

# SOFTWARE PARA EL DISEÑO DE FILTROS SALLEN-KEY

## Software for the Design of Sallen-Key Filters

### RESUMEN

En este documento se presentan los resultados obtenidos al realizar el desarrollo de una interfaz gráfica para el diseño de filtros activos tipo Sallen-Key, en las configuraciones pasa bajas, pasa altas, pasa banda y rechazo de banda; implementada en el entorno de desarrollo integrado *Matlab*<sup>®</sup>. También se muestran las simulaciones de algunos circuitos diseñados en la herramienta *Proteus*<sup>®</sup>, con el fin de verificar que la respuesta en frecuencia obtenida concuerde con la arrojada por el algoritmo.

**PALABRAS CLAVES:** Amplificador operacional, diagrama de Bode, filtro, función de transferencia, ganancia, respuesta en frecuencia.

### ABSTRACT

*This paper presents the results obtained to carry out the development of software for Sallen-Key active filter design in the settings low pass, high pass, band pass and band rejection; implemented in Matlab<sup>®</sup> integrated development environment. Also shown are simulations of some circuits designed Proteus<sup>®</sup> tool, to verify that the frequency response obtained is consistent with that from the algorithm.*

**KEYWORDS:** Operational amplifier, Bode diagram, filter, transfer function, gain, frequency response.

## 1. INTRODUCCIÓN

Desde los comienzos de las telecomunicaciones electrónicas se observó la necesidad de enviar información procedente de distintas fuentes sobre un mismo medio de transmisión, bien sea a través de una red eléctrica (comunicación alámbrica), o el aire (comunicación inalámbrica); con lo cual se hace impetuosa la creación de estrategias que permitan separar las señales recibidas por el receptor.

El primer inconveniente, consistente en la transmisión de datos de distintas procedencias en un mismo medio, se suplió al realizar el envío de estos en diferentes frecuencias mediante la aplicación de alguna técnica de modulación. En cuanto a lo que tiene que ver con el transmisor, es posible afirmar que resulta sencilla la solución. En el sistema de recepción, es necesario idear circuitos capaces de elegir la información deseada y eliminar el ruido que se le puede inducir por campos magnéticos o eléctricos a los cuales se encuentran expuestos los medios de transmisión. Este proceso se conoce como filtrado y el subsistema que realiza esta tarea es el filtro [1].

Los primeros filtros se basaron en los modelos  $\tau$  y  $\pi$ , apoyados en desarrollos meramente teóricos. Estos estaban conformados por componentes pasivos tales

como resistencias, condensadores e inductores (comúnmente llamados filtros pasivos), y se

caracterizaban por que no necesitaban fuentes externas para su funcionamiento, es decir, que no agregan potencia a las señales filtradas. Aunque estos filtros ofrecen una alta confiabilidad, no son la mejor solución, ya que los componentes son relativamente grandes y difíciles de adquirir comercialmente [2].

Debido al gran avance en la electrónica, los filtros pasivos fueron reemplazados por filtros activos, los cuales se basan en amplificadores operacionales. Estos filtros tienen como característica fundamental la posibilidad de amplificar la señal filtrada (aumento de la ganancia en la banda pasante), además de una mejor precisión en la frecuencia de corte, un menor tamaño, y la abolición de la necesidad de involucrar los incómodos inductores [3] [4].

Este trabajo contiene los resultados obtenidos al desarrollar una interfaz gráfica en la utilidad GUIDE (*Graphical User Interface Development Environment*) de *Matlab*<sup>®</sup>, que permite diseñar filtros activos de segundo orden en configuración pasa bajas, pasa altas, pasa banda y rechazo de banda; utilizando para ello redes tipo *Sallen-Key*. El programa también permite obtener la función de transferencia del sistema dinámico diseñado, además del diagrama de Bode (respuesta en frecuencia) del filtro, el cual puede ser comparado con la simulación del circuito arrojado al ejecutar el algoritmo,

### ARLEY BEJARANO

Estudiante Ingeniería Electrónica  
Universidad Tecnológica de Pereira  
[arleybejarano@gmail.com](mailto:arleybejarano@gmail.com)

### OSCAR ANDRES AYALA

Estudiante Ingeniería Electrónica  
Universidad Tecnológica de Pereira  
[oaayala@utp.edu.co](mailto:oaayala@utp.edu.co)

### EDWIN ANDRÉS QUINTERO

Ingeniero Electrónico  
Candidato a Magíster en  
Instrumentación Física  
Profesor Asistente  
Universidad Tecnológica de Pereira  
[equintero@utp.edu.co](mailto:equintero@utp.edu.co)

con el fin de verificar el correcto funcionamiento del mismo.

