

**UNIVERSIDAD DE ALCALÁ**

**Escuela Politécnica Superior**

**GRADO EN INGENIERÍA EN ELECTRÓNICA  
Y AUTOMÁTICA INDUSTRIAL**

**Trabajo Fin de Grado**

**“Interfaz gráfica para simulación de  
convertidores DC/AC para entorno  
docente”**

**Josua Cabeza Picazo  
2015**



**UNIVERSIDAD DE ALCALÁ**

**Escuela Politécnica Superior**

**GRADO EN INGENIERÍA EN ELECTRÓNICA  
Y AUTOMÁTICA INDUSTRIAL**

**Trabajo Fin de Grado**

**“Interfaz gráfica para simulación de  
convertidores DC/AC para entorno  
docente”**

**Autor:** Josua Cabeza Picazo  
**Directores:** M<sup>a</sup> Carmen Pérez Rubio  
Jesús Ureña Ureña

**Tribunal**

**Presidente:** Emilio José Bueno Peña

**Vocal 1º:** Manuel Guerra Martínez

**Vocal 2º:** M<sup>a</sup> Carmen Pérez Rubio

**Calificación:**.....

**Fecha:**.....



# Agradecimientos

Este proyecto se lo quiero dedicar a las cuatro personas más importantes de mi vida y que más han sufrido estos años de carrera.

En primer lugar a ti Joanna, mi hermana, con este proyecto quiero enseñarte que con esfuerzo todo se consigue, que eres una luchadora y aunque te parezca lejano, se que algún día seré yo quien te acompañe en un momento como éste.

A ti mamá, por estar siempre preocupada por mi aunque no siempre me diera cuenta, porque gracias a ti soy lo que soy, y has conseguido que lo que eran piedras en el camino me hicieran más fuerte en vez de hacerme tirar la toalla, porque gracias a tu apoyo aquellas etapas de mi vida en las que era diferente al resto ahora son un recuerdo del que me río.

Papa, gracias a ti soy lo que soy, porque como padre has sido una inspiración para dar los pasos que he ido dando a lo largo de la vida, has hecho que el esfuerzo y el trabajo bien hecho sea por sí mismo una recompensa y ahora que acabo, de alguna manera, siento que alcanzo una meta junto a ti.

Por último gracias Cris, porque juntos hemos superado lo más difícil de esta etapa, hemos conseguido acabar lo que hace tiempo parecía imposible y juntos hemos alcanzado la primera de muchas metas que están por llegarnos, porque éste es sólo un pequeño paso de todos los que nos quedan por dar.



# Índice General

<b>Resumen.....</b>	<b>1</b>
Resumen .....	3
Abstract .....	5
Resumen Extendido .....	7
<b>Memoria.....</b>	<b>9</b>
Introducción.....	11
1.1 Estructura del documento .....	12
Revisión de convertidores DC/AC.....	13
2.1 Convertidor monofásico en puente completo .....	13
2.1.1 Introducción .....	13
2.1.2 Modulación PWM bipolar .....	17
2.1.3 Modulación PWM unipolar.....	20
2.1.4 Modulación mediante cancelación de tensión.....	21
2.2 Convertidor monofásico Push Pull.....	22
2.3 Convertidor trifásico en puente completo.....	24
2.3.1 Introducción .....	24
2.3.2 Modulación PWM clásica .....	24
2.3.3 Modulación PWM con inyección de secuencia cero: método inyección tercer armónico (THIPWM).....	27
2.3.4 Modulación PWM con inyección de secuencia cero: método PWM con vectores espaciales basada en portadora (CB-SVPWM).....	29
2.3.5 Modulación PWM con inyección de secuencia cero: método discontinuo (DPWM).....	29
2.3.6 Modulación vectorial .....	30
Herramienta docente para la enseñanza de convertidores DC/AC.....	35
3.1 Introducción.....	35
3.2 Interfaz gráfica de convertidores monofásicos .....	35

3.2.1	Modulación PWM Bipolar .....	38
3.2.2	Modulación PWM Unipolar .....	52
3.2.3	Modulación mediante cancelación de tensión .....	60
3.2.4	Inversor Push Pull.....	68
3.3	Interfaz gráfica de convertidores trifásicos .....	77
3.3.1	Modulación PWM clásica .....	78
3.3.2	Modulación PWM con inyección de secuencia cero: método inyección tercer armónico (THIPWM).....	92
3.3.3	Modulación PWM con inyección de secuencia cero: método PWM con vectores espaciales basada en portadora (CB-SVPWM).....	103
3.3.4	Modulación PWM con inyección de secuencia cero: método discontinuo (DPWM).....	114
3.3.5	Modulación vectorial .....	125
	Conclusiones y Líneas Futuras .....	141
4.1	Conclusiones .....	141
4.2	Líneas Futuras.....	141
	<b>Pliego de condiciones.....</b>	<b>143</b>
	<b>Presupuesto.....</b>	<b>147</b>
	<b>Diagramas.....</b>	<b>151</b>
	<b>Manual de usuario.....</b>	<b>179</b>
	<b>Bibliografía .....</b>	<b>183</b>



# Índice de Figuras

Figura 2.1: Circuito inversor en Puente completo [6].	13
Figura 2.2: Señales portadora y moduladora [6].	14
Figura 2.3: Zonas de funcionamiento en función de $ma$ [6].	16
Figura 2.4: Efecto de tiempos muertos en la tensión de salida [6].	17
Figura 2.5: Tensión de salida y señales de control en modulación PWM bipolar [6].	17
Figura 2.6: Armónicos en zona de sobremodulación [6].	18
Figura 2.7: Tensión y corriente de salida en modulación bipolar [6].	19
Figura 2.8: Tensión de salida en modulación PWM unipolar [6].	20
Figura 2.9: Corriente de salida modulación unipolar para el caso lineal [6].	21
Figura 2.10: Tensión de salida en modulación mediante cancelación de tensión [6].	21
Figura 2.11: Armónicos y THD de la tensión de salida en modulación mediante cancelación de tensión [6].	22
Figura 2.12: Circuito inversor Push Pull [6].	23
Figura 2.13: Circuito inversor trifásico en tres ramas.	24
Figura 2.14: Señales de control en modulación PWM y tensión de salida [6].	25
Figura 2.15: Zonas de funcionamiento en función de $ma$ [6].	27
Figura 2.16: Corriente de entrada de un inversor trifásico [6].	27
Figura 2.17: Esquema de funcionamiento de modulación PWM con inyección de secuencia cero.	28
Figura 2.18: Señales de control en modulación THIPWM [7].	28
Figura 2.19: Señales de control con modulación CB-SVPWM [7].	29
Figura 2.20: Señales de control en modulación DPWM [7].	30
Figura 2.21: Posibles estados de los transistores del puente en modulación vectorial.	31
Figura 2.22: a) Hexágono de vectores estacionarios, b) Primer semisextante en detalle.	31
Figura 2.23: Secuencia de conmutación del primer semisextante [8].	32
Figura 2.24: a) Tensión de salida en modulación SVM, b) Armónicos de la tensión de salida [8].	33
Figura 2.25: Secuencias de conmutación tipo A y tipo B en modulación vectorial [8].	33
Figura 2.26: Hexágono de vectores estacionarios en función de secuencia tipo A y tipo B.	34
Figura 2.27: a) Tensión de salida en modulación SVM con eliminación de armónicos pares, b) Armónicos de la tensión de salida [8].	34
Figura 3.1: Pantalla de inicio.	36
Figura 3.2: Pantalla principal de inversores monofásicos.	37

Figura 3.3: Esquema del puente completo.....	38
Figura 3.4: Modelo Simulink en puente completo monofásico en modulación bipolar.....	38
Figura 3.5: Circuito Simulink de modulación PWM bipolar monofásico.....	39
Figura 3.6: Cálculos con modulación PWM bipolar y datos por defecto.....	41
Figura 3.7: Mensaje de advertencia. ....	42
Figura 3.8: Transparencias sobre inversores monofásicos de la asignatura. ....	42
Figura 3.9: Pantalla de representación de formas de onda con PWM.....	43
Figura 3.10: Señales PWM, TA+ y TB+ con modulación PWM bipolar. ....	44
Figura 3.11: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM bipolar.....	45
Figura 3.12: Corriente de salida ( $i_o$ ) en modulación PWM bipolar.....	46
Figura 3.13: Corrientes del circuito en modulación PWM bipolar. ....	47
Figura 3.14: Armónicos de tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM bipolar.....	48
Figura 3.15: Ciclos de los transistores en modulación PWM bipolar. ....	49
Figura 3.16: Mensaje de advertencia sobre la necesidad de cálculo. ....	50
Figura 3.17: Pantalla de barrido.....	50
Figura 3.18: Imagen precalculada del barrido en modulación bipolar. ....	51
Figura 3.19: Circuito Simulink con modulación PWM unipolar.....	52
Figura 3.20: Cálculos con modulación PWM unipolar y datos por defecto.....	53
Figura 3.21: Señales PWM, TA+ y TB+ en modulación PWM unipolar. ....	54
Figura 3.22: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM unipolar.....	55
Figura 3.23: Corriente de salida ( $i_o$ ) en modulación PWM unipolar.....	56
Figura 3.24: Corrientes del circuito en modulación PWM unipolar. ....	57
Figura 3.25: Armónicos de tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM unipolar.....	58
Figura 3.26: Ciclos de los transistores en modulación PWM unipolar. ....	59
Figura 3.27: Imagen precalculada del barrido en modulación PWM unipolar. ....	60
Figura 3.28: Circuito Simulink de modulación mediante cancelación de tensión. ....	60
Figura 3.29: Cálculos de modulación mediante cancelación de tensión con datos por defecto. ....	61
Figura 3.30: Pantalla de representación formas de onda sin PWM. ....	62
Figura 3.31: Ciclos de los transistores en modulación mediante cancelación de tensión para $\alpha=45^\circ$ . ....	63
Figura 3.32: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación mediante cancelación de tensión. ....	64
Figura 3.33: Corriente de salida ( $i_o$ ) en modulación mediante cancelación de tensión. ....	65
Figura 3.34: Corrientes del circuito en modulación mediante cancelación de tensión. ....	66
Figura 3.35: Armónicos de tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación mediante cancelación de tensión. ....	67

Figura 3.36: Imagen precalculada de barrido en modulación mediante cancelación de tensión. ....	68
Figura 3.37: Esquema del inversor Push Pull. ....	68
Figura 3.38: Cálculos del inversor Push Pull con datos por defecto. ....	69
Figura 3.39: Circuito Simulink del inversor Push Pull. ....	70
Figura 3.40: Circuito Simulink del Generador PWM en inversor Push Pull. ....	70
Figura 3.41: Señal PWM, TA+ y TB+ en inversor Push Pull. ....	71
Figura 3.42: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en inversor Push Pull. ....	72
Figura 3.43: Corriente de salida ( $i_o$ ) en inversor Push Pull. ....	73
Figura 3.44: Corrientes del circuito en inversor Push Pull. ....	74
Figura 3.45: Armónicos tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en inversor Push Pull. ....	75
Figura 3.46: Ciclos de los transistores en inversor Push Pull. ....	76
Figura 3.47: Imagen precalculada del barrido en inversor Push Pull. ....	76
Figura 3.48: Pantalla principal de inversores trifásicos. ....	77
Figura 3.49: Esquema del inversor en tres ramas. ....	78
Figura 3.50: Modelo Simulink en puente completo. ....	78
Figura 3.51: Circuito Simulink de modulación PWM. ....	79
Figura 3.52: Transparencias sobre inversores trifásicos de la asignatura. ....	80
Figura 3.53: Cálculos con modulación PWM y datos por defecto. ....	81
Figura 3.54: Pantalla de representación formas de onda trifásica con PWM. ....	82
Figura 3.55: Señales PWM, TA+, TB+ y TC+ con modulación PWM. ....	83
Figura 3.56: Tensiones $v_{AB}$ , $v_{BC}$ y $v_{CA}$ en modulación PWM. ....	84
Figura 3.57: Tensiones $v_{AN}$ , $v_{BN}$ y $v_{CN}$ en modulación PWM. ....	85
Figura 3.58: Tensiones $v_{An}$ , $v_{Bn}$ y $v_{Cn}$ en modulación PWM. ....	86
Figura 3.59: Tensiones $v_{AN}$ , $v_{An}$ y $v_{Nn}$ en modulación PWM. ....	87
Figura 3.60: Corrientes de fase $i_A$ , $i_B$ y $i_C$ en modulación PWM. ....	88
Figura 3.61: Corrientes de fase $i_A$ , $i_{A1}$ y $i_{Aripple}$ en modulación PWM. ....	89
Figura 3.62: Corrientes $i_L$ , $i_{CI}$ y $i_{RI}$ e $i_A$ en modulación PWM. ....	90
Figura 3.63: Armónicos de tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación PWM. ....	91
Figura 3.64: Imagen precalculada del barrido en modulación PWM. ....	92
Figura 3.65: Circuito Simulink de modulación THIPWM. ....	92
Figura 3.66: Cálculos con modulación THIPWM y datos por defecto. ....	93
Figura 3.67: Señales PWM, TA+, TB+ y TC+ con modulación THIPWM. ....	94
Figura 3.68: Tensiones $v_{AB}$ , $v_{BC}$ y $v_{CA}$ en modulación THIPWM. ....	95
Figura 3.69: Tensiones $v_{AN}$ , $v_{BN}$ y $v_{CN}$ en modulación THIPWM. ....	96
Figura 3.70: Tensiones $v_{An}$ , $v_{Bn}$ y $v_{Cn}$ en modulación THIPWM. ....	97
Figura 3.71: Tensiones $v_{AN}$ , $v_{An}$ y $v_{Nn}$ en modulación THIPWM. ....	98

Figura 3.72: Corrientes de fase $i_A$ , $i_B$ y $i_C$ en modulación THIPWM. ....	99
Figura 3.73: Corriente de fase $i_A$ , $i_{A1}$ y $i_{Aripple}$ en modulación THIPWM. ....	100
Figura 3.74: Corrientes $i_L$ , $i_{CI}$ y $i_{RI}$ e $i_A$ en modulación THIPWM.....	101
Figura 3.75: Armónicos de tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación THIPWM. ....	102
Figura 3.76: Imagen precalculada del barrido en modulación THIPWM. ....	103
Figura 3.77: Circuito Simulink de modulación CB-SVPWM. ....	103
Figura 3.78: Cálculos con modulación CB-SVPWM y datos por defecto. ....	104
Figura 3.79: Señales PWM, $TA+$ , $TB+$ y $TC+$ con modulación CB-SVPWM. ....	105
Figura 3.80: Tensiones $v_{AB}$ , $v_{BC}$ y $v_{CA}$ en modulación CB-SVPWM.....	106
Figura 3.81: Tensiones $v_{AN}$ , $v_{BN}$ y $v_{CN}$ en modulación CB-SVPWM. ....	107
Figura 3.82: Tensiones $v_{An}$ , $v_{Bn}$ y $v_{Cn}$ en modulación CB-SVPWM. ....	108
Figura 3.83: Tensiones $v_{AN}$ , $v_{An}$ y $v_{Nn}$ en modulación CB-SVPWM. ....	109
Figura 3.84: Corrientes de fase $i_A$ , $i_B$ y $i_C$ en modulación CB-SVPWM. ....	110
Figura 3.85: Corriente de fase $i_A$ , $i_{A1}$ y $i_{Aripple}$ en modulación CB-SVPWM. ....	111
Figura 3.86: Corrientes $i_L$ , $i_{CI}$ y $i_{RI}$ e $i_A$ en modulación CB-SVPWM.....	112
Figura 3.87: Armónicos tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación CB-SVPWM. ....	113
Figura 3.88: Imagen precalculada del barrido en modulación CB-SVPWM. ....	114
Figura 3.89: Circuito Simulink de modulación DPWM.....	114
Figura 3.90: Cálculos con modulación DPWM y datos por defecto. ....	115
Figura 3.91: Señales PWM, $TA+$ , $TB+$ y $TC+$ con modulación DPWM.....	116
Figura 3.92: Tensiones $v_{AB}$ , $v_{BC}$ y $v_{CA}$ en modulación DPWM. ....	117
Figura 3.93: Tensiones $v_{AN}$ , $v_{BN}$ y $v_{CN}$ en modulación DPWM.....	118
Figura 3.94: Tensiones $v_{An}$ , $v_{Bn}$ y $v_{Cn}$ en modulación DPWM.....	119
Figura 3.95: Tensiones $v_{AN}$ , $v_{An}$ y $v_{Nn}$ en modulación DPWM.....	120
Figura 3.96: Corrientes de fase $i_A$ , $i_B$ y $i_C$ en modulación DPWM.....	121
Figura 3.97: Corrientes de fase $i_A$ , $i_{A1}$ y $i_{Aripple}$ en modulación DPWM. ....	122
Figura 3.98: Corrientes $i_L$ , $i_{CI}$ y $i_{RI}$ e $i_A$ en modulación DPWM. ....	123
Figura 3.99: Armónicos tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación DPWM.....	124
Figura 3.100: Imagen precalculada barrido en modulación DPWM.....	124
Figura 3.101: Circuito Simulink de modulación vectorial. ....	125
Figura 3.102: Cálculos con modulación vectorial normal y valores por defecto.....	127
Figura 3.103: Pantalla representación formas de onda trifásica con modulación vectorial. ....	128
Figura 3.104: Hexágono de estados y señales $TA+$ , $TB+$ y $TC+$ con modulación vectorial. ....	129
Figura 3.105: Tensiones $v_{AB}$ , $v_{BC}$ y $v_{CA}$ en modulación vectorial.....	130

Figura 3.106: Tensiones $v_{AB}$ , $v_{BC}$ y $v_{CA}$ en modulación vectorial con eliminación de armónicos pares.....	131
Figura 3.107: Tensiones $v_{AN}$ , $v_{BN}$ y $v_{CN}$ en modulación vectorial. ....	132
Figura 3.108: Tensiones $v_{An}$ , $v_{Bn}$ y $v_{Cn}$ en modulación vectorial. ....	133
Figura 3.109: Tensiones $v_{AN}$ , $v_{An}$ y $v_{Nn}$ en modulación vectorial. ....	134
Figura 3.110: Corrientes de fase $i_A$ , $i_B$ y $i_C$ en modulación discontinua. ....	135
Figura 3.111: Corrientes de fase $i_A$ , $i_{A1}$ y $i_{Aripple}$ en modulación vectorial. ....	136
Figura 3.112: Corrientes $i_I$ , $i_{CI}$ y $i_{RI}$ e $i_A$ en modulación vectorial.....	137
Figura 3.113: Armónicos de tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación vectorial normal. ....	138
Figura 3.114: Armónicos tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación vectorial con eliminación de armónicos pares para un valor de $mf = 24$ . ....	139
Figura 3.115: Imagen precalculada del barrido en modulación vectorial. ....	139



# Índice de Tablas

Tabla 1: Valor amplitud de los armónicos normalizado $v_{ohVI}$ según valor de $ma$ [6]. .....	18
Tabla 2: Valor eficaz de armónicos en zona lineal normalizado $v_{LLVI}$ según valor de $ma$ para valores de $mf$ grandes, impares y múltiplo de tres [6]. .....	25
Tabla 3: Posibles estados de las tres ramas del inversor trifásico. ....	30
Tabla 4: Secuencias de conmutación en modulación vectorial [8]. ....	32
Tabla 5: Presupuesto de ejecución por material (PEM). ....	149
Tabla 6: Honorarios del empleado. ....	149
Tabla 7: Presupuesto Total del TFG .....	150





Parte I

Resumen



# Resumen

En este Trabajo Fin de Grado se ha realizado un sistema software de simulación de inversores de potencia para uso docente.

El proyecto ha sido desarrollado mediante Simulink y GUIDE, ambas herramientas de Matlab, creando una interfaz gráfica sencilla y didáctica.

La aplicación permite la simulación de las diferentes tipologías de inversor estudiadas en distintas asignaturas relacionadas con la electrónica de potencia. En este Trabajo Fin de Grado se han estudiado diferentes técnicas de modulación PWM en inversores monofásicos y trifásicos, incluida la modulación vectorial en el inversor trifásico. Modificando los parámetros del circuito de una manera sencilla, el alumno puede visualizar el comportamiento del mismo, lo que facilita su comprensión.

**Palabras clave:** electrónica de potencia, inversores, simulación, educación.



# Abstract

This work proposes a software simulation system of power inverters using Matlab-Simulink and a Guide application.

The application allows to simulate different typologies of inverters studied in subjects related to Power Electronics. In this work several PWM modulation techniques in single-phase and three-phase inverters are studied, including space-vector modulation in three-phase inverters. The proposed application allows the students to better understand the different modulation techniques in power inverters and the effect of the different parameters in the whole system.

**Keywords:** power electronics, inverters, simulation tools, education.



# Resumen Extendido

En este Trabajo Fin de Grado se ha diseñado una herramienta para la simulación interactiva de convertidores DC/AC. La finalidad del proyecto es crear una herramienta para la docencia de inversores DC/AC en asignaturas relacionadas con la electrónica de potencia. Así, gracias a la herramienta de simulación propuesta, los alumnos podrán evaluar de forma empírica y autónoma los efectos que los diferentes parámetros tienen sobre el sistema.

En los últimos años, existe una motivación para modificar las técnicas educativas acorde a la nueva sociedad de la información. Esta nueva perspectiva implica incorporar las TIC's [1] en los procesos de enseñanza-aprendizaje.

Este proyecto consta de dos partes diferenciadas. En primer lugar se realiza la parte correspondiente a los inversores monofásicos y sus diferentes técnicas de modulación, las cuales se enumeran a continuación:

- Modulación PWM bipolar.
- Modulación PWM unipolar.
- Modulación mediante cancelación de tensión.
- Inversor Push Pull.

En segundo lugar se realiza la parte correspondiente a los inversores trifásicos, en los cuales también se emplean diferentes técnicas de modulación:

- Modulación PWM básica.
- Modulación PWM con inyección de tercer armónico.
- Modulación PWM con inyección de secuencia cero.
- Modulación PWM discontinua.
- Modulación Vectorial.
  - Normal.
  - Con eliminación de armónicos pares.

En ambos casos se ha realizado una interfaz gráfica con la herramienta GUIDE de Matlab, la cual se conecta con los diferentes modelos creados con la herramienta Simulink.

La herramienta es capaz de realizar la simulación de las diferentes modulaciones permitiendo modificar los parámetros de entrada del circuito, pudiéndose visualizar los valores de las variables calculadas más importantes. Además permite visualizar diferentes señales en el tiempo así como realizar simulaciones paramétricas.

Junto con todas las funcionalidades de simulación se incluyen los apuntes de la asignatura de Electrónica de Potencia así como dos videotutoriales para permitir al alumno trabajar de forma más autónoma y facilitando el auto-aprendizaje.

Este Trabajo Fin de Grado se ha desarrollado en el seno del proyecto de innovación docente “Necesidades y estrategias de formación en asignaturas de grado y postgrado relacionadas con la Ingeniería Electrónica” [ref. UAH/EV712].



Parte II

Memoria



# Capítulo 1

## Introducción

El estudio de los convertidores DC/AC es una parte relevante dentro de aquellas ingenierías con estudios de electrónica ya que permite transformar la energía de una forma a otra para poder utilizarla en su aplicación final. Sus aplicaciones son realmente variadas, ya que abarcan campos de la ingeniería tan dispares como la generación de energía eléctrica mediante renovables a la regulación de máquinas eléctricas, destacando las siguientes aplicaciones:

- Pequeñas fuentes de alimentación.
- Conversión de corriente continua generada por paneles solares fotovoltaicos.
- Puesta en marcha de motores de corriente alterna.

El objetivo de este Trabajo Fin de Grado consiste en facilitar la comprensión de los convertidores DC/AC gracias a una herramienta software, la cual permite interactuar con el usuario de forma sencilla e intuitiva.

Para la creación de este proyecto se han empleado herramientas del software Matlab, Simulink [2] y GUIDE [3]. Con la herramienta Simulink se han realizado los modelos sobre los diferentes tipos de circuitos y modulaciones estudiadas. Con la herramienta GUIDE se ha realizado la interfaz gráfica, en la cual se pueden configurar los diferentes parámetros para realizar la simulación deseada.

Este TFG (Trabajo Fin de Grado) se ha desarrollado en el seno del proyecto de innovación docente “Necesidades y estrategias de formación en asignaturas de grado y postgrado relacionadas con la Ingeniería Electrónica” [ref. UAH/EV712]. Además, este TFG supone la continuación al proyecto “Sistema HW de control PWM gestionado desde Matlab” [4] en el cual se realiza una herramienta similar para la simulación de convertidores DC/DC.

El uso de herramientas TIC para la mejora de la docencia se ha visto impulsado debido al cambio surgido con el Plan Bolonia, en el cual el alumno pasa de ser un sujeto pasivo a un sujeto activo dentro del aula, lo cual se facilita con este tipo de herramientas [5]. Además la aplicación incluye un videotutorial así como apuntes teóricos sobre estos circuitos. Gracias a esto se consigue aumentar la autonomía del alumno debido a que se simplifica la comprensión de la herramienta, promoviendo una forma de aprendizaje más autodidáctica.

## 1.1 Estructura del documento

Este documento está dividido en nueve grandes apartados, los cuales se resumen a continuación:

El primer apartado, como se ha podido observar, es un resumen del proyecto en el cual se explica en líneas generales.

En este segundo apartado se describen las bases del proyecto, explicando el funcionamiento de la herramienta y contextualizando las motivaciones.

En el tercer apartado se realiza una revisión teórica de los convertidores DC/AC tanto monofásicos como trifásicos.

En el cuarto apartado se explica el funcionamiento de la herramienta, analizando los resultados obtenidos y las características implementadas.

En el quinto apartado se resumen las conclusiones del proyecto así como las futuras ampliaciones de este.

En el sexto apartado se analiza el pliego de condiciones que se ha necesitado para desarrollar la herramienta.

En el séptimo apartado se calcula el presupuesto del proyecto.

En el octavo apartado se muestran algunos ejemplos de las funciones implementadas en el proyecto.

Por último, se incluye la bibliografía utilizada.

# Capítulo 2

## Revisión de convertidores DC/AC

### 2.1 Convertidor monofásico en puente completo

#### 2.1.1 Introducción

Un convertidor monofásico en puente completo transforma una señal de entrada continua en una señal de salida alterna.

El circuito que lo forma es el mostrado en la Figura 2.1.

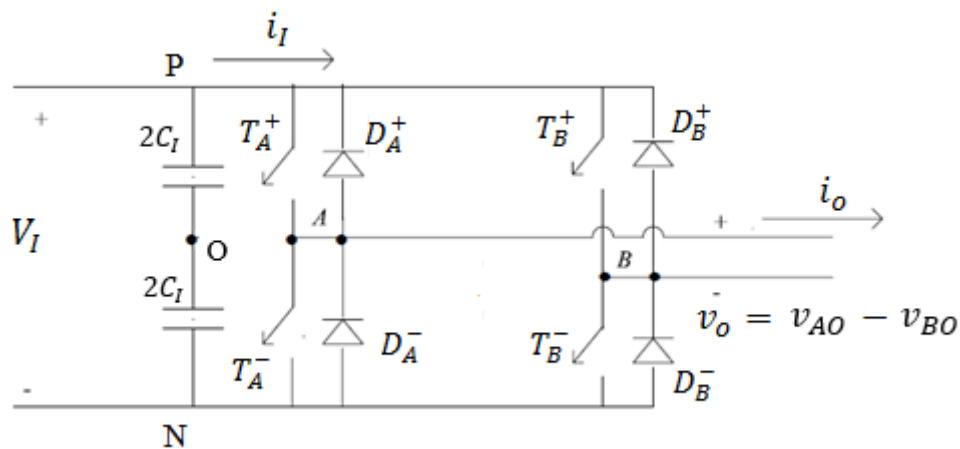


Figura 2.1: Circuito inversor en Puente completo [6].

Este tipo de implementación tiene la principal ventaja de soportar mayores tensiones de salida y, en consecuencia mayores potencias de salida, que en el caso del semipuerto, en el cual se tiene únicamente una rama.

Para esta topología de inversor se debe cumplir que los transistores de una misma rama siempre estén en estados contrarios, es decir:  $TA^+ = \overline{TA^-}$  y  $TB^+ = \overline{TB^-}$ .

Para el control de este convertidor se implementan diferentes tipos de modulación. En concreto, las que se estudiarán en este proyecto corresponden al tipo PWM (*Pulse Width Modulation*).

- Modulación PWM con tensión bipolar.
- Modulación PWM con tensión unipolar.
- Modulación PWM mediante cancelación de tensión.

Antes de entrar en detalle en cada una de las diferentes modulaciones se procederá a explicar algunos conceptos esenciales para su comprensión.

### Plan de conmutación por PWM

Para controlar las conmutaciones de los transistores se tiene una señal  $V_{control}$ , también llamada señal moduladora, que se compara con una señal triangular  $V_{tri}$  o señal portadora. Se pueden observar ambas señales en la Figura 2.2.

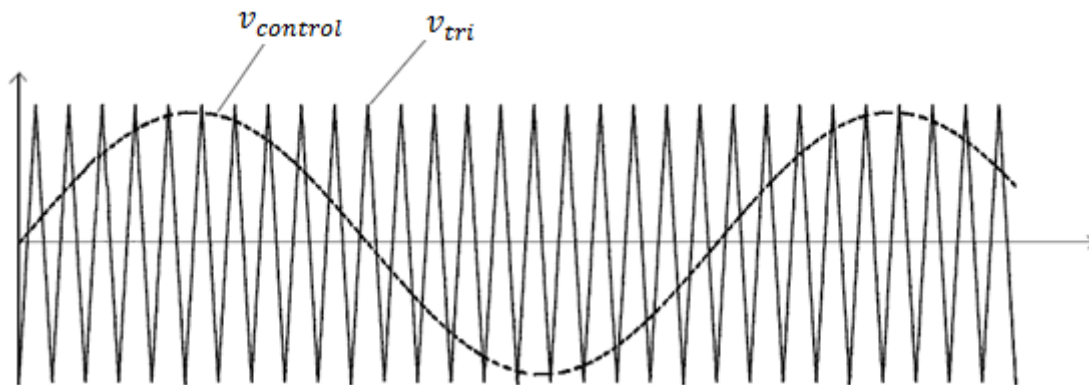


Figura 2.2: Señales portadora y moduladora [6].

La onda triangular  $V_{tri}$  es de una frecuencia igual a la frecuencia de conmutación de los interruptores,  $f_s$ , mientras que la frecuencia de la señal moduladora,  $f_1$ , es la frecuencia fundamental deseada en la tensión de salida del inversor.

La relación entre la frecuencia de la señal portadora y la frecuencia de la señal moduladora define el índice de modulación en frecuencia  $m_f$ , expresado en la ecuación (1).

$$m_f = \frac{f_s}{f_1} \quad (1)$$

Por otra parte, la relación entre las amplitudes de ambas señales define a su vez el índice de modulación en amplitud  $m_a$ , expresado en la ecuación (2).

$$m_a = \frac{\widehat{V}_{control}}{\widehat{V}_{tri}} \quad (2)$$

Una vez se han definido estos índices es posible describir diferentes situaciones en función del valor de los mismos.

- $m_f$  pequeña ( $m_f \leq 21$ ):

Para valores pequeños de  $m_f$  las señales  $V_{control}$  y  $V_{tri}$  deberían estar sincronizadas entre sí. La razón por la que se debe usar ambas señales en sincronía es debido a que, en caso de no hacerlo, la señal de salida estará compuesta por subarmónicos, que son indeseables en la mayoría de las aplicaciones.

- $m_f$  grande ( $m_f \geq 21$ ):

En caso de estar trabajando con valores de  $m_f$  grandes se podrán utilizar señales de control asíncronas, ya que los subarmónicos generados son de pequeño valor. No obstante, en ciertas aplicaciones como la alimentación de un motor de alterna, estos subarmónicos podrían generar grandes corrientes que serán indeseadas por lo que será conveniente tener precaución en la elección de PWM asíncrona.

- Zona lineal ( $m_a \leq 1$ ):

Para valores de  $m_a \leq 1$  el inversor trabaja en zona lineal. Este comportamiento lineal conlleva que la amplitud de salida sea proporcional a la tensión de salida, tal y como se indica en la expresión (3).

$$\widehat{V}_{o1} = m_a \cdot V_I \quad (3)$$

- Sobremodulación ( $1 \leq m_a \leq 3.24$ ):

En esta zona de trabajo el comportamiento de la señal no es lineal. En la zona de sobremodulación se consiguen amplitudes de salida mayores a la tensión  $V_I$ . El valor máximo de  $m_a$  con el cual se sigue trabajando en zona de sobremodulación es variable, siendo función de  $m_f$ . Además, estar en esta zona de trabajo conlleva obtener una señal de salida con mayor contenido armónico, el cual se analizará más en profundidad en las diferentes estrategias de modulación. El comportamiento de esta zona de trabajo se rige mediante la ecuación (4).

$$V_I \leq \widehat{V}_{o1} \leq \frac{4 * V_I}{\pi} \quad (4)$$

- Conmutación con onda cuadrada ( $m_a \geq 3.24$ ):

En este plan de conmutación, cada interruptor del inversor está en “on” durante medio periodo ( $180^\circ$ ). Esta zona de trabajo mantiene constante la amplitud de la señal de salida independientemente del valor de  $m_a$ , siendo esta la indicada en la ecuación (5).

$$\widehat{V}_{o1} \leq \frac{4 * V_I}{\pi} \quad (5)$$

La ventaja más importante de esta zona de trabajo consiste en que el número de cambios de estado de los interruptores (2 veces por ciclo) es menor con respecto a las otras zonas de trabajo. Esto que implica menores niveles de potencia consumidos. Por otro lado, el principal inconveniente está en que, para controlar la tensión de salida es necesario modificar la tensión de entrada ya que no existe ningún parámetro que permita regular esta amplitud.

La Figura 2.3 representa la síntesis de las tres zonas de trabajo mencionadas en función de  $m_a$ .

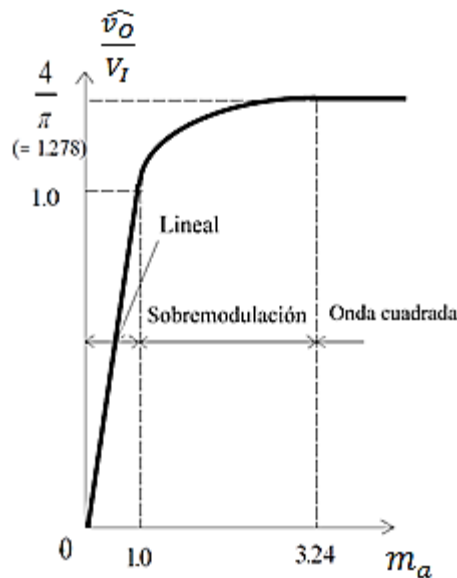


Figura 2.3: Zonas de funcionamiento en función de  $m_a$  [6].

Para este análisis se va a despreciar el efecto del tiempo muerto  $t_d$ . De aquí en adelante el paso de los interruptores de “on” a “off” y de “off” a “on” en los transistores de una misma rama se realizará de manera simultánea. Sin embargo, para evitar cortocircuitos o conducciones cruzadas se retrasa el paso de los interruptores de “off” a “on” para asegurar que nunca están ambos interruptores activados. El valor de tiempo muerto seleccionado dependerá del tipo de transistor, siendo de unos microsegundos en caso de estar trabajando con un MOSFET o de tiempos mayores para dispositivos de conmutaciones más lentas.

Los efectos de los tiempos muertos consisten en una disminución de la amplitud del primer armónico de la tensión de salida así como un incremento de armónicos de bajo orden, tal y como se observa en la Figura 2.4.



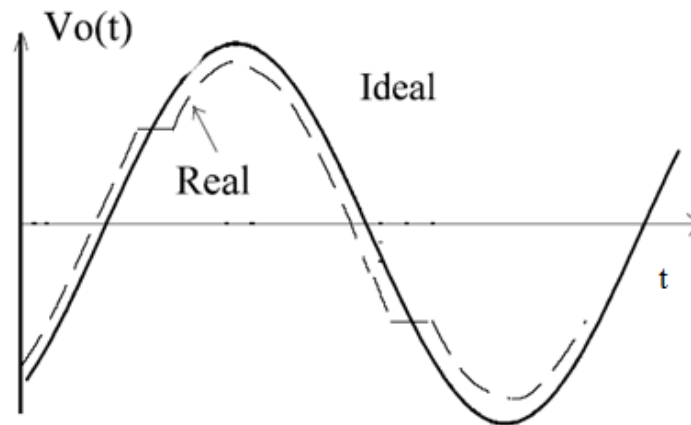


Figura 2.4: Efecto de tiempos muertos en la tensión de salida [6].

### 2.1.2 Modulación PWM bipolar

El esquema de modulación PWM bipolar se basa en que los interruptores de cada diagonal, es decir,  $(TA^+, TB^-)$  por un lado y  $(TA^-, TB^+)$  por el otro, se encuentran en el mismo estado. La salida de la rama A del inversor es negativa con respecto a la salida de la rama B, por lo que se tienen las expresiones (6) y (7).

$$V_{BO}(t) = -V_{AO}(t) \tag{6}$$

$$V_o(t) = V_{AO}(t) - V_{BO}(t) = 2V_{AO}(t) \tag{7}$$

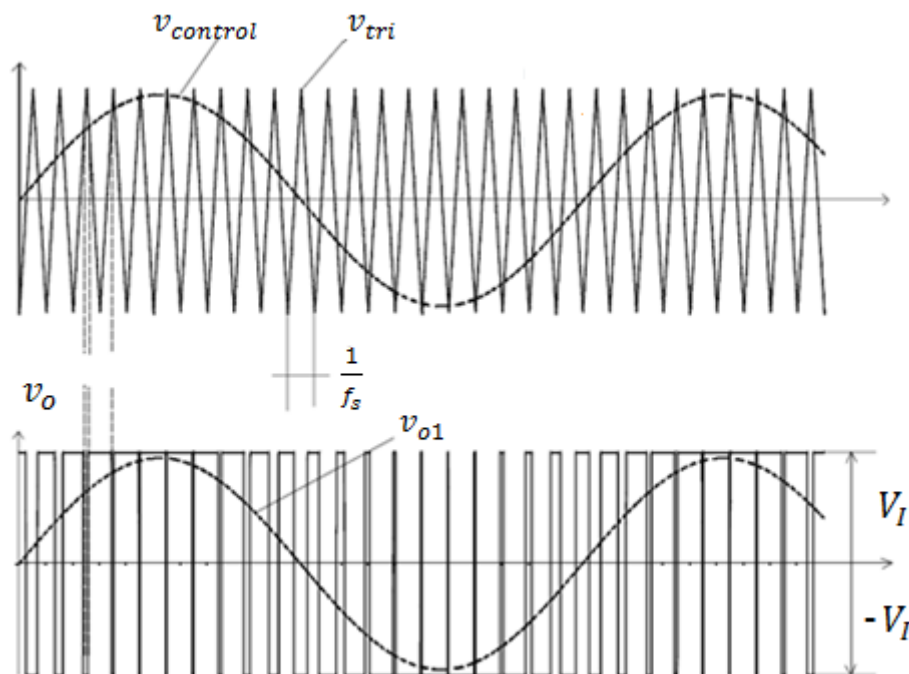


Figura 2.5: Tensión de salida y señales de control en modulación PWM bipolar [6].

En la Figura 2.5 se representa la forma de onda de la señal de salida  $v_o$ , así como la generación de los pulsos a partir de las señales moduladora y portadora. Además, en la Figura 2.5 se observa también que la tensión de salida conmuta entre  $\pm V_I$ ; es por ello que a este tipo de conmutación se le llama PWM con tensión bipolar.

Las amplitudes de los diferentes armónicos se pueden calcular a partir de la Tabla 1 mientras se trabaja en zona lineal, sin embargo, en zona de sobremodulación las componentes armónicas se distribuyen como en la Figura 2.6.

armónico/ $m_a$	0,2	0,4	0,6	0,8	1
<b>h 1 Fundamental</b>	0,2	0,4	0,6	0,8	1
$m_f$	1,242	1,150	1,006	0,818	0,601
$m_f \pm 2$		0,061	0,131	0,220	0,318
$m_f \pm 4$					0,018
$2m_f \pm 1$	0,193	0,326	0,370	0,314	0,181
$2m_f \pm 3$		0,024	0,071	0,139	0,212
$2m_f \pm 5$				0,013	0,033
$3m_f$	0,335	0,123	0,083	0,171	0,113
$3m_f \pm 2$	0,044	0,139	0,203	0,176	0,062
$3m_f \pm 4$		0,012	0,047	0,104	0,157
$3m_f \pm 6$				0,016	0,044
$4m_f \pm 1$	0,163	0,157	0,008	0,105	0,068
$3m_f \pm 3$	0,012	0,070	0,132	0,115	0,009
$3m_f \pm 5$				0,084	0,119
$3m_f \pm 7$				0,017	0,050

Tabla 1: Valor amplitud de los armónicos normalizado  $\widehat{v_{oh}}/V_I$  según valor de  $m_a$  [6].

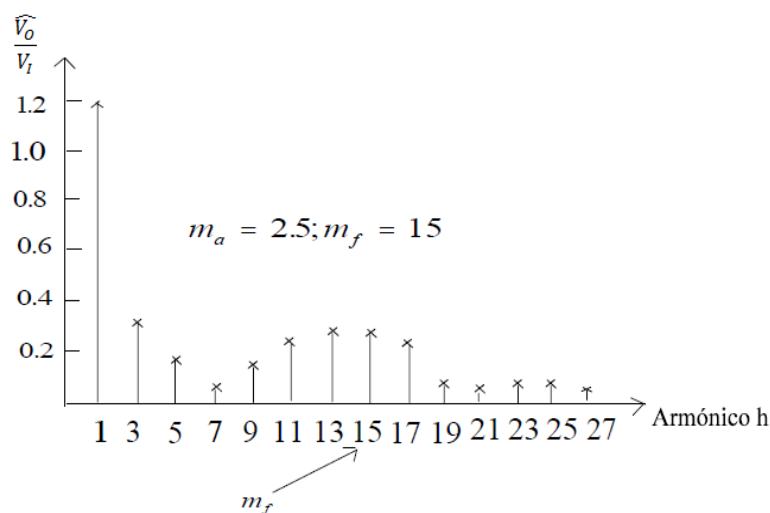


Figura 2.6: Armónicos en zona de sobremodulación [6].

Otro parámetro cuyo estudio es interesante es la corriente de entrada  $i_I$ . Para su cálculo se desprecian las componentes de alta frecuencia producidas por la tensión de entrada en

dispositivos reales. Esto se debe a que en el estudio realizado, la tensión de entrada  $V_I$  no tiene componente alterna, a pesar de que en la realidad sí contendrá componentes armónicas ya que normalmente será obtenida de una señal alterna de red rectificadas. Una vez asumida esta hipótesis se puede calcular la corriente de entrada a partir de un análisis de potencia, ya que la potencia instantánea a la entrada debe ser igual a la potencia instantánea a la salida, pues no hay dispositivos en el convertidor que almacenen energía.

A partir de este planteamiento se obtiene la expresión (8), donde  $\Phi$  es el desfase entre  $I_O$  y  $V_O$ .

$$i_1(t) = \frac{V_O I_O}{V_I} \cos\Phi - \frac{V_O I_O}{V_I} \cos(2\omega t - \Phi) \quad (8)$$

La expresión (8) da como resultado una componente continua y un armónico del doble de frecuencia que la frecuencia fundamental, los cuales están definidos en las ecuaciones (9) y (10).

$$I_1 = \frac{V_O I_O}{V_I} \cos\Phi \quad (9)$$

$$I_{12} = \frac{1}{\sqrt{2}} \frac{V_O I_O}{V_I} \quad (10)$$

Este comportamiento se representa en la Figura 2.7.

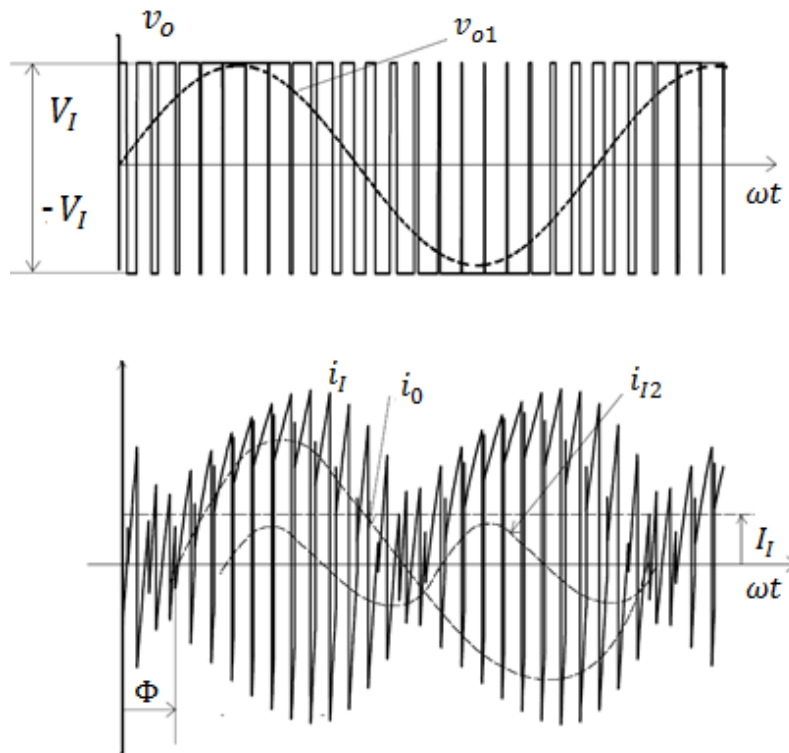


Figura 2.7: Tensión y corriente de salida en modulación bipolar [6].

### 2.1.3 Modulación PWM unipolar

En el esquema de modulación PWM unipolar, a diferencia de la modulación PWM bipolar antes vista, los interruptores del inversor no conmutan por diagonales. En esta ocasión cada una de las ramas del puente completo comparte la señal portadora, estando gobernadas por su propia señal moduladora,  $V_{control}$  y  $-V_{control}$  respectivamente.

Este planteamiento genera las señales de control dadas por la expresión (11).

$$\begin{aligned}
 v_{control} > v_{tri}; & TA^+ \text{ on y } v_{AN} = V_I \\
 v_{control} < v_{tri}; & TA^- \text{ on y } v_{AN} = 0 \\
 -v_{control} > v_{tri}; & TB^+ \text{ on y } v_{BN} = V_I \\
 -v_{control} < v_{tri}; & TB^- \text{ on y } v_{BN} = 0
 \end{aligned} \tag{11}$$

Estas cuatro señales de control generan cuatro estados diferentes descritos en la expresión (12).

$$\begin{aligned}
 TA^+, TB^- \text{ on: } & v_{AN} = V_I; V_{BN} = 0; v_O = V_I \\
 TA^-, TB^+ \text{ on: } & v_{AN} = 0; V_{BN} = V_I; v_O = -V_I \\
 TA^+, TB^+ \text{ on: } & v_{AN} = V_I; V_{BN} = V_I; v_O = 0 \\
 TA^-, TB^- \text{ on: } & v_{AN} = 0; V_{BN} = 0; v_O = 0
 \end{aligned} \tag{12}$$

En la Figura 2.8 se puede observar la evolución temporal de las señales de control, así como de la tensión de salida y la tensión en cada una de las ramas.

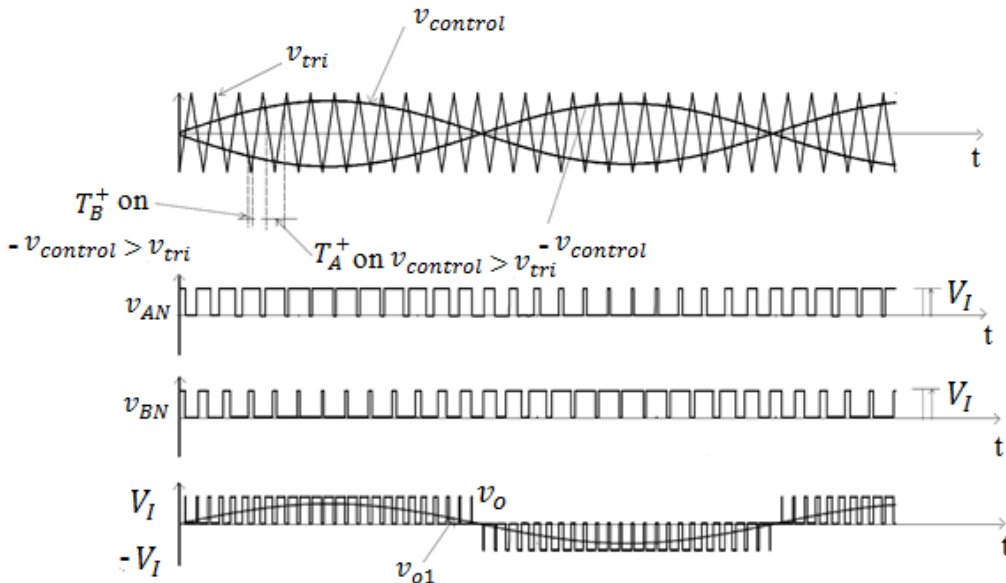


Figura 2.8: Tensión de salida en modulación PWM unipolar [6].

Como se observa en Figura 2.8, la tensión de salida  $v_O$  conmuta entre 0 y  $+V_I$  o entre 0 y  $-V_I$ . Es por esta razón que a esta modulación se la denomina unipolar.

Esta modulación presenta dos ventajas con respecto a la modulación bipolar. La primera de ellas reside en el menor salto de tensión en las conmutaciones, ya que en la modulación anterior el salto era de  $2V_I$  mientras que en esta modulación el salto se reduce a la mitad,  $V_I$ . La otra gran ventaja de esta modulación se debe a que se trabaja al doble de frecuencia de conmutación que en el caso bipolar. Esto produce que los armónicos más bajos aparezcan como bandas laterales del doble de frecuencia de conmutación. Además, las bandas laterales de la frecuencia de conmutación desaparecen.

Al igual que se estudió en la modulación bipolar, en la Figura 2.9 se puede visualizar la corriente a la entrada del inversor. En esta imagen se aprecia como el rizado es notablemente menor al visto en el caso bipolar.

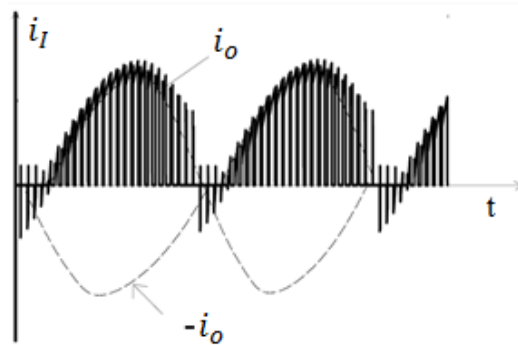


Figura 2.9: Corriente de salida modulación unipolar para el caso lineal [6].

### 2.1.4 Modulación mediante cancelación de tensión

La técnica de modulación mediante cancelación de tensión se basa en la combinación de conmutación con onda cuadrada y modulación PWM unipolar. En la Figura 2.10 se muestran las tensiones de salida del inversor.

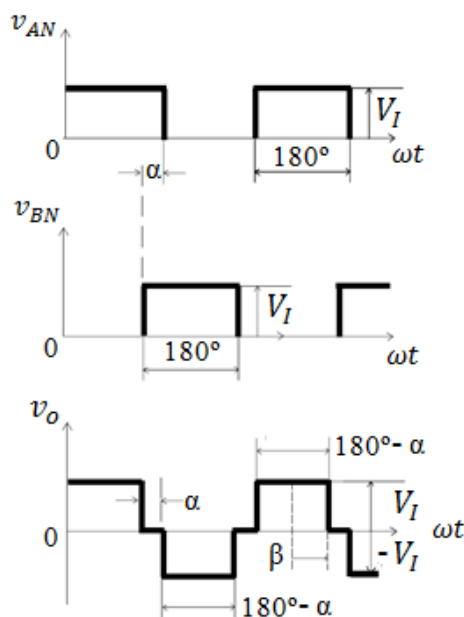


Figura 2.10: Tensión de salida en modulación mediante cancelación de tensión [6].

En la Figura 2.10 puede verse como los interruptores de ambas ramas son controlados de forma independiente, como ocurre en la modulación PWM unipolar. En este tipo de modulación todos los interruptores tienen un ciclo de trabajo de 0.5, siendo el parámetro  $\alpha$  el empleado para controlar el inversor, ya que en esta modulación no se tienen los parámetros  $m_a$  ni  $m_f$ .

El parámetro  $\alpha$ , mostrado en en la Figura 2.10, define el periodo de tiempo durante el cual los transistores de la rama superior están en el mismo estado.

En los intervalos de solapamiento la salida es igual a 0 debido al estado de “on” de los dos interruptores superiores o de los dos interruptores inferiores.

La componente fundamental viene dada por las expresiones (13) y (14). Para su cálculo así como el de los diferentes armónicos de la señal, se emplea el parámetro  $\beta = 90^\circ - 0.5\alpha$  simplificando así el resultado obtenido.

$$\widehat{V}_{O,h} = \frac{2}{\pi} \int_{-\pi/2}^{\pi/2} V_O \cos(h\theta) d\theta \quad (13)$$

$$\widehat{V}_{O,h} = \frac{4}{\pi h} V_I \text{sen}(h\beta) \quad (14)$$

En la Figura 2.11 se puede apreciar la variación de la componente fundamental de la tensión de salida así como de los armónicos en función del parámetro  $\alpha$ , normalizado con respecto a la componente fundamental con onda cuadrada ( $\alpha=0$ ).

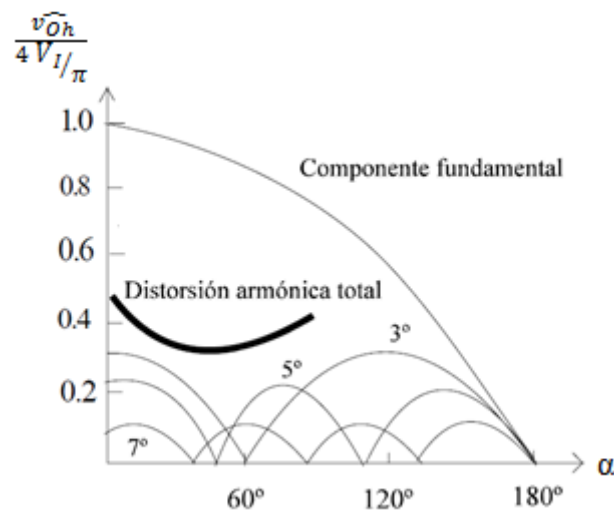


Figura 2.11: Armónicos y THD de la tensión de salida en modulación mediante cancelación de tensión [6].

## 2.2 Convertidor monofásico Push Pull

El inversor monofásico Push Pull, a diferencia del antes estudiado, requiere de un transformador que conecte la entrada con la salida. En la Figura 2.12 se muestra el circuito de este inversor.

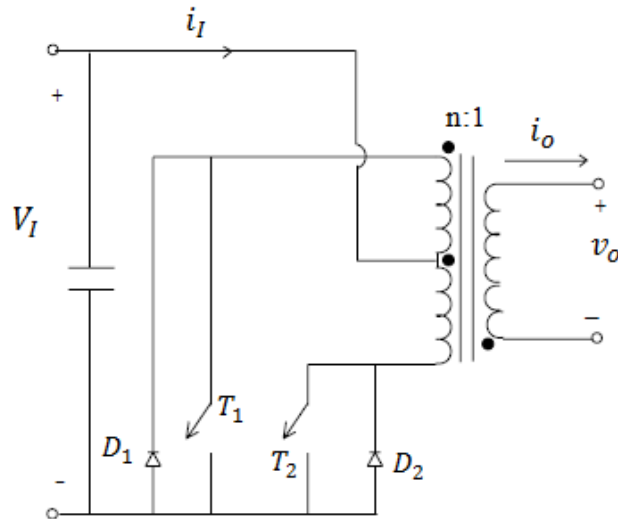


Figura 2.12: Circuito inversor Push Pull [6].

El inversor Push Pull puede trabajar en modo PWM o en modo de onda cuadrada, siendo las formas de control análogas a las estudiadas para el puente completo con la modulación PWM unipolar. Dichas formas de control vienen dadas por las expresiones (15) y (16), añadiendo un factor  $N$  cuyo significado viene dado por la relación de espiras del transformador.

$$\widehat{V}_{o1} = m_a * \frac{V_I}{n} \quad m_a \leq 1 \quad (15)$$

$$\frac{V_I}{n} \leq \widehat{V}_{o1} \leq \frac{4 * V_I}{\pi * n} \quad m_a \geq 1 \quad (16)$$

La principal ventaja del circuito en Push Pull es que no conduce más de un interruptor en serie en un mismo instante de tiempo. Esto es importante si la entrada continua del convertidor proviene de una fuente de tensión de baja impedancia, ya que una caída de tensión sobre más de un interruptor supone una reducción del rendimiento del inversor.

Además se debe tener en cuenta que, en cada conmutación la corriente del primario se desplaza de una mitad a la otra, lo que requiere un acoplamiento magnético muy bueno entre los dos medios devanados del primario para reducir la energía disipada. Para la disipación de esta energía, además de los interruptores, se cuenta con un circuito amortiguador “red snubber” que se implementa para protegerlo.

Por último debido al gran número de espiras del transformador se tiene una alta inductancia de conmutación, lo que dificulta que un inversor Push Pull con PWM funcione con frecuencias de conmutación superiores a 1KHz.

La principal ventaja con respecto al puente completo es el aislamiento eléctrico entre la entrada y la salida del inversor, lo cual puede resultar muy interesante para ciertas aplicaciones.

## 2.3 Convertidor trifásico en puente completo

### 2.3.1 Introducción

Un inversor trifásico está formado por tres inversores monofásicos, donde cada inversor produce una salida desfasada  $120^\circ$  con respecto a la anterior, manteniendo la misma amplitud y frecuencia entre ellas.

El circuito inversor trifásico más empleado está formado por tres ramas, tal y como se muestra en la Figura 2.13. Esta configuración permite que cada rama, las cuales están asociadas cada una a una fase, dependa únicamente de  $V_I$  y del estado de los interruptores siendo independiente de la corriente por la carga, ya que alguno de los interruptores de la carga está siempre en “on”.

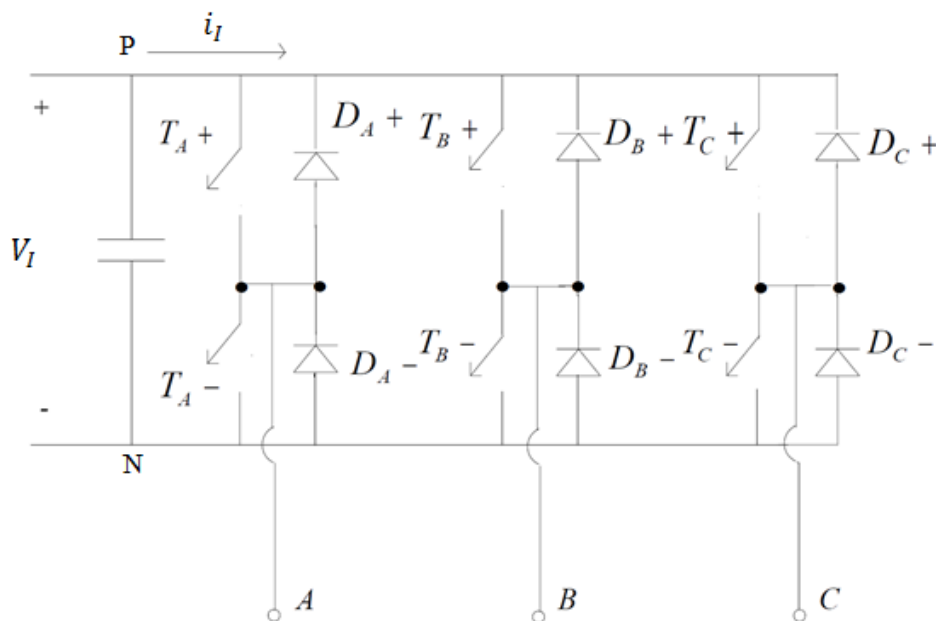


Figura 2.13: Circuito inversor trifásico en tres ramas.

En los inversores trifásicos con modulación PWM, al igual que ocurría en los inversores monofásicos, hay una señal triangular  $V_{tri}$  y una señal de control para cada fase,  $V_{controlA}$ ,  $V_{controlB}$  y  $V_{controlC}$ .

### 2.3.2 Modulación PWM clásica

Al igual que ocurría en los inversores monofásicos, el objetivo de estos inversores con modulación PWM consiste en controlar la tensión de salida en magnitud y frecuencia con una tensión de entrada constante. Para conseguir esta tensión, al igual que ocurría en la modulación PWM bipolar, donde se comparaba una señal triangular con una señal sinusoidal de control, se compara una señal triangular con tres señales sinusoidales, una para cada rama del puente, desfasadas  $120^\circ$  entre sí tal y como se representa en la Figura 2.14.



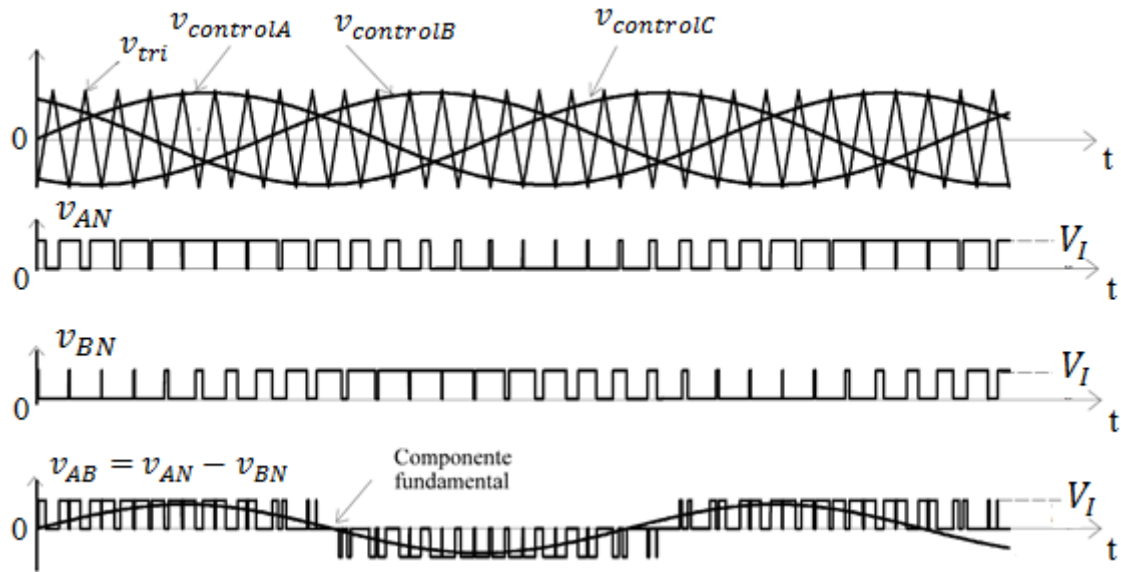


Figura 2.14: Señales de control en modulación PWM y tensión de salida [6].

Como se observa en la Figura 2.14, el valor de la componente continua de la tensión  $v_{AN}$  es igual al valor en la tensión  $v_{BN}$ . Por tanto, al igual que ocurría en los inversores monofásicos, la componente continua de la tensión de línea  $v_{AB}$ , así como del resto, es igual a cero.

En los inversores trifásicos es interesante analizar el contenido armónico de las tensiones de línea. Para reducir el contenido armónico de estas tensiones la selección más apropiada de valores de  $m_f$  es de valores impares y múltiplos de tres ya que de esta forma se consigue eliminar los armónicos pares, así como los armónicos múltiplos de  $m_f$ , por lo que únicamente quedarían las bandas laterales de  $m_f$  con armónicos pares. El cálculo de estos armónicos se resume en la Tabla 2.

armónico/ $m_a$	0,2	0,4	0,6	0,8	1
<b>h 1</b> <b>Fundamental</b>	0.122	0.245	0.367	0.490	0.612
$m_f \pm 2$	0.010	0.037	0.080	0.135	0.195
$m_f \pm 4$				0.005	0.011
$2m_f \pm 1$	0.116	0.200	0.227	0.192	0.111
$2m_f \pm 5$				0.008	0.020
$3m_f \pm 2$	0.027	0.085	0.124	0.108	0.038
$3m_f \pm 4$		0.007	0.029	0.064	0.096
$4m_f \pm 1$	0.100	0.096	0.005	0.064	0.042
$3m_f \pm 5$			0.3021	0.051	0.073
$3m_f \pm 7$				0.010	0.030

Tabla 2: Valor eficaz de armónicos en zona lineal normalizado  $v_{LL}/V_I$  según valor de  $m_a$  para valores de  $m_f$  grandes, impares y múltiplo de tres [6].

Al igual que en el caso de los inversores monofásicos, se pueden definir diferentes zonas de trabajo en función de los valores de  $m_a$ .

