

APLICACIÓN DE LAS CURVAS DE BÉZIER EN LA GENERACIÓN DE UNA SEÑAL ELÉCTRICA

Juan Carlos Pérez Merlos

Universidad Autónoma del Estado de
México, Facultad de Ingeniería
Toluca - Estado de México

Mireya Salgado Gallegos

Universidad Autónoma del Estado de
México, Facultad de Ingeniería
Toluca - Estado de México

Eduardo Rodríguez Ángeles

Universidad Autónoma del Estado de
México, Facultad de Ingeniería
Toluca - Estado de México

Silvia Edith Albarrán Trujillo

Universidad Autónoma del Estado de
México, Facultad de Ingeniería
Toluca - Estado de México

All content in this magazine is licensed under a Creative Commons Attribution License. Attribution-Non-Commercial-Non-Derivatives 4.0 International (CC BY-NC-ND 4.0).



Resumen: Los polinomios de Bernstein son el fundamento de las curvas de Bézier, éstas tienen una amplia aplicación en la generación de figuras geométricas complejas y son base para el desarrollo de programas dedicados a la generación de dibujos, así como también para la robótica móvil y videojuegos, especialmente en la planeación de trayectorias. En este trabajo se hace uso de las curvas de Bézier para generar señales eléctricas mediante generadores de señales en equipos que proporcionan diferentes formas de onda (senoidal, pulsos o escalón unitario y otras funciones periódicas), las cuales son utilizadas para inyectarlas a circuitos eléctricos con la finalidad de ver su respuesta en tiempo o en frecuencia a través de un osciloscopio. Lo anterior se logró a través de la programación de los algoritmos de la curva de Bézier y la interfaz de usuario en Labview, así como la simulación en Matlab de estas curvas. Para la generación física de la señal se usó el puerto serie entre una computadora y un microcontrolador. El resultado obtenido mostró la factibilidad de que la elaboración de señales eléctricas físicas mediante este medio puede sustituir a un generador de señales de equipo electrónico.

Palabras clave: Curvas de Bézier, generación de una señal eléctrica, aplicación con arduino, pwm.

INTRODUCCIÓN

Antes que los gráficos por computadora existieran, los ingenieros diseñaban las alas de aviones y el chasis de los automóviles mediante el uso de trazadores. Un trazador es una banda flexible de madera o plástico con una sección transversal rectangular mantenida en un lugar por varias posiciones hechas por pesas de metal conocidas como *ducks* (patos), ese sistema mecánico era torpe y difícil para trabajar y no podía ser usado para dar una descripción permanente y reproducible de una curva (Haralambous, 2007).

Pierre Bézier (1910-1999), un ingeniero de Renault desde 1933, comenzó a hacer una investigación para encontrar un sistema para el modelado de curvas, o incluso superficies, que serían fáciles de calcular y también de manipular para el ser humano. Las curvas que llevan su nombre satisfacen estas condiciones, pero él no era el único en trabajar en estas técnicas, también estuvieron Paul Casteljaou, un ingeniero de Citroën; y Birkhoff, Garabedian y Boor de General Motors, en los Estados Unidos (Haralambous, 2007).

El trabajo de Bézier tuvo como aportación las ecuaciones de estas curvas que hoy se conocen en todo el mundo porque están en el corazón de los sistemas de diseño asistido por computadora; el trabajo de Casteljaou dio el algoritmo de dichas curvas (Haralambous, 2007).

De acuerdo con Farin (2002) (Farin, 2002), es necesario entender que el hablar de las curvas de Bézier se tiene que hacer referencia a los polinomios de Bernstein, los cuales se basan en el teorema del binomio de Newton: para cualquier número natural n , y números reales a y b , se cumple:

$$(a + b)^n = \sum_{i=0}^n \binom{n}{i} a^i b^{n-i} \quad \text{Ec. (1)}$$

Donde $\binom{n}{i}$, que también es denotado como C_i^n , es llamado coeficiente binomial y es igual a $\frac{n!}{(n-i)!i!}$.

Si $a = t$ y $b = (1 - t)$ donde $t \in [0, 1]$, se obtiene la identidad expresada en la Ec. (2):

$$1 = 1^n = (t + (1 - t))^n = \sum_{i=0}^n \binom{n}{i} t^i (1 - t)^{n-i} \quad \text{Ec. (2)}$$

La Ec. (2) permite introducir los llamados polinomios de Bernstein o funciones en la base de Bernstein de grado n , los cuales se representan con la Ec. (3):

$$B_{i,n}(t) = \sum_{i=0}^n \binom{n}{i} (1 - t)^{n-i} t^i \quad \text{Ec. (3)}$$

Donde

$B_{i,n}(t)$: Polinomio de Bernstein de grado n .

n : Grado del polinomio de Bernstein.

t : Parámetro que varía en el intervalo $[0, 1]$.