## 2. FILTROS ACTIVOS SALLEN KEY

Las redes *Sallen-Key* son un tipo de filtros electrónicos particularmente valiosos por su simplicidad. Para obtener un filtro de orden mayor se pueden poner en cascada varias etapas [5].

Para su implementación se utilizan filtros estándar que permiten diseñar de forma rápida y mecánica. Teniendo en cuenta el diseño de las impedancias de las redes *Sallen-Key*, es posible obtener filtros pasa bajos o pasa altos de dos polos usando dos resistencias, dos condensadores y un amplificador. La figura 1 contiene la representación general de las redes *Sallen-Key* [6].

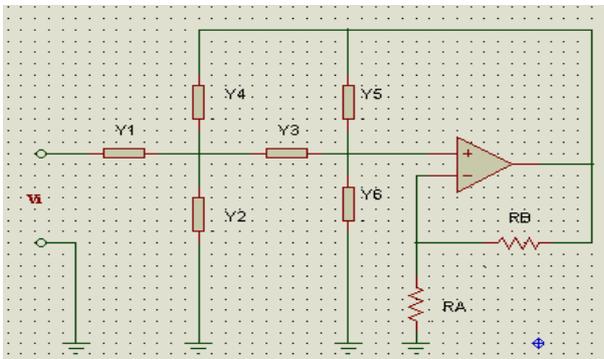


Figura 1. Estructura de una red *Sallen-Key*.

A continuación es posible reemplazar parte de la estructura que conecta a  $R_a$  y  $R_b$  (figura 2) por la que se muestra en la figura 3, esto con el fin de facilitar el análisis matemático.

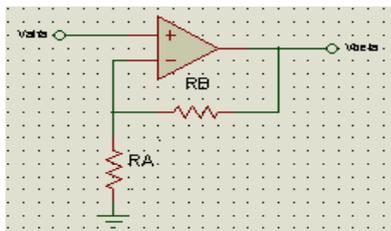


Figura 2. Red de amplificación no inversora.

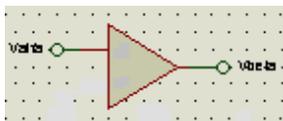


Figura 3. Etapa de amplificación de ganancia  $\mu$ .

Por lo tanto, de las figuras 2 y 3 se tiene que:

$$V_B = \left(1 + \frac{R_B}{R_A}\right) * V_A \quad (1)$$

$$\mu = 1 + \frac{R_B}{R_A} \quad (2)$$

La figura 4 contiene el circuito final obtenido al reescribir el circuito en términos de la red de ganancia  $\mu$ .

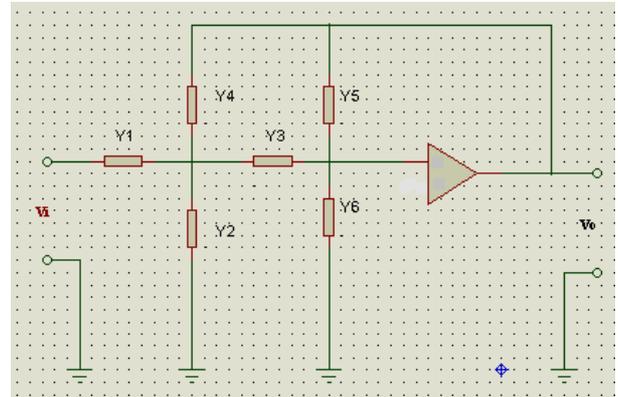


Figura 4. Red *Sallen-Key* con etapa de amplificación de ganancia  $\mu$ .

