



UNIVERSIDAD POLITÉCNICA SALESIANA

Carrera de Ingeniería Electrónica

**“ANÁLISIS DE LAS TÉCNICAS MODERNAS DE MODULACIÓN
APLICADAS A LOS SISTEMAS CD/CA”**

Tesis previa a la obtención del
título de Ingeniero en
Electrónica

AUTOR:

William Manuel González Gallegos

DIRECTOR:

Ing. Flavio Quizhpi

CUENCA-ECUADOR

2012

Certifico qué bajo mi dirección la tesis “**ANÁLISIS DE LAS TÉCNICAS MODERNAS DE MODULACIÓN APLICADAS A LOS SISTEMAS CD/CA**” fue realizado por el señor William Manuel González Gallegos

(f).....

Ing. Flavio Quizhpi

Director de Tesis

DEDICATORIA

El cumplimiento de esta meta va dedicado en especial a mi Mami quien con mucho esfuerzo y sacrificio me ayudo para dar este gran paso en mi vida, pues fueron fundamentales sus consejos, guía, cariño y su lucha por mi y mis hermanos, gracias Mamiqui por todo. A mi Papi que siempre me acompaña y guía. Con cariño para mi esposa Priscy quien me incentivó a culminar con éxitos esta tesis, gracias por llegar a mi vida y llenarla de felicidad.

A mis hermanos Janneth y Geovanny que siempre estuvieron conmigo y me acompañaron en mi lucha diaria, con quienes compartí alegrías y tristezas gracias por impulsarme a no desfallecer en mi lucha; a mi sobrino Joaquincito por toda su dulzura y cariño, tú también me impulsaste a finalizar mi trabajo. Para mis tíos Ana e Iván pues no dudaron nunca en ayudarme cuando acudí a ellos, gracias también por todo su afecto.

AGRADECIMIENTO

Un sincero agradecimiento a la “Universidad Politécnica Salesiana” y a todo su cuerpo docente por su excelente labor educativa y humana, pues ellos me han sabido orientar hacia esta meta. Mi profundo agradecimiento al Ing. Flavio Quizhpi pues siempre estuvo dispuesto a brindarme su ayuda y conocimientos para culminar mi tesis, gracias Flavio por guiarme en la secundaria y ahora en la universidad.

Un cordial agradecimiento a mis Papis Ruth y Pedro, esposa Priscy, tios Ana e Iván, hermanos Janneth y Geovanny, cuñado Andrés, sobrino Joaquincito y mi prima Andrea por sus palabras de aliento, ya que sin su apoyo no hubiese podido culminar mi carrera. Los quiero mucho a todos.

RESPONSABILIDAD

Yo, William Manuel González Gallegos con CI.: 0103093795; estudiante de la carrera de Ingeniería Electrónica, me hago responsable por todos los efectos sociales y tecnológicos que conllevan la realización de esta tesis. Además de los beneficios y problemas que podría ocasionar este proyecto. Y autorizo a la Universidad Politécnica Salesiana el uso de la misma con fines académicos.



William M. González Gallegos

CI.: 0103093795

ÍNDICE

INTRODUCCIÓN	VII
CAPÍTULO 1	9
INVERSORES MONOFÁSICOS	9
1.1. Introducción.....	9
1.1.1. Principio de Funcionamiento.....	11
1.1.2. TIPOS DE INVERSORES.....	17
1.2. Expresión de Corriente en Régimen Permanente	21
1.3. Tensión efectiva	24
1.4. Expresión series de Fourier	24
1.5. Factor de Distorsión Armónica	25
1.5.1 Normalización	25
1.5.2. Regulación del CONELEC No. 004/01.	25
1.5.3. Norma IEEE 519-1992.....	27
CAPÍTULO II	31
INVERSORES TRIFÁSICOS	31
2.1. Introducción.....	31
2.2. Tensión en Series de Fourier	34
2.3. Tensión Efectiva.....	35
2.4. Factor de Distorsión Armónica Total	35
2.5. Modelo en Vectores Espaciales	35
2.5.1. Inversor	35
2.5.2. Carga	40

CAPÍTULO III	43
TÉCNICAS DE MODULACIÓN	43
3.1. Modulación por ancho de pulso PWM.....	43
3.1.1. Índice de modulación de frecuencia.....	44
3.1.2. Índice de modulación de amplitud.....	44
3.1.3. Contenido Armónico	45
3.1.4. Algoritmo para simulación.....	47
3.2. Modulación por ancho de pulso Modificada SPWM.....	48
3.2.1. Algoritmo para simulación.....	50
CAPÍTULO IV.....	53
TÉCNICAS AVANZADAS DE MODULACIÓN	53
4.1. Introducción.....	53
4.2. Trapezoidal	53
4.2.1. Algoritmo para simulación	55
4.3. Por inyección de Armónicas.....	56
4.3.1. Algoritmo para simulación	58
4.4. Escalera	59
4.4.1. Algoritmo para simulación.....	64
4.5. Por Pasos	66
4.5.1. Algoritmo para la simulación.....	67
ANEXOS	71
A.1. MANUAL DEL PROGRAMA.....	71
A.1.1. Partes y componentes del programa.....	72
A.1.2. Funcionamiento del programa.....	73
A.1.3. ARCHIVO M-FILE DEL PROGRAMA.....	80

INTRODUCCIÓN

Los convertidores CD/CA tienen por objetivo la transformación de tensión CD a tensión CA de amplitud y/o frecuencia variable; dichos circuitos de conversión tienen una amplia aplicación en la industria. A estos también se les llama inversores, los cuales son utilizados en variadores de velocidad, sistemas de alimentación ininterrumpida, filtros activos, etc. El proceso de conversión de voltaje se logra mediante la implementación de técnicas de modulación, las cuales actúan sobre un puente inversor monofásico o trifásico. Según las características de estas técnicas, las propiedades de eficiencia en la conversión, contenido armónico de la señal de salida y pérdidas en el puente inversor cambian en función del tipo de modulación aplicado al inversor.

En las aplicaciones donde se usan los inversores generalmente es necesario controlar el voltaje de salida de los inversores para hacer frente a las variaciones de la entrada cd, para la regulación del voltaje de los inversores y para los requisitos de control constante del voltaje y la frecuencia. El método más eficiente de controlar la ganancia(y el voltaje de salida) es incorporar en los inversores el control de modulación del ancho de pulso(PWM).

La modulación por ancho de pulso (PWM, Pulse Width Modulation) proporciona un método para disminuir el factor de distorsión armónica en la corriente que el inversor suministra a la carga. Si bien la salida con PWM posee un contenido alto de armónicas, estas son de frecuencias elevadas lo cual facilita su filtrado y atenuación por parte de la carga. Si es que usamos otra referencia obtenemos otros tipos de modulación como la trapezoidal, por inyección de armónicas, por pasos, por escalera y SPWM, cada una de estas presentando ciertas ventajas en la salida por ejemplo la reducción del contenido armónico.

Con el desarrollo de esta tesis se ha realizado el análisis de las técnicas modernas de modulación aplicadas a los inversores pues con dicho análisis se presenta una herramienta importante al momento de estudiar los sistemas CD/CA, como ayuda para esto no sólo se ha realizado es estudio teórico sino también un software para simulación desarrollado en Matlab. Al final en anexos presentamos el código M de la programación y un manual de cómo usar un software; además a esta tesis se adjunta el software para que pueda ser usado.

CAPÍTULO 1

INVERSORES MONOFÁSICOS

1.1. Introducción

Los inversores, son circuitos que tienen como finalidad suministrar tensión o corriente alterna, variable en magnitud y frecuencia a partir de una fuente de corriente continua. Los rectificadores controlados en algunos casos y dependiendo del ángulo de disparo pueden trabajar como inversores. Las principales aplicaciones de los inversores son el control de velocidad y posición de los máquinas de corriente alterna, la fabricación de fuentes ininterrumpidas de potencia (UPS) para cargas críticas y dispositivos de corriente alterna que funciones a partir de una batería como los vehículos eléctricos.

Los convertidores de cd a ca se conocen como inversores. La función de un inversor es cambiar un voltaje de entrada en cd a un voltaje simétrico de salida en ca, con la magnitud y frecuencia deseadas. Tanto el voltaje de salida como la frecuencia pueden ser fijos o variables. Si se modifica el voltaje de entrada de cd y la ganancia del inversor se mantiene constante, es posible obtener un voltaje variable de salida. Por otra parte, si el voltaje de entra en cd es fijo y no es controlable, se puede obtener un voltaje de salida variable si se varia la ganancia del inversor, esto por lo general se hace controlando la modulación del ancho de pulso (PWM) dentro del inversor. La ganancia del inversor se puede definir como la relación entre el voltaje de salida en ca y el voltaje de entrada en cd.

En los inversores ideales, las formas de onda del voltaje de salida deberían ser senoidales. Sin embargo, en los inversores reales no son senoidales y contienen ciertas armónicas. Para aplicaciones de mediana y baja potencia, se pueden aceptar los valores de onda cuadrada o casi cuadrada; para aplicaciones de alta potencia, son necesarias las formas de onda senoidales de baja distorsión. Dada la

disponibilidad de los dispositivos semiconductores de potencia de alta velocidad, es posible minimizar o reducir significativamente el contenido armónico del voltaje de salida mediante las técnicas de conmutación.

Los inversores pueden clasificarse básicamente en dos tipos:

- Inversores monofásicos.
- Inversores trifásicos.

Cada tipo de inversor puede utilizar dispositivos de conmutación con activación y desactivación controlada como lo son los transistores bipolares BJT, los transistores bipolares de compuerta aislada IGBT, los transistores de efecto de campo de metal-óxido MOSFET, los tiristores, entre otros.

Estos inversores utilizan por lo general señales de control moduladas por ancho de pulso PWM, para producir un voltaje de salida de corriente alterna. El voltaje y la corriente de entrada del inversor pueden ser fijos o variables, y puede ser obtenido del voltaje de línea o de un generador de voltaje de corriente alterna a través de un rectificador y un filtro. También, puede ser obtenido de una batería, una celda de combustible, o de un arreglo de celdas solares.

A un inversor se le llama Inversor de Fuente de Voltaje VSI, si el voltaje de entrada se mantiene constante y la corriente es variable; o Inversor de Fuente de Corriente CSI si la corriente de entrada se mantiene constante y el voltaje variable.

Los inversores son ampliamente usados y algunas de sus aplicaciones pueden ser las siguientes:

- Controladores de motores de corriente alterna.
- Fuentes de poder ininterrumpibles (UPS).

- Fuentes de Corriente alterna.
- Generadores o compensadores estáticos de potencia reactiva.

1.1.1. Principio de Funcionamiento

Los inversores se pueden clasificar dependiendo del número de fases de voltaje de corriente alterna que generen a la salida, cuando se genera una sola fase de voltaje a la salida se conoce como inversor monofásico, y en inversor trifásico cuando se generan tres fases de voltaje a la salida. Cada tipo de inversor puede subdividirse en inversores de tres niveles que sintetizan una señal de corriente alterna con tres niveles de voltaje de CD, y en inversores multinivel los cuales sintetizan la señal de voltaje de corriente alterna a la salida con más de tres niveles de voltaje de CD.

En la figura 1.1., se presenta el esquema de un inversor monofásico. Este convertidor esta conformado por cuatro interruptores bidireccionales de corriente. La operación sincronizada de los interruptores Sw permite aplicar sobre la carga tensiones positivas (+VDC), negativas (-VDC) y cero (0).

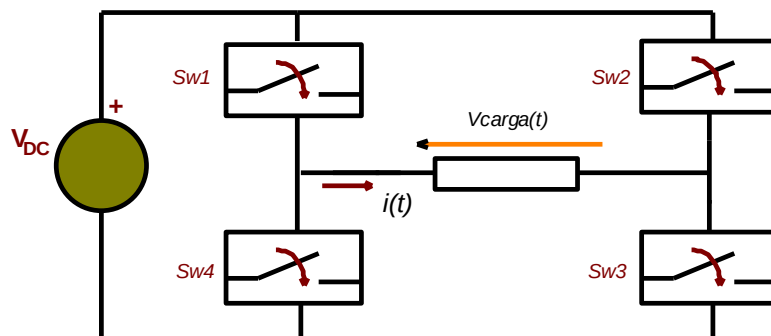


Figura 1.1. Esquema Inversor Monofásico

Para obtener tensión positiva (+VDC) en la carga, es necesario cerrar los interruptores Sw1y Sw3, mientras que Sw2y Sw4 permanecen abiertos. En la figura 1.2. se presenta la topología del convertidor para esta secuencia de operación de los interruptores.

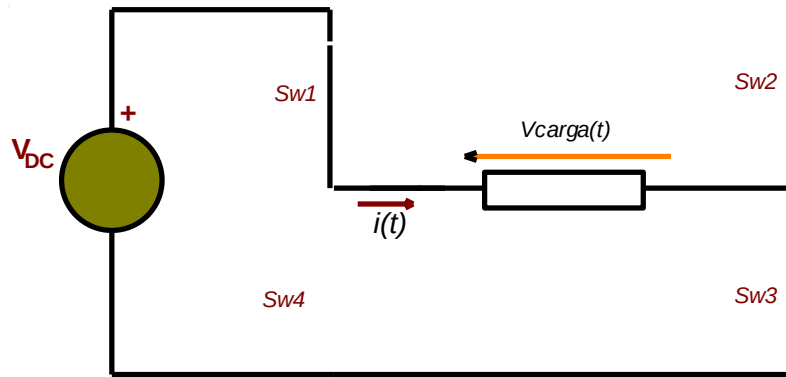


Figura 1.2. Topología funcionamiento Sw1 y Sw3 del inversor

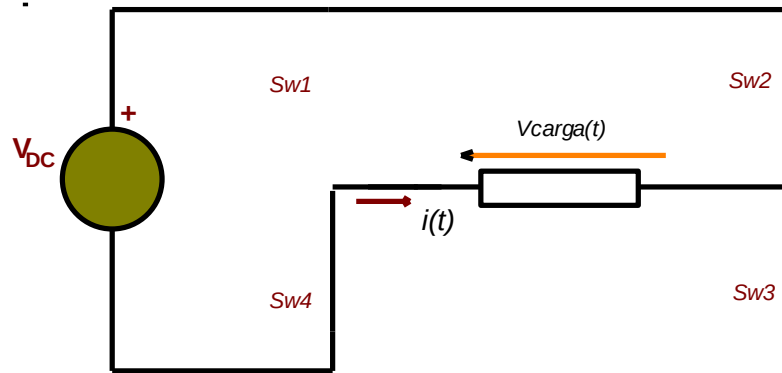


Figura 1.3. Topología funcionamiento Sw2 y Sw4 del inversor

Para obtener tensión negativa (-VDC) en la carga, es necesario cerrar los interruptores Sw2 y Sw4, mientras que Sw1 y Sw3 permanecen abiertos. En la figura 1.3. se presenta la topología del convertidor para esta secuencia de operación de los interruptores.

Para obtener tensión cero (0) en la carga, es necesario cerrar los interruptores Sw2 y Sw1 o Sw3 y Sw4 mientras que los demás permanecen abiertos.

Generalmente se alterna las dos secuencias de disparo, de forma simétrica, para obtener tensión cero en la carga con la finalidad que todas las componentes manejen los mismos niveles de pérdidas.

En la tabla 1.1. se presenta un resumen de la secuencia de operación de los interruptores para obtener cada una de las tensiones $+V_{DC}$, $-V_{DC}$ y 0 sobre la carga. Controlando el tiempo que el convertidor permanece en cada uno de los estados de la tabla 1.1., se puede controlar la frecuencia y magnitud efectiva de la tensión o corriente sobre la carga. Los puentes inversores pueden trabajar con carga pasiva o activa alterna.

Interruptores Cerrados	Tensión sobre la Carga
Sw1 y Sw3	$+V_{DC}$
Sw2 y Sw4	$-V_{DC}$
Sw1 y Sw2	0
Sw3 y Sw4	0

Tabla 1.1. Secuencia operación interruptores y voltaje a la carga

En la figura 1.4. se presenta la topología del convertidor para esta secuencia de operación de los interruptores.

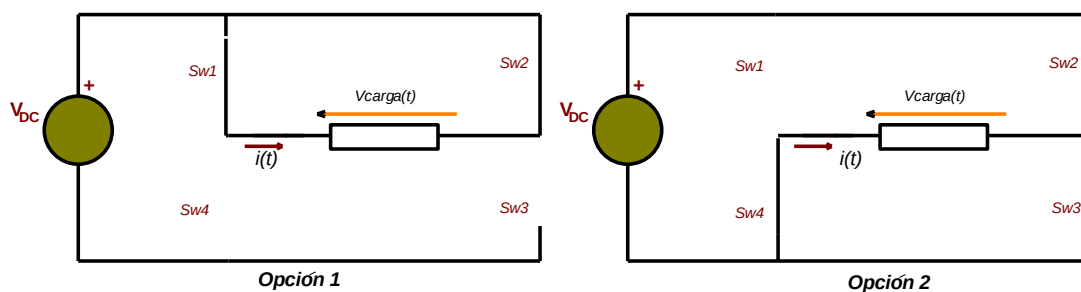


Figura 1.4. Topología de los interruptores para obtener 0v

En la figura 1.5., se presenta la forma de onda de tensión sobre la carga para una operación simétrica del inversor en dos estados ($+V_{DC}$, $-V_{DC}$). Controlado el tiempo de conmutación de los interruptores ($T/2$), se puede modificar la frecuencia

de la onda de tensión de solida. La tensión efectiva sobre la carga se puede calcular como:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T V_{DC}^2 dt} = V_{DC}$$

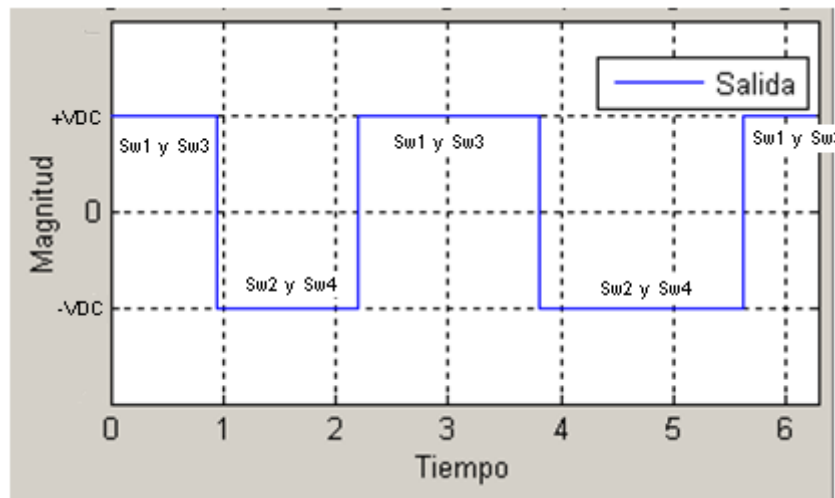


Fig. 1.5. Tensión en la carga para un inversor monofásico en operación de 2 estados

Para modificar el valor efectivo de la onda de salida del inversor, es necesario modular el valor de la fuente DC en cada semiciclo de la onda de alterna de forma simétrica. Esta modulación se puede realizar de forma análoga a la operación de los controladores DC - DC (chopper) en donde durante el tiempo de conducción de las componentes se realizaban cortocircuitos en la carga a fin de disminuir el valor de la tensión media sobre esta. A este tipo de operación se le conoce como control por tres estados (+VDC, -VDC, 0). Otra posibilidad de reducir el valor medio de la fuente DC, durante el semi ciclo de operación de la onda alterna es invertir el valor de la fuente durante un instante de tiempo, a esta operación se le conoce como control por dos estados (+VDC, -VDC).

En la figura 6, se presenta la forma de onda de tensión sobre la carga para un inversor con control de 3 estados. Para este caso la tensión efectiva sobre la carga es:

$$V_{rms} = \sqrt{\frac{1}{T} \int_x^{a+x} V_{DC}^2 dt + \int_{x+\frac{T}{2}}^{a+x+\frac{T}{2}} V_{DC}^2 dt} = \sqrt{\frac{2}{T} \int_x^{a+x} V_{DC}^2 dt} = V_{DC} \sqrt{\frac{2a}{T}}$$

Donde:

$$0 \leq a \leq \frac{T}{2}$$

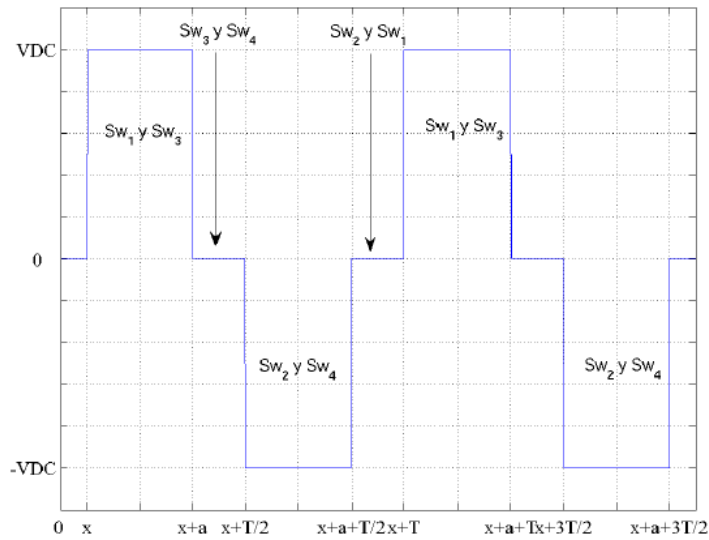


Figura 1.6. Tensión en la carga para un inversor monofásico en operación de 3 estados (Rashid, 1995)

Realizando el cambio de variable $a = T/2 - 2x$ en la expresión anterior, se obtiene:

$$V_{rms} = V_{DC} \sqrt{1 - \frac{4x}{T}}$$

La tensión de salida del inversor de la figura 6, aprovechando su simetría, se puede expresar en series de Fourier como:

$$v(t) = \sum_{n=1,2,3,\dots}^{\infty} C_n \text{sen}(n\omega t)$$

Donde:

$$C_n = \frac{4}{T} \int_x^{\frac{T}{2}-x} V_{DC} \text{sen}(n\omega t) d\omega t = \left(\frac{8V_{DC}}{nT} \right) \cos(nx)$$

La variación del valor de "x" permite modificar el valor efectivo de la señal de salida, así como la amplitud de cada armónica de la onda. Por esta razón el contenido armónico de la señal puede ser controlado con una escogencia adecuada del valor de "x". Por ejemplo un valor de $x = T/12$, anula la tercera armónica de la señal y sus múltiplos. En la figura 7, se presenta un ejemplo gráfico de la eliminación de las terceras y quinta armónica en tres formas de ondas generadas por el inversor. Se puede observar en la figura 7.a y 7.b como el área positiva y negativa en ambas gráficas se compensan, ocasionando la anulación de la armónica respectiva. Por otro lado en la Figura 7.c se evidencia que en el espectro armónico de la señal no hay presencia de tercera ni quinta armónica.