Como ya se mencionó, la representación analítica de las curvas de Bézier está basada en los polinomios de Bernstein y éstas se obtienen dibujando precisamente estos polinomios, de esta manera, las curvas de Bézier son curvas polinómicas que tienen una representación matemática particular. Una curva de Bézier de grado n es especificada por una secuencia de $n + 1$ puntos (**puntos de control**), el polígono se obtiene de la unión de estos puntos (**polígono de control**) con los segmentos de línea en el orden prescrito.

Dados $n+1$ puntos de control P_0, P_1, \dots, P_n , la curva de Bézier de grado n está definida como:

$$B(t) = \sum_{i=0}^n P_i B_{i,n}(t) \quad \text{Ec. (4)}$$

Donde

$B(t)$: Curva de Bézier de grado n .

$$B_{i,n}(t) = \begin{cases} \binom{n}{i}(1-t)^{n-i}t^i, & \text{si } 0 \leq i \leq n \\ 0, & \text{otro} \end{cases}$$

(Polinomios de Bernstein).

P_i : Puntos de control (x_i, y_i) , $i = 0, 1, \dots, n$.

n : Grado del polinomio de Bernstein.

t : Parámetro que varía en el intervalo $[0, 1]$.

Así, el polígono de control de Bézier está formado por la unión de los puntos de control P_0, \dots, P_n en el orden especificado.

La Ec. (4) es una ecuación vector y puede ser expresada por sus funciones de coordenadas paramétricas de las Ecs. (5) y (6):

$$b_x(t) = x(t) = \sum_{i=0}^n x_i B_{i,n}(t) \quad \text{Ec. (5)}$$

$$b_y(t) = y(t) = \sum_{i=0}^n y_i B_{i,n}(t) \quad \text{Ec. (6)}$$

Una *curva de Bézier lineal* es un segmento de recta que une dos puntos de control $P_0(x_0, y_0)$ y $P_1(x_1, y_1)$, sustituyendo $n = 1$ en la Ec. (4), se obtiene la Ec. (7):

$$B(t) = P_0(1-t) + P_1t \quad \text{Ec. (7)}$$

Donde

$B(t)$: Curva de Bézier de grado $n = 1$

P_0 y P_1 : Puntos de control con coordenadas (x_i, y_i) , $i = 0, 1$.

t : Parámetro que varía en el rango de $[0, 1]$.

La Ec. (7) es la representación vectorial de una curva de Bézier lineal.

Para una curva de Bézier cuadrática se establecen tres puntos de control $P_0(x_0, y_0)$, $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$, sustituyendo $n = 2$ en la Ec. (4), se obtiene la Ec. (8):

$$B(t) = P_0(1-t)^2 + P_12t(1-t) + P_2t^2 \quad \text{Ec. (8)}$$

Donde

$B(t)$: Curva de Bézier de grado

P_0, P_1 y P_2 : Puntos de control con coordenadas (x_i, y_i) , $i = 0, 1, 2$.

t : Parámetro que varía en el rango de $[0, 1]$.

Asimismo, para una curva de Bézier cúbica se establecen cuatro puntos de control $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ y $P_3(x_3, y_3)$ sustituyendo $n = 3$ en la Ec. (4), se obtiene la Ec. (9):

$$B(t) = P_0(1-t)^3 + P_13t(1-t)^2 + P_23t^2(1-t) + P_3t^3 \quad \text{Ec. (9)}$$

Donde

$B(t)$: Curva de Bézier de grado

P_0, P_1, P_2 y P_3 : Puntos de control con coordenadas (x_i, y_i) , $i = 0, 1, 2, 3$.

t : Parámetro que varía en el rango de $[0, 1]$.

La derivada e integral de las curvas de Bézier se obtienen a partir de la fórmula general (Ec. 4).

La matemática anteriormente descrita es utilizada en este trabajo para la elaboración de señales eléctricas físicas que permitan, en algunos casos, sustituir a un generador de señales.

Por un lado, los generadores de señales son equipos electrónicos que se originaron para

generar señales eléctricas las cuales poseen características como frecuencia y amplitud (San Miguel, 2010). Existe un gran número de generadores que tienen funciones parecidas siendo estos aplicados en un sinfín de áreas (Alonso Rodríguez, 2014), tal es el caso de Alonso (2014) quien realizó un generador de señales virtuales con Labview, donde la tarjeta arduino fue utilizada como interfaz para la salida física de la señal usando la técnica de síntesis digital directa (Alonso Rodríguez, 2014).