- Zona lineal ( $m_a \leq 1$ ):

Para valores de  $m_a \leq 1$  el inversor trabaja en zona lineal. Este comportamiento lineal conlleva que la amplitud de salida sea proporcional a la tensión de salida, tal y como se indica en la expresión (17).

$$\widehat{V_{AN1}} = m_a * \frac{V_I}{2} \quad (17)$$

De la expresión (17), se puede deducir que la tensión eficaz de línea es la dada por (18):

$$V_{LL1} = \frac{\sqrt{3}}{\sqrt{2}} \widehat{V_{AN1}} = \frac{\sqrt{3}}{2*\sqrt{2}} m_a * V_I \approx 0.612 m_a * V_I \quad (18)$$

- Sobremodulación ( $1 \leq m_a \leq 3.24$ ):

En esta zona de trabajo el comportamiento de la señal deja de ser lineal, lo que permite tener mayores amplitudes de  $V_{LL1}$ . Trabajar en esta zona de trabajo conlleva obtener una señal de salida con mayor cantidad de armónicos, sin embargo los armónicos dominantes no poseen amplitudes tan elevadas como en la zona lineal. El comportamiento de esta zona de trabajo se rige mediante la ecuación (19).

$$0.612 V_I \leq V_{LL1} \leq 0.78 V_I \quad (19)$$

- Conmutación con onda cuadrada ( $m_a \geq 3.24$ ):

En este plan de conmutación, cada interruptor del inversor está en “on” durante medio periodo ( $180^\circ$ ), por tanto, en cualquier instante de tiempo hay tres interruptores en “on”. Esta zona de trabajo mantiene constante la amplitud de la señal de salida independientemente del valor de  $m_a$ , siendo este el indicado en la ecuación (20).

$$V_{LL1} = \frac{\sqrt{6} * V_I}{\pi} \approx 0.78 V_I \quad (20)$$

La Figura 2.15 representa la síntesis de las tres zonas de trabajo mencionadas en función de  $m_a$ .

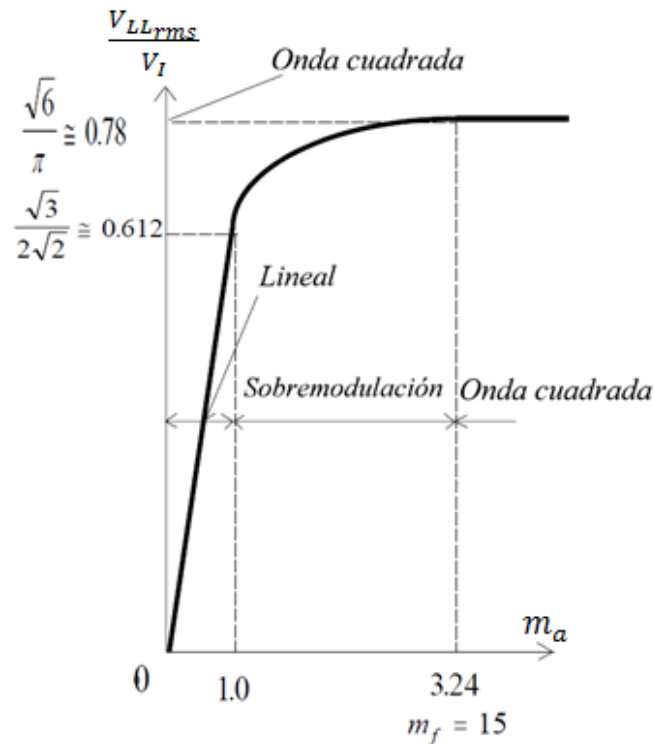


Figura 2.15: Zonas de funcionamiento en función de  $m_a$  [6].

A diferencia de lo que ocurría en los inversores monofásicos, la corriente de entrada no contiene un armónico del doble de frecuencia que la frecuencia de salida. No obstante, como puede observarse en la Figura 2.16, esta corriente de entrada está formada por componentes de alta frecuencia, las cuales tendrían un efecto negativo en un inversor trifásico real.

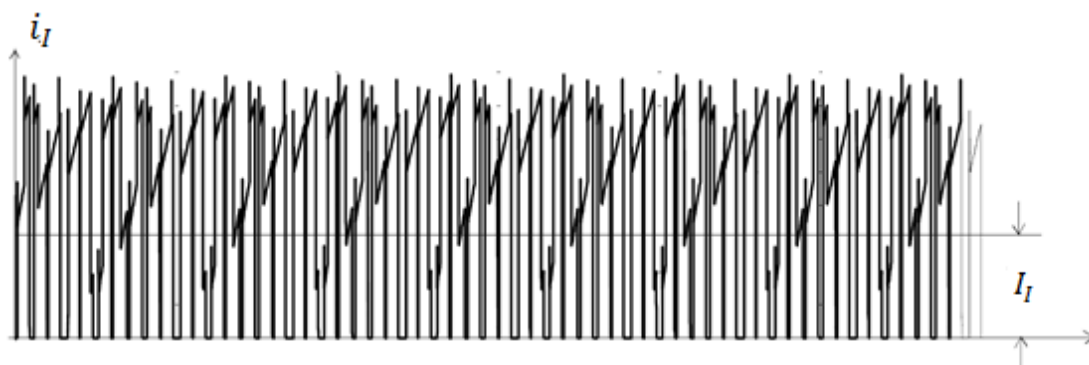


Figura 2.16: Corriente de entrada de un inversor trifásico [6].

### 2.3.3 Modulación PWM con inyección de secuencia cero: método inyección tercer armónico (THIPWM)

La técnica de modulación THIPWM (*Third harmonic injection PWM*) es una variante de la modulación PWM básica. Consiste en inyectar el tercer armónico a la señal moduladora con la

que se compara la señal triangular. Para ilustrar el funcionamiento de esta técnica de modulación se tiene en la Figura 2.17 un esquema con su funcionamiento simplificado.

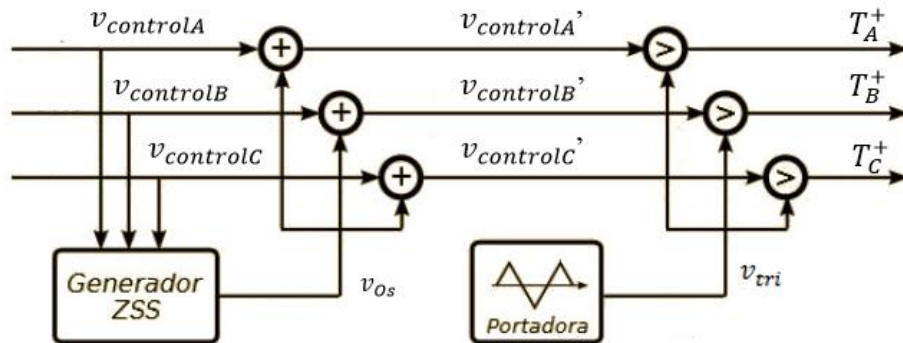


Figura 2.17: Esquema de funcionamiento de modulación PWM con inyección de secuencia cero.

La señal moduladora para esta técnica de modulación, así como las señales a partir de las que es generada, se representan en la Figura 2.18.

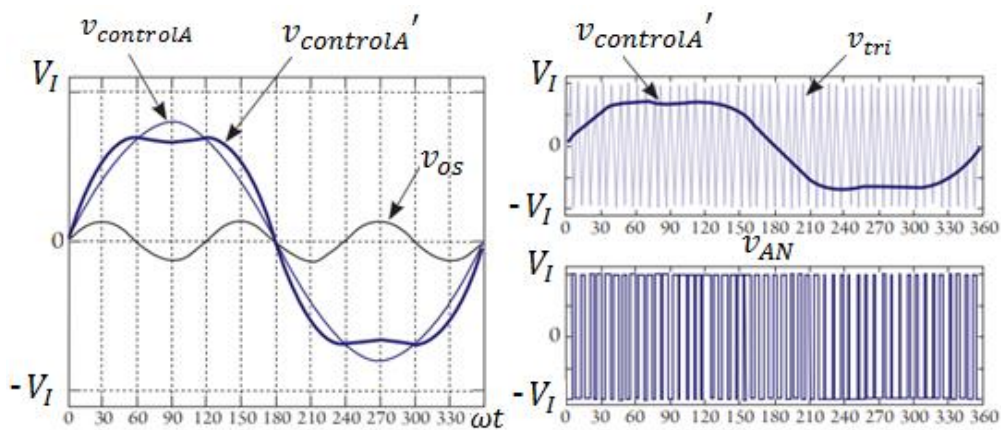


Figura 2.18: Señales de control en modulación THIPWM [7].

El objetivo de esta técnica de modulación consiste en aumentar la componente fundamental de tensión sin salir de la zona lineal, aumentando en un 11.7% para un tercer armónico de amplitud  $\frac{A}{4}$  y de un 15% para un tercer armónico de amplitud  $\frac{A}{6}$ . Este segundo caso aumenta la distorsión por lo general. El tercer armónico se configura según (21).

$$v_{0s} = \frac{A}{4} \sin(3\omega t) \quad (21)$$

El mayor inconveniente de esta técnica de modulación es debido al cálculo del tercer armónico, el cual puede generar una sobrecarga computacional.

### 2.3.4 Modulación PWM con inyección de secuencia cero: método PWM con vectores espaciales basada en portadora (CB-SVPWM)

La técnica de modulación CB-SVPWM (*Carrier Based Space Vector PWM*) se basa en el mismo principio de la modulación PWM con inyección de tercer armónico, sin embargo se inyecta una secuencia cero calculada mediante la expresión (22).

$$v_{0s} = -\frac{1}{2} [\max\{v_{controlA}, v_{controlB}, v_{controlC}\} + \min\{v_{controlA}, v_{controlB}, v_{controlC}\}] \quad (22)$$

El resultado de esta señal es una señal triangular con el triple de frecuencia de la frecuencia fundamental y amplitud  $\frac{A}{4}$ , tal y como se muestra en la Figura 2.19.

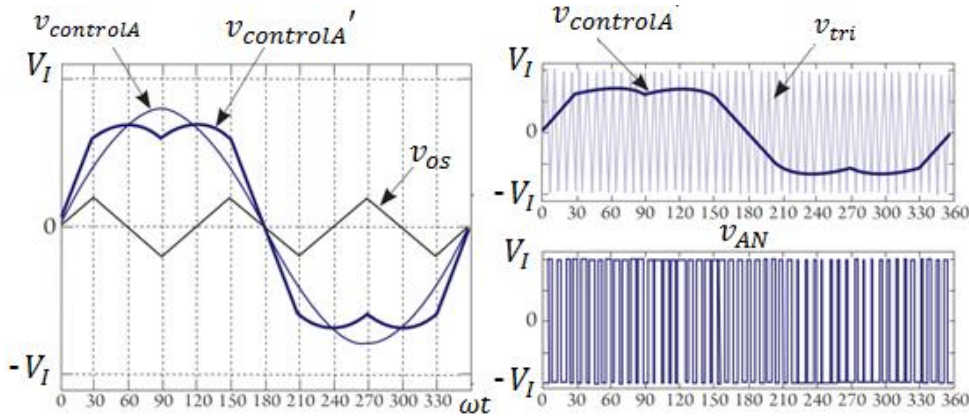


Figura 2.19: Señales de control con modulación CB-SVPWM [7].

El objetivo perseguido con esta técnica de modulación es mejorar el contenido armónico de la señal, concentrando los armónicos en la frecuencia de la portadora y sus múltiplos.

### 2.3.5 Modulación PWM con inyección de secuencia cero: método discontinuo (DPWM)

La técnica de modulación DPWM (*Discontinuous PWM*) genera conmutaciones durante 2/3 del periodo de la señal moduladora. La secuencia inyectada consigue llevar a una de las tres señales moduladoras al nivel del pico positivo o negativo de la señal triangular.

Para elegir la fase saturada se aplican diferentes reglas de selección, sin embargo en este trabajo únicamente se estudia una de ellas, que es mostrada en la condición (23).

$$IF(|v_{controlA}| \geq |v_{controlB}| \text{ AND } |v_{controlA}| \geq |v_{controlC}|) \quad (23)$$

Una vez se cumple la ecuación (23) se calcula la secuencia inyectada en la ecuación (24).

$$v_{0s} = \text{sign}(v_{controlA}) \frac{V_I}{2} - v_{controlA} \quad (24)$$

Esta selección genera una señal moduladora como la mostrada en la Figura 2.20.

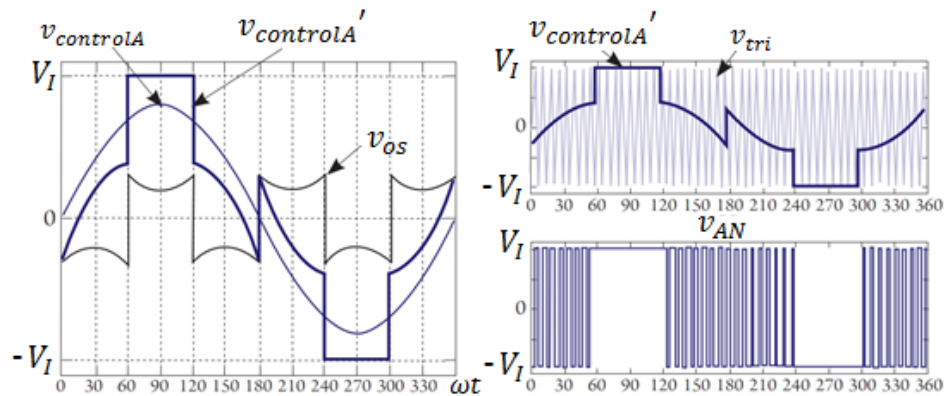


Figura 2.20: Señales de control en modulación DPWM [7].

Esta técnica de modulación, independientemente de la regla de selección, persigue mejorar el contenido armónico en zona lineal, así como minimizar las pérdidas por conmutación debido a menor número de conmutaciones.

### 2.3.6 Modulación vectorial

Esta técnica de modulación es posiblemente la más popular de todas, especialmente en el campo del control de máquinas eléctricas.

Este nuevo enfoque realizado en esta técnica de modulación implica sintetizar el estado de los transistores del puente en un vector  $\alpha\beta$ .

La generación del vector  $\alpha\beta$  se realiza mediante una combinación lineal de las coordenadas (a,b,c), tal y como se muestra en la ecuación (25).

$$\begin{pmatrix} x_\alpha \\ x_\beta \end{pmatrix} = k \begin{pmatrix} 1 & -1 & -1 \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \end{pmatrix} \begin{pmatrix} x_a \\ x_b \\ x_c \end{pmatrix} \quad (25)$$

Siendo  $k = \frac{2}{3}$  si la transformación es invariante a la amplitud o bien  $k = \sqrt{\frac{2}{3}}$  si la transformada es invariante a la potencia.

El estado de los interruptores del convertidor puede representarse por estados de conmutación, teniendo dos posibles estados para cada rama, los cuales se resumen en la Tabla 3.

Estado	Rama A			Rama B			Rama C		
	$T_A^+$	$T_A^-$	$v_{AN}$	$T_B^+$	$T_B^-$	$v_{BN}$	$T_C^+$	$T_C^-$	$v_{CN}$
<b>P</b>	On	Off	$V_I$	On	Off	$V_I$	On	Off	$V_I$
<b>O</b>	Off	On	0	Off	On	0	Off	On	0

Tabla 3: Posibles estados de las tres ramas del inversor trifásico.

Esta tabla de estados genera ocho posibles estados de conmutación, tal y como se muestra en la Figura 2.21.

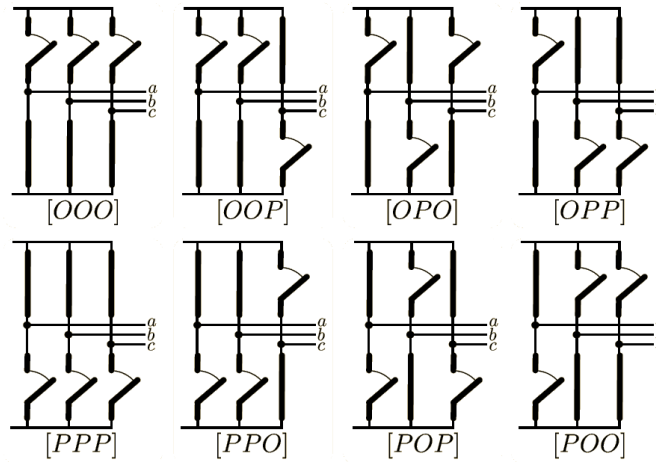


Figura 2.21: Posibles estados de los transistores del puente en modulación vectorial.

Estos ocho posibles estados pueden representarse como un hexágono tal y como se muestra en la Figura 2.22 a), formando una base de vectores. Cada  $T_s$ , que es el periodo durante el cual se realizará una secuencia de conmutación, se selecciona un nuevo vector  $v_{abc}$ . Este se transforma en coordenadas  $v_{\alpha\beta}$ . Utilizando los vectores estacionarios más próximos la tensión de salida durante ese periodo de tiempo  $T_s$  debe ser, de media, igual a la tensión  $v_{\alpha\beta}$ .

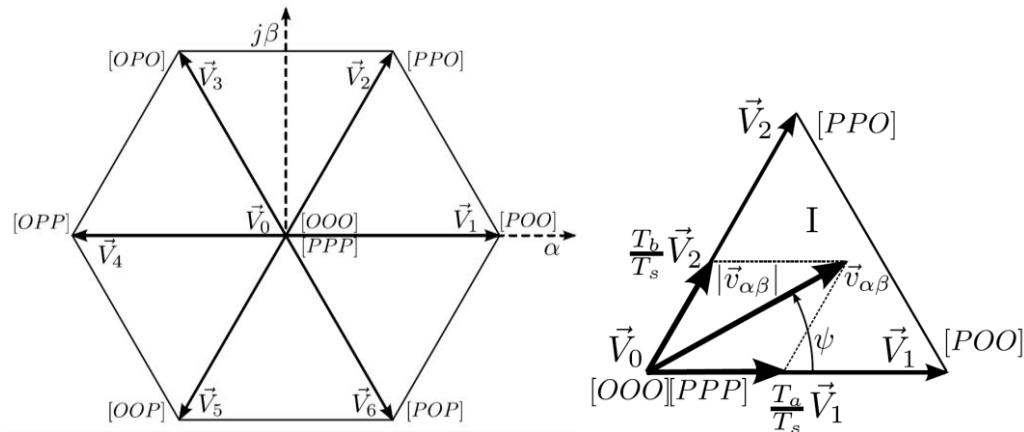


Figura 2.22: a) Hexágono de vectores estacionarios, b) Primer semisextante en detalle.

Para conseguir este objetivo, se deben calcular tres tiempos de conmutación definidos por la expresión (25), los cuales serán función del módulo y la fase del vector  $v_{\alpha\beta}$ .

$$T_a = \frac{\sqrt{3}T_s |v_{\alpha\beta}|}{V_1} \text{sen}\left(\frac{\pi}{3} - \psi\right)$$

$$T_b = \frac{\sqrt{3}T_s |v_{\alpha\beta}|}{V_1} \text{sen}(\psi) \quad (25)$$

$$T_c = T_s - (T_a + T_b)$$

Una vez conocidos los vectores que se deben emplear y la duración de cada estado debe definirse la secuencia a emplear. Esta secuencia de cambio debe involucrar un cambio en una sola rama por transición, así como emplear los estados redundantes para reducir el número de conmutaciones. Esto genera unas secuencias de cambio sintetizadas en la Tabla 4.

Sector	1	2	3	4	5	6	7
I	$\vec{v}_0$ [OOO]	$\vec{v}_1$ [POO]	$\vec{v}_2$ [PPO]	$\vec{v}_0$ [PPP]	$\vec{v}_2$ [PPO]	$\vec{v}_1$ [POO]	$\vec{v}_0$ [OOO]
II	$\vec{v}_0$ [OOO]	$\vec{v}_3$ [OPO]	$\vec{v}_2$ [PPO]	$\vec{v}_0$ [PPP]	$\vec{v}_2$ [PPO]	$\vec{v}_3$ [OPO]	$\vec{v}_0$ [OOO]
III	$\vec{v}_0$ [OOO]	$\vec{v}_3$ [OPO]	$\vec{v}_4$ [OPP]	$\vec{v}_0$ [PPP]	$\vec{v}_4$ [OPP]	$\vec{v}_3$ [OPO]	$\vec{v}_0$ [OOO]
IV	$\vec{v}_0$ [OOO]	$\vec{v}_5$ [OOP]	$\vec{v}_4$ [OPP]	$\vec{v}_0$ [PPP]	$\vec{v}_4$ [OPP]	$\vec{v}_5$ [OOP]	$\vec{v}_0$ [OOO]
V	$\vec{v}_0$ [OOO]	$\vec{v}_5$ [OOP]	$\vec{v}_6$ [POP]	$\vec{v}_0$ [PPP]	$\vec{v}_6$ [POP]	$\vec{v}_5$ [OOP]	$\vec{v}_0$ [OOO]
VI	$\vec{v}_0$ [OOO]	$\vec{v}_1$ [POO]	$\vec{v}_6$ [POP]	$\vec{v}_0$ [PPP]	$\vec{v}_6$ [POP]	$\vec{v}_1$ [POO]	$\vec{v}_0$ [OOO]

Tabla 4: Secuencias de conmutación en modulación vectorial [8].

Un ejemplo de secuencia de cambio en el primer sextante es la representada en la Figura 2.23.

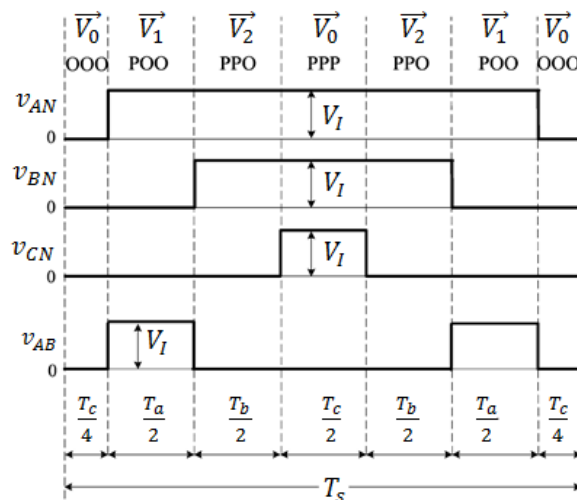


Figura 2.23: Secuencia de conmutación del primer semisextante [8].



En la Figura 2.24, se puede observar las tensiones generadas, así como el contenido armónico para una señal de valores  $f_1 = 60\text{Hz}$ ,  $f_s = 720\text{Hz}$  y  $m_a = 0.8$ .

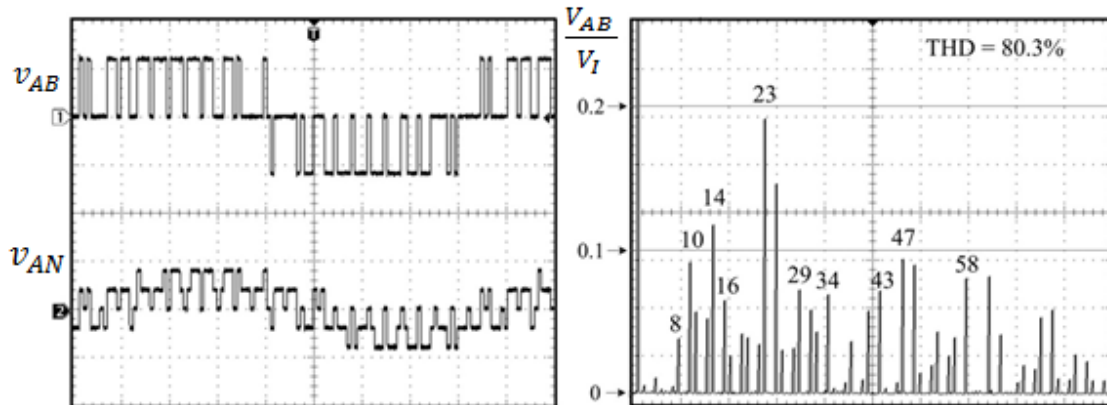


Figura 2.24: a) Tensión de salida en modulación SVM, b) Armónicos de la tensión de salida [8].

Al no tener simetría en media onda la señal, tal y como puede verse en la Figura 2.24, se generan armónicos pares en la señal. Estos armónicos no pueden tener cualquier valor, ya que para ciertas aplicaciones de red deben cumplir con ciertos estándares (IEEE 519-1992).

Con el fin de evitar este problema se puede modificar esta técnica de modulación para eliminar, o al menos minimizar, el valor de estos armónicos.

La técnica de modulación vectorial con eliminación de armónicos pares se basa en crear dos tipos de secuencia en función de la colocación de los vectores nulos. Una secuencia tipo A, donde se comienza por el vector nulo [OOO] y una secuencia tipo B, donde se comienza por el vector nulo [PPP]. En la Figura 2.25 puede observarse la secuencia tipo A y tipo B para un vector perteneciente al sector IV.

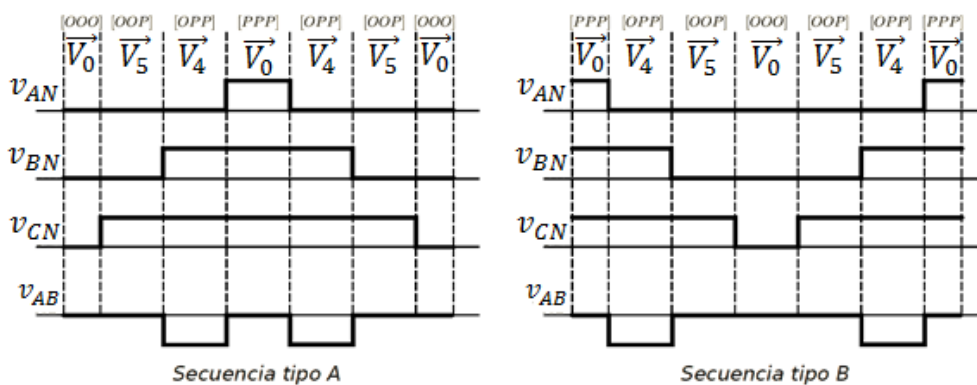


Figura 2.25: Secuencias de conmutación tipo A y tipo B en modulación vectorial [8].

Para definir cuándo se debe emplear un tipo de secuencia de conmutación u otra, se dividen los sextantes del hexágono en dos semisextantes tal y como se muestra en la Figura 2.26.

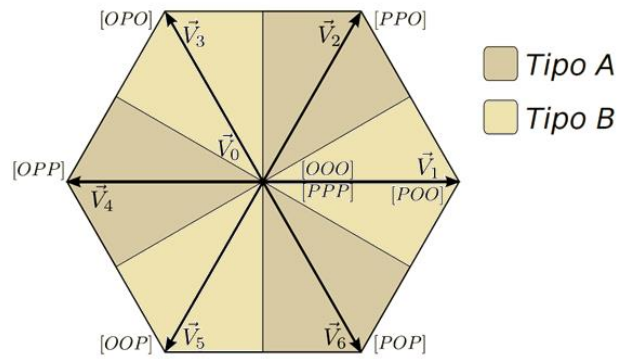


Figura 2.26: Hexágono de vectores estacionarios en función de secuencia tipo A y tipo B.

La principal desventaja de esta técnica para la eliminación de armónicos pares es el aumento del número de conmutaciones y en consecuencia de las pérdidas por este motivo.

En la Figura 2.2 se puede corroborar que la señal de salida resultante de esta técnica de modulación tiene la simetría par necesaria, además de la eliminación de los armónicos pares de esta señal. Estas señales, al igual que las mostradas en la Figura 2.24, han sido calculadas con unos valores de  $f_1 = 60\text{Hz}$ ,  $f_s = 720\text{Hz}$  y  $m_a = 0.8$ .

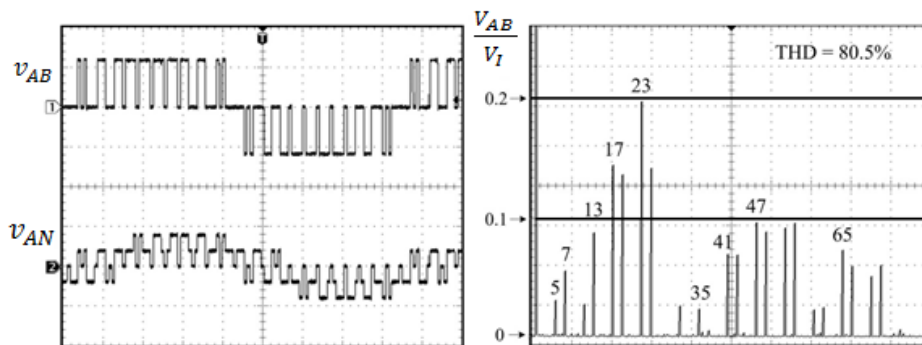


Figura 2.27: a) Tensión de salida en modulación SVM con eliminación de armónicos pares, b) Armónicos de la tensión de salida [8].

## Capítulo 3

# Herramienta docente para la enseñanza de convertidores DC/AC

### 3.1 Introducción

La herramienta desarrollada en este Trabajo Fin de Grado tiene como objetivo ayudar al alumno en el estudio de los inversores de potencia DC/AC estudiados en clase de Electrónica de Potencia.

Esta herramienta está compuesta por una interfaz gráfica desarrollada mediante la aplicación GUIDE [3] que será la encargada de poder de configurar los distintos parámetros a analizar, así como de seleccionar las diferentes opciones permitidas por la herramienta.

La interfaz gráfica enlaza con otra aplicación llamada Simulink [2], que permite modelar y simular los modelos de los inversores y modulaciones considerados.

Ambas herramientas pertenecen al software Matlab.

### 3.2 Interfaz gráfica de convertidores monofásicos

Como se ha explicado con anterioridad, la interfaz gráfica es la encargada de realizar las diferentes acciones del proyecto, tales como modificar los valores, simular circuitos o visualizar diferentes señales.

Tras ejecutar el *script* principal se abre la pantalla de inicio mostrada en la Figura 3.1.



Figura 3.1: Pantalla de inicio.

Como se puede ver, esta pantalla contiene cuatro botones diferentes para cada una de las opciones:

- Convertidor DC/DC

Este botón se deja preparado para posteriormente ejecutar los modelos de convertidores DC/DC realizados en el proyecto anterior a este [4].

- Convertidor DC/DC Puente en H

Al igual que el botón anterior, este botón se deja preparado para implementar la interfaz gráfica asociada a su correspondiente convertidor DC/DC.

- Inversor DC/AC Monofásico

Este es el primer botón operativo de la interfaz. Con él se abre la interfaz gráfica asociada a los inversores de potencia monofásicos con sus diferentes variantes, es decir, el convertidor en puente completo con modulación PWM bipolar, con modulación PWM unipolar y con modulación mediante cancelación de tensión, además del convertidor Push Pull. Este botón será estudiado en el apartado 3.2.

- Inversor DC/AC Trifásico

Este botón permite abrir la interfaz gráfica asociada a los inversores de potencia trifásicos y sus diferentes variantes, es decir, al convertidor de tres ramas con modulación PWM, con modulación THIPWM, con modulación CB-SVPWM, con modulación DPWM y con modulación vectorial, tanto en su forma normal como con eliminación de armónicos pares. Este botón será estudiado en el apartado 3.3.

Al pulsar el botón **“Inversor DC/AC Monofásico”** se abre la ventana mostrada en la Figura 3.2 con la interfaz gráfica asociada a estos inversores.

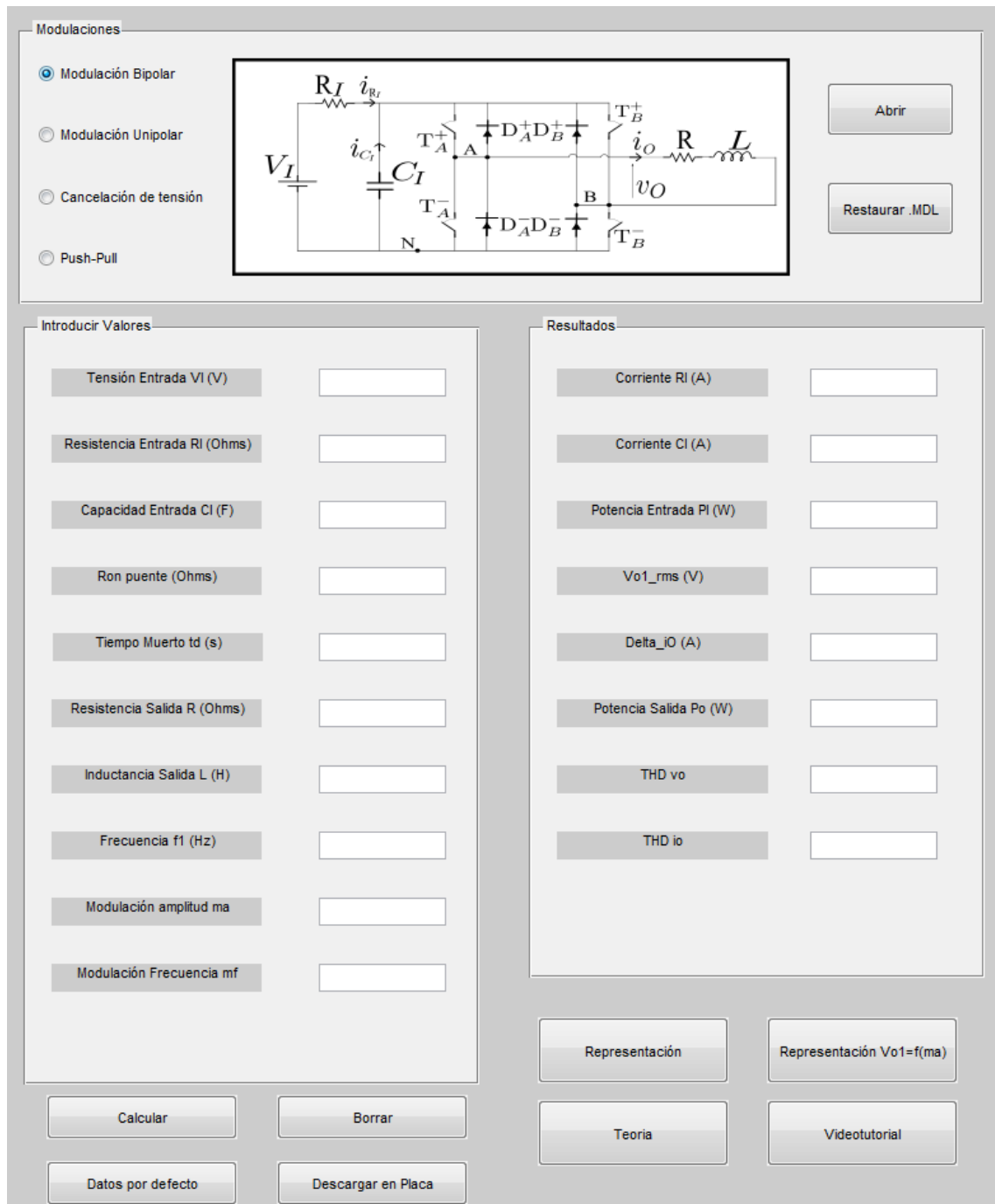


Figura 3.2: Pantalla principal de inversores monofásicos.

En primer lugar está la pestaña **“Modulación”**. En esta pestaña se permite seleccionar el tipo de modulación que se quiere analizar.

Una vez en esta pestaña, se analizará cada modulación por separado.

### 3.2.1 Modulación PWM Bipolar

Siguiendo en la pestaña “**Modulación**” se puede observar el circuito inversor en puente completo de la Figura 3.3 en el cual se implementa la modulación.

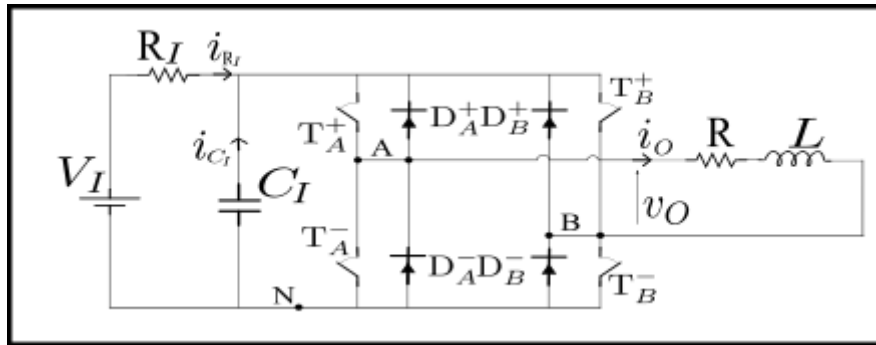


Figura 3.3: Esquema del puente completo.

Pulsando el botón “**Abrir**”, se abrirá el archivo .mdl que modela este tipo de modulación.

El archivo .mdl correspondiente a la modulación PWM bipolar es el mostrado en la Figura 3.4.

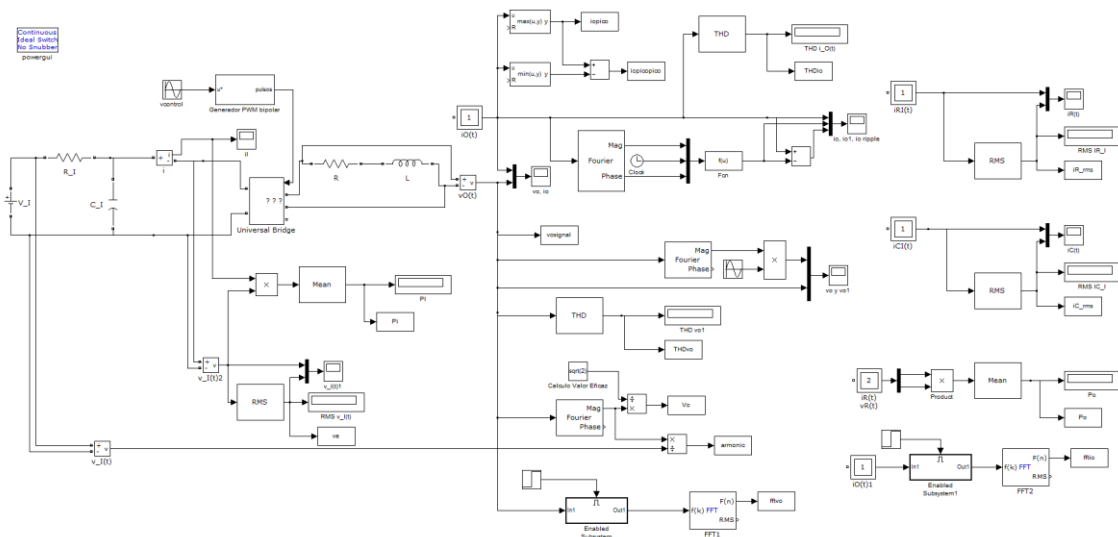


Figura 3.4: Modelo Simulink en puente completo monofásico en modulación bipolar.

A continuación se explican los bloques más importantes empleados en los modelos Simulink:

**Universal Bridge:** Con este bloque se ha implementado el puente completo en los inversores monofásicos y el puente de tres ramas en los inversores trifásicos. En ambos casos el puente está formado por Mosfet's y diodos aunque permite su implementación con otro tipo de transistores.

**FFT:** Este bloque se ha empleado para calcular los armónicos de la tensión y de la corriente de salida. En ambos casos son activados tras el transitorio de la señal para evitar el ruido generado por este.

**THD:** Con este bloque se ha calculado el THD (del inglés *Total Harmonic Distortion*) de la tensión y de la corriente de salida.

**RMS:** Este bloque se utiliza para calcular el valor eficaz de diferentes magnitudes como la tensión de salida o la corriente de entrada.

**Fourier:** Este bloque se ha utilizado para el cálculo del primer armónico de diferentes señales, ya que descompone la señal de entrada en la amplitud y la fase del armónico seleccionado.

**Relay:** Este bloque es un comparador por histéresis que se emplea para comparar las señales  $v_{control}$  y  $v_{tri}$ .

**On/Off Delay:** Este bloque sirve para introducir los tiempos muertos  $t_d$  en la simulación.

**Max/Min:** Con este bloque, configurable para calcular el máximo o el mínimo de una señal, se ha calculado el rizado de la corriente de salida.

**Multi-Winding Transformer:** Este bloque únicamente se utiliza en el convertidor Push Pull. Representa un transformador con dos devanados de entrada y uno de salida, gracias al cual podemos ajustar la relación de espiras N.

La diferencia con el resto de modulaciones implementadas en el puente completo está en el bloque “**Generador PWM bipolar**”, el cual se puede ver en la Figura 3.5. En esta técnica de modulación se calcula la diferencia entre las señales  $v_{control}$  con  $v_{tri}$ . Después con el comparador con histéresis se definen los estados de “on” y “off”. En esta técnica de modulación,  $T_A^+$  y  $T_B^-$  por un lado, y  $T_A^-$  y  $T_B^+$  por otro, están en el mismo estado.

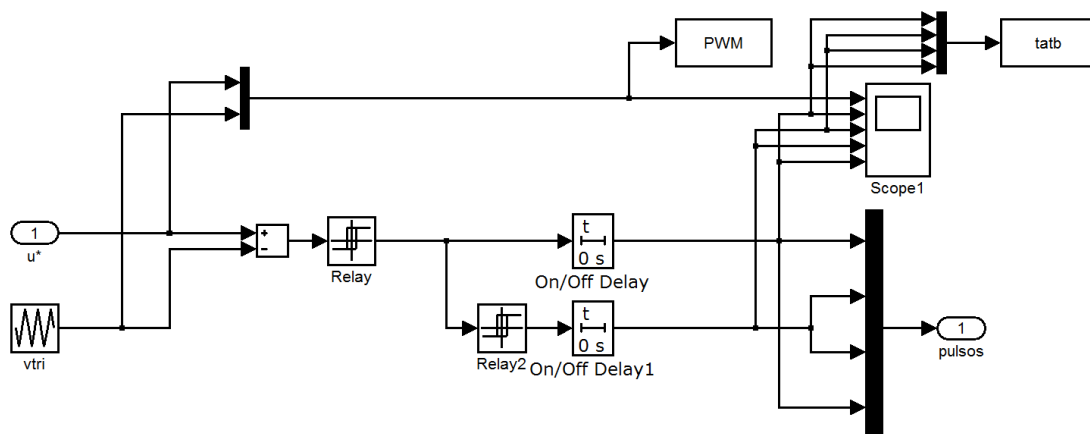


Figura 3.5: Circuito Simulink de modulación PWM bipolar monofásico.

Sobre estos archivos se pueden realizar y guardar las modificaciones que desee el usuario para estudiar otros comportamientos que no hayan sido considerados en este Trabajo Fin de Grado.

Pulsando el botón “**Restaurar .mdl**” se restaura el modelo a ejecutar por una copia guardada del original y se eliminan los posibles cambios que se hayan guardado en el modelo.

La siguiente pestaña es “**Introducir Valores**” en la cual se introducen los valores que serán asignados al modelo Simulink.

Los valores que se pueden introducir no serán los mismos en todos los modelos aunque se dan ciertas variables comunes a todos ellos:

- Tensión Entrada  $V_I$  (V)
- Resistencia Entrada  $R_I$  (Ohmios)
- Capacidad Entrada  $C_I$  (F)
- $R_{on}$  puente (Ohmios)
- Tiempo Muerto  $t_d$  (s)
- Resistencia Salida  $R$  (Ohmios)
- Inductancia Salida  $L$  (H)

Además habrá otras variables específicas de cada modulación, siendo para el caso de modulación PWM bipolar:

- Frecuencia  $f_1$  (Hz)
- Modulación Amplitud  $m_a$
- Modulación Frecuencia  $m_f$

La otra pestaña perteneciente a la interfaz es “**Resultado**”. En esta pestaña se obtienen los resultados calculados en la simulación. Para las diferentes técnicas de modulación los resultados son los siguientes:

- Corriente  $R_I$  (A)
- Corriente  $C_I$  (A)
- Potencia Entrada  $P_I$  (W)
- $V_{01\_rms}$  (V)
- $\Delta i_o$  (A)
- Potencia Salida  $P_o$  (W)
- $THD_{v_o}$
- $THD_{i_o}$

En la parte inferior de la interfaz hay una serie de botones con diversas funcionalidades que serán explicados a continuación.



El botón “**Calcular**” ejecuta el modelo y tras el tiempo de simulación aparecen los resultados. Esta funcionalidad se puede ver en la Figura 3.6.

The interface is divided into several sections:

- Modulaciones:** Contains radio buttons for 'Modulación Bipolar' (selected), 'Modulación Unipolar', 'Cancelación de tensión', and 'Push-Pull'. To the right are 'Abrir' and 'Restaurar .MDL' buttons.
- Introducir Valores:** A table of input parameters with text boxes for values:
 

Tensión Entrada $V_I$ (V)	100
Resistencia Entrada $R_I$ (Ohms)	0.1
Capacidad Entrada $C_I$ (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto $t_d$ (s)	0
Resistencia Salida $R$ (Ohms)	10
Inductancia Salida $L$ (H)	0.01
Frecuencia $f_1$ (Hz)	50
Modulación amplitud $m_a$	0.8
Modulación Frecuencia $m_f$	15
- Resultados:** A table of output results:
 

Corriente $R_I$ (A)	4.75271
Corriente $C_I$ (A)	2.82674
Potencia Entrada $P_I$ (W)	305.696
$V_{o1\_rms}$ (V)	56.1827
$\Delta_{iO}$ (A)	17.3882
Potencia Salida $P_o$ (W)	305.633
THD $v_o$	1.46569
THD $i_o$	0.252642
- Buttons:** 'Calcular' (highlighted), 'Borrar', 'Representación', 'Representación  $V_{o1}=f(m_a)$ ', 'Teoria', 'Videotutorial', 'Datos por defecto', and 'Descargar en Placa'.

Figura 3.6: Cálculos con modulación PWM bipolar y datos por defecto.

El siguiente botón en pantalla es el botón “**Borrar**”, encargado de suprimir todos los valores de la interfaz, tanto de la pestaña “**Introducir Valores**” como de la pestaña “**Resultados**”.

El botón “**Datos por defecto**” genera unos valores predeterminados para realizar la simulación, los cuales podrán servir como ayuda para comprobar el correcto funcionamiento del convertidor.

El botón “**Descargar en Placa**” se ha dejado preparado para una posible futura ampliación de la herramienta. Esta ampliación consistiría en poder descargar las señales del modelo Simulink en una tarjeta FPGA para poder trabajar con un circuito inversor real, tal y como se realizó en el proyecto anterior a este [4]. Actualmente muestra el mensaje de advertencia de la Figura 3.7.

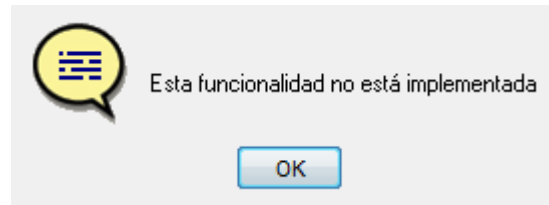


Figura 3.7: Mensaje de advertencia.

El botón “**Teoría**” abre los apuntes de la asignatura, lo cual sirve para complementar la herramienta y ayudar al alumno a buscar respuesta a diferentes problemas que puedan surgir. La portada de estos apuntes puede verse en la Figura 3.8.



Figura 3.8: Transparencias sobre inversores monofásicos de la asignatura.

El botón “**Videotutorial**” abre un video explicativo con el funcionamiento de la herramienta que sirve como manual de uso de la misma.

El botón “**Representación**” abrirá la ventana de la Figura 3.9.

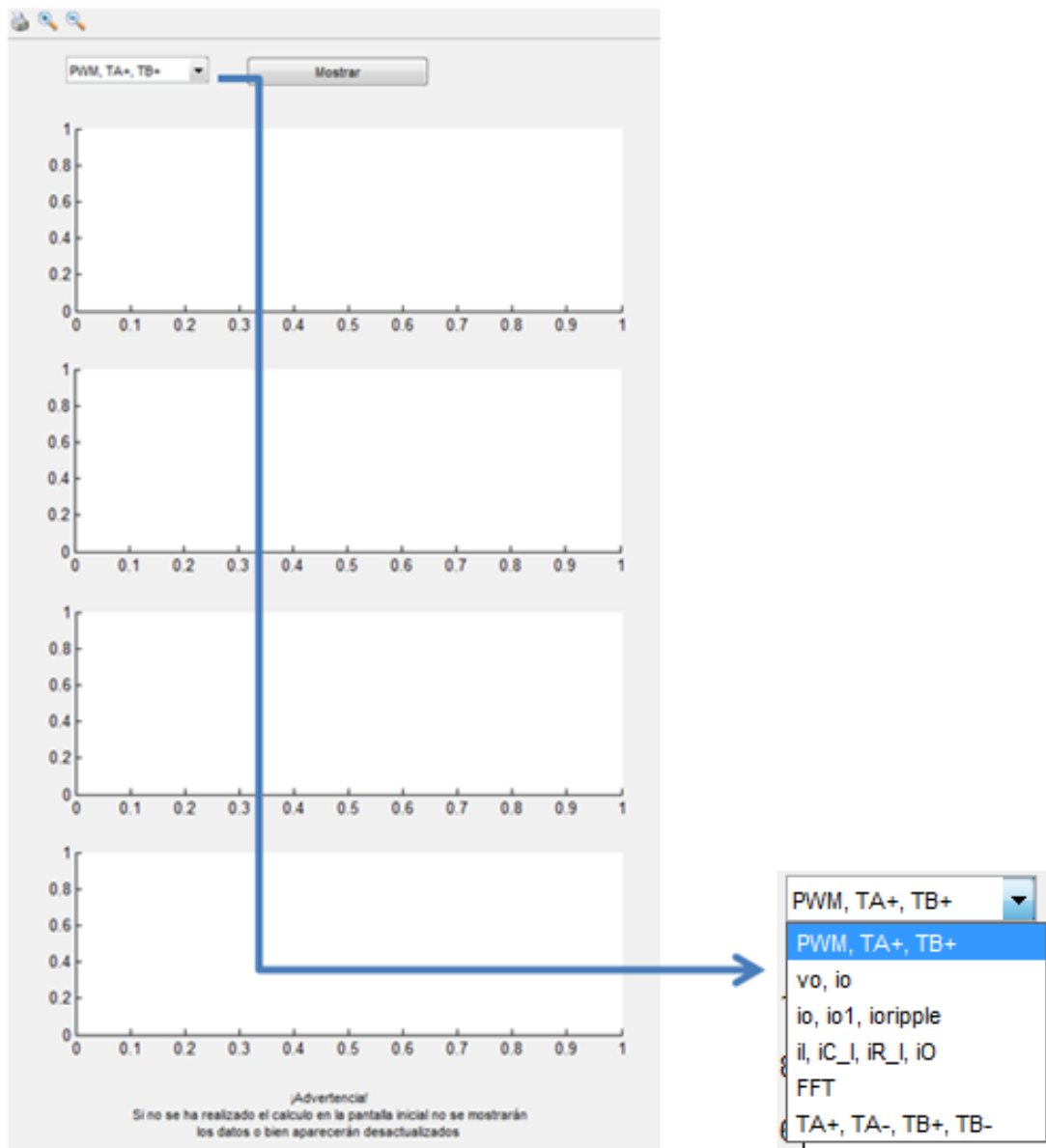


Figura 3.9: Pantalla de representación de formas de onda con PWM.

En esta ventana hay un desplegable que permite seleccionar entre seis diferentes opciones de visualización, las cuales aparecerán en las gráficas tras pulsar el botón “**Mostrar**”.

Además, en la parte superior izquierda de la ventana se encuentra la herramienta de ampliación. Esta herramienta permite ampliar la sección deseada de una gráfica mientras que el resto se reajusta al mismo nivel de aumento.

Al seleccionar “**PWM, TA+, TB+**” se obtiene las señales de modulación PWM así como el estado de los transistores superiores del puente, tal y como se observa en la Figura 3.10.

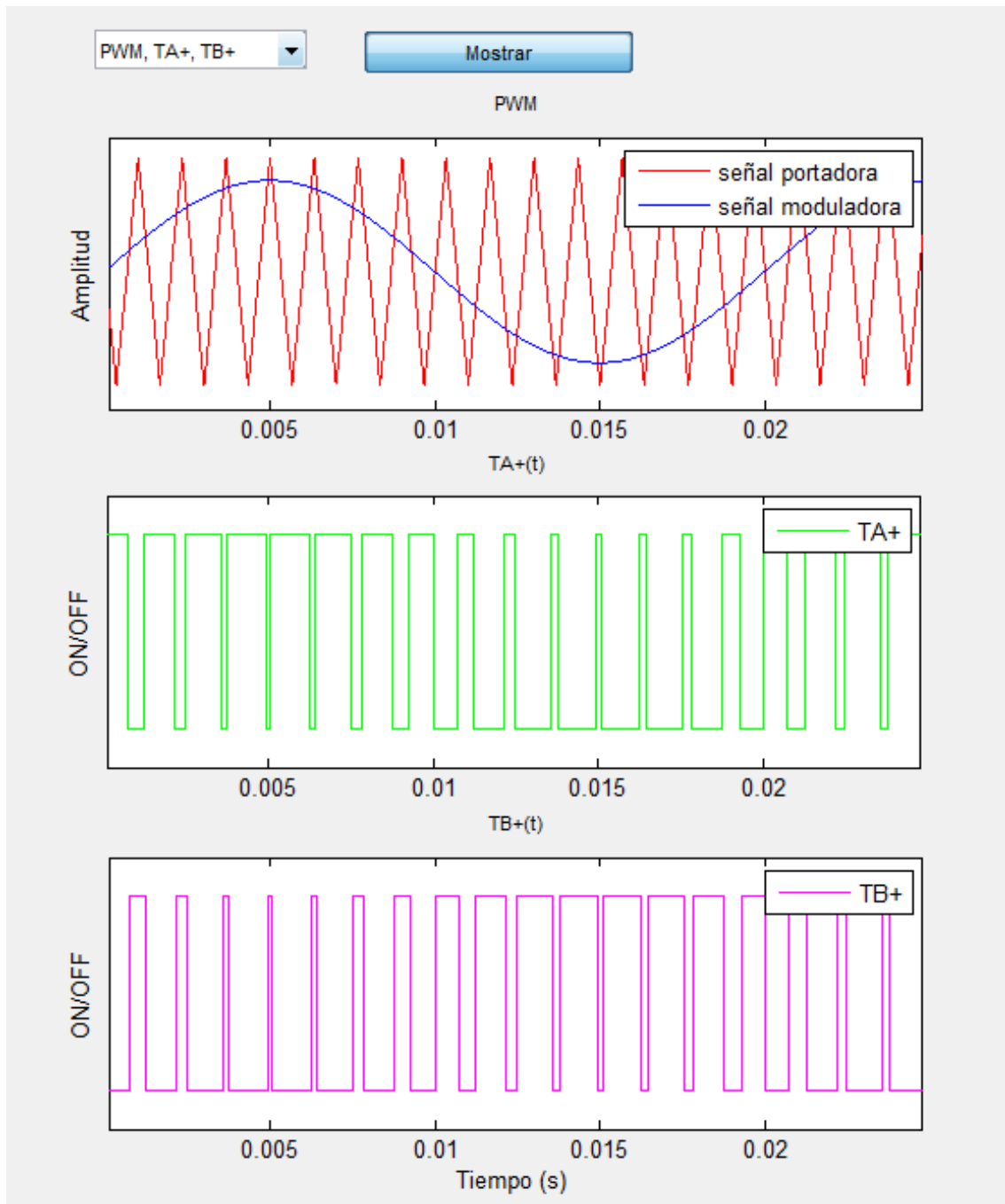


Figura 3.10: Señales PWM, TA+ y TB+ con modulación PWM bipolar.

Al pulsar sobre “ **$v_o$ ,  $i_o$** ” se mostrará la tensión y la corriente de salida con sus respectivos primeros armónicos, correspondiente a la Figura 3.11.

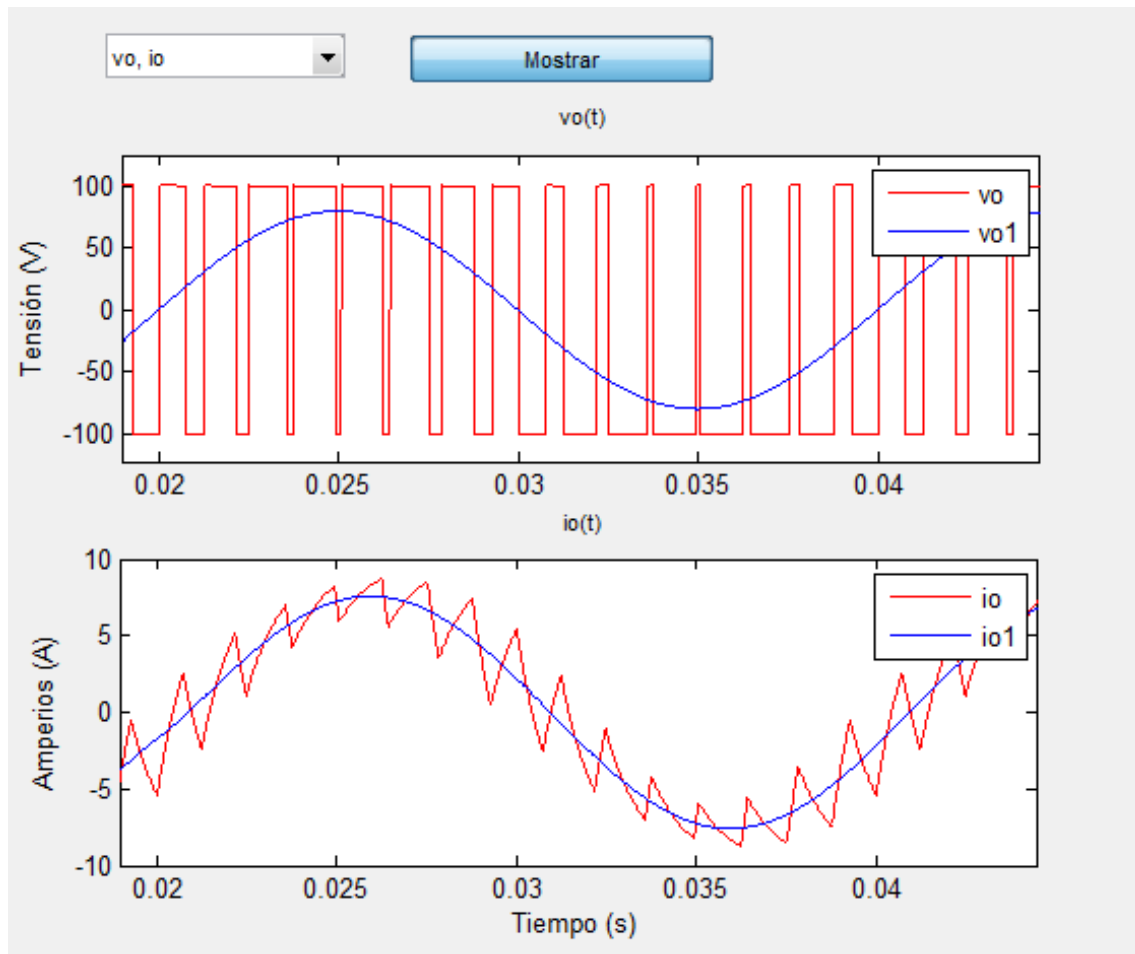


Figura 3.11: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM bipolar.

Seleccionando “**io**, **io1**, **ioripple**” puede verse la corriente de salida en la primera ventana, su primer armónico en la segunda y el rizado en la tercera ventana. Ver Figura 3.12.

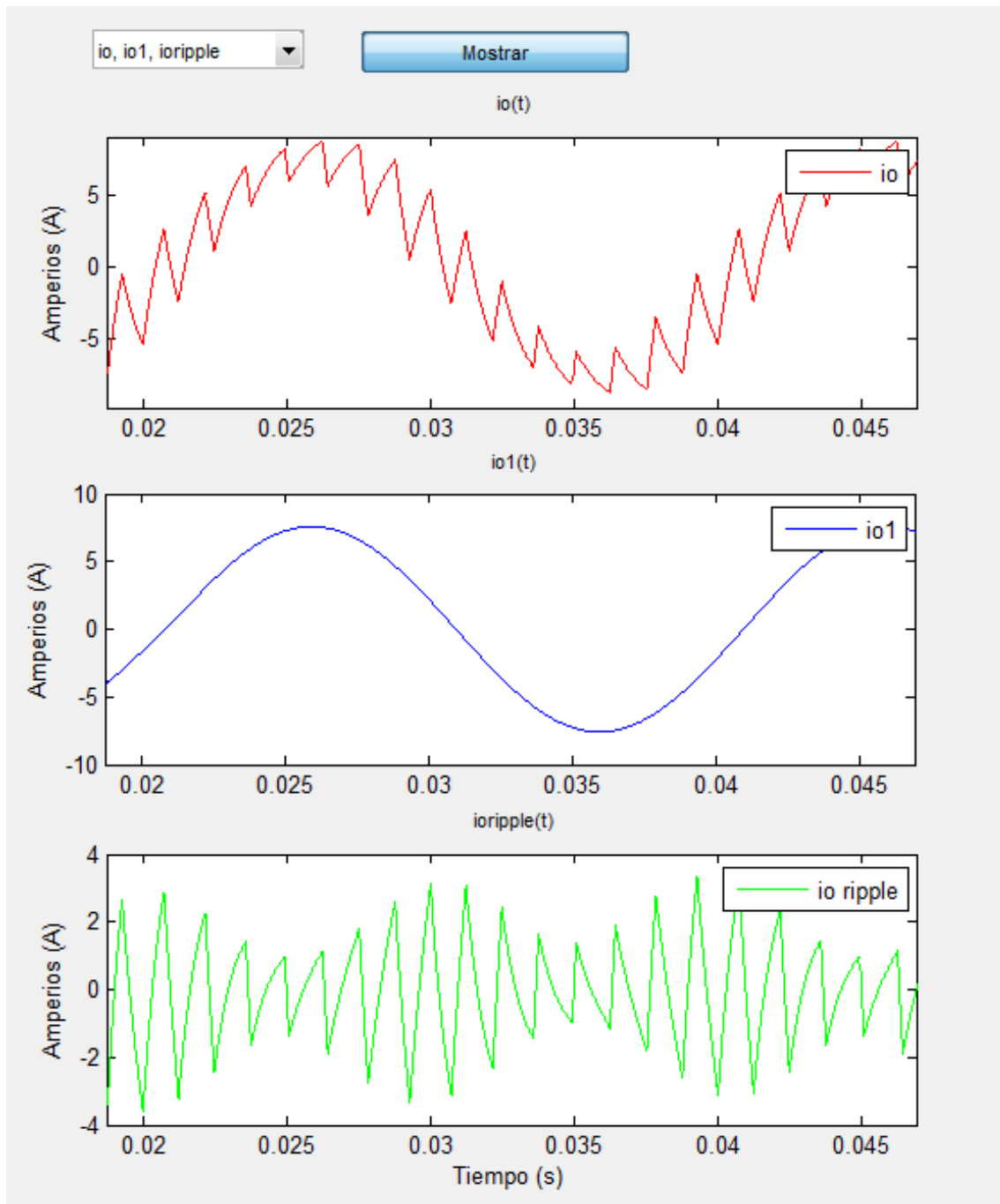


Figura 3.12: Corriente de salida ( $i_o$ ) en modulación PWM bipolar.

Al seleccionar “**iI, iC\_I, iR\_I, iO**” se representa la evolución temporal de las diferentes corrientes del circuito, las cuales se ilustran en Figura 3.13:

- La corriente a la entrada
- La corriente por el condensador de entrada
- La corriente por la resistencia de entrada
- La corriente de salida

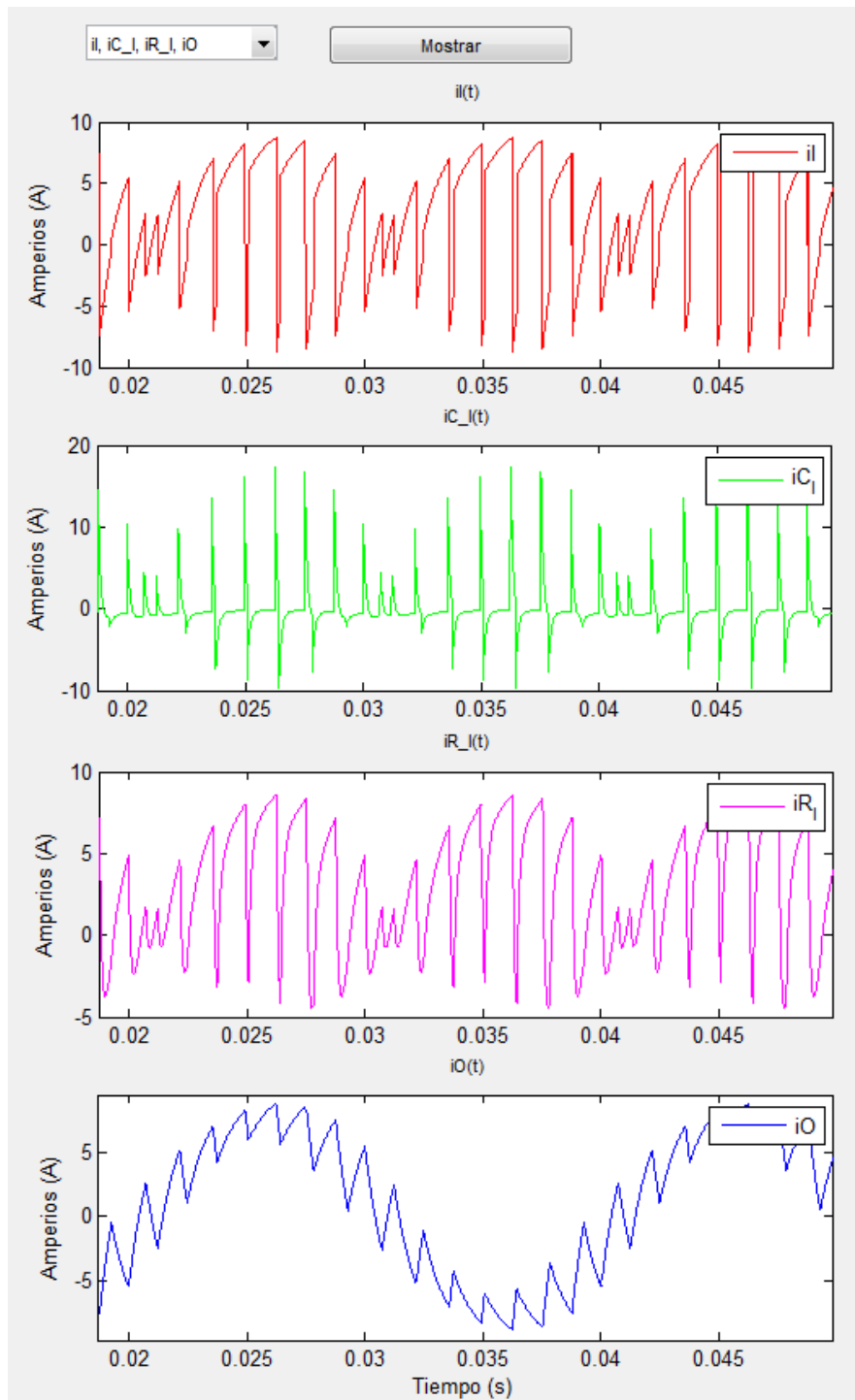


Figura 3.13: Corrientes del circuito en modulación PWM bipolar.