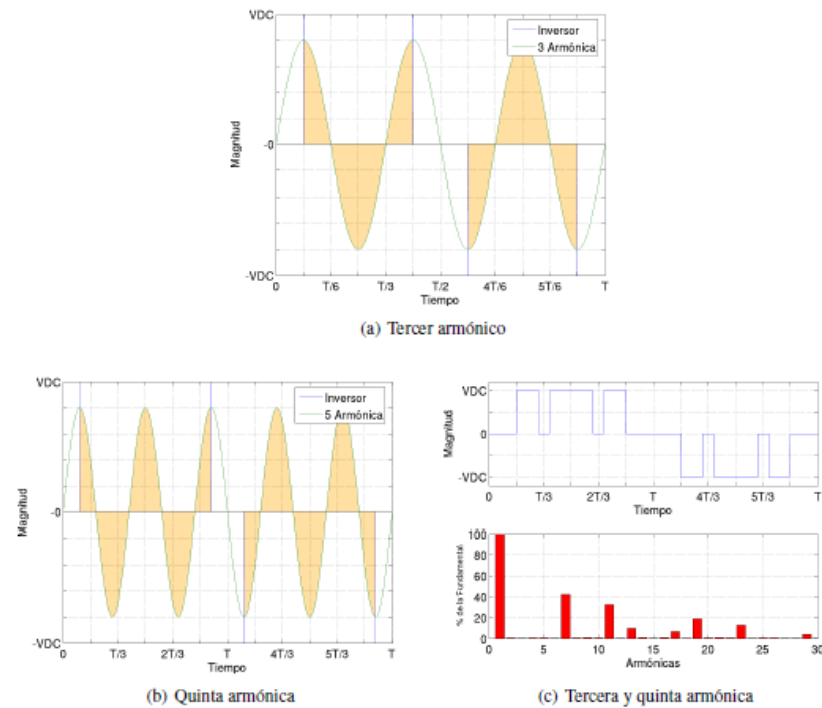


Figura 1.7. Eliminación de armónicos. (Bueno, 2007)

1.1.2. TIPOS DE INVERSORES

Los inversores se pueden clasificar dependiendo del número de fases de voltaje de corriente alterna que generen a la salida, cuando se genera una sola fase de voltaje a la salida se conoce como inversor monofásico, y en inversor trifásico cuando se generan tres fases de voltaje a la salida. Cada tipo de inversor puede subdividirse en inversores de tres niveles que sintetizan una señal de corriente alterna con tres niveles de voltaje de CD, y en inversores multinivel los cuales sintetizan la señal de voltaje de corriente alterna a la salida con más de tres niveles de voltaje de CD.

1.1.2.1. Inversor de Medio Puente

La configuración más simple de un inversor monofásico es el inversor de medio puente, figura 1.8.

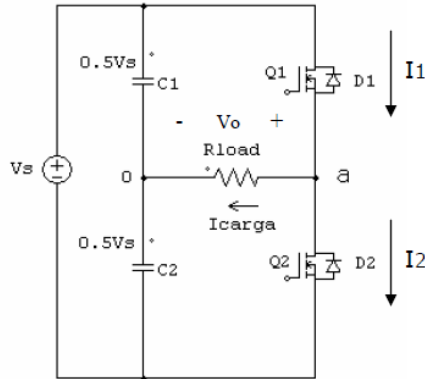


Figura 1.8. Inversor monofásica medio puente (Rashid, 1995)

El circuito consiste en un par de dispositivos de conmutación Q1 y Q2 conectados en serie a través de una fuente de voltaje de corriente directa V_s , y una carga conectada entre el punto a y el punto central 0 en la unión de los dos capacitores C1 y C2, los cuales dividen el voltaje de entrada V_s . Los dispositivos Q1 y Q2 conmutan en forma alterna en un ángulo ϕ para generar el voltaje de salida.

Mediante la figura 1.8 se puede explicar el principio de funcionamiento del inversor de medio puente. Inicialmente, se activa el transistor Q1 durante un tiempo $T_o/2$, el voltaje instantáneo a través de la carga V_o será $V_s/2$. Después, el transistor Q2 es activado por lo tanto Q1 desactivado durante un tiempo $T_o/2$. El voltaje $-V_s/2$ aparece en la carga.

En la figura 1.9 se pueden observar las formas de onda del voltaje y la corriente en los dispositivos de conmutación de inversor de medio puente con carga resistiva. En realidad, una pequeña brecha (T_d) o banda muerta es

conservada entre conmutaciones como se indica en la figura 1.9 para prevenir un corto circuito en la estructura.

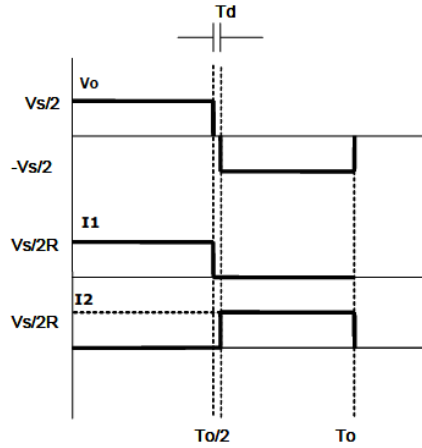


Figura 1.9. Forma de onda inversor medio puente (Rashid, 1995)

El voltaje rms de salida V_0 se puede encontrar a partir de:

$$V_o = \sqrt{\frac{2}{T_o} \int_0^{T_o/2} \frac{V_s^2}{4} dt} = \frac{V_s}{2}$$

El voltaje instantáneo de salida v_0 puede expresarse en una serie de Fourier como:

$$v_o = \sum_{n=1,3,5,\dots}^{\infty} \frac{2V_s}{n\pi} \sin(n\omega t)$$

$$v_o = 0 \text{ para } n = 2, 4, 6, \dots$$

Donde $\omega = 2\pi f_o$ es la frecuencia del voltaje de salida en radianes/segundo. Para $n=1$, la ecuación proporciona el valor rms de la componente fundamental como:

$$V_1 = \frac{2V_s}{\sqrt{2\pi}} = 0,45V_s$$

1.1.2.2. Inversor de Puente Completo o Puente H

Dos inversores de medio puente pueden ser conectados para formar un inversor de puente completo, o puente H, como el de la figura 10.

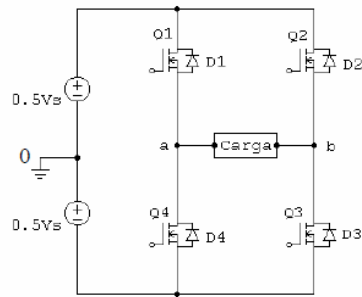


Figura 1.10. Inversor puente completo (Rashid, 1995)

La ausencia de los capacitores es notoria y no se utilizan en este caso. La carga esta conectada entre los puntos a y b. En el modo de operación de señal de onda cuadrada, la cual es mostrada en la figura 1.11, el par de dispositivos Q1Q3 y Q2Q4 conmutan en forma alterna para generar la señal de voltaje de salida de onda cuadrada de amplitud V_s .

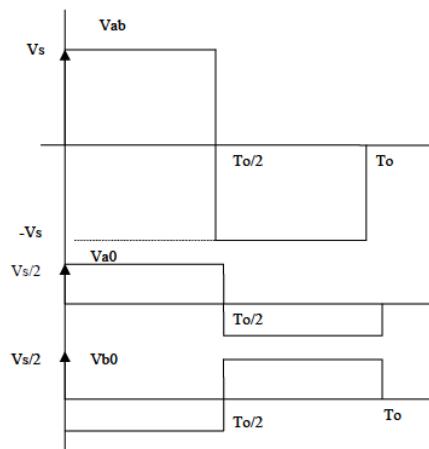


Figura 1.11. Forma de onda inversor puente completo (Rashid, 1995)

Para esta estructura tanto los diodos y los MOSFET están dimensionados para soportar el voltaje de la fuente V_s cuando estos se encuentran polarizados inversamente y en corte respectivamente. La corriente pico que deberán conducir los MOSFET es I_m y la corriente que conducirán los diodos es $I_m \sen\varphi$.

El voltaje rms de salida se puede determinar a partir de:

$$V_o = \sqrt{\frac{2}{T_o} \int_0^{T_o/2} V_s^2 dt} = V_s$$

El voltaje instantáneo de salida está determinado por:

$$v_o = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_s}{n\pi} \sen(n\omega t)$$

Y para $n=1$, la ecuación proporciona el valor rms de la componente fundamental como:

$$V_{1=} = \frac{4V_s}{\sqrt{2\pi}} = 0,9V_s$$

1.2. Expresión de Corriente en Régimen Permanente

Para el inversor monofásico de la figura 1.12, se puede apreciar que la para la configuración de media onda la tensión sobre la carga varía entre $\pm V_{DC}/2$, mientras que para el de onda completa varía ente $\pm V_{DC}$. La ecuación diferencial que describe el circuito para una carga del tipo RL viene dada por:

$$vf(t) = Ri(t) + L \frac{di(t)}{dt}$$

Donde:

$$vf(t) = \begin{cases} V_1 & 0 \leq t \leq \frac{T}{2} \\ -V_1 & 0 < t < T \end{cases}$$

Para el puente inversor de media onda $V_1 = V_{DC}/2$, mientras que para el de onda completa $V_1 = V_{DC}$.

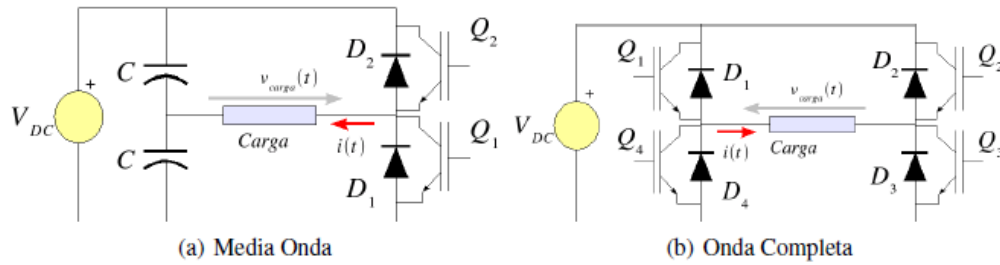


Figura 1.12. Inversor Monofásico (Bueno, 2007)

Resolviendo la ecuación diferencial anterior, para cada uno de los semiciclos obtenemos para $0 \leq t \leq \frac{T}{2}$:

$$i(t) = k_1 e^{-\frac{t}{\tau}} + \frac{V_1}{R}$$

Evaluando la condición inicial $i(0) = -I_{min}$ en la expresión anterior se obtiene:

$$i(t) = \frac{V_1}{R} \left(1 - e^{-\frac{t}{\tau}} \right) - I_{min} e^{-\frac{t}{\tau}}$$

Evaluamos la condición final del intervalo:

$$i\left(\frac{T}{2}\right) = I_{max} = \frac{V_1}{R} \left(1 - e^{-\frac{T}{2\tau}} \right) - I_{min} e^{-\frac{T}{2\tau}}$$

para $\frac{T}{2} \leq t \leq T$:

$$i(t) = k_2 e^{-\frac{t}{\tau}} - \frac{V_1}{R}$$

Evaluando la condición inicial $i\left(\frac{T}{2}\right) = I_{max}$ en la expresión anterior, se obtiene:

$$i(t) = \frac{V_1}{R} \left(e^{-\frac{t-\frac{T}{2}}{\tau}} - 1 \right) + I_{max} e^{-\frac{t-\frac{T}{2}}{\tau}}$$

Evaluamos la condición final del intervalo

$$i(T) = I_{min} = \frac{V_1}{R} \left(e^{-\frac{T}{2\tau}} - 1 \right) + I_{max} e^{-\frac{T}{2\tau}}$$

Por simetría de la onda se cumple que:

$$I_{min} = -I_{max}$$

Sustituyendo la condición de simetría en la ecuación final del intervalo, se obtiene:

$$I_{max} = \frac{v_1}{R} \left(1 - e^{-\frac{T}{2\tau}} \right) - I_{max} e^{-\frac{T}{2\tau}}$$

$$I_{max} \left(1 + e^{-\frac{T}{2\tau}} \right) = \frac{v_1}{R} \left(1 - e^{-\frac{T}{2\tau}} \right)$$

$$I_{max} = I_{min} = \frac{v_1}{R} \frac{\left(1 - e^{-\frac{T}{2\tau}} \right)}{\left(1 + e^{-\frac{T}{2\tau}} \right)}$$

1.3. Tensión efectiva

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^{T/2} V_1^2 dt} = V_1$$

1.4. Expresión series de Fourier

1.4.1. Tensión

$$v(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_1}{n\pi} \operatorname{sen}\left(\frac{2\pi nt}{T}\right)$$

1.4.2. Corriente

$$v(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_1}{n\pi} \frac{1}{Z} \operatorname{sen}\left(\frac{2\pi nt}{T} - \phi_n\right)$$

Donde:

$$Z = \sqrt{R^2 + (n\omega L)^2}$$

$$\phi_n = \arctan\left(\frac{n\omega L}{R}\right)$$

1.5. Factor de Distorsión Armónica

1.5.1 Normalización

Con respecto a los armónicos, se toma en cuenta la Regulación del CONELEC No. 004/01 del Reglamento de Suministro de Servicio Eléctrico en el Ecuador y la IEEE en la norma 519-1992, en estas se presentan algunas recomendaciones y valores mínimos en lo que se refiere al factor de distorsión armónica.

1.5.2. Regulación del CONELEC No. 004/01.

El Consejo Nacional de Electricidad considerando que es necesario asegurar un nivel satisfactorio de la prestación de los servicios eléctricos a los consumidores determina que conveniente dictar las Regulaciones sobre la Calidad del Servicio Eléctrico, cuyos aspectos respecto a la distorsión armónica se resaltan a continuación.

Para las mediciones de distorsiones armónicas, se deberá realizar mensualmente lo siguiente:

1. Un registro en cada uno de los puntos de medición, en un número equivalente al 0.15% de los transformadores de distribución, en los bornes de bajo voltaje, no menos de 5.
2. Para la elección de los puntos se consideran los niveles de voltaje, el tipo de zona, y la topología de la red, a fin de que las mediciones sean representativas de todo el sistema. Una vez realizada la selección de los puntos, la Empresa Distribuidora debe notificar al CONELEC, por lo menos 2 meses antes de efectuar las mediciones.

3. Simultáneamente con este registro se deberá medir la energía entregada a efectos de conocer la que resulta suministrada en malas condiciones de calidad.
4. En cada punto de medición, para cada mes, el registro se efectuará durante un periodo no inferior a 7 días continuos, en intervalos de medición de 10 minutos.

Con la finalidad de ubicar de una manera más eficiente los medidores de distorsiones armónicas, se efectuarán mediciones de monitoreo de armónicas, de manera simultánea con las mediciones de voltaje indicadas anteriormente; por lo que los medidores de voltaje deberán estar equipados para realizar tales mediciones de monitoreo. Los valores eficaces de los voltajes armónicos individuales y la THD, expresados como porcentaje del voltaje nominal del punto de medición respectivo, no deben superar los valores límites (%Vh y THD) señalados en la tabla 1.2.

Orden de la Armónica (h)	THD	
	V>40KV	V≤40KV
Impares no múltiplos de 3		
3	2	6
5	2	5
11	1,5	3,5
13	1,5	3,0
17	1	2,0
19	1	1,5
23	0,7	1,5
25	0,7	1,5

>25	0,1+0,6*25/h	0,2+1,3*25/h
Impares múltiplos de 3		
3	1,5	5
9	1	1,5
15	0,3	0,3
21	0,2	0,2
>21	0,2	0,2
Pares		
2	1,5	2
4	1	1
6	0,5	0,5
8	0,2	0,5
10	0,2	0,5
12	0,2	0,2
>12	0,2	0,5

Tabla 1.2. Tasa de distorsión armónica individual y THD para voltaje.

1.5.3. Norma IEEE 519-1992.

El aumento del uso de convertidores estáticos de energía, tanto en equipos de control industrial como en aplicaciones domésticas, combinado con el aumento del uso de condensadores para el mejoramiento del factor de potencia, han creado problemas generalizados. Debido a lo extenso de estos problemas, ha sido necesario desarrollar técnicas y lineamientos para la instalación de equipos y control de armónicos. Este segmento discute estos lineamientos y su importancia en el diseño de sistemas.

Las normas estadounidenses con respecto a los armónicos han sido agrupadas en la norma 519-1992, el propósito de la IEEE 519-1992 es de recomendar límites en la distorsión armónica según dos criterios distintos, específicamente:

1. Existe una limitación sobre la cantidad de corriente armónica que un consumidor puede inyectar a la red de distribución eléctrica.
2. Se establece una limitación en el nivel de voltaje armónico que una compañía de distribución de electricidad puede suministrar al consumidor.

Uno de los temas que se habla es la distorsión de la corriente de entrada y el cumplimiento con las especificaciones de armónicas. De estas especificaciones, ninguna se cita más frecuentemente que la IEEE 519-1992. La filosofía estándar que se cito en el numeral 2.2 es directa: cada consumidor individual de suministro eléctrico debe controlar su distorsión de corriente, y la compañía eléctrica podrá entonces suministrar voltaje adecuadamente, libre de distorsión a todos sus usuarios. Para lograr este objetivo, el estándar recomienda niveles de distorsión de corriente para cada usuario en base a la medida relativa de la carga para la capacidad del sistema.

La distorsión del voltaje de entrada es una función de tres factores: los niveles preexistentes de armónicos de voltaje, la impedancia del sistema de potencia en el punto de acoplamiento común y el perfil de la corriente armónica que está siendo generada por el controlador.

Los inversores monofásicos para la generación generalmente son proporcionados para menos de 10 kW y típicamente no pueden causar problemas para una compañía en números pequeños. Como su uso se incrementa, sin embargo, grandes números de inversores ligados a la misma alimentación pueden causar problemas si los armónicos de los inversores son excesivos.

Los inversores trifásicos para la generación son típicamente proporcionados desde los 10KW hasta 1MW y probablemente son, por lo menos en términos cercanos, causa de formas de onda inaceptables para la subestación alimentadora si los inversores de las formas de onda de salida contienen altos porcentajes de armónicos. Como los inversores monofásicos, los inversores trifásicos pueden ser topologías de conmutación lineal o de conmutación propia (algunas veces llamada conmutación propia).

Límites de Distorsión			
THD	Aplicaciones Especiales	Sistemas Generales	Sistemas Especiales
	3%	5%	10%

Tabla.1.3. Límites de distorsión en sistemas de bajo voltaje

Orden de la Armónica (h)	THD
1	100
3	13,78
5	5,05
7	2,59
9	1,57
11	1,05
13	0,75
15	0,57
17	0,44
19	0,35
21	0,29
23	0,24
25	0,20

Tabla.1.4. Límites de distorsión armónica individual

El factor de distorsión armónica (THD) para tensión en inversores es:

$$THD = \frac{\sqrt{v_1^2 - \left(\frac{4v_1}{n\pi}\right)^2}}{\frac{4v_1}{n\pi}} = 0,48343$$

CAPÍTULO II

INVERSORES TRIFÁSICOS

2.1. Introducción

Los inversores trifásicos se usan comúnmente en aplicaciones de alta potencia, la conexión en paralelo de tres inversores monofásicos ya sean de medio puente o completo forman un inversor Trifásico tal como puede observarse en la figura 2.1, aquí el inversor está constituido por transistores IGBT y diodos de descarga libre antiparalelo.

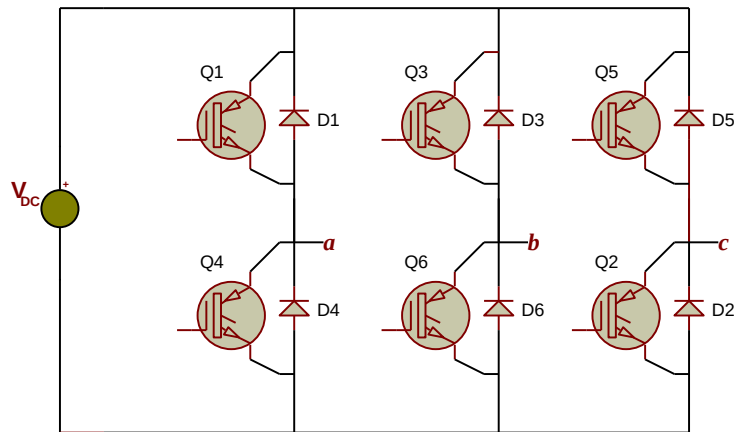


Fig. 2.1. Inversor Trifásico

El sistema trifásico generado a partir de la fuente de corriente continua debe cumplir las siguientes condiciones:

1. Las tensiones en las tres fases deben poseer igual módulo.
2. Debe existir un desfase de $\frac{2}{3}\pi$ entre las fases.
3. El sistema de tensiones debe tener una secuencia (a,b,c) o (a,c,b)
4. La suma de las tensiones en cada instante de tiempo debe ser cero:

$$(v_{ab}(t) + v_{bc}(t) + v_{ca}(t) = 0)$$

En la figura 2.2. se presenta un posible sistema de tensiones trifásicas generadas por el inversor. Analizando este sistema de tensiones, se puede evidenciar que cumple las tres primeras condiciones pero la sumatoria de tensiones instantáneas en las fases es diferente de cero.

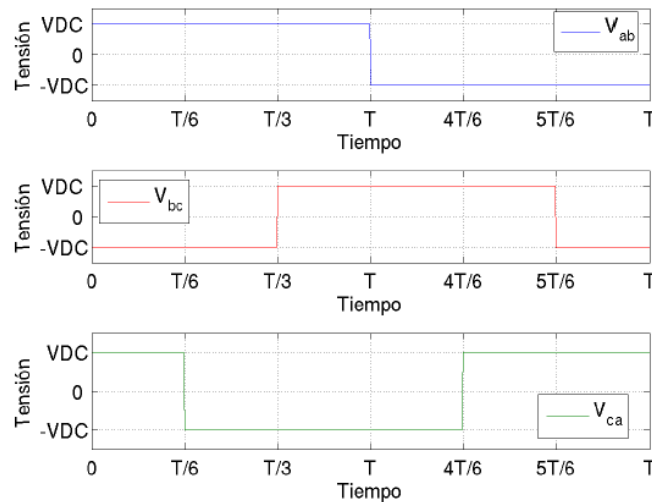


Fig. 2.2. Sistema de tensiones trifásica (Bueno, 2007)

Estudiando la secuencia de disparo de los IGBT para generara este sistema trifásico de la figura 2.2, se pude apreciar como se requiere la operación simultánea de los dos componentes pertenecientes a la misma rama, esto ocasionaría un cortocircuito en la fuente de corriente continua, razón por la cual esta forma de onda no puede ser generada por este puente convertidor.