En el mismo contexto, Mite (2017) presenta un generador de señales con arduino y electrónica para manejar la amplitud de la señal (Mite Rivera, 2017); Montiel (2012) describió un generador de señales diente de sierra con electrónica analógica incluyendo la rectificación de la fuente de alimentación de corriente alterna y utilizó la detección de cruce por cero (Montiel, 2012); Soriano (2014), analizó transitorios de circuitos eléctricos con un generador de señales utilizando instrumentos virtuales (Soriano, Merconchini, & Cruz, 2014).

Por otro lado, las curvas de Bézier han sido implementadas en diferentes aplicaciones por ejemplo, Aumann (2003) presenta un algoritmo para generar superficies manipulables de grado y configuración arbitraria (Marin & Jaimez, 2008), Montés (2008) utilizó las curvas para calcular, en tiempo real, trayectorias curvas en robots móviles autónomos (Montés, Herraiz, Armesto, & Tornero, 2008). Asimismo, existen autores que trabajan con las curvas de Bézier para la generación de figuras geométricas y planificación de trayectorias (Cai & Wang, 2009; Fougerolle, Lanquetin, Neveu, & Lauthelier, 2008; Marin & Jaimez, 2008; Marsh, 2006; Yan, 2006).

Como se puede observar, existen trabajos de generadores de señales que utilizan software y tarjetas como arduino, así también

aplicaciones para hacer figuras geométricas y para definir trayectorias. Específicamente este trabajo se enfoca en la generación de señales eléctricas usando las curvas de Bézier.

DESARROLLO

DESCRIPCIÓN DEL MÉTODO

Para el desarrollo del generador de señales eléctricas a través de las curvas de Bézier se siguió la siguiente metodología:

- Investigación documental
- Diagrama de bloques del sistema.
- Programación de los algoritmos de la curva de Bézier.
- La interfaz de usuario en Labview.
- Simulación en Matlab de las curvas de Bézier.
- Programación de la comunicación serial entre la computadora y el microcontrolador asimismo la generación de la modulación ancho de pulso (PWM).
- Diseño del filtro para la señal generada.
- Pruebas y resultados.
- Conclusiones.

IMPLEMENTACIÓN

Las ecuaciones antes mencionadas, otras como las curvas de Bézier con intervalo arbitrario, las derivadas e integrales de las curvas de Bézier y la integral de Fourier fueron usadas para la generación y análisis de la señal eléctrica.

Primeramente, se diseñó un diagrama a bloques del generador de señales eléctricas a través de las curvas de Bézier como se observa en la Figura 1.

Cabe mencionar que el hardware que se utilizó como interfaz del usuario fue una computadora portátil y el software utilizado fue Labview versión 8.5.

Con base en la Figura 1, las funciones del sistemas son: calcular los puntos de la curva de Bézier elegida, posteriormente mandar

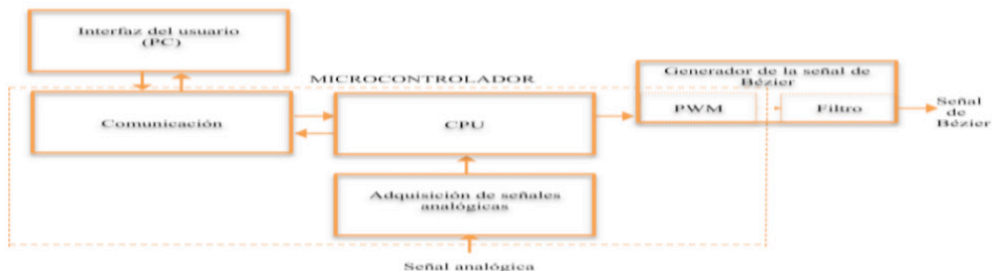


Figura 1. Diagrama de flujo del sistema.

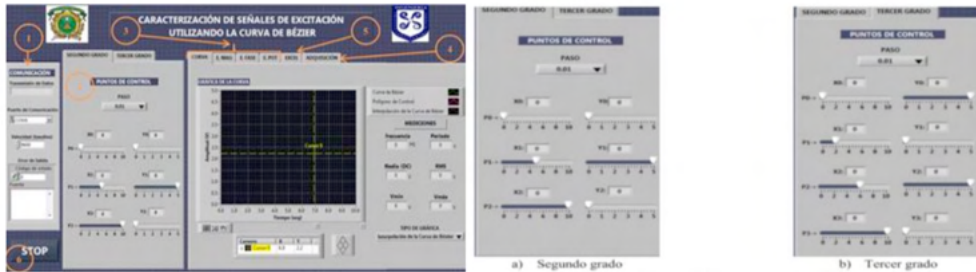


Figura 2. Interfaz de usuario en la computadora (lado izquierdo) y módulo de curvas de Bézier (lado derecho).

estos datos al microcontrolador por medio del bloque denominado comunicación, almacenar los datos obtenidos en la adquisición y generación de las señales, obtener el espectro de la señal y almacenar los datos para después, si el usuario lo desea, exportarlos a un libro de trabajo en el software de Excel.