La siguiente opción que se puede seleccionar es “**FFT**”. En esta pestaña se representa el contenido armónico de la tensión de salida  $v_o$  y de la corriente de salida  $i_o$ . Estas gráficas se pueden visualizar en la Figura 3.14. Esta pestaña resulta interesante para poder analizar la calidad de las señales en función de su modulación.

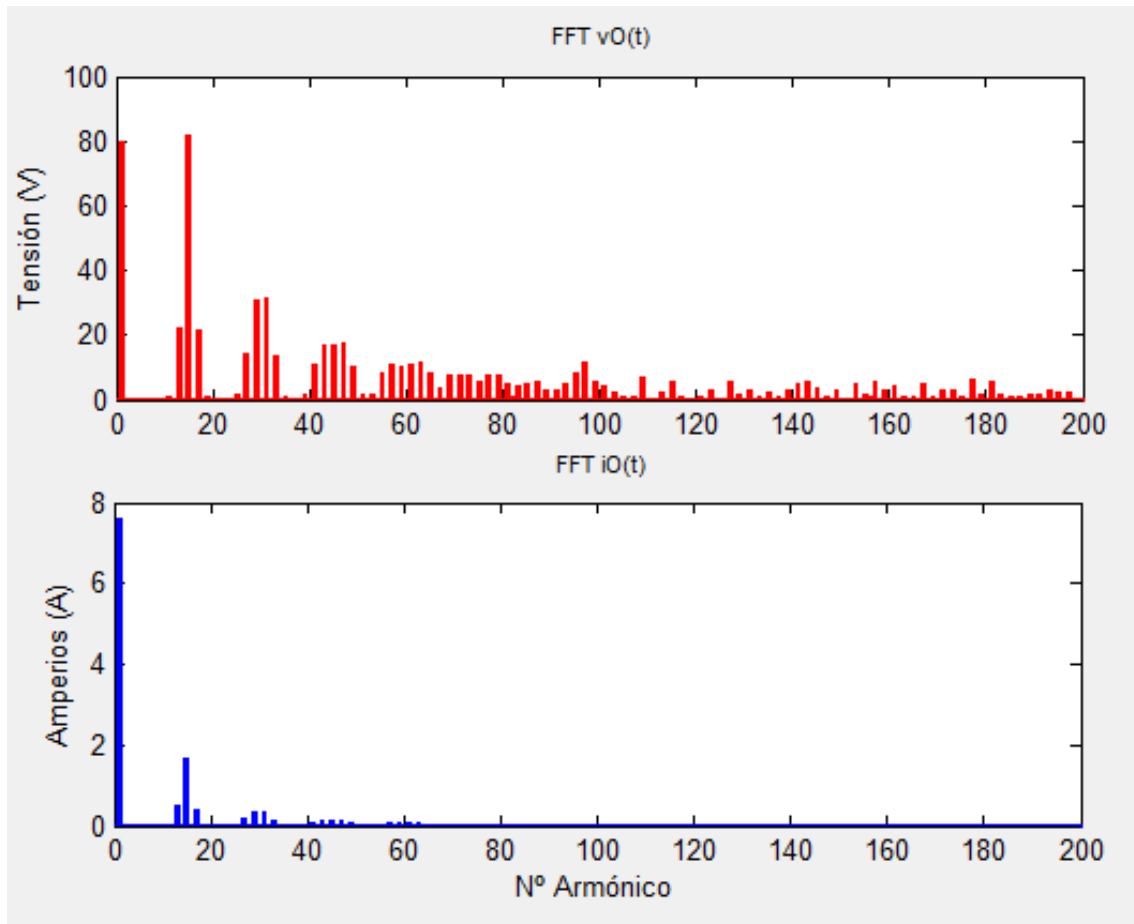


Figura 3.14: Armónicos de tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM bipolar.



La última opción disponible para seleccionar es “TA+, TA-, TB+, TB-“. En esta opción se muestra el estado de los cuatro transistores del circuito, como puede verse en la Figura 3.15.

Esta opción es especialmente interesante para observar el comportamiento del inversor frente a los tiempos muertos.

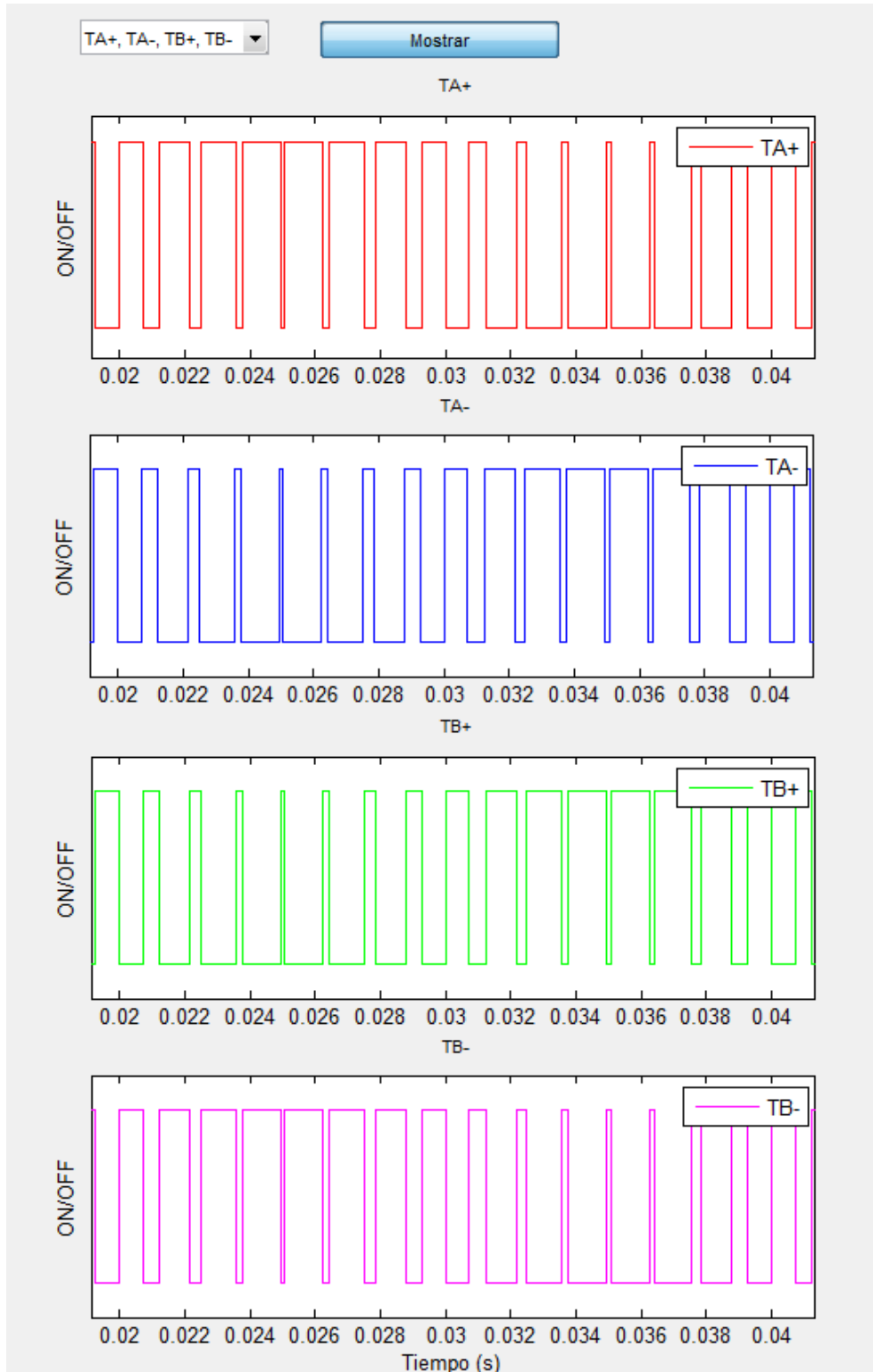


Figura 3.15: Ciclos de los transistores en modulación PWM bipolar.

Cuando se está trabajando en esta pestaña de representación, es imprescindible haber realizado la simulación previamente, ya que si se modifica la técnica de modulación o los parámetros de entrada y no se pulsa el botón “**Calcular**” las gráficas que se representarán serán las correspondientes al último cálculo realizado. A modo de recordatorio se muestra en la ventana de representación el mensaje de la Figura 3.16.

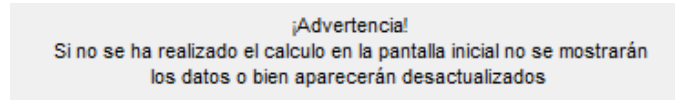


Figura 3.16: Mensaje de advertencia sobre la necesidad de cálculo.

Una vez analizadas todas las opciones que se pueden visualizar en la ventana de representación se vuelve a la interfaz gráfica monofásica.

El último botón por analizar en esta ventana es el botón “**Representación Vo1=f(ma)**”, el cual, al ser pulsado, abrirá la ventana de la Figura 3.17.

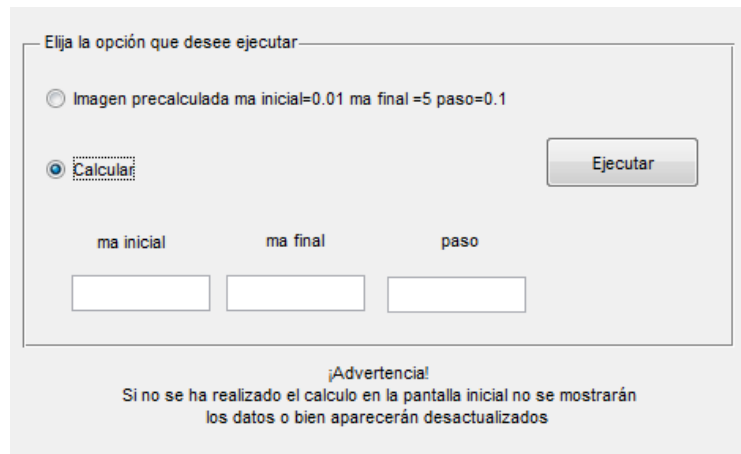
Una interfaz de usuario con el título "Elija la opción que desee ejecutar". Hay dos opciones de radio: "Imagen precalculada ma inicial=0.01 ma final =5 paso=0.1" (desseleccionada) y "Calcular" (seleccionada). A la derecha hay un botón "Ejecutar". Debajo de "Calcular" hay tres campos de entrada etiquetados "ma inicial", "ma final" y "paso". En la parte inferior del recuadro principal hay un mensaje de advertencia idéntico al de la Figura 3.16.

Figura 3.17: Pantalla de barrido.

Dentro de esta ventana existen dos opciones diferentes. La primera de ellas consiste en una imagen precalculada como la de la Figura 3.18 con unos parámetros ya definidos, debido a que este cálculo puede tardar cierto tiempo. La segunda opción consiste en realizar el cálculo con los valores de  $m_a$  inicial,  $m_a$  final y el paso deseado.

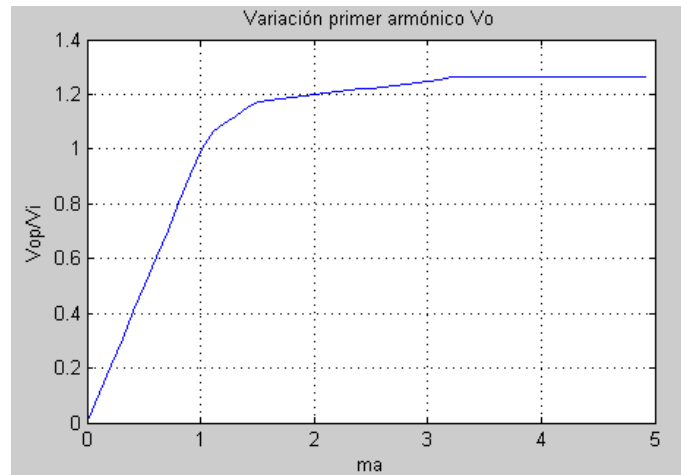


Figura 3.18: Imagen precalculada del barrido en modulación bipolar.

Al igual que ocurría en la ventana de representación, el usuario debe haber realizado el cálculo previamente en la pantalla principal, ya que en caso de no haberlo hecho se ejecutará con los parámetros y modulación seleccionados con anterioridad.

### 3.2.2 Modulación PWM Unipolar

En la modulación PWM unipolar y en las siguientes (modulación mediante cancelación de tensión e inversor Push Pull), únicamente se explicarán aquellos aspectos que sean diferentes respecto a la modulación PWM bipolar.

Al seleccionar la opción de Modulación Unipolar no se producen cambios en la interfaz gráfica monofásica, debido a que el esquema del circuito y los parámetros de entrada se mantienen iguales al de la modulación unipolar.

Pulsando el botón “Abrir”, se abrirá el archivo .mdl en el cual está implementado esta modulación.

El archivo .mdl que se muestra en la modulación PWM unipolar es igual que en el caso de la modulación PWM bipolar, a excepción del bloque “Generador PWM unipolar”. Este circuito se observa en la Figura 3.19.

En esta técnica de modulación se calcula la diferencia entre las señales  $v_{control}$  con  $v_{tri}$  para los transistores de la rama A, y se niega la señal  $v_{control}$  para el cálculo de las señales de la rama B.

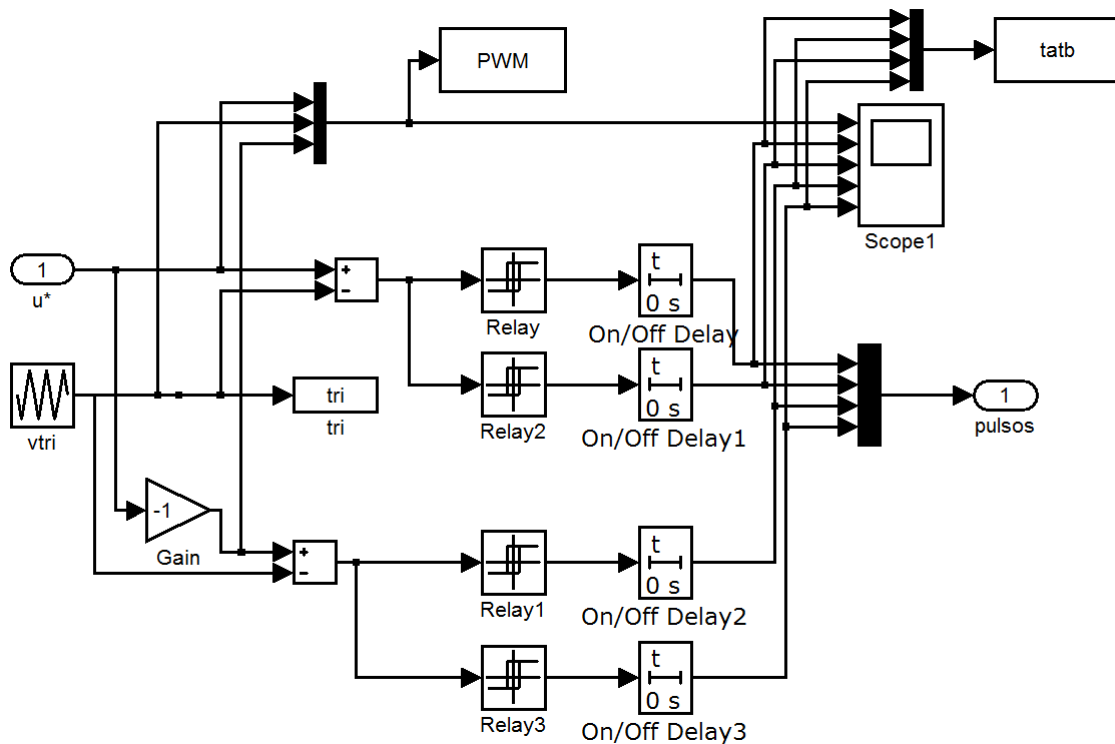


Figura 3.19: Circuito Simulink con modulación PWM unipolar.

El botón “**Datos por defecto**” genera unos valores predeterminados para realizar la simulación, los cuales son diferentes a los observados en la modulación bipolar. Los datos por defecto y los resultados obtenidos al calcular se visualizan en la Figura 3.20.

The interface is divided into several sections:

- Modulaciones:** Radio buttons for 'Modulación Bipolar', 'Modulación Unipolar' (selected), 'Cancelación de tensión', and 'Push-Pull'.
- Circuit Diagram:** A schematic of a unipolar PWM bridge with an input filter (resistor  $R_I$ , capacitor  $C_I$ ) and an output filter (resistor  $R$ , inductor  $L$ ). Transistors are labeled  $T_A^+$ ,  $T_A^-$ ,  $T_B^+$ ,  $T_B^-$  and diodes  $D_A^+$ ,  $D_B^+$ ,  $D_A^-$ ,  $D_B^-$ . Currents  $i_{R_I}$ ,  $i_{C_I}$ ,  $i_O$  and voltages  $V_I$ ,  $v_O$  are indicated.
- Introducir Valores:** Input parameters table.
 

Tensión Entrada $V_I$ (V)	100
Resistencia Entrada $R_I$ (Ohms)	0.1
Capacidad Entrada $C_I$ (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto $t_d$ (s)	0
Resistencia Salida $R$ (Ohms)	10
Inductancia Salida $L$ (H)	0.01
Frecuencia $f_1$ (Hz)	50
Modulación amplitud $m_a$	0.8
Modulación Frecuencia $m_f$	14
- Resultados:** Simulation results table.
 

Corriente $R_I$ (A)	3.93252
Corriente $C_I$ (A)	1.86347
Potencia Entrada $P_I$ (W)	289.953
$V_{o1\_rms}$ (V)	56.2766
$\Delta i_O$ (A)	16.3379
Potencia Salida $P_o$ (W)	289.897
THD $v_o$	0.765965
THD $i_o$	0.0754355
- Buttons:** 'Abrir', 'Restaurar .MDL', 'Calcular', 'Borrar', 'Teoría', 'Videotutorial', 'Datos por defecto' (highlighted), and 'Descargar en Placa'.

Figura 3.20: Cálculos con modulación PWM unipolar y datos por defecto.

El botón “**Representación**” abrirá nuevamente la ventana de la Figura 3.9.

Seleccionando “**PWM, TA+, TB+**” se obtienen las señales de modulación PWM unipolar así como el estado de los transistores superiores del puente, tal y como se visualiza en la Figura 3.21.

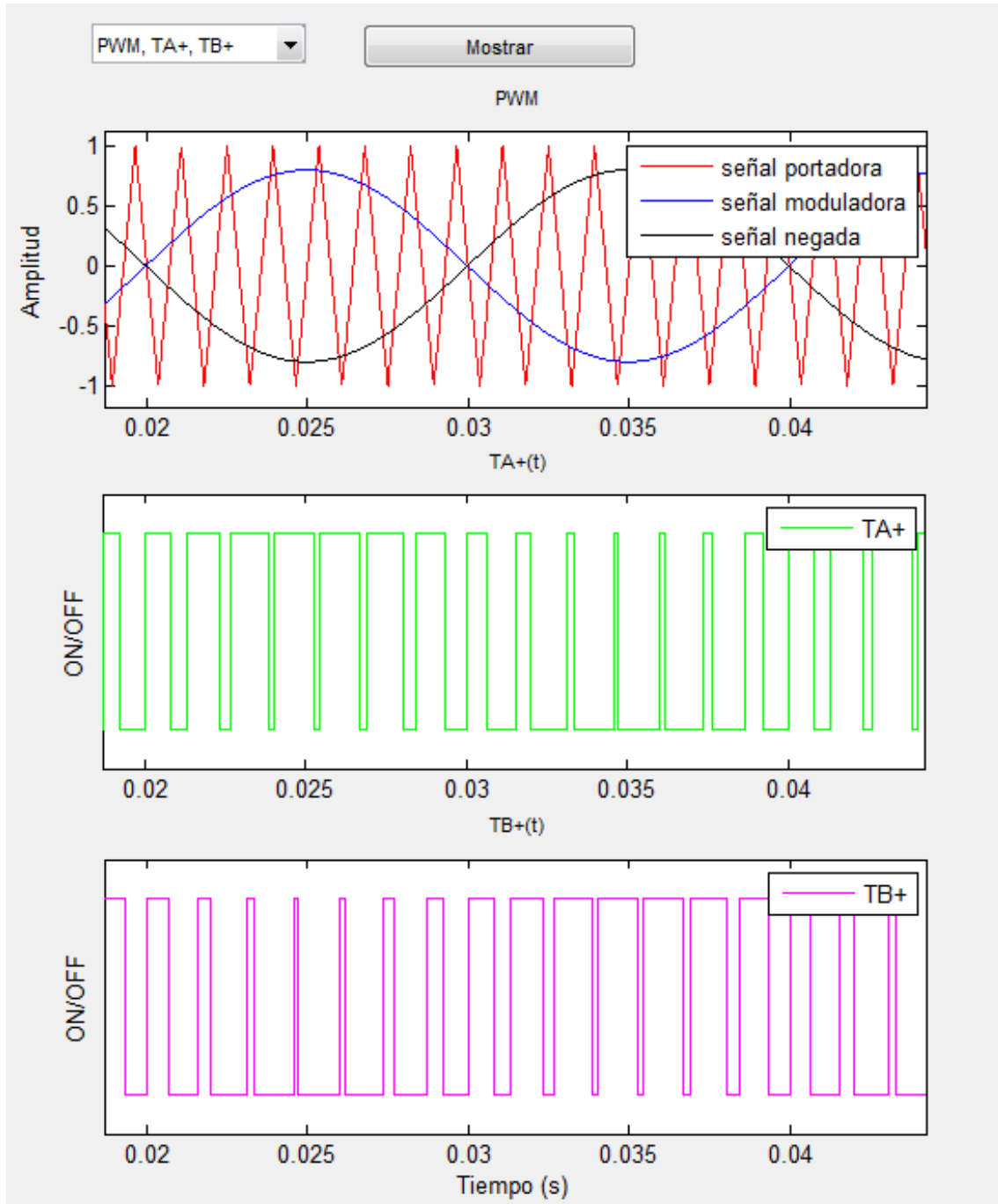


Figura 3.21: Señales PWM, TA+ y TB+ en modulación PWM unipolar.

Al seleccionar “**v<sub>o</sub>, i<sub>o</sub>**” se mostrara la tensión y la corriente de salida con sus respectivos primeros armónicos, correspondiente a la Figura 3.22

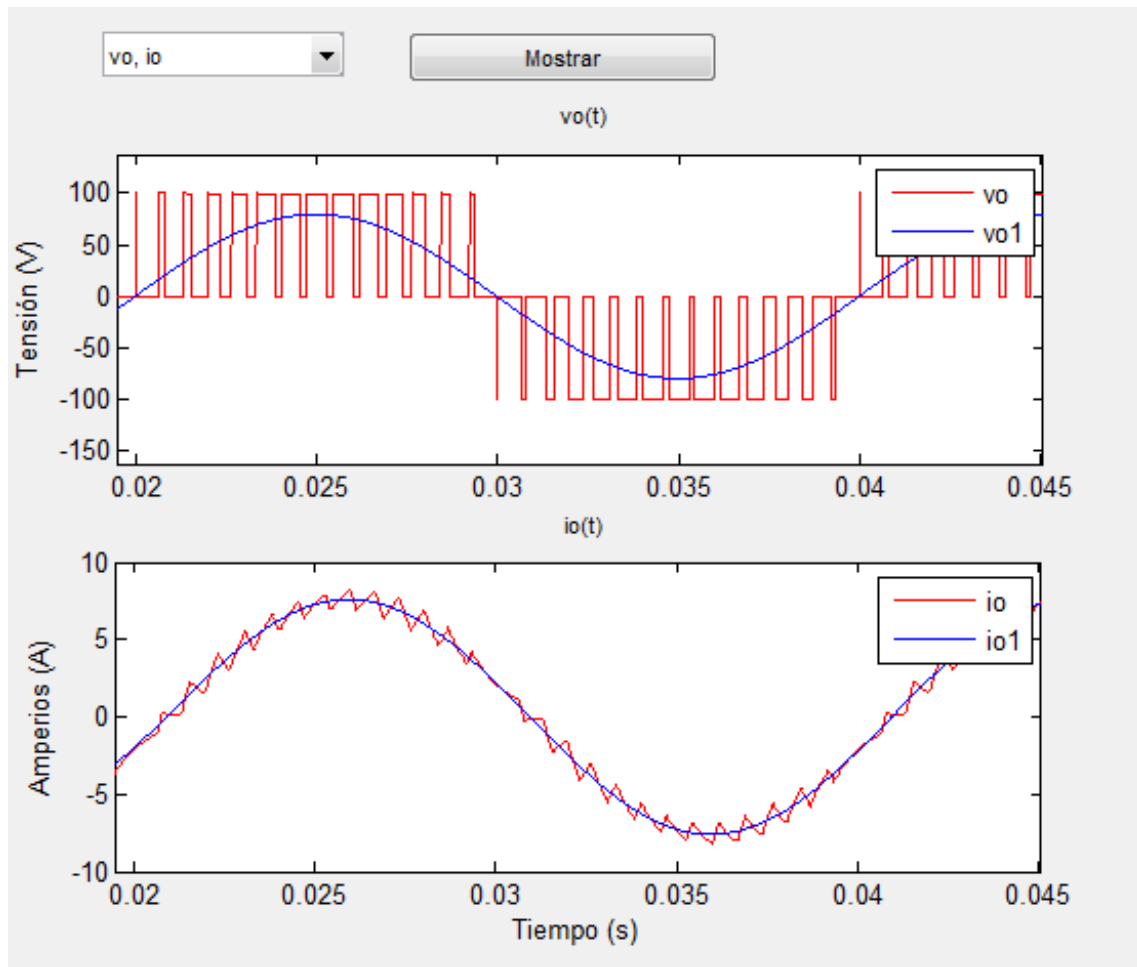


Figura 3.22: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM unipolar.

Al seleccionar sobre “**io, io1, ioripple**” se puede visualizar la corriente de salida en la primera ventana, su primer armónico en la segunda y el rizado en la tercera. Ver Figura 3.23.

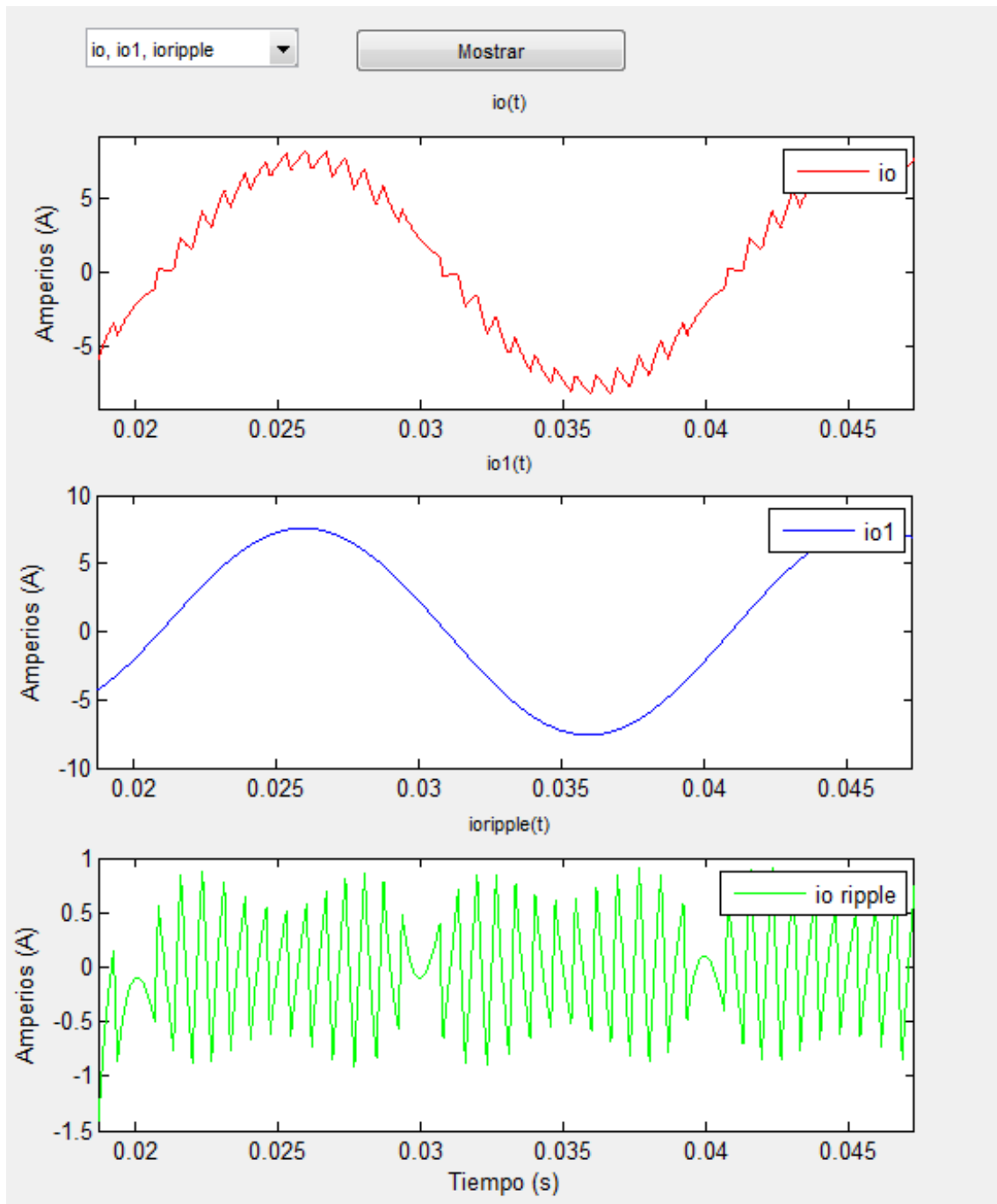


Figura 3.23: Corriente de salida ( $i_o$ ) en modulación PWM unipolar.



En la opción “**iI, iC\_I, iR\_I, iO**” se representa la evolución de las diferentes corrientes del circuito. Esta evolución se muestra en la Corrientes del circuito en modulación PWM unipolar. Figura 3.24.

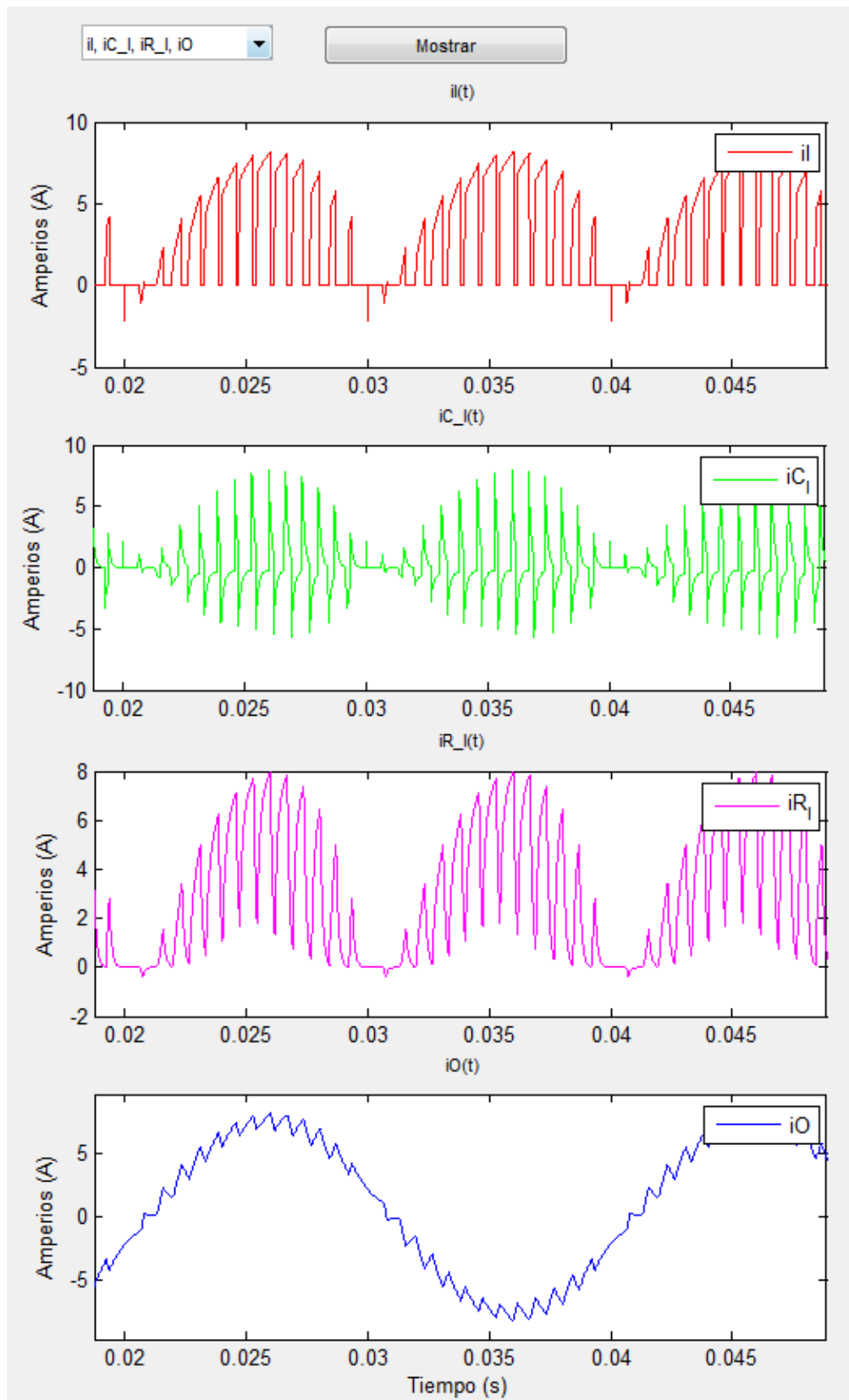


Figura 3.24: Corrientes del circuito en modulación PWM unipolar.

La siguiente opción que se permite seleccionar es “**FFT**”. En esta pestaña se muestra el contenido armónico de la tensión de salida  $v_o$  y de la corriente de salida  $i_o$ . Estas gráficas se pueden ver en la Figura 3.25.

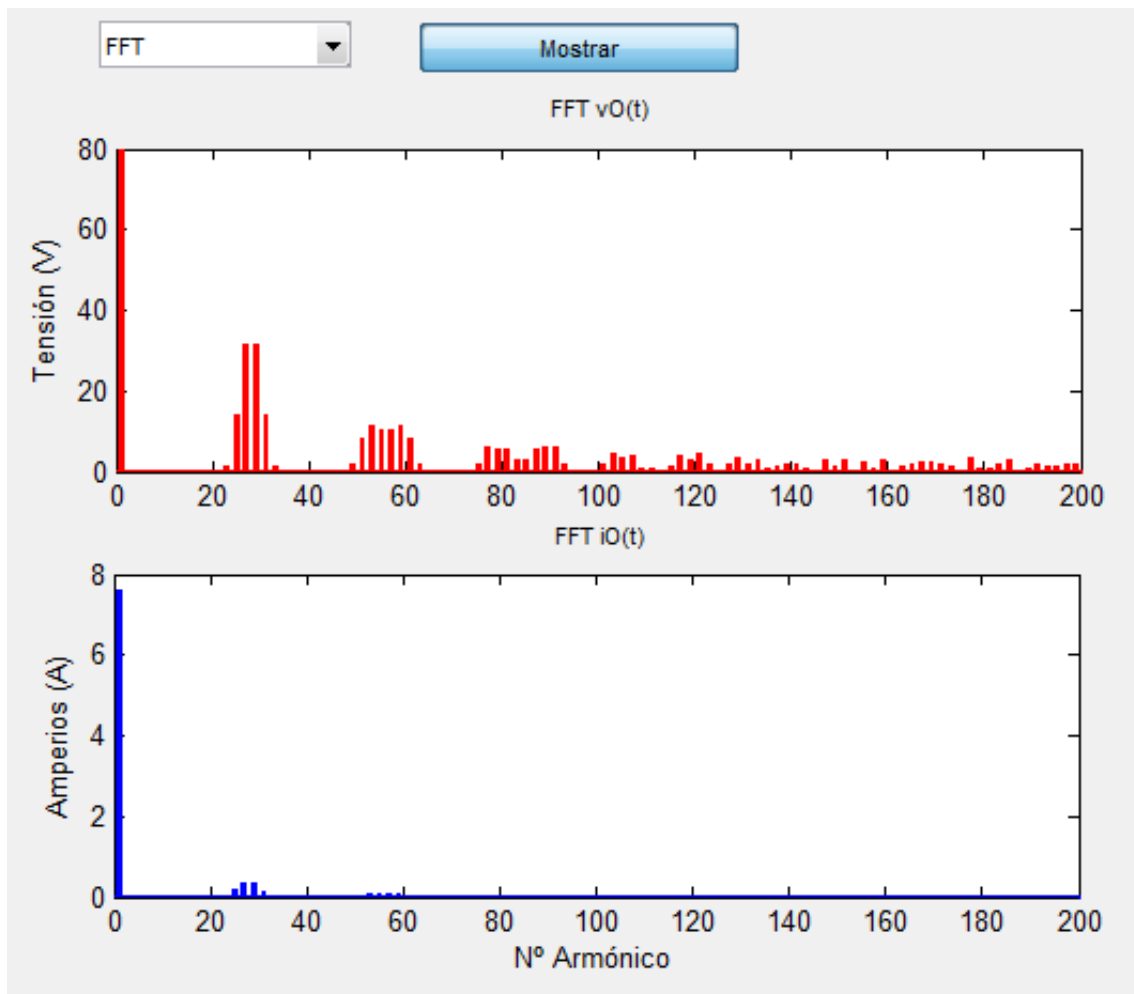


Figura 3.25: Armónicos de tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación PWM unipolar.

La última opción disponible para seleccionar es “TA+, TA-, TB+, TB-“. En esta opción se muestra el estado de los cuatro transistores del circuito. Este comportamiento se representa en la Figura 3.26

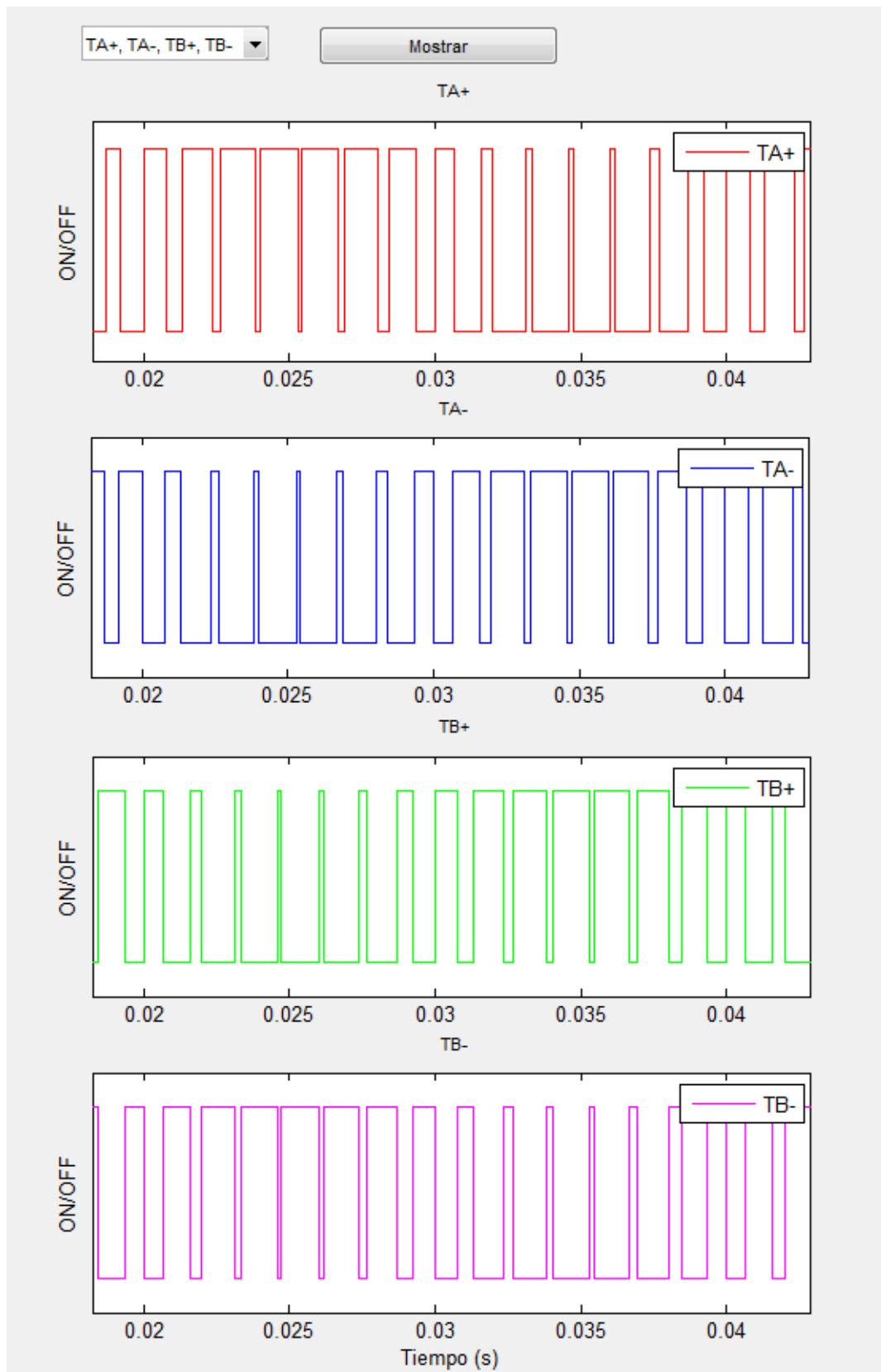


Figura 3.26: Ciclos de los transistores en modulación PWM unipolar.

La ventana asociada al botón “**Representación  $V_{o1}=f(m_a)$** ” de la interfaz gráfica abre la Figura 3.17. Nuevamente se permite la opción de realizar la simulación paramétrica o bien abrir la imagen precalculada, sin embargo la imagen precalculada guardada para la modulación PWM unipolar es diferente al caso bipolar. Esto corresponde a la imagen de la Figura 3.27.

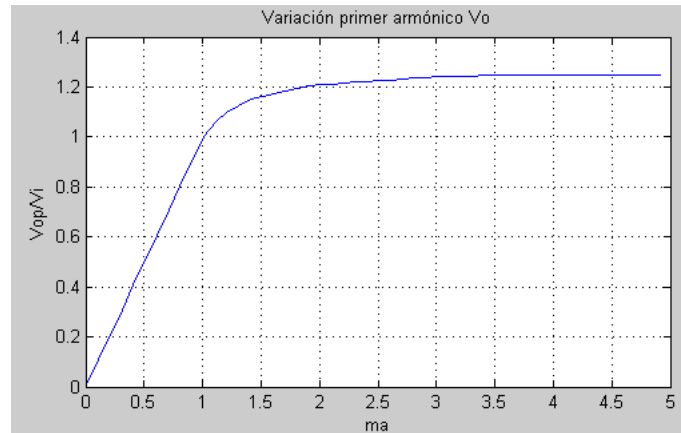


Figura 3.27: Imagen precalculada del barrido en modulación PWM unipolar.

### 3.2.3 Modulación mediante cancelación de tensión

Al seleccionar la opción de Cancelación de tensión, se producen cambios en la pestaña de “Introducir Valores”, ya que para esta modulación los parámetros de entrada no son iguales a los parámetros de las modulaciones PWM unipolar y bipolar.

Los nuevos parámetros son:

- Frecuencia Conmutación (Hz)
- Alpha (deg).

Al igual que ocurre en las modulaciones anteriores, al pulsar el botón “**Abrir**”, se abrirá el archivo .mdl en el cual está implementada esta modulación. Este modelo es igual al anterior a excepción del bloque “**Cancelación de tensión**” mostrado en la Figura 3.28. En esta técnica de modulación se trabaja con señales de onda cuadrada, dentro de las cuales se emplea el parámetro  $\alpha$ .

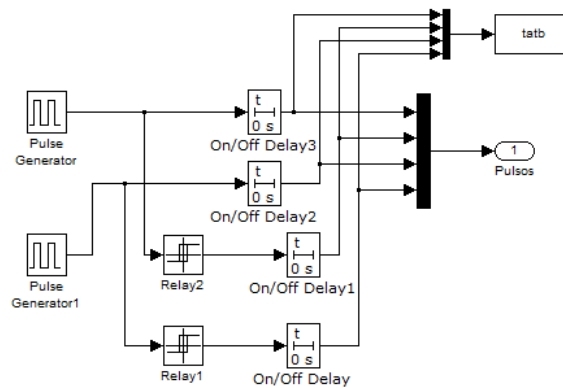


Figura 3.28: Circuito Simulink de modulación mediante cancelación de tensión.

El botón “**Datos por defecto**” genera unos valores predeterminados para realizar la simulación, los cuales son diferentes debido a los nuevos parámetros. Esta nueva situación se puede observar en la Figura 3.29.

**Modulaciones**

- Modulación Bipolar
- Modulación Unipolar
- Cancelación de tensión
- Push-Pull

**Introducir Valores**

Tensión Entrada $V_I$ (V)	100
Resistencia Entrada $R_I$ (Ohms)	0.1
Capacidad Entrada $C_I$ (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto $t_d$ (s)	0
Resistencia Salida $R$ (Ohms)	10
Inductancia Salida $L$ (H)	0.01
Frecuencia Conmutación (Hz)	50
Alpha (deg)	0

**Resultados**

Corriente $R_I$ (A)	8.76528
Corriente $C_I$ (A)	1.33545
Potencia Entrada $P_I$ (W)	786.254
$V_{o1\_rms}$ (V)	89.2109
Delta_ $i_O$ (A)	19.7969
Potencia Salida $P_o$ (W)	786.099
THD $v_o$	0.486061
THD $i_o$	0.291928

Botones: Abrir, Restaurar .MDL, Representación, Representación  $V_{o1}=f(\alpha)$ , Teoria, Videotutorial, Calcular, Borrar, Datos por defecto, Descargar en Placa

Figura 3.29: Cálculos de modulación mediante cancelación de tensión con datos por defecto.

El botón “**Representación**” abrirá la ventana de formas de onda diferente a los casos con modulación PWM, ya que en este caso no hay señal PWM que representar.

La nueva ventana con el nuevo desplegable se puede visualizar en la Figura 3.30.

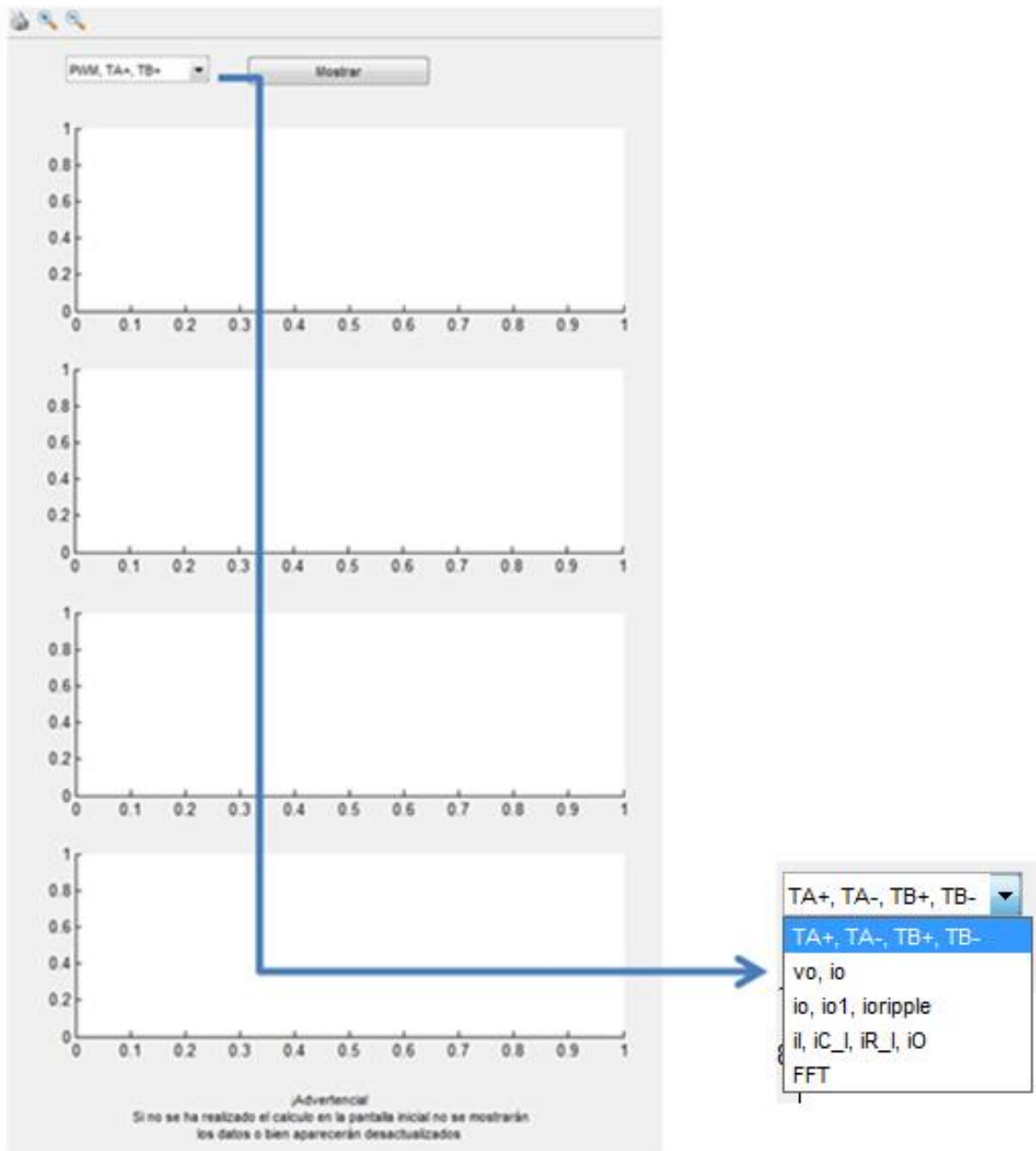


Figura 3.30: Pantalla de representación formas de onda sin PWM.

En la primera opción, “**TA+**, **TA-**, **TB+**, **TB-**”, se representa el estado de los cuatro transistores del circuito, tal y como se muestra en la Figura 3.31. Esta imagen se ha calculado para un valor de  $\alpha=45^\circ$ .

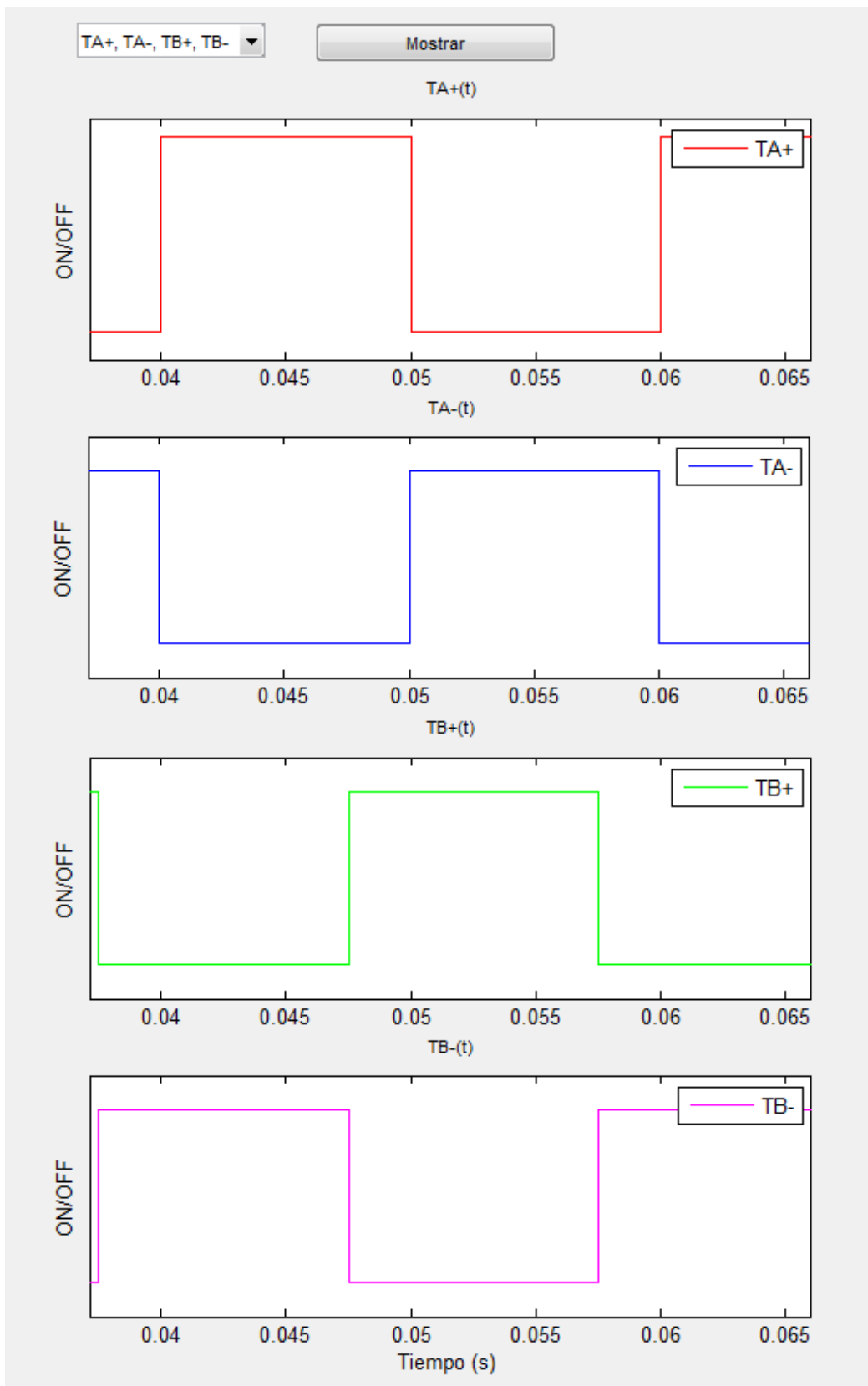


Figura 3.31: Ciclos de los transistores en modulación mediante cancelación de tensión para  $\alpha=45^\circ$ .

Pulsando sobre “**vo, io**” se representa la tensión y la corriente de salida con sus respectivos primeros armónicos. Estas dos señales se pueden observar en la Figura 3.32.

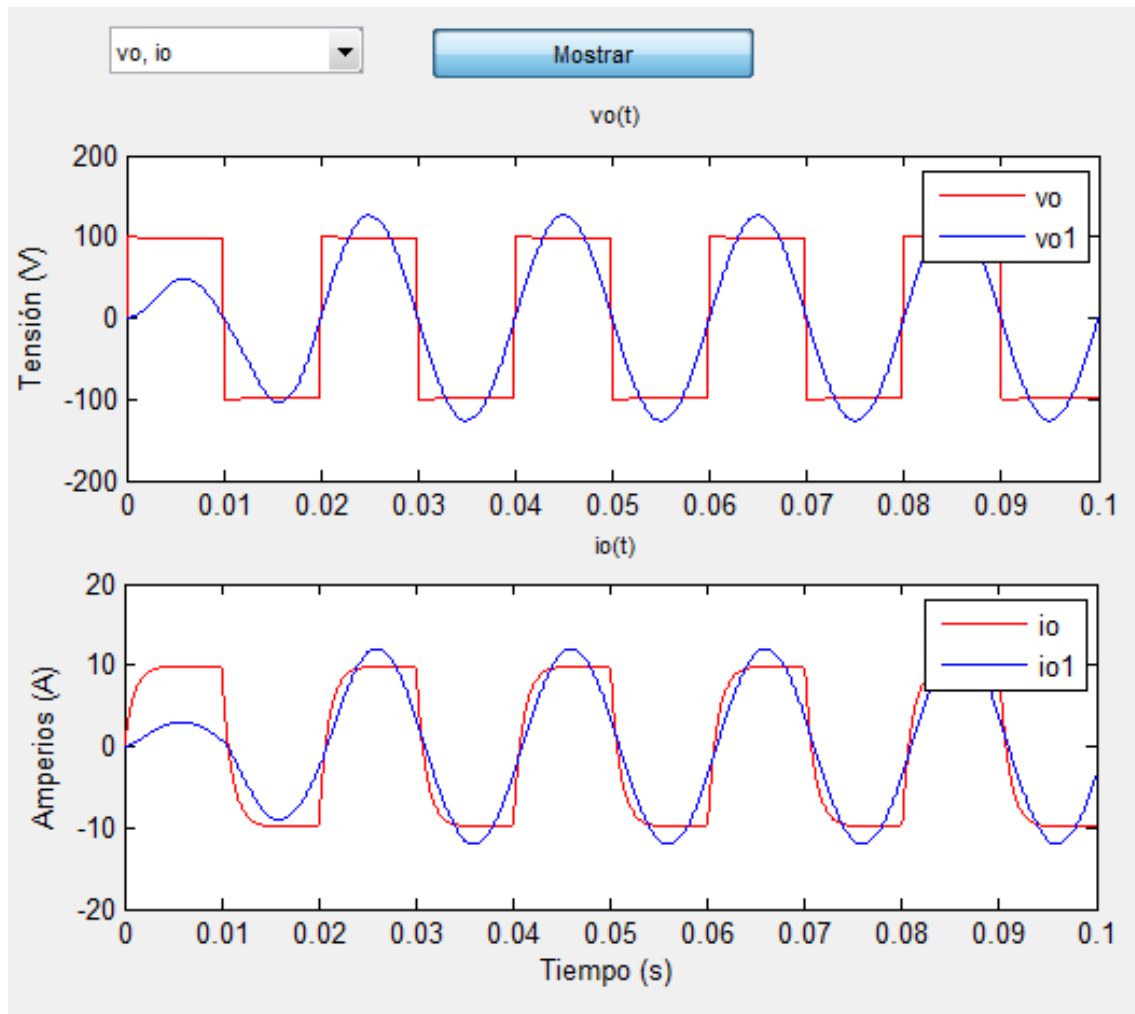


Figura 3.32: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación mediante cancelación de tensión.



La opción “**io, io1, ioripple**” representa la corriente de salida en la primera ventana, su primer armónico en la segunda y el rizado en la tercera. Estas señales pueden verse en la Figura 3.33.

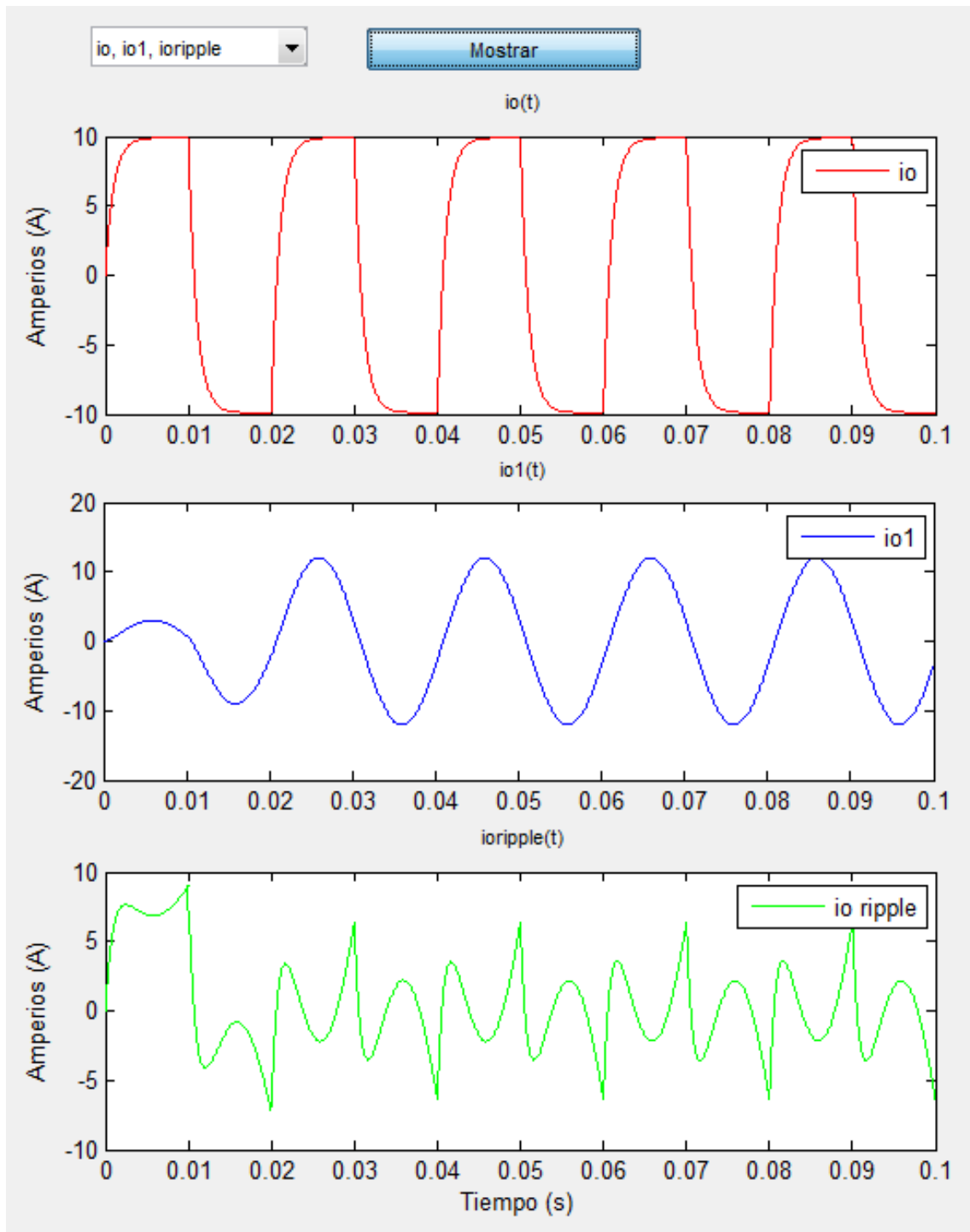


Figura 3.33: Corriente de salida ( $i_o$ ) en modulación mediante cancelación de tensión.

Seleccionando “**iI**, **iC\_I**, **iR\_I**, **iO**” se ve, tal y como se muestra en la Figura 3.34, la evolución temporal de las diferentes corrientes del circuito.