Para cumplir la condición que la sumatoria instantánea de tensiones entre las fases sea igual a cero, las formas de onda generadas por el puente inversor no pueden poseer tercer armónico. Esto garantiza que no exista operación simultánea de dos interruptores de la misma rama. En la figura 2.3, se presenta un sistema de tensiones trifásica sin tercer armónico, con su respectiva secuencia de disparo de las componentes.

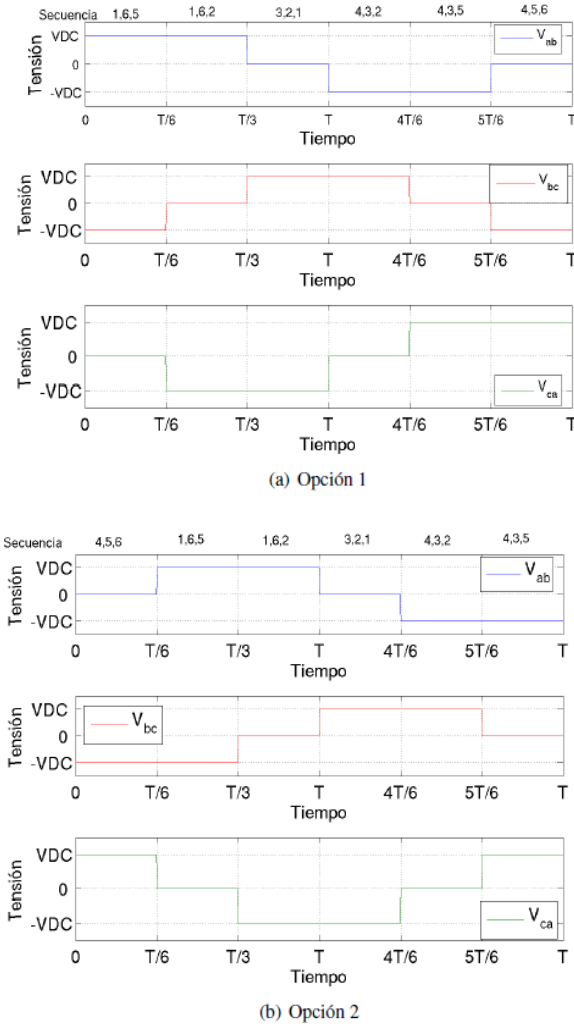


Fig. 2.3. Sistema de tensiones trifásicas sin presencia de tercer armónico (Bueno, 2007)

En la figura 2.4, se presenta el contenido armónico de la tensión v_{ab} para las formas de onda de las figuras 2.2 y 2.3. Se puede apreciar la ausencia de terceros armónicos y sus múltiplos en el contenido armónico de las dos ondas correspondientes a la figura 2.3

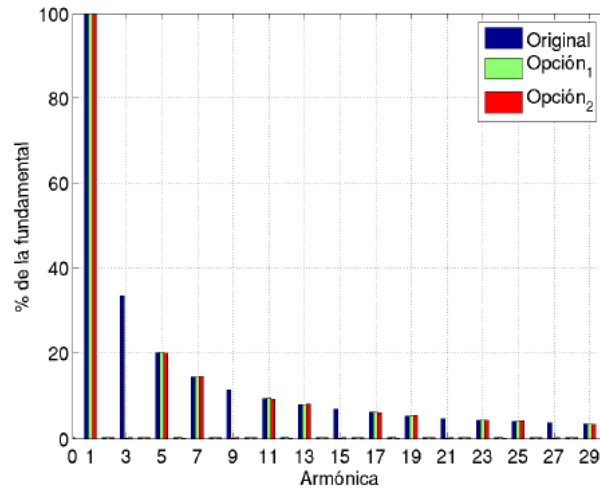


Fig. 2.4. Contenido armónico del sistema de tensiones trifásicas con y sin tercer armónico.
(Bueno, 2007)

2.2. Tensión en Series de Fourier

La expresión en series de Fourier de la tensión línea a línea del inversor trifásico sobre la carga es:

$$v_{ab}(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_{DC}}{n\pi} \cos\left(\frac{n\pi}{6}\right) \sin\left(n\left(\omega t + \frac{\pi}{6}\right)\right)$$

$$v_{bc}(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_{DC}}{n\pi} \cos\left(\frac{n\pi}{6}\right) \sin\left(n\left(\omega t + \frac{\pi}{2}\right)\right)$$

$$v_{ca}(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_{DC}}{n\pi} \cos\left(\frac{n\pi}{6}\right) \sin\left(n\left(\omega t + \frac{7\pi}{6}\right)\right)$$

Cabe denotar que para $n = 3$ y sus múltiplos los coeficientes de de la serie son iguales a cero.

2.3. Tensión Efectiva

La tensión efectiva total línea a línea sobre la carga es:

$$V_{rms} = \sqrt{\frac{2}{3}} V_{DC}$$

2.4. Factor de Distorsión Armónica Total

El factor de distorsión armónica total en tensión es:

$$THD = \frac{\sqrt{V_{rms}^2 - V_{rms1}^2}}{V_{rms1}} = 0,31084$$

2.5. Modelo en Vectores Espaciales

2.5.1. Inversor

Recordando la definición del vector espacial de tensión línea neutro:

$$\vec{v}_{fn} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} \end{bmatrix} \begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix}$$

Calculando el vector espacial de tensión aplicado por el inversor sobre la carga, a partir de las tensiones línea a línea, se obtiene:

$$\vec{v}_u = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} \end{bmatrix} \begin{bmatrix} v_{ab}(t) \\ v_{bc}(t) \\ v_{ca}(t) \end{bmatrix}$$

$$\vec{v}_u = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} \end{bmatrix} \left[\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} - \begin{bmatrix} v_b(t) \\ v_c(t) \\ v_a(t) \end{bmatrix} \right]$$

$$\vec{v}_{ll} = \left(1 - e^{j\frac{2\pi}{3}}\right) \vec{v}_{fn}$$

$$\vec{v}_{ll} = \sqrt{3} e^{j\frac{\pi}{6}} \vec{v}_{fn}$$

El resultado de la expresión anterior, es análogo al obtenido en régimen sinusoidal permanente al pasar de tensiones de línea a tensiones de fase. En la figura 2.5, se presenta el vector espacial de tensiones que aplica el inversor a la carga en por unidad de la tensión de Corriente continua V_{DC} .

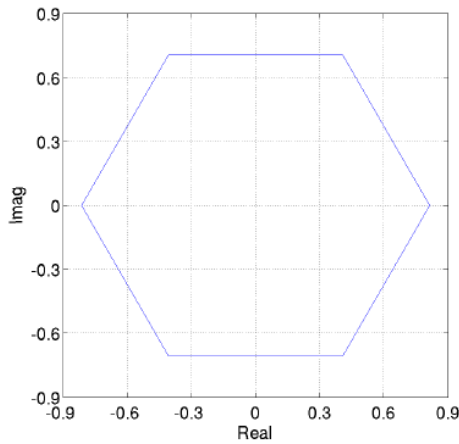


Fig. 2.5. Tensión espacial del inversor trifásico. (Bueno, 2007)

El hexágono de la figura 2.5, corresponde a cada una de las posibles combinaciones de los seis interruptores que conforman el puente inversor, respetando que los interruptores de la misma rama operan de manera complementaria entre sí, a fin de evitar cortocircuitos sobre la fuente de corriente continua.

$$Sw_4 = \overline{Sw_1}$$

$$Sw_6 = \overline{Sw_3}$$

$$Sw_2 = \overline{Sw_5}$$

Renombrando los interruptores de la figura 2.1 en función de las fases obtenemos:

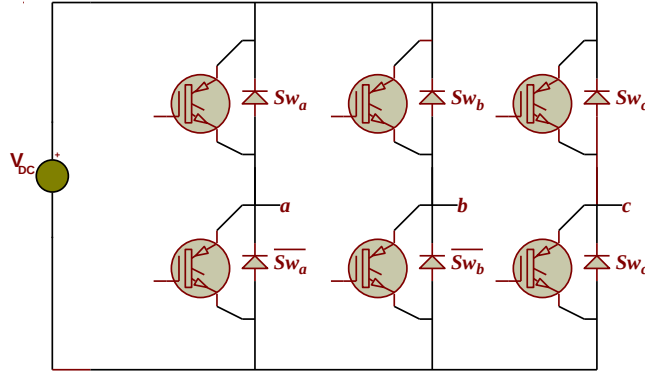


Fig. 2.6. Esquema del inversor trifásico con operación complementaria de interruptores

En la tabla 2.1, se presentan los vectores espaciales obtenidos con el inversor trifásico para cada una de las posibles combinaciones de los interruptores de la figura 2.6.

Sw_a	Sw_b	Sw_c	$\vec{v}_{fn}$
0	0	0	0
0	0	1	$-\sqrt{\frac{2}{3}} V_{DC} e^{j\frac{\pi}{3}}$
0	1	0	$-\sqrt{\frac{2}{3}} V_{DC} - e^{j\frac{\pi}{3}}$
0	1	1	$-\sqrt{\frac{2}{3}} V_{DC}$
1	0	0	$\sqrt{\frac{2}{3}} V_{DC}$
1	0	1	$\sqrt{\frac{2}{3}} V_{DC} - e^{j\frac{\pi}{3}}$
1	1	0	$\sqrt{\frac{2}{3}} V_{DC} e^{j\frac{\pi}{3}}$
1	1	1	0

Tabla 2.1. Vectores espaciales de tensiones del inversor trifásico

$$\Re(\overrightarrow{v_{fn}}) = \sqrt{\frac{2}{3}} \left(v_a(t) - \frac{1}{2}(v_b(t) + v_c(t)) \right)$$

Como el sistema no tiene neutro conectado, se tiene que:

$$v_a(t) + v_b(t) + v_c(t) = 0 \Rightarrow v_a(t) = -(v_b(t) + v_c(t))$$

Sustituyendo entre las dos ecuaciones anteriores se tiene:

$$v_a(t) = \sqrt{\frac{2}{3}} \Re(\overrightarrow{v_{fn}})$$

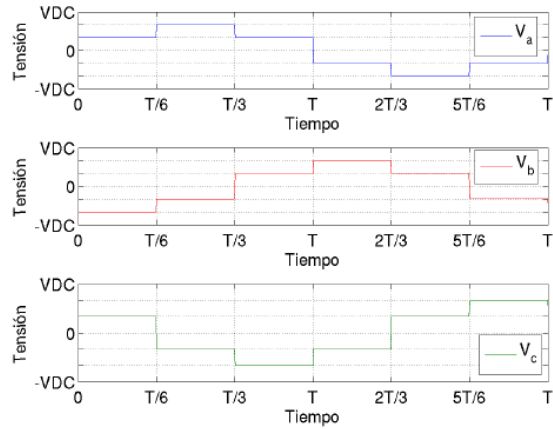
Si rotamos el vector espacial de tensión línea neutro en $e^{j\frac{4\pi}{3}}$ y utilizando un proceso análogo al utilizado en la expresión anterior se obtiene:

$$\overrightarrow{v_{fn}} e^{j\frac{4\pi}{3}} = \sqrt{\frac{2}{3}} \begin{bmatrix} e^{j\frac{4\pi}{3}} & 1 & e^{j\frac{2\pi}{3}} \end{bmatrix} \begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} \Rightarrow v_b(t) = \sqrt{\frac{2}{3}} \Re(\overrightarrow{v_{fn}} e^{j\frac{4\pi}{3}})$$

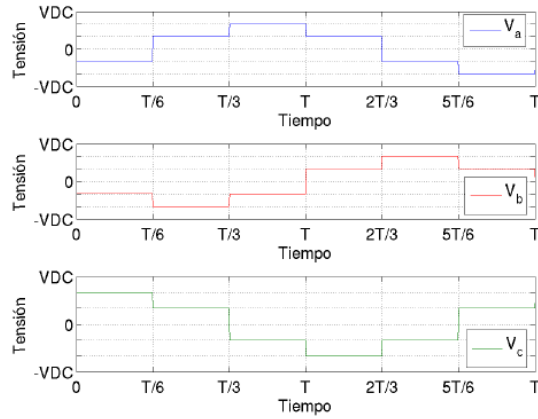
De la ecuación sin neutro conectado, se obtiene el valor de $v_c(t)$ como:

$$v_c(t) = -(v_a(t) + v_b(t))$$

En la figura 2.7., se presentan la tensión fase neutro generada por el inversor en las dos opciones de conmutación mostradas en la figura 2.3.



(a) Opción 1



(b) Opción 2

Fig. 2.7. Tensiones fase neutro del inversor trifásico. (Bueno, 2007)

En la figura 2.8, se presenta el detalle de la tensión de la fase "a" para ambas opciones.

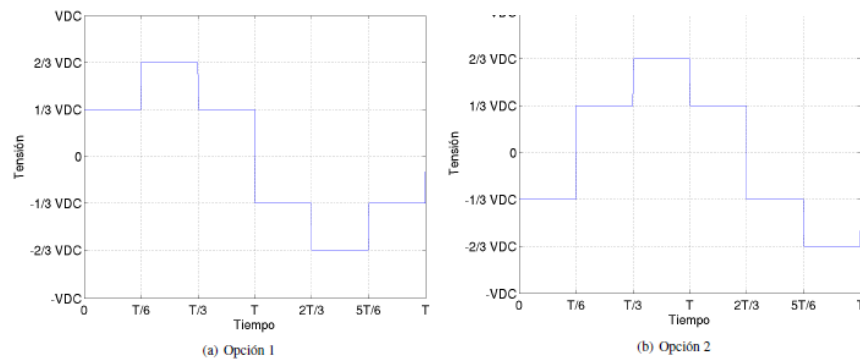


Fig. 2.8. Detalle de la tensión en la fase "a"

2.5.2. Carga

En la figura 2.9, se presenta el modelo trifásico equilibrado de una carga activa y/o pasiva conectada en delta y estrella en bornes del inversor. El modelo en vectores espaciales del inversor y la carga se puede expresar como:

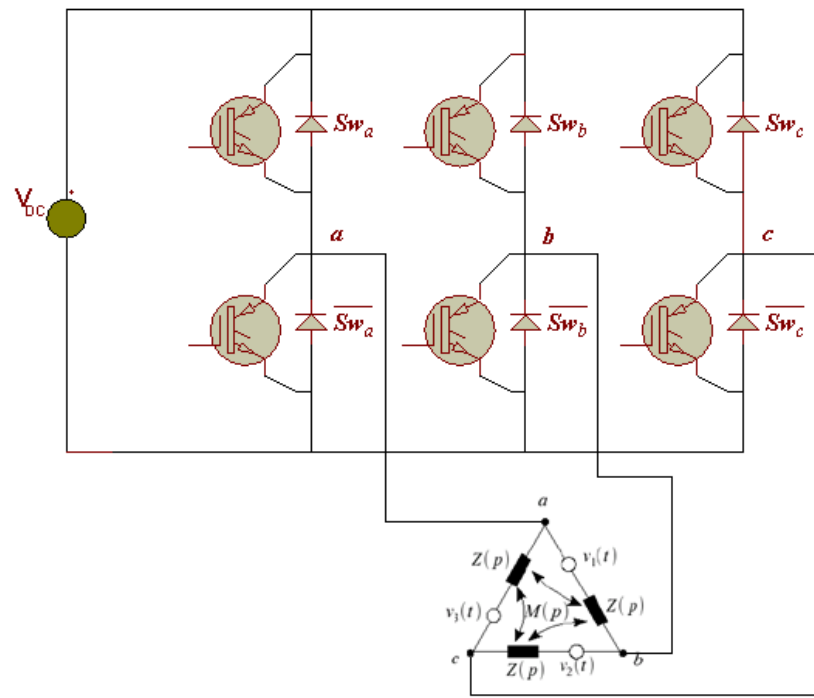
$$\overrightarrow{v_{fn}} = k \overrightarrow{e} + [Z(p) - M(p)] \vec{i}$$

donde:

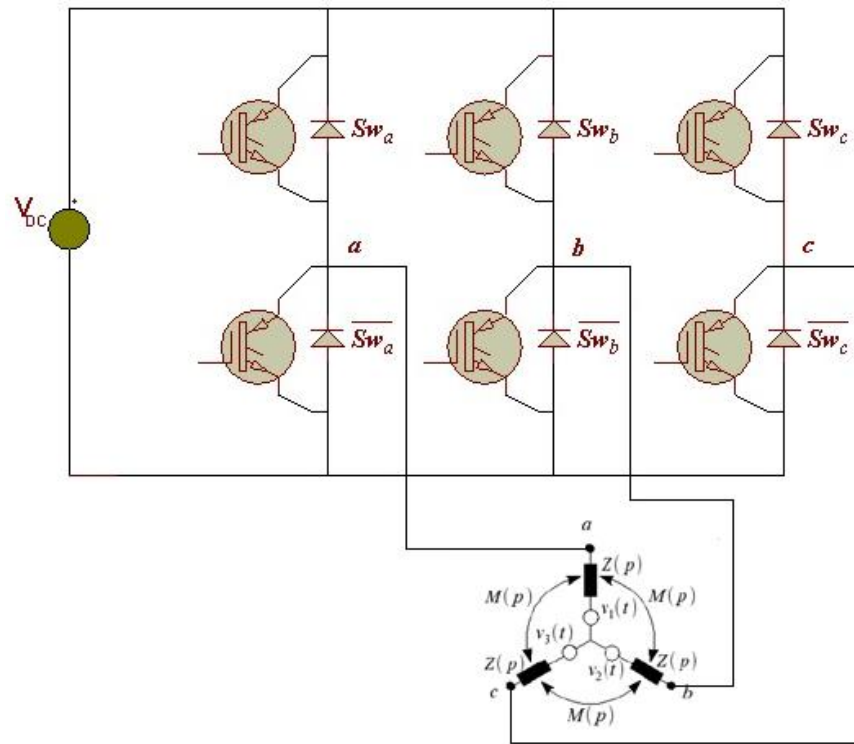
$$\overrightarrow{v_{fn}} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} \end{bmatrix} [Sw_a \ Sw_b \ Sw_c]^t$$

$$\overrightarrow{e} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} \end{bmatrix} [v_1(t) \ v_2(t) \ v_3(t)]^t$$

$$p = \frac{d}{dt}$$



(a) Delta



(b) Estrella

Fig. 2.9. Inversor con carga activa y/o pasiva trifásica

En la tabla 2.2, se muestran los valores de la impedancia operacional $Z(p)$ y $M(p)$ de la expresión de $\overrightarrow{v_{fn}}$ para los elementos resistivos, inductivos y capacitivos.

Elemento	k_Y	$Z_Y(p)$	$M_Y(p)$	k_Δ	$Z_\Delta(p)$	$M_\Delta(p)$
Resistencia	1	R	0	$\frac{e^{-j\frac{\pi}{6}}}{\sqrt{3}}$	$\frac{R}{3}$	0
Inductancia	1	L_p	M_p	$\frac{e^{-j\frac{\pi}{6}}}{\sqrt{3}}$	$\frac{L}{3}p$	$\frac{M}{3}p$
Capacitancia	1	$\frac{1}{C_p}$	0	$\frac{e^{-j\frac{\pi}{6}}}{\sqrt{3}}$	$\frac{1}{3C_p}$	0

Tabla 2.2. Impedancias operacionales en conexión estrella y delta

CAPÍTULO III

TÉCNICAS DE MODULACIÓN

3.1. Modulación por ancho de pulso PWM

En muchas aplicaciones industriales, a menudo es necesario controlar el voltaje de salida de los inversores para hacer frente a las variaciones de entrada cd, para la regulación del voltaje de los inversores y para los requisitos de control constante del voltaje y la frecuencia. El método más eficiente de controlar la ganancia(y el voltaje de salida) es incorporar en los inversores el control de modulación del ancho de pulso(PWM).

La modulación por ancho de pulso (PWM, Pulse Width Modulation) proporciona un método para disminuir el factor de distorsión armónica (THD) en la corriente que suministra el inversor a la carga. La salida de un inversor con PWM con algo de filtrado, cumple las regulaciones de distorsión armónica total más fácilmente que un inversor con salida mediante ondas cuadradas. Si bien la salida con PWM posee un contenido alto de armónicas, estas son de frecuencias elevadas lo cual facilita su filtrado y atenuación por parte de la carga.

La modulación PWM controla la amplitud de la tensión de salida utilizando diferentes formas de onda moduladoras o de referencia. Dos ventajas de esta modulación son la reducción de los requerimientos de filtrado y el control de la amplitud de la salida. Entre las desventajas podemos citar el incremento en las pérdidas del dispositivo interruptor por el mayor número de conmutaciones realizadas y una mayor complejidad de los circuitos de control.

La modulación PWM puede ser realizada de dos formas:

- Bipolar: Cuando el inversor utiliza dos estados $+V_{DC}$ y $-V_{DC}$.
- Unipolar: Cuando el inversor utiliza tres estados $+V_{DC}$, $-V_{DC}$ y 0.

En la figura 3.1, se presenta el esquema de modulación unipolar y bipolar para una onda sinusoidal de referencia y una triangular de portadora.

3.1.1. Índice de modulación de frecuencia

El índice de modulación de frecuencia m_f se define como el cociente entre la frecuencia de la portadora y de la referencia:

$$m_f = \frac{f_{portadora}}{f_{referencia}}$$

La señal de salida del PWM posee la misma frecuencia fundamental que la onda de referencia y armónicas en y alrededor de los múltiplos del índice de modulación. La escogencia de índices de modulación elevados facilita el filtrado de la onda de salida, pero incrementa las pérdidas en los dispositivos electrónicos de potencia utilizados en la conmutación.

3.1.2. Índice de modulación de amplitud

El índice de modulación de amplitud m_f se define como la relación entre la amplitud de la señal de referencia y la portadora:

$$m_f = \frac{V_{pico\ referencia}}{V_{pico\ portadora}}$$

Si $m_a \leq 1$, la amplitud de la componente fundamental de la salida del PWM es linealmente proporcional a m_a es decir:

$$V_{rms1} = \sqrt{2}m_a V_{DC}$$

De esta forma se puede controlar la amplitud de la componente de frecuencia fundamental de la salida del PWM al variar m_a . Si m_a es mayor que uno, la amplitud de la fundamental de salida se incrementa pero de forma no lineal.

