

Simulación Estructurada del Filtro de Kalman para Identificación de Estados en un Motor de C.C. Sobre una Computadora Embebida

José Luis Cano Rosas, Gustavo Delgado Reyes, Pedro GuevaraLópez

Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del Instituto Politécnico Nacional
Av. Santa Ana No. 1000, Col. San Francisco Culhuacan, Deleg. Coyoacán, D.F., México, CP 04260.
lucskyr@gmail.com, dengue_mgs4@hotmail.com, pguevara@ipn.mx

2014 Published by *DIFU*_{100ci}@ <http://nautilus.uaz.edu.mx/difu100cia>

Resumen

El Filtro de Kalman es un algoritmo identificador de estados que minimiza los ruidos de un sistema, esto motivó al desarrollo del presente trabajo, donde se presenta una simulación estructurada del Filtro de Kalman. La programación se hizo en una computadora embebida Raspberry Pi, ésta cuenta con el sistema operativo Raspbian y un compilador de lenguaje C que permite la implementación de algoritmos iterativos y recursivos. Como ejemplo de estudio se simula el modelo matemático de un motor de C.C. y se identifican sus estados a través del Filtro de Kalman, para ello se programaron una serie de Shell Scripts desarrollados como librerías para realizar operaciones matriciales y algunas funciones para análisis estadístico. Para presentar las gráficas de los resultados, los algoritmos de la computadora embebida almacenan los datos de las respuestas obtenidas de los estados identificados y errores cuadráticos medios para finalmente graficarse con Matlab.

Palabras clave: Embebido, Filtro de Kalman, Motor, Raspberry Pi, Simulación.

1. Introducción

De acuerdo a [1], un filtro digital es un dispositivo que elimina, extrae, predice, reconstruye y describe parte de la información de un sistema, de acuerdo a un criterio previamente establecido. Con base en esto en [2] y [3], el Filtro de Kalman se describe como un algoritmo basado en el modelo de espacio de estados de un sistema para estimar el estado futuro y la

salida futura realizando un filtrado óptimo a la señal de salida y dependiendo del retraso de las muestras que se le ingresan puede cumplir la función de estimador de parámetros o identificador de estados, en ambos casos se obtiene la disminución de ruido [4] y [5]. El Filtro de Kalman tiene muchas aplicaciones prácticas, una de ellas es la disminución de ruidos gaussianos en motores eléctricos de corriente continua. La construcción de un motor eléctrico se basa en dos partes que se deno-

minan campo y armadura; en ambas partes, llevan un devanado que al ser excitado con energía eléctrica se crean dos campos magnéticos que al acoplarse desarrolla un par de fuerzas que dan origen a la rotación, creando así un movimiento en el rotor por lo que esta energía mecánica se manifiesta en la flecha del motor. [6]

Por otra parte, un Sistema Embebido es un término asociado a los circuitos electrónicos que contienen diferentes componentes o subcircuitos encapsulados en un solo dispositivo, como ejemplo los microprocesadores, microcontroladores, arreglo de compuertas, etc. Una gran variedad de organizaciones han diseñado diferentes tarjetas electrónicas de desarrollo que permiten interactuar con los sistemas embebidos, por mencionar algunas están DE0 de Terasic, Arduino Mega 2560, Freescale y Raspberry Pi, este último por su practicidad es el sistema en el cual se desarrolla el presente trabajo, sirviendo como plataforma para la implantación del filtro de Kalman.

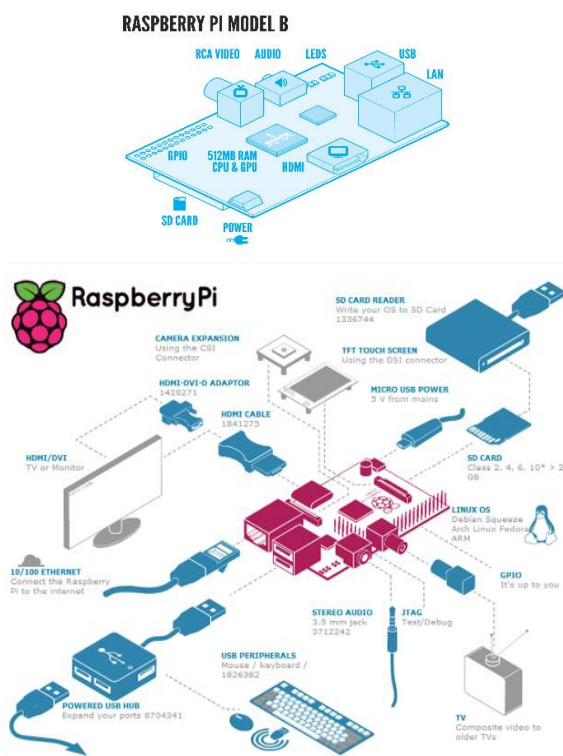


Figura 1. Esquema del sistema embebido y su conectividad.[8]