Construyendo la matriz definida de admitancias para la red de la figura 4 se obtiene:

$$\begin{bmatrix} i_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} Y_1 & -Y_1 & 0 & 0 \\ -Y_1 & Y\alpha_1 & -Y_3 & 0 \\ 0 & -Y_3 & Y\alpha_2 & -Y_5 \\ 0 & -Y_4 & -Y_4 & Y\alpha_3 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ uV_3 \end{bmatrix}$$

Donde:

$$\begin{aligned} Y\alpha_1 &= Y_1 + Y_2 + Y_3 + Y_4 \\ Y\alpha_2 &= Y_3 + Y_5 + Y_6 \\ Y\alpha_3 &= Y_4 + Y_6 \\ \mu V_3 &= V_4 \end{aligned}$$

La ganancia de voltaje está determinada por:

$$A_v = \frac{V_0}{V_i} = \mu \frac{V_B}{V_1} \quad (3)$$

Ahora, desarrollando el análisis matricial respectivo y sabiendo que:

$$\begin{aligned} Y_1 &= \frac{1}{R_1}; \quad Y_2 = \frac{1}{\infty}; \quad Y_3 = \frac{1}{R_3}; \quad Y_5 = \frac{1}{\infty}; \quad Y_4 = C_4 * S; \\ Y_6 &= C_6 * S; \quad R_1 = R_3 = R; \quad C_4 = C_6 = C \end{aligned}$$

Se obtiene la ganancia del filtro:

$$A_v = \mu \frac{1}{C^2 R^2} \frac{1}{S^2 + \left[\frac{3}{RC} - \frac{\mu}{RC}\right] S + \frac{1}{C^2 R^2}} \quad (4)$$

Comparando con la función de transferencia de un filtro pasa bajas de segundo orden:

$$G(s) = \frac{W_c^2}{s^2 + \frac{W_c}{Q} s + W_c^2} \quad (5)$$

Se obtiene que:

$$W_c = \frac{1}{RC} ; \rightarrow \text{Frecuencia de corte del filtro}$$

$$\mu = 3 - \frac{1}{Q} ; \rightarrow \text{Factor de calidad o ganancia del circuito Sallen - key.}$$

La topología de los filtros pasa alto Sallen-Key es reciproca a los de pasa bajo, es decir, las resistencias estarán donde estaban los condensadores y viceversa.

Para construir filtros pasa banda a partir de redes Sallen-Key, basta con conectar un filtro pasa bajo cuya frecuencia de corte  $\omega_c$  corresponda a la frecuencia superior del pasa banda  $\omega_H$ ; en cascada con un paso alto en el cual la frecuencia de corte coincide con la frecuencia inferior del selector de banda  $\omega_L$ .

En el caso de los filtros de rechazo de banda, es suficiente con sumar las salidas de los filtros pasa bajas y pasa altas, configurados a la frecuencia inferior  $\omega_L$  y superior  $\omega_H$  respectivamente.