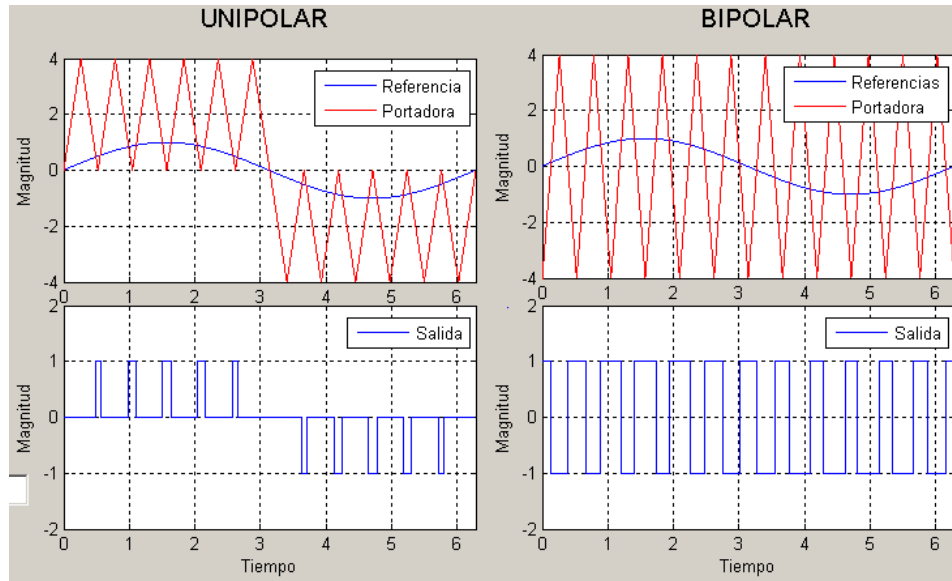


Fig. 3.1. Modulación PWM Bipolar y Unipolar

3.1.3. Contenido Armónico

En la figura 3.2, se presenta el contenido armónico de la salida del PWM unipolar y bipolar de la figura 3.1, para este caso se utilizó un índice de modulación $m_f = 12$ y $m_a = 4$.

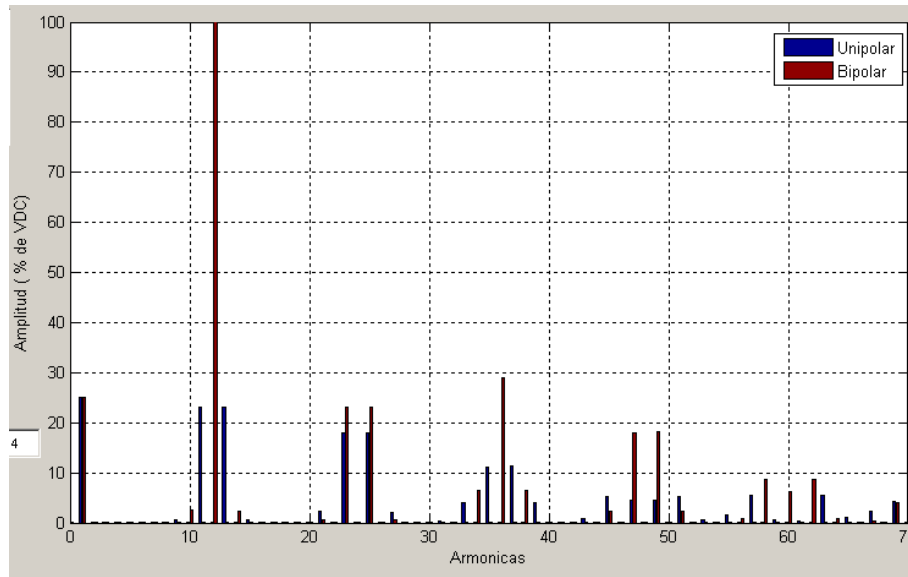


Fig. 3.2. Contenido armónico de la modulación PWM

En la figura 3.2 se puede destacar que el valor de la fundamental tanto para la salida bipolar como para la unipolar, coincide con el índice de modulación de amplitud m_a . Los mayores contenidos armónicos se localizan en los alrededores del índice de modulación de frecuencia m_f . La salida bipolar presenta mayor contenido armónico que la señal unipolar.

Si modificamos el índice de modulación de amplitud a uno ($m_a = 1$) obtendremos los resultados mostrados en la tabla 3.1 de valor efectivo en por unidad del valor de la tensión de corriente continua (V_{DC}) y distorsión armónica total para la modulación unipolar y bipolar:

	<i>Valor efectivo total</i> (V_{rms})	<i>Valor efectivo 1^{ra} armónica</i> (V_{rms1})	<i>THD</i>
Unipolar	0,7792	0,7077	0,4606
Bipolar	0,9690	0,7095	0,930

Tabla 3.1. Característica de la modulación PWM para referencia sinusoidal con $m_a = 1$

3.1.4. Algoritmo para simulación

Para simular la modulación PWM se siguió el siguiente algoritmo

- Generación de la portadora (onda triangular)
- Generación de la referencia (onda sinusoidal)
- Cálculo de la modulación entre la portadora y referencia
- Calculo del contenido armónico
- Presentación de gráficas

A continuación presentamos la programación en el m-file para la simulación de la modulación PWM

```
t=0:0.001:2*pi; %Definicion del tiempo
ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal

% Modulacion PWM

mf=handles.indicef
ma=handles.indicea

rangov=handles.rangov
co=rangov

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

% Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2);end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas
```



```

%Para Graficar
i=1
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);
axes(handles.grafical)

plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/1.5 co/1.5]); %%%Aquí
cambie
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/1.5 co/1.5]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

```

3.2. Modulación por ancho de pulso Modificada SPWM

En esta modulación se utiliza una sinusoidal como referencia pero la portadora se modifica a fin de disminuir el número de conmutaciones del puente inversor. La portadora que se utiliza varía como una diente de sierra en los extremos de cada semi ciclo de la referencia, que corresponde a los sitios donde más varía la sinusoidal mientras que en la cresta se mantiene un pulso cuadrado.

La modulación por diente de sierra se aplica en los siguientes rangos: $[0, \pi/3]$, y $[2\pi/3, 4\pi/3]$ y $[5\pi/3, 2\pi]$. En el rango $[\pi/3, 2\pi/3]$ y $[4\pi/3, 5\pi/3]$ la portadora es un pulso cuadrado. En la figura 3.3 se presenta un esquema de esta modulación para $m_a = 1$ y $m_f = 12$ en modulación unipolar y bipolar.

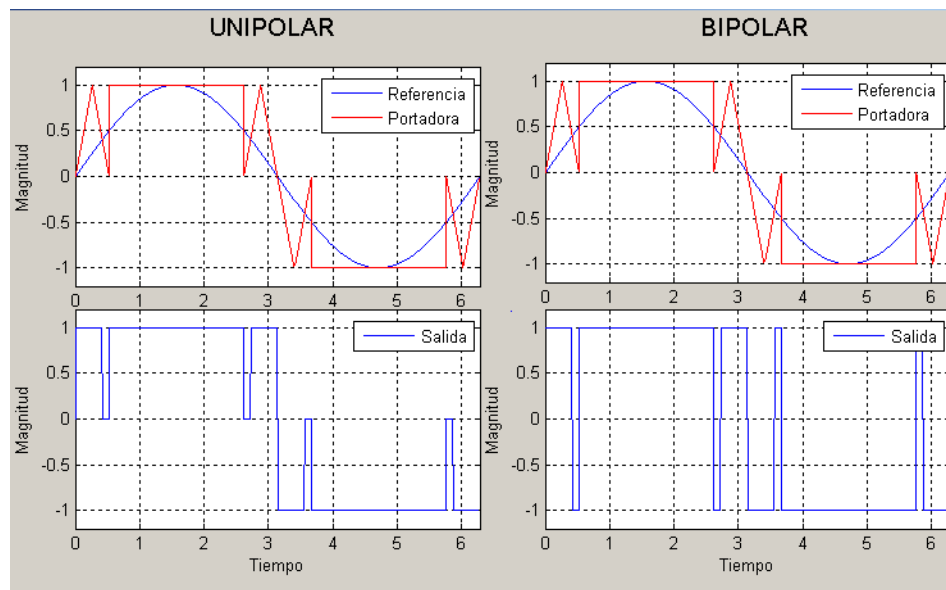


Fig. 3.3. Modulación SPWM

En la figura 3.4, se presenta el contenido armónico de la modulación por ancho de pulso modificada unipolar y bipolar de la figura 3.3.

En la tabla 3.2 se presenta el valor efectivo en por unidad del valor de la tensión de corriente continua (V_{DC}) y distorsión armónica total para la modulación SPWM unipolar y bipolar:

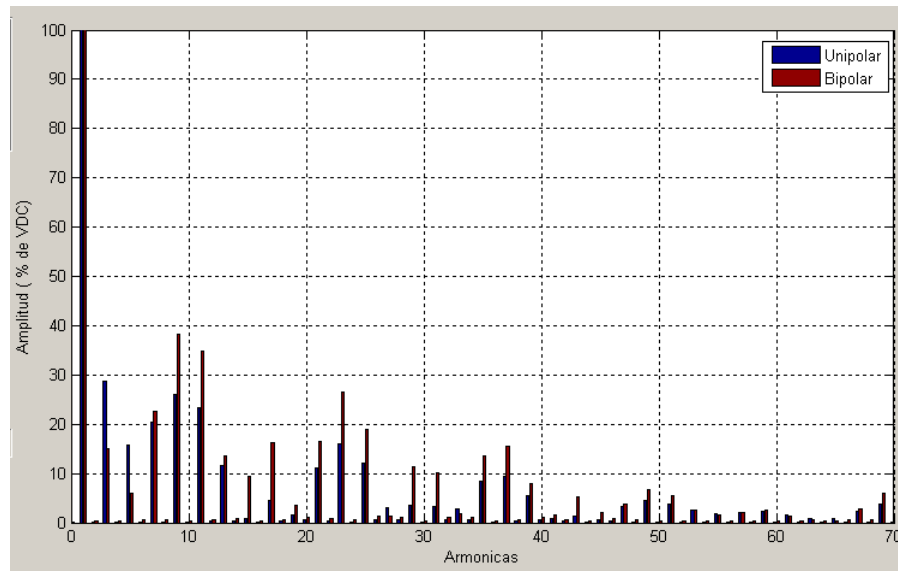


Fig. 3.4. Contenido Armónico de la modulación SPWM

	<i>Valor efectivo total (V_{rms})</i>	<i>Valor efectivo 1^{ra} armónica (V_{rms1})</i>	<i>THD</i>
Unipolar	0,9585	0,8554	0,5057
Bipolar	0,9857	0,8104	0,6923

Tabla 3.2. Característica de la modulación SPWM para referencia sinusoidal con $m_a = 1$

Esta modulación disminuye el número de conmutaciones del puente inversor disminuyendo sus pérdidas por este concepto, aumenta el valor efectivo total y de la 1ra. armónica de tensión comparado con la modulación PWM clásica y disminuye la distorsión armónica total generada por el puente convertidor.

3.2.1. Algoritmo para simulación

Para simular la modulación SPWM se siguió el siguiente algoritmo

- Generación de la portadora (onda triangular)
- Generación de la referencia (onda diente de sierra con pulso cuadrado)
- Cálculo de la modulación entre la portadora y referencia
- Cálculo del contenido armónico
- Presentación de gráficas

A continuación se presenta la programación en el m-file para la simulación de la modulación SPWM

```
mf=handles.indicef
ma=handles.indicea

rangov=handles.rangov
co=rangov

t=0:0.01:2*pi; % Definición del tiempo
ref=sin(t); % Referencia

% Modulación SPWM
PWM1=ma*(sawtooth(mf*t,.5)+1)*.5;
```

```

ang_m=30*pi/180;
ang_mod=ang_m/(2*pi)*100;
desfa1=0;
desfa2=150*pi/180;
desfa3=180*pi/180;
desfa4=330*pi/180;
Mod2=((square(t-desfa1,ang_mod)+1)*.5+(square(t-desfa2,ang_mod)+1)*.5))
Mod2=Mod2+(square(t-desfa3,ang_mod)+1)*.5;
Mod2=(Mod2+(square(t-desfa4,ang_mod)+1)*.5).*PWM1
Mod2=Mod2+(square(t-ang_m,120/3.6)+1)*.5
Mod2=(Mod2+(square(t-ang_m-pi,120/3.6)+1)*.5).*square(t,50);
Unipolar=Mod2;Bipolar=Mod2;
for i=1:length(t)
if abs(Unipolar(i))>=abs(ref(i));
Sunipolar(i)=sign(ref(i));
else
unipolar(i)=0;
end
if (Bipolar(i))>=(ref(i));
Sbipolar(i)=1;
else
Sbipolar(i)=-1;
end
end

% Calculo Contenido Armónico
Bi=fft(Sbipolar)/(length(t)/2); Bi(1)=Bi(1)/2; Bi=abs(Bi);
Ui=fft(Sunipolar)/(length(t)/2); Ui(1)=Ui(1)/2; Ui=abs(Ui);
VrmsBin=sqrt(sum(Bi(2:72).^2))/sqrt(2);
VrmsBi=sqrt(Bi(1)^2+VrmsBin^2)
THDBi=sqrt(VrmsBi^2-Bi(2)^2/2)/(Bi(2)/sqrt(2))
VrmsBil=Bi(2)/sqrt(2)
VrmsUin=sqrt(sum(Ui(2:72).^2))/sqrt(2);
VrmsUi=sqrt(Ui(1)^2+VrmsUin^2)
THDUi=sqrt(VrmsUi^2-Ui(2)^2/2)/(Ui(2)/sqrt(2))
VrmsUil=Ui(2)/sqrt(2)

%Para Graficar
i=1
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafica1)
plot(t,ref,t,Unipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref,t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

```

```
axes(handles.grafica4)
plot(t,Sbipolar); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

figure(1); clf; n=0:72;
bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 100]);
xlabel('Armonicas'); ylabel('Amplitud ( % de
VDC) ');legend('Unipolar','Bipolar')

set(handles.armonicos,'enable','on')
```

CAPÍTULO IV

TÉCNICAS AVANZADAS DE MODULACIÓN

4.1. Introducción

Aquí se aplica la modulación PWM bipolar y unipolar a diferentes ondas de referencia, desde el punto de vista de distorsión armónica total generada, valor efectivo de la señal de salida y valor efectivo de la 1ra armónica. Durante la comparación se utilizará una modulación de amplitud de uno ($m_a = 1$) y de frecuencia de doce ($m_f = 12$).

4.2. Trapezoidal

En este caso la onda de referencia es una trapezoidal, esta onda se construye a partir de un triangular la cual se recorta a partir de una amplitud específica la cual puede ser ajustada. En la figura 4.1, se presenta el esquema de esta modulación unipolar y bipolar, para esta referencia.

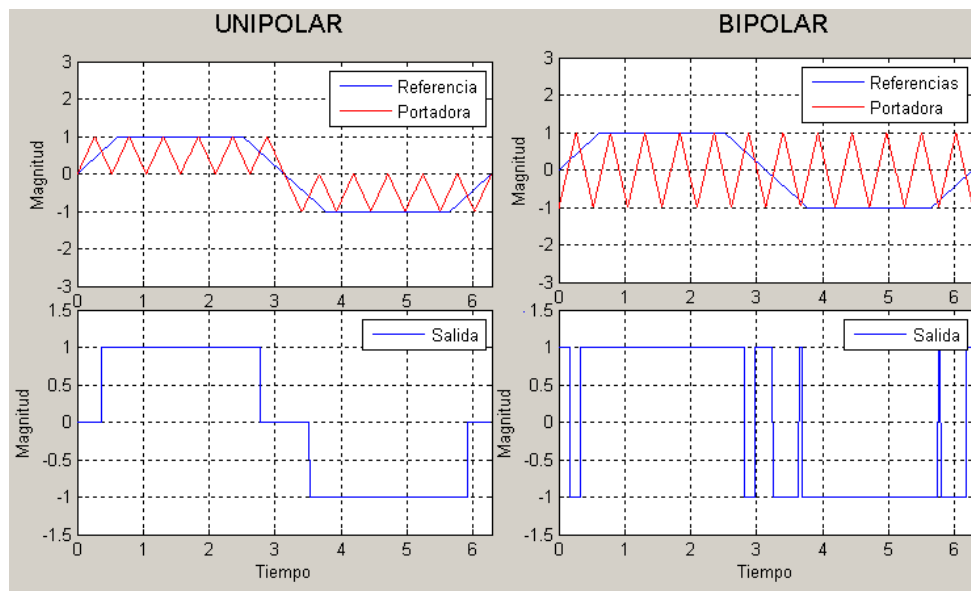


Fig. 4.1 Modulación Trapezoidal

En la figura 4.2, se presenta el contenido armónico de la modulación por ancho de pulso modificada unipolar y bipolar de la figura 4.1.

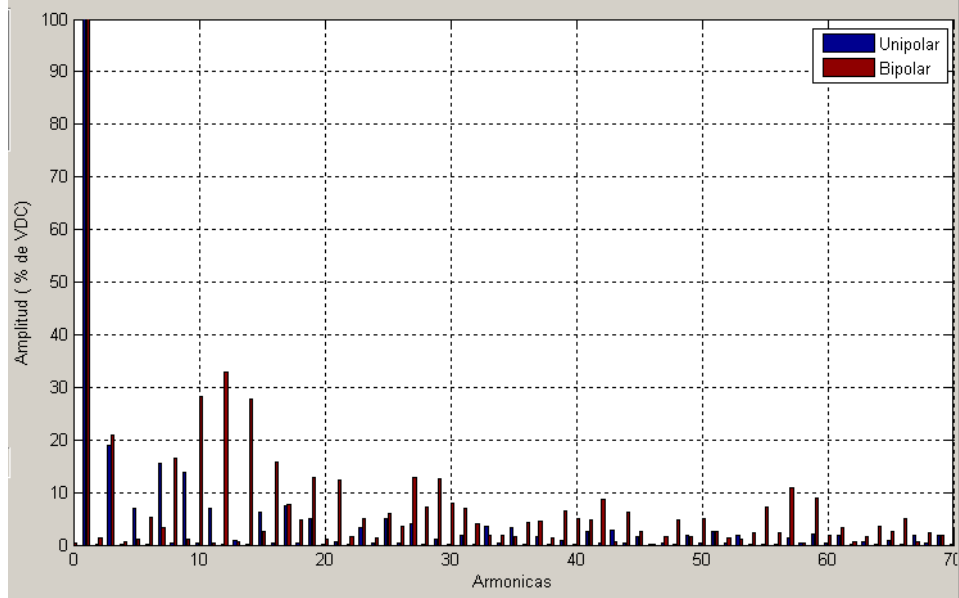


Fig. 4.2. Contenido armónico para la modulación PWM con referencia trapezoidal

En la tabla 4.1 se presenta el valor efectivo en por unidad del valor de la tensión de corriente continua (V_{DC}) y distorsión armónica total para la modulación PWM unipolar y bipolar, con referencia trapezoidal.

	<i>Valor efectivo total</i> (V_{rms})	<i>Valor efectivo 1^{ra} armónica</i> (V_{rms1})	<i>THD</i>
Unipolar	0,8729	0,8395	0,2850
Bipolar	0,9854	0,8399	0,6137

Tabla 4.1. Característica de la modulación PWM con referencia trapezoidal

4.2.1. Algoritmo para simulación

Para simular la modulación PWM con referencia trapezoidal se siguió el siguiente algoritmo:

- Generación de la portadora (onda triangular)
- Generación de la referencia (onda Trapezoidal)
- Cálculo de la modulación entre la portadora y referencia
- Calculo del contenido armónico
- Presentación de gráficas

A continuación se presenta la programación en el m-file para la simulación de la modulación PWM con referencia trapezoidal.

```
t=0:0.001:2*pi; %Definicion del tiempo

% Referencia Trapezoidal
Trap=-2.5*sawtooth(t-pi/2,.5);
for i=1:length(t);
if abs(Trap(i))>1 Trap(i)=sign(Trap(i));
end; end;
ref(2,:)=Trap;

% Modulacion PWM

mf=handles.indicef
ma=handles.indicea

rangov=handles.rangov
co=rangov

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

% Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
```



```

Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2); end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=1
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);
axes(handles.grafical)

plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/1.5 co/1.5]); %%%Aqui
cambie
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/1.5 co/1.5]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

```

4.3. Por inyección de Armónicas

En este caso la onda de referencia es una onda sinusoidal de frecuencia fundamental, con inyección de un contenido armónico específico de tercera y novena armónica. La expresión inferior, muestra el contenido armónico más utilizado en esta modulación. En la figura 4.3, se presenta el esquema de esta modulación unipolar y bipolar, para esta referencia.

$$v_{ref}(t) = 1,15 \sin(\omega t) + 0,27 \sin(3\omega t) - 0,029 \sin(9\omega t)$$

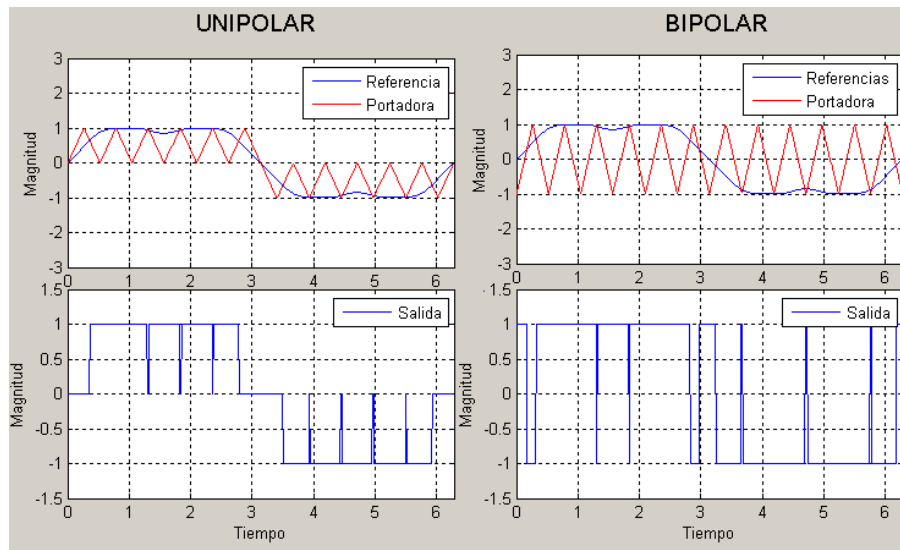


Fig. 4.3. Modulación por inyección de armónicas

En la figura 4.4, se presenta el contenido armónico de la modulación por ancho de pulso modificada unipolar y bipolar de la figura 4.3.