En la interfaz del usuario que se puede observar en la Figura 2, el usuario puede: establecer una comunicación con el microcontrolador, seleccionar el orden de la curva de Bézier, asignar los puntos de control y la elección de almacenar los datos de las curvas y adquirir datos desde el microcontrolador, etc.

Haciendo referencia a la Figura 2, para el establecimiento y monitoreo de la comunicación serial, es recomendable que el valor de estos controles se modifique antes de ejecutar el VI (*Virtual Instruments*, instrumento virtual) del prototipo. El indicador denominado "Transmisión de Datos" muestra los datos que se van enviando a través del puerto Usb de la computadora. El indicador denominado "Error de Salida"

muestra el código y la fuente de error en el VI si éste existe en el momento de ejecución del programa. Adicionalmente se tiene el módulo de parámetros de la curva de Bézier, esta parte se utiliza para la manipulación de los puntos de control, el "paso" y el grado de la curva de Bézier que se desea generar.

La ventana denominada "Segundo grado" de la Figura 2, contiene siete controles y seis indicadores numéricos. El control denominado "Paso" establece el tamaño del paso con el cual la curva de Bézier es generada. Hay tres valores que se pueden elegir en este control, los cuales son: 0.1, 0.01 y 0.001 (el tamaño del paso por defecto es 0.01). Este valor sólo afecta la gráfica de la curva de Bézier que muestra la interfaz. Los puntos de control de la curva de segundo grado son: $\mathbf{P}_0(x_0, y_0)$, $\mathbf{P}_1(x_1, y_1)$, $\mathbf{P}_2(x_2, y_2)$, por lo tanto, a cada coordenada de cada punto de control le corresponde un control y un indicador numérico respectivamente.

La ventana denominada "Tercer grado" (Figura 2) contiene nueve controles y ocho indicadores numéricos. El control denominado "Paso" tiene la misma función

que el control que se encuentra en la ventana “Segundo grado” y que es denominado con el mismo nombre. Los puntos de control de la curva de Bézier de tercer grado son: $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ y $P_3(x_3, y_3)$, y también, a cada coordenada de cada punto de control le corresponde un control y un indicador numérico respectivamente. El intervalo de la coordenada x de los puntos de control es de 0 a 10, y el intervalo de la coordenada y es de 0 a 5. Los valores de estos controles se pueden modificar en cualquier momento aún si el VI se está ejecutando.

El control denominado “Tipo de Gráfica” establece la gráfica que se desea visualizar, la cual puede ser la curva de Bézier o la interpolación de la curva de Bézier.

Los indicadores numéricos denominados “Frecuencia” y “Periodo” muestran la frecuencia y el periodo de la señal generada respectivamente. Los indicadores denominados “Media (DC)” y “RMS” indican el valor promedio y el valor cuadrático medio (Root Media Square, RMS) de la señal generada respectivamente. Los indicadores denominados “V_{mín}” y “V_{máx}” indican el valor de amplitud mínimo y valor de amplitud máximo de la señal generada respectivamente. La gráfica tiene una propiedad en la cual el cursor denominado “Cursor 0” se puede desplazar a través de la gráfica de la señal y mostrar el valor de sus coordenadas (x, y).

En la parte “Gráficas y espectros de la señal generada” se muestra la gráfica de la curva de

Bézier que se está generando cuando el VI se está ejecutando, así como también, las gráficas de los espectros de magnitud, fase y potencia.

Los bloques denominados Comunicación, CPU, Adquisición de señales analógicas y Generador física de la señal de Bézier y la modulación ancho de pulso (PWM) se realizaron con la tarjeta Arduino Mega. El bloque Generador de la señal de Bézier está dividido en dos partes: una el PWM, que es donde se genera la señal de Bézier en forma digital y se encuentra dentro del microcontrolador; otra el Filtro, es donde se convierte la señal digital en una señal analógica mediante un filtro pasa-bajo.

Como se mencionó, la etapa denominada “Generador de la señal de Bézier” es la encargada de generar la señal de excitación, la cual está basada en el algoritmo de Bézier.

El método que se utilizó en esta etapa es la Modulación por Ancho de Pulso (*Pulse-Width Modulation*, PWM) y un filtro pasa-bajo de componentes pasivos como se muestra en la Figura 3.