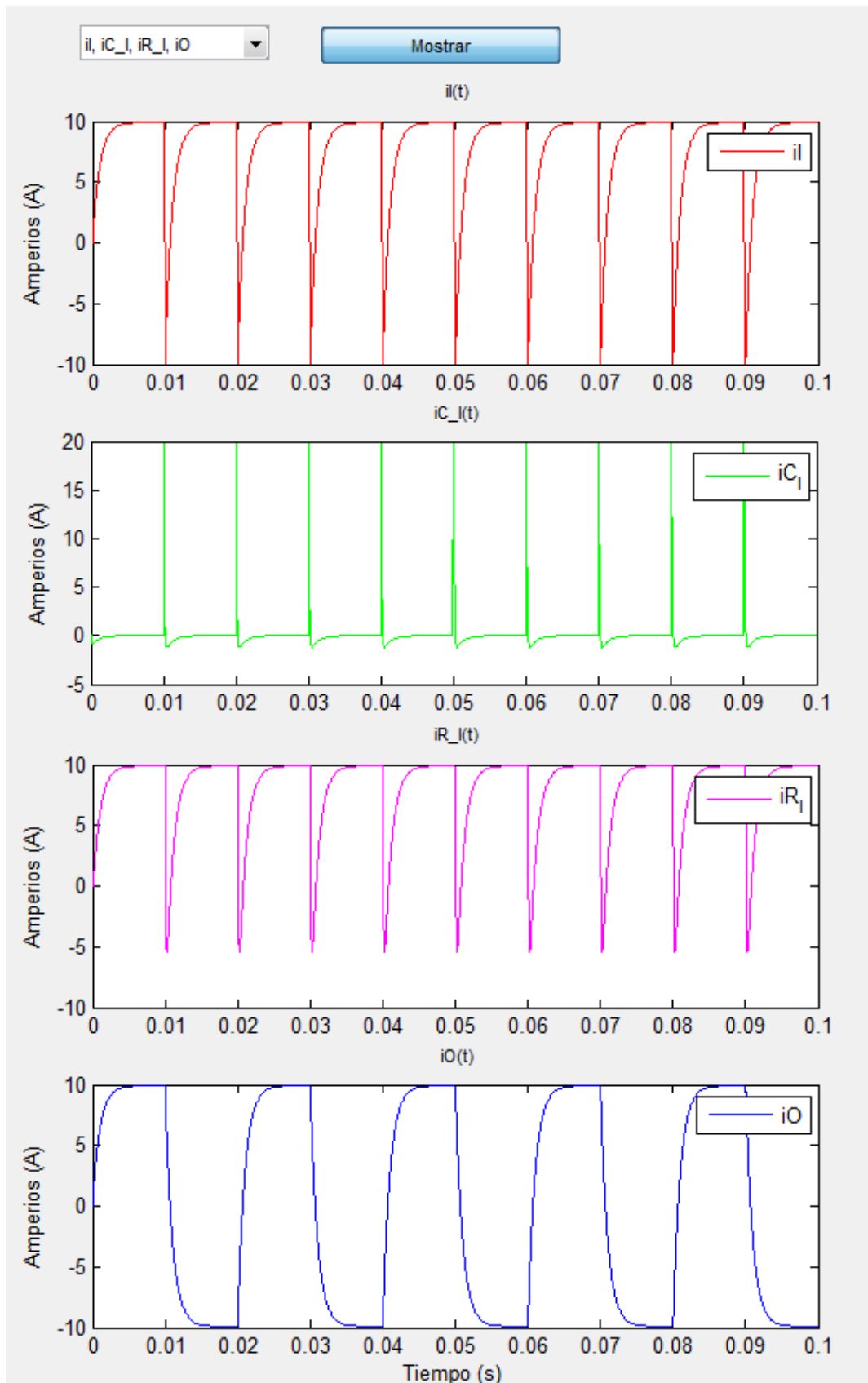


Figura 3.34: Corrientes del circuito en modulación mediante cancelación de tensión.

En esta ventana de representación, la última opción que se permite seleccionar es “**FFT**”. Dicha pestaña muestra el contenido armónico de la tensión de salida  $v_o$  y de la corriente de salida  $i_o$ . Estas gráficas pueden observarse en la Figura 3.35.

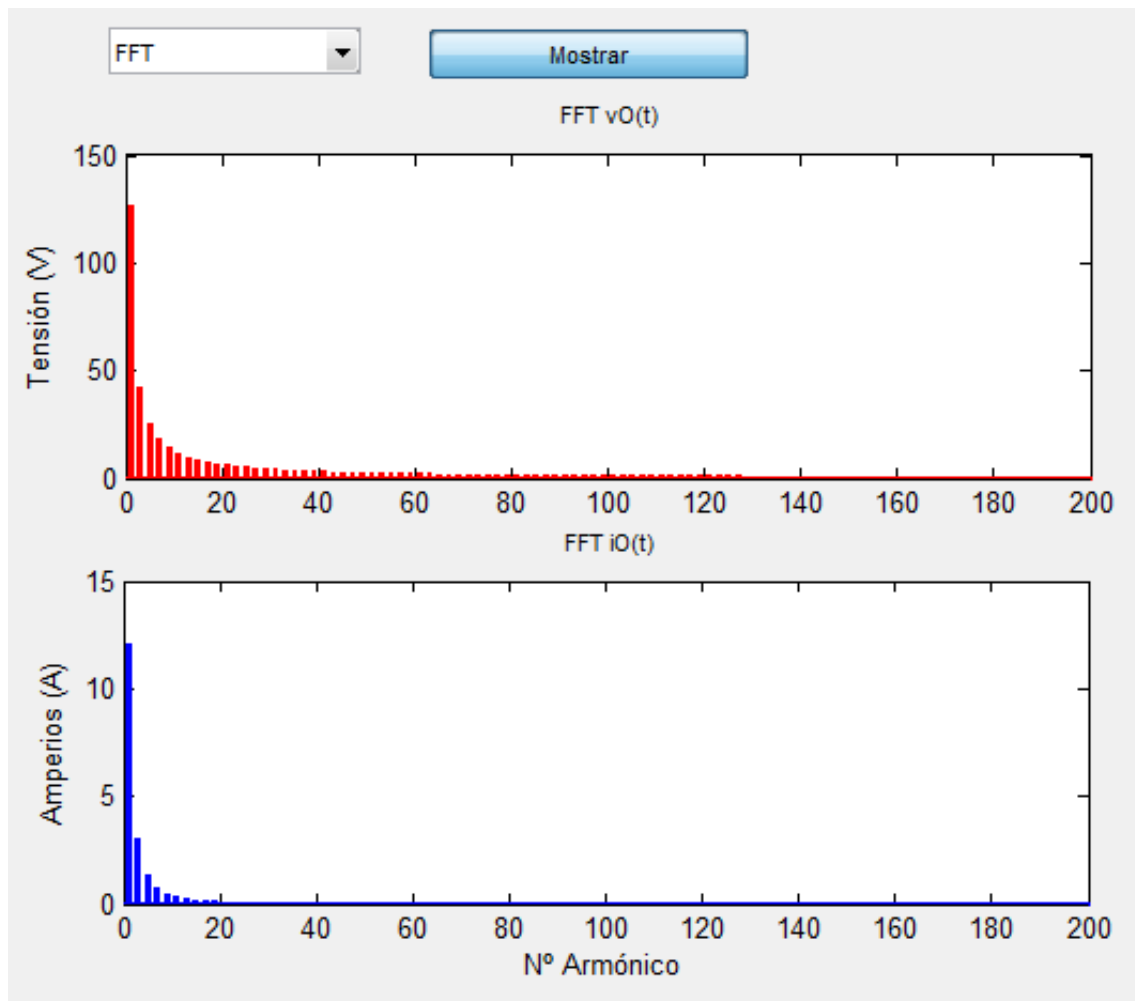


Figura 3.35: Armónicos de tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en modulación mediante cancelación de tensión.

La ventana asociada al botón “**Representación Vo1=f(alpha)**” de la interfaz gráfica básica tiene la misma forma que la Figura 3.17, sin embargo la imagen precalculada guardada para esta modulación es diferente a las técnicas PWM, ya que además del primer armónico se muestra el tercero, el quinto y el séptimo. Estos armónicos son los representados en la Figura 3.36.

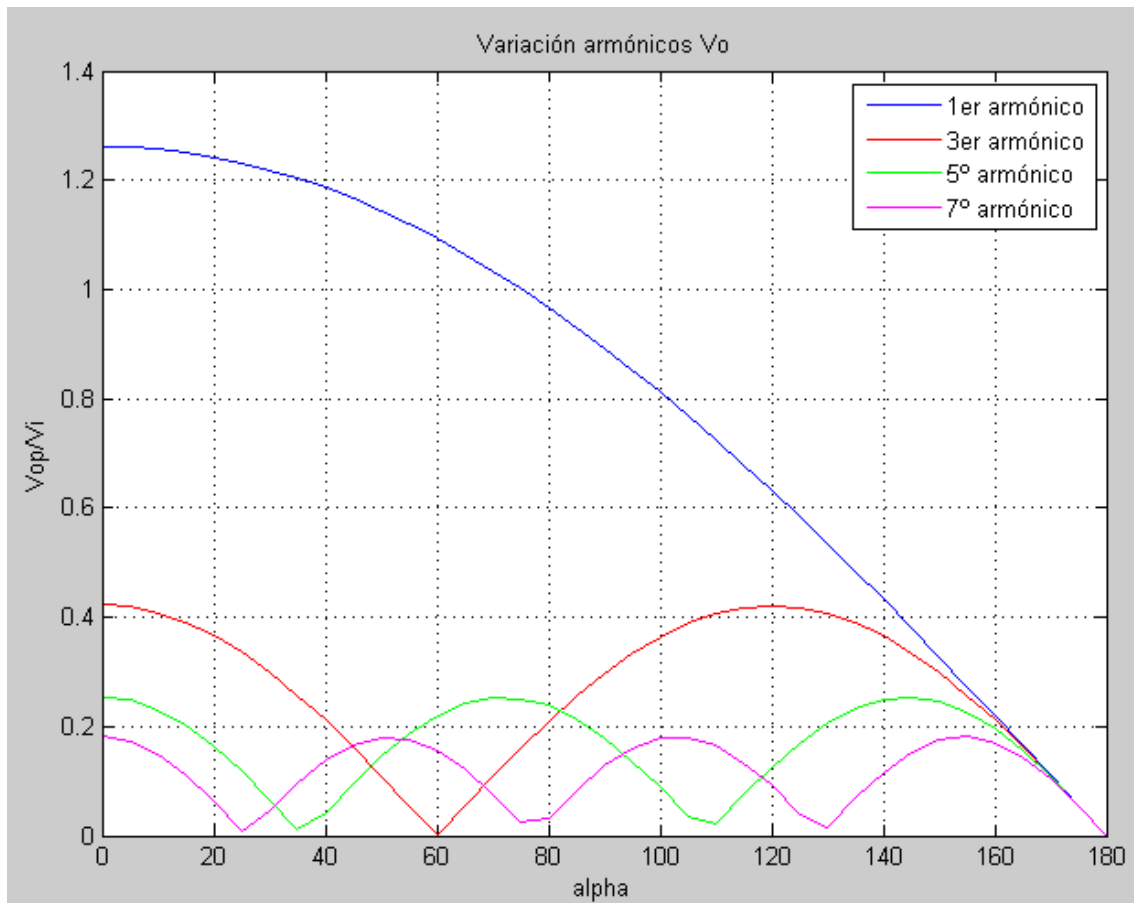


Figura 3.36: Imagen precalculada de barrido en modulación mediante cancelación de tensión.

### 3.2.4 Inversor Push Pull

Al seleccionar la opción Push Pull, el esquema básico cambia respecto a las tres técnicas de modulación anteriores. Esto es debido a que las anteriores técnicas estaban implementadas en un inversor en puente completo, mientras que en este caso hay un transformador a la salida del puente. El nuevo esquema es el mostrado en la Figura 3.37.

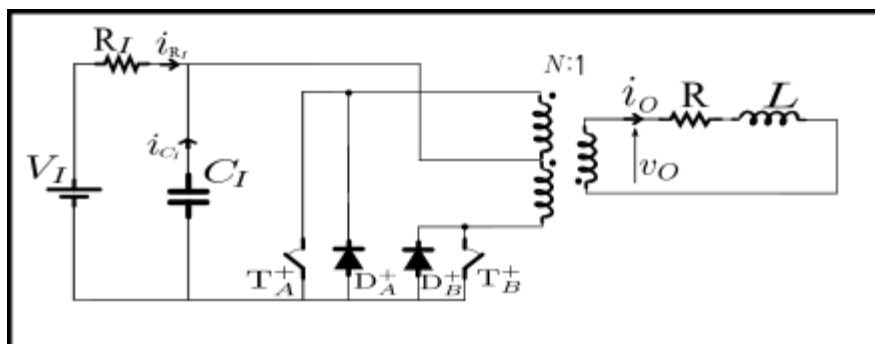


Figura 3.37: Esquema del inversor Push Pull.

En este tipo de inversor, además de los parámetros de entrada vistos en las modulaciones PWM unipolar y bipolar, se añade un nuevo parámetro:

- Relación de espiras (N).

Este nuevo parámetro provoca que al pulsar el botón “Valores por defecto” se observen diferencias respecto a las técnicas anteriores. Este cambio, así como los cambios en “Resultados” se pueden observar en la Figura 3.38.

The screenshot shows a software interface for calculating a Push-Pull inverter. It is divided into several sections:

- Modulaciones:** Radio buttons for 'Modulación Bipolar', 'Modulación Unipolar', 'Cancelación de tensión', and 'Push-Pull' (selected).
- Circuit Diagram:** A schematic of a push-pull inverter with a transformer (N:1), input capacitor (C<sub>I</sub>), and output filter (R, L). Labels include V<sub>I</sub>, R<sub>I</sub>, i<sub>RI</sub>, i<sub>CI</sub>, T<sub>A</sub><sup>+</sup>, D<sub>A</sub><sup>+</sup>, D<sub>B</sub><sup>+</sup>, T<sub>B</sub><sup>+</sup>, i<sub>O</sub>, R, L, and v<sub>O</sub>.
- Introducir Valores:** Input parameters table.
 

Tensión Entrada V <sub>I</sub> (V)	100
Resistencia Entrada R <sub>I</sub> (Ohms)	0.1
Capacidad Entrada C <sub>I</sub> (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto t <sub>d</sub> (s)	0
Resistencia Salida R (Ohms)	10
Inductancia Salida L (H)	0.01
Frecuencia f <sub>1</sub> (Hz)	50
Modulación amplitud ma	0.8
Modulación Frecuencia mf	15
Relación espiras (N)	1
- Resultados:** Output results table.
 

Corriente R <sub>I</sub> (A)	4.85646
Corriente C <sub>I</sub> (A)	2.50997
Potencia Entrada P <sub>I</sub> (W)	358.625
Vo1_rms (V)	55.9503
Delta_iO (A)	17.3167
Potencia Salida P <sub>o</sub> (W)	303.209
THD v <sub>o</sub>	1.46905
THD i <sub>o</sub>	0.253309
- Buttons:** 'Abrir', 'Restaurar .MDL', 'Calcular', 'Borrar', 'Teoria', 'Videotutorial', 'Datos por defecto', 'Descargar en Placa', 'Representación', and 'Representación Vo1=f(ma)'.

Figura 3.38: Cálculos del inversor Push Pull con datos por defecto.

Al pulsar el botón “Abrir”, se abrirá el archivo .mdl de la Figura 3.39. Dentro del bloque “Generador PWM” están los bloques de la Figura 3.40 los cuales implementan la modulación de este inversor. En este tipo de inversor únicamente hay dos transistores, en los cuales se definen los estados al igual que en la modulación PWM bipolar.

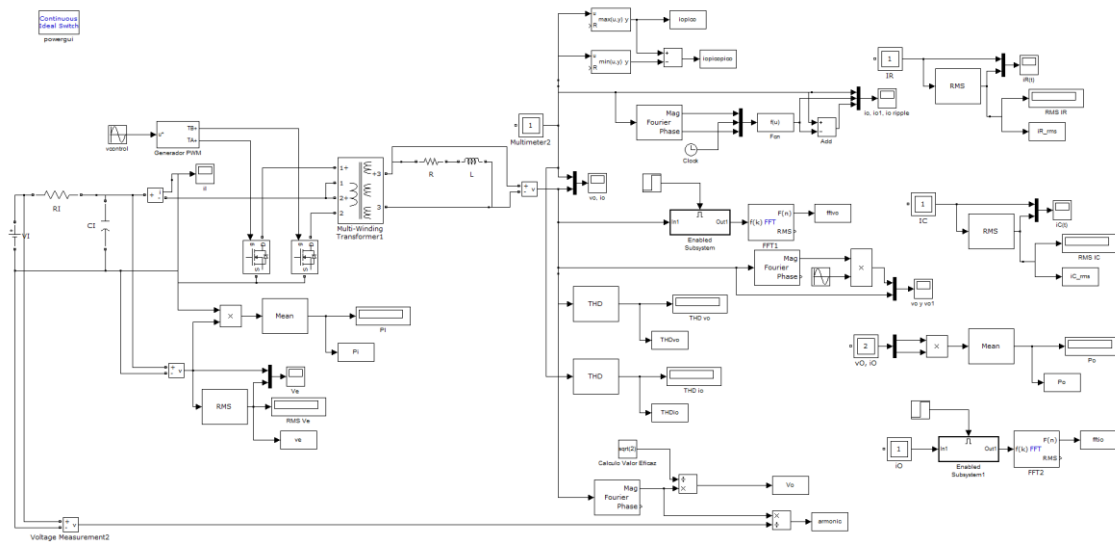


Figura 3.39: Circuito Simulink del inversor Push Pull.

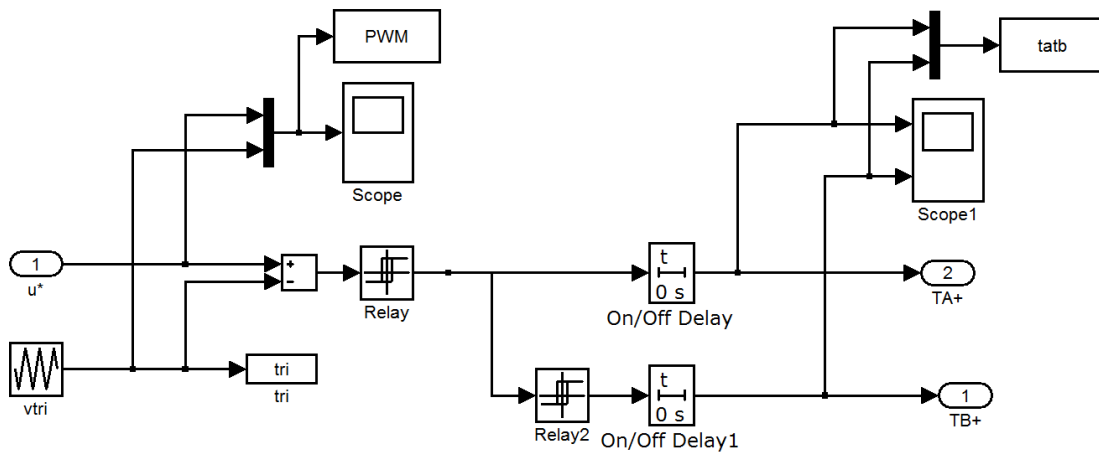


Figura 3.40: Circuito Simulink del Generador PWM en inversor Push Pull.

El botón “**Representación**” abrirá la ventana de representación de la Figura 3.9 .

Al seleccionar “**PWM, TA+, TB+**” se obtiene las señales de modulación PWM unipolar así como el estado de los transistores del puente. Ver Figura 3.41.

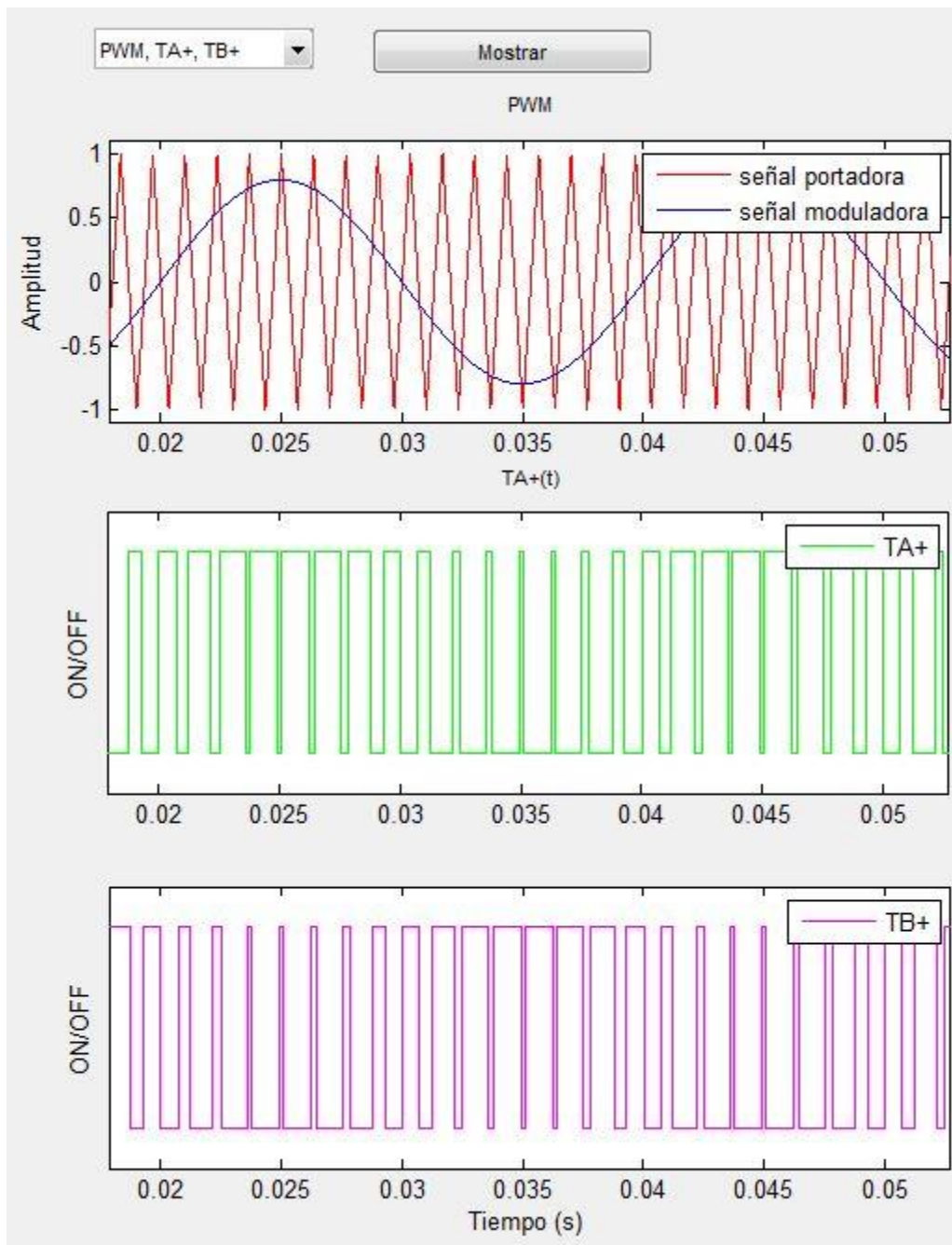


Figura 3.41: Señal PWM, TA+ y TB+ en inversor Push Pull.

Al pulsar sobre “**vo, io**” se mostrara la tensión y la corriente de salida con sus respectivos primeros armónicos. Estas dos señales de salida son representadas en la Figura 3.42.

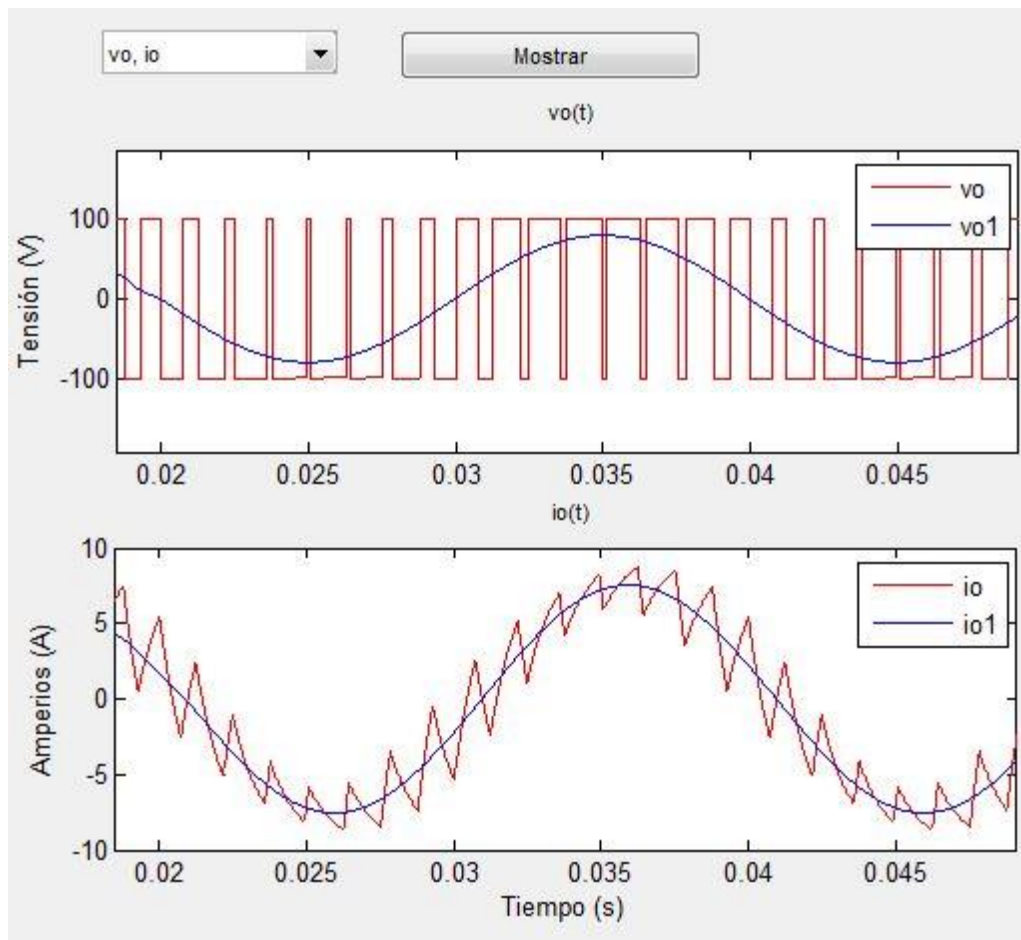


Figura 3.42: Tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en inversor Push Pull.



Seleccionando “**io, io1, ioripple**” se representa la corriente de salida en la primera ventana, su primer armónico en la segunda y el rizado en la tercera tal y como se muestra en la Figura 3.43.

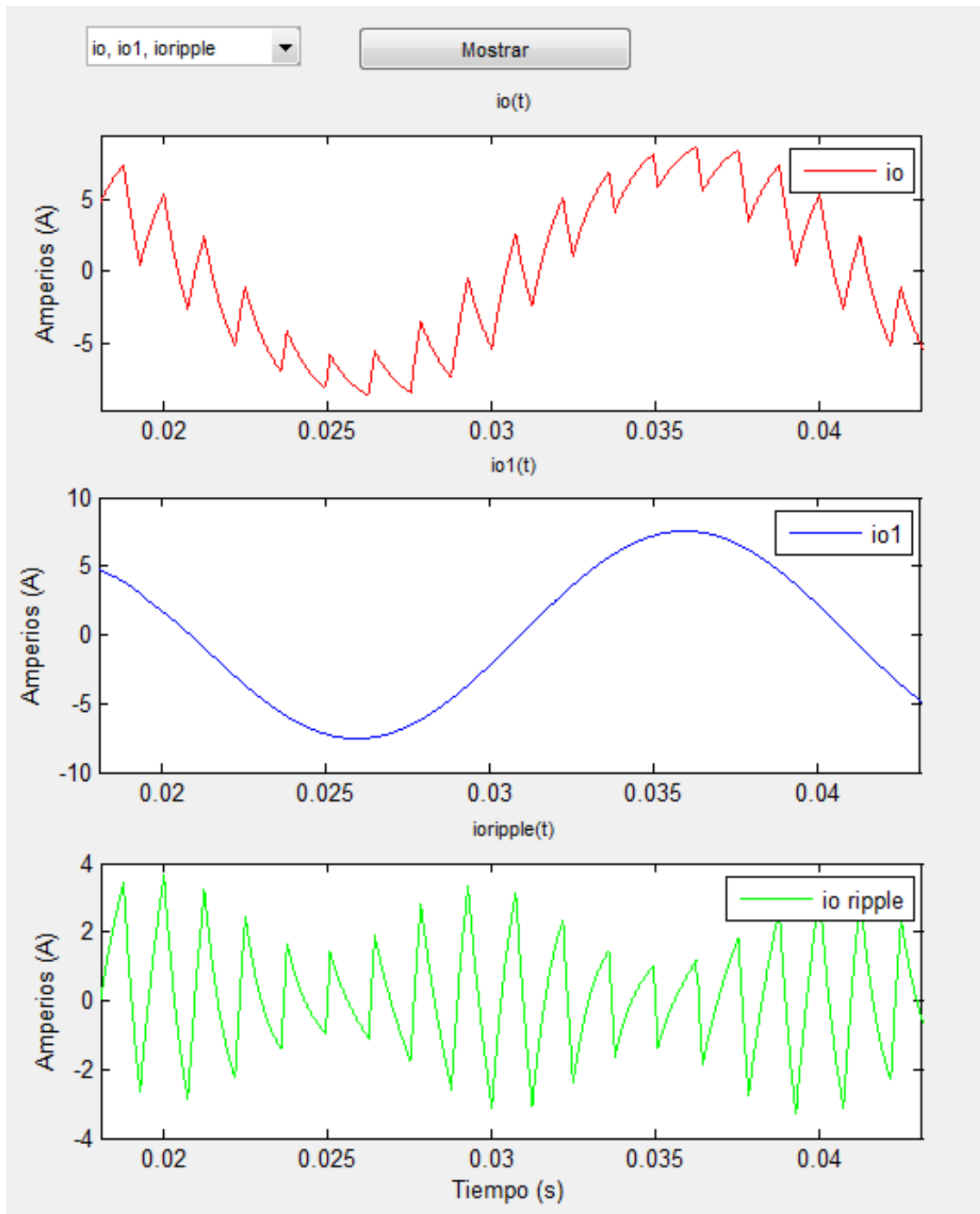


Figura 3.43: Corriente de salida ( $i_o$ ) en inversor Push Pull.

Si se selecciona “**iI, iC<sub>I</sub>, iR<sub>I</sub>, iO**” se puede visualizar la evolución temporal de las diferentes corrientes del circuito. Este comportamiento se muestra en la Figura 3.44.

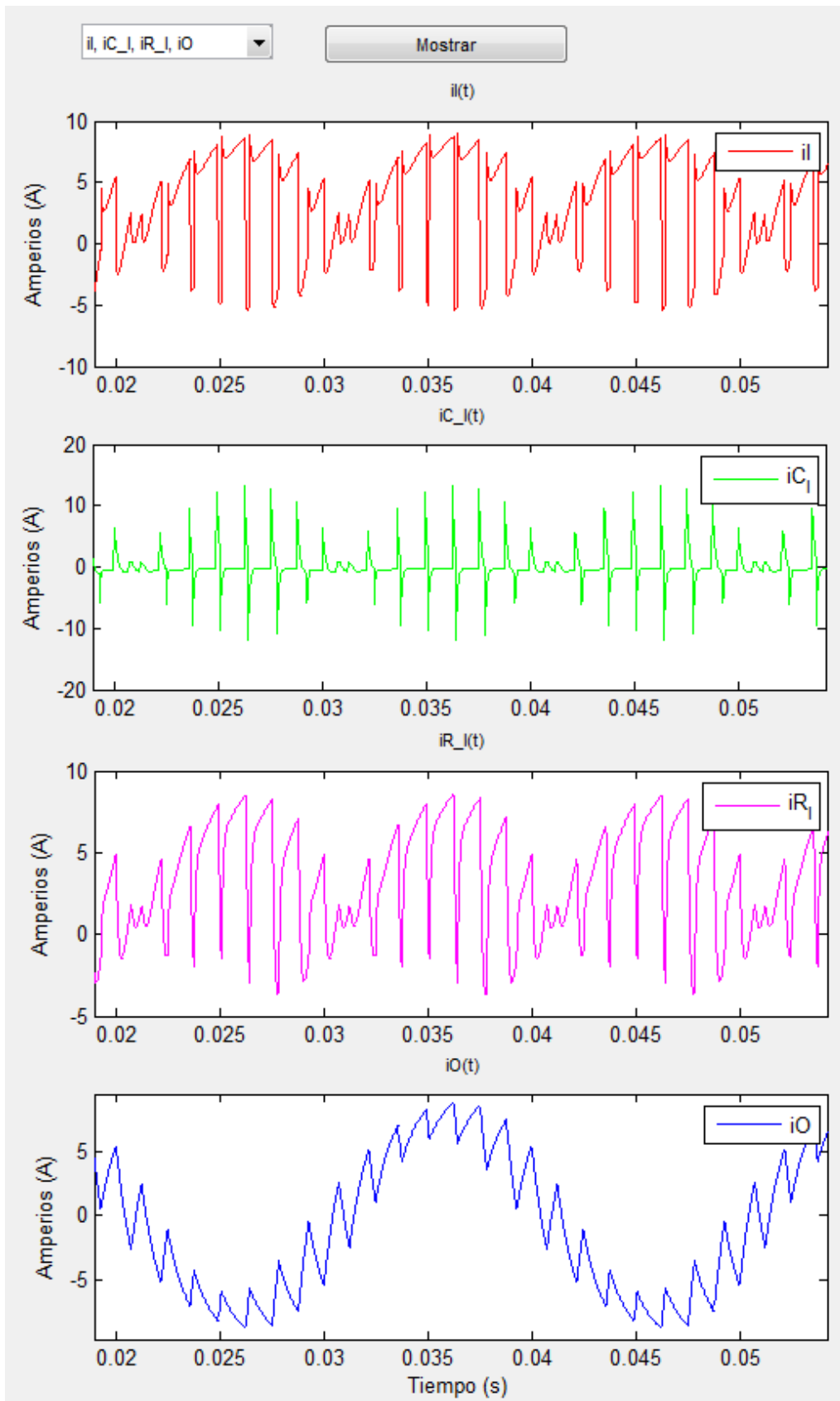


Figura 3.44: Corrientes del circuito en inversor Push Pull.

La siguiente opción que se permite seleccionar es “**FFT**”. En esta pestaña se muestra el contenido armónico de la tensión de salida  $v_o$  y de la corriente de salida  $i_o$ . Estos armónicos pueden verse en la Figura 3.45.

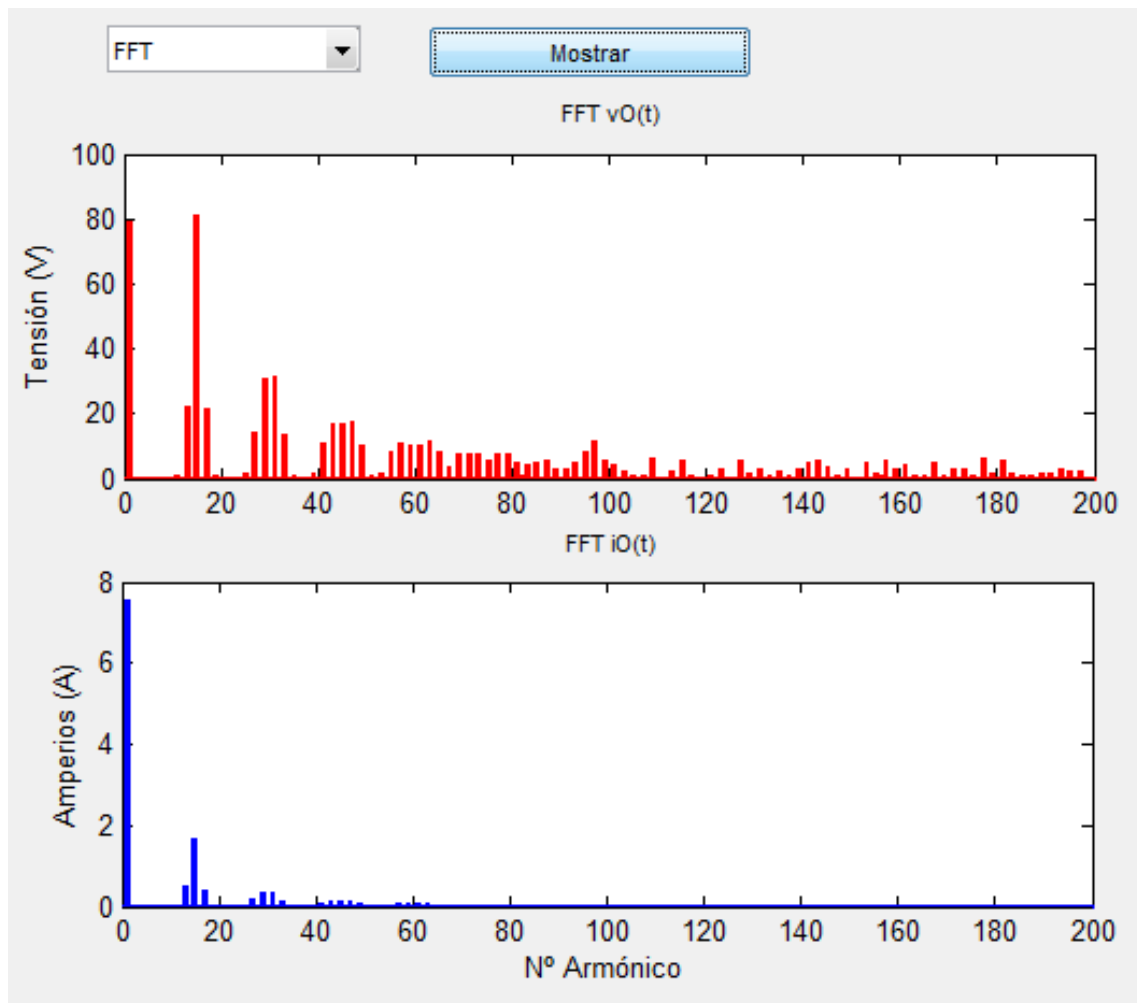


Figura 3.45: Armónicos tensión ( $v_o$ ) y corriente ( $i_o$ ) de salida en inversor Push Pull.

La última opción disponible para seleccionar es “TA+, TA-, TB+, TB-“. En esta opción, correspondiente con la Figura 3.46, se muestra el estado de los dos transistores del circuito (cabe recordar que este inversor sólo cuenta con dos transistores).

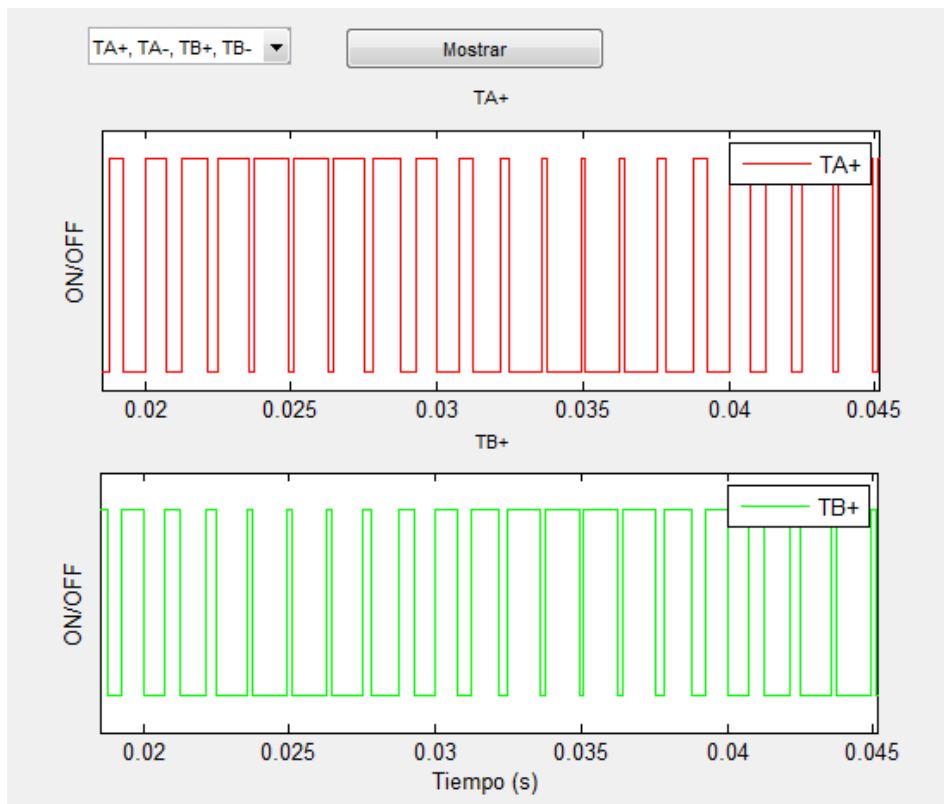


Figura 3.46: Ciclos de los transistores en inversor Push Pull.

La ventana asociada al botón “**Representación  $V_{o1}=f(ma)$** ” de la interfaz gráfica básica es la mostrada en la Figura 3.17. La imagen precalculada guardada para esta modulación es diferente a los casos anteriores como se muestra en la Figura 3.47.

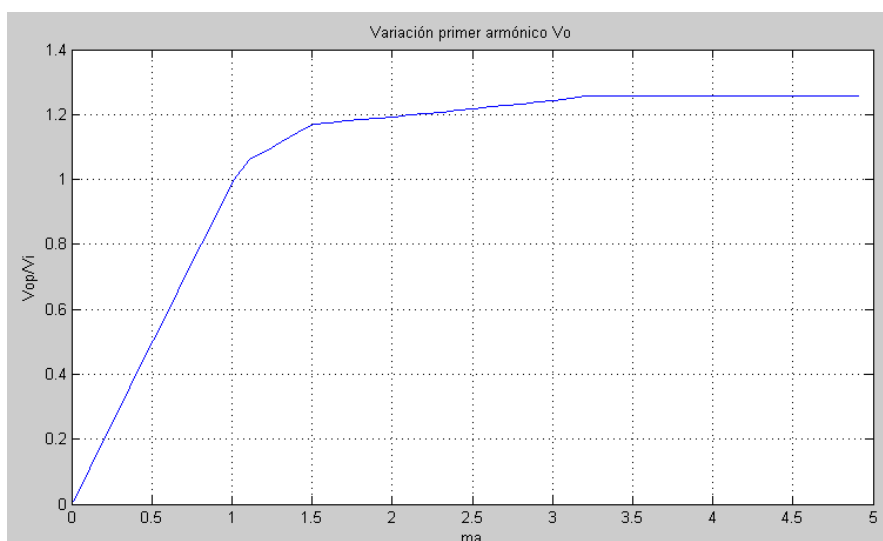


Figura 3.47: Imagen precalculada del barrido en inversor Push Pull.

### 3.3 Interfaz gráfica de convertidores trifásicos

Una vez explicado el funcionamiento de la interfaz gráfica monofásica en el capítulo anterior, se abordará a lo largo de este capítulo el funcionamiento de la interfaz gráfica trifásica.

Tras ejecutar nuevamente el script principal se abrirá la pantalla de inicio de la Figura 3.1.

- Inversor DC/AC Trifásico

Este botón permite abrir la interfaz gráfica asociada con los inversores de potencia trifásicos y sus diferentes variantes, es decir, con el convertidor de tres ramas con modulación PWM, con modulación PWM con inyección de tercer armónico, con modulación PWM con inyección de secuencia cero, con modulación PWM discontinua y con modulación vectorial, tanto en su forma normal como con eliminación de armónicos pares.

Al pulsar el botón “**Inversor DC/AC Trifásico**” se abre la ventana de la Figura 3.48 con la interfaz gráfica asociada a estos inversores.

Los botones “**Borrar**”, “**Descargar en placa**” y “**Videotutorial**” ya han sido explicados en el capítulo anterior, por lo que no se hará referencia a ellos en este capítulo a pesar de estar presentes en la interfaz gráfica trifásica.

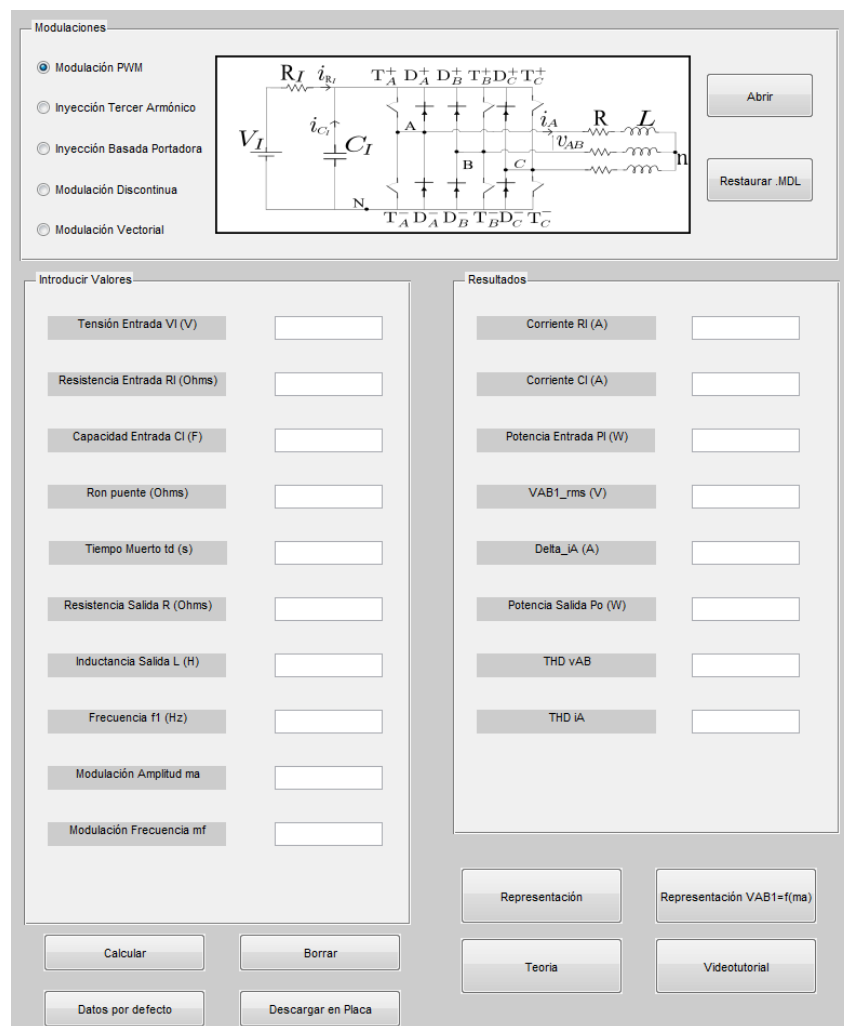


Figura 3.48: Pantalla principal de inversores trifásicos.

En primer lugar está la pestaña “**Modulación**”. En esta pestaña se permite seleccionar el tipo de modulación que se quiere analizar.

Una vez en esta pestaña, se analizará cada modulación por separado.

### 3.3.1 Modulación PWM clásica

Siguiendo en la pestaña “**Modulación**” se puede observar el circuito inversor en tres ramas de la Figura 3.49 en el cual se implementan todas las modulaciones.

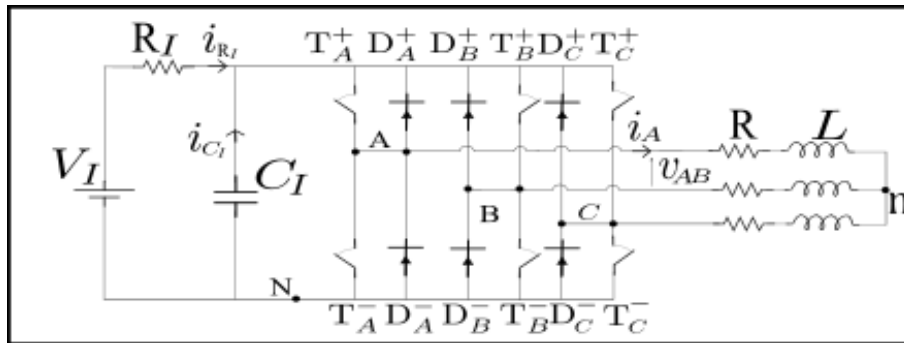


Figura 3.49: Esquema del inversor en tres ramas.

Pulsando el botón “**Abrir**”, se abrirá el archivo .mdl en el cual están implementadas todas las modulaciones, con la excepción del bloque de modulación. En esta técnica de modulación el bloque diferente es el bloque “**Generador PWM**”. En la Figura 3.50 está el archivo .mdl con el puente en tres ramas, mientras que en la Figura 3.51 está la modulación PWM.

En esta técnica de modulación el cálculo de los estados de los transistores se realiza de la misma manera que en la modulación PWM bipolar del caso monofásico, con la diferencia que en este caso hay tres señales de control en lugar de una.

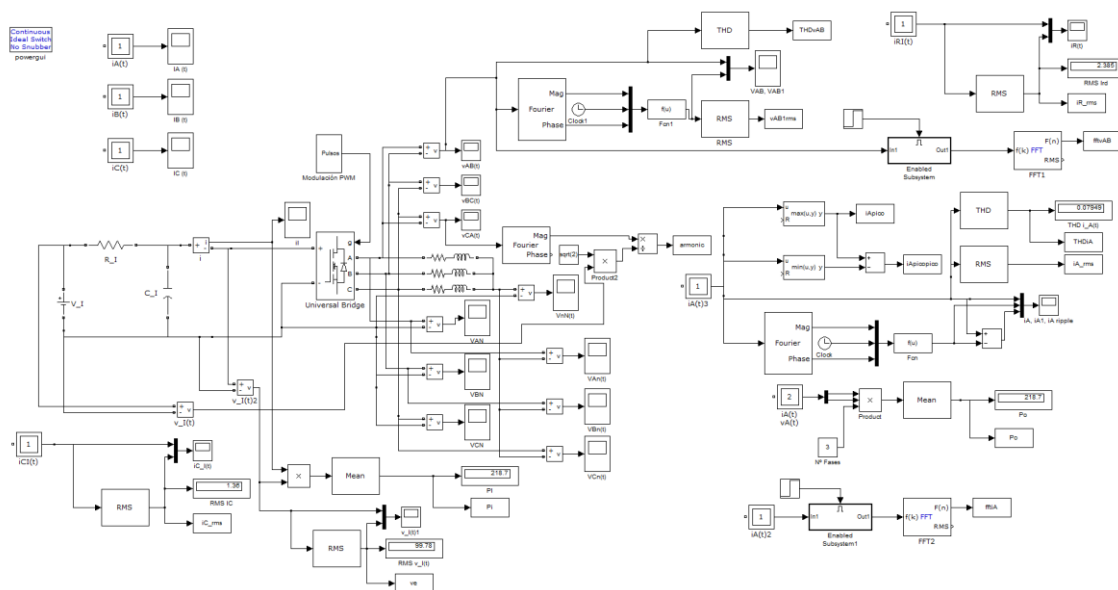


Figura 3.50: Modelo Simulink en puente completo.

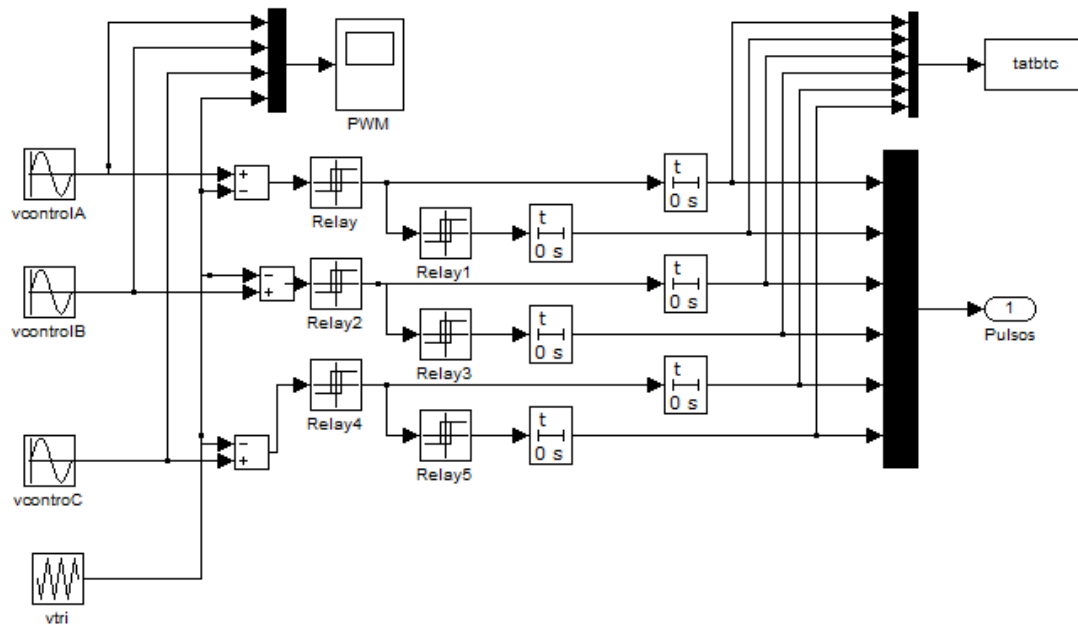


Figura 3.51: Circuito Simulink de modulación PWM.

La siguiente pestaña es “**Introducir Valores**” en la cual se introducen los valores que serán asignados al modelo Simulink.

A diferencia de los inversores monofásicos, en las diferentes modulaciones trifásicas las variables de entrada son las siguientes para todos los modelos:

- Tensión Entrada  $V_I (V)$
- Resistencia Entrada  $R_I$  (Ohmios)
- Capacidad Entrada  $C_I$  (F)
- $R_{on}$  puente (Ohmios)
- Tiempo Muerto  $t_d$  (s)
- Resistencia Salida  $R$  (Ohmios)
- Inductancia Salida  $L$  (H)
- Frecuencia  $f_1$  (Hz)
- Modulación Amplitud  $m_a$
- Modulación Frecuencia  $m_f$

La otra pestaña perteneciente a la interfaz es “**Resultado**”. En esta pestaña se obtienen los resultados calculados en la simulación. Para las diferentes técnicas de modulación los resultados son los siguientes:

- Corriente  $R_I$  (A)
- Corriente  $C_I$  (A)
- Potencia Entrada  $P_I$  (W)
- $V_{01_{rms}}$  (V)
- $\Delta i_o$  (A)
- Potencia Salida  $P_0$  (W)
- $THD_{v_o}$
- $THD_{i_o}$

En la parte inferior de la interfaz hay una serie de botones con diversas funcionalidades que serán explicados a continuación.

El botón “**Teoría**” abre los apuntes de la asignatura sobre inversores trifásicos, lo que sirve para complementar la herramienta y ayudar al alumno a buscar respuesta a diferentes problemas que puedan surgir. La portada de estas transparencias es la mostrada en la Figura 3.52.

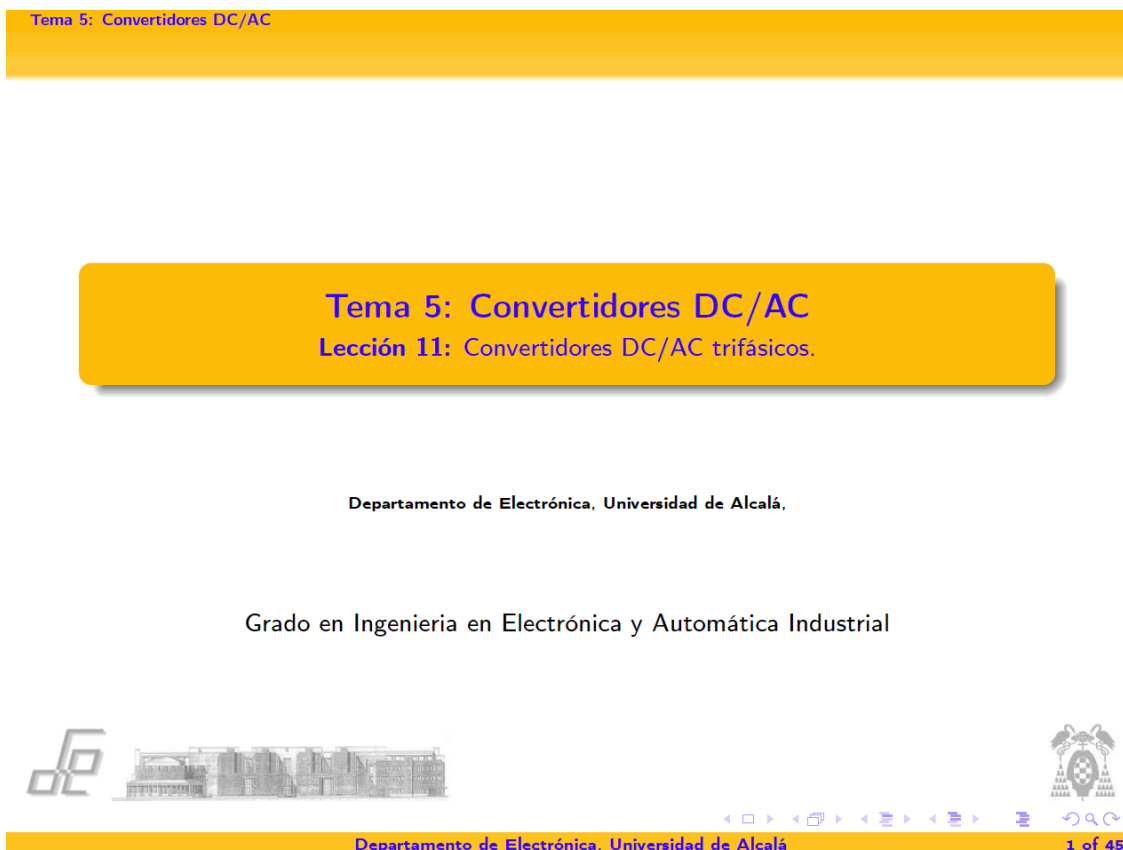


Figura 3.52: Transparencias sobre inversores trifásicos de la asignatura.



El botón “**Calcular**” ejecuta el modelo y tras el tiempo de simulación aparecen los resultados. Esta funcionalidad se puede ver en la Figura 3.53.

**Modulaciones**

- Modulación PWM
- Inyección Tercer Armónico
- Inyección Basada Portadora
- Modulación Discontinua
- Modulación Vectorial

**Introducir Valores**

Tensión Entrada $V_I$ (V)	100
Resistencia Entrada $R_I$ (Ohms)	0.1
Capacidad Entrada $C_I$ (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto $t_d$ (s)	0
Resistencia Salida $R$ (Ohms)	10
Inductancia Salida $L$ (H)	0.01
Frecuencia $f_1$ (Hz)	50
Modulación Amplitud $m_a$	0.8
Modulación Frecuencia $m_f$	21

**Resultados**

Corriente $R_I$ (A)	2.38533
Corriente $C_I$ (A)	1.3603
Potencia Entrada $P_I$ (W)	218.686
$V_{AB1\_rms}$ (V)	48.8609
Delta_ $i_A$ (A)	8.4979
Potencia Salida $P_o$ (W)	218.664
THD $v_{AB}$	0.913628
THD $i_A$	0.0794865

Buttons: **Calcular**, **Borrar**, **Datos por defecto**, **Descargar en Placa**, **Abrir**, **Restaurar .MDL**, **Representación**, **Teoría**, **Videotutorial**

Figura 3.53: Cálculos con modulación PWM y datos por defecto.

El botón “**Representación**” abrirá la ventana mostrada en la Figura 3.54.

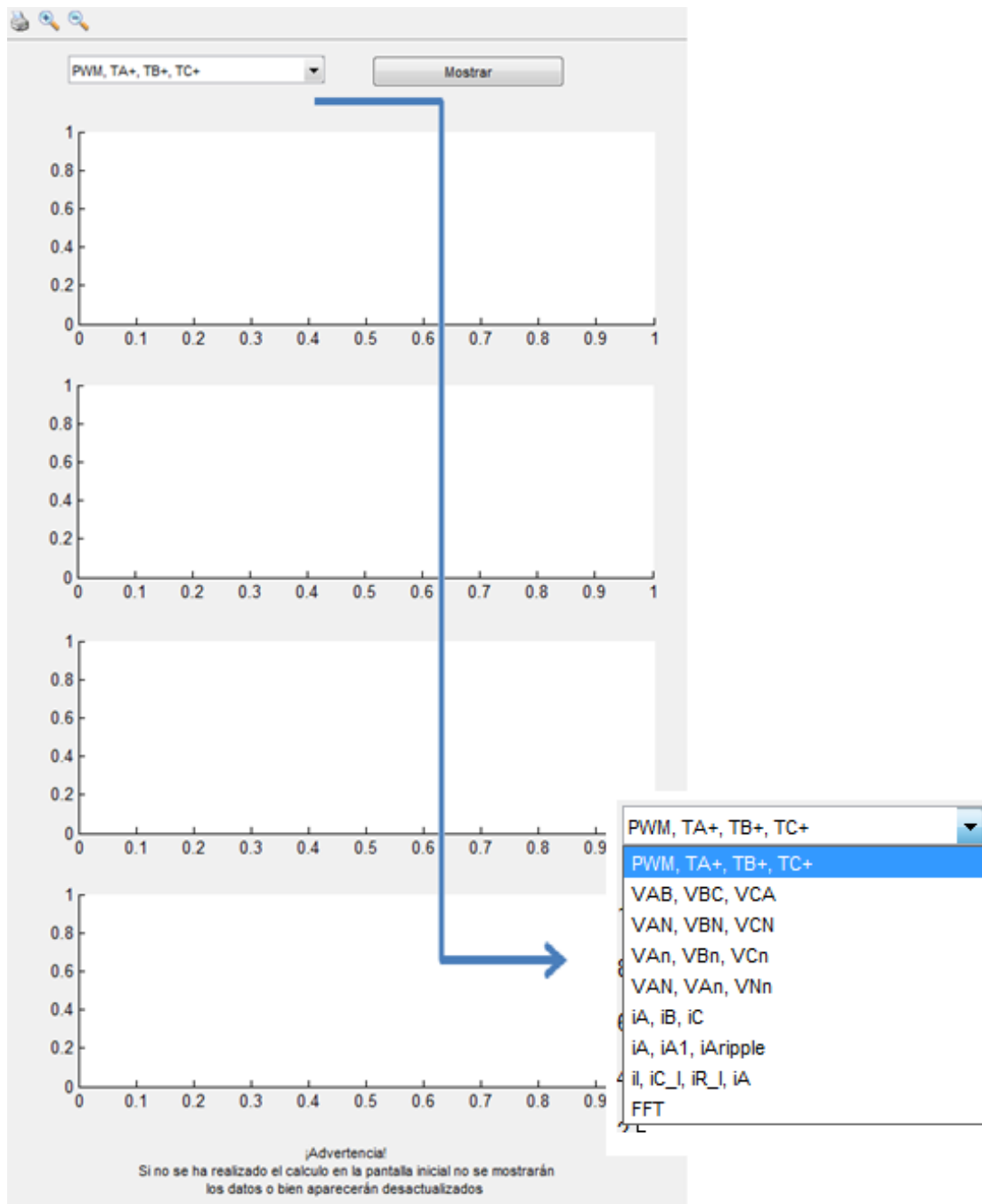


Figura 3.54: Pantalla de representación formas de onda trifásica con PWM.

En esta ventana hay un desplegable que permite seleccionar entre nueve diferentes opciones de visualización, las cuales aparecerán en las gráficas tras pulsar el botón “**Mostrar**”.

Al seleccionar “**PWM, TA+, TB+, TC+**” se obtiene las señales de modulación PWM así como el estado de los transistores superiores del inversor, tal y como puede verse en la Figura 3.55.

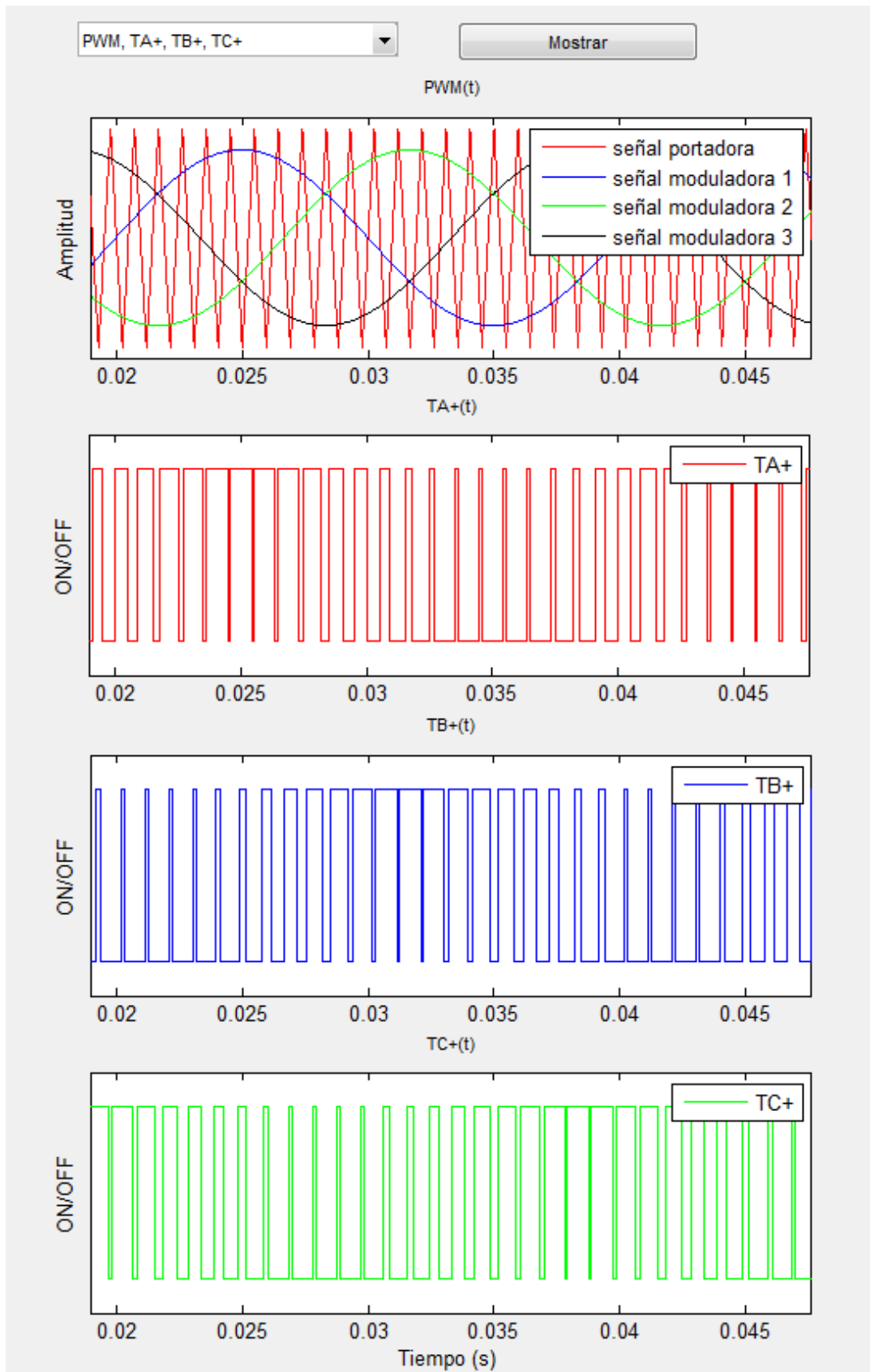


Figura 3.55: Señales PWM, TA+, TB+ y TC+ con modulación PWM.