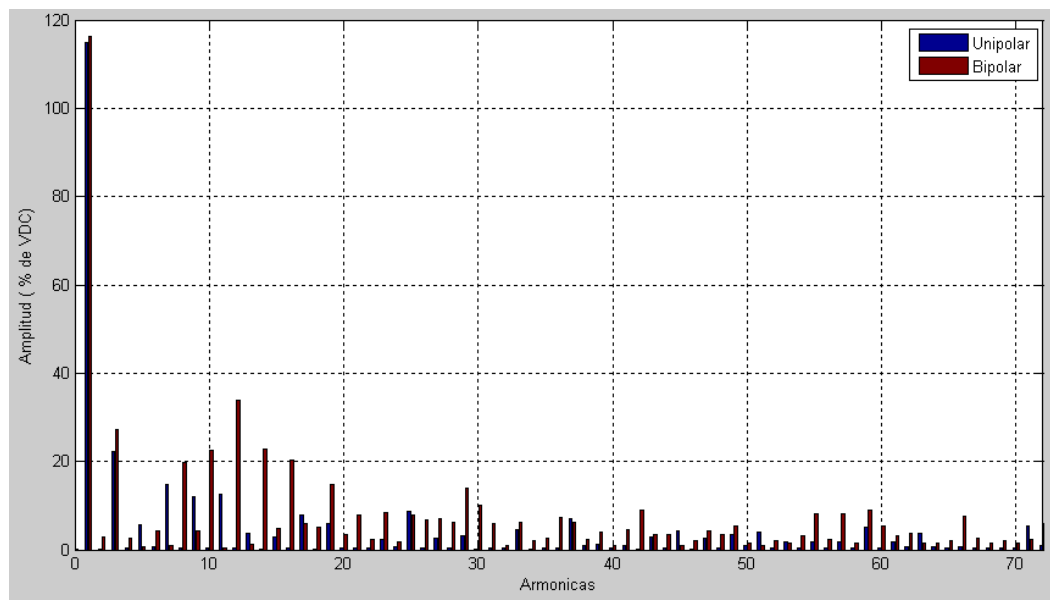


Fig. 4.4. Contenido armónico para la modulación PWM con referencia armónica

En la tabla 4.2 se presenta el valor efectivo en por unidad del valor de la tensión de corriente continua (V_{DC}) y distorsión armónica total para la modulación PWM unipolar y bipolar, con referencia por inyección de armónicas.

	<i>Valor efectivo total (V_{rms})</i>	<i>Valor efectivo 1^{ra} armónica (V_{rms1})</i>	<i>THD</i>
Unipolar	0,8576	0,8127	0,3369
Bipolar	0,9754	0,8227	0,6369

Tabla 4.2. Característica de la modulación PWM con referencia armónica

4.3.1. Algoritmo para simulación

Para simular la modulación PWM con inyección de armónicas se siguió el siguiente algoritmo:

- Generación de la portadora (onda triangular)
- Generación de la referencia (onda con armónicas)
- Cálculo de la modulación entre la portadora y referencia
- Calculo del contenido armónico
- Presentación de gráficas

A continuación se presenta la programación en el m-file para la simulación de la modulación PWM:

```
t=0:0.001:2*pi; %Definicion del tiempo
arm=1.15*sin(t)+0.27*sin(3*t)-0.029*sin(9*t);
ref(3,:)=arm; % Referencia Armónica

% Modulacion PWM

mf=handles.indicef
ma=handles.indicea

rangov=handles.rangov
co=rangov

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
```

```

Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

% Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2);end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=1
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);
axes(handles.grafical)

plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/1.5 co/1.5]); %%%Aquí
cambie
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/1.5 co/1.5]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

```

4.4. Escalera

Esta modulación aproxima una referencia sinusoidal por niveles o peldaños, generalmente se utilizan de dos a cuatro peldaños en las aproximaciones. Los niveles de los escalones se calculan para eliminar armónicas específicas y para

cada número de niveles se recomienda un índice de modulación de frecuencia específico m_f . Para obtener un valor elevado de la fundamental con baja distorsión armónica se recomienda los siguientes índices de modulación:

- Dos niveles: $m_f = 15$
- Tres niveles: $m_f = 21$
- Cuatro niveles: $m_f = 27$

A continuación observaremos las formas de onda y contenido armónico para aproximaciones de dos, tres y cuatro niveles con los índices de modulación recomendados, para la modulación PWM bipolar y unipolar.

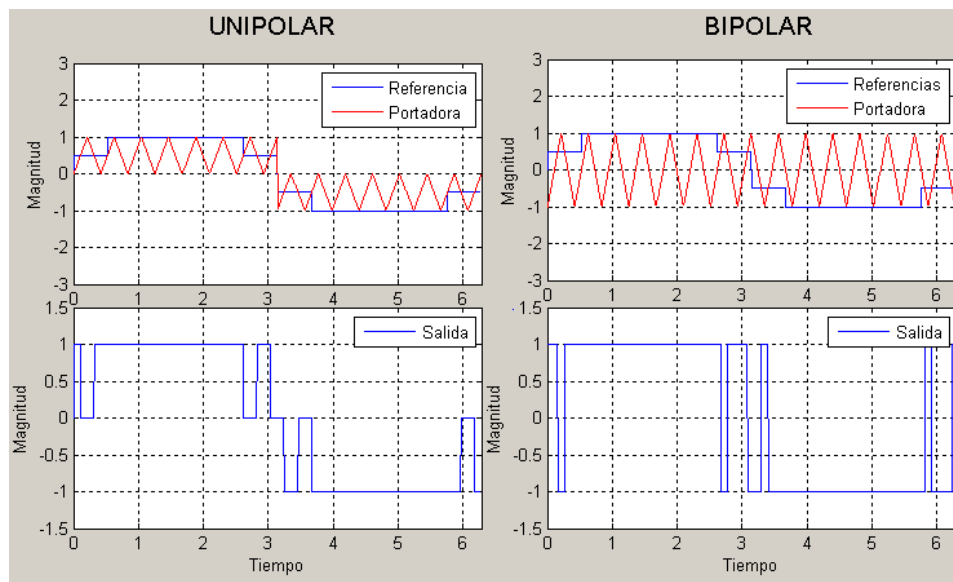


Fig. 4.5. Modulación escalera dos niveles

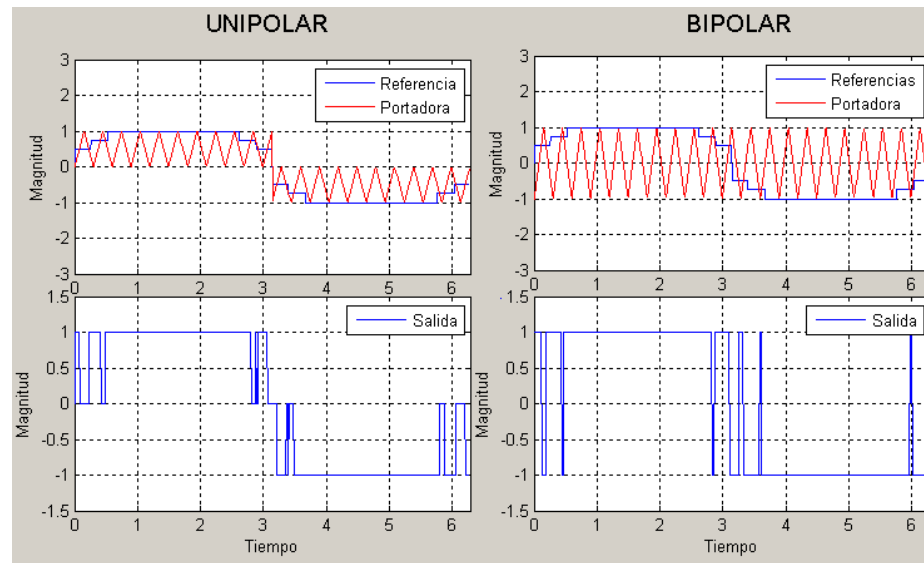


Fig. 4.6. Modulación escalera tres niveles

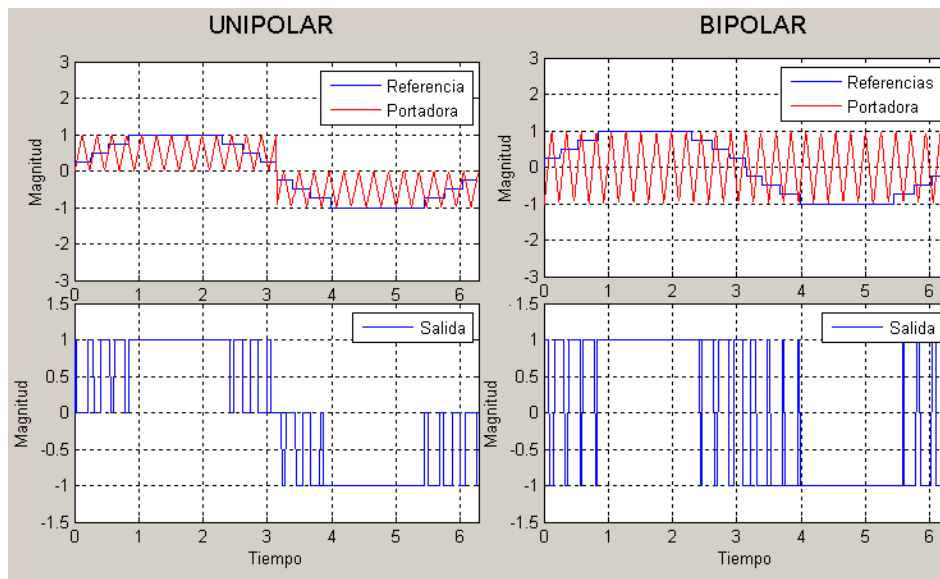


Fig. 4.7. Modulación escalera cuatro niveles

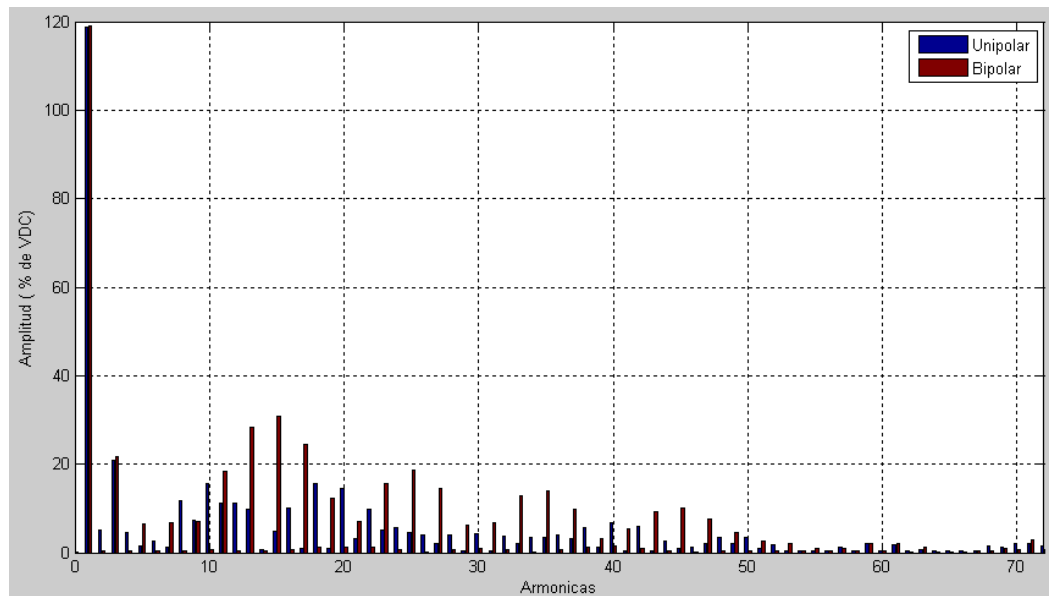


Fig. 4.8. Contenido Armónico de la modulación escalera dos niveles

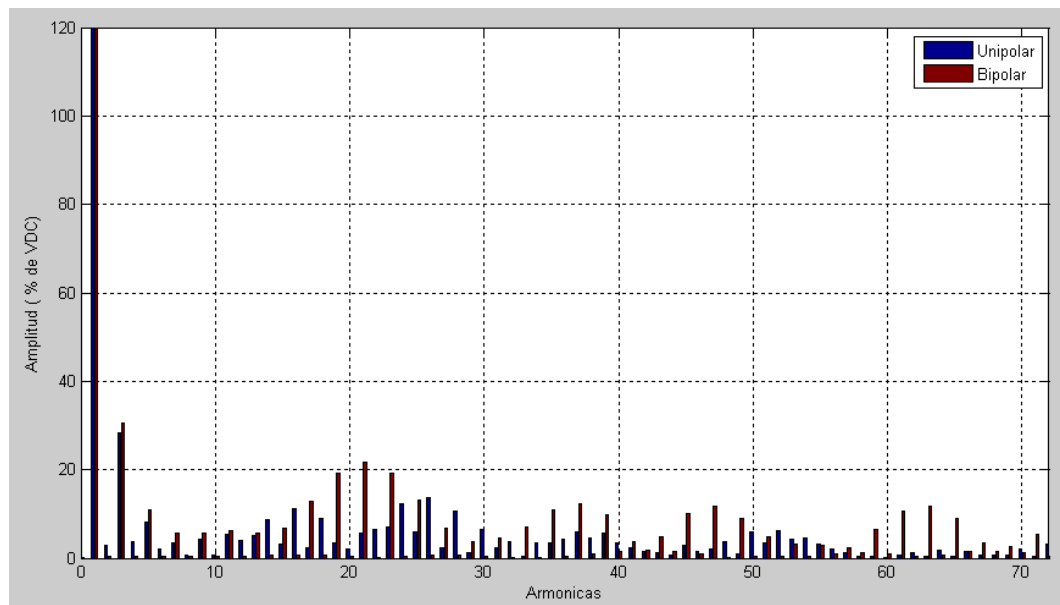


Fig. 4.9. Contenido Armónico de la modulación escalera tres niveles

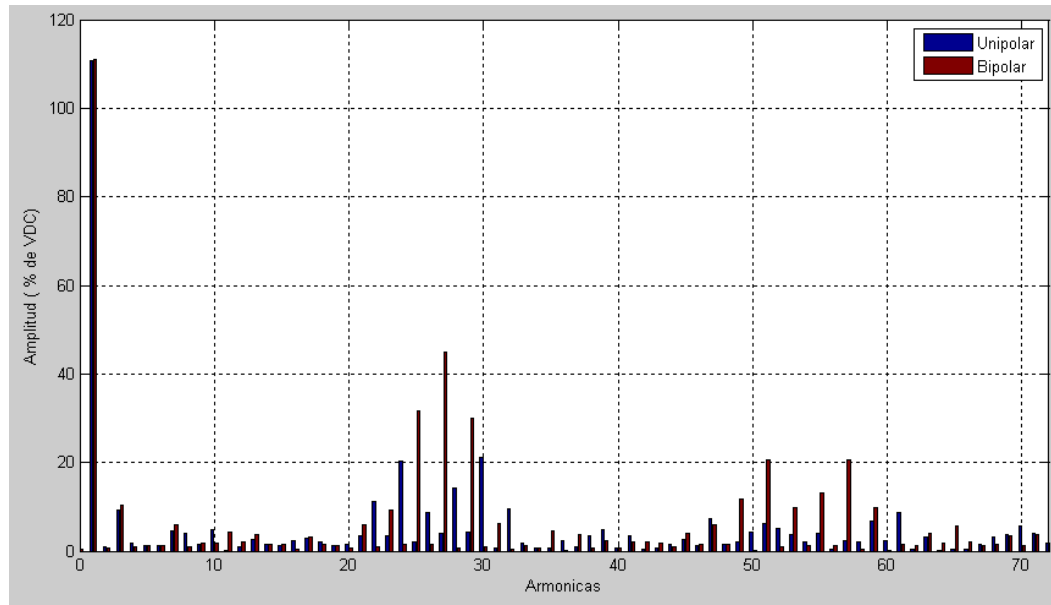


Fig. 4.10. Contenido Armónico de la modulación escalera cuatro niveles

En la tabla 4.3 se presenta el valor efectivo en por unidad del valor de la tensión de corriente continua (V_{DC}) y distorsión armónica total para la modulación PWM unipolar y bipolar, con referencia escalonada para dos, tres y cuatro niveles.

		Dos Niveles	Tres Niveles	Cuatro Niveles
Índice de modulación de frecuencia m_f		15	21	27
Unipolar	<i>Valor efectivo total (V_{rms})</i>	0,9068	0,9261	0,8461
	<i>Valor efectivo 1^{ra} armónica (V_{rms1})</i>	0,8394	0,8614	0,7825
	<i>THD</i>	0,4089	0,3947	0,4112
Bipolar	<i>Valor efectivo total (V_{rms})</i>	0,9846	0,9799	0,9531
	<i>Valor efectivo 1^{ra} armónica (V_{rms1})</i>	0,8425	0,8680	0,7850
	<i>THD</i>	0,6048	0,5239	0,6885

Tabla 4.3. Característica de la modulación PWM para referencia escalonada

4.4.1. Algoritmo para simulación

Para simular la modulación PWM para referencia escalonada se siguió el siguiente algoritmo:

- Generación de la portadora (onda triangular)
- Generación de la referencia1 (escalera 2 peldaños)
- Generación de la referencia2 (escalera 3 peldaños)
- Generación de la referencia3 (escalera 4 peldaños)
- Cálculo de la modulación entre la portadora y referencia
- Calculo del contenido armónico
- Presentación de gráficas

A continuación se presenta la programación en el m-file para la simulación de la modulación PWM para referencia escalonada:

```
t=0:0.001:2*pi; %Definicion del tiempo
ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
% Referencia Escalera de 2 peldaños
for i=1:length(t)
if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
else y2(i)=sign(sin(t(i))); end; end;
ref(4,:)=y2;
% Escalera 3 peldaños
for i=1:length(t)
if abs(y(i))<.25; y3(i)=sign(y(i))*.5;
else; if abs(y(i))<.5; y3(i)=sign(y(i))*.75;
else; y3(i)=sign(y(i)); end; end; end;
ref(5,:)=y3;
% Escalera 4 peldaños
for i=1:length(t)
if abs(y(i))<.25; y4(i)=sign(y(i))*.25;
else; if abs(y(i))<.5; y4(i)=sign(y(i))*.5;
else; if abs(y(i))<.75; y4(i)=sign(y(i))*.75;
else; y4(i)=sign(y(i)); end; end; end; end;
ref(6,:)=y4;
y5=(round(y*10))/10; ref(7,:)=y5;

% Modulacion PWM

mf=handles.indicef
ma=handles.indicea
```

```

rangov=handles.rangov
co=rangov

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWMl=square(t,50); Unipolar=ma*PWMl.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

% Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2);end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=1
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);
axes(handles.grafical)

plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/1.5 co/1.5]); %%%Aquí
cambie
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/1.5 co/1.5]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

```

4.5. Por Pasos

La modulación por pasos consiste en aproximar una onda sinusoidal de referencia por niveles, esta discretización se realiza cada $\pi/9$. En la figura 4.11, se presenta la forma de onda para la modulación PWM unipolar y bipolar con un índice de modulación de amplitud de uno y de frecuencia de doce.

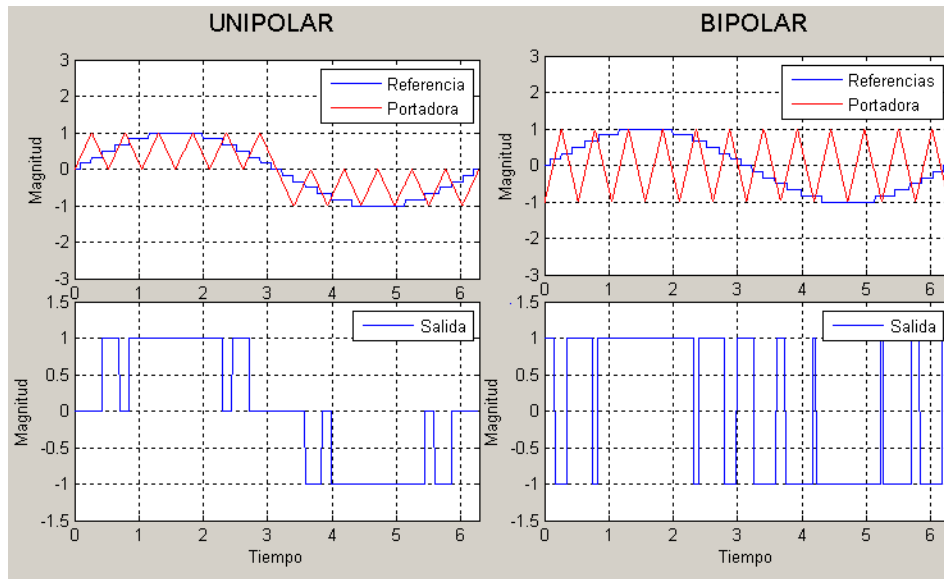


Fig. 4.11. Modulación PWM con referencia a pasos

En la figura 4.12, se presenta el contenido armónico de la modulación por ancho de pulso modificada unipolar y bipolar de la figura 4.11.

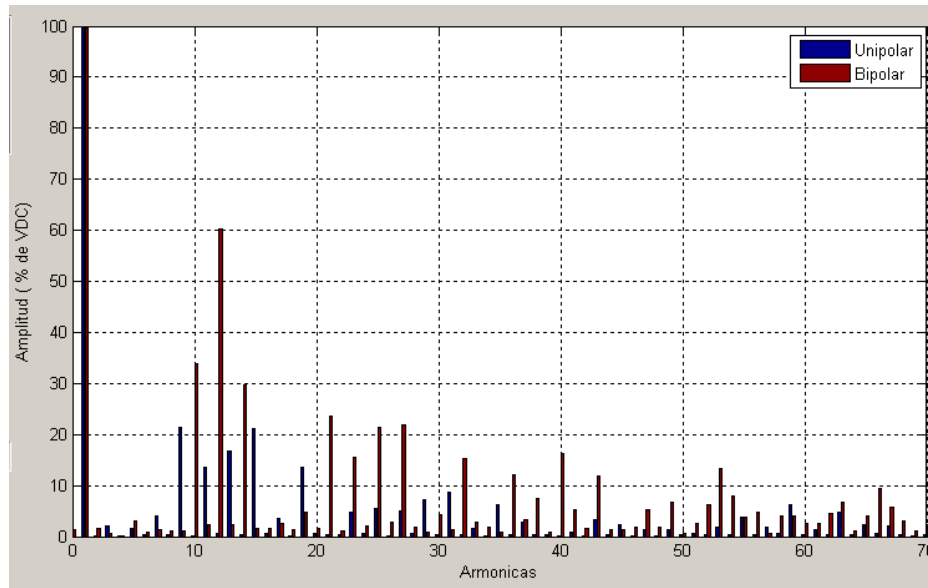


Fig. 4.12. Contenido armónico para la modulación PWM con referencia a pasos

En la tabla 4.4 se presenta el valor efectivo en por unidad del valor de la tensión de corriente continua (V_{DC}) y distorsión armónica total para la modulación PWM unipolar y bipolar, con referencia por pasos.