Una Raspberry Pi [8] (ver figura 1) es una computadora embebida con arquitectura system on a chip, que integra la mayoría de los componentes en un solo circuito integrado. Esta tarjeta de desarrollo está compuesta con un chip BCM2835 que contiene un procesador ARM1176JZF-S de 700MHz (el cual se puede someter a overclock), procesador gráfico VideoCore IV y 512 MB

de memoria RAM. Es preciso mencionar que existen diferentes aplicaciones del filtro de Kalman en sistemas embebidos diversos, e inclusive en software de programación y análisis como lo son Labview y Matlab, el trabajo que presenta este artículo tiene como uno de los principales aportes, el desarrollo de librerías que realizan el algoritmo del Filtro de Kalman en Raspberry Pi.

2. Filtro de Kalman como Identificador de Estados

2.1. Esquema del Filtro de Kalman para identificación de estados

El Filtro de Kalman opera bajo un esquema observador para identificación de estados como el que se aprecia en la figura 2.

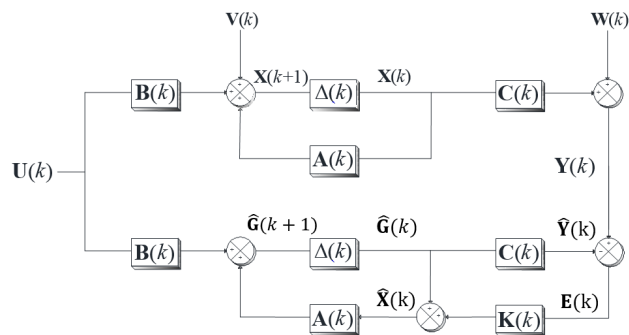


Figura 2. Filtro de Kalman para identificación de estados.

2.2. Estructura de la simulación del Filtro de Kalman para identificación de estados

Donde para la identificación de los estados del sistema es necesario conocer las matrices propias del sistema dinámico $A(k)$, $B(k)$, $C(k)$ y $D(k)$, los vectores de ruidos agregados $V(k)$ y $W(k)$ y las matrices de covarianza $Q(k)$ y $R(k)$, de esta manera para utilizar el filtro de Kalman se aplica procedimiento mostrado en la figura 3.

3. Modelado del Motor Eléctrico de Corriente Continua

Para el desarrollo del modelo matemático del motor se partió del circuito electromecánico mostrado en la figura 4. La descripción de cada una de las variables mostradas en el circuito de la figura anterior se muestra en la tabla 1.

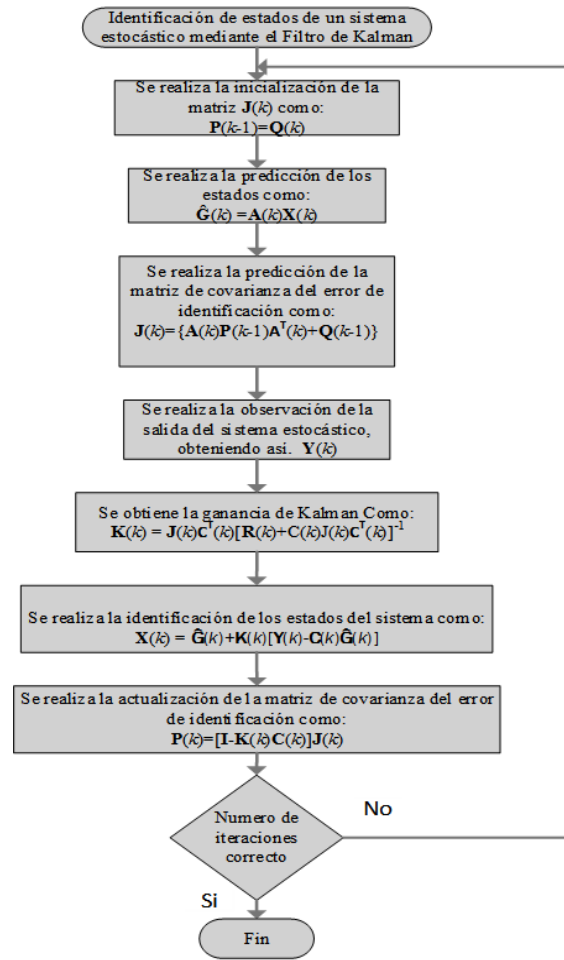


Figura 3. Algoritmo para identificación de estados mediante el Filtro de Kalman.