### 3. PROGRAMA DESARROLLADO

La figura 5 contiene una porción del código fuente, mientras la figura 6 presenta el diagrama de flujo del algoritmo elaborado.

```

function pushbutton1_Callback(hObject, eventdata, handles)
    mu=1.55;
    if and(handles.opcion==2,handles.opcion1==2)==1
        fc=str2num(get(handles.edit6,'String'));
        c=str2num(get(handles.edit5,'String'));
        c=c*(1e-6);
        wc=2*pi*fc;
        z=1/(c*wc);
        set(handles.text1,'Visible','on');
        set(handles.edit1,'Visible','on');
        set(handles.edit1,'String',z);
        Av=fc*(1e-2),[1 (3-mu)/(1/wc) wc^2];
        axes(handles.axes2)
        bodemag(Av)
    end
    if and(handles.opcion==2,handles.opcion1==3)==1
        c=str2num(get(handles.edit5,'String'));
        c=c*(1e-6);
        fc=str2num(get(handles.edit1,'String'));
        wc=1/(c*c);
        wc=2*pi*fc;
        z=1/(c*wc);
        set(handles.text6,'Visible','on');
        set(handles.edit6,'Visible','on');
        set(handles.edit6,'String',fc);
        Av=fc*(1e-2),[1 (3-mu)/(1/wc) wc^2];
        axes(handles.axes2)
        bodemag(Av)
    end
    if and(handles.opcion==3,handles.opcion1==2)==1
        c=str2num(get(handles.edit5,'String'));
        c=c*(1e-6);
        fc=str2num(get(handles.edit6,'String'));
        wc=1/(c*c);
        wc=2*pi*fc;
        z=1/(c*wc);
        set(handles.text6,'Visible','on');
        set(handles.edit6,'Visible','on');
        set(handles.edit6,'String',fc);
        Av=fc*(1e-2),[1 (3-mu)/(1/wc) wc^2];
        axes(handles.axes2)
        bodemag(Av)
    end
    if and(handles.opcion==4,handles.opcion1==2)==1
        fc=str2num(get(handles.edit7,'String'));
        c=str2num(get(handles.edit5,'String'));
        c=c*(1e-6);
        wc=2*pi*fc;
        z=1/(c*wc);
        set(handles.text1,'Visible','on');
        set(handles.edit1,'Visible','on');
        set(handles.edit1,'String',z);
        Av=fc*(1e-2),[1 (3-mu)/(1/wc) wc^2];
        %Diseño Pasa alto
        fcH=str2num(get(handles.edit8,'String'));
        cH=str2num(get(handles.edit9,'String'));
        cH=cH*(1e-6);
        wcH=2*pi*fcH;
        zH=1/(cH*wcH);
        set(handles.text7,'Visible','on');
        set(handles.edit7,'Visible','on');
        set(handles.edit7,'String',zH);
        AvH=fcH*(1e-2),[1 (3-mu)/(1/wcH) wcH^2];
        %Diseño Pasa bajo
        fcL=str2num(get(handles.edit7,'String'));
        cL=str2num(get(handles.edit8,'String'));
        cL=cL*(1e-6);
        wcL=2*pi*fcL;
        zL=1/(cL*wcL);
        set(handles.text2,'Visible','on');
        set(handles.edit2,'Visible','on');
        set(handles.edit2,'String',zL);
        AvL=fcL*(1e-2),[1 (3-mu)/(1/wcL) wcL^2];
        %Filtro pasa banda
        Av=AvL+AvH;
        axes(handles.axes2)
        bodemag(Av)
    end
    if and(handles.opcion==4,handles.opcion1==3)==1
        %Respuesta en frecuencia pasa bajo
        c=str2num(get(handles.edit5,'String'));
        c=c*(1e-6);
        fc=str2num(get(handles.edit1,'String'));
        wc=1/(c*c);
        wc=2*pi*fc;
        z=1/(c*wc);
        set(handles.text7,'Visible','on');
        set(handles.edit7,'Visible','on');
        set(handles.edit7,'String',fc);
        Av=fc*(1e-2),[1 (3-mu)/(1/wc) wc^2];
        %Respuesta en frecuencia pasa alto
        fcH=str2num(get(handles.edit8,'String'));
        cH=str2num(get(handles.edit9,'String'));
        cH=cH*(1e-6);
        wcH=2*pi*fcH;
        zH=1/(cH*wcH);
        set(handles.text8,'Visible','on');
        set(handles.edit8,'Visible','on');
        set(handles.edit8,'String',fcH);
        AvH=fcH*(1e-2),[1 (3-mu)/(1/wcH) wcH^2];
        %Respuesta Total
        Av=AvL+AvH;
        axes(handles.axes2)
        bodemag(Av)
    end
    if and(handles.opcion==6,handles.opcion1==2)==1
        fc=str2num(get(handles.edit8,'String'));
    end
    
```

Figura 5. Código fuente del software para el diseño de filtros activos de segundo orden.