Para lo anterior se utilizó la tarjeta arduino Mega (Figura 3) y de sus recursos el Temporizador/Contador en el modo de operación denominado como PWM Fase y Frecuencia Correcta (*PWM Phase and Frequency Correct*). En este modo de operación el contador cuenta repetidamente de BOTTOM (0x0000) a TOP(1600) y después de TOP a BOTTOM.

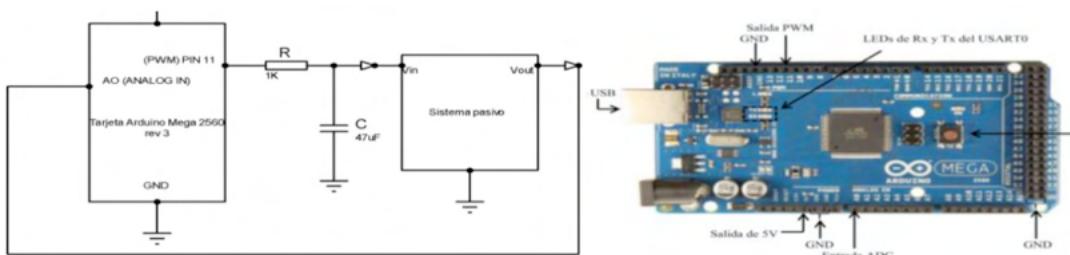


Figura 3. Diagrama esquemático del prototipo y la tarjeta arduino Mega.

La frecuencia de la señal PWM para este modo de operación es calculada por la Ec. (10):

$$f_{OC1APFCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP} \quad \text{Ec. (10)}$$

Donde

$f_{OC1APFCPWM}$: Frecuencia de la señal PWM

$f_{clk_I/O}$: Frecuencia del reloj

N : El divisor del *prescaler* (1, 8, 64, 256, ó 1024)

La frecuencia de reloj de la tarjeta de Arduino es 16 MHz, el divisor del *prescaler* es 1, entonces la frecuencia de la señal PWM es 5 KHz.

La frecuencia de la señal PWM que se genera en el microcontrolador es de 5 KHz. Para el filtro pasa-bajo se eligió una frecuencia de corte de 3 Hz con el fin de obtener el menor rizo posible. Se seleccionó el valor del capacitor de 47 μF y se sustituyen los valores anteriores en la Ec. (11) para obtener el valor de la resistencia, de esta manera, se tiene que:

$$R = \frac{1}{2\pi(47 \mu\text{F})(3 \text{ Hz})} = 1.129 \times 10^3 \Omega \quad \text{Ec. (11)}$$

Para evitar utilizar varias resistencias el valor se aproximó a 1 Kohm, de manera que el valor de la frecuencia de corte con ese valor de resistencia es de 3.38 Hz. En la Figura 3 se muestra la conexión de la tarjeta arduino con el filtro pasa-bajas con los valores utilizados.

La respuesta en frecuencia del filtro paso-bajas es dada por la Ec. (12):

$$H(j\omega) = \frac{1}{1+0.047(j\omega)} \quad \text{Ec. (12)}$$

En la ecuación anterior, $H(j\omega)$ es la respuesta en frecuencia. La frecuencia de corte angular es igual a: $\omega_c = 2\pi f_c$, sustituyendo el valor de la frecuencia de corte, se obtiene que $\omega_c = 2\pi(3.38) = 21,23 \text{ rad/seg}$.

La Figura 4, muestra la gráfica de respuesta en frecuencia del filtro pasa-bajo del prototipo. El eje x representa a la frecuencia angular en unidades de rad/seg. El eje y representa a la magnitud $H(j\omega)$ en dB. Se puede observar en dicha figura que cuando el valor de la magnitud es de -3 dB, el valor de la frecuencia angular es igual a 21.2, la cual es la frecuencia de corte angular. El ancho de banda es 21.2 rad/seg.

PRUEBAS Y RESULTADOS

El sistema completo como prototipo se muestra en la Figura 5, donde en la computadora se tiene la interfaz de usuario que envía los datos generados a la tarjeta arduino, también los puede almacenar o enviar al Excel. Asimismo, una vez que el microcontrolador recibe la información anterior desde la computadora, utiliza el ancho del pulso para generar la señal eléctrica la cual es pasada por un filtro pasa-bajo para suavizar la señal.