	<i>Valor efectivo total</i> (V_{rn})	<i>Valor efectivo 1^{ra} armónica</i> (V_{rms})	<i>THD</i>
Unipolar	0,7875	0,7197	0,4443
Bipolar	0,9736	0,7177	0,9166

Tabla 4.4. Característica de la modulación PWM con referencia a pasos

4.5.1. Algoritmo para la simulación

Para simular la modulación PWM con referencia a pasos se siguió el siguiente algoritmo

- Generación de la portadora (onda triangular)
- Determinar numero de pasos
- Generación de la referencia (onda sinusoidal)
- Cálculo de la modulación entre la portadora y referencia
- Calculo del contenido armónico
- Presentación de gráficas

A continuación se presenta la programación en el m-file para la simulación de la modulación PWM con referencia a pasos:

```

numpasos=get(handles.npasos,'String')
numpasos=str2double(numpasos)
if numpasos==0 %%%%%%%%%%el end de este if esta al ultimo de case8
errordlg({'Por favor escriba','el numero de pasos'},'MODULACIONES
by Willy')
else
numpasos=get(handles.npasos,'String')
numpasos=str2double(numpasos)

t=0:0.01:2*pi; % Definición del tiempo
ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
y5=(round(y*numpasos))/numpasos; ref(7,:)=y5; % Escalonada

%Modulacion PWM

mf=handles.indicef
ma=handles.indicea
rangoa=get(handles.rangoa,'String')
rangoa=str2double(rangoa)
co=rangoa

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

%Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2);end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

```

```

%Para Graficar

i=7
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafica1)
plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

    end %este es el end de el if del inicio de case8

```

En la tabla 4.5, se presenta un resumen de las modulaciones PWM unipolar y bipolar presentadas en este capítulo para un índice de modulación de amplitud de uno ($m_a = 1$) y de frecuencia de doce ($m_f = 12$). A excepción de la modulación escalera la cual se realizó con los índices de modulación de frecuencia recomendados según el número de niveles.

<i>Referencia de la Modulación</i>	<i>Unipolar</i>			<i>Bipolar</i>		
	V_{rms}	V_{rms1}	THD	V_{rms}	V_{rms1}	THD
<i>Sinusoidal</i>	0,7792	0,7077	0,4606	0,9690	0,7095	0,9300
<i>SPWM</i>	0,9585	0,8554	0,5057	0,9857	0,8104	0,6923
<i>Trapezoidal</i>	0,8729	0,8395	0,2850	0,9854	0,8399	0,6137
<i>Por inyección de armónicas</i>	0,8576	0,8127	0,3369	0,9754	0,8227	0,6369
<i>Escalera (2 niveles)</i>	0,9068	0,8394	0,4089	0,9846	0,8425	0,6048
<i>Escalera (3 niveles)</i>	0,9261	0,8614	0,3947	0,9799	0,8680	0,5239
<i>Escalera (4 niveles)</i>	0,8461	0,7825	0,4112	0,9531	0,7850	0,6885
<i>Por Pasos</i>	0,7875	0,7197	0,4443	0,9736	0,7177	0,9166

Tabla 4.5. Modulaciones PWM para las diferentes referencias

En la tabla anterior se puede observar como la modulación unipolar presenta menor distorsión armónica a la carga que la bipolar, pero esto requiere un puente inversor de tres estados. Ambas modulaciones obtienen valor efectivos de primera armónica similares. La escogencia de una referencia específica depende de los estados que maneje el puente convertidor y la máxima frecuencia de conmutación que soporten las componentes para los requerimientos de la carga.

ANEXOS

A.1. MANUAL DEL PROGRAMA

Para simular las modulaciones PWM, SPWM y las Técnicas Avanzadas de Modulación se desarrollo el software “Listo1”, este fue desarrollado en Matlab 2008 y se usó para generar las gráficas presentadas en los capítulos 3 y 4. En la figura A.1. se presenta la pantalla principal del programa.

Para que el software se ejecute se debe tener instalado Matlab en la PC, además se recomienda crear un carpeta con cualquier nombre en el disco duro C:, y en esta copiar el archivo .m y .fig del programa. No se debe ejecutar desde el cd, si no hacemos esto no podremos grabar las imágenes.

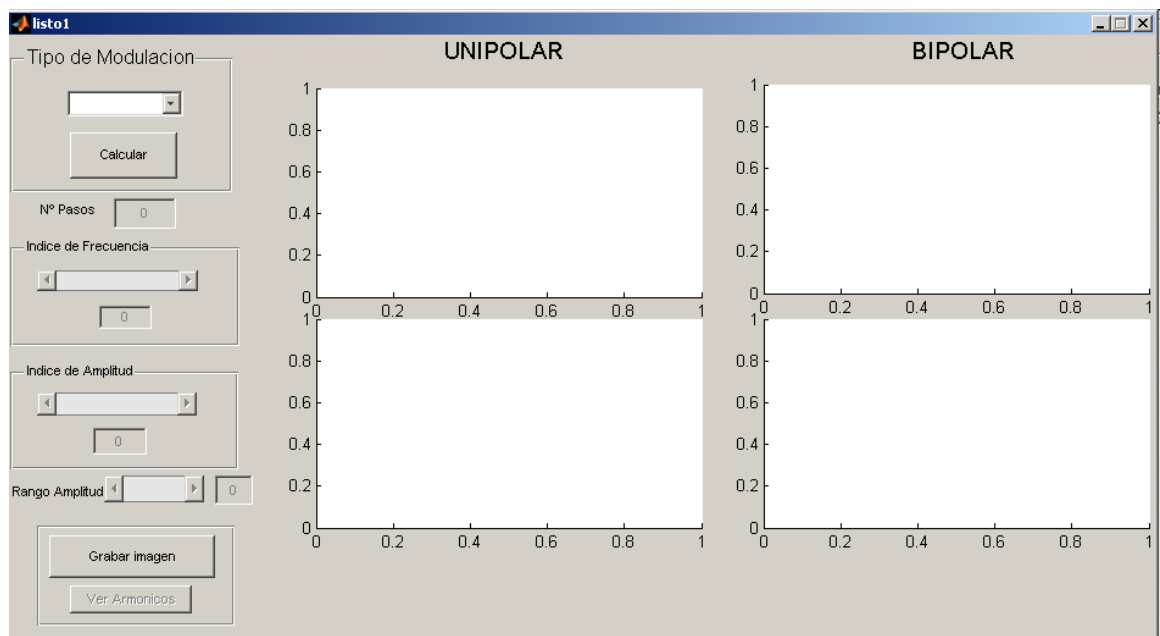


Fig. A.1. Ventana Principal del Software de Simulación “Listo1”

A.1.1. Partes y componentes del programa

Para una mejor interfaz de usuario, el funcionamiento del programa se lo dividió en tres bloques:

- Ingreso de datos
- Calculo
- Presentación gráfica

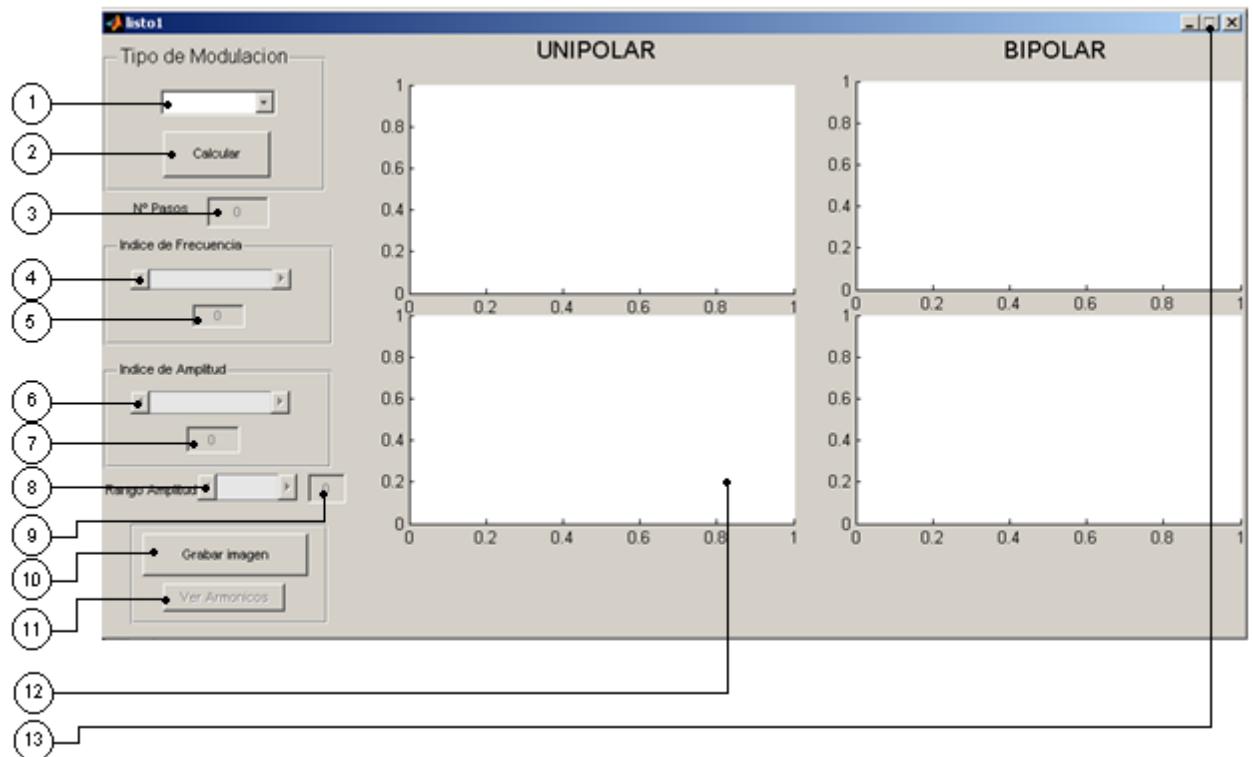


Fig. A.2. Partes y componentes del Programa

Cada una de estas partes consta de Botones (Button), Barras deslizantes (Slider), Cuadros de texto (Edit Text), Listas (List Box) y Cuadros de gráfico (Axes) para presentar gráficos, los cuales están numerados en la figura A.2., a continuación presentamos el nombre de cada uno de estos y su función:

1. *Lista de Tipos de Modulación:* Aquí puedo escoger el tipo de modulación que deseo simular.
2. *Botón Calcular:* Al pulsar el programa se encarga de realizar los cálculos de la simulación y presentar los gráficos en los axes correspondientes.
3. *Cuadro de Texto Número de pasos:* Cuando escojo la modulación por pasos este cuadro se activa y debo escribir el número de pasos para la simulación.
4. *Barra deslizante Índice de frecuencia:* Esta barra nos sirve para escoger el índice de frecuencia deseado para la simulación.
5. *Cuadro de texto Índice de frecuencia:* Aquí se nos presenta el valor del índice de frecuencia escogido en la barra anterior.
6. *Barra deslizante Índice de amplitud:* Esta barra nos sirve para escoger el índice de amplitud deseado para la simulación.
7. *Cuadro de texto Índice de amplitud:* Aquí se nos presenta el valor del índice de frecuencia escogido en la barra anterior.
8. *Barra deslizante Rango de amplitud:* En esta podemos determinar el rango del voltaje en las gráficas y tener una mejor visualización de las mismas.
9. *Cuadro de texto Rango de amplitud:* Aquí se nos presenta el valor del rango de amplitud escogido.
10. *Botón Grabar:* Nos sirve para grabar la imagen de la simulación ya sea de las modulaciones o de los armónicos.
11. *Botón Armónicos:* Se usa para visualizar gráficamente el contenido armónico de la modulación simulada.
12. *Cuadros de gráficos:* Aquí se presentan los gráficos de la simulación.
13. *Botones generales:* Con estos puedo minimizar o cerrar el programa.

A.1.2. Funcionamiento del programa

En primera instancia debemos escoger el tipo de modulación (Fig. A.3), luego de esto determinaremos el índice de frecuencia, amplitud y rango de amplitud con las barras deslizantes respectivas, los valores escogidos podrán verse en los

cuadros de texto de cada barra, es importante revisar que estos valores sean diferentes de cero pues no se podrá realizar la simulación.



Fig. A.3. Escogiendo el tipo de modulación

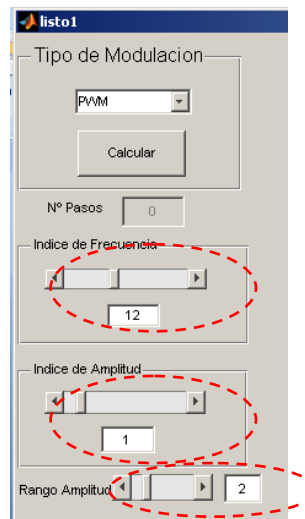


Fig. A.4. Determinando índices de frecuencia, amplitud y rango de amplitud

Después de haber realizado los pasos anteriores podemos presionar el botón Calcular, enseguida vamos a ver como se nos realizan las gráficas de la modulación unipolar y bipolar tal como se ve en la figura A.5.

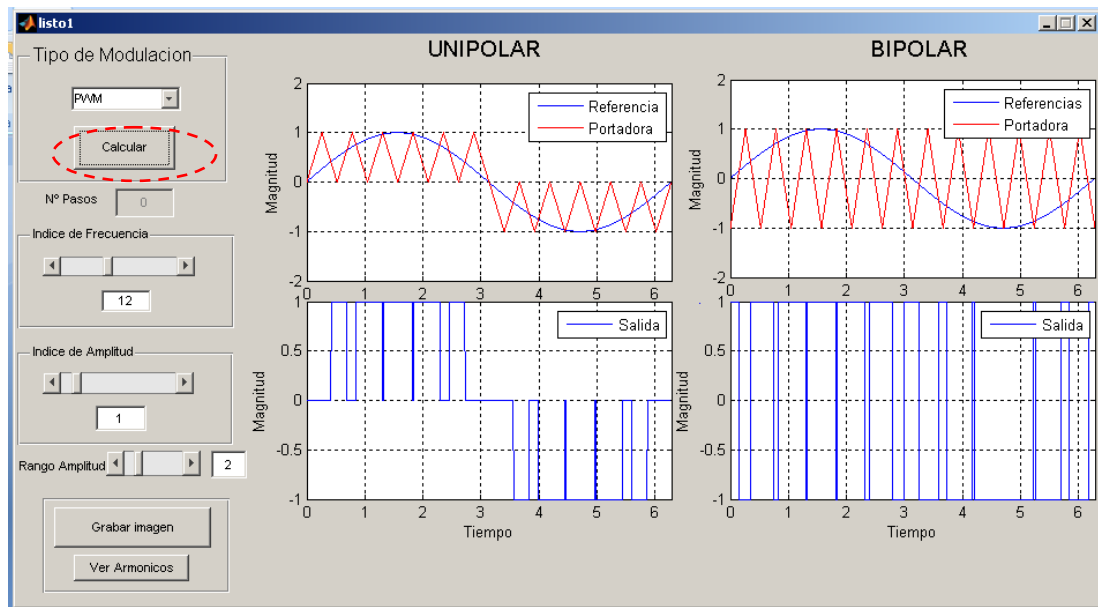


Fig. A.5. Simulación de modulación PWM con $m_f = 12$ y $m_a = 1$

Ahora como puede ser necesario tener un registro de la simulación realizada, se implemento la opción de grabar la imagen generada, para esto debemos hacer click en el botón grabar y se nos abrirá una ventana similar a la presentada en la figura A.6.

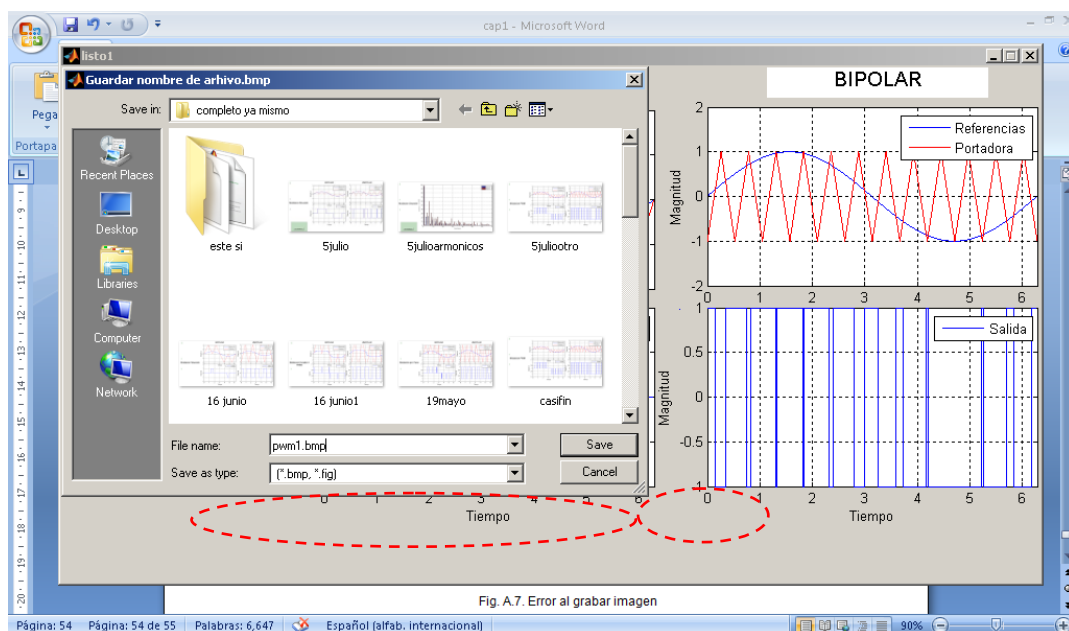


Fig. A.6. Grabando la simulación

En File name debemos colocar el nombre con el que deseamos guardar a la gráfica por ejemplo pwm1.bmp, si observa será importante color el **.bmp** que no es más que la extensión del archivo gráfico; si es que solamente ponemos el nombre del archivo “pwm1” nos va a salir el siguiente cuadro de error:



Fig. A.7. Error al grabar imagen

Así que debemos colocar la extensión .bmp para no tener este inconveniente, si por el contrario no colocamos nada en File name, se nos pedirá que coloquemos el nombre del archivo otra vez.

Por default el lugar donde se guardarán las imágenes es en la carpeta donde están los archivos .m y .fig del programa.

También podemos visualizar el contenido armónico de la modulación presionando el botón Armónicos, en este caso se borrarán las gráficas anteriores y se nos presentará lo observado en la figura A.8.

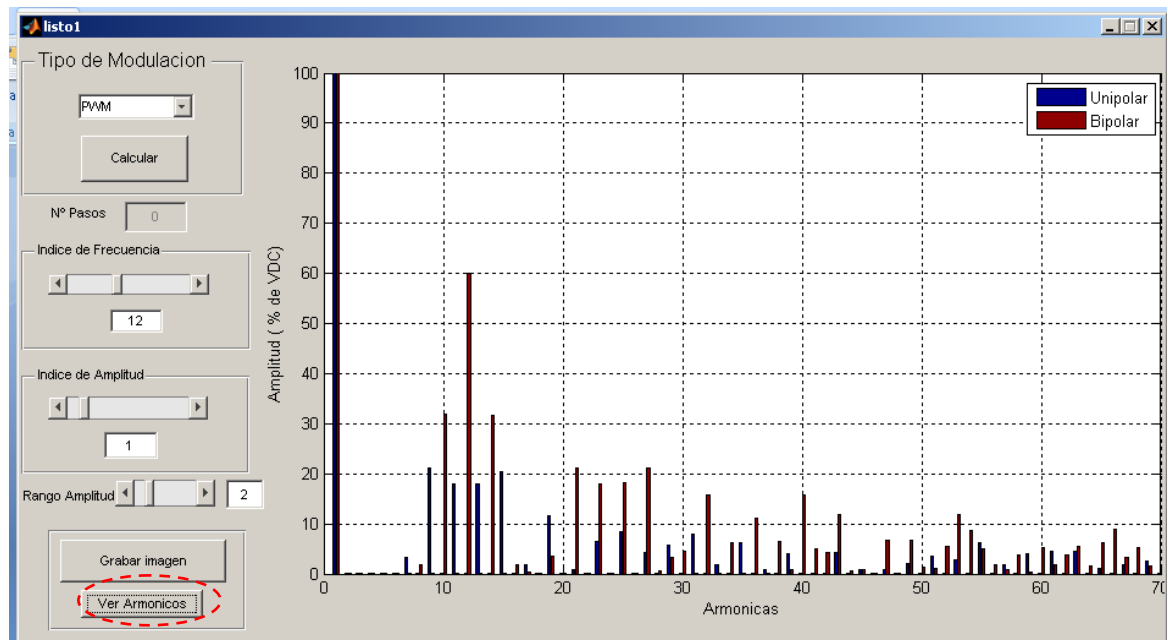


Fig. A.8. Presentación del contenido armónico de la modulación.

Si es que deseamos realizar otra simulación, no existe ningún inconveniente, puedo escoger el tipo de modulación deseada y continuar con los pasos descritos anteriormente.

Existen algunos aspectos importantes que se han implementado para mejorar la interfaz gráfica con el usuario, los cuales se describen a continuación:

- a. Si es que no escojo el tipo de modulación, es decir se mantiene en blanco el cuadro tipo de modulación, y pulso el boton calcular el programa nos presentará el siguiente mensaje:

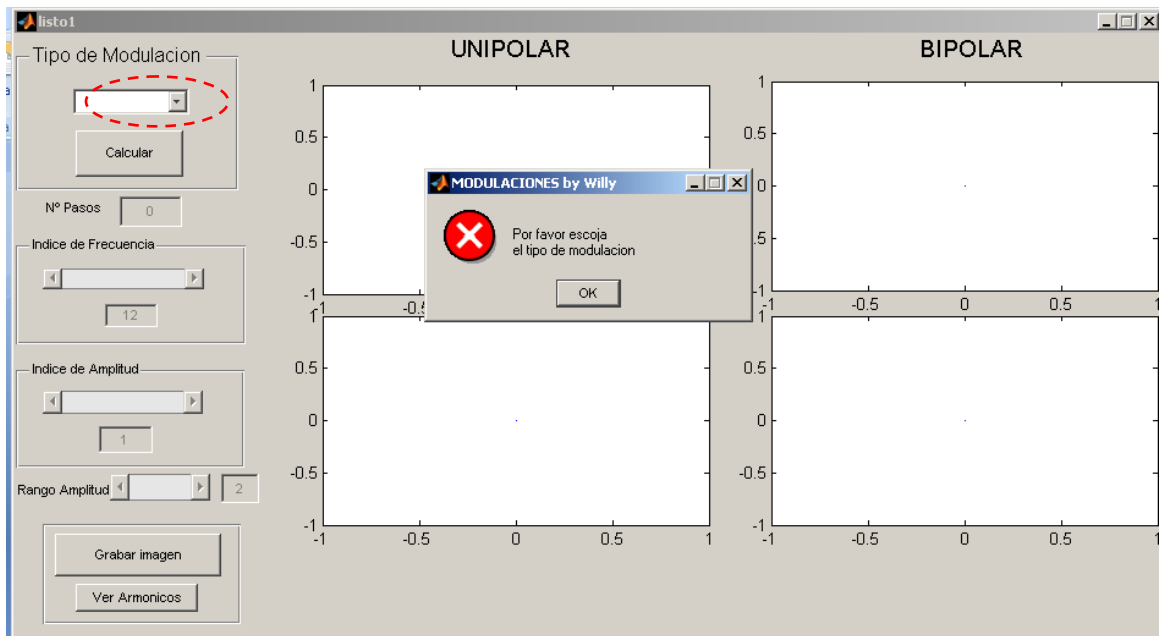


Fig. A.9. Error por no haber escogido el Tipo de modulación

- a. Otro aspecto a considerar es que si no he realizado una simulación al pulsar el botón Calcular no se habilita el Botón Armónicos.