Al pulsar sobre “**VAB, VBC, VCA**” se mostrarán las tensiones de línea del sistema, tal y como se observa en la Figura 3.56.

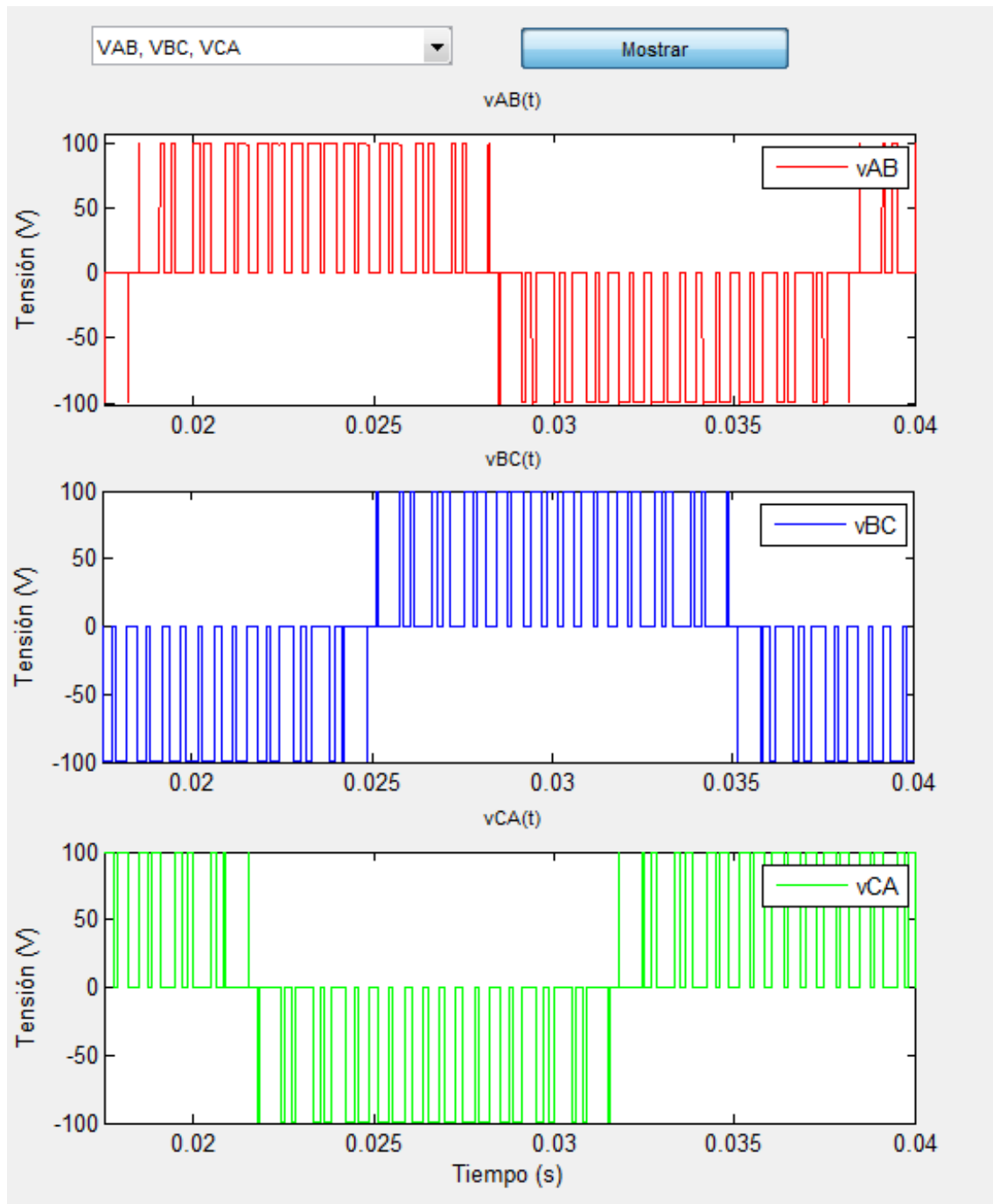


Figura 3.56: Tensiones  $v_{AB}$ ,  $v_{BC}$  y  $v_{CA}$  en modulación PWM.

Al pulsar sobre “VAN, VBN, VCN” se mostrarán las tensiones de cada fase con respecto a la entrada. Esto se observa en la Figura 3.57.

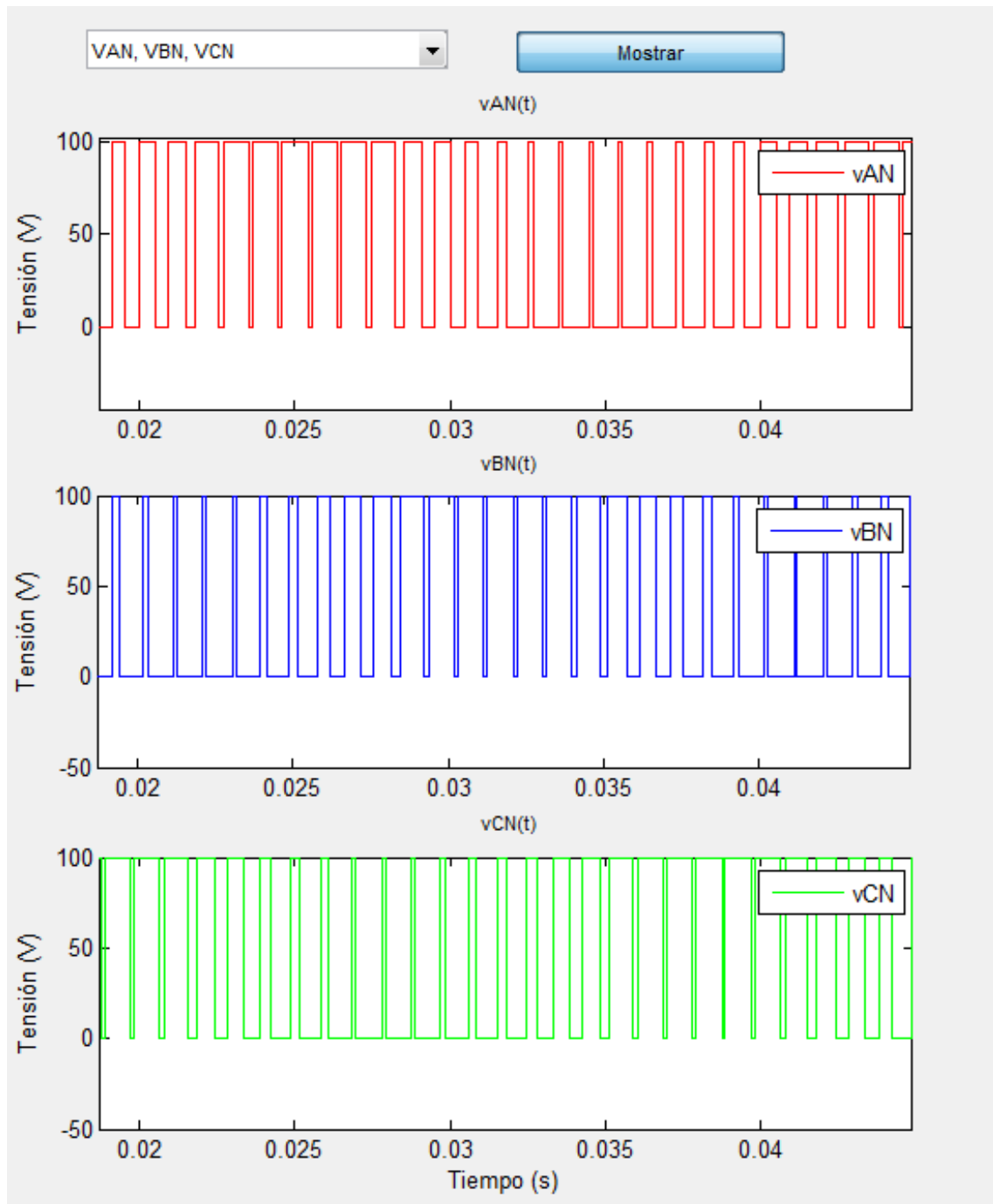


Figura 3.57: Tensiones  $v_{AN}$ ,  $v_{BN}$  y  $v_{CN}$  en modulación PWM.

Seleccionando sobre “**VAn, VBn, VCn**” se mostrarán las tensiones de cada fase, lo que puede verse en la Figura 3.58.

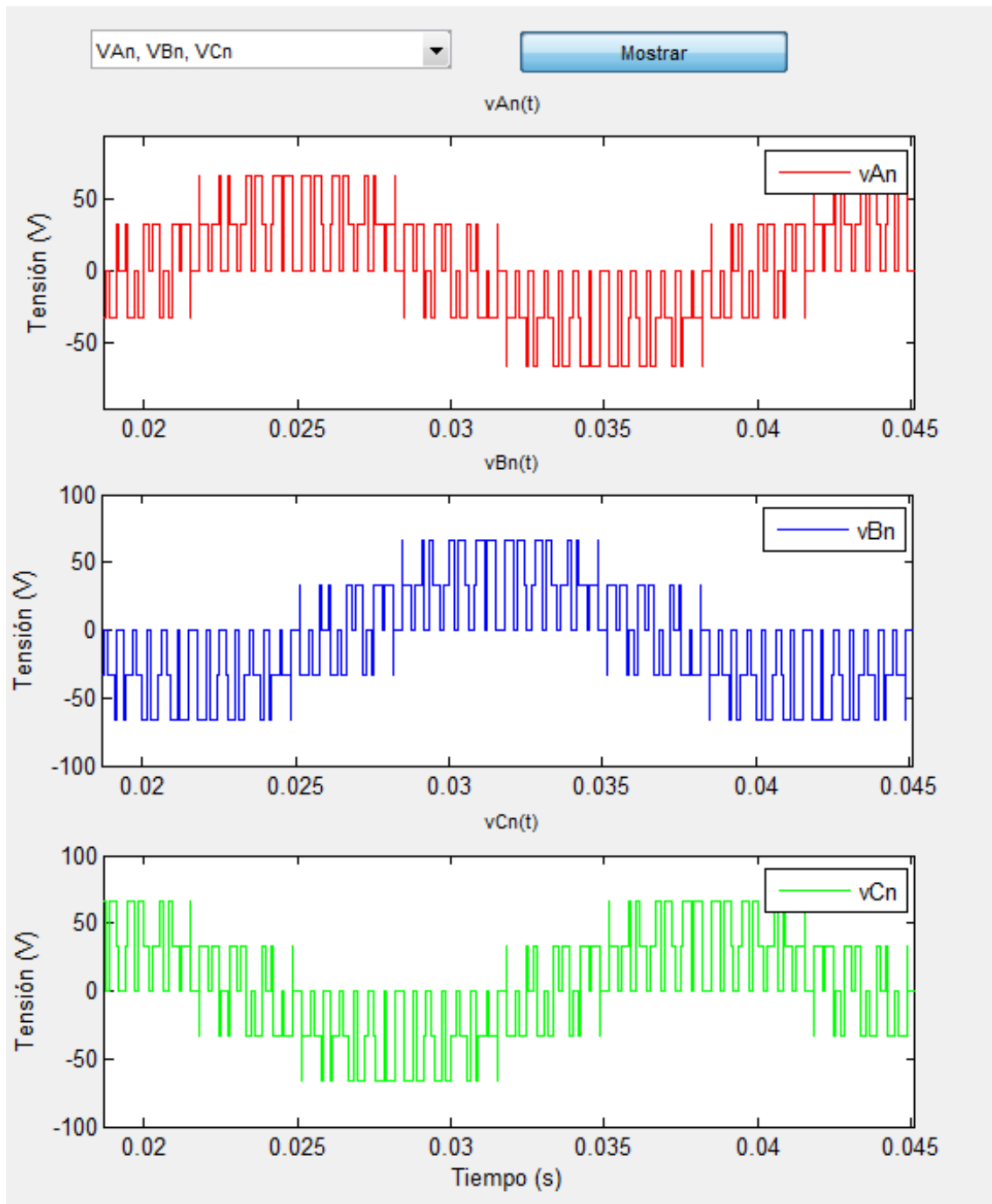


Figura 3.58: Tensiones  $v_{An}$ ,  $v_{Bn}$  y  $v_{Cn}$  en modulación PWM.

La siguiente opción que se puede seleccionar es “VAN, VAn, VNn”. En esta opción se visualizan las dos tensiones anteriores en la fase A además de la diferencia entre ambas, las cuales se aprecian en la Figura 3.59.

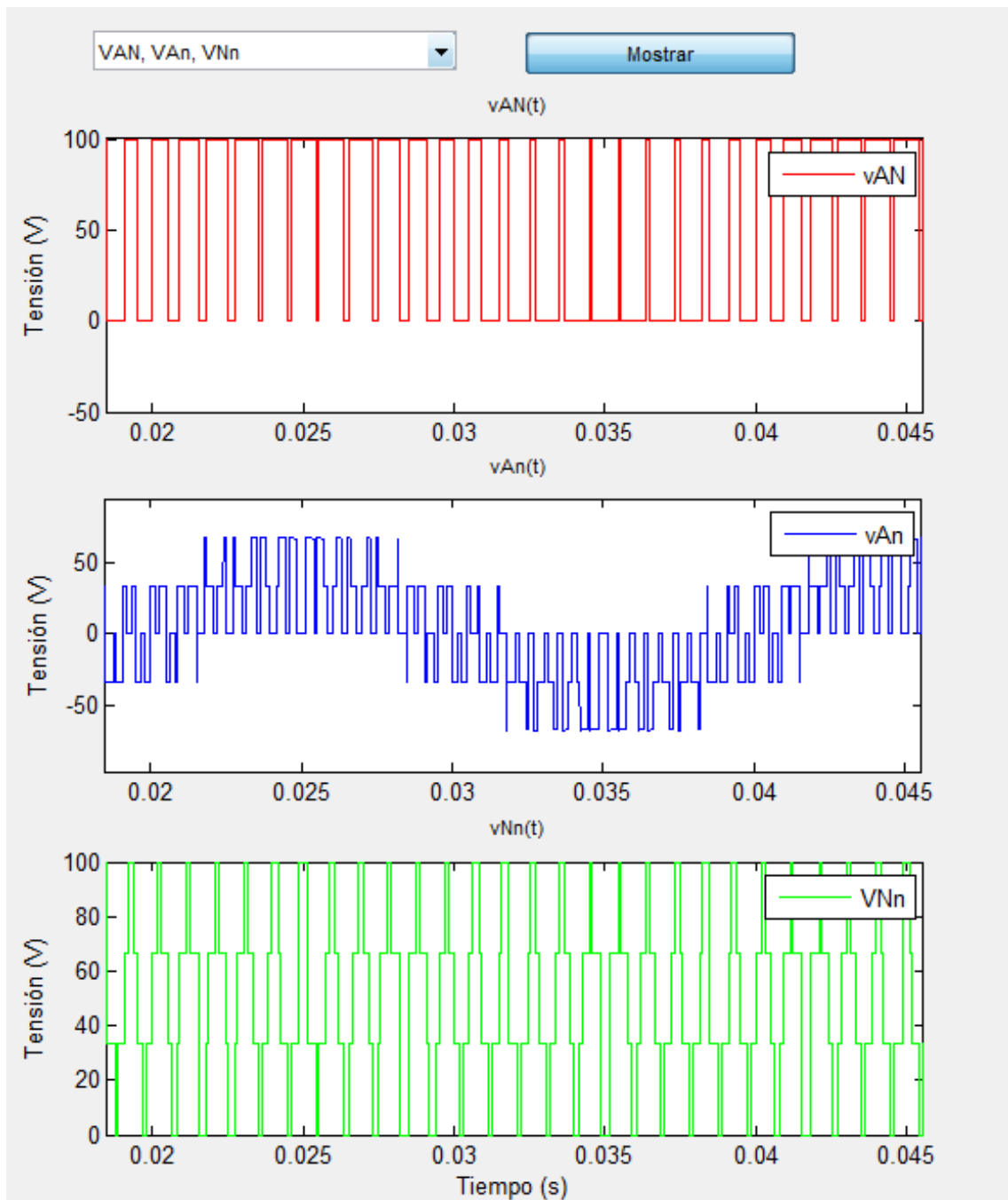


Figura 3.59: Tensiones  $v_{AN}$ ,  $v_{An}$  y  $v_{Nn}$  en modulación PWM.

Vistas todas las tensiones interesantes del circuito, la siguiente opción “**iA, iB, iC**” representa las corrientes de fase de la salida del inversor. Estas corrientes se muestran en la Figura 3.60.

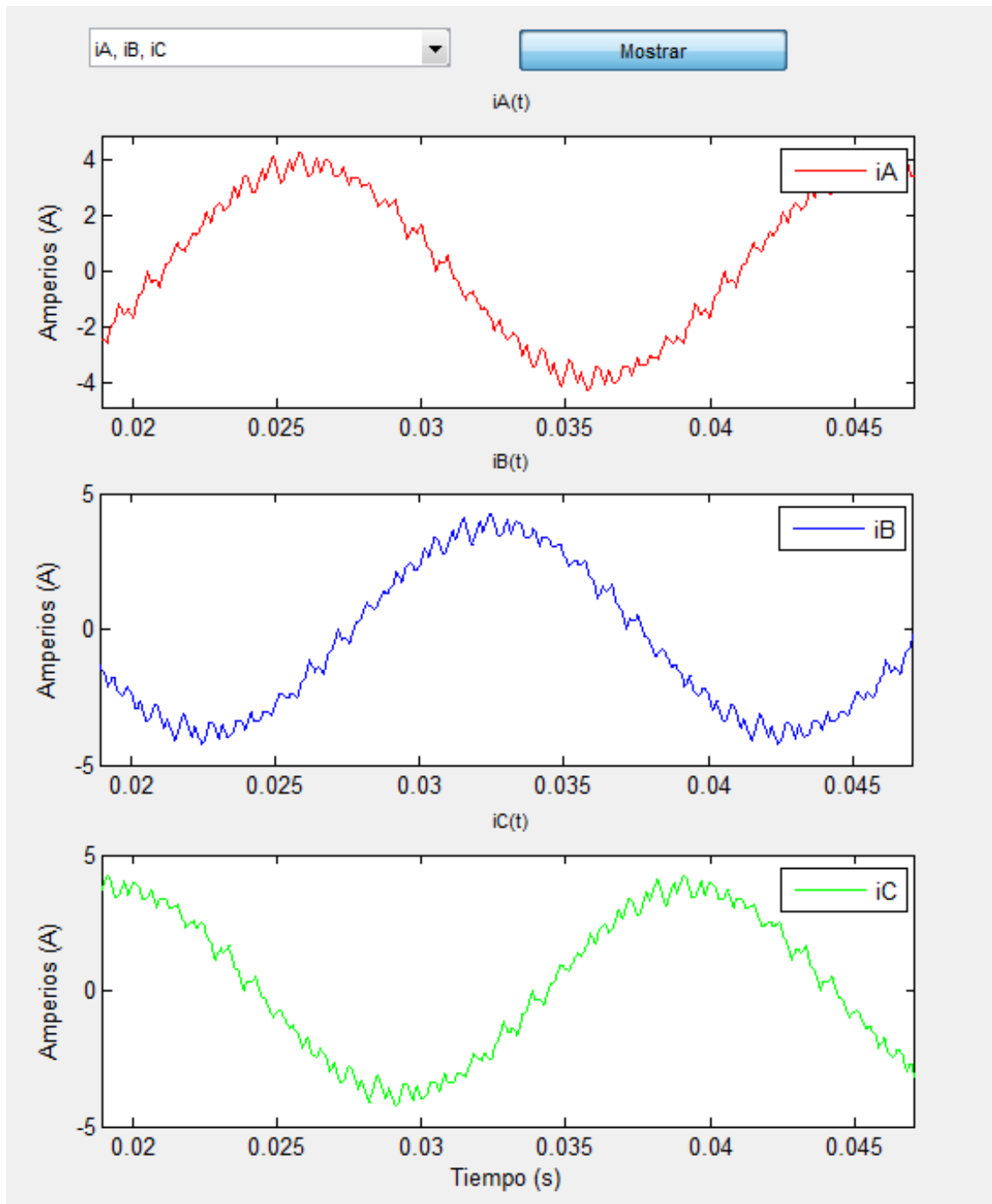


Figura 3.60: Corrientes de fase  $i_A$ ,  $i_B$  y  $i_C$  en modulación PWM.



La siguiente opción, “**iA, iA1, iAripple**”, muestra la corriente por la fase A, además de su primer armónico y del rizado de la señal, tal y como se muestra en la Figura 3.61.

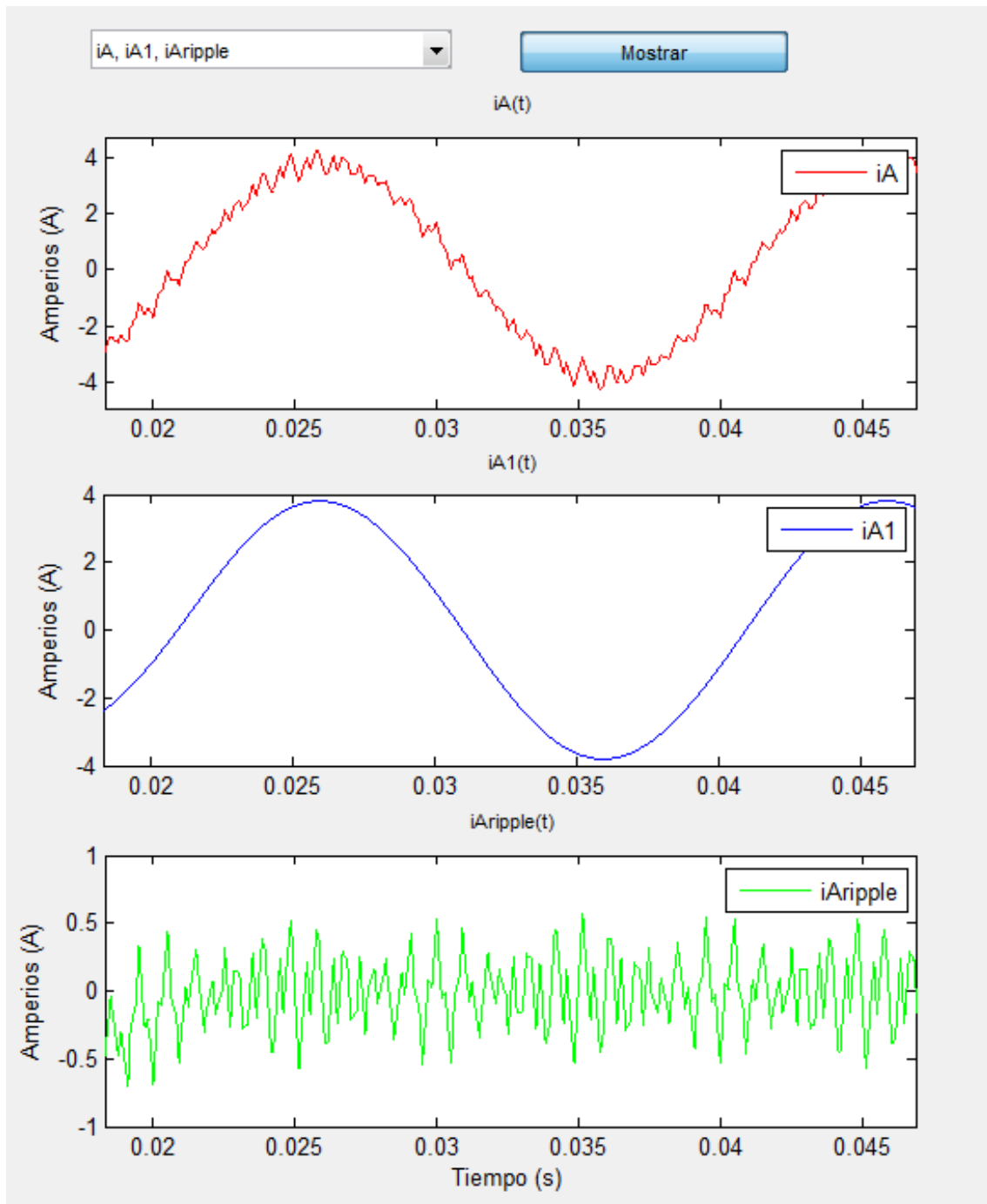


Figura 3.61: Corrientes de fase  $i_A$ ,  $i_{A1}$  y  $i_{Aripple}$  en modulación PWM.

Al igual que ocurría en el caso de inversores monofásicos, en la pestaña “ $i_l$ ,  $i_{C_I}$ ,  $i_{R_I}$ ,  $i_A$ ” se muestran las diferentes corrientes del circuito. En la Figura 3.62 se observan dichas corrientes.

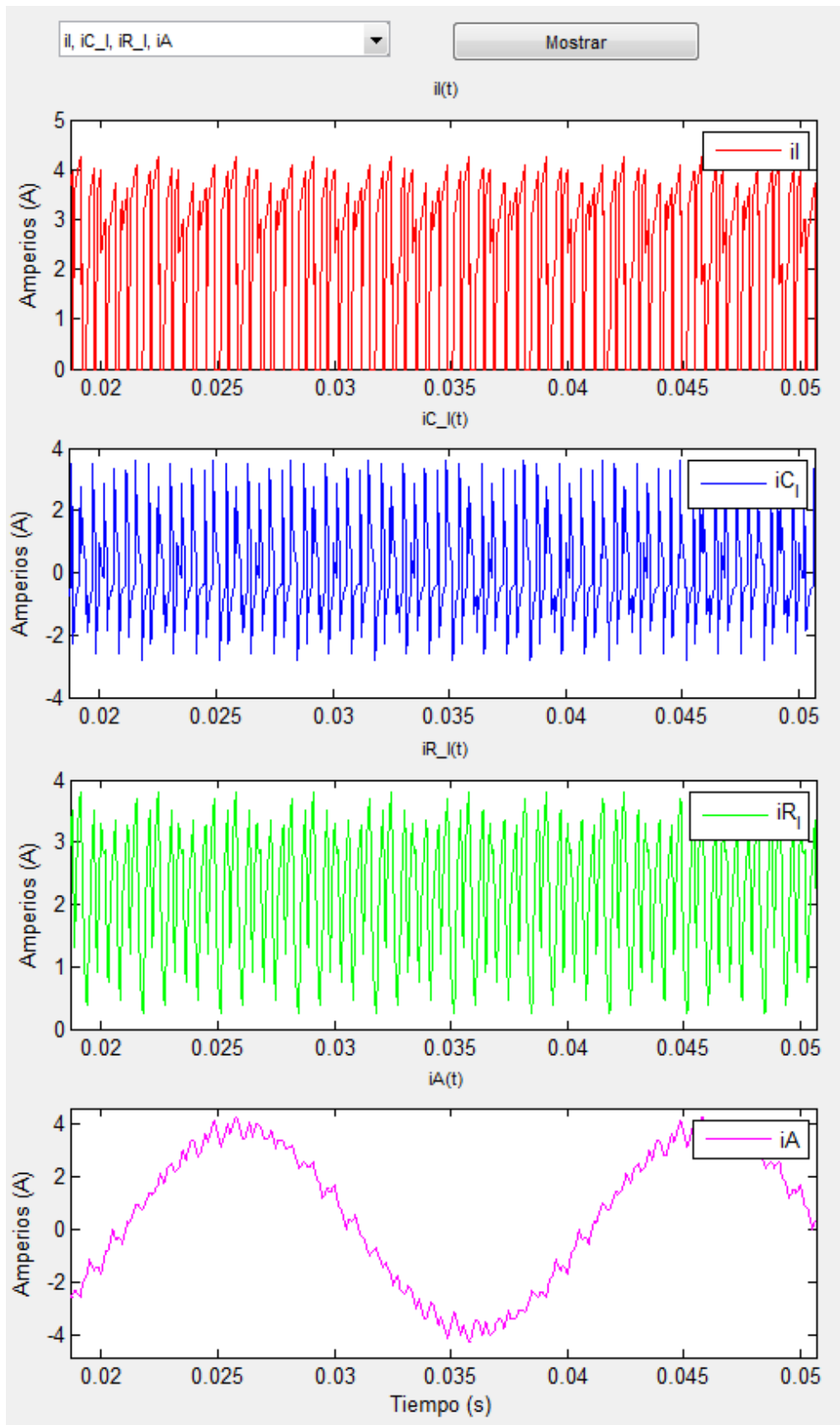


Figura 3.62: Corrientes  $i_l$ ,  $i_{C_I}$  y  $i_{R_I}$  e  $i_A$  en modulación PWM.

Por último, en la pestaña “**FFT**” mostrada en la Figura 3.63, se puede ver el análisis de armónicos de la tensión de línea y de la corriente de fase.

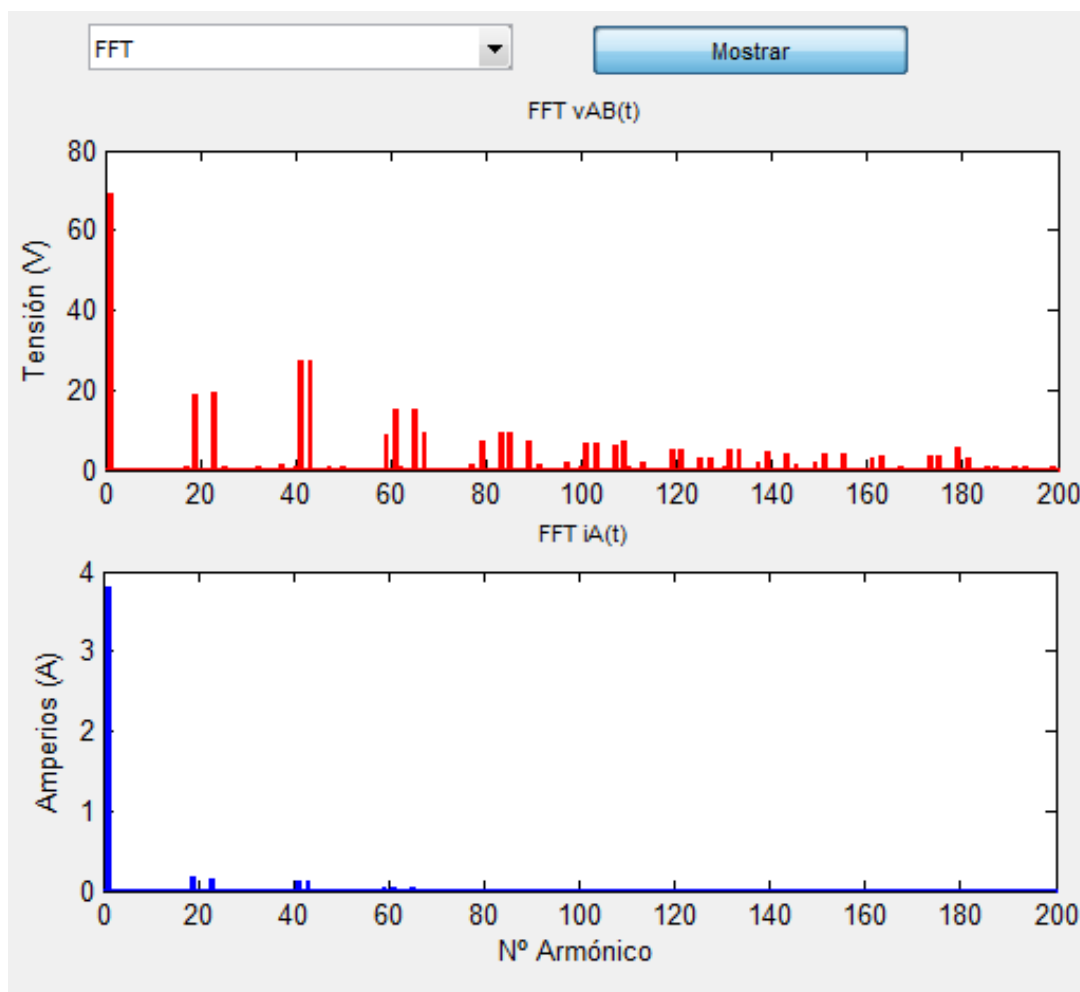


Figura 3.63: Armónicos de tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación PWM.

Una vez analizadas todas las opciones que se pueden visualizar en la ventana de representación se vuelve a la interfaz gráfica trifásica.

El último botón por analizar en esta ventana es el botón “**Representación VAB1=f(ma)**”, el cual, al ser pulsado, abrirá la ventana de la Figura 3.17. Al igual que ocurría en el caso de los inversores monofásicos, dentro de esta ventana existen dos opciones diferentes. La primera de ellas consiste en una imagen precalculada como la de la Figura 3.64 con unos parámetros ya definidos, debido a que este cálculo puede tardar cierto tiempo. La segunda opción consiste en realizar el cálculo con los valores de  $m_a$  inicial,  $m_a$  final y el paso deseado.

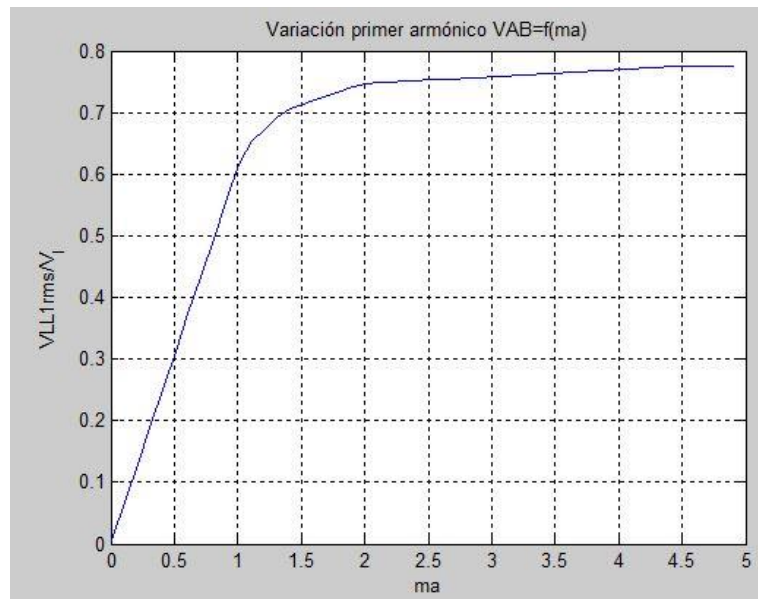


Figura 3.64: Imagen precalculada del barrido en modulación PWM.

Al igual que ocurría en la ventana de representación, el usuario debe haber realizado el cálculo inicialmente en la interfaz gráfica monofásica, ya que en caso de no haberlo hecho se ejecutará con los parámetros y modulación seleccionada con anterioridad.

### 3.3.2 Modulación PWM con inyección de secuencia cero: método inyección tercer armónico (THIPWM)

En esta nueva modulación, pulsando el botón “Abrir”, se abrirá el archivo .mdl mostrado en la Figura 3.50 a excepción del bloque de modulación llamado “Modulación PWM con inyección de tercer armónico”, el cual se puede observar en la Figura 3.65.

En esta técnica de modulación se suma a las señales de control el armónico  $v_{os}$ , lo que genera las señales  $v_{CONTROLA}'$ ,  $v_{CONTROLB}'$  y  $v_{CONTROLC}'$ .

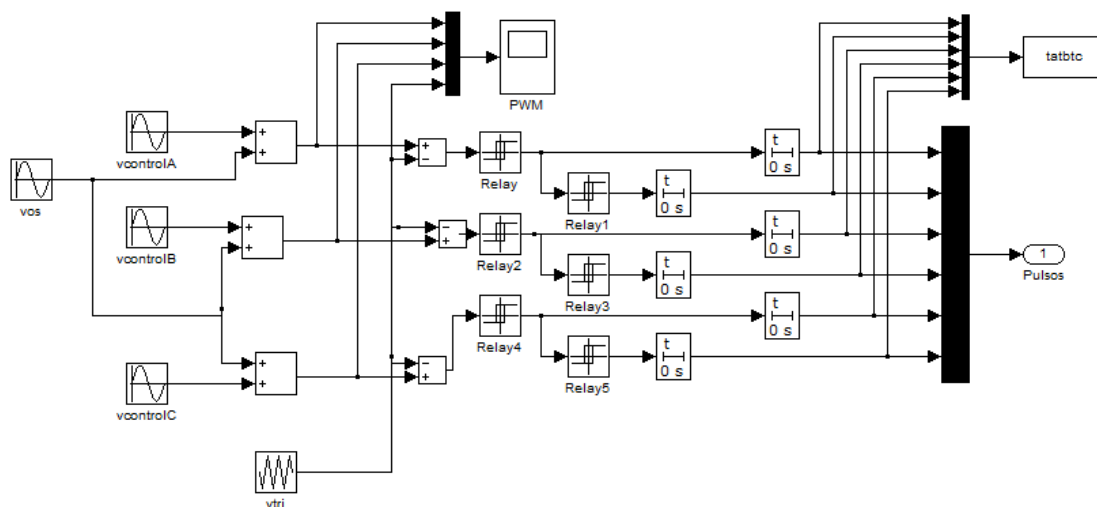


Figura 3.65: Circuito Simulink de modulación THIPWM.

Pulsando el botón “**Calcular**” se realiza la simulación, generando nuevamente los resultados de la simulación. Los resultados obtenidos con los valores por defecto pueden observarse en la Figura 3.66.

En la Figura 3.66 puede verse que, en la pestaña de “Introducir Valores”, hay un nuevo parámetro llamado Relación de Tercer Armónico. Este parámetro permite seleccionar entre  $m_a/4$  y  $m_a/6$ , ya que estas dos son las amplitudes del tercer armónico más utilizadas.

The screenshot shows a software interface for a THIPWM simulation. It is divided into several sections:

- Modulaciones:** Contains radio buttons for different modulation types:
  - Modulación PWM
  - Inyección Tercer Armónico** (selected)
  - Inyección Basada Portadora
  - Modulación Discontinua
  - Modulación Vectorial
- Circuit Diagram:** A schematic diagram of a three-phase inverter bridge with a load (R, L) and a neutral point (n). It shows the switching transistors (T<sub>A</sub><sup>+</sup>, T<sub>A</sub><sup>-</sup>, T<sub>B</sub><sup>+</sup>, T<sub>B</sub><sup>-</sup>, T<sub>C</sub><sup>+</sup>, T<sub>C</sub><sup>-</sup>) and diodes (D<sub>A</sub><sup>+</sup>, D<sub>A</sub><sup>-</sup>, D<sub>B</sub><sup>+</sup>, D<sub>B</sub><sup>-</sup>, D<sub>C</sub><sup>+</sup>, D<sub>C</sub><sup>-</sup>). Input parameters include V<sub>I</sub>, R<sub>I</sub>, i<sub>R<sub>I</sub></sub>, C<sub>I</sub>, i<sub>C<sub>I</sub></sub>, and N.
- Introducir Valores:** A table of input parameters:
 

Tensión Entrada V <sub>I</sub> (V)	100
Resistencia Entrada R <sub>I</sub> (Ohms)	0.1
Capacidad Entrada C <sub>I</sub> (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto t <sub>d</sub> (s)	0
Resistencia Salida R (Ohms)	10
Inductancia Salida L (H)	0.01
Frecuencia f <sub>1</sub> (Hz)	50
Modulación Amplitud m <sub>a</sub>	0.8
Modulación Frecuencia m <sub>f</sub>	21
Relación Tercer Armónico	ma/4
- Resultados:** A table of output results:
 

Corriente R <sub>I</sub> (A)	2.3627
Corriente C <sub>I</sub> (A)	1.39265
Potencia Entrada P <sub>I</sub> (W)	218.45
V <sub>AB1_rms</sub> (V)	48.8629
Delta_i <sub>A</sub> (A)	8.24837
Potencia Salida P <sub>o</sub> (W)	218.428
THD v <sub>AB</sub>	0.911858
THD i <sub>A</sub>	0.0717757
- Buttons:**
  - Abrir
  - Restaurar .MDL
  - Calcular
  - Borrar
  - Representación
  - Representación V<sub>AB1</sub>=f(m<sub>a</sub>)
  - Teoria
  - Videotutorial
  - Datos por defecto
  - Descargar en Placa

Figura 3.66: Cálculos con modulación THIPWM y datos por defecto.

El botón “**Representación**” abrirá la ventana de la Figura 3.54.

Al seleccionar “**PWM, TA+, TB+, TC+**” se obtiene las señales de modulación PWM así como el estado de los transistores superiores del inversor, tal y como se observa en la Figura 3.67.

En esta técnica de modulación las señales moduladoras no tienen los máximos en la zona central, sino que la señal es más aplanada, lo que permite permanecer en zona lineal para valores de  $m_a > 1$ .

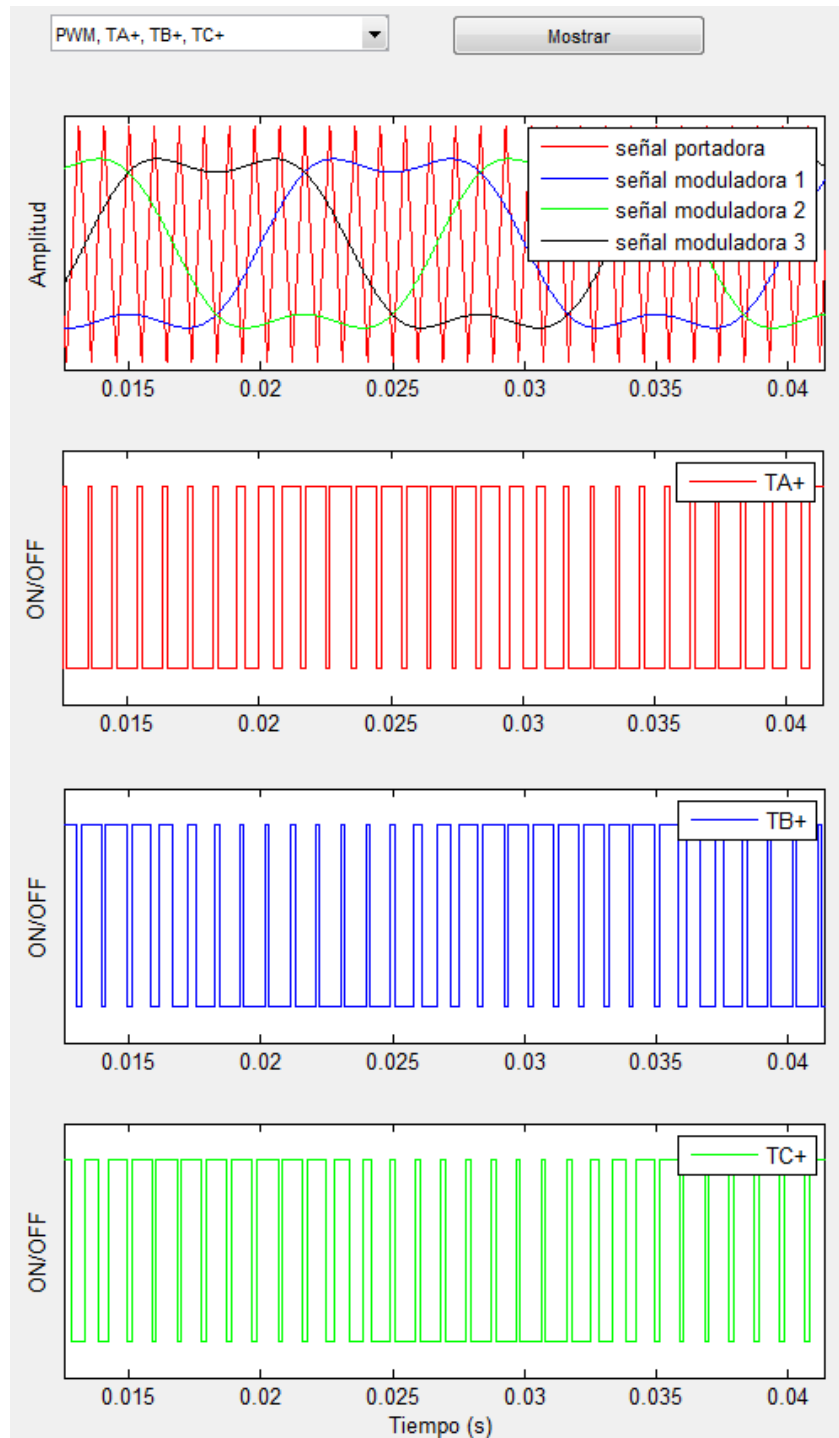


Figura 3.67: Señales PWM, TA+, TB+ y TC+ con modulación THIPWM.

Al pulsar sobre “**VAB, VBC, VCA**” se mostrarán las tensiones de línea del sistema, tal y como se muestra en la Figura 3.68.

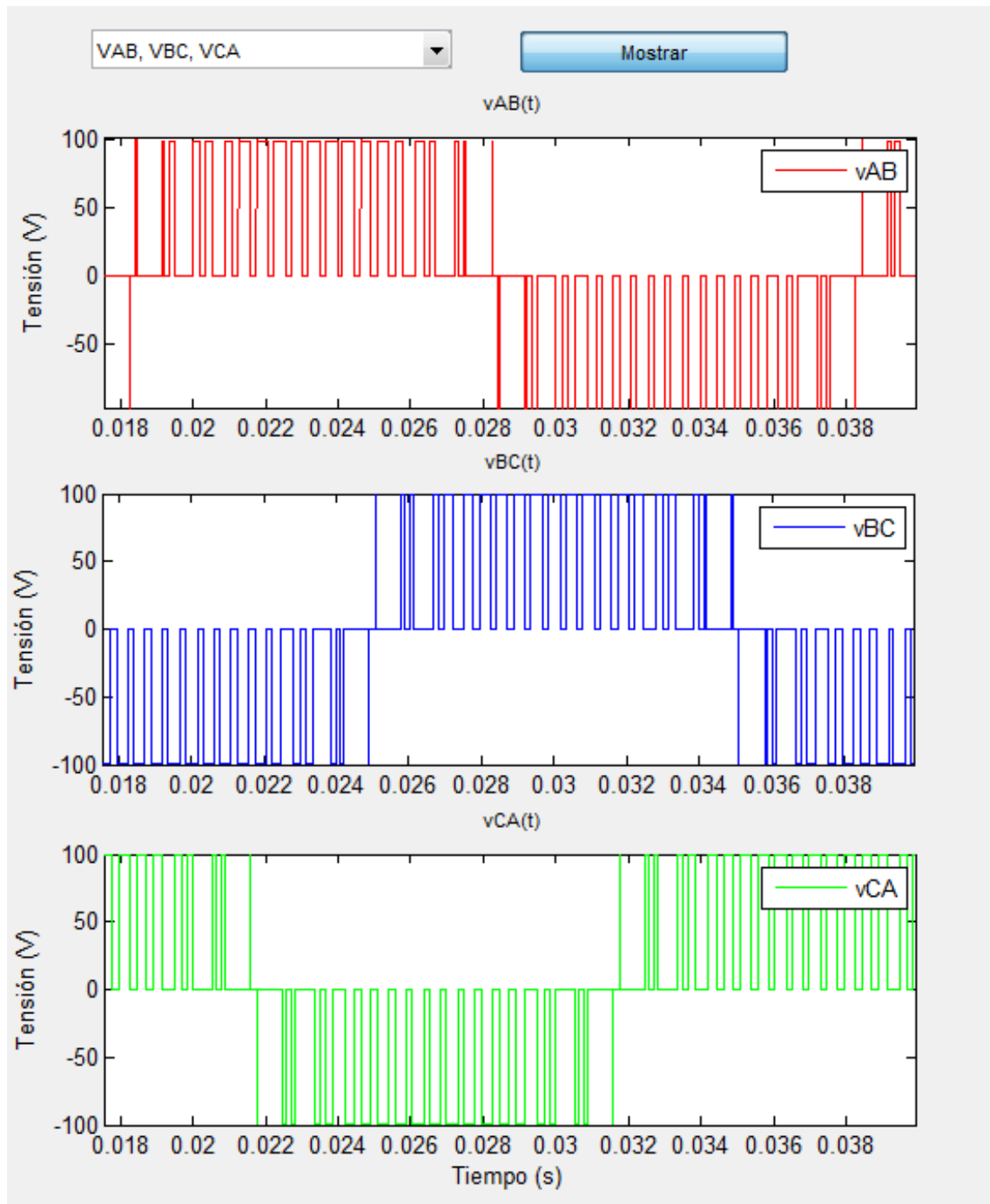


Figura 3.68: Tensiones  $v_{AB}$ ,  $v_{BC}$  y  $v_{CA}$  en modulación THIPWM.

Al pulsar sobre “VAN, VBN, VCN” se representarán las tensiones de cada fase con respecto a la entrada. Ver Figura 3.69.

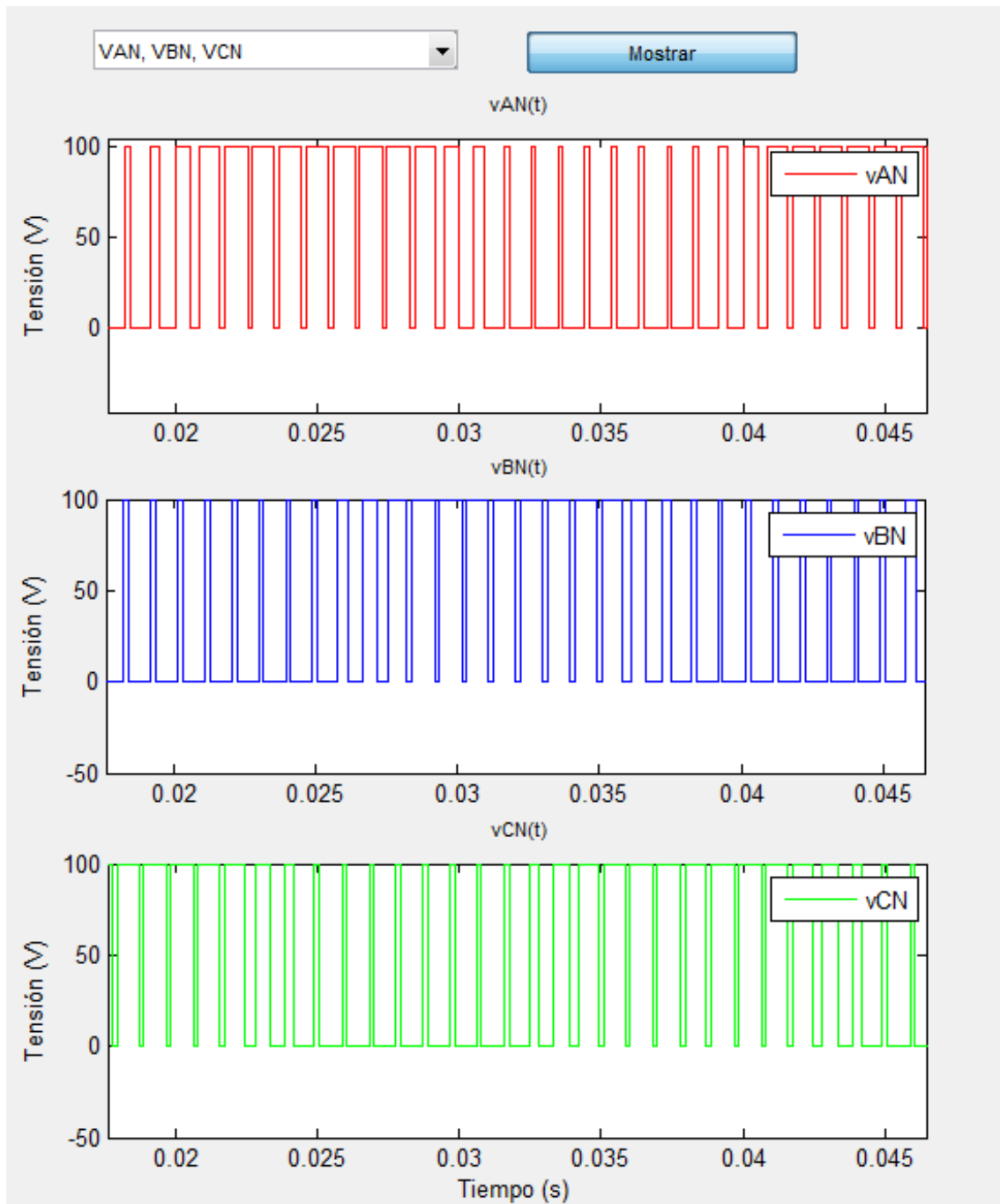


Figura 3.69: Tensiones  $v_{AN}$ ,  $v_{BN}$  y  $v_{CN}$  en modulación THIPWM.



Seleccionando sobre “**VAn, VBn, VCn**” se mostrarán las tensiones de cada fase. En la Figura 3.70 se pueden observar estas tensiones.

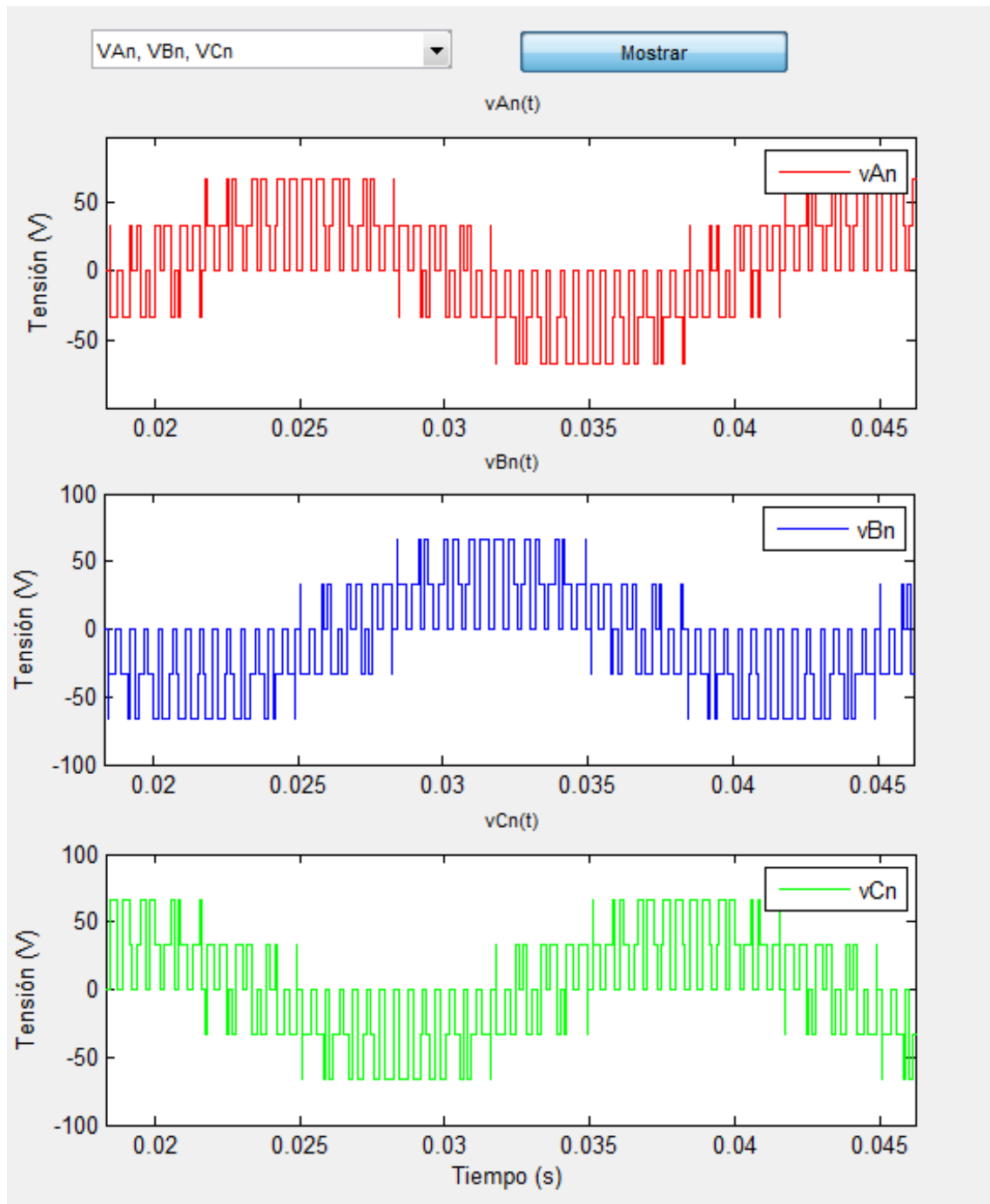


Figura 3.70: Tensiones  $v_{An}$ ,  $v_{Bn}$  y  $v_{Cn}$  en modulación THIPWM.

La siguiente opción que se puede seleccionar es “VAN, VAn, VNn”. En esta opción se visualizan las dos tensiones anteriores en la fase A además de la diferencia entre ambas. Estas tensiones se visualizan en la Figura 3.71.

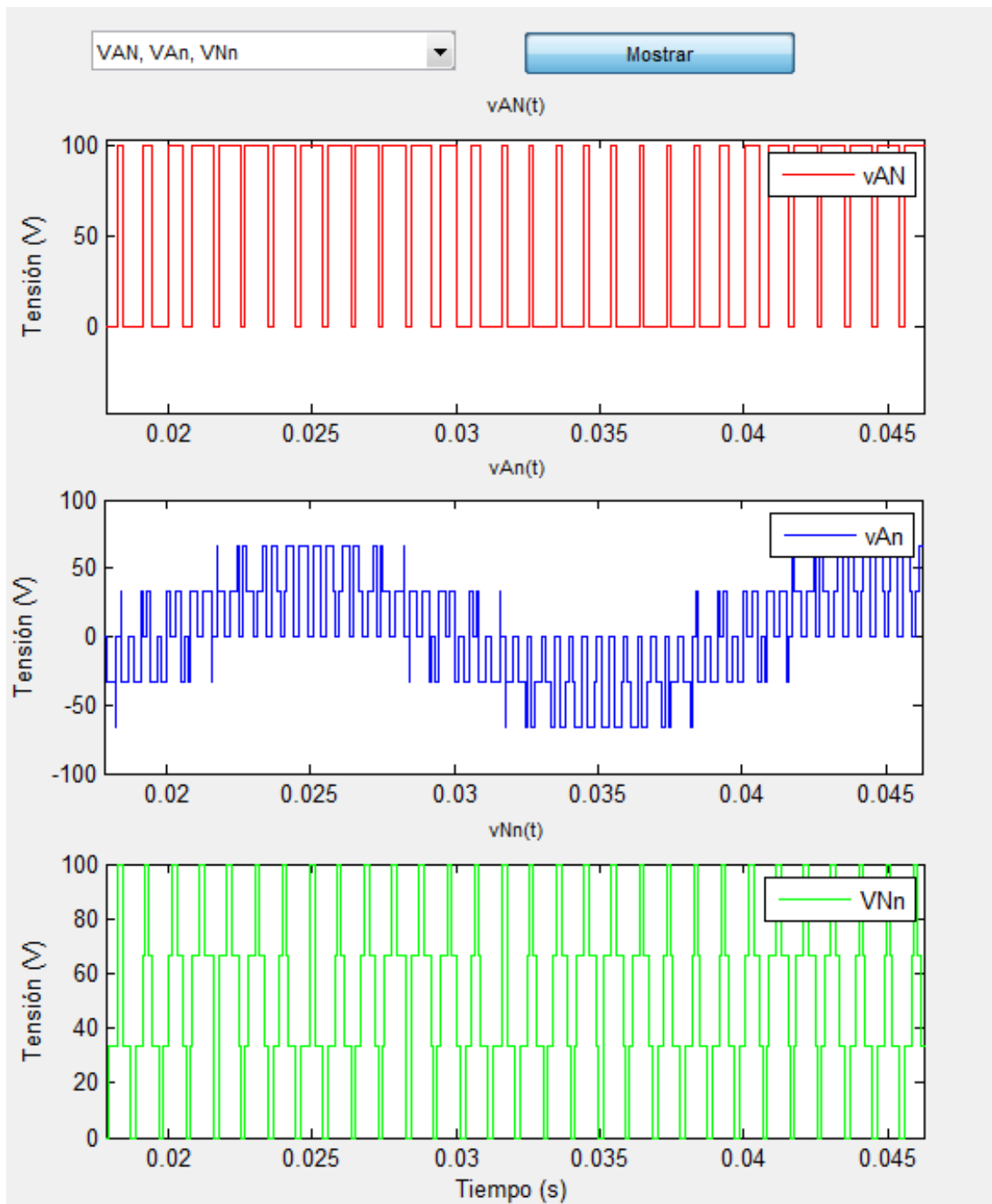


Figura 3.71: Tensiones  $v_{AN}$ ,  $v_{An}$  y  $v_{Nn}$  en modulación THIPWM.

Al finalizar de representar las diferentes tensiones del circuito, la siguiente opción posible es “**iA, iB, iC**” que representa las corrientes de fase de la salida del inversor. Estas corrientes se muestran en la Figura 3.72.

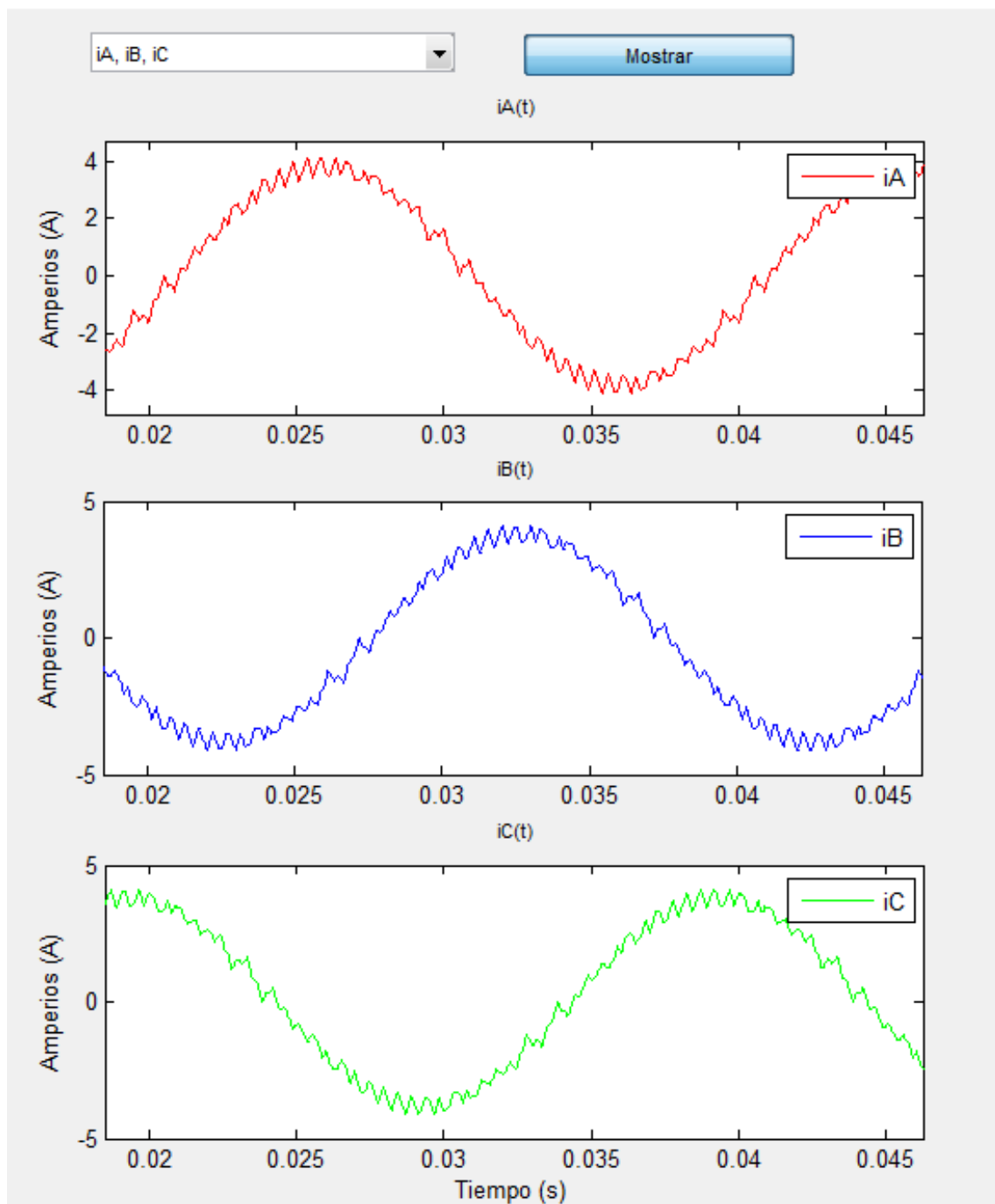


Figura 3.72: Corrientes de fase  $i_A$ ,  $i_B$  y  $i_C$  en modulación THIPWM.

La siguiente opción, “ **$i_A$ ,  $i_{A1}$ ,  $i_{Aripple}$** ”, muestra la corriente por la fase A, además de su primer armónico y del rizado de la señal, tal y como se muestra en la Figura 3.73.