El algoritmo para generar las curvas de Bézier también se realizó en Matlab y en

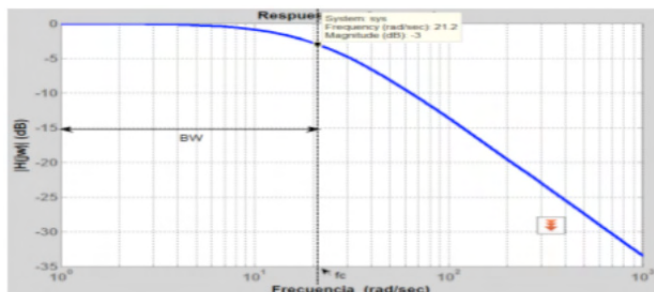


Figura 4. Respuesta en frecuencia del filtro pasa-bajo del prototipo

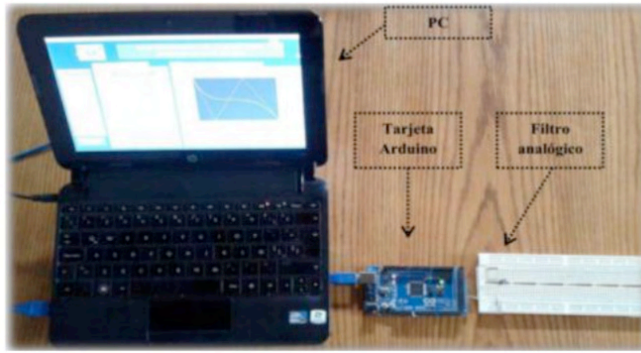


Figura 5. Interfaz de usuario y el sistema final

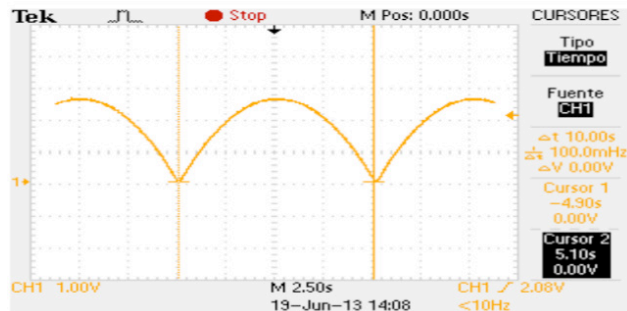
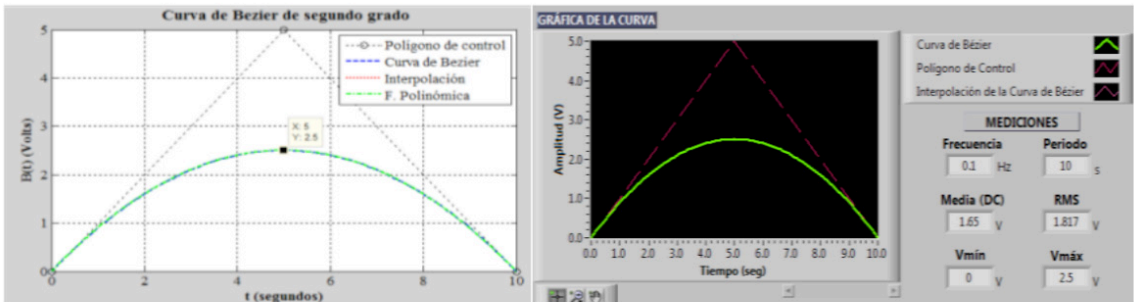


Figura 6. Curva de Bézier en Matlab, en Labview y el osciloscopio.