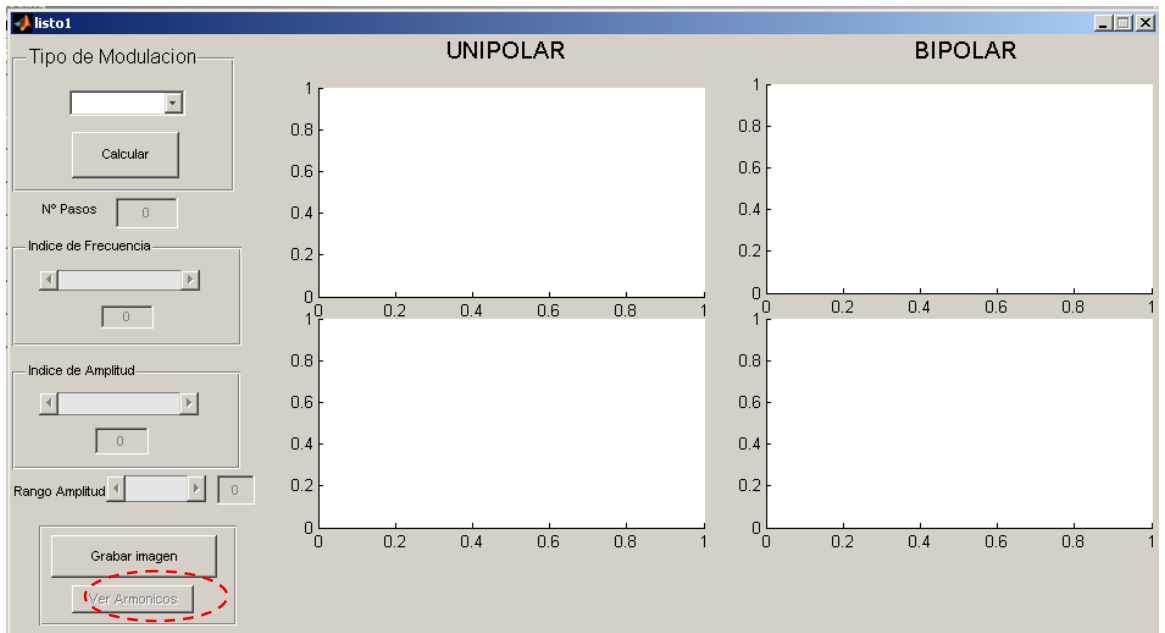


Fig. A.10. Botón Armónicos Deshabilitado

- b. Luego de realizar una simulación (pulsar botón calcular) y haber mostrado los armónicos (pulsar botón armónicos), se puede volver a ver las gráficas de voltaje presionando otra vez el botón calcular.

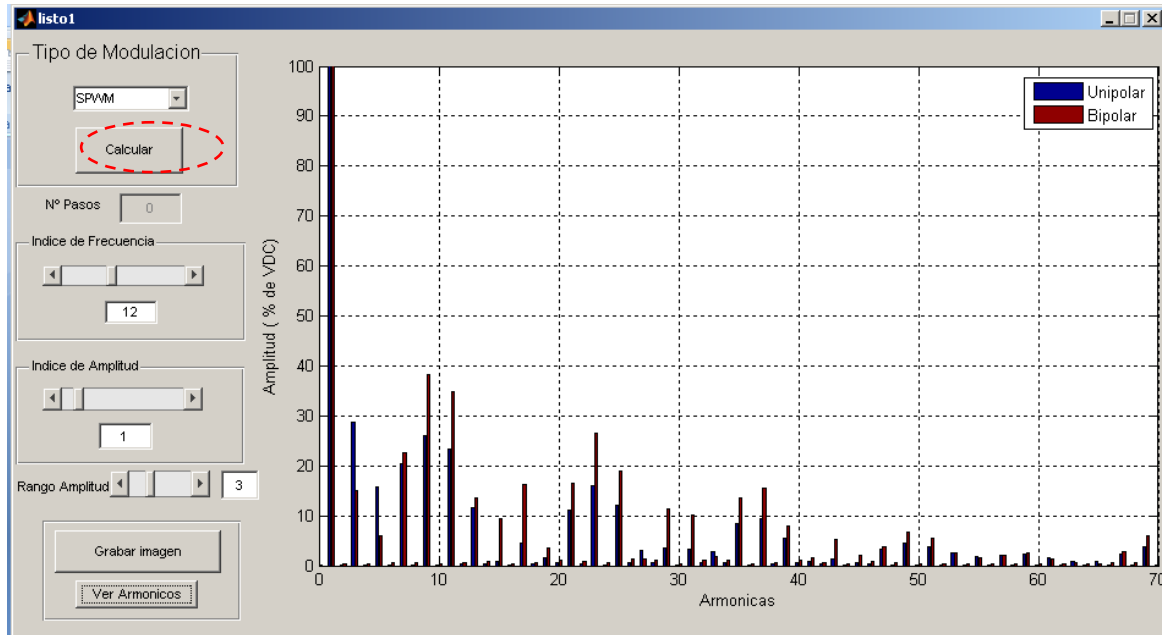


Fig. A.11. Para volver a ver gráficas iniciales

- c. El cuadro de texto *número de pasos* se habilita solamente cuando selecciono la modulación por pasos, además si es que no escribo un valor diferente de cero y presiono el botón calcular se nos muestra el mensaje de la figura A.12.

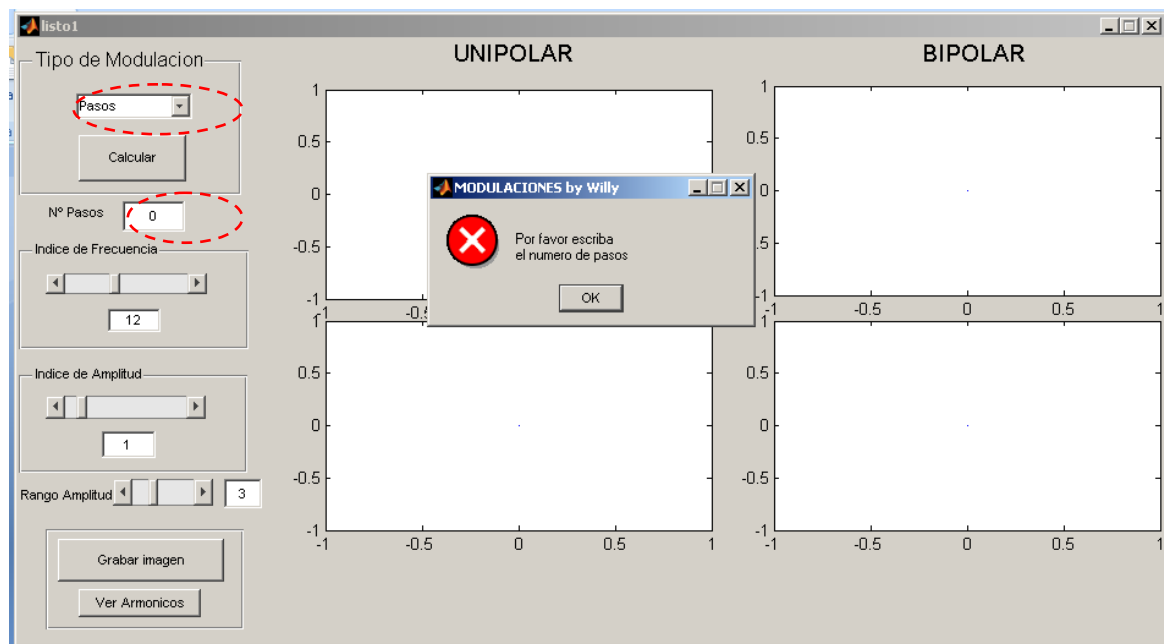


Fig. A.12. Cuadro número de pasos habilitado y error

A.1.3. ARCHIVO M-FILE DEL PROGRAMA

A continuación se presenta las líneas de programación del archivo .m de Matlab:

```
function varargout = listo1(varargin)
% LISTO1 M-file for listo1.fig
%     LISTO1, by itself, creates a new LISTO1 or raises the existing
%     singleton*.
%
%     H = LISTO1 returns the handle to a new LISTO1 or the handle to
%     the existing singleton*.
%
%     LISTO1('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in LISTO1.M with the given input arguments.
%
%     LISTO1('Property','Value',...) creates a new LISTO1 or raises the
%     existing singleton*. Starting from the left, property value pairs
are
%     applied to the GUI before listo1_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to listo1_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%     instance to run (singleton)".
%
```

```

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help listol

% Last Modified by GUIDE v2.5 05-Jul-2012 12:38:47

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @listol_OpeningFcn, ...
                  'gui_OutputFcn',  @listol_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before listol is made visible.
function listol_OpeningFcn(hObject, eventdata, handles, varargin)
    handles.i=0
    set(handles.grafic,'visible','off')
    guidata(hObject,handles)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to listol (see VARARGIN)

% Choose default command line output for listol
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes listol wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = listol_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on slider movement.
function indicefreq_Callback(hObject, eventdata, handles)
indicef=get(handles.indicefreq,'Value')
handles.indicef=indicef

guidata(hObject,handles)
set(handles.grafic,'visible','off')
set(handles.valorf,'String',handles.indicef)
% hObject      handle to indicefreq (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%          get(hObject,'Min') and get(hObject,'Max') to determine range of
slider

% --- Executes during object creation, after setting all properties.
function indicefreq_CreateFcn(hObject, eventdata, handles)

% hObject      handle to indicefreq (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function indiceampli_Callback(hObject, eventdata, handles)
indicea=get(handles.indiceampli,'Value')
handles.indicea=indicea
guidata(hObject,handles)
set(handles.valora,'String',handles.indicea)

% hObject      handle to indiceampli (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles       structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%          get(hObject,'Min') and get(hObject,'Max') to determine range of
slider

% --- Executes during object creation, after setting all properties.
```

```
function indiceampli_CreateFcn(hObject, eventdata, handles)
% hObject    handle to indiceampli (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
function valorf_Callback(hObject, eventdata, handles)
% hObject    handle to valorf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valorf as text
%        str2double(get(hObject,'String')) returns contents of valorf as a
double
```

```
% --- Executes during object creation, after setting all properties.
function valorf_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valorf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function valora_Callback(hObject, eventdata, handles)
% hObject    handle to valora (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of valora as text
%        str2double(get(hObject,'String')) returns contents of valora as a
double
```

```
% --- Executes during object creation, after setting all properties.
function valora_CreateFcn(hObject, eventdata, handles)
% hObject    handle to valora (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function rangoa_Callback(hObject, eventdata, handles)
% hObject     handle to rangoa (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rangoa as text
%         str2double(get(hObject,'String')) returns contents of rangoa as a
double

% --- Executes during object creation, after setting all properties.
function rangoa_CreateFcn(hObject, eventdata, handles)
% hObject     handle to rangoa (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in modulaciones.
function modulaciones_Callback(hObject, eventdata, handles)

valormod=get(handles.modulaciones,'Value') %Acerando graficos y mostrando
solo ventanas
axes(handles.grafic)
plot(0,0)
set(handles.grafica1,'visible','on')
set(handles.grafica2,'visible','on')
set(handles.grafica3,'visible','on')
set(handles.grafica4,'visible','on')

switch valormod
case 1
    guidata(hObject,handles)
    set(handles.grafic,'visible','off')
    set(handles.npasos,'enable','off')
    set(handles.valorf,'enable','off')
    set(handles.valora,'enable','off')
```

```
set(handles.rangoa,'enable','off')
set(handles.indicefreq,'enable','off')
set(handles.indiceampli,'enable','off')
set(handles.rangovoltaje,'enable','off')
```

case 2

```
guidata(hObject,handles)
set(handles.grafic,'visible','off')
set(handles.npasos,'enable','off')
set(handles.grafic,'visible','off')
set(handles.valorf,'enable','on')
set(handles.valora,'enable','on')
set(handles.rangoa,'enable','on')
set(handles.indicefreq,'enable','on')
set(handles.indiceampli,'enable','on')
set(handles.rangovoltaje,'enable','on')
```

case 3

```
guidata(hObject,handles)
set(handles.grafic,'visible','off')
set(handles.npasos,'enable','off')
set(handles.valorf,'enable','on')
set(handles.valora,'enable','on')
set(handles.rangoa,'enable','on')
set(handles.indicefreq,'enable','on')
set(handles.indiceampli,'enable','on')
set(handles.rangovoltaje,'enable','on')
```

case 4

```
guidata(hObject,handles)
set(handles.grafic,'visible','off')
set(handles.npasos,'enable','off')
set(handles.valorf,'enable','on')
set(handles.valora,'enable','on')
set(handles.rangoa,'enable','on')
set(handles.indicefreq,'enable','on')
set(handles.indiceampli,'enable','on')
set(handles.rangovoltaje,'enable','on')
```

case 5

```
guidata(hObject,handles)
set(handles.grafic,'visible','off')
set(handles.npasos,'enable','off')
set(handles.valorf,'enable','on')
set(handles.valora,'enable','on')
set(handles.rangoa,'enable','on')
set(handles.indicefreq,'enable','on')
set(handles.indiceampli,'enable','on')
set(handles.rangovoltaje,'enable','on')
```

case 6

```
guidata(hObject,handles)
set(handles.grafic,'visible','off')
```

```

        set(handles.npasos,'enable','off')
        set(handles.valorf,'enable','on')
        set(handles.valora,'enable','on')
        set(handles.rangoa,'enable','on')
        set(handles.indicefreq,'enable','on')
        set(handles.indiceampli,'enable','on')
        set(handles.rangovoltaje,'enable','on')
    case 7

        guidata(hObject,handles)
        set(handles.grafic,'visible','off')
        set(handles.npasos,'enable','off')
        set(handles.valorf,'enable','on')
        set(handles.valora,'enable','on')
        set(handles.rangoa,'enable','on')
        set(handles.indicefreq,'enable','on')
        set(handles.indiceampli,'enable','on')
        set(handles.rangovoltaje,'enable','on')
    case 8

        guidata(hObject,handles)
        set(handles.grafic,'visible','off')
        set(handles.npasos,'enable','off')
        set(handles.valorf,'enable','on')
        set(handles.valora,'enable','on')
        set(handles.rangoa,'enable','on')
        set(handles.indicefreq,'enable','on')
        set(handles.indiceampli,'enable','on')
        set(handles.rangovoltaje,'enable','on')

    case 9

        guidata(hObject,handles)
        set(handles.grafic,'visible','off')
        set(handles.npasos,'enable','on')
        set(handles.valorf,'enable','on')
        set(handles.valora,'enable','on')
        set(handles.rangoa,'enable','on')
        set(handles.indicefreq,'enable','on')
        set(handles.indiceampli,'enable','on')
        set(handles.rangovoltaje,'enable','on')
end
% hObject    handle to modulaciones (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns modulaciones contents as
% cell array
%           contents{get(hObject,'Value')} returns selected item from
%           modulaciones

% --- Executes during object creation, after setting all properties.
function modulaciones_CreateFcn(hObject, eventdata, handles)

```

```

% hObject      handle to modulaciones (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Calcular.
function Calcular_Callback(hObject, eventdata, handles)
% hObject      handle to Calcular (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

axes(handles.grafic)% Acerando graficas y mostrando los axis(solo ventanas
sin graficos)
plot(0,0)
set(handles.grafic,'visible','off')
set(handles.text3,'visible','on')
set(handles.text4,'visible','on')
set(handles.grafical,'visible','on')
set(handles.grafica2,'visible','on')
set(handles.grafica3,'visible','on')
set(handles.grafica4,'visible','on')

valormod=get(handles.modulaciones,'Value')
switch valormod
    case 1

        errordlg({'Por favor escoja','el tipo de
modulacion'},'MODULACIONES by Willy')
    case 2

t=0:0.001:2*pi; %Definicion del tiempo
ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
% Referencia Trapezoidal
Trap=-2.5*sawtooth(t-pi/2,.5);
for i=1:length(t);if abs(Trap(i))>1 Trap(i)=sign(Trap(i)); end; end;
ref(2,:)=Trap;
arm=1.15*sin(t)+.27*sin(3*t)-.029*sin(9*t); ref(3,:)=arm; % Referencia
Armónica
% Referencia Escalera de 2 peldaños
for i=1:length(t)
if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
else y2(i)=sign(sin(t(i))); end; end; ref(4,:)=y2;
% Escalera 3 peldaños
for i=1:length(t)
if abs(y(i))<.25; y3(i)=sign(y(i))*0.5;

```



```

else; if abs(y(i))<.5; y3(i)=sign(y(i))*0.75;
else; y3(i)=sign(y(i)); end; end; end; ref(5,:)=y3;
% Escalera 4 peldaños
for i=1:length(t)
if abs(y(i))<.25; y4(i)=sign(y(i))*0.25;
else; if abs(y(i))<.5; y4(i)=sign(y(i))*0.5;
else; if abs(y(i))<.75; y4(i)=sign(y(i))*0.75;
else; y4(i)=sign(y(i)); end; end; end; end; ref(6,:)=y4;
y5=(round(y*10))/10; ref(7,:)=y5; % Escalonada

% Modulacion PWM

mf=handles.indicef
ma=handles.indicea

rangov=handles.rangov
co=rangov

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

% Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2);end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=1
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);
axes(handles.grafical)

plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)

```

```

plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/1.5 co/1.5]); %%%Aquí
cambie
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/1.5 co/1.5]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

% figure(1); clf; n=0:72;
% bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 100]);
% xlabel('Armonicass'); ylabel('Amplitud ( % de
VDC)');legend('Unipolar','Bipolar')

case 3

mf=handles.indicef
ma=handles.indicea

rangov=handles.rangov
co=rangov

t=0:0.01:2*pi; % Definición del tiempo
ref=sin(t); % Referencia

% Modulacion SPWM
PWM1=ma*(sawtooth(mf*t,.5)+1)*.5;
ang_m=30*pi/180;
ang_mod=ang_m/(2*pi)*100;
desfa1=0;
desfa2=150*pi/180;
desfa3=180*pi/180;
desfa4=330*pi/180;
Mod2=((square(t-desfa1,ang_mod)+1)*.5+(square(t-desfa2,ang_mod)+1)*.5)
Mod2=Mod2+(square(t-desfa3,ang_mod)+1)*.5;
Mod2=(Mod2+(square(t-desfa4,ang_mod)+1)*.5).*PWM1
Mod2=Mod2+(square(t-ang_m,120/3.6)+1)*.5
Mod2=(Mod2+(square(t-ang_m-pi,120/3.6)+1)*.5).*square(t,50);
Unipolar=Mod2;Bipolar=Mod2;
for i=1:length(t)
if abs(Unipolar(i))>=abs(ref(i));
Sunipolar(i)=sign(ref(i));
else
unipolar(i)=0;
end
if (Bipolar(i))>=(ref(i));
Sbipolar(i)=1;
else
Sbipolar(i)=-1;

```

```

end
end

% Calculo Contenido Armónico
Bi=fft(Sbipolar)/(length(t)/2); Bi(1)=Bi(1)/2; Bi=abs(Bi);
Ui=fft(Sunipolar)/(length(t)/2); Ui(1)=Ui(1)/2; Ui=abs(Ui);
VrmsBin=sqrt(sum(Bi(2:72).^2))/sqrt(2);
VrmsBi=sqrt(Bi(1)^2+VrmsBin^2)
THDBi=sqrt(VrmsBi^2-Bi(2)^2/2)/(Bi(2)/sqrt(2))
VrmsBi1=Bi(2)/sqrt(2)
VrmsUin=sqrt(sum(Ui(2:72).^2))/sqrt(2);
VrmsUi=sqrt(Ui(1)^2+VrmsUin^2)
THDUi=sqrt(VrmsUi^2-Ui(2)^2/2)/(Ui(2)/sqrt(2))
VrmsUi1=Ui(2)/sqrt(2)

%Para Graficar
i=1
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafica1)
plot(t,ref,t,Unipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref,t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

figure(1); clf; n=0:72;
bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 100]);
xlabel('Armonicas'); ylabel('Amplitud ( % de VDC)');legend('Unipolar','Bipolar')

set(handles.armonicos,'enable','on')

case 4
    t=0:0.01:2*pi; % Definición del tiempo
    ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
    % Referencia Trapezoidal
    Trap=-2.5*sawtooth(t-pi/2,.5);
    for i=1:length(t);if abs(Trap(i))>1 Trap(i)=sign(Trap(i)); end; end;
    ref(2,:)=Trap;

```

```

arm=1.15*sin(t)+.27*sin(3*t)-.029*sin(9*t); ref(3,:)=arm; % Referencia
Armónica
% Referencia Escalera de 2 peldaños
for i=1:length(t)
if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
else y2(i)=sign(sin(t(i))); end; end; ref(4,:)=y2;
% Escalera 3 peldaños
for i=1:length(t)
if abs(y(i))<.25; y3(i)=sign(y(i))*.5;
else; if abs(y(i))<.5; y3(i)=sign(y(i))*.75;
else; y3(i)=sign(y(i)); end; end; end; ref(5,:)=y3;
% Escalera 4 peldaños
for i=1:length(t)
if abs(y(i))<.25; y4(i)=sign(y(i))*.25;
else; if abs(y(i))<.5; y4(i)=sign(y(i))*.5;
else; if abs(y(i))<.75; y4(i)=sign(y(i))*.75;
else; y4(i)=sign(y(i)); end; end; end; end; ref(6,:)=y4;
y5=(round(y*10))/10; ref(7,:)=y5; % Escalonada

% Modulacion PWM

mf=handles.indicef
ma=handles.indicea
rangoa=get(handles.rangoa,'String')
rangoa=str2double(rangoa)
co=rangoa

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

%Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBi1(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUi1(i)=Ui(i,2)/sqrt(2);end;
salida=[THDBi',VrmsBi',VrmsBi1',THDUi',VrmsUi',VrmsUi1']' %Salidas

%Para Graficar
i=2
handles.i=i;