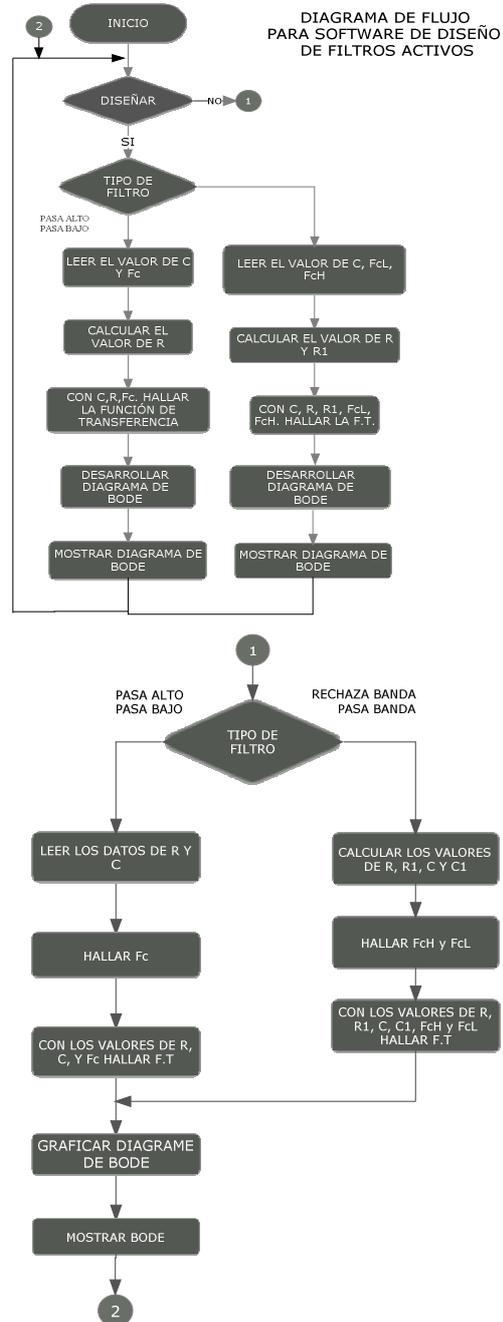


Figura 6. Diagrama de flujo del programa desarrollado.

Inicialmente el usuario elige el tipo de filtro que desea diseñar. Posteriormente ingresa el valor del condensador a utilizar y las frecuencias de corte del filtro. El programa procede a visualizar el circuito final y el diagrama de Bode del filtro, gracias a lo cual es posible verificar el correcto funcionamiento de la red. Las figuras 7 a 11 muestran las diferentes ventanas que ofrece la interfaz gráfica.



El diagrama de bode para el filtro diseñado puede observarse en las figuras 14 y 15. Nótese las frecuencias inferior (1kHz) y superior (9kHz).

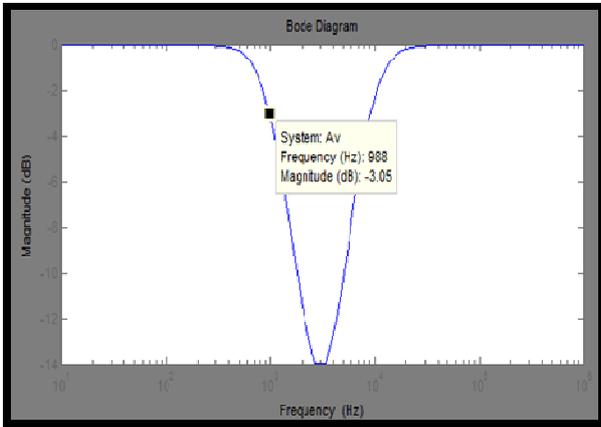


Figura 14. Frecuencia inferior para el filtro desarrollado.

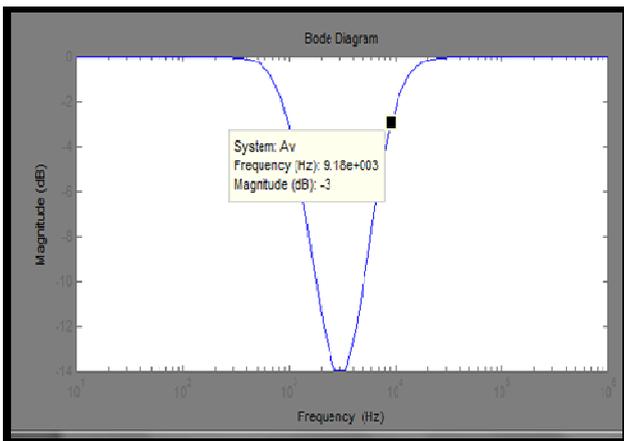


Figura 15. Frecuencia superior para el filtro desarrollado.

El circuito arrojado por la interfaz gráfica fue simulado en la herramienta de simulación de circuitos electrónicos *Proteus*<sup>®</sup>, con el objetivo de verificar el correcto funcionamiento del mismo (ver figura 16). La figura 17 contiene la respuesta en frecuencia del circuito de la figura 16 obtenida en la herramienta informática anteriormente mencionada. Nótese la similitud que presenta esta gráfica con las mostradas en las figuras 14 y 15.

Además de lo expuesto anteriormente, el programa también permite realizar el análisis de filtros activos de segundo orden seleccionando la opción “*Observar Respuesta*”, dentro del menú “*Diseñar/Respuesta en Frecuencia*”. En este ítem el usuario puede seleccionar el tipo de sistema dinámico que desea analizar, ingresar el valor de los condensadores y resistencias que conforman el circuito, para finalmente observar el diagrama de Bode

(respuesta en frecuencia) de la red de interés arrojada por la interfaz gráfica.