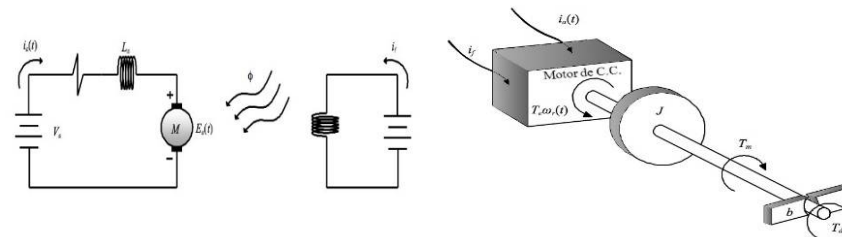


Figura 4. Diagrama Eléctrico del motor de C.C. y Diagrama mecánico del motor de C.C.

Tabla 1. Variables del motor de C.C

Variable	Descripción	Unidades
R_a	Resistencia de la armadura	Ω
L_a	Inductancia de la armadura	mH
i_a	Corriente de la armadura	A
i_f	Corriente del campo	A
V_a	Tensión aplicada a la armadura	V
ℓ_b	Fuerza contraelectromotriz	V
W_r	Velocidad angular del eje del motor	rad/seg
T	Par desarrollado por el motor	Nm
J	Momento de inercia	kgm^2
b	Coefficiente de fricción viscosa	$\frac{Nms}{rad}$

Las ecuaciones (1) y (2) representan el modelo del motor en espacio de estados en el dominio del tiempo continuo:

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{A}\mathbf{X}(t) + \mathbf{B}\mathbf{U}(t) + \mathbf{V}(t) \quad (1)$$

$$\mathbf{Y}(t) = \mathbf{C}\mathbf{X}(t) + \mathbf{W}(t) \quad (2)$$

Donde:

$$\mathbf{A} = \begin{bmatrix} \frac{-R_a}{L_a} & \frac{-K_a i_f}{L_a} \\ \frac{-K_a i_f}{J} & \frac{-b}{J} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{1}{J} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{X}(t) = \begin{bmatrix} i_a(t) \\ \omega_r(t) \end{bmatrix} \quad \mathbf{U}(t) = \begin{bmatrix} V_a(t) \\ T_d(t) \end{bmatrix}$$

$$\mathbf{V}(t) = \begin{bmatrix} V_1(t) \\ V_2(t) \end{bmatrix} \quad \mathbf{W}(t) = \begin{bmatrix} W_1(t) \\ W_2(t) \end{bmatrix}$$

4. Identificación de los Estados del Motor de Corriente Continua Utilizando el Filtro de Kalman

El Filtro de Kalman se ha discretizado por la ecuaciones (1) y (2) con un periodo de muestreo de 10 milisegundos, Obteniéndose el sistema en espacio de estados en el dominio del tiempo discreto que se observa en (3) y (4):

$$\mathbf{X}(k+1) = \begin{bmatrix} -0.0001 & -0.3291 \\ 0.0003 & 0.9869 \end{bmatrix} \mathbf{X}(k) + \dots$$

$$\dots + \begin{bmatrix} -0.3323 & 0.0032 \\ 0.0032 & -0.0099 \end{bmatrix} \begin{bmatrix} V_a(k) \\ T_d(k) \end{bmatrix} \quad (3)$$

$$\mathbf{Y}(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{X}(k) + \begin{bmatrix} 5.7600 & 0 \\ 0 & 5.7600 \end{bmatrix} \mathbf{W}(k) \quad (4)$$