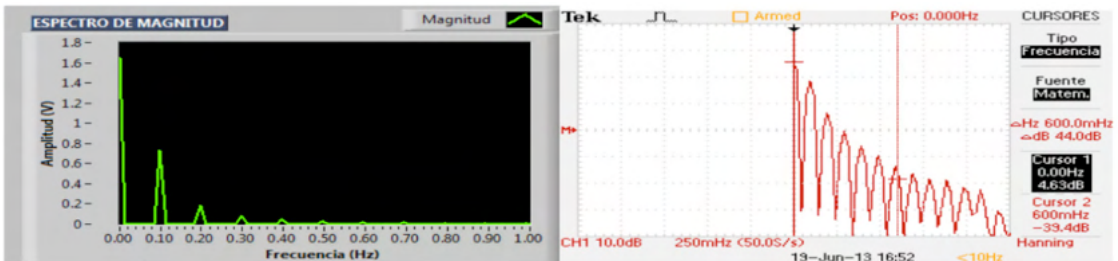


Figura 7. Espectro de la señal generada, con Labview y el Osciloscopio

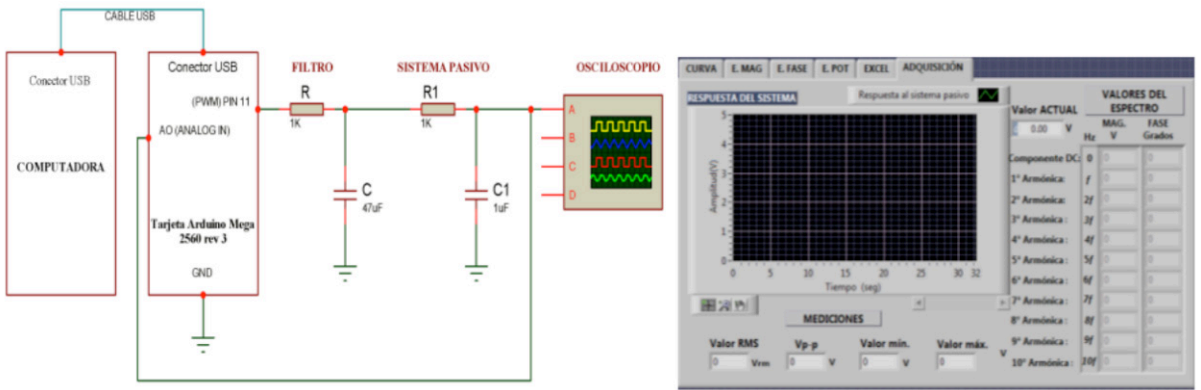


Figura 8. Circuito RC de prueba y ventana de adquisición de señales en LabView.

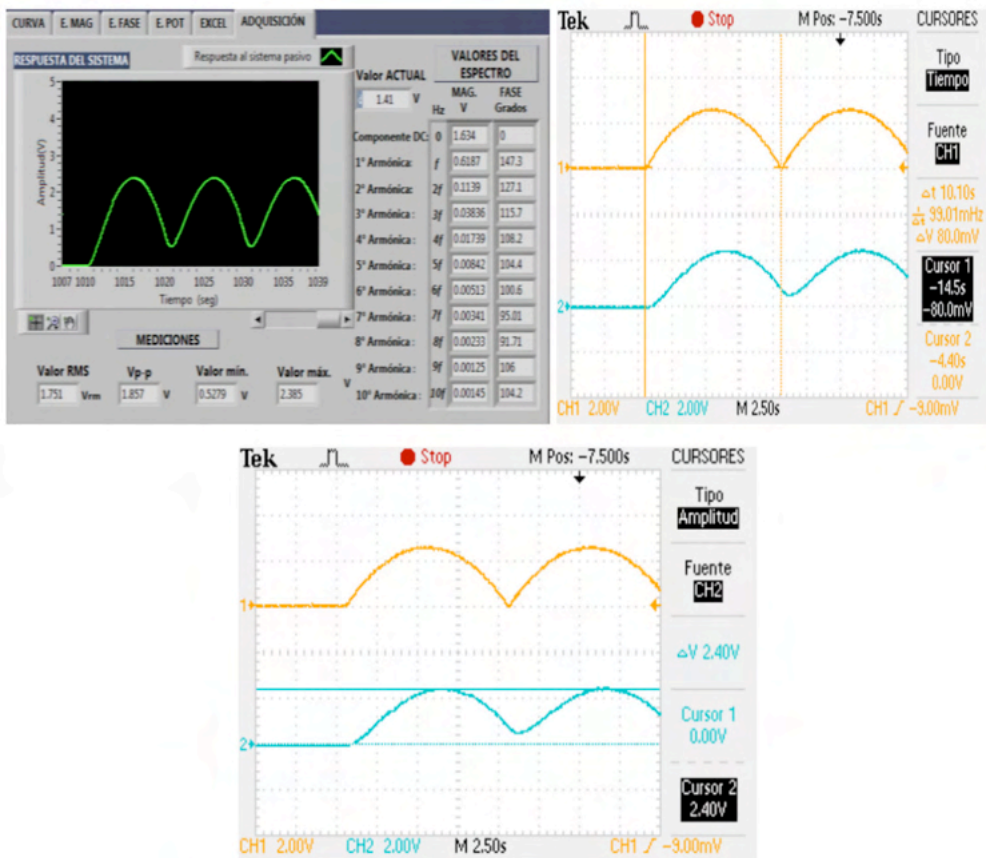


Figura 9. Medición de la respuesta del circuito RC, en LabView y el osciloscopio (Tiempo y amplitud).

Labview, en la Figura 6 se muestran ambas gráficas. Se puede observar en la misma Figura el resultado de los datos después de arduino y el filtro medidos con un osciloscopio.

El eje x representa el tiempo de la señal en segundos y el eje y representa el valor de tensión en Volts. También en la gráfica obtenida en la simulación por MatLab se puede apreciar que la amplitud máxima de la curva es de 2.5 y el valor del periodo de 10 segundos. La gráfica de la simulación realizada en Labview muestra que la amplitud máxima de la curva es de 2.5 V y el periodo es de 10 segundos. Los valores de amplitud máxima y periodo obtenidos en las simulaciones en Matlab, Labview y el osciloscopio son los mismos. Se obtuvo de igual forma el espectro de frecuencia de la señal en Labview y en el osciloscopio como se muestran en las gráficas de la Figura 7.