```

```

handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafical)

plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');
axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

% figure(1); clf; n=0:72;
% bar(n,[Ui(i,n+1),Bi(i,n+1)]*100);grid; axis([0 70 0 120]);
% xlabel('Armonicas'); ylabel('Amplitud ( % de
VDC)');legend('Unipolar','Bipolar')

case 5
t=0:0.01:2*pi; % Definición del tiempo
ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
% Referencia Trapezoidal
Trap=-2.5*sawtooth(t-pi/2,.5);
for i=1:length(t);if abs(Trap(i))>1 Trap(i)=sign(Trap(i)); end; end;
ref(2,:)=Trap;
arm=1.15*sin(t)+0.27*sin(3*t)-0.029*sin(9*t); ref(3,:)=arm; % Referencia
Armónica
% Referencia Escalera de 2 peldaños
for i=1:length(t)
if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
else y2(i)=sign(sin(t(i))); end; end; ref(4,:)=y2;
% Escalera 3 peldaños
for i=1:length(t)
if abs(y(i))<.25; y3(i)=sign(y(i))*.5;
else; if abs(y(i))<.5; y3(i)=sign(y(i))*.75;
else; y3(i)=sign(y(i)); end; end; end; ref(5,:)=y3;
% Escalera 4 peldaños
for i=1:length(t)
if abs(y(i))<.25; y4(i)=sign(y(i))*.25;
else; if abs(y(i))<.5; y4(i)=sign(y(i))*.5;
else; if abs(y(i))<.75; y4(i)=sign(y(i))*.75;
else; y4(i)=sign(y(i)); end; end; end; end; ref(6,:)=y4;
y5=(round(y*10))/10; ref(7,:)=y5; % Escalonada

%Modulacion PWM

```

```

mf=handles.indicef
ma=handles.indicea
rangoa=get(handles.rangoa,'String')
rangoa=str2double(rangoa)

co=rangoa

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sshipolar(j,i)=-1; else Sshipolar(j,i)=1;end;
end; end;

%Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sshipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2); end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil'] %Salidas

%Para Graficar
i=3
handles.i=i;
guidata(hObject,handles);

axes(handles.grafica1)
plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sshipolar(i,:)); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

% figure(1); clf; n=0:72;

```

```

% bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 120]);
% xlabel('Armonicas'); ylabel('Amplitud ( % de
VDC) ');legend('Unipolar','Bipolar')

case 6

    t=0:0.01:2*pi; % Definición del tiempo
    ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
    % Referencia Trapezoidal
    Trap=-2.5*sawtooth(t-pi/2,.5);
    for i=1:length(t);if abs(Trap(i))>1 Trap(i)=sign(Trap(i)); end; end;
    ref(2,:)=Trap;
    arm=1.15*sin(t)+0.27*sin(3*t)-0.029*sin(9*t); ref(3,:)=arm; % Referencia
    Armónica
    % Referencia Escalera de 2 peldaños
    for i=1:length(t)
    if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
    else y2(i)=sign(sin(t(i))); end; end; ref(4,:)=y2;
    % Escalera 3 peldaños
    for i=1:length(t)
    if abs(y(i))<.25; y3(i)=sign(y(i))*0.5;
    else; if abs(y(i))<.5; y3(i)=sign(y(i))*0.75;
    else; y3(i)=sign(y(i)); end; end; end; ref(5,:)=y3;
    % Escalera 4 peldaños
    for i=1:length(t)
    if abs(y(i))<.25; y4(i)=sign(y(i))*0.25;
    else; if abs(y(i))<.5; y4(i)=sign(y(i))*0.5;
    else; if abs(y(i))<.75; y4(i)=sign(y(i))*0.75;
    else; y4(i)=sign(y(i)); end; end; end; end; ref(6,:)=y4;
    y5=(round(y*10))/10; ref(7,:)=y5; % Escalonada

    % Modulacion PWM

    mf=handles.indicef
    ma=handles.indicea
    rangoa=get(handles.rangoa,'String')
    rangoa=str2double(rangoa)
    co=rangoa

    PWM=(sawtooth(mf*t,.5)+1)*.5;
    PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
    for j=1:7; for i=1:length(t);
    if abs(Unipolar(i))>=abs(ref(j,i));
    Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
    if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
    end; end;

    % Calculo Contenido Armónico
    Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
    for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
    Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
    Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
    Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));

```

```

VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2); end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=4
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafica1)
plot(t,ref(i,:),t,Unipolar,'r');grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

% figure(1); clf; n=0:72;
% bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 120]);
% xlabel('Armonicass'); ylabel('Amplitud ( % de
VDC)');legend('Unipolar','Bipolar')

case 7
    t=0:0.01:2*pi; % Definición del tiempo
    ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
    % Referencia Trapezoidal
    Trap=-2.5*sawtooth(t-pi/2,.5);
    for i=1:length(t);if abs(Trap(i))>1 Trap(i)=sign(Trap(i)); end; end;
    ref(2,:)=Trap;
    arm=1.15*sin(t)+.27*sin(3*t)-.029*sin(9*t); ref(3,:)=arm; % Referencia
    Armónica
    % Referencia Escalera de 2 peldaños
    for i=1:length(t)
        if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
        else y2(i)=sign(sin(t(i))); end; end; ref(4,:)=y2;
    % Escalera 3 peldaños
    for i=1:length(t)
        if abs(y(i))<.25; y3(i)=sign(y(i))*0.5;

```



```

else; if abs(y(i))<.5; y3(i)=sign(y(i))*0.75;
else; y3(i)=sign(y(i)); end; end; end; ref(5,:)=y3;
% Escalera 4 peldaños
for i=1:length(t)
if abs(y(i))<.25; y4(i)=sign(y(i))*0.25;
else; if abs(y(i))<.5; y4(i)=sign(y(i))*0.5;
else; if abs(y(i))<.75; y4(i)=sign(y(i))*0.75;
else; y4(i)=sign(y(i)); end; end; end; end; ref(6,:)=y4;
y5=(round(y*10))/10; ref(7,:)=y5; % Escalonada

%Modulacion PWM

mf=handles.indicef
ma=handles.indicea
rangoa=get(handles.rangoa,'String')
rangoa=str2double(rangoa)
co=rangoa

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

%Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2);end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=5
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafical)
plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/2 co/2]);

```

```

ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

% figure(1); clf; n=0:72;
% bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 120]);
% xlabel('Armonicas'); ylabel('Amplitud ( % de
VDC)');legend('Unipolar','Bipolar')

    case 8
        t=0:0.01:2*pi; % Definición del tiempo
        ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
        % Referencia Trapezoidal
        Trap=-2.5*sawtooth(t-pi/2,.5);
        for i=1:length(t);if abs(Trap(i))>1 Trap(i)=sign(Trap(i)); end; end;
        ref(2,:)=Trap;
        arm=1.15*sin(t)+0.27*sin(3*t)-0.029*sin(9*t); ref(3,:)=arm; % Referencia
        Armónica
        % Referencia Escalera de 2 peldaños
        for i=1:length(t)
            if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
            else y2(i)=sign(sin(t(i))); end; end; ref(4,:)=y2;
        % Escalera 3 peldaños
        for i=1:length(t)
            if abs(y(i))<.25; y3(i)=sign(y(i))*0.5;
            else; if abs(y(i))<.5; y3(i)=sign(y(i))*0.75;
            else; y3(i)=sign(y(i)); end; end; end; ref(5,:)=y3;
        % Escalera 4 peldaños
        for i=1:length(t)
            if abs(y(i))<.25; y4(i)=sign(y(i))*0.25;
            else; if abs(y(i))<.5; y4(i)=sign(y(i))*0.5;
            else; if abs(y(i))<.75; y4(i)=sign(y(i))*0.75;
            else; y4(i)=sign(y(i)); end; end; end; end; ref(6,:)=y4;
        y5=(round(y*10))/10; ref(7,:)=y5; % Escalonada

    %Modulacion PWM

mf=handles.indicef
ma=handles.indicea
rangoa=get(handles.rangoa,'String')
rangoa=str2double(rangoa)
% handles.rangoa=rangoa
% co=rangoa;
co=rangoa

PWM=(sawtooth(mf*t,.5)+1)*.5;
PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);

```

```

for j=1:7; for i=1:length(t);
if abs(Unipolar(i))>=abs(ref(j,i));
Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
end; end;

%Calculo Contenido Armónico
Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);
Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2); end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=6
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafica1)
plot(t,ref(i,:),t,Unipolar,'r');grid ; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid ; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

% figure(1); clf; n=0:72;
% bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 120]);
% xlabel('Armonicas'); ylabel('Amplitud ( % de
VDC)');legend('Unipolar','Bipolar')

case 9

numpasos=get(handles.npasos,'String')
numpasos=str2double(numpasos)

```

```

        if numpasos==0 %%%%%%%%%%el end de este if esta al ultimo de case8
            errordlg({'Por favor escriba','el numero de pasos'},'MODULACIONES
by Willy')
        else
            numpasos=get(handles.npasos,'String')
            numpasos=str2double(numpasos)

            t=0:0.01:2*pi; % Definición del tiempo
            ref(1,:)=sin(t); y=ref(1,:); % Referencia sinusoidal
            % Referencia Trapezoidal
            Trap=-2.5*sawtooth(t-pi/2,.5);
            for i=1:length(t);if abs(Trap(i))>1 Trap(i)=sign(Trap(i)); end; end;
            ref(2,:)=Trap;
            arm=1.15*sin(t)+0.27*sin(3*t)-0.029*sin(9*t); ref(3,:)=arm; % Referencia
            Armónica
            % Referencia Escalera de 2 peldaños
            for i=1:length(t)
                if abs(sin(t(i)))<0.5; y2(i)=sign(sin(t(i)))*.5;
                else y2(i)=sign(sin(t(i))); end; end; ref(4,:)=y2;
            % Escalera 3 peldaños
            for i=1:length(t)
                if abs(y(i))<.25; y3(i)=sign(y(i))*.5;
                else; if abs(y(i))<.5; y3(i)=sign(y(i))*.75;
                else; y3(i)=sign(y(i)); end; end; end; ref(5,:)=y3;
            % Escalera 4 peldaños
            for i=1:length(t)
                if abs(y(i))<.25; y4(i)=sign(y(i))*.25;
                else; if abs(y(i))<.5; y4(i)=sign(y(i))*.5;
                else; if abs(y(i))<.75; y4(i)=sign(y(i))*.75;
                else; y4(i)=sign(y(i)); end; end; end; end; ref(6,:)=y4;
            y5=(round(y*numpasos))/numpasos; ref(7,:)=y5; % Escalonada

            %Modulacion PWM

            mf=handles.indicef
            ma=handles.indicea
            rangoa=get(handles.rangoa,'String')
            rangoa=str2double(rangoa)
            co=rangoa

            PWM=(sawtooth(mf*t,.5)+1)*.5;
            PWM1=square(t,50); Unipolar=ma*PWM1.*PWM; Bipolar=ma*sawtooth(mf*t,.5);
            for j=1:7; for i=1:length(t);
                if abs(Unipolar(i))>=abs(ref(j,i));
                Sunipolar(j,i)=0; else; Sunipolar(j,i)=sign(ref(j,i)); end;
                if (Bipolar(i))>=(ref(j,i)); Sbipolar(j,i)=-1; else Sbipolar(j,i)=1;end;
            end; end;

            %Calculo Contenido Armónico
            Bi=zeros(7,length(t)); Ui=zeros(7,length(t));
            for i=1:7 Bi(i,:)=fft(Sbipolar(i,:))/(length(t)/2);
            Bi(i,1)=Bi(i,1)/2; Bi(i,:)=abs(Bi(i,:));
            Ui(i,:)=fft(Sunipolar(i,:))/(length(t)/2);

```

```

Ui(i,1)=Ui(i,1)/2; Ui(i,:)=abs(Ui(i,:));
VrmsBin(i)=sqrt(sum(Bi(i,2:72).^2))/sqrt(2);
VrmsBi(i)=sqrt(Bi(i,1)^2+VrmsBin(i)^2);
THDBi(i)=sqrt(VrmsBi(i)^2-Bi(i,2)^2/2)/(Bi(i,2)/sqrt(2));
VrmsBil(i)=Bi(i,2)/sqrt(2); VrmsUin(i)=sqrt(sum(Ui(i,2:72).^2))/sqrt(2);
VrmsUi(i)=sqrt(Ui(i,1)^2+VrmsUin(i)^2);
THDUi(i)=sqrt(VrmsUi(i)^2-Ui(i,2)^2/2)/(Ui(i,2)/sqrt(2));
VrmsUil(i)=Ui(i,2)/sqrt(2); end;
salida=[THDBi',VrmsBi',VrmsBil',THDUi',VrmsUi',VrmsUil']' %Salidas

%Para Graficar
i=7
handles.i=i;
handles.Ui=Ui
handles.Bi=Bi
guidata(hObject,handles);

axes(handles.grafical)
plot(t,ref(i,:),t,Unipolar,'r');grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencia','Portadora');

axes(handles.grafica2)
plot(t,Sunipolar(i,:));grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida');

axes(handles.grafica3)
plot(t,ref(i,:),t,Bipolar,'r'); grid; axis([0 2*pi -co co]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Referencias','Portadora');

axes(handles.grafica4)
plot(t,Sbipolar(i,:)); grid; axis([0 2*pi -co/2 co/2]);
ylabel('Magnitud'); xlabel('Tiempo'); legend('Salida')
set(handles.armonicos,'enable','on')

% figure(1); clf; n=0:72;
% bar(n,[Ui(i,n+1)',Bi(i,n+1)']*100);grid; axis([0 70 0 120]);
% xlabel('Armonicass'); ylabel('Amplitud ( % de
VDC)');legend('Unipolar','Bipolar')
    end %este es el end de el if del inicio de case8
end

% --- Executes during object creation, after setting all properties.
function grafical_CreateFcn(hObject, eventdata, handles)
% hObject    handle to grafical (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate grafical

% --- Executes during object deletion, before destroying properties.
function grafical_DeleteFcn(hObject, eventdata, handles)

```

```
% hObject      handle to grafical (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on slider movement.
function rangovoltaje_Callback(hObject, eventdata, handles)
rangov=get(handles.rangovoltaje,'Value')
handles.rangov=rangov
guidata(hObject,handles)%%%%%%%%%%%%%%-----OJO ES PARA
GRABAR
set(handles.rangoa,'String',handles.rangov)

% hObject      handle to rangovoltaje (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of
slider

% --- Executes during object creation, after setting all properties.
function rangovoltaje_CreateFcn(hObject, eventdata, handles)
% hObject      handle to rangovoltaje (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function npasos_Callback(hObject, eventdata, handles)
% hObject      handle to npasos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of npasos as text
%         str2double(get(hObject,'String')) returns contents of npasos as a
double

% --- Executes during object creation, after setting all properties.
function npasos_CreateFcn(hObject, eventdata, handles)
% hObject      handle to npasos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called
```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in grabar.
function grabar_Callback(hObject, eventdata, handles)
% h=figure('Position',[800 800 800 800]);
if handles.i==0
errordlg({'Por favor escoja','el tipo de modulacion'},'MODULACIONES by
Willy');return,end;

%uiwait(msgbox('Coloque el nombre del archivo y la extension
.bmp','Modulaciones by Willy','modal'));
set(handles.uipanel3,'Visible','off')
set(handles.uipanel3,'Title',' ')
set(handles.Calcular,'Visible','off')
set(handles.modulaciones,'Visible','off')
set(handles.npasos,'Visible','off')
set(handles.text5,'Visible','off')
set(handles.uipanel1,'Visible','off')
set(handles.uipanel1,'Title',' ')
set(handles.indicefreq,'Visible','off')
set(handles.valorf,'Visible','off')
set(handles.uipanel2,'Visible','off')
set(handles.uipanel2,'Title',' ')
set(handles.indiceampli,'Visible','off')
set(handles.valora,'Visible','off')
set(handles.rangovoltaje,'Visible','off')
set(handles.rangoa,'Visible','off')
set(handles.grabar,'Visible','off')
set(handles.text2,'Visible','off')
set(handles.uipanel6,'visible','off')
set(handles.armonicos,'visible','off')
set(handles.nombremodu,'Visible','on')
set(handles.text4,'BackgroundColor','w')%%%hacer color blanco background
texto de unipolar
set(handles.text3,'BackgroundColor','w')%%%hacer color blanco background
texto de bipolar
if handles.i==1 set(handles.nombremodu,'String','Modulacion PWM');
elseif handles.i==2 set(handles.nombremodu,'String','Modulacion SPWM');
    elseif handles.i==3 set(handles.nombremodu,'String','Modulacion
Trapezoidal');
        elseif handles.i==4 set(handles.nombremodu,'String','Modulacion
Armonicas');
            elseif handles.i==5
set(handles.nombremodu,'String','Modulacion Escalera 2 niveles');[.9 .9
.9]
                elseif handles.i==6
set(handles.nombremodu,'String','Modulacion Escalera 3 niveles');
                    elseif handles.i==7
set(handles.nombremodu,'String','Modulacion Escalera 4 niveles');

```

```

                                elseif handles.i==8
set(handles.nombremodu,'String','Modulacion por Pasos');
end
axes(handles.grafical);
% plot(x,tan(x))
%handles.axes1

[filename,pathname]=uiputfile({'*.bmp;*.fig'},'Guardar nombre de
archivo.bmp')

while filename==0 uiwait(errordlg({'ERROR, Escriba el nombre del archivo y
la extension .bmp'},'Modulaciones by Willy','modal'));
[filename,pathname]=uiputfile({'*.bmp;*.fig'},'Guardar nombre de
archivo.bmp');

end
[a,b,c]=fileparts(filename)%%%en c se guarda la extension del archivo y b
es el nombre
compext=strmatch('.bmp',c,'exact')%%%compara extension grabada con .bmp
compext=isempty(compext)

while compext==1
uiwait(errordlg({'ERROR de extension de la imagen, coloque .bmp como
extension'},'Modulaciones by Willy','modal'));
[filename,pathname]=uiputfile({'*.bmp;*.fig'},'Guardar nombre de
archivo.bmp')
[a,b,c]=fileparts(filename)
compext=strmatch('.bmp',c,'exact')%%%compara extension grabada con .bmp
compext=isempty(compext)
end
saveas(gcf,filename)
set(handles.uipanel3,'Visible','on')
set(handles.uipanel3,'Title','Tipo de Modulacion ')
set(handles.Calcular,'Visible','on')
set(handles.modulaciones,'Visible','on')
set(handles.npasos,'Visible','on')
set(handles.text5,'Visible','on')
set(handles.uipanel1,'Visible','on')
set(handles.uipanel1,'Title','Indice de Frecuencia')
set(handles.indicefreq,'Visible','on')
set(handles.valorf,'Visible','on')
set(handles.uipanel2,'Visible','on')
set(handles.uipanel2,'Title','Indice de Amplitud')
set(handles.indiceampli,'Visible','on')
set(handles.valora,'Visible','on')
set(handles.rangovoltaje,'Visible','on')
set(handles.rangoa,'Visible','on')
set(handles.grabar,'Visible','on')
set(handles.uipanel6,'visible','on')
set(handles.armonicos,'visible','on')
set(handles.text2,'Visible','on')
set(handles.text4,'BackgroundColor',[.831 .816 .784])%%%%%%hacer color
blanco background texto de unipolar
set(handles.text3,'BackgroundColor',[.831 .816 .784])%%%%%%hacer color
blanco background texto de unipolar

```



```
set(handles.nombremodu, 'Visible', 'off')

% close(h)
% hObject    handle to grabar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in armonicos.
function armonicos_Callback(hObject, eventdata, handles)
% hObject    handle to armonicos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

axes(handles.grafica1)
plot(0,0)
axes(handles.grafica2)
plot(0,0)
axes(handles.grafica3)
plot(0,0)
axes(handles.grafica4)
plot(0,0)

guidata(hObject,handles)
set(handles.text3, 'visible', 'off')
set(handles.text4, 'visible', 'off')
set(handles.grafica1, 'visible', 'off')
set(handles.grafica2, 'visible', 'off')
set(handles.grafica3, 'visible', 'off')
set(handles.grafica4, 'visible', 'off')
Ui=handles.Ui
Bi=handles.Bi
i=handles.i
axes(handles.grafic)

%figure(1); clf; n=0:72;
n=0:72;
bar(n, [Ui(i,n+1)', Bi(i,n+1)'] * 100); grid; axis([0 70 0 100]);
xlabel('Armonicas'); ylabel('Amplitud ( % de
VDC)'); legend('Unipolar', 'Bipolar')
```

CONCLUSIONES Y RECOMENDACIONES:

Como conclusiones se puede determinar que el uso de las técnicas modernas de modulación en los sistemas CD/CA tiene algunas ventajas como la reducción de los requerimientos de filtrado y el control de la amplitud de la salida. Dependiendo de la referencia que usemos obtenemos otra técnica de modulación que puede ser PWM, SPWM, Trapezoidal, Por inyección de armónicas, Por pasos, escalera según el índice de frecuencia y amplitud podemos alterar las características presentadas en la salida.

Esta tesis servirá como material de referencia cuando se trabaje en el tema, pues además de sólo obtener información teórica acerca de este, también podremos realizar las simulaciones de cada Modulación para observar su comportamiento en la carga. Además podremos guardar las gráficas presentadas para futuras referencias.

BIBLIOGRAFÍA:

- [1] BUENO Alexander. ELECTRÓNICA DE POTENCIA. Aspectos Generales y Convertidores
Electrónicos, Venezuela: Universidad Simón Bolívar, 2007
- [2] RASHID Muhammad. ELECTRÓNICA DE POTENCIA. Circuitos, dispositivos y aplicaciones,
Tercera Edición, México: Prentice Hall, 2004
- [3] MathWorks Inc. Matlab Overview[en línea]. MathWorks, 1994-2011, [ref. de 15-06-2011]. Disponible
en Web: <http://www.mathworks.com/products/matlab/>
- [4] Norma IEEE 519-1992, Recomendaciones prácticas y requerimientos de la IEEE para el Control de Armónicos en Sistemas Eléctricos de Potencia.
- [4] Regulación CONELEC No. 004-01, Regulación para la Calidad del Servicio Eléctrico de Distribución en el Ecuador.