En la ecuación (3) y (4) se aprecian las matrices de covarianza **Q** y **R**, las cuales se define en (5) y (7):

$$\mathbf{Q} = \begin{bmatrix} \sigma_V(k)\sigma_V(k) & 0 \\ 0 & \sigma_V(k)\sigma_V(k) \end{bmatrix}$$

$$= \begin{bmatrix} 0.0196 & 0 \\ 0 & 0.0196 \end{bmatrix} \quad (5)$$

Al analizar la ecuación (5) se ve que cada uno de los elementos de la matriz define el momento de covarianza de **V(k)**, éste es igual al segundo momento central del ruido agregado a la entrada del sistema; por lo que en (6) se establece el intervalo en el cual puede ser

evaluado el momento de covarianza para dos variables aleatorias es decir:

$$0 \leq k_{xy} \leq \sigma_y \sigma_x \quad (6)$$

De acuerdo a la ecuación (4) la matriz **R** se define como:

$$\mathbf{R} = \begin{bmatrix} \sigma_W(k)\sigma_W(k) & 0 \\ 0 & \sigma_W(k)\sigma_W(k) \end{bmatrix}$$

$$= \begin{bmatrix} 5.7600 & 0 \\ 0 & 5.7600 \end{bmatrix} \quad (7)$$

La matriz **R** define el momento de covarianza de **W(k)**.

5. El Filtro de Kalman en el Sistema Embebido Implementado en la Computadora Raspberry Pi

Para la implementación del Filtro de Kalman y la simulación del motor en la computadora embebida, se utilizó el sistema operativo Raspbian, diseñando una biblioteca de desarrollo propio llamada matrices.h, esta se muestra en la figura 5.

```
void imprimir_matriz(double**, int, int);
void imprimir_vector(double*, int);
double **producto_matriz(double**, double**, int, int, int);
double **iniciar_matriz(int, int, double);
double **inversa(double**, int);
double **matriz_identidad(int);
double **traspuesta(double**, int, int);
double **llenar_matriz(int, int);
double **sumar_matriz(double **, double **, int, int);
double **restar_matriz(double **, double **, int, int);
double *iniciar_vector(int, float);
double **pseudo_inversa(double**, int, int);
double **perturbacion(double**, int, int, double);
double **matriz_igual(double **b, double **a, m, n);
double **valor_inicial(double **a, int m, int n, double x);
```

Figura 5. Funciones para operaciones matriciales utilizadas por el filtro de Kalman.

Con las funciones de la figura 5 se realiza la simulación del sistema dinámico, obteniendo **Y(k)**. La entrada restante al filtro se obtiene al agregar ruidos a **U(k)**. El algoritmo para programar el Filtro de Kalman se describe mediante el diagrama de flujo que de manera simplificada se observa en la figura 3, donde se describen las operaciones que permiten la identificación y filtrado, al compilarse y ejecutarse permiten la obtención de

los resultados experimentales que se muestran en la siguiente sección.

6. Resultados Experimentales

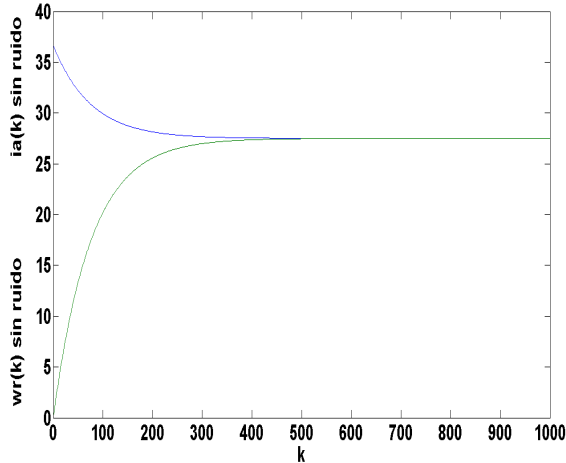


Figura 6. Gráfica de $X(k)$ sin ruidos (respuesta ideal del sistema).

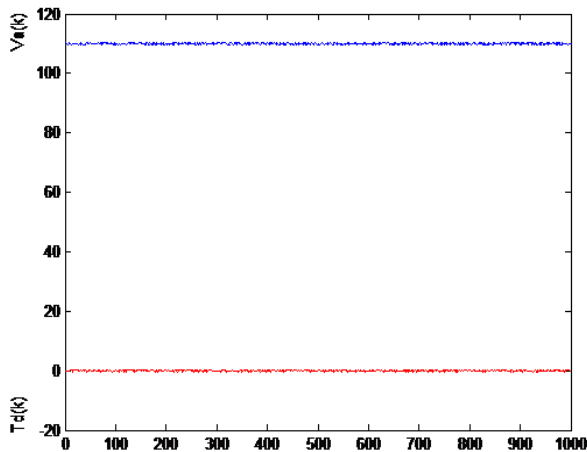


Figura 7. Gráfica de $U(k)$, donde $V_a(k)$ es la curva superior y $T(k)$ corresponde a la gráfica inferior para 1000 iteraciones.