Se tomaron valores del espectro de frecuencia de la señal generada en Matlab, Labview y el osciloscopio. Tomando como referencia el valor del osciloscopio, el error en amplitud fue del 4.78% de 5 Voltios y en cuanto al periodo del 1% de 10 seg., el cual se puede decir que es aceptable. Adicionalmente, se aplicaron estas señales a un circuito eléctrico sencillo "RC" $R=1\text{Kohm}$ y $C=1\mu\text{F}$ y se midió su respuesta. La señal eléctrica de respuesta del circuito se obtuvo en la computadora a través del convertidor analógico digital del mismo arduino y se analizó con el software realizado en LabVIEW, como se muestra en la Figura 8.

La ventana presenta la respuesta del circuito de prueba, obteniendo valores rms (root mean square), pico a pico, valor mínimo, valor máximo y los 10 primeros armónicos de la señal (magnitud y fase). Dichas mediciones también fueron realizadas con osciloscopio, como lo muestra la Figura 9. El periodo de la señal es de 10.1 seg. y la amplitud de 2.4 Voltios.

CONCLUSIONES

El prototipo genera señales eléctricas de 0 a 5 voltios basadas en las curvas de Bézier y la modulación de ancho del pulso.

Los resultados de las pruebas realizadas a las señales de Bézier de segundo y tercer grado muestran que el error relativo promedio del valor de la amplitud entre señales simuladas y reales fue de 4.78% de 5 voltios.

Con respecto al valor del periodo de las señales obtenidas en las simulaciones y reales fue de 1% de 10 seg.

La modulación ancho de pulso fue utilizada para la generación de la señal y el filtro pasabajo para la recuperación analógica de la misma, esto es una limitación debido a que la salida está sujeta a dicho diseño.

Se utilizó el convertidor analógico-digital de Arduino para regresar la señal del circuito de prueba a la computadora.

Se aplicó la señal a un circuito RC y se midió la respuesta tanto en el software hecho en LabVIEW como en el osciloscopio, De las pruebas realizadas se concluye que el valor de tensión de la señal adquirida en LabVIEW en cualquier instante, está dentro del intervalo de $[V_{sa} \pm 0.018V]$, donde V_{sa} es el valor de tensión de la señal adquirida.

REFERENCIAS

- Alonso Rodríguez, R. (2014). *Generador virtual de funciones de onda predefinida y arbitraria con herramientas hardware y software ARDUINO y LABVIEW*. Universidad de Valladolid, Escuela Técnica Superior de Ingenieros de Telecomunicación, Valladolid, España.
- Cai, H.-h., & Wang, G.-j. (2009). *A new method in highway route design: joining circular arcs by a single C-BÁ©zier curve with shape parameter*. *Journal of Zhejiang University-SCIENCE A*, 10(4), 562-569.
- Farin, G. E. (2002). *Curves and surfaces for CAGD: a practical guide*: Morgan Kaufmann.
- Fougerolle, Y. D., Lanquetin, S., Neveu, M., & Lauthelier, T. (2008). *A geometric algorithm for ray/Bézier surfaces intersection using quasi-interpolating control net*. Paper presented at the Signal Image Technology and Internet Based Systems, 2008. SITIS'08. IEEE International Conference on.
- Haralambous, Y. (2007). *Fonds & Encoding*: O'Reilly Media Inc.
- Marin, J. V., & Jaimez, C. O. (2008). *Diseño y construcción de un robot rotulador de movimiento carteziano para superficies planas mediante las curvas de Bézier*. Universidad Autónoma del Estado de México.
- Marsh, D. (2006). *Applied geometry for computer graphics and CAD*: Springer Science & Business Media.
- Mite Rivera, J. O. (2017). *Diseño de un generador de señales para estudiantes en la Facultad de Ingeniería Industrial*. Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de Ingeniería en Teleinformática.
- Montés, N., Herraéz, A., Armesto, L., & Tornero, J. (2008). *Real-time clothoid approximation by Rational Bezier curves*. Paper presented at the Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on.
- Montiel, M. M. (2012). *Elementos para la generación de diente de sierra por detección de cruce por cero*: Instituto tecnológico Superior de Teziutlan: Academia de Ingeniería Mecatrónica.
- San Miguel, P. A. (2010). *Electrónica aplicada*: Editorial Paraninfo.
- Soriano, G. B., M., Merconchini, F.-G., & Cruz, C., Llohandy. (2014). *Study of transient phenomena in first order circuits*. *Ingeniería Energética*, 35(3), 295-304.
- Yan, L. (2006). *Adjustable Bézier Curves with Simple Geometric Continuity Conditions*. *Mathematical and Computational Applications*, 21(4), 44.