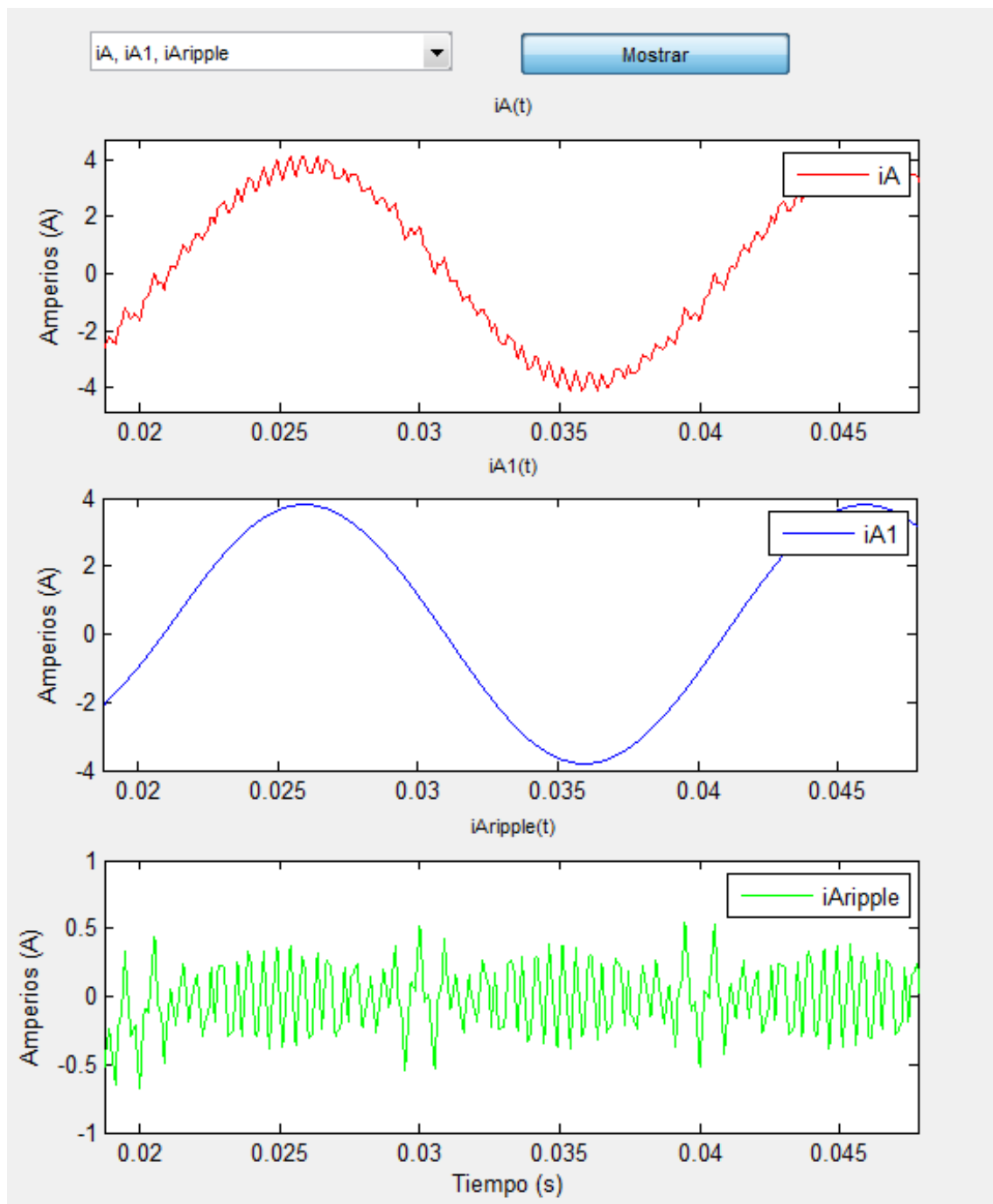


Figura 3.73: Corriente de fase  $i_A$ ,  $i_{A1}$  y  $i_{Aripple}$  en modulación THIPWM.

En la pestaña “ $i_I$ ,  $i_{C_I}$ ,  $i_{R_I}$ ,  $i_A$ ” se muestran las diferentes corrientes del circuito. En la Figura 3.74 se representan dichas corrientes.

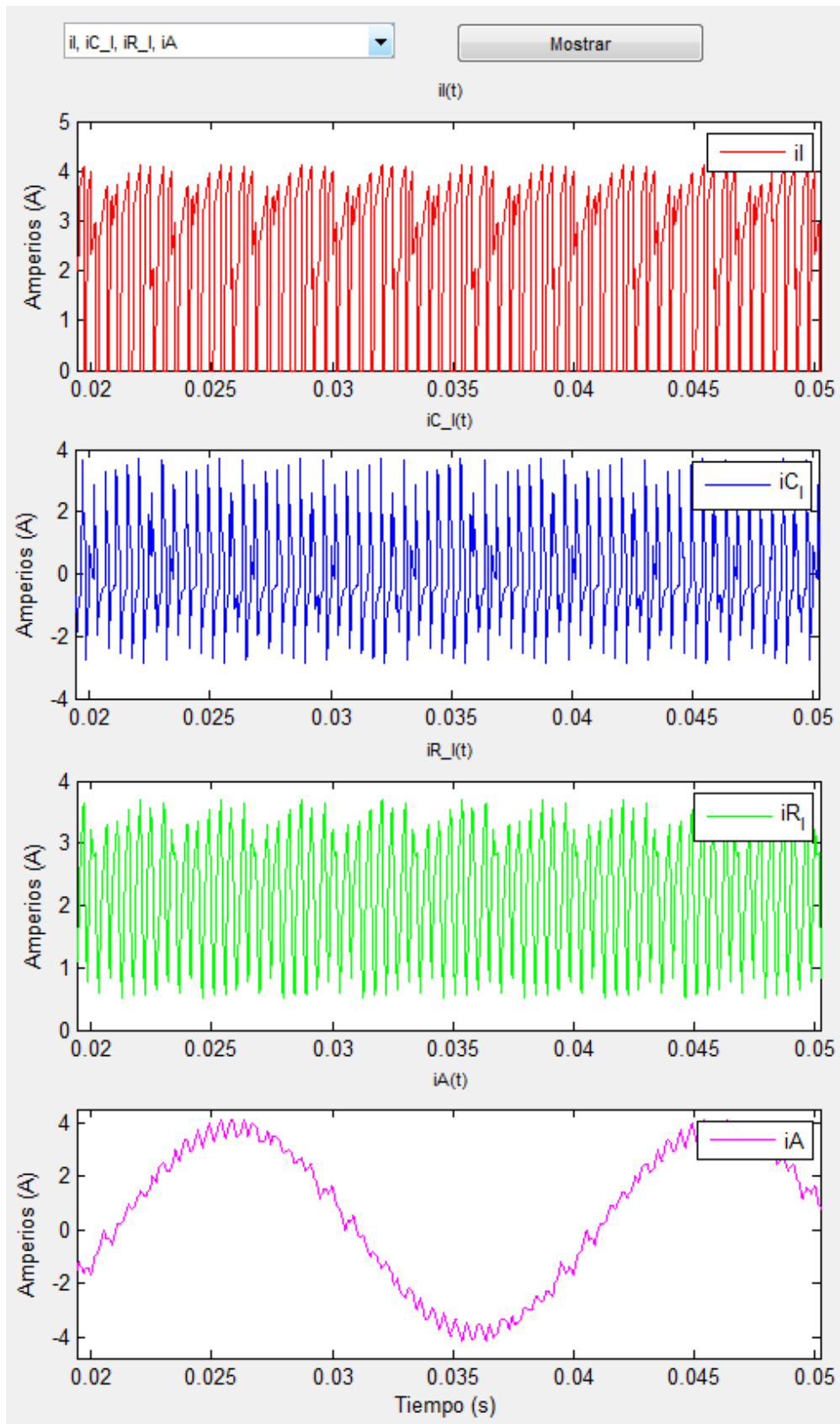


Figura 3.74: Corrientes  $i_I$ ,  $i_{C_I}$  y  $i_{R_I}$  e  $i_A$  en modulación THIPWM.

Por último, la pestaña “**FFT**” se muestra en la Figura 3.75, en la cual se puede ver el análisis de armónicos de la tensión de línea y de la corriente de fase.

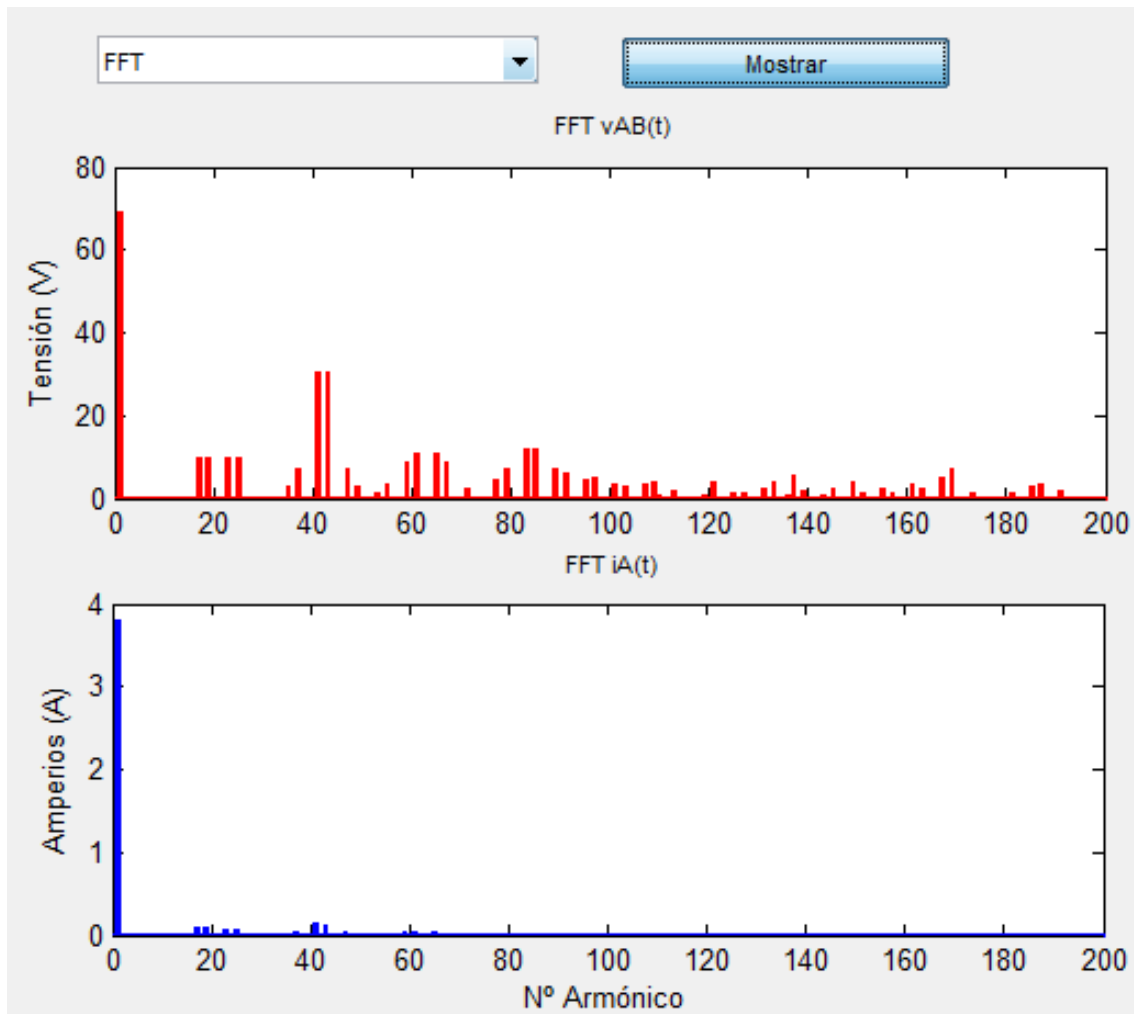


Figura 3.75: Armónicos de tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación THIPWM.

Una vez analizadas todas las opciones que se pueden visualizar en la ventana de representación se vuelve a la interfaz gráfica trifásica.

El último botón por analizar en esta ventana es el botón “**Representación VAB1=f(ma)**”, el cual, al ser pulsado, abrirá la ventana mostrada en la Figura 3.17.

Nuevamente hay una primera opción que consiste en una imagen precalculada como la de la Figura 3.76 con unos parámetros ya definidos, debido a que este cálculo puede tardar cierto tiempo. La segunda opción consiste en realizar el cálculo con los valores de  $m_a$  inicial,  $m_a$  final y el paso deseado.

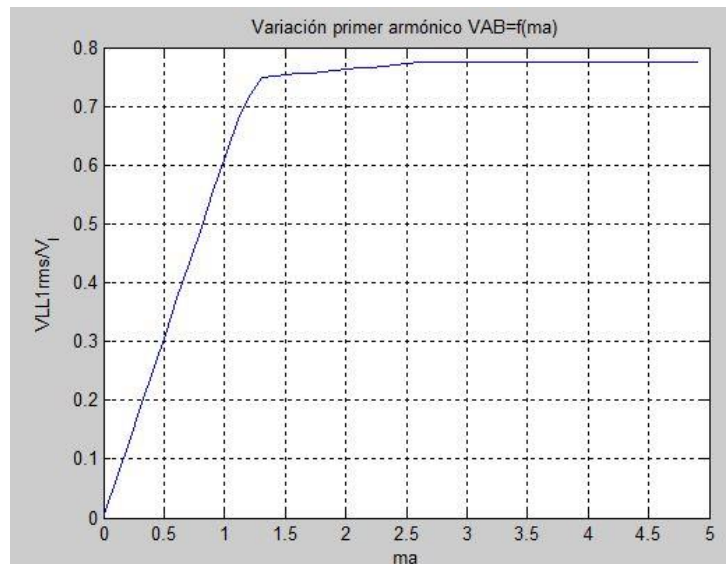


Figura 3.76: Imagen precalculada del barrido en modulación THIPWM.

En la Figura 3.76 se puede observar cómo la zona lineal del circuito se mantiene para valores de  $m_a > 1$ . En esta técnica de modulación  $m_a$  toma valores cercanos a 1.2 como límite de zona lineal.

### 3.3.3 Modulación PWM con inyección de secuencia cero: método PWM con vectores espaciales basada en portadora (CB-SVPWM)

En esta nueva modulación, pulsando el botón “Abrir”, se abrirá el archivo .mdl mostrado en la Figura 3.50 a excepción del bloque de modulación llamado “Modulación PWM con inyección de secuencia cero”, el cual se representa en la Figura 3.77. En esta técnica de modulación se calcula la señal inyectada  $v_{0s}$  a partir de los máximos y los mínimos de la señal.

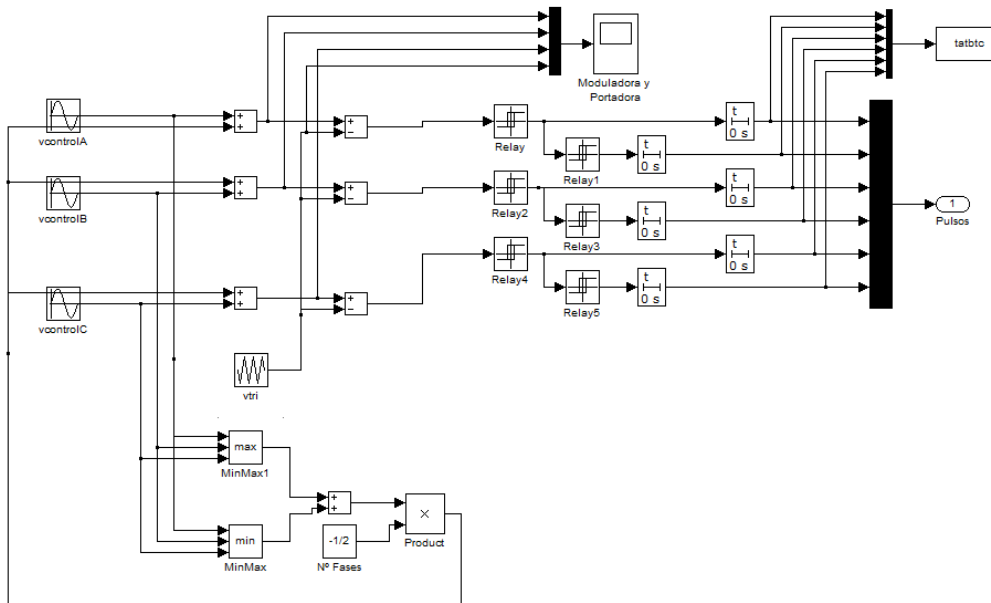


Figura 3.77: Circuito Simulink de modulación CB-SVPWM.

Al pulsar el botón “**Calcular**” nuevamente se tienen los resultados de la simulación. Los resultados realizados con valores por defecto son los mostrados en la Figura 3.78.

The interface is divided into several sections:

- Modulaciones:** Radio buttons for selecting modulation types:
  - Modulación PWM
  - Inyección Tercer Armónico
  - Inyección Basada Portadora
  - Modulación Discontinua
  - Modulación Vectorial
- Circuit Diagram:** A schematic of a three-phase inverter bridge with a DC source  $V_I$ , input resistance  $R_I$ , input capacitor  $C_I$ , and an RL load. The bridge consists of six transistors ( $T_A^+, T_A^-, T_B^+, T_B^-, T_C^+, T_C^-$ ) and six diodes ( $D_A^+, D_A^-, D_B^+, D_B^-, D_C^+, D_C^-$ ). The output voltage  $v_{AB}$  and current  $i_A$  are indicated.
- Introducir Valores:** Input fields for parameters:
 

Tensión Entrada $V_I$ (V)	100
Resistencia Entrada $R_I$ (Ohms)	0.1
Capacidad Entrada $C_I$ (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto $t_d$ (s)	0
Resistencia Salida $R$ (Ohms)	10
Inductancia Salida $L$ (H)	0.01
Frecuencia $f_1$ (Hz)	50
Modulación Amplitud $m_a$	0.8
Modulación Frecuencia $m_f$	21
- Resultados:** Output fields for simulation results:
 

Corriente $R_I$ (A)	2.3791
Corriente $C_I$ (A)	1.39649
Potencia Entrada $P_I$ (W)	220.073
$V_{AB1\_rms}$ (V)	49.0443
Delta $i_A$ (A)	8.30592
Potencia Salida $P_o$ (W)	220.051
THD $v_{AB}$	0.907853
THD $i_A$	0.0716954
- Buttons:**
  - Abrir
  - Restaurar .MDL
  - Calcular (highlighted)
  - Borrar
  - Datos por defecto
  - Descargar en Placa
  - Representación
  - Representación  $V_{AB1}=f(m_a)$
  - Teoria
  - Videotutorial

Figura 3.78: Cálculos con modulación CB-SVPWM y datos por defecto.



El botón “**Representación**” abrirá la ventana de la Figura 3.54.

Al seleccionar “**PWM, TA+, TB+, TC+**” se obtiene las señales de modulación PWM así como el estado de los transistores superiores del inversor, tal y como se observa en la Figura 3.79.

Al igual que ocurren en la modulación PWM con inyección de tercer armónico, las señales moduladoras tienen los picos debidos al tercer armónico, lo que permite permanecer en zona lineal para valores de  $m_a > 1$ .

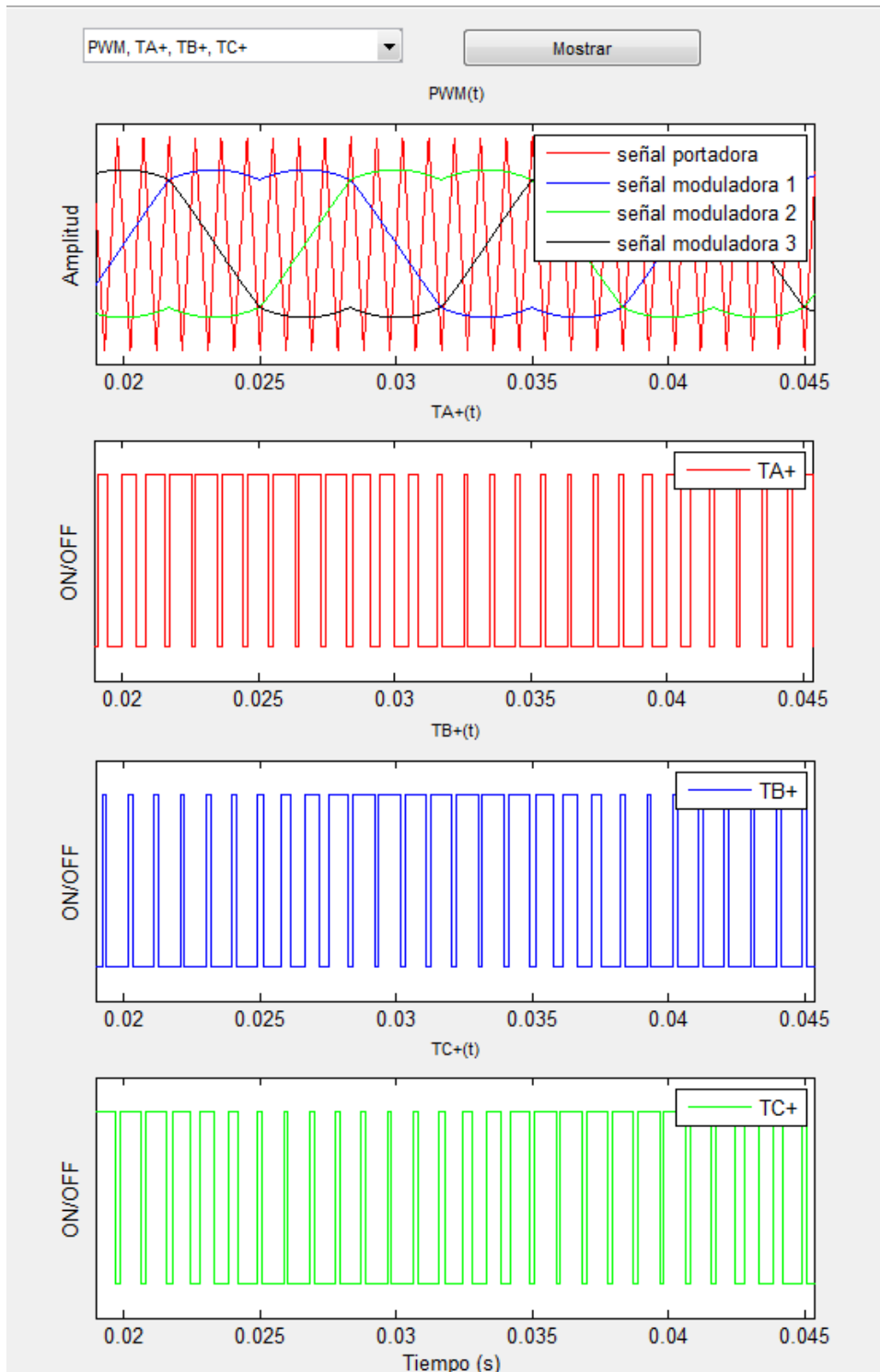


Figura 3.79: Señales PWM, TA+, TB+ y TC+ con modulación CB-SVPWM.

Pulsando sobre “**VAB, VBC, VCA**” se mostrarán las tensiones de línea del sistema, tal y como se muestra en la Figura 3.80.

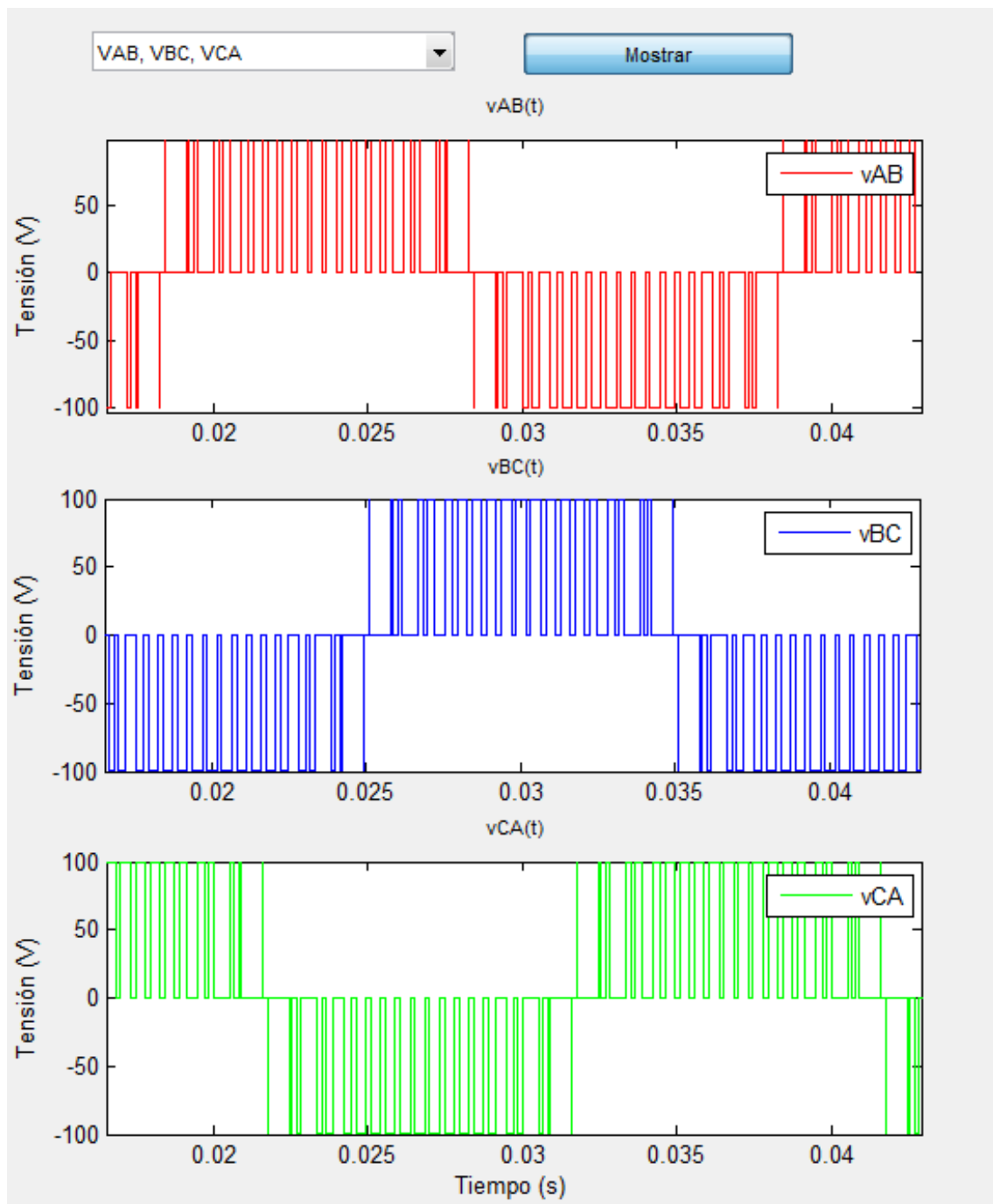


Figura 3.80: Tensiones  $v_{AB}$ ,  $v_{BC}$  y  $v_{CA}$  en modulación CB-SVPWM.

Al pulsar sobre “VAN, VBN, VCN” se mostrarán las tensiones de cada fase con respecto a la entrada. Una imagen de las mismas se puede ver en la Figura 3.81.

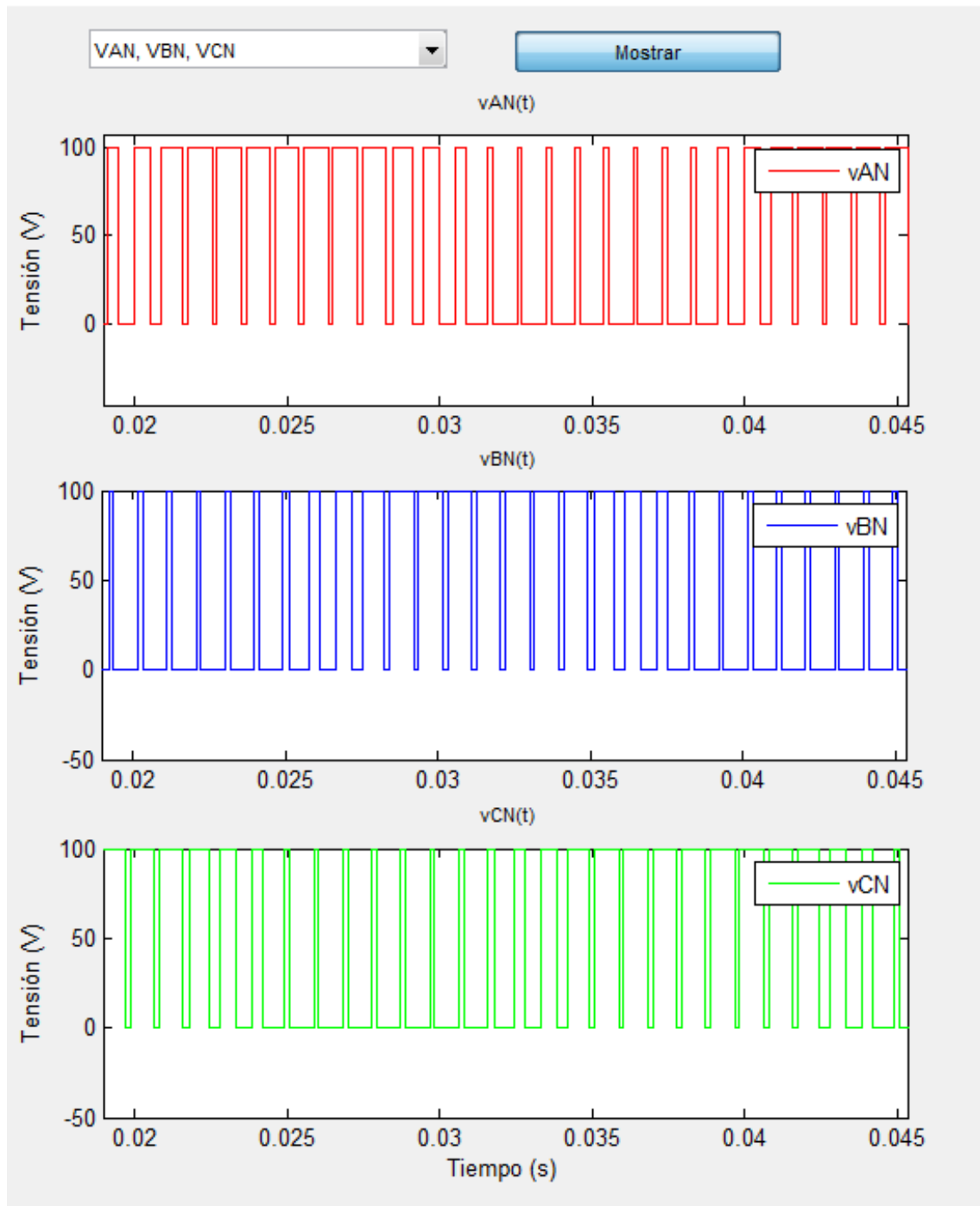


Figura 3.81: Tensiones  $v_{AN}$ ,  $v_{BN}$  y  $v_{CN}$  en modulación CB-SVPWM.

Al seleccionar sobre la opción “**VAn, VBn, VCn**” se mostrarán las tensiones de cada fase. En la Figura 3.82 se representan estas tensiones.

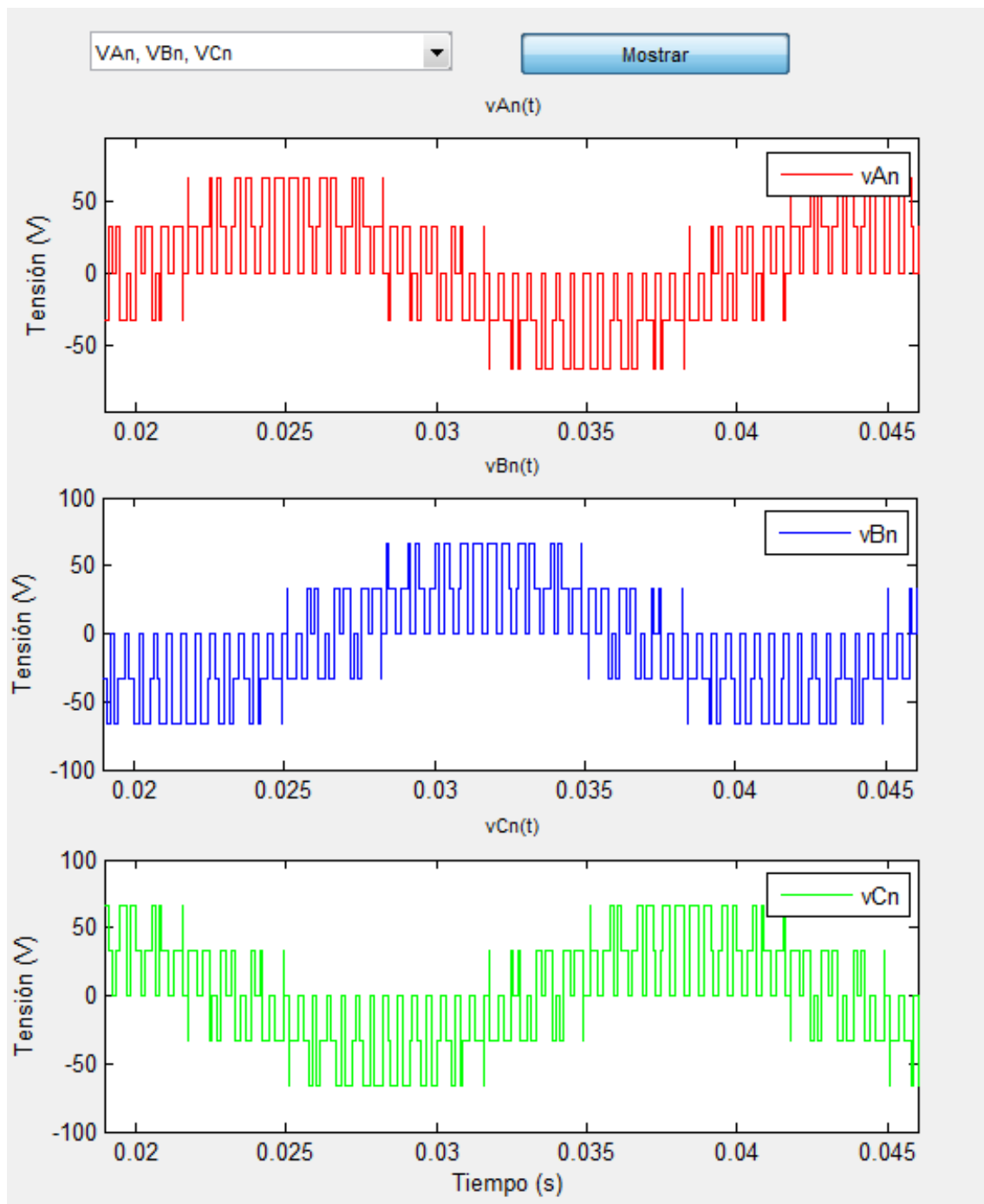


Figura 3.82: Tensiones  $v_{An}$ ,  $v_{Bn}$  y  $v_{Cn}$  en modulación CB-SVPWM.

Nuevamente, la última opción relacionada sobre representación de tensiones es “VAN, VAn, VNn”. En esta opción se visualizan las dos tensiones anteriores en la fase A además de la diferencia entre ambas. Ver Figura 3.83.

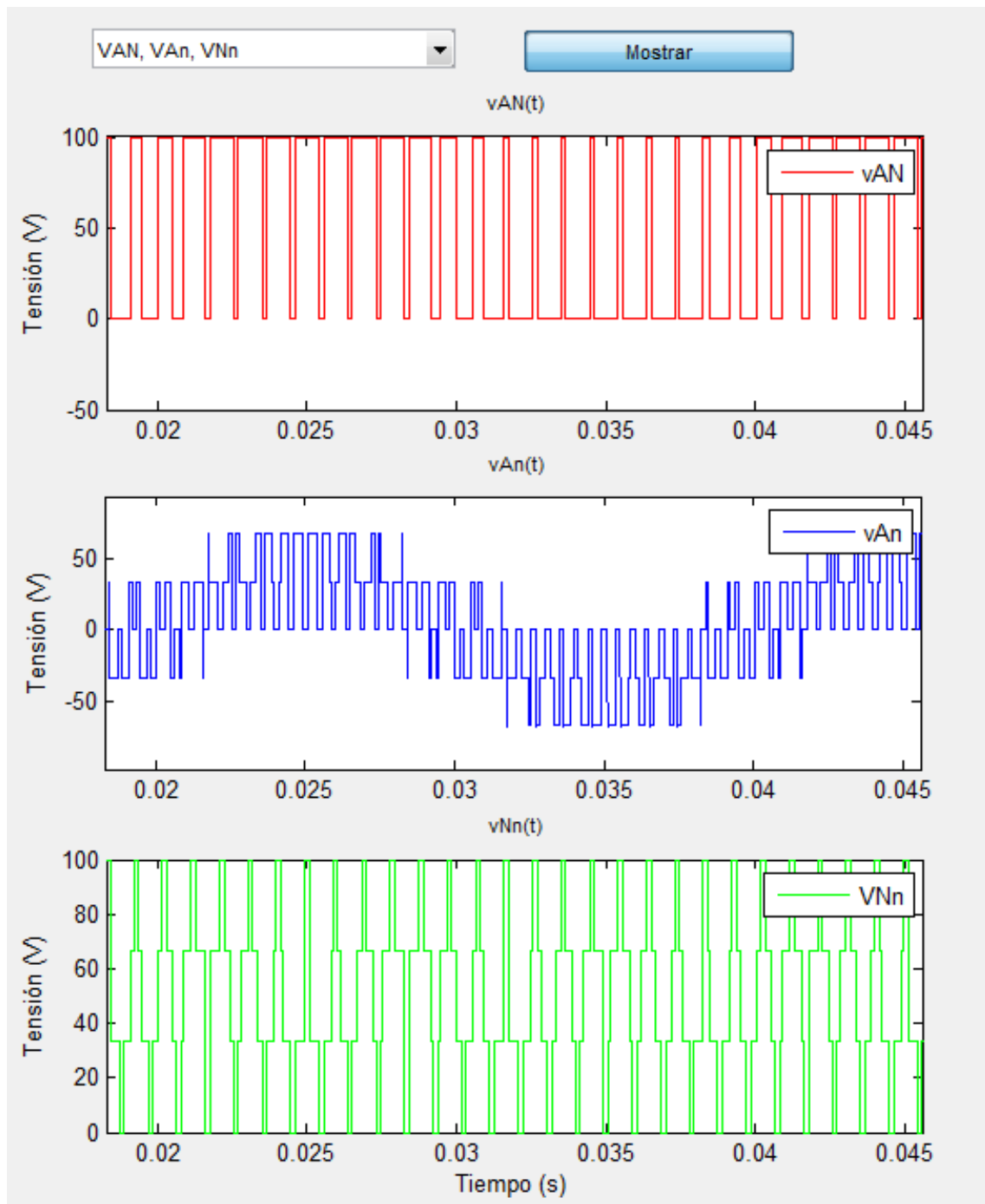


Figura 3.83: Tensiones  $v_{AN}$ ,  $v_{An}$  y  $v_{Nn}$  en modulación CB-SVPWM.

Al terminar de mostrar las diferentes tensiones del circuito, la siguiente opción, “**iA, iB, iC**”, muestra las corrientes de fase de la salida del inversor. Estas corrientes se observan en la Figura 3.84.

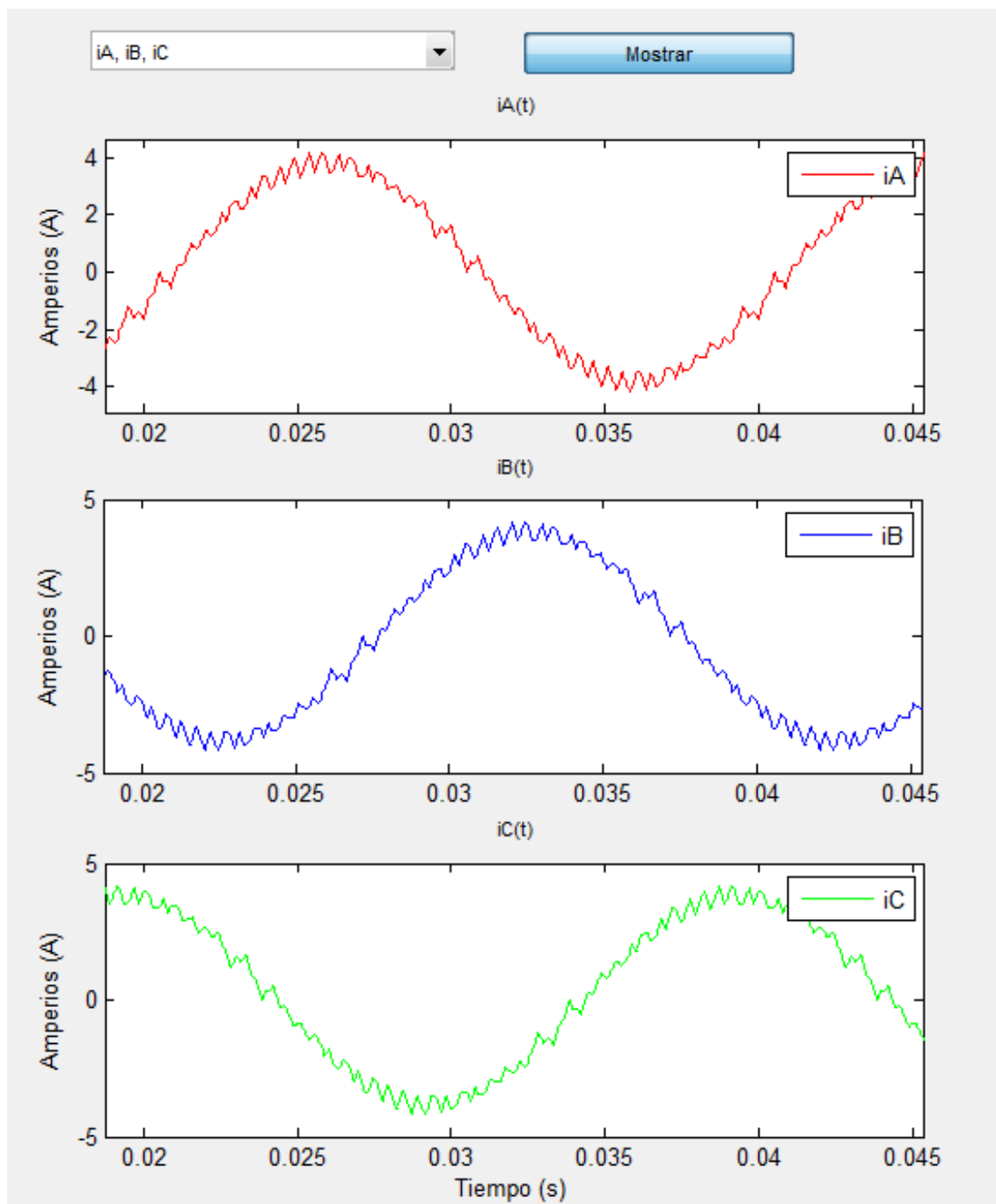


Figura 3.84: Corrientes de fase  $i_A$ ,  $i_B$  y  $i_C$  en modulación CB-SVPWM.

La siguiente opción, “**iA, iA1, iAripple**”, muestra la corriente por la fase A, además de su primer armónico y del rizado de la señal, tal y como se muestra en la Figura 3.85.

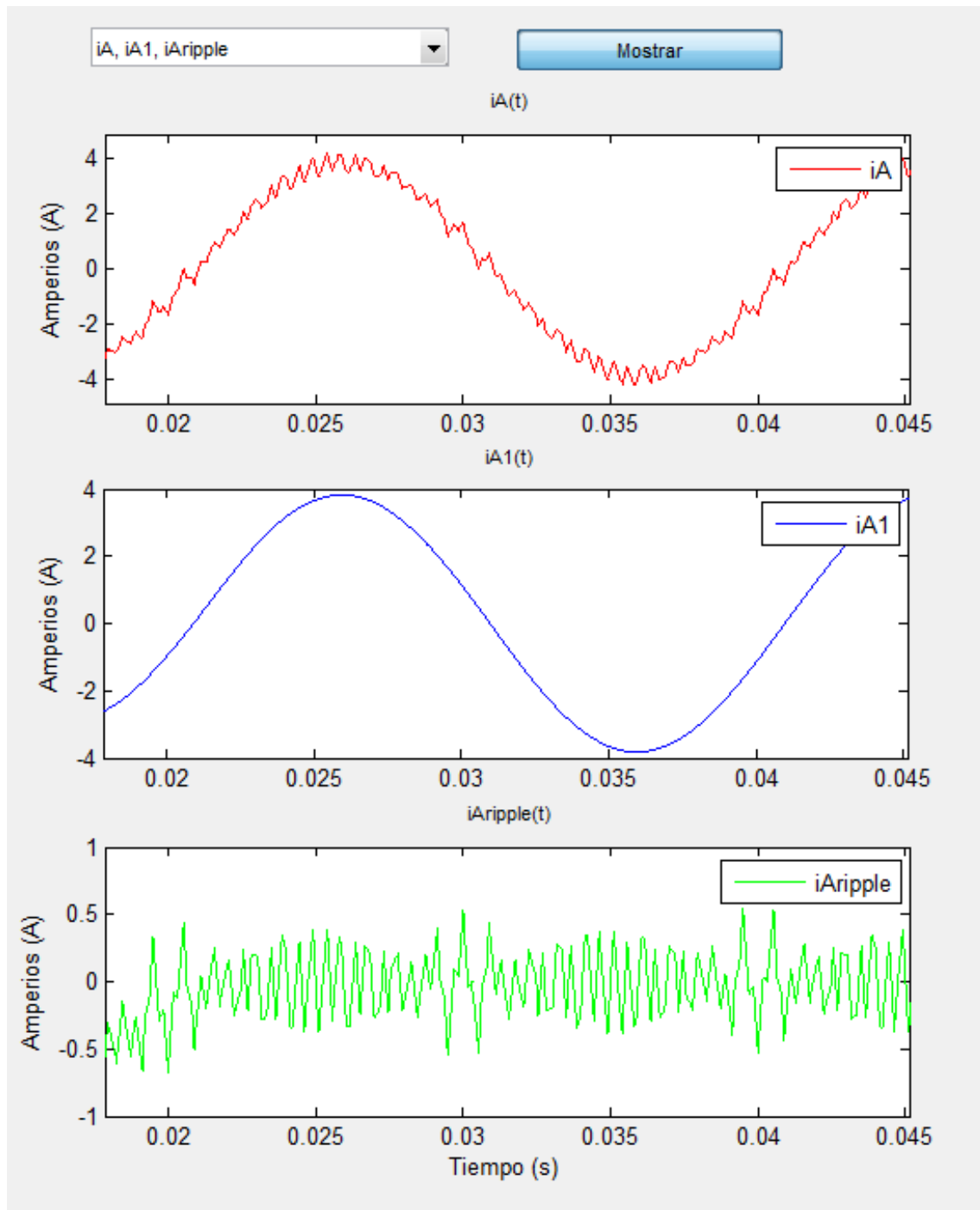


Figura 3.85: Corriente de fase  $i_A$ ,  $i_{A1}$  y  $i_{Aripple}$  en modulación CB-SVPWM.

Al igual que ocurría en el caso de inversores monofásicos, en la pestaña “**iI, iC\_I, iR\_I, iA**” se muestran las diferentes corrientes del circuito. En la Figura 3.86 se visualizan dichas corrientes.

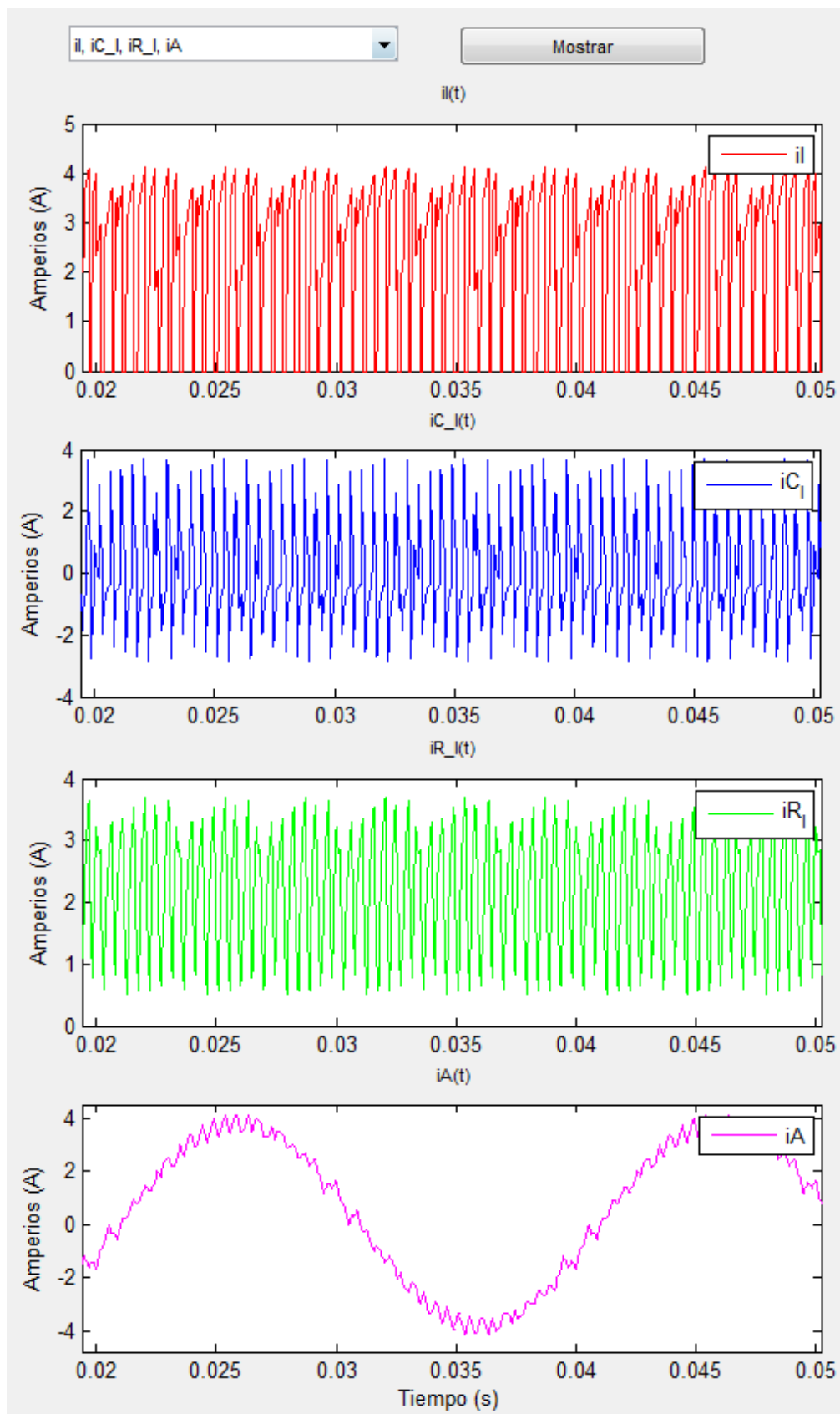


Figura 3.86: Corrientes  $i_I$ ,  $i_{C_I}$  y  $i_{R_I}$  e  $i_A$  en modulación CB-SVPWM.



Por último, en la pestaña “**FFT**”, que se muestra en la Figura 3.87, se puede ver el análisis de armónicos de la tensión de línea y de la corriente de fase.

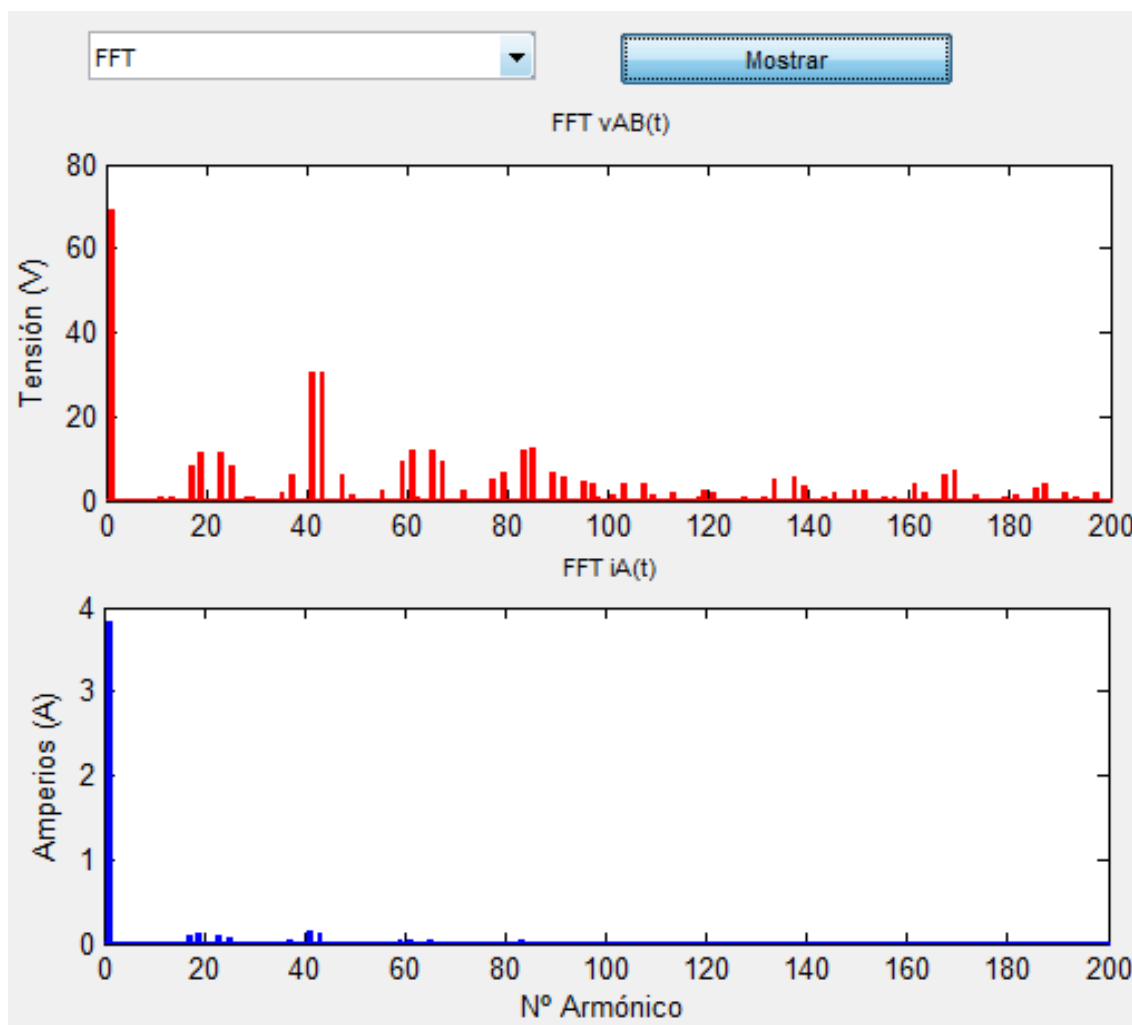


Figura 3.87: Armónicos tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación CB-SVPWM.

Una vez analizadas todas las opciones que se pueden visualizar en la ventana de representación se vuelve a la interfaz gráfica trifásica.

El último botón por analizar en esta ventana es el botón “**Representación  $V_{AB1}=f(m_a)$** ”, el cual, al ser pulsado, abrirá la ventana mostrada en la Figura 3.17.

Nuevamente se tiene una primera opción que consiste en una imagen precalculada como la de la Figura 3.88. La segunda opción consiste en realizar el cálculo con los valores de  $m_a$  inicial,  $m_a$  final y el paso deseado.

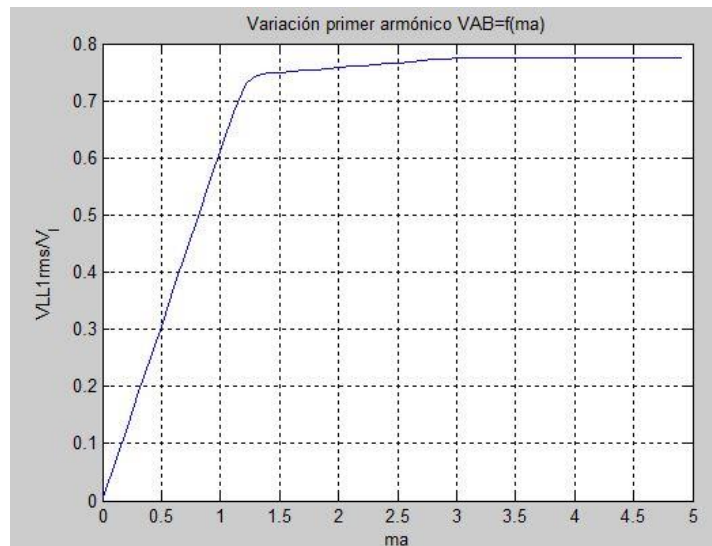


Figura 3.88: Imagen precalculada del barrido en modulación CB-SVPWM.

En la Figura 3.88 se puede observar cómo la zona lineal del circuito se mantiene para valores de  $m_a > 1$ . En esta técnica de modulación  $m_a$  llega a valer 1.2 y el inversor sigue trabajando en zona lineal, por lo que se consigue aumentar esta zona de trabajo.

### 3.3.4 Modulación PWM con inyección de secuencia cero: método discontinuo (DPWM)

En la opción modulación PWM discontinua, pulsando el botón “Abrir”, se abrirá el mismo archivo .mdl que se abría en el resto de modulaciones PWM, tal y como se muestra en la Figura 3.50, a excepción del bloque de modulación llamado “Modulación PWM discontinua” mostrado en la Figura 3.89. En esta técnica de modulación, el estado de los transistores se calcula comparando las señales de control con el valor máximo de la señal triangular, seleccionando la señal a comparar con el bloque *Multiport Switch*. El bloque *Multiport Switch* selecciona la señal a comparar según los ángulos para los cuales cada señal es mayor en valor absoluto.

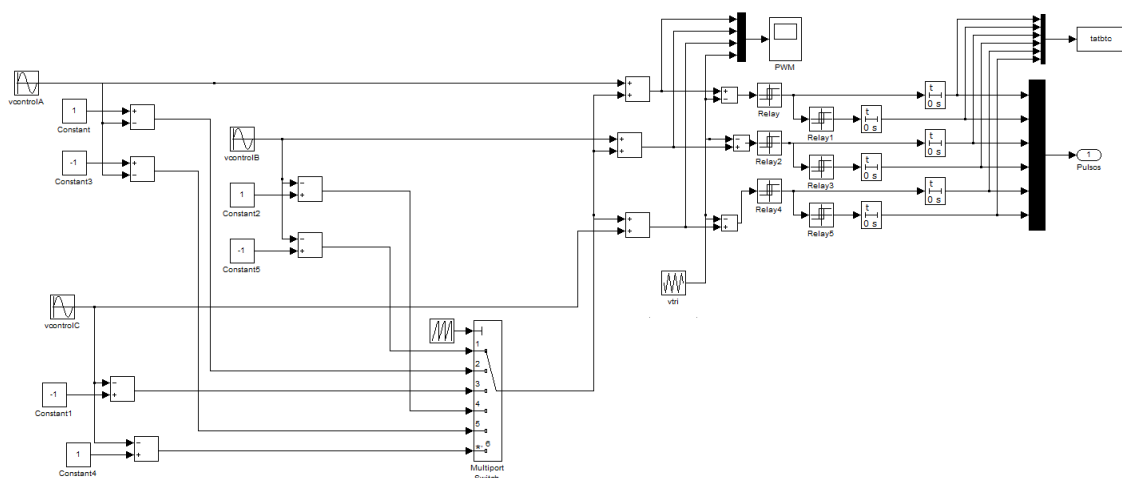


Figura 3.89: Circuito Simulink de modulación DPWM.

Al pulsar el botón “**Calcular**” se generan nuevamente los resultados de la simulación. En la Figura 3.90 se muestran estos resultados calculados con los valores por defecto.

The interface is divided into several sections:

- Modulaciones:**
  - Modulación PWM
  - Inyección Tercer Armónico
  - Inyección Basada Portadora
  - Modulación Discontinua
  - Modulación Vectorial
- Circuit Diagram:** Shows a three-phase inverter bridge with a DC source  $V_I$ , resistor  $R_I$ , and capacitor  $C_I$ . The bridge consists of six transistors ( $T_A^+, T_A^-, T_B^+, T_B^-, T_C^+, T_C^-$ ) and six diodes ( $D_A^+, D_A^-, D_B^+, D_B^-, D_C^+, D_C^-$ ). The output is connected to a load with resistor  $R$  and inductor  $L$ . Currents  $i_{R_I}$ ,  $i_{C_I}$ , and  $i_A$  are indicated.
- Introducir Valores:**

Tensión Entrada $V_I$ (V)	100
Resistencia Entrada $R_I$ (Ohms)	0.1
Capacidad Entrada $C_I$ (F)	0.001
Ron puente (Ohms)	0.001
Tiempo Muerto $t_d$ (s)	0
Resistencia Salida $R$ (Ohms)	10
Inductancia Salida $L$ (H)	0.01
Frecuencia $f_1$ (Hz)	50
Modulación Amplitud $m_a$	0.8
Modulación Frecuencia $m_f$	21
- Resultados:**

Corriente $R_I$ (A)	2.83787
Corriente $C_I$ (A)	1.08379
Potencia Entrada $P_I$ (W)	249.184
$V_{AB1\_rms}$ (V)	51.9411
Delta $i_A$ (A)	9.02127
Potencia Salida $P_o$ (W)	249.159
THD $v_{AB}$	0.855185
THD $i_A$	0.121225
- Buttons:**
  - Abrir
  - Restaurar .MDL
  - Calcular
  - Borrar
  - Datos por defecto
  - Descargar en Placa
  - Representación
  - Representación  $V_{AB1}=f(m_a)$
  - Teoria
  - Videotutorial

Figura 3.90: Cálculos con modulación DPWM y datos por defecto.

El botón “**Representación**” abrirá la ventana mostrada en la Figura 3.54.

Al seleccionar “**PWM, TA+, TB+, TC+**” se obtienen las señales de modulación PWM así como el estado de los transistores superiores del inversor, tal y como se observa en la Figura 3.91.

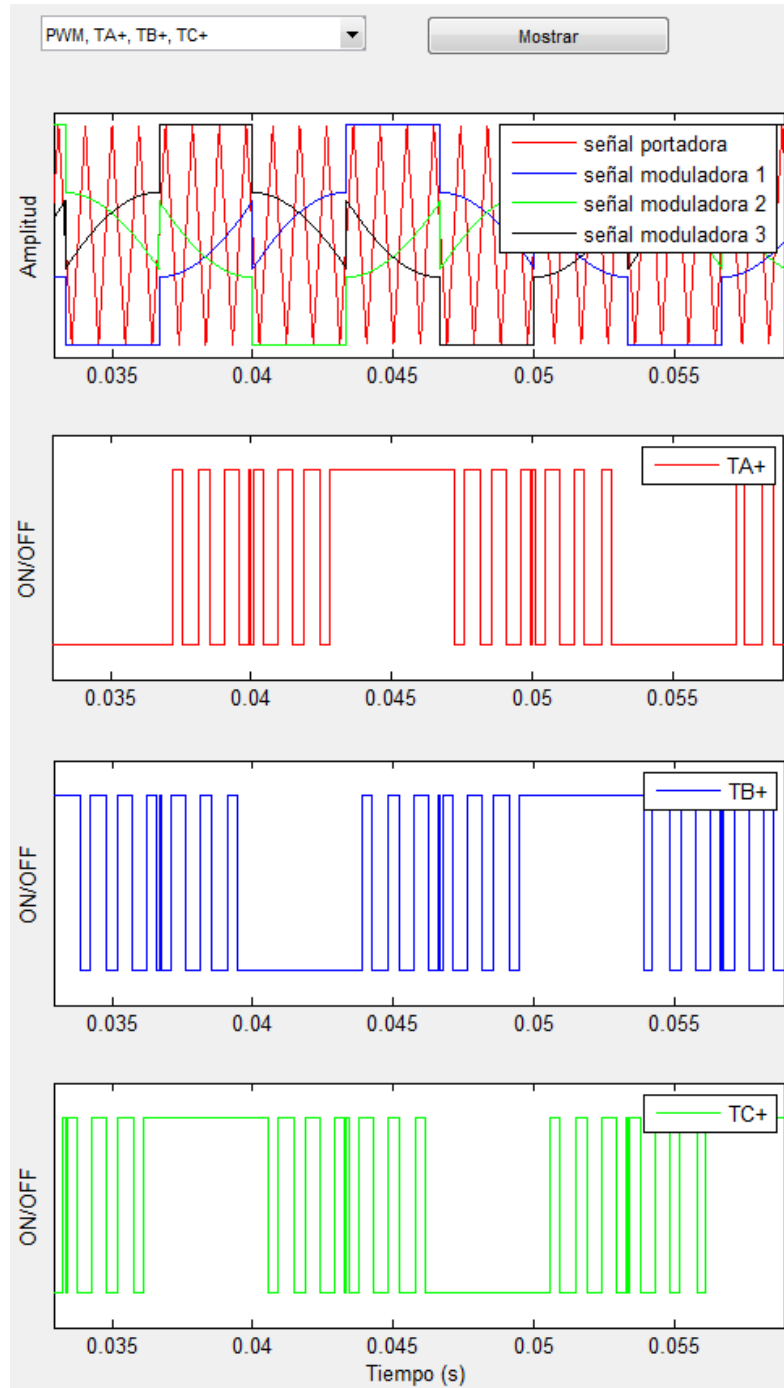


Figura 3.91: Señales PWM, TA+, TB+ y TC+ con modulación DPWM.

En la Figura 3.91 se puede apreciar el motivo del nombre de modulación PWM discontinua, ya que trabajando en zona lineal hay intervalos de tiempo en los cuales se satura la señal.

Pulsando sobre “**VAB, VBC, VCA**” se representan las tensiones de línea del sistema, tal y como se muestra en la Figura 3.92.

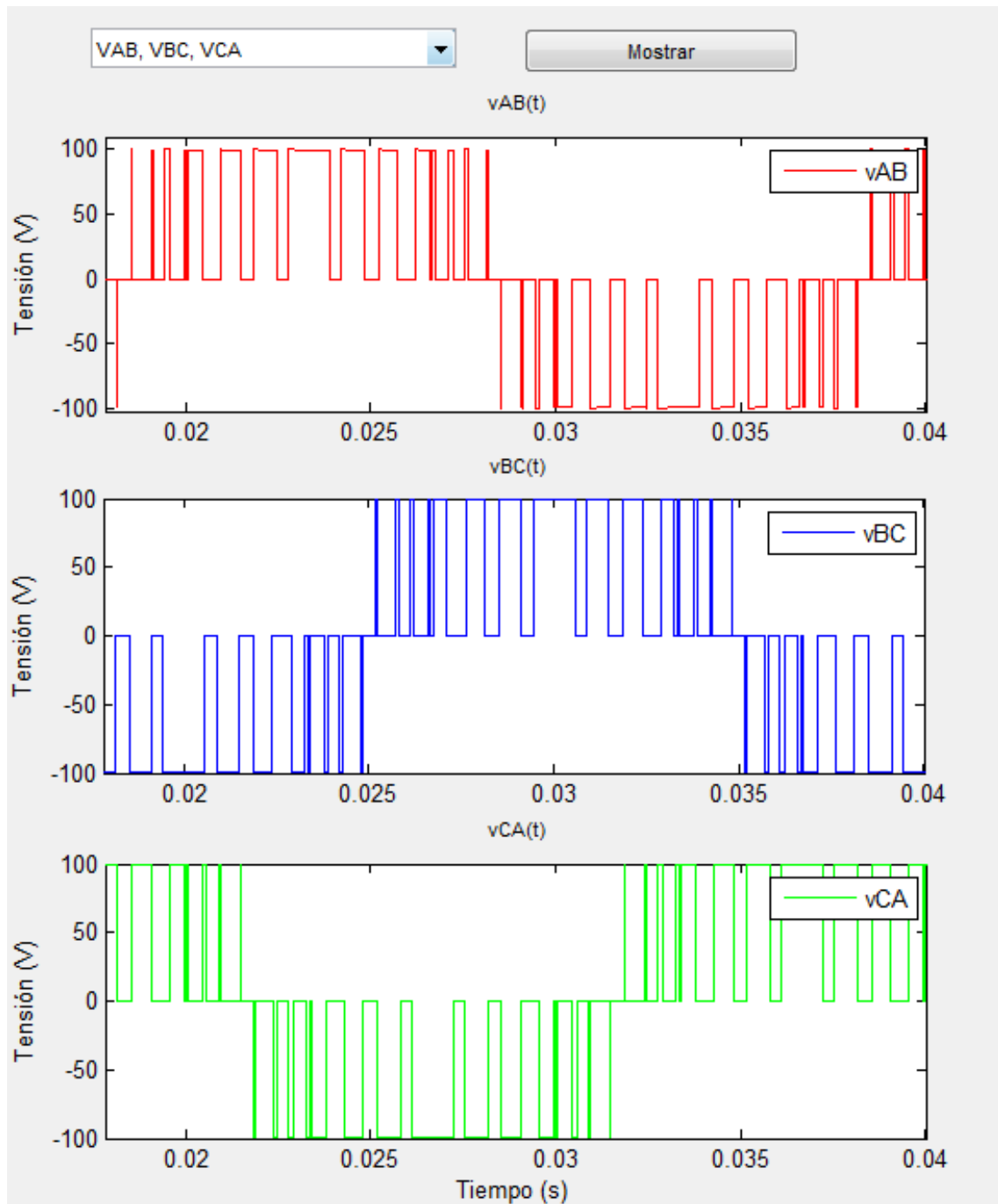


Figura 3.92: Tensiones  $v_{AB}$ ,  $v_{BC}$  y  $v_{CA}$  en modulación DPWM.

Al pulsar sobre “VAN, VBN, VCN” se mostrarán las tensiones de cada fase con respecto a la entrada. Ver Figura 3.93.

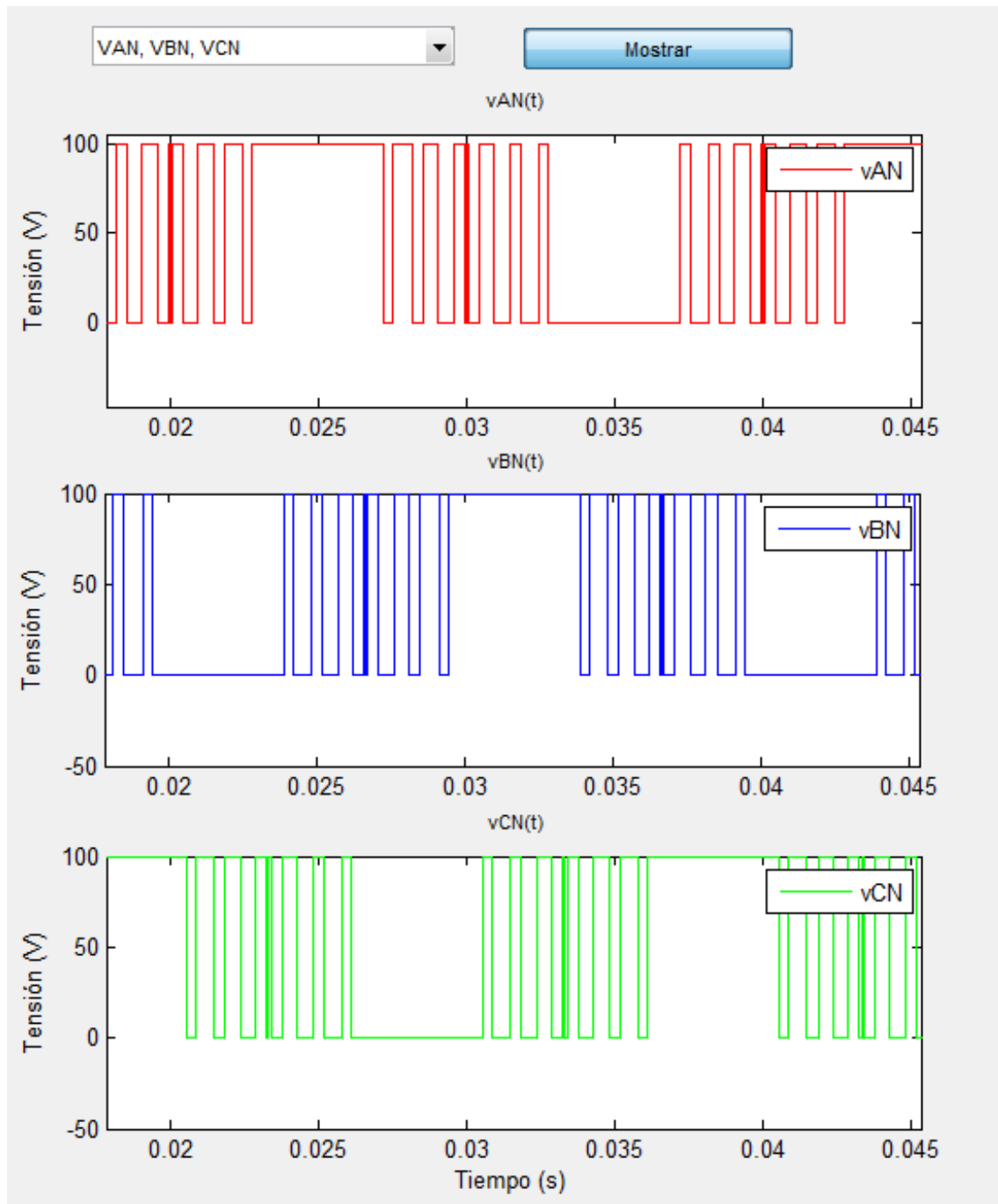


Figura 3.93: Tensiones  $v_{AN}$ ,  $v_{BN}$  y  $v_{CN}$  en modulación DPWM.

Al seleccionar sobre la opción “ $V_{An}$ ,  $V_{Bn}$ ,  $V_{Cn}$ ” se representan las tensiones de cada fase. Pueden observarse en la Figura 3.94.

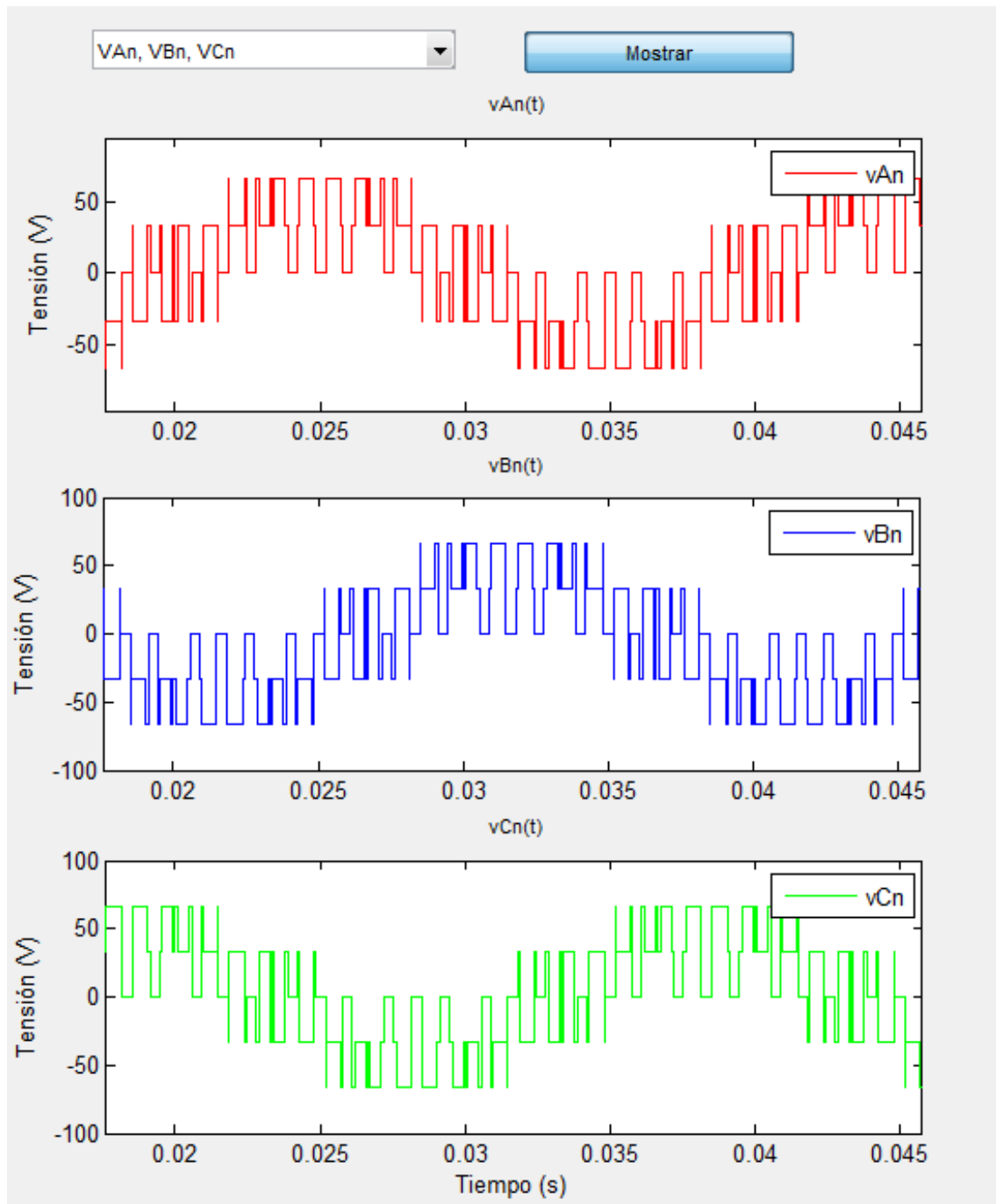


Figura 3.94: Tensiones  $v_{An}$ ,  $v_{Bn}$  y  $v_{Cn}$  en modulación DPWM.

Nuevamente, la última opción relacionada sobre representación de tensiones es “**VAN, VAn, VNn**”. En esta opción se visualizan las dos tensiones anteriores en la fase A además de la diferencia entre ambas, las cuales se visualizan en la Figura 3.95.

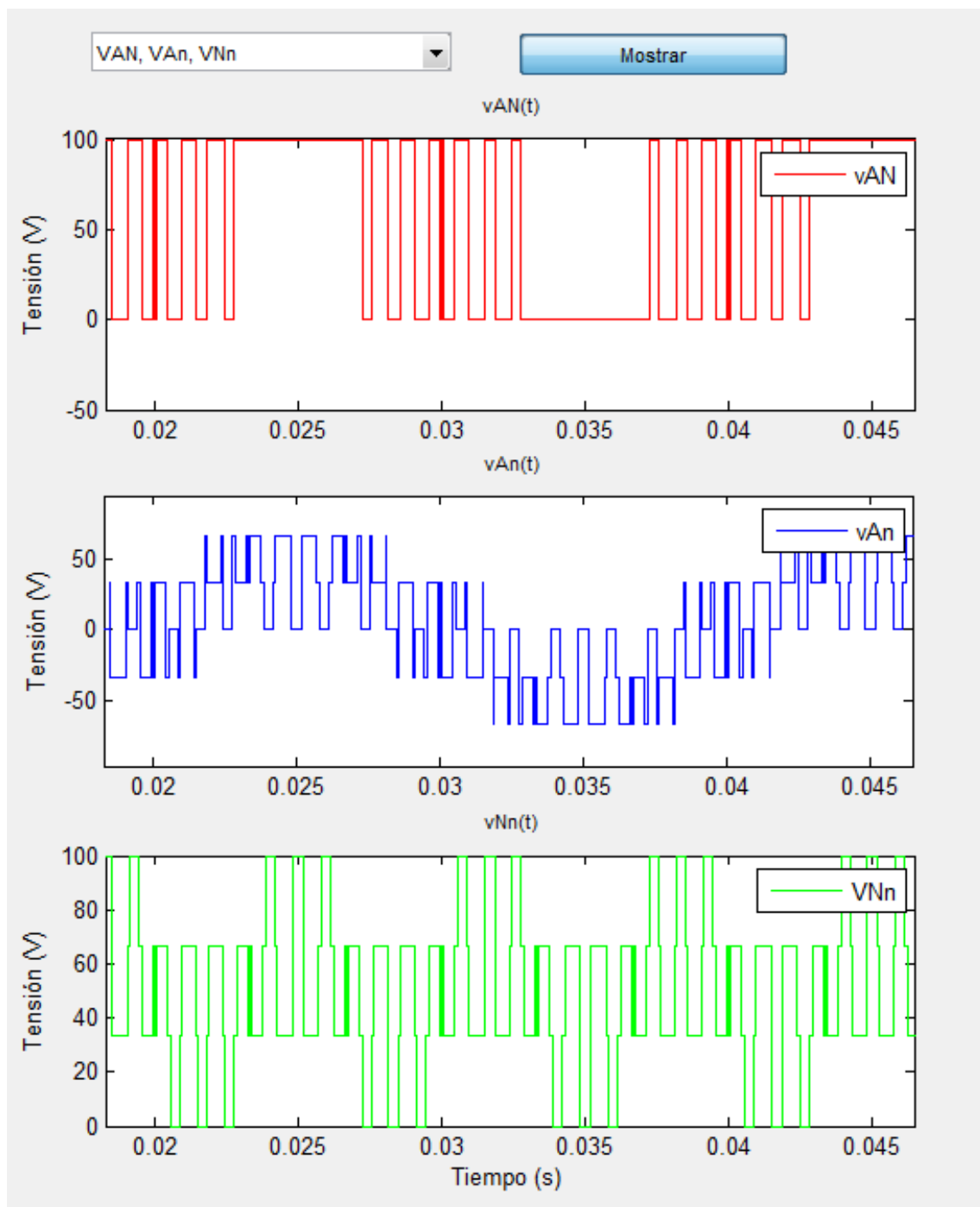


Figura 3.95: Tensiones  $v_{AN}$ ,  $v_{An}$  y  $v_{Nn}$  en modulación DPWM.



Una vez mostradas las diferentes tensiones del circuito, la siguiente opción “**iA, iB, iC**” representa las corrientes de fase de la salida del inversor. Estas corrientes se pueden ver en la Figura 3.96.

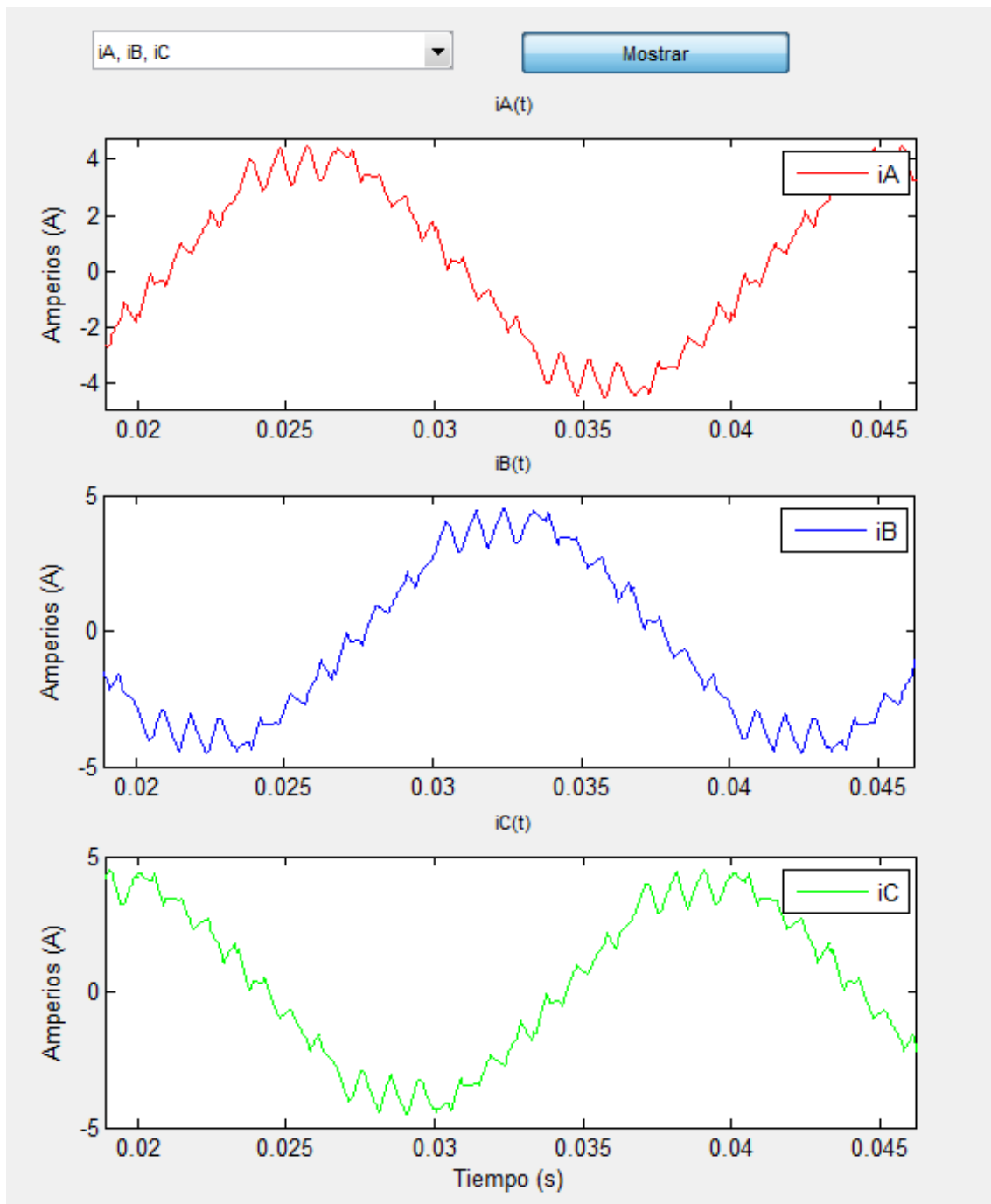


Figura 3.96: Corrientes de fase  $i_A$ ,  $i_B$  y  $i_C$  en modulación DPWM.

La siguiente opción, “**iA, iA1, iAripple**”, muestra la corriente por la fase A, además de su primer armónico y del rizado de la señal, tal y como se ve en la Figura 3.97.

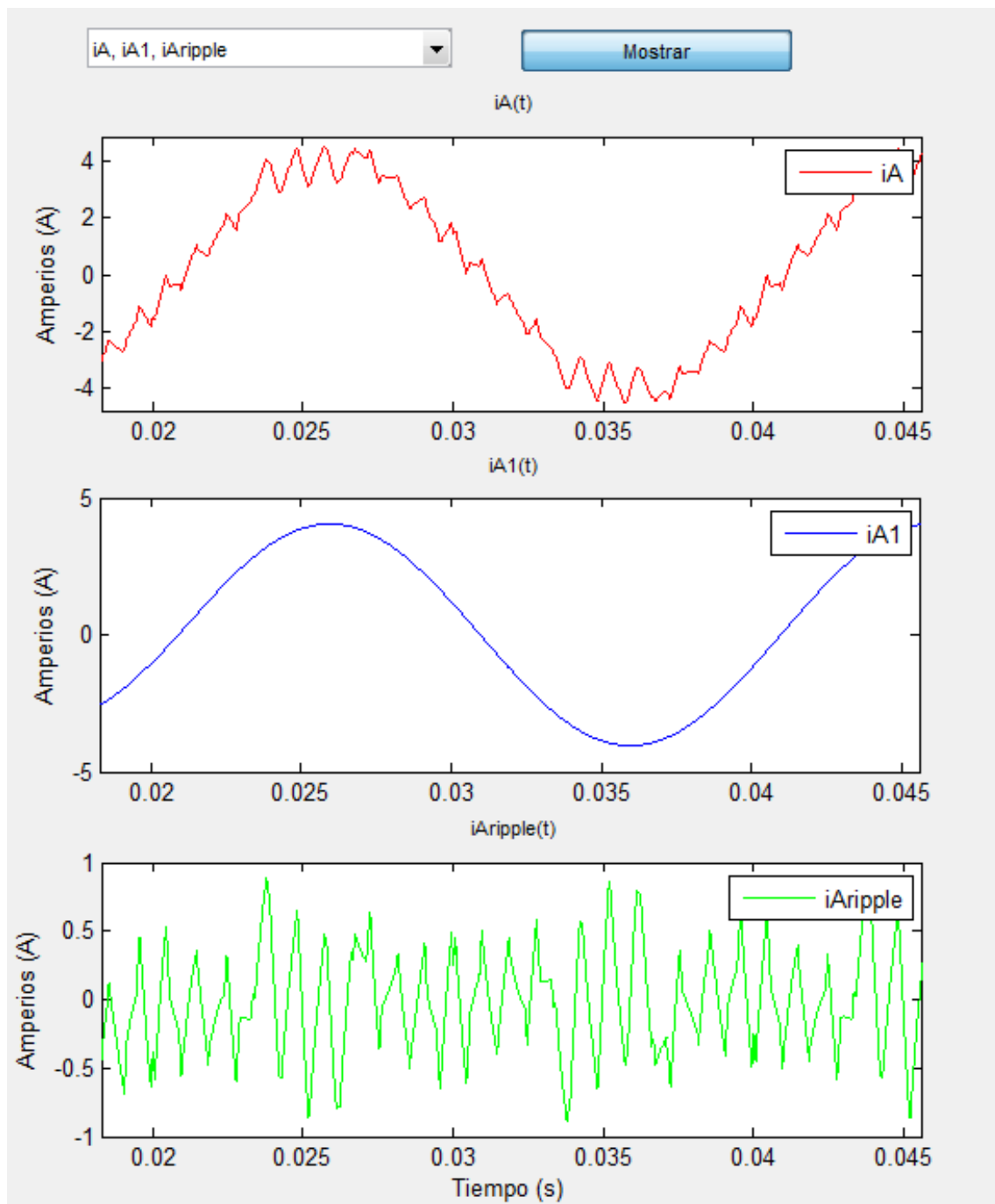


Figura 3.97: Corrientes de fase  $i_A$ ,  $i_{A1}$  y  $i_{Aripple}$  en modulación DPWM.

En la pestaña “ $i_I$ ,  $i_{C_I}$ ,  $i_{R_I}$ ,  $i_A$ ” se representan nuevamente las diferentes corrientes del circuito. La Figura 3.98 muestra dichas corrientes.

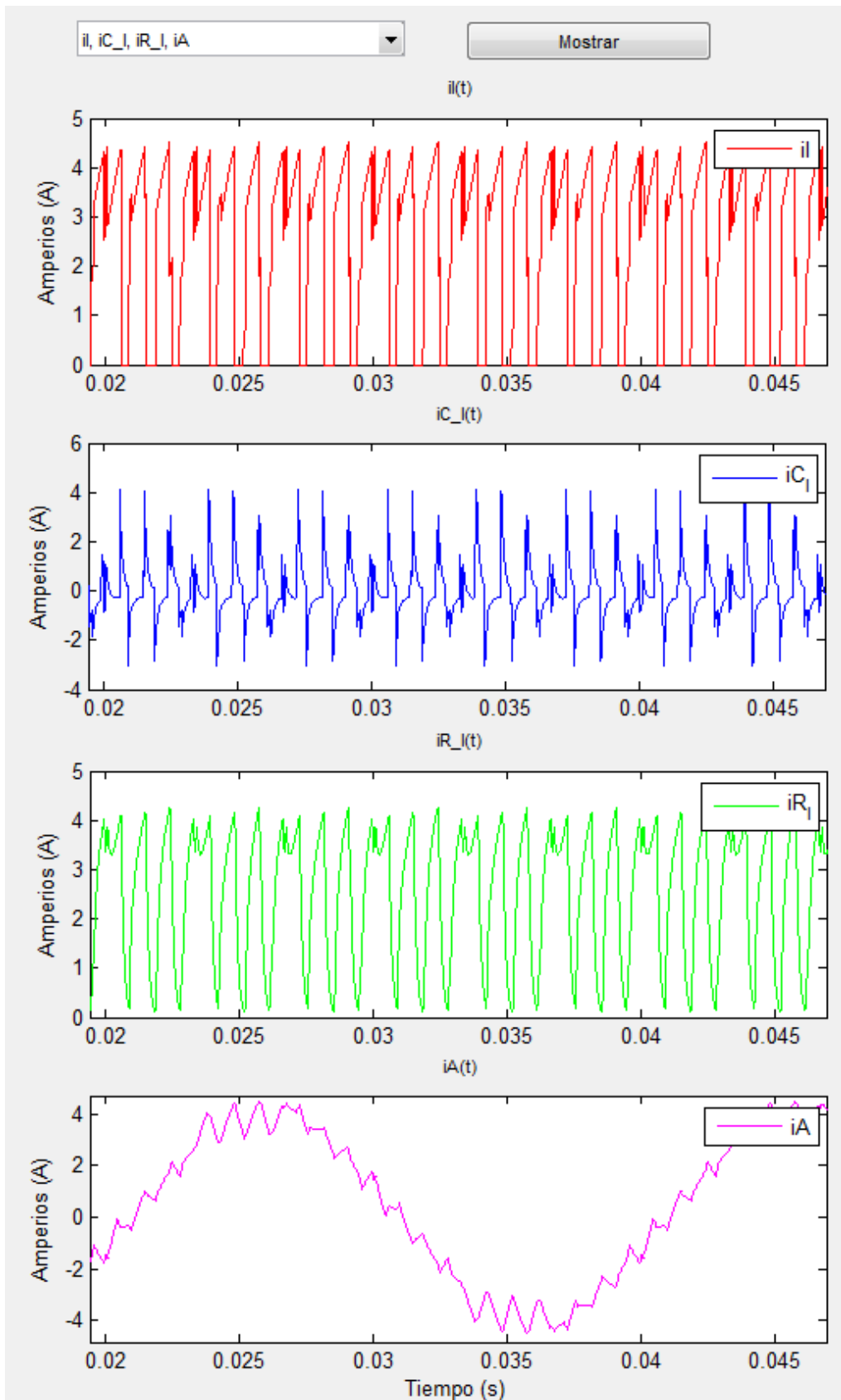


Figura 3.98: Corrientes  $i_I$ ,  $i_{C_I}$  y  $i_{R_I}$  e  $i_A$  en modulación DPWM.

Por último, en la pestaña “**FFT**”, que se muestra en la Figura 3.99, se puede ver el análisis de armónicos de la tensión de línea y de la corriente de fase.

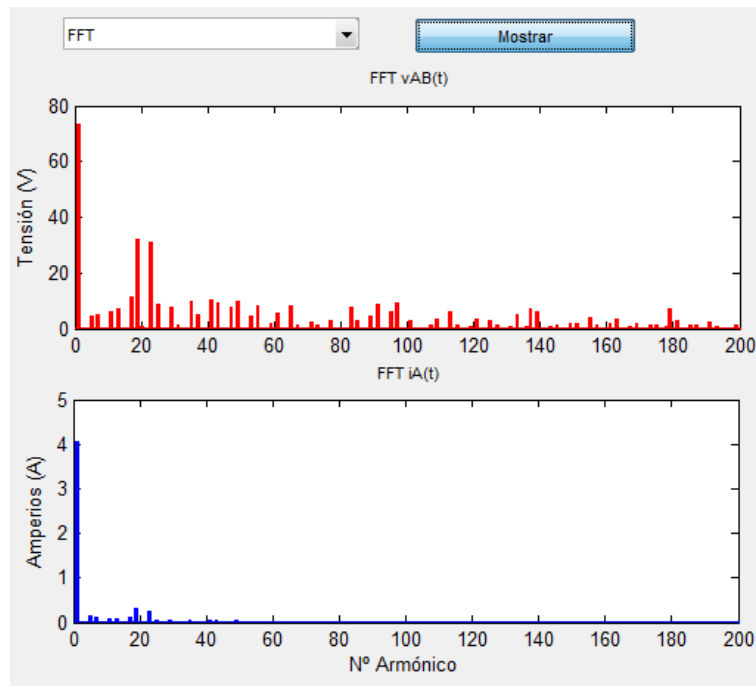


Figura 3.99: Armónicos tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación DPWM.

Una vez analizadas todas las opciones que se pueden visualizar en la ventana de representación se vuelve a la interfaz gráfica trifásica.

El último botón por analizar en esta ventana es el botón “**Representación VAB1=f(ma)**”, el cual, al ser pulsado, abrirá la ventana mostrada en la Figura 3.17.

Nuevamente se tiene una primera opción que consiste en una imagen precalculada como la de la Figura 3.100. La segunda opción consiste en realizar el cálculo con los valores de  $m_a$  inicial,  $m_a$  final y el paso deseado.

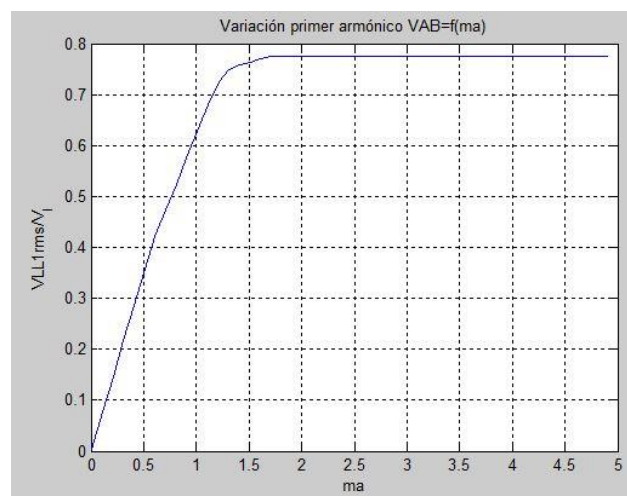


Figura 3.100: Imagen precalculada barrido en modulación DPWM.

### 3.3.5 Modulación vectorial

La modulación vectorial o modulación SVM es la última estudiada en este Trabajo Fin de Grado.

A diferencia de todas las modulaciones anteriores esta técnica no está basada en señales PWM sino en vectores.

Al pulsar el botón “Abrir”, se abrirá el mismo archivo .mdl que en el resto de las modulaciones trifásicas, tal y como se muestra en la Figura 3.50, a excepción del bloque de modulación llamado “Modulación Vectorial”, el cual se puede observar en la Figura 3.101. En esta técnica de modulación hay dos conjuntos de señales. Las primeras son las señales de onda cuadrada, empleadas en caso de que el vector  $\alpha\beta$  este fuera del hexágono de vectores estacionarios. El segundo conjunto de señales contiene las señales de control calculadas en caso de que el vector  $\alpha\beta$  esté dentro del hexágono de vectores estacionarios. Por último se dispone de un bloque que selecciona en cuál de las dos zonas de trabajo se está.

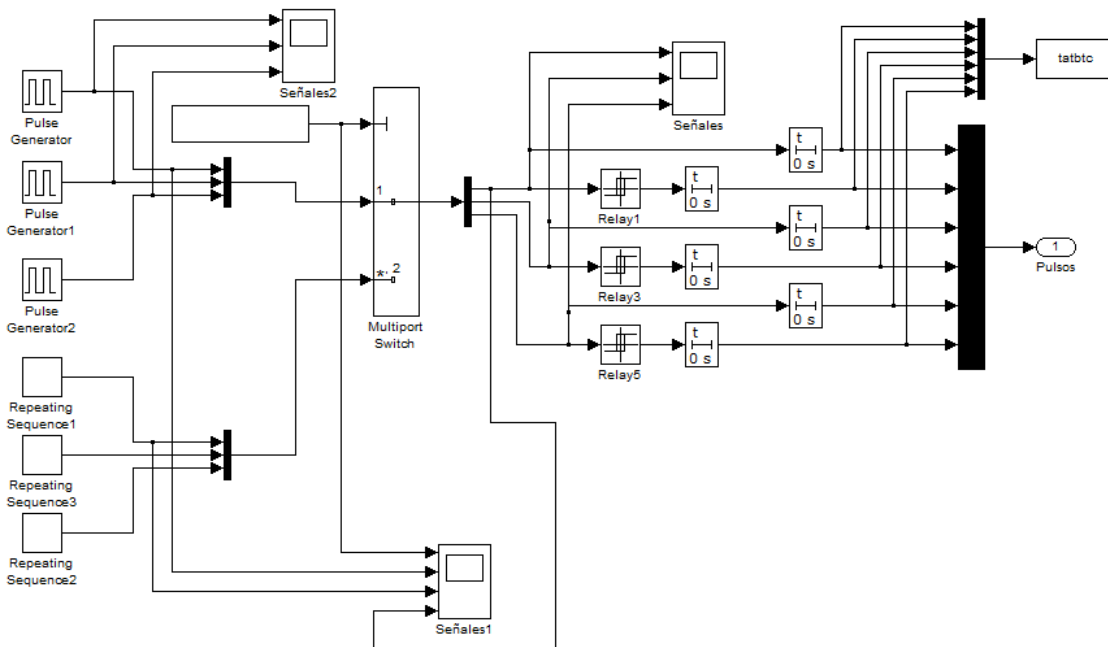


Figura 3.101: Circuito Simulink de modulación vectorial.

Al pulsar el botón “Calcular” se pueden ver nuevamente los resultados de la simulación.

El cálculo de estos resultados es diferente al procedimiento realizado en todas las modulaciones anteriores, ya que antes de ejecutarse el modelo Simulink se ejecuta un archivo .m en el que se generan las variables que utiliza el modelo. Es decir, en lugar de tener una conexión directa entre la interfaz y el modelo, la interfaz debe conectar con un script y este, posteriormente, con el modelo.

En este script lo primero que se realiza es calcular el vector  $\alpha\beta$ :

```
xalpha=(2*sqrt(2)*1.2248*ma*Vi*sin((j-1)*2*pi/mf+pi/10))/9-
(sqrt(2)*1.2248*ma*Vi*sin((j-1)*2*pi/mf-2*pi/3+pi/10))/9-
(sqrt(2)*1.2248*ma*Vi*sin((j-1)*2*pi/mf+2*pi/3+pi/10))/9;
xbeta=(sqrt(6)*1.2248*ma*Vi*sin((j-1)*2*pi/mf-2*pi/3+pi/10))/9-
(sqrt(6)*1.2248*ma*Vi*sin((j-1)*2*pi/mf+2*pi/3+pi/10))/9;
a(j)=xalpha;
b(j)=xbeta;
ab=sqrt(xalpha^2+xbeta^2);
alpha=atan2(xbeta, xalpha);
alpha=alpha*180/pi;
deg=alpha;
Ts=1/(mf*frec);
```

Una vez se tiene este vector, se calculan los tiempos de conmutación  $T_a$ ,  $T_b$  y  $T_c$ :

```
Ta=(sqrt(3)*Ts*ab*sin(pi/3-angulo*pi/180))/Vi;
Tb=(sqrt(3)*Ts*ab*sin(angulo*pi/180))/Vi;
Tc=Ts-Ta-Tb;
```

Una vez calculados los tiempos se asigna el vector al sextante, o subsextante en caso de estar en el modo de eliminación de armónicos pares, y se genera la secuencia de conmutación. En el siguiente ejemplo se calcula la secuencia para el primer sextante:

```
if (sextante == 1)
    t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta/2+Tb+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

timeA=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeB=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeC=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

signA = [0 0 1 1 1 1 1 1 0 0];
signB = [0 0 0 1 1 1 1 0 0 0];
signC = [0 0 0 0 1 1 0 0 0 0];
```

Este proceso se realiza  $m_f$  veces, generándose una secuencia para todo un periodo:

```
timeAtotal=[timeAtotal, timeA+(j-1)*Ts.*ones(1,10)];
timeBtotal=[timeBtotal, timeB+(j-1)*Ts.*ones(1,10)];
timeCtotal=[timeCtotal, timeC+(j-1)*Ts.*ones(1,10)];
signAtotal=[signAtotal, signA];
signBtotal=[signBtotal, signB];
signCtotal=[signCtotal, signC];
```

Durante todo este cálculo se va analizando si el modo de funcionamiento es lineal o si por el contrario se está trabajando en zona de saturación u onda cuadrada. Para ello se comprueba si el módulo del vector es mayor que  $\frac{2}{3}V_l$ , ya que si se cumple esta primera condición automáticamente se está trabajando en zona de saturación. Posteriormente se calcula si el tiempo  $T_c$  es  $\leq 0$ , ya que si es así también se estará trabajando en zona de onda cuadrada.

Una vez generada toda la secuencia de conmutación para un periodo se ejecuta el modelo de Simulink.

Como se puede observar en la Figura 3.102, en la cual se muestran estos resultados calculados con los valores por defecto, existe una nueva variable llamada “**Modo**”. Esta variable permite seleccionar entre modulación vectorial básica y modulación vectorial con eliminación de armónicos pares.

The interface is divided into several sections:

- Modulaciones:** A list of modulation types with radio buttons. "Modulación Vectorial" is selected.
- Circuit Diagram:** A schematic of a three-phase inverter bridge with a load (R, L) and a neutral point (n). Labels include  $V_I$ ,  $R_I$ ,  $i_{R_I}$ ,  $C_I$ ,  $i_{C_I}$ ,  $T_A^+$ ,  $D_A^+$ ,  $D_B^+$ ,  $T_B^+$ ,  $D_C^+$ ,  $T_C^+$ ,  $T_A^-$ ,  $D_A^-$ ,  $D_B^-$ ,  $T_B^-$ ,  $D_C^-$ ,  $T_C^-$ ,  $i_A$ ,  $R$ ,  $L$ ,  $U_{AB}$ , and  $n$ .
- Introducir Valores:** Input fields for:
  - Tensión Entrada  $V_I$  (V): 100
  - Resistencia Entrada  $R_I$  (Ohms): 0.1
  - Capacidad Entrada  $C_I$  (F): 0.001
  - Ron puente (Ohms): 0.001
  - Tiempo Muerto  $t_d$  (s): 0
  - Resistencia Salida  $R$  (Ohms): 10
  - Inductancia Salida  $L$  (H): 0.01
  - Frecuencia  $f_1$  (Hz): 50
  - Modulación Amplitud  $m_a$ : 0.8
  - Modulación Frecuencia  $m_f$ : 21
  - Modo: Normal (dropdown menu)
- Resultados:** Output fields for:
  - Corriente  $R_I$  (A): 3.01307
  - Corriente  $C_I$  (A): 1.48699
  - Potencia Entrada  $P_I$  (W): 286.976
  - $V_{AB1\_rms}$  (V): 55.9545
  - Delta  $i_A$  (A): 9.37012
  - Potencia Salida  $P_o$  (W): 287.836
  - THD  $v_{AB}$ : 0.780526
  - THD  $i_A$ : 0.0657601
- Buttons:** "Abrir", "Restaurar .MDL", "Calcular", "Borrar", "Datos por defecto", "Descargar en Placa", "Representación", "Representación  $V_{AB1}=f(m_a)$ ", "Teoría", and "Videotutorial".

Figura 3.102: Cálculos con modulación vectorial normal y valores por defecto.

El botón “**Representación**” abrirá una ventana de representación diferente a los casos anteriores y que se puede ver en la Figura 3.103. Esta ventana es diferente debido a que en la modulación vectorial no hay señal PWM que representar. En su lugar se representa el hexágono con los vectores principales y los estados tomados.

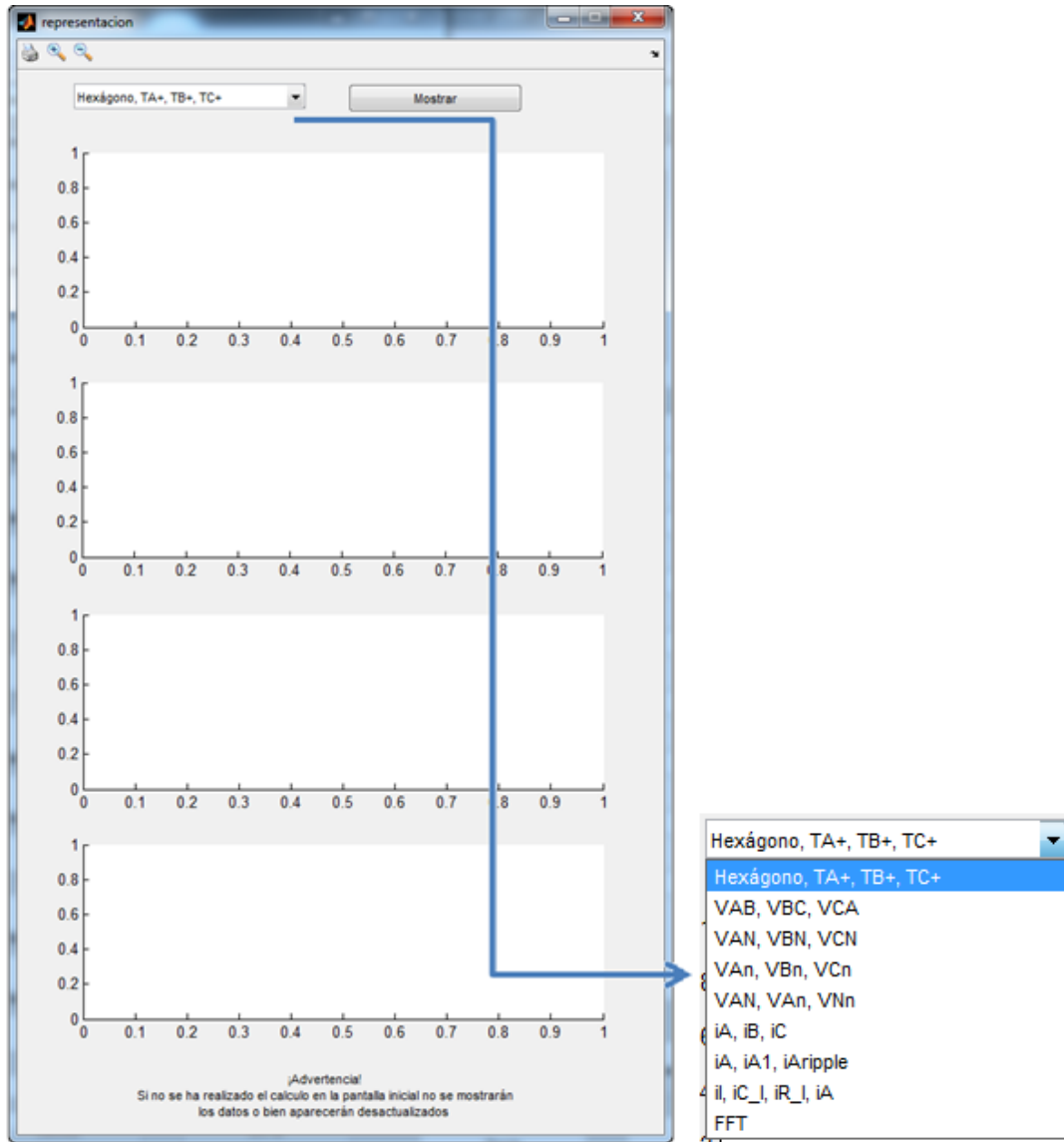


Figura 3.103: Pantalla representación formas de onda trifásica con modulación vectorial.



Al seleccionar “**Hexágono, TA+, TB+, TC+**” se observa el hexágono de estados, así como el estado de los transistores superiores del inversor, tal y como se aprecia en la Figura 3.104.

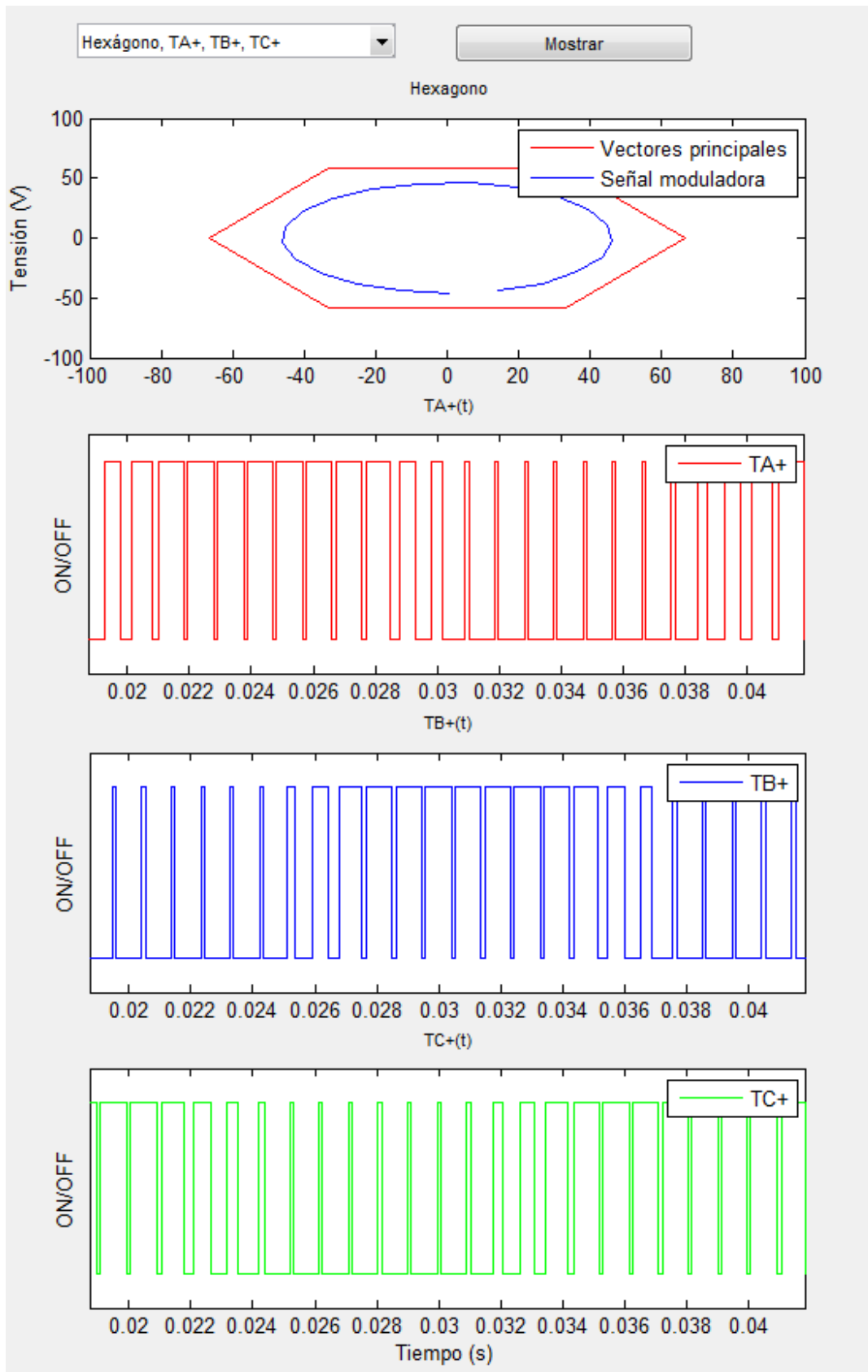


Figura 3.104: Hexágono de estados y señales TA+, TB+ y TC+ con modulación vectorial.

Pulsando sobre “**VAB, VBC, VCA**” se representan las tensiones de línea del sistema, tal y como se muestra en la Figura 3.105. En esta figura se puede observar como las tensiones de línea tienen simetría par respecto al paso por cero, lo cual es algo característico de esta modulación.

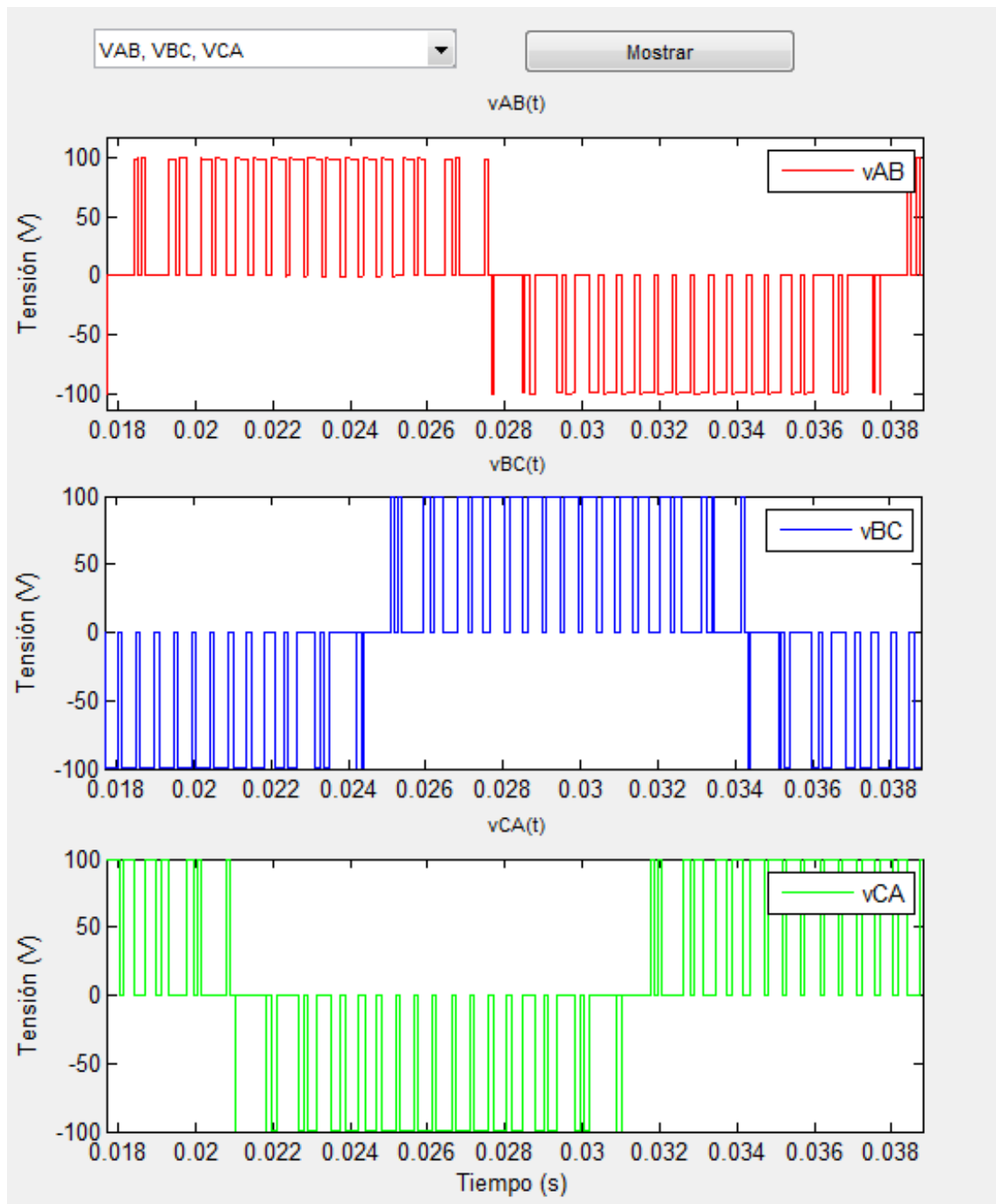


Figura 3.105: Tensiones  $v_{AB}$ ,  $v_{BC}$  y  $v_{CA}$  en modulación vectorial.

Nuevamente en la opción “**VAB, VBC, VCA**”, en caso de haber seleccionado la opción de “**Elim. Armónicos Pares**” y para un valor de  $m_f$  múltiplo de doce, se puede ver la simetría par y la simetría impar en la señal. Gracias a esto se consigue que esta tensión carezca de armónicos pares. En la Figura 3.106 se muestran las tensiones de línea para un valor de  $m_f = 24$ .

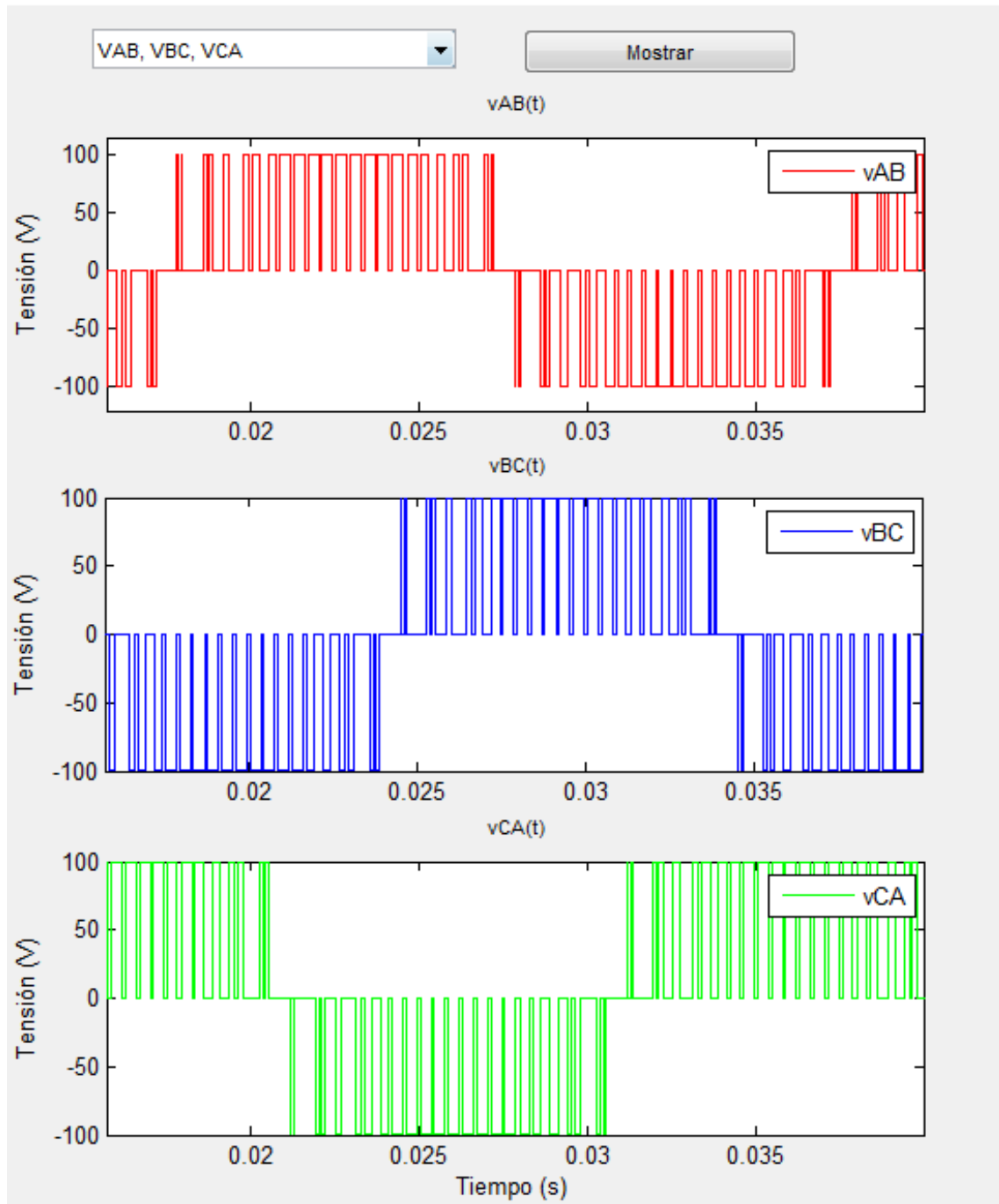


Figura 3.106: Tensiones  $v_{AB}$ ,  $v_{BC}$  y  $v_{CA}$  en modulación vectorial con eliminación de armónicos pares.

En la opción “VAN, VBN, VCN” se mostrarán las tensiones de cada fase con respecto a la entrada. En la Figura 3.107 se pueden observar dichas tensiones.

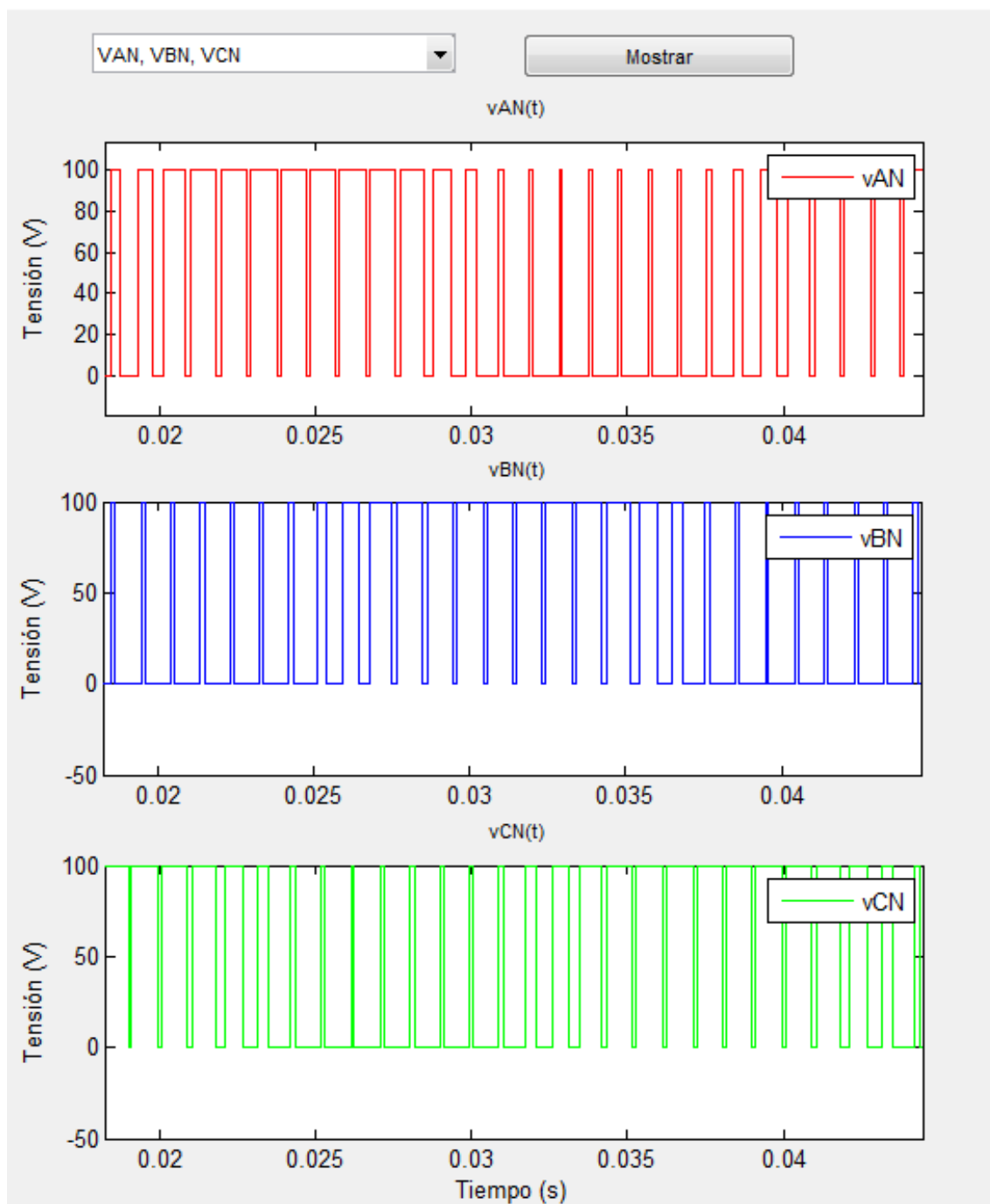


Figura 3.107: Tensiones  $v_{AN}$ ,  $v_{BN}$  y  $v_{CN}$  en modulación vectorial.

Al seleccionar sobre la opción “**VAn, VBn, VCn**” se generan las tensiones de cada fase. En la Figura 3.108 se pueden observar estas tensiones.

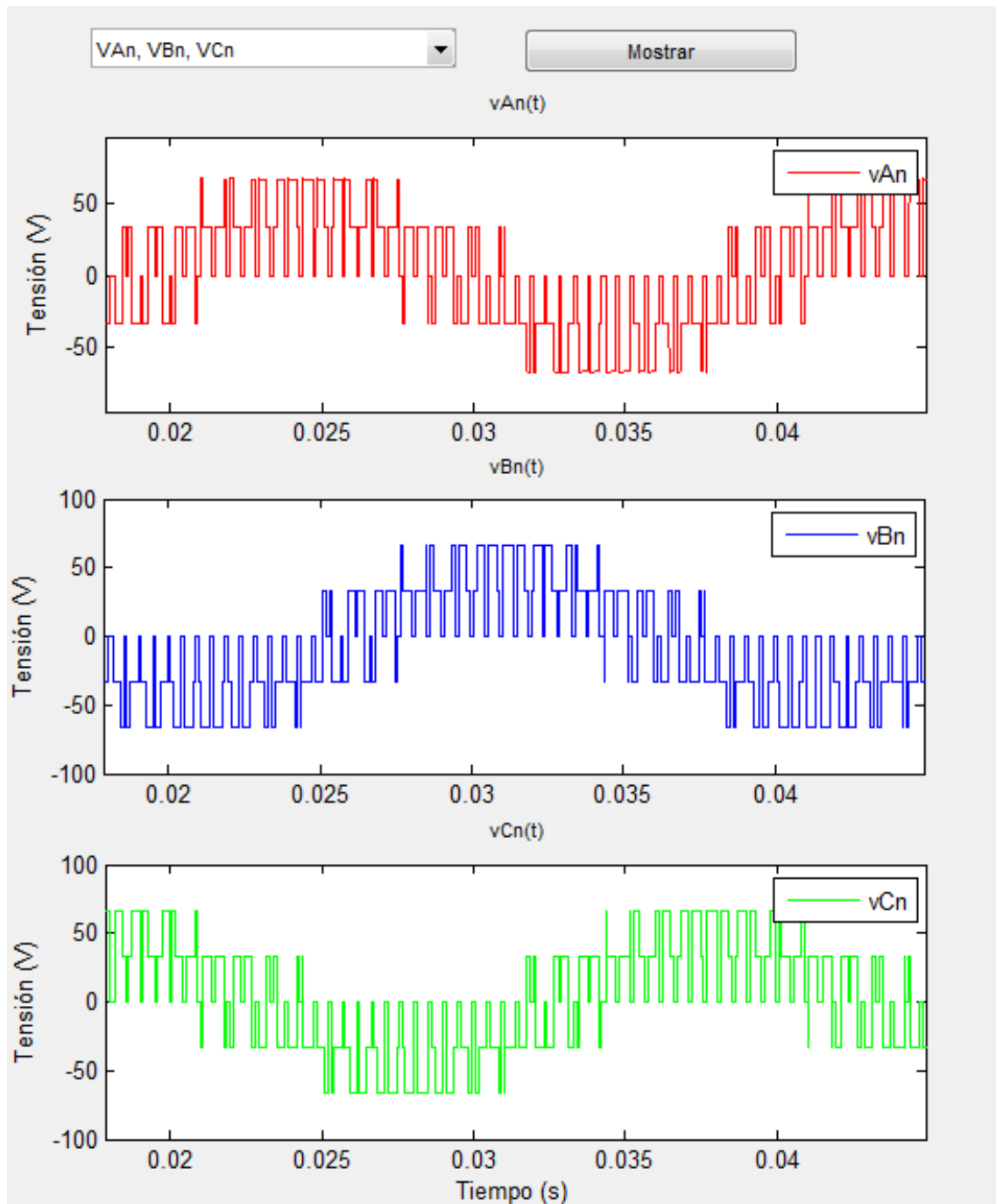


Figura 3.108: Tensiones  $v_{An}$ ,  $v_{Bn}$  y  $v_{Cn}$  en modulación vectorial.