Para obtener los resultados experimentales de las figuras 6, 7, 8, 9, 10 y 11 en la Raspberry Pi, se programó la simulación en el sistema operativo Raspbian y se permite la escritura de los datos en una localidad del disco duro, conformándose vectores en formato de texto que son graficados en Matlab.

Los errores cuadráticos medios han sido obtenidos en cada iteración del algoritmo programado mediante el cálculo de la esperanza matemática del error que existe entre las respuestas del sistema ideal y las respuestas $Y(k)$ (las cuales contienen ruido y no han sido filtradas)

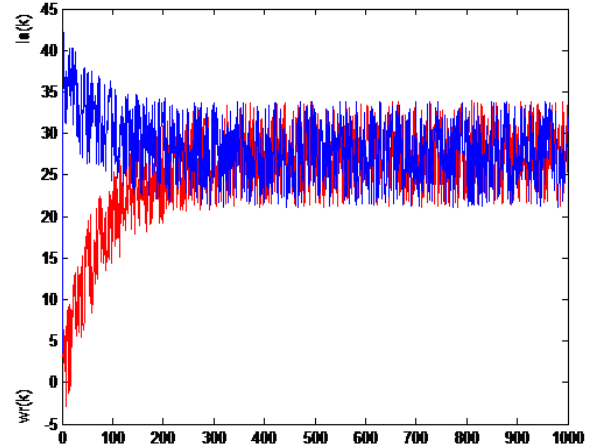


Figura 8. Gráfica de $Y(k)$, donde $i_a(k)$ con ruido es la curva superior y $W_r(k)$ con ruido corresponde a la gráfica inferior para 1000 iteraciones.

para el caso de la figura 10, y el error existente entre la salida del sistema ideal y los estados identificados (los cuales presentan una disminución considerable en los ruidos ver figura 9 producto del filtrado al que han sido sometidos) para el caso de la figura 11, mediante un análisis a estas figuras se establece que a medida que se logra la disminución del ruido en las respuestas del sistema, los errores cuadráticos convergen de manera asintótica a valores cada vez más cercanos a 0; sin embargo cuando el filtrado no es adecuado esta convergencia no tiene como límite un valor cercano a 0 (como se aprecia en la figura 10), es de esta manera como se validan los resultados de filtrado obtenidos en el presente trabajo, en el que se ha demostrado que los errores cuadráticos obtenidos para los estados identificados convergen a valores muy pequeños (0.025), en tanto que los obtenidos para las respuestas sin filtrar lo hacen a valores distantes de 0 (14.0), observe las figuras 10 y 11.

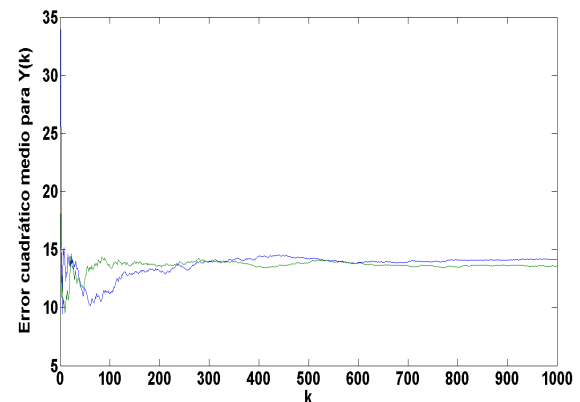


Figura 10. Gráfica de error cuadrático medio de $Y(k)$.