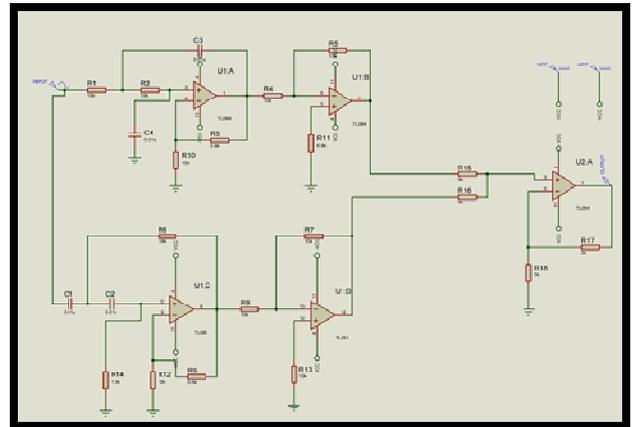


Figura 16. Filtro Elimina-bandas en la herramienta de simulación *Proteus*<sup>®</sup>.

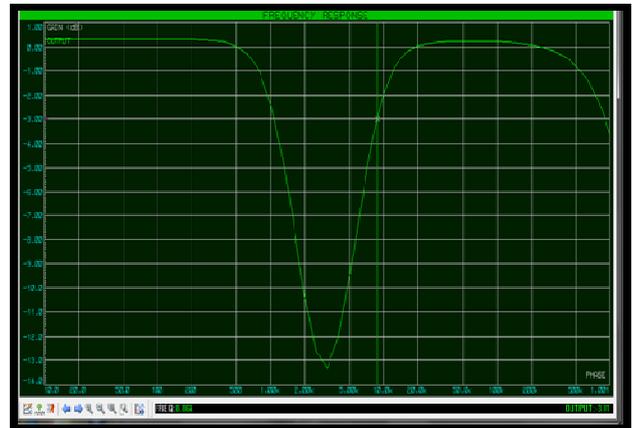


Figura 17. Respuesta en frecuencia del Filtro Elimina-bandas obtenida en *Proteus*<sup>®</sup>.

## 5. CONCLUSIONES

La facilidad de implementación de las redes *Sallen-Key* permite establecer algoritmos que sistematicen el diseño de los mismos. Además, gracias a la teoría de filtros, es posible pasar rápidamente de un pasa bajas a un pasa altas al obtener el complemento del primero; a un pasa banda mediante la conexión en cascada de un pasa bajas y un pasa altas; y a un rechazo de banda a partir de la suma en paralelo de los dos filtros elementales [7].

A pesar de que los filtros *Sallen-Key* ofrecen simplicidad a la hora del diseño, estos requieren de una etapa posterior que normalice la ganancia en la banda pasante, pues el factor de calidad deseado (0.707 generalmente) obliga a la utilización de cierta ganancia mayor que uno para la red de amplificación no inversora (ver figuras 2 y 3). Esto significa que no es posible ajustar dentro del filtro mismo el valor de ganancia para la banda de paso según el gusto del usuario, sino que esta debe ser

asignada a partir de una red adicional, lo cual, naturalmente, eleva los costos involucrados en el desarrollo del sistema.

## 6. BIBLIOGRAFÍA

- [1] José E. López, *Síntesis de Redes: Impedancias y Filtros*, I Edición, Delta Publicaciones, 2008.
- [2] Adel S. Sedra, *Circuitos Microelectrónicos*, IV Edición, Oxford University Press, 1998.
- [3] Robert B. Northrop, *Analog Electronic Circuits*, I Edición, Addison-Wesley Publishing Company, 1990.
- [4] Muhammad H. Rashid, *Circuitos Microelectrónicos: Análisis y Diseño*, I Edición, International Thomson Editors, 2000.
- [5] Robert Spence, *Tolerance Desing of Electronic Circuits*, I Edición, Imperial College Press, 1988.
- [6] Luis E. Avendaño, *Sistemas Electrónicos Analógicos: Un Enfoque Matricial*, III Edición, Centro de Publicaciones Universidad Tecnológica de Pereira.
- [7] C. J. Savant Jr., *Diseño Electrónico: Circuitos y Sistemas*, III Edición, Pearson Education, 1997.