La última opción relacionada sobre representación de tensiones es “VAN, VAn, VNn”. Como en los casos anteriores, en esta opción se visualizan las dos tensiones anteriores en la fase A además de la diferencia entre ambas, las cuales se visualizan en la Figura 3.109.

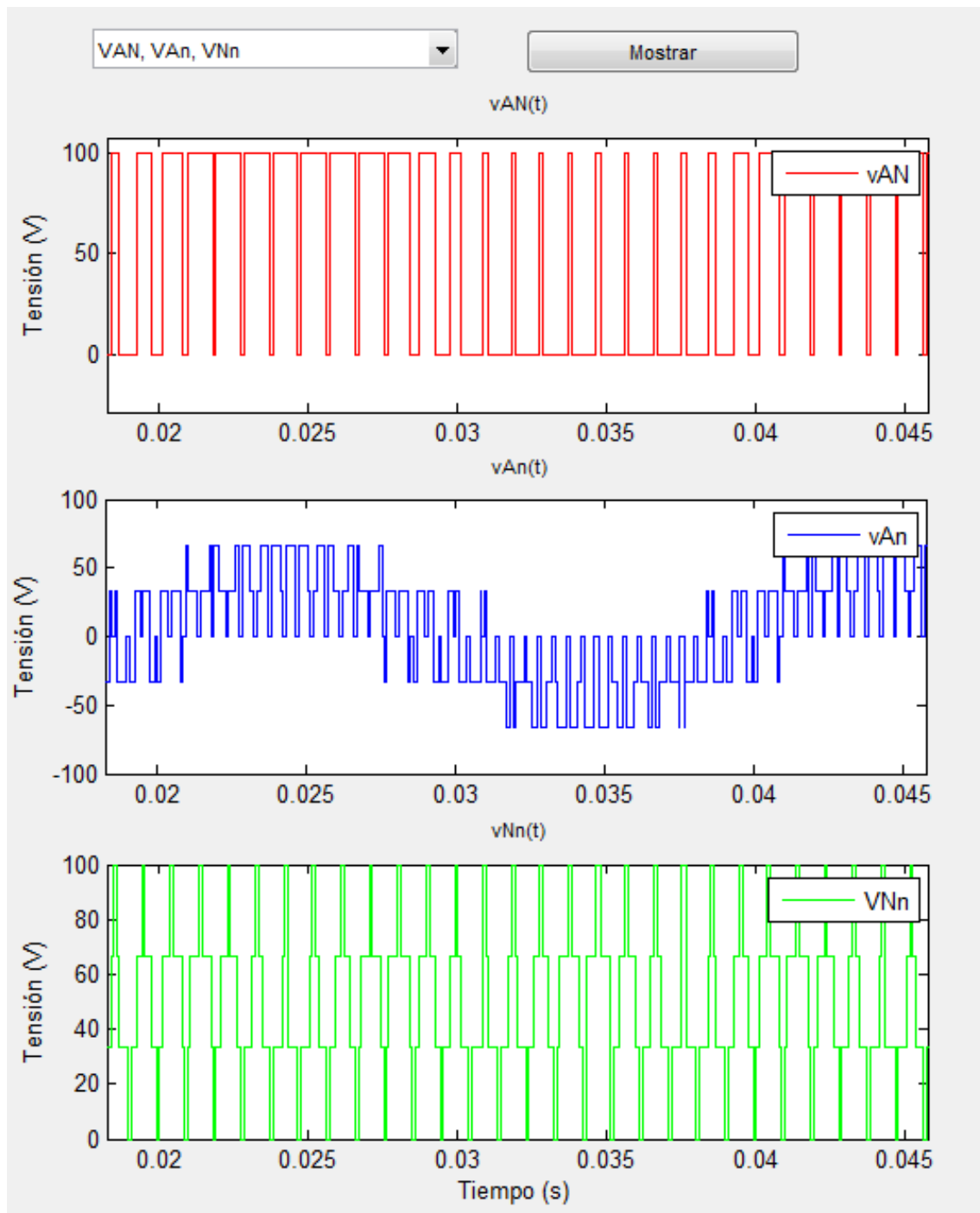


Figura 3.109: Tensiones  $v_{AN}$ ,  $v_{An}$  y  $v_{Nn}$  en modulación vectorial.

La siguiente opción que se permite seleccionar es “**iA, iB, iC**”. En esta opción se representan las corrientes de fase de la salida del inversor. Estas corrientes se muestran en la Figura 3.110.

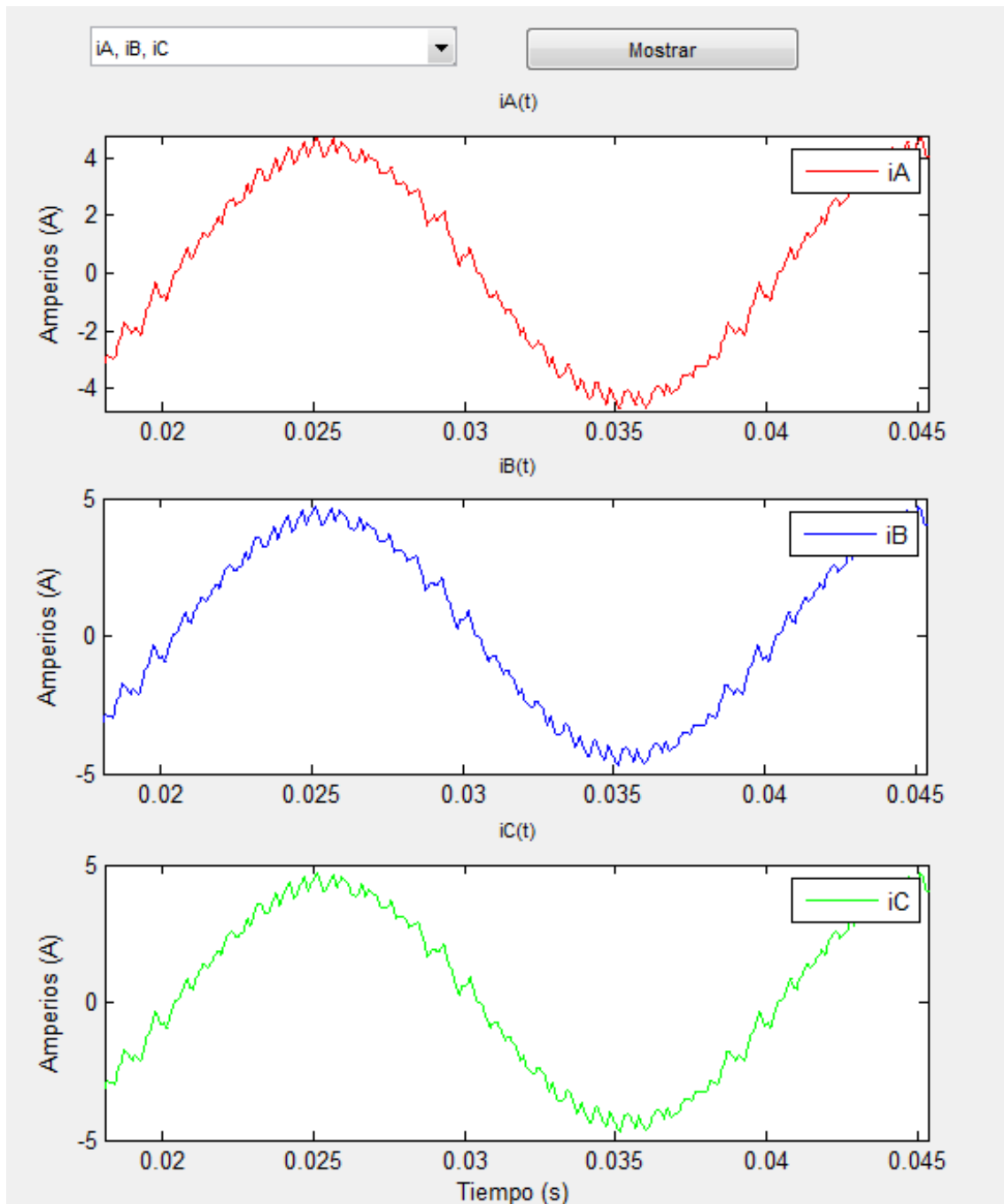


Figura 3.110: Corrientes de fase  $i_A$ ,  $i_B$  y  $i_C$  en modulación discontinua.

La siguiente opción, “**iA, iA1, iAripple**”, muestra la corriente por la fase A, además de su primer armónico y del rizado de la señal, tal y como se muestra en la Figura 3.111.

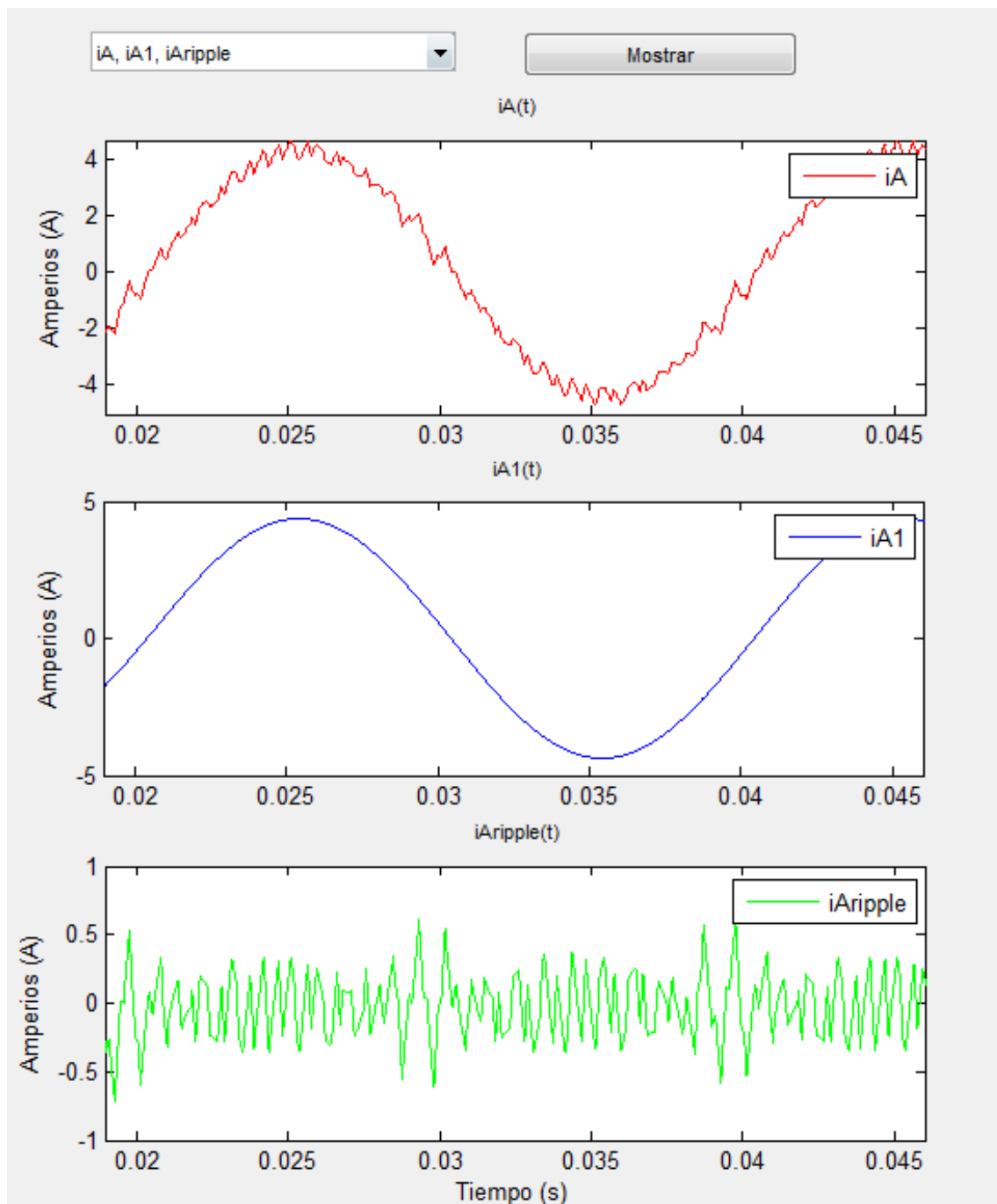


Figura 3.111: Corrientes de fase  $i_A$ ,  $i_{A1}$  y  $i_{Aripple}$  en modulación vectorial.



En la pestaña “ $i_l$ ,  $i_{C_I}$ ,  $i_{R_I}$ ,  $i_A$ ” se representan las diferentes corrientes del circuito. En la Figura 3.112 se observan dichas corrientes.

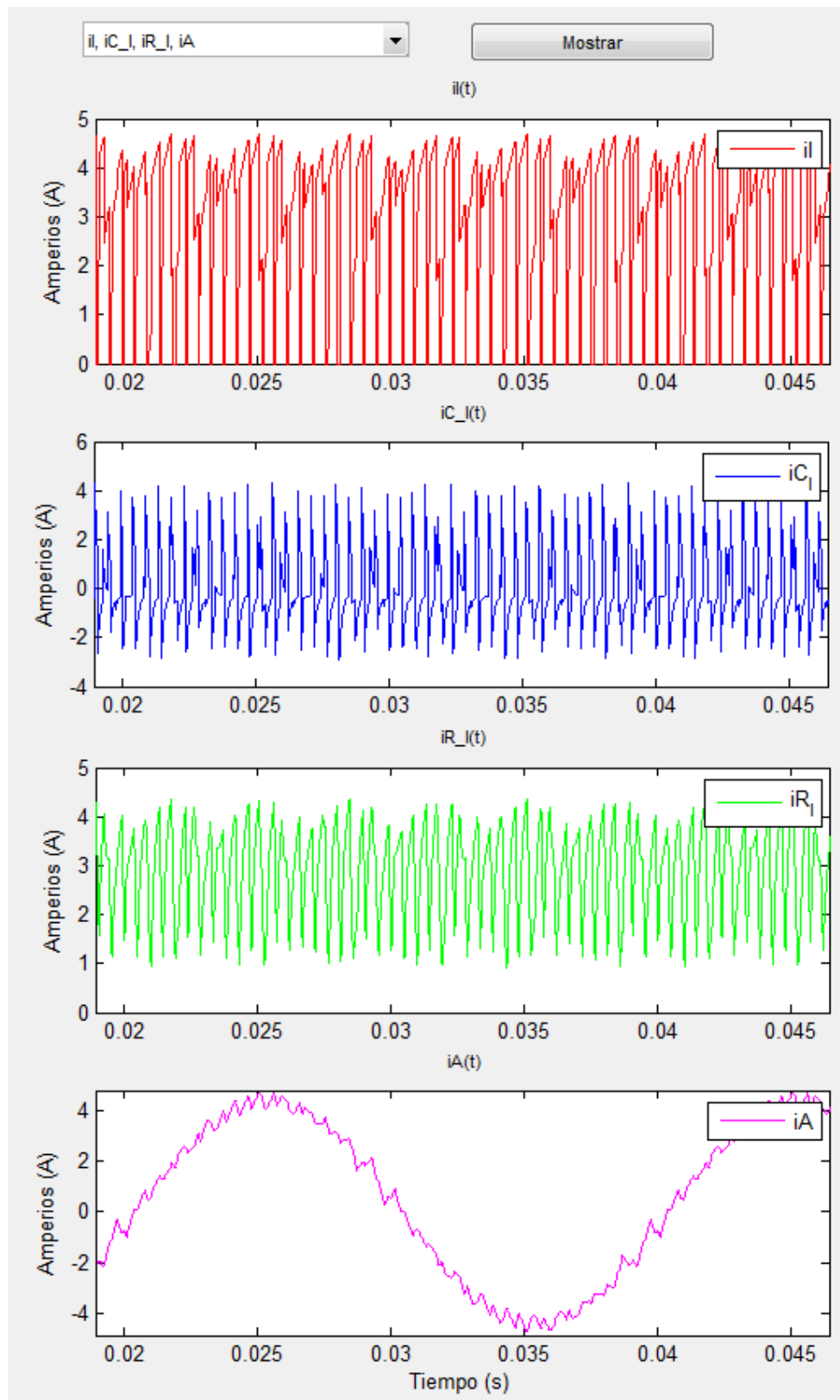


Figura 3.112: Corrientes  $i_l$ ,  $i_{C_I}$  y  $i_{R_I}$  e  $i_A$  en modulación vectorial.

Por último, en la pestaña “**FFT**”, se puede ver el análisis de armónicos de la tensión de línea y de la corriente de fase.

En la opción de modulación vectorial normal se puede ver como hay armónicos pares e impares en la Figura 3.113, sin embargo, en la Figura 3.114 se puede observar como en la modulación vectorial con eliminación de armónicos pares estos armónicos se reducen hasta casi su desaparición.

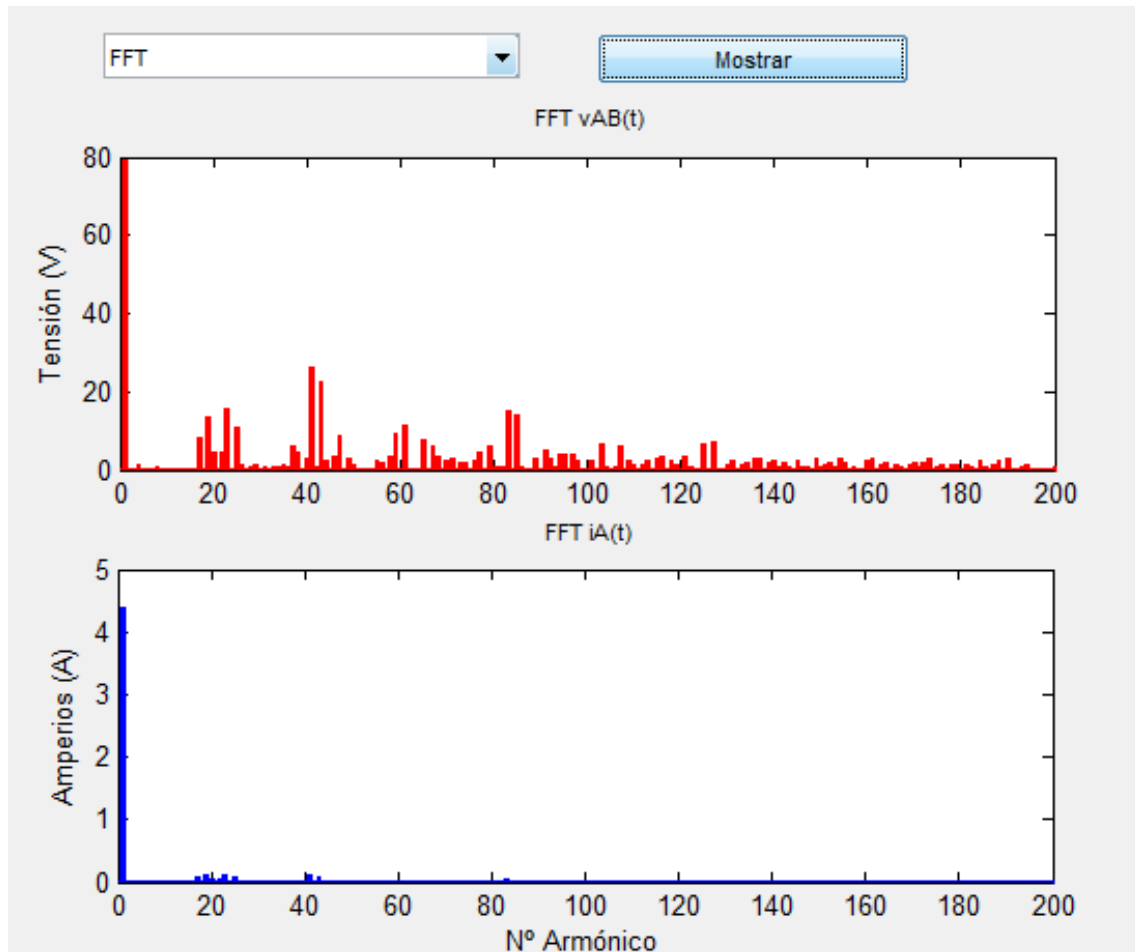


Figura 3.113: Armónicos de tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación vectorial normal.

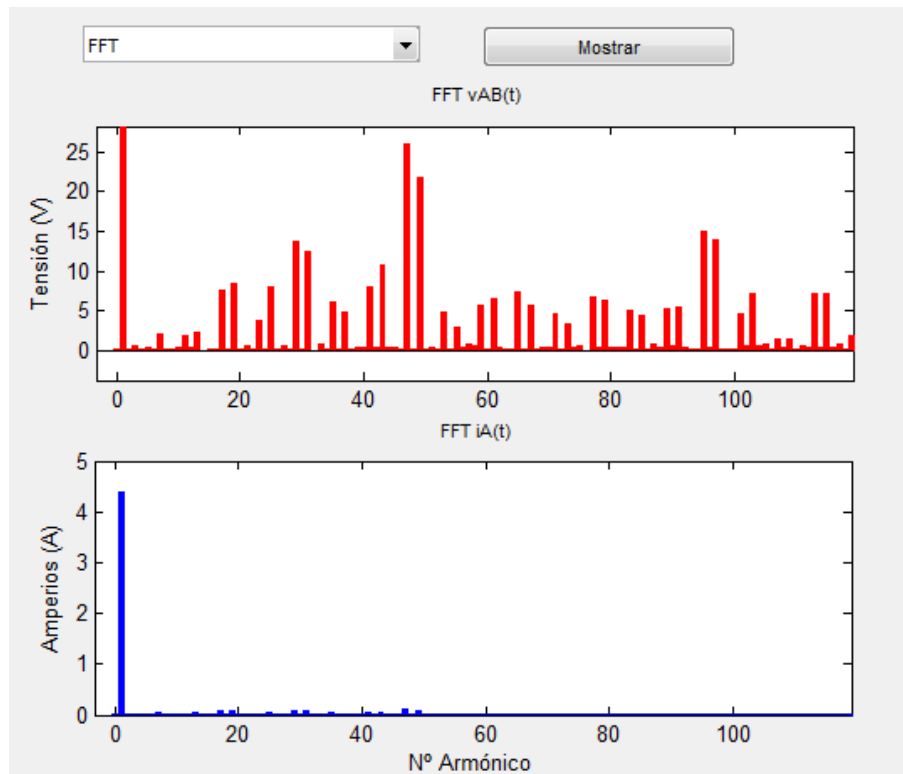


Figura 3.114: Armónicos tensión de línea ( $v_{AB}$ ) y corriente de fase ( $i_A$ ) en modulación vectorial con eliminación de armónicos pares para un valor de  $m_f = 24$ .

Una vez analizadas las diferentes opciones que se pueden visualizar en la ventana de representación se vuelve a la interfaz gráfica trifásica.

El último botón por analizar en esta ventana es el botón “**Representación VAB1=f(ma)**”, el cual, al ser pulsado, abrirá la ventana de la Figura 3.17.

La primera opción consiste en una imagen precalculada como la de la Figura 3.115. La segunda opción consiste en realizar el cálculo con los valores de  $m_a$  inicial,  $m_a$  final y el paso deseado.

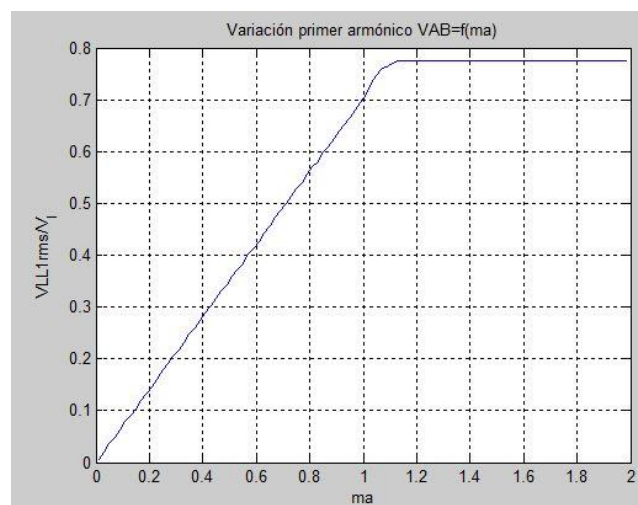


Figura 3.115: Imagen precalculada del barrido en modulación vectorial.



# Capítulo 4

## Conclusiones y Líneas Futuras

### 4.1 Conclusiones

Se ha realizado una herramienta educativa que ayudará al alumno a comprender el funcionamiento de los convertidores DC/AC, tanto monofásicos como trifásicos, con diferentes técnicas de modulación habitualmente estudiadas en asignaturas relacionadas con la electrónica de potencia. De esta forma, el alumno puede comprobar experimentalmente los conocimientos teóricos estudiados y evaluar el efecto de los diferentes parámetros sobre los elementos del sistema.

Además esta herramienta complementa a la ya creada sobre convertidores DC/DC [4] lo cual consigue acercar el objetivo general de crear una herramienta global que permita simular todas las tipologías de circuito estudiadas generalmente en electrónica de potencia.

En el ámbito personal, este proyecto me ha permitido mejorar mis conocimientos sobre inversores de potencia. Además me ha permitido mejorar mis conocimientos en el software Matlab, mejorando mis aptitudes en la herramienta Simulink y aprendiendo el funcionamiento de la herramienta GUIDE.

En cuanto a la temática del proyecto me ha parecido muy interesante. Ya que esta asignatura fue una de las más gratificantes del grado y gracias a este proyecto, he creado una herramienta que se utilizará en dicha asignatura.

### 4.2 Líneas Futuras

Una vez finalizado este Trabajo Fin de Grado se tienen varias posibilidades de ampliación y mejora.

La primera mejora posible a realizar es la implementación de la parte de descarga en placa de los modelos Simulink para pruebas reales sobre convertidores DC/AC. Con esta tarea se permitiría, mediante una tarjeta FPGA y un circuito de acondicionamiento, ver el funcionamiento

real de los inversores en un cubo Semiteach\_IGBT tal y como se ha realizado en el proyecto predecesor a este sobre convertidores DC/DC [4]. Para ello se debería implementar un programa con la herramienta System Generator de Xilinx por ejemplo que, empleando ciertos bloques de

Simulink, permite realizar una implementación en FPGA automáticamente para poder realizar pruebas en un circuito inversor real.

La siguiente ampliación posible para este Trabajo Fin de Grado consiste en completarlo con los diferentes tipos de circuito, añadiendo a los ya implementados convertidores DC/DC y DC/AC, los convertidores AC/DC también estudiados en la asignatura.

Por último cabe destacar la posibilidad de realizar herramientas similares a la creada en este Trabajo Fin de Grado para otras asignaturas. Con esto se consigue que el alumno sea más autodidacta. Además, el software Matlab tiene librerías sobre diferentes temáticas por lo que se podrían crear herramientas de simulación sobre asignaturas tanto de electrónica como de otras áreas del conocimiento.

Parte III

# Pliego de condiciones





# Pliego de condiciones

- Ordenador
  - Sistema Operativo: Windows XP o posteriores.
  - Procesador: Intel Pentium (Pentium IV y posterior), Inter Celeron, Intel Xeon, Intel Core, AMD Athlon, AMD Optaron.
  - Disco duro: se requieren al menos de 500MB de espacio disponible en disco.
  - RAM: 512MB como mínimo.
  - Video: Tarjeta gráfica de 16, 24 o 32 bits con OpenGL y DirectX9.
- Matlab versión 2010<sup>a</sup> o posteriores.
- Software Word del paquete Microsoft Office.



Parte IV

# Presupuesto



# Presupuesto

El presupuesto de este TFG se basa en la cuantía de los materiales utilizados y los honorarios de desarrollador. En la Tabla 5 se recogen los cálculos realizados para el Presupuesto de Ejecución de Material (PEM), debido principalmente a los equipos informáticos y al software utilizado.

<b>Presupuesto de ejecución por material (PEM)</b>			
<b>Concepto</b>	<b>Cantidad</b>	<b>Precio unitario</b>	<b>Coste Total</b>
<b>Material:</b>			
Ordenador	1	550,00 €	550,00 €
Material Oficina	1	20,00 €	20,00 €
<b>Material Software:</b>			
Windows 7.0 Professional	1	145,00 €	145,00 €
Office 365 Hogar	1	99,00 €	99,00 €
Matlab 2012 <sup>1</sup>	1	163,00 €	163,00 €
<b>Total PEM</b>			<b>1.432,92 €</b>

Tabla 5: Presupuesto de ejecución por material (PEM).

Para este TFG se ha contratado a un Ingeniero para el desarrollo de la aplicación. Al finalizar el proyecto, el Ingeniero realizará las labores de redacción de memoria. En la Tabla 6 se calculan los gastos totales debidos al personal contratado.

<b>Honorarios</b>				
<b>Concepto</b>	<b>Nº personal</b>	<b>Tiempo (horas)</b>	<b>Precio unitario (€/hora)</b>	<b>Coste Total</b>
<b>Ingeniería</b>	1	180	50,00 €	9.000,00 €
<b>Mecanografiado</b>	1	35	15,00 €	525,00 €
<b>Total</b>				<b>9.525,00 €</b>

Tabla 6: Honorarios del empleado.

<sup>1</sup> Incluye software Matlab con herramienta Simulink y librería SimPowerSystem.

El presupuesto total del TFG será, por tanto, la suma del PEC y de los honorarios del empleado. Al resultado obtenido se le aplica el Impuesto de Valor Añadido (IVA). En la Tabla 7: Presupuesto Total del TFG se detallan los gastos totales del proyecto.

<b>Presupuesto</b>	
<b>Concepto</b>	<b>Coste Total</b>
PEC	1.432,92 €
Honorarios	9.525,00 €
Material fungible:	
Gastos de impresión	30,00 €
Encuadernación	90,00 €
<b>Subtotal</b>	<b>11.077,92 €</b>
IVA (21%)	2.326,36 €
<b>Total presupuesto</b>	<b>13.404,28 €</b>

Tabla 7: Presupuesto Total del TFG

Parte V

# Diagramas





# Diagramas

## Código Matlab modulación vectorial normal trifásica

```
e=0;

clear alfa modulo;
clear timeC timeB timeA signA signB signC;

hexang=[0 60 120 180 240 300 360];
hexamod=[Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3];

timeAtotal=[];
timeBtotal=[];
timeCtotal=[];
signAtotal=[];
signBtotal=[];
signCtotal=[];
control=[];
tiempocontrol=[];

for j=1:mf

    xalpha=1.2248*((2*sqrt(2)*ma*Vi*sin((j-1)*2*pi/mf+pi/10))/9-
(sqrt(2)*ma*Vi*sin((j-1)*2*pi/mf-2*pi/3+pi/10))/9-
(sqrt(2)*ma*Vi*sin((j-1)*2*pi/mf+2*pi/3+pi/10))/9);
    xbeta=1.2248*((sqrt(6)*ma*Vi*sin((j-1)*2*pi/mf-2*pi/3+pi/10))/9-
(sqrt(6)*ma*Vi*sin((j-1)*2*pi/mf+2*pi/3+pi/10))/9);
    a(j)=xalpha;
    b(j)=xbeta;
    ab=sqrt(xalpha^2+xbeta^2);
    alpha=atan2(xbeta, xalpha);
    alpha=alpha*180/pi;
    deg=alpha;
    Ts=1/(mf*frec);

    if ab>=2*Vi/3

        estado=1;

    else
    if alpha<=0
        alpha=alpha+360;
    end

    sextante=ceil(alpha/59.999);
```

```

if sextante==7
    sextante=1;
end

angulo=mod(alpha, 59.99);

%para control
alfa(j)=alpha;
modulo(j)=ab;
sextant(j)=sextante;
angul(j)=angulo;

Ta=(sqrt(3)*Ts*ab*sin(pi/3-angulo*pi/180))/Vi;
Tb=(sqrt(3)*Ts*ab*sin(angulo*pi/180))/Vi;
Tc=Ts-Ta-Tb;

if Tc<0
    estado=1;

timeA=[0 0 Ts Ts];
timeB=[0 0 Ts Ts];
timeC=[0 0 Ts Ts];
signA = [0 1 1 0];
signB = [0 1 1 0];
signC = [0 1 1 0];

timeAtotal=[timeAtotal, timeA+(j-1)*Ts.*ones(1,4)];
timeBtotal=[timeBtotal, timeB+(j-1)*Ts.*ones(1,4)];
timeCtotal=[timeCtotal, timeC+(j-1)*Ts.*ones(1,4)];
signAtotal=[signAtotal, signA];
signBtotal=[signBtotal, signB];
signCtotal=[signCtotal, signC];

else
    if (sextante == 1)

        t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Ts-Tc/4-Ta/2; t6=Ts-Tc/4; t7=Ts;

        timeA=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
        timeB=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
        timeC=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

        signA = [0 0 1 1 1 1 1 1 0 0];
        signB = [0 0 0 1 1 1 1 0 0 0];
        signC = [0 0 0 0 1 1 0 0 0 0];

    elseif (sextante == 2)

        t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Ts-Tc/4-Tb/2; t6=Ts-Tc/4; t7=Ts;

        timeB=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
        timeA=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
        timeC=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

        signB = [0 0 1 1 1 1 1 1 0 0];

```

```

signA = [0 0 0 1 1 1 1 0 0 0];
signC = [0 0 0 0 1 1 0 0 0 0];

elseif (sextante == 3)

    t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Ts-Tc/4-Ta/2; t6=Ts-Tc/4; t7=Ts;

    timeB=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeC=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeA=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signB = [0 0 1 1 1 1 1 1 0 0];
    signC = [0 0 0 1 1 1 1 0 0 0];
    signA = [0 0 0 0 1 1 0 0 0 0];

elseif (sextante == 4)

    t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Ts-Tc/4-Tb/2; t6=Ts-Tc/4; t7=Ts;

    timeC=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeB=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeA=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signC = [0 0 1 1 1 1 1 1 0 0];
    signB = [0 0 0 1 1 1 1 0 0 0];
    signA = [0 0 0 0 1 1 0 0 0 0];

elseif (sextante == 5)

    t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Ts-Tc/4-Ta/2; t6=Ts-Tc/4; t7=Ts;

    timeC=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeA=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeB=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signC = [0 0 1 1 1 1 1 1 0 0];
    signA = [0 0 0 1 1 1 1 0 0 0];
    signB = [0 0 0 0 1 1 0 0 0 0];

elseif (sextante == 6)

    t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Ts-Tc/4-Tb/2; t6=Ts-Tc/4; t7=Ts;

    timeA=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeC=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeB=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signA = [0 0 1 1 1 1 1 1 0 0];
    signC = [0 0 0 1 1 1 1 0 0 0];
    signB = [0 0 0 0 1 1 0 0 0 0];

end

timeAtotal=[timeAtotal, timeA+(j-1)*Ts.*ones(1,10)];
timeBtotal=[timeBtotal, timeB+(j-1)*Ts.*ones(1,10)];

```

```

timeCtotal=[timeCtotal, timeC+(j-1)*Ts.*ones(1,10)];
signAtotal=[signAtotal, signA];
signBtotal=[signBtotal, signB];
signCtotal=[signCtotal, signC];

estado=2;
end
end

control=[control,[estado estado]];
tiempocontrol=[tiempocontrol,[(j-1)*Ts j*Ts]];

est(j)=estado;
end

comp=isempty(timeAtotal);

if comp==1
    timeAtotal=[0 1];
    timeBtotal=[0 1];
    timeCtotal=[0 1];
    signAtotal=[0 1];
    signBtotal=[0 1];
    signCtotal=[0 1];
end

assignin('base','timeAtotal',floor(timeAtotal*1e5)/1e5);
assignin('base','timeBtotal',floor(timeBtotal*1e5)/1e5);
assignin('base','timeCtotal',floor(timeCtotal*1e5)/1e5);
assignin('base','signAtotal',floor(signAtotal*1e5)/1e5);
assignin('base','signBtotal',floor(signBtotal*1e5)/1e5);
assignin('base','signCtotal',floor(signCtotal*1e5)/1e5);

assignin('base','control',control);
assignin('base','tiempocontrol',tiempocontrol);

```

### **Código Matlab modulación vectorial con eliminación armónicos pares trifásica**

```

e=0;

clear alfa modulo;
clear timeC timeB timeA signA signB signC;

hexang=[0 60 120 180 240 300 360];
hexamod=[Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3 Vi*2/3];

timeAtotal=[];
timeBtotal=[];
timeCtotal=[];
signAtotal=[];
signBtotal=[];
signCtotal=[];
control=[];
tiempocontrol=[];

for j=1:mf

```

```

    xalpha=(2*sqrt(2)*1.2248*ma*Vi*sin((j-1)*2*pi/mf+pi/10))/9-
(sqrt(2)*1.2248*ma*Vi*sin((j-1)*2*pi/mf-2*pi/3+pi/10))/9-
(sqrt(2)*1.2248*ma*Vi*sin((j-1)*2*pi/mf+2*pi/3+pi/10))/9;
    xbeta=(sqrt(6)*1.2248*ma*Vi*sin((j-1)*2*pi/mf-2*pi/3+pi/10))/9-
(sqrt(6)*1.2248*ma*Vi*sin((j-1)*2*pi/mf+2*pi/3+pi/10))/9;
    a(j)=xalpha;
    b(j)=xbeta;
    ab=sqrt(xalpha^2+xbeta^2);
    alpha=atan2(xbeta, xalpha);
    alpha=alpha*180/pi;
    deg=alpha;
    Ts=1/(mf*frec);

    if ab>=2*Vi/3
        estado=1;

    else

    if alpha<=0
        alpha=alpha+360;
    end

    sextante=ceil(alpha/29.999);
    if sextante==13
        sextante=1;
    end

    angulo=mod(alpha, 59.999);

    %para control
    alfa(j)=alpha;
    modulo(j)=ab;
    sextant(j)=sextante;
    angul(j)=angulo;

    Ta=(sqrt(3)*Ts*ab*sin(pi/3-angulo*pi/180))/Vi;
    Tb=(sqrt(3)*Ts*ab*sin(angulo*pi/180))/Vi;
    Tc=Ts-Ta-Tb;

    if Tc<0
        estado=1;

        timeA=[0 0 Ts Ts];
        timeB=[0 0 Ts Ts];
        timeC=[0 0 Ts Ts];
        signA = [0 1 1 0];
        signB = [0 1 1 0];
        signC = [0 1 1 0];

        timeAtotal=[timeAtotal, timeA+(j-1)*Ts.*ones(1,4)];
        timeBtotal=[timeBtotal, timeB+(j-1)*Ts.*ones(1,4)];
        timeCtotal=[timeCtotal, timeC+(j-1)*Ts.*ones(1,4)];
        signAtotal=[signAtotal, signA];
        signBtotal=[signBtotal, signB];
        signCtotal=[signCtotal, signC];
    else
        if (sextante == 1)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta/2+Tb+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeA=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeB=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeC=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signA = [0 0 1 1 1 1 1 1 0 0];
signB = [0 0 0 1 1 1 1 0 0 0];
signC = [0 0 0 0 1 1 0 0 0 0];

```

```

elseif (sextante == 2)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta+Tb/2+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeC=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeB=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeA=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signC = [1 1 0 0 0 0 0 0 1 1];
signB = [1 1 1 0 0 0 0 1 1 1];
signA = [1 1 1 1 0 0 1 1 1 1];

```

```

elseif (sextante == 3)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta/2+Tb+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeC=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeA=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeB=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signC = [1 1 0 0 0 0 0 0 1 1];
signA = [1 1 1 0 0 0 0 1 1 1];
signB = [1 1 1 1 0 0 1 1 1 1];

```

```

elseif (sextante == 4)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta+Tb/2+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeB=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeA=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeC=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signB = [0 0 1 1 1 1 1 1 0 0];
signA = [0 0 0 1 1 1 1 0 0 0];
signC = [0 0 0 0 1 1 0 0 0 0];

```

```

elseif (sextante == 5)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta/2+Tb+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeB=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeC=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeA=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signB = [0 0 1 1 1 1 1 1 0 0];
signC = [0 0 0 1 1 1 1 0 0 0];
signA = [0 0 0 0 1 1 0 0 0 0];

elseif (sextante == 6)
    t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta+Tb/2+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

    timeA=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeC=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeB=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signA = [1 1 0 0 0 0 0 0 1 1];
    signC = [1 1 1 0 0 0 0 1 1 1];
    signB = [1 1 1 1 0 0 1 1 1 1];

elseif (sextante == 7)
    t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta/2+Tb+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

    timeA=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeB=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeC=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signA = [1 1 0 0 0 0 0 0 1 1];
    signB = [1 1 1 0 0 0 0 1 1 1];
    signC = [1 1 1 1 0 0 1 1 1 1];

elseif (sextante == 8)
    t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta+Tb/2+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

    timeC=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeB=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeA=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signC = [0 0 1 1 1 1 1 1 0 0];
    signB = [0 0 0 1 1 1 1 0 0 0];
    signA = [0 0 0 0 1 1 0 0 0 0];

elseif (sextante == 9)
    t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta/2+Tb+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

    timeC=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
    timeA=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
    timeB=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

    signC = [0 0 1 1 1 1 1 1 0 0];
    signA = [0 0 0 1 1 1 1 0 0 0];
    signB = [0 0 0 0 1 1 0 0 0 0];

elseif (sextante == 10)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta+Tb/2+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeB=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeA=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeC=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signB = [1 1 0 0 0 0 0 0 1 1];
signA = [1 1 1 0 0 0 0 1 1 1];
signC = [1 1 1 1 0 0 1 1 1 1];

```

```

elseif (sextante == 11)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Ta/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta/2+Tb+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeB=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeC=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeA=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signB = [1 1 0 0 0 0 0 0 1 1];
signC = [1 1 1 0 0 0 0 1 1 1];
signA = [1 1 1 1 0 0 1 1 1 1];

```

```

elseif (sextante == 12)

```

```

t0=0; t1= Tc/4; t2=Tc/4+Tb/2; t3=Tc/4+Ta/2+Tb/2;
t4=Tc/4+Ta/2+Tb/2+Tc/2; t5=Tc/4+Ta+Tb/2+Tc/2; t6=Tc/4+Ta+Tb+Tc/2;
t7=Ts;

```

```

timeA=[t0 t1 t1+e t2 t3 t4 t5 t6 t6+e t7];
timeC=[t0 t1 t2 t2+e t3 t4 t5 t5+e t6 t7];
timeB=[t0 t1 t2 t3 t3+e t4 t4+e t5 t6 t7];

```

```

signA = [0 0 1 1 1 1 1 1 0 0];
signC = [0 0 0 1 1 1 1 0 0 0];
signB = [0 0 0 0 1 1 0 0 0 0];

```

```

end

```

```

timeAtotal=[timeAtotal, timeA+(j-1)*Ts.*ones(1,10)];
timeBtotal=[timeBtotal, timeB+(j-1)*Ts.*ones(1,10)];
timeCtotal=[timeCtotal, timeC+(j-1)*Ts.*ones(1,10)];
signAtotal=[signAtotal, signA];
signBtotal=[signBtotal, signB];
signCtotal=[signCtotal, signC];

```

```

estado=2;

```

```

end
end

```

```

control=[control,[estado estado]];
tiempocontrol=[tiempocontrol,[(j-1)*Ts j*Ts]];

```

```

end

```

```

comp=isempty(timeAtotal);

```

```

if comp==1

```

```

timeAtotal=[0 1];
timeBtotal=[0 1];

```



```

    timeCtotal=[0 1];
    signAtotal=[0 1];
    signBtotal=[0 1];
    signCtotal=[0 1];
end

assignin('base','timeAtotal',floor(timeAtotal*1e5)/1e5);
assignin('base','timeBtotal',floor(timeBtotal*1e5)/1e5);
assignin('base','timeCtotal',floor(timeCtotal*1e5)/1e5);
assignin('base','signAtotal',floor(signAtotal*1e5)/1e5);
assignin('base','signBtotal',floor(signBtotal*1e5)/1e5);
assignin('base','signCtotal',floor(signCtotal*1e5)/1e5);

assignin('base','control',control);
assignin('base','tiempocontrol',tiempocontrol);

```

## **Ejemplos código Matlab interfaz gráfica GUIDE**

### **Función Pantalla Inicio**

```

function basica_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

```

```

delete handles.mat
delete workspace.mat

```

```

guidata(hObject, handles);

```

```

function varargout = basica_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

```

```

function monof_Callback(hObject, eventdata, handles)
addpath 'Interfaz Grafica Monofasico'
addpath 'Inversor Monofásico'
addpath 'Copias seguridad'
addpath 'Teoria'
addpath 'Imágenes Barrido'
addpath 'Imágenes'
addpath 'Videotutoriales'

```

CalculadorMonofasico

```

function Trif_Callback(hObject, eventdata, handles)

```

```

addpath 'Interfaz Grafica Trifasico'
addpath 'Inversor Trifasico'
addpath 'Copias seguridad'
addpath 'Teoria'
addpath 'Imágenes Barrido'
addpath 'Imágenes'
addpath 'Videotutoriales'

```

CalculadorTrifasico

---

### **Botón “Calcular” en pantalla Inversores Monofásicos**

```

function Calcular_Callback(hObject, eventdata, handles)

Vi = str2double (get(handles.inputvi, 'string'));
assignin('base', 'Vi', Vi);
RI = str2double (get(handles.inputri, 'string'));
assignin('base', 'RI', RI);
CI = str2double (get(handles.inputci, 'string'));
assignin('base', 'CI', CI);
R = str2double (get(handles.inputr, 'string'));
assignin('base', 'R', R);
L = str2double (get(handles.inputl, 'string'));
assignin('base', 'L', L);
ma = str2double (get(handles.inputma, 'string'));
assignin('base', 'ma', ma);
mf = str2double (get(handles.inputmf, 'string'));
assignin('base', 'mf', mf);
frec = str2double (get(handles.inputf, 'string'));
assignin('base', 'frec', frec);
Ron = str2double (get(handles.inputron, 'string'));
assignin('base', 'Ron', Ron);
td = str2double (get(handles.td, 'string'));
assignin('base', 'td', td);
alpha = str2double (get(handles.alpha, 'string'));
assignin('base', 'alpha', alpha);
N = str2double (get(handles.inputN, 'string'));
assignin('base', 'N', N);

unip = get(handles.Unipolar, 'Value');
bip = get(handles.Bipolar, 'Value');
cancel = get(handles.Ctension, 'Value');
pushp= get(handles.Pushpull, 'Value');

if unip==1

    sim 'ModulacionUnipolar.mdl';
    close figure 1;

    set(handles.outputpi, 'String', Pi)
    set(handles.outputiR_rms, 'String', iR_rms)
    set(handles.outputiC_rms, 'String', iC_rms)
    set(handles.outputvo, 'String', Vo)
    set(handles.outputiopp, 'String', iopicopico)
    set(handles.outputpo, 'String', Po)
    set(handles.outputthdvo, 'String', THDvo)
    set(handles.outputthdio, 'String', THDdio)
    modulacion = str2double('2');
    assignin('base', 'modulacion', modulacion);

    save workspace

elseif bip==1

    sim 'ModulacionBipolar.mdl';
    close figure 1;

    set(handles.outputpi, 'String', Pi)
    set(handles.outputiR_rms, 'String', iR_rms)
    set(handles.outputiC_rms, 'String', iC_rms)
    set(handles.outputvo, 'String', Vo)
    set(handles.outputiopp, 'String', iopicopico)

```

```

set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvo)
set(handles.outputthdio, 'String', THDio)
modulacion = str2double('1');
assignin('base', 'modulacion', modulacion);
save workspace

elseif cancel==1

sim 'CancelacionTension.mdl';
close figure 1;

set(handles.outputpi, 'String', Pi)
set(handles.outputiR_rms, 'String', iR_rms)
set(handles.outputiC_rms, 'String', iC_rms)
set(handles.outputvo, 'String', Vo)
set(handles.outputiopp, 'String', iopicopico)
set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvo)
set(handles.outputthdio, 'String', THDio)
modulacion = str2double('3');
assignin('base', 'modulacion', modulacion);
save workspace

elseif pushp==1

sim 'PushPull.mdl';
close figure 1;

set(handles.outputpi, 'String', Pi)
set(handles.outputiR_rms, 'String', iR_rms)
set(handles.outputiC_rms, 'String', iC_rms)
set(handles.outputvo, 'String', Vo)
set(handles.outputiopp, 'String', iopicopico)
set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvo)
set(handles.outputthdio, 'String', THDio)
modulacion = str2double('4');
assignin('base', 'modulacion', modulacion);
save workspace

end

```

---

### Botón “Valores por defecto” en pantalla Inverosores Monofásicos

```

function defect_Callback(hObject, eventdata, handles)

set(handles.inputvi, 'string', '100');
set(handles.inputci, 'string', '0.001');
set(handles.inputri, 'string', '0.1');
set(handles.inputr, 'string', '10');
set(handles.inputl, 'string', '0.01');
set(handles.inputma, 'string', '0.8');
set(handles.inputmf, 'string', '15');
set(handles.inputf, 'string', '50');
set(handles.inputtron, 'string', '0.001');
set(handles.td, 'string', '0');
set(handles.alpha, 'string', '0');
set(handles.inputN, 'string', '1');

```

```

unip = get(handles.Unipolar, 'Value');
bip = get(handles.Bipolar, 'Value');
cancel = get(handles.Ctension, 'Value');
pushp = get(handles.Pushpull, 'Value');

if unip==1
    set(handles.inputmf, 'string', '14');
elseif bip ==1
    set(handles.inputmf, 'string', '15');
elseif cancel==1
    set(handles.inputmf, 'string', '0');
elseif pushp==1
    set(handles.inputmf, 'string', '15');
end

```

---

### Selección de modulación en pantalla Inversores Monofásicos

```

function uipanel5_SelectionChangeFcn(hObject, eventdata, handles)

newButton=get(eventdata.NewValue, 'tag');
switch newButton
case 'Unipolar'
    img=imread('Imagen1.png');
    imshow(img);

    set(handles.alpha, 'visible', 'off');
    set(handles.alpha, 'string', '0');

    set(handles.N, 'visible', 'off');
    set(handles.inputN, 'string', '0');
    set(handles.inputN, 'visible', 'off');

    set(handles.ma, 'string', 'Modulación Amplitud ma');
    set(handles.inputma, 'visible', 'on');
    set(handles.inputma, 'string', ' ');

    set(handles.mf, 'visible', 'on');
    set(handles.inputmf, 'visible', 'on');
    set(handles.inputmf, 'string', ' ');

    set(handles.f1, 'string', 'Frecuencia f1 (Hz)')

    set(handles.inputvi, 'string', ' ');
    set(handles.inputci, 'string', ' ');
    set(handles.inputri, 'string', ' ');
    set(handles.inputr, 'string', ' ');
    set(handles.inputl, 'string', ' ');
    set(handles.inputma, 'string', ' ');
    set(handles.inputmf, 'string', ' ');
    set(handles.inputf, 'string', ' ');
    set(handles.inputron, 'string', ' ');
    set(handles.td, 'string', ' ');
    set(handles.alpha, 'string', ' ');

    set(handles.outputpi, 'String', ' ');
    set(handles.outputiR_rms, 'String', ' ');
    set(handles.outputiC_rms, 'String', ' ');

```

```

set(handles.outputvo,'String',' ');
set(handles.outputiopp,'String',' ');
set(handles.outputpo,'String',' ');
set(handles.outputthdvo,'String',' ');
set(handles.outputthdio,'String',' ');

set(handles.repparam,'string','Representación Vol=f(ma) ');

case 'Bipolar'
img=imread('Imagen1.png');
imshow(img);

set(handles.alpha,'visible','off');
set(handles.alpha,'string','0');

set(handles.N,'visible','off');
set(handles.inputN,'string','0');
set(handles.inputN,'visible','off');

set(handles.ma,'string','Modulación Amplitud ma');
set(handles.inputma,'visible','on');
set(handles.inputma,'string',' ');

set(handles.mf,'visible','on');
set(handles.inputmf,'visible','on');
set(handles.inputmf,'string',' ');

set(handles.f1,'string','Frecuencia f1 (Hz)')

set(handles.inputvi,'string',' ');
set(handles.inputci,'string',' ');
set(handles.inputtri,'string',' ');
set(handles.inputr,'string',' ');
set(handles.inputl,'string',' ');
set(handles.inputma,'string',' ');
set(handles.inputmf,'string',' ');
set(handles.inputf,'string',' ');
set(handles.inputron,'string',' ');
set(handles.td,'string',' ');
set(handles.alpha,'string',' ');

set(handles.outputpi,'String',' ');
set(handles.outputiR_rms,'String',' ');
set(handles.outputiC_rms,'String',' ');
set(handles.outputvo,'String',' ');
set(handles.outputiopp,'String',' ');
set(handles.outputpo,'String',' ');
set(handles.outputthdvo,'String',' ');
set(handles.outputthdio,'String',' ');

set(handles.repparam,'string','Representación Vol=f(ma) ');

case 'Ctension'
img=imread('Imagen1.png');
imshow(img);

set(handles.alpha,'visible','on');
set(handles.alpha,'string',' ');

```

```

set(handles.N,'visible','off');
set(handles.inputN,'string','0');
set(handles.inputN,'visible','off');

set(handles.ma,'string','Alpha (deg)');
set(handles.inputma,'visible','off');
set(handles.inputma,'string','0');

set(handles.mf,'visible','off');
set(handles.inputmf,'visible','off');
set(handles.inputmf,'string','0');

set(handles.f1,'string','Frecuencia Conmutación (Hz)')

set(handles.inputvi,'string',' ');
set(handles.inputci,'string',' ');
set(handles.inputri,'string',' ');
set(handles.inputr,'string',' ');
set(handles.inputl,'string',' ');
set(handles.inputma,'string',' ');
set(handles.inputmf,'string',' ');
set(handles.inputf,'string',' ');
set(handles.inputron,'string',' ');
set(handles.td,'string',' ');
set(handles.alpha,'string',' ');

set(handles.outputpi,'String',' ')
set(handles.outputiR_rms,'String',' ')
set(handles.outputiC_rms,'String',' ')
set(handles.outputvo,'String',' ')
set(handles.outputiopp,'String',' ')
set(handles.outputpo,'String',' ')
set(handles.outputthdvo,'String',' ')
set(handles.outputthdio,'String',' ')

set(handles.repparam,'string','Representación Vol=f(alpha)');

case 'Pushpull'
img=imread('Imagen2.png');
imshow(img);

set(handles.ma,'string','Modulación Amplitud ma');
set(handles.inputma,'visible','on');
set(handles.inputma,'string',' ');

set(handles.mf,'visible','on');
set(handles.inputmf,'visible','on');
set(handles.inputmf,'string',' ');

set(handles.alpha,'visible','off');
set(handles.alpha,'string','0');

set(handles.N,'visible','on');
set(handles.inputN,'visible','on');
set(handles.inputN,'string',' ');

set(handles.f1,'string','Frecuencia f1 (Hz)')

```

```

set(handles.inputvi,'string',' ');
set(handles.inputci,'string',' ');
set(handles.inputri,'string',' ');
set(handles.inputr,'string',' ');
set(handles.inputl,'string',' ');
set(handles.inputma,'string',' ');
set(handles.inputmf,'string',' ');
set(handles.inputf,'string',' ');
set(handles.inputron,'string',' ');
set(handles.td,'string',' ');
set(handles.alpha,'string',' ');

set(handles.outputpi,'String',' ');
set(handles.outputiR_rms,'String',' ');
set(handles.outputiC_rms,'String',' ');
set(handles.outputvo,'String',' ');
set(handles.outputiopp,'String',' ');
set(handles.outputpo,'String',' ');
set(handles.outputthdvo,'String',' ');
set(handles.outputthdio,'String',' ');

set(handles.repparam,'string','Representación Vol=f(ma) ');
end

```

---

### Botón “Restaurar” en pantalla Inversores Monofásicos

```

function restaurar_Callback(hObject, eventdata, handles)

unip = get(handles.Unipolar,'Value');
bip = get(handles.Bipolar,'Value');
cancel = get(handles.Ctension,'Value');
pushp= get(handles.Pushpull,'Value');

if unip==1
    open_system('ModulacionUnipolar_fijo');
    save_system('ModulacionUnipolar_fijo','Inversor
Monofásico/ModulacionUnipolar')
    close_system('ModulacionUnipolar_fijo')
    close_system('ModulacionUnipolar')
elseif bip==1
    open_system('ModulacionBipolar_fijo');
    save_system('ModulacionBipolar_fijo','Inversor
Monofásico/ModulacionBipolar')
    close_system('ModulacionBipolar_fijo')
    close_system('ModulacionBipolar')
elseif cancel==1
    open_system('CancelacionTension_fijo');
    save_system('CancelacionTension_fijo','Inversor
Monofásico/CancelacionTension')
    close_system('CancelacionTension_fijo')
    close_system('CancelacionTension')
elseif pushp==1
    open_system('PushPull_fijo');
    save_system('PushPull_fijo','Inversor Monofásico/PushPull')
    close_system('PushPull_fijo')
    close_system('PushPull')
end

```

---

**Botón “Teoría” en pantalla Inversores Monofásicos**

```
function teoria_Callback(hObject, eventdata, handles)
open ('Inversores Monofasicos.pdf')
```

---

**Botón “Videotutorial” en pantalla Inversores Monofásicos**

```
function pushbutton14_Callback(hObject, eventdata, handles)
winopen ('Videotutoriales/videotutorial monofásico.mp4')
```

---

**Desplegable en representación en Inversores Monofásicos**

```
function mostrar_Callback(hObject, eventdata, handles)
slide=get(handles.menu, 'Value');

save handles
load workspace.mat
load handles.mat

if slide==1
    axes(handles.axes4);
    plot(0.0)

    if modulacion==1
        axes(handles.axes1)

plot(PWM.time,PWM.signals.values(:,2), 'r', PWM.time,PWM.signals.values(
:,1), 'b')
    legend('señal portadora','señal moduladora')
    ylim([-1.2 1.2])
    ylabel ('Amplitud')
    set(handles.axes1, 'ytick', [])

    axes(handles.axes2);
    plot(tatb.time,tatb.signals.values(:,1), 'g')
    legend('TA+')
    ylabel ('ON/OFF')
    ylim([-0.2 1.2])
    set(handles.axes2, 'ytick', [])

    axes(handles.axes3);
    plot(tatb.time,tatb.signals.values(:,2), 'm')
    legend('TB+')
    ylabel ('ON/OFF')
    xlabel ('Tiempo (s)')
    ylim([-0.2 1.2])
    set(handles.axes3, 'ytick', [])

    set(handles.axes3, 'visible', 'on');

elseif modulacion==2
```



```

        axes (handles.axes1)

plot(PWM.time,PWM.signals.values(:,2),'r',PWM.time,PWM.signals.values(
(:,1),'b',PWM.time,PWM.signals.values(:,3),'k')
    legend('señal portadora','señal moduladora','señal negada')
    ylim([-1.2 1.2])
    ylabel ('Amplitud')

    axes(handles.axes2);
    plot(tatb.time,tatb.signals.values(:,1),'g')
    legend('TA+')
    ylabel ('ON/OFF')
    ylim([-0.2 1.2])
    set(handles.axes2, 'ytick', [])

    axes(handles.axes3);
    plot(tatb.time,tatb.signals.values(:,2),'m')
    legend('TB+')
    ylabel ('ON/OFF')
    xlabel ('Tiempo (s)')
    ylim([-0.2 1.2])
    set(handles.axes3, 'ytick', [])

    set(handles.axes3, 'visible', 'on');

elseif modulacion==3
    axes(handles.axes3);
    plot(0.0)

    axes(handles.axes1);
    plot(tatb.time,tatb.signals.values(:,1),'g')
    legend('TA+')
    ylabel ('ON/OFF')
    ylim([-0.2 1.2])
    set(handles.axes1, 'ytick', [])

    axes(handles.axes2);
    plot(tatb.time,tatb.signals.values(:,2),'m')
    legend('TB+')
    ylabel ('ON/OFF')
    xlabel ('Tiempo (s)')
    ylim([-0.2 1.2])
    set(handles.axes2, 'ytick', [])

    set(handles.axes3, 'visible', 'off');
    set(handles.titulo3, 'visible', 'off');

elseif modulacion==4

    axes (handles.axes1)

plot(PWM.time,PWM.signals.values(:,1),'r',PWM.time,PWM.signals.values(
(:,2),'b')
    legend('señal portadora','señal moduladora')
    ylim([-1.2 1.2])
    ylabel ('Amplitud')

    axes(handles.axes2);
    plot(tatb.time,tatb.signals.values(:,1),'g')

```

```

    legend('TA+')
    ylabel ('ON/OFF')
    ylim([-0.2 1.2])
    set(handles.axes2, 'ytick', [])

    axes(handles.axes3);
    plot(tatb.time,tatb.signals.values(:,2), 'm')
    legend('TB+')
    ylabel ('ON/OFF')
    xlabel ('Tiempo (s)')
    ylim([-0.2 1.2])
    set(handles.axes3, 'ytick', [])

    set(handles.axes3, 'visible', 'on');
end

set(handles.axes1, 'visible', 'on');
set(handles.axes2, 'visible', 'on');
set(handles.axes3, 'visible', 'on');
set(handles.axes4, 'visible', 'off');

set(handles.titulo1, 'string', 'PWM');
set(handles.titulo2, 'string', 'TA+(t)');
set(handles.titulo3, 'string', 'TB+(t)');
set(handles.titulo4, 'visible', 'off');

linkaxes([handles.axes1 handles.axes2 handles.axes3], 'x');

elseif slide==2
    axes(handles.axes3);
    plot(0.0)

    axes(handles.axes4);
    plot(0.0)

    axes(handles.axes1)

plot(vovo1.time,vovo1.signals.values(:,2), 'r',vovo1.time,vovo1.signals
.values(:,1), 'b')
    legend('vo', 'vol')
    ylabel ('Tensión (V)')

    axes(handles.axes2);

plot(ioio1.time,ioio1.signals.values(:,1), 'r',ioio1.time,ioio1.signals
.values(:,2), 'b')
    legend('io', 'io1')
    xlabel ('Tiempo (s)')
    ylabel ('Amperios (A)')

set(handles.axes1, 'visible', 'on');
set(handles.axes2, 'visible', 'on');
set(handles.axes3, 'visible', 'off');
set(handles.axes4, 'visible', 'off');

set(handles.titulo1, 'string', 'vo(t)');
set(handles.titulo2, 'string', 'io(t)');

```

```

set(handles.titulo3,'visible','off');
set(handles.titulo4,'visible','off');

linkaxes([handles.axes1 handles.axes2], 'x');

elseif slide==3
axes(handles.axes4);
plot(0.0)

axes(handles.axes1);
plot(ioio1.time,ioio1.signals.values(:,1),'r')
legend('io')
ylabel ('Amperios (A)')

axes(handles.axes2);
plot(ioio1.time,ioio1.signals.values(:,2),'b')
legend('io1')
ylabel ('Amperios (A)')

axes(handles.axes3);
plot(ioio1.time,ioio1.signals.values(:,3),'g')
legend('io ripple')
ylabel ('Amperios (A)')
xlabel ('Tiempo (s)')

set(handles.axes1,'visible','on');
set(handles.axes2,'visible','on');
set(handles.axes3,'visible','on');
set(handles.axes4,'visible','off');

set(handles.titulo3,'visible','on');

set(handles.titulo1,'string','io(t)');
set(handles.titulo2,'string','io1(t)');
set(handles.titulo3,'string','ioripple(t)');
set(handles.titulo4,'visible','off');

linkaxes([handles.axes1 handles.axes2 handles.axes3], 'x');

elseif slide==4

axes(handles.axes1);
plot(iI1.time,iI1.signals.values(:,1),'r')
legend('iI')
ylabel ('Amperios (A)')

axes(handles.axes2);
plot(iC.time,iC.signals.values(:,1),'g')
legend('iC_I')
ylabel ('Amperios (A)')

axes(handles.axes3);
plot(iR.time,iR.signals.values(:,1),'m')
legend('iR_I')
ylabel ('Amperios (A)')

axes(handles.axes4);
plot(ioio1.time,ioio1.signals.values(:,1),'b')
legend('iO')

```

```

ylabel ('Amperios (A)')
xlabel ('Tiempo (s)')

set(handles.axes1,'visible','on');
set(handles.axes2,'visible','on');
set(handles.axes3,'visible','on');
set(handles.axes4,'visible','on');

set(handles.titulo3,'visible','on');
set(handles.titulo4,'visible','on');

set(handles.titulo1,'string','iI(t)');
set(handles.titulo2,'string','iC_I(t)');
set(handles.titulo3,'string','iR_I(t)');
set(handles.titulo4,'string','iO(t)');

linkaxes([handles.axes1 handles.axes2 handles.axes3
handles.axes4], 'x');

elseif slide==5

axes(handles.axes3);
plot(0.0)

axes(handles.axes4);
plot(0.0)

axes(handles.axes1)
bar((0:200),fftvo.signals.values(1:201,:,2),'r')
xlim([0 200])
ylabel ('Tensión (V)')

axes(handles.axes