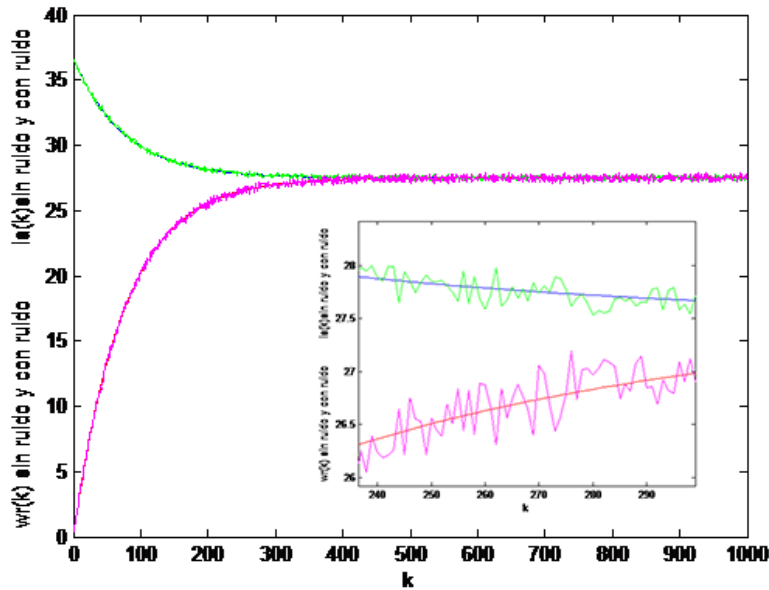


Figura 9. Gráfica de $X(k)$ y $\hat{X}(k)$ correspondiente a 1000 iteraciones.

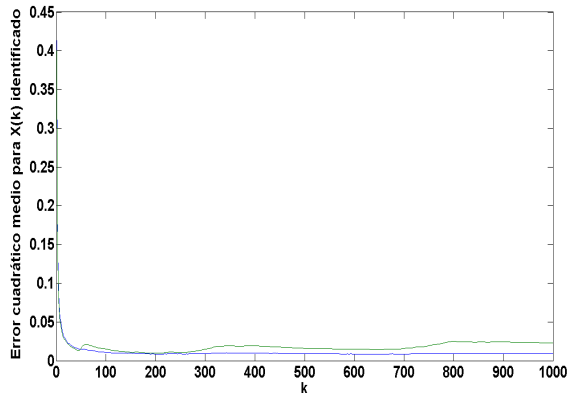


Figura 11. Gráfica de error cuadrático medio de la identificación de $X(k)$.

7. Conclusiones

Los sistemas dinámicos pueden modelarse a través de sus estados observables y estados internos, los primeros siempre son conocidos, mientras que los segundos tienen que identificarse para conocer la dinámica completa del sistema; una herramienta muy poderosa para identificación es el Filtro de Kalman, que además de identificar estados internos, elimina el ruido.

Para la implementación de la simulación del sistema en estudio y del Filtro de Kalman en una computadora embebida es necesario expresar el modelo en forma discreta y recursiva con el fin de obtener un algoritmo

de baja complejidad computacional; en este sentido, la propuesta fue utilizar una computadora embebida Raspberry Pi para la simulación del Filtro de Kalman utilizado en la identificación de estados de un motor de corriente continua. Cabe mencionar que se hizo una búsqueda exhaustiva en la literatura y no se encontraron trabajos que implementen el Filtro de Kalman en una computadora Raspberry Pi.

Los algoritmos fueron desarrollados en Lenguaje C, compilados y ejecutados en Raspbian, para ello se desarrolló una biblioteca con diferentes funciones matriciales para la implementación del motor eléctrico y del Filtro. Los resultados de la simulación se observan en las figuras 6 a 9; se aprecia que la identificación de estados es muy buena así como la eliminación de ruidos, para ello se presentan las gráficas de las figuras 10 y 11 donde se observa su calidad a través de las magnitudes de los errores cuadráticos medios.

Referencias

- [1] J. Abonyi. "Fuzzy model identification for control 1 st ed (Birk hauser Boston 2003). P. 19."
- [2] J. G. Díaz, A. M. Mejía y F. Arteaga. "Aplicación del filtro de Kalman a sistemas de Control," 2009.
- [3] Haykin S. "Cognition is the Key to the Next Generation of Radar Systems," 2009.
- [4] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". Transactions of the ASME-Journal of Basic Engineering, 82. Series D, 1960, P.35.
- [5] J.J. Medel Juárez "M.T. Zagaceta Álvarez, Estimación- identificación como filtro digital integrado: descripción e im-

- plementación recursiva, Revista Mexicana de Física No. 56,” 2010.
- [6] J. S. Valdez Martinez. (2008), “Modelado discreto, Simulación, y Control de un Motor de un motor de corriente continua tipo serie considerando pérdidas. ”. Mexico D.F.
- [7] Gustafsson F. “Adaptive filtering and change detection”. Ed. John Wiley and Sons , LTD, Linkoping University Sweden, 2000.
- [8] Rapsberry PI. <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/> ,Septiembre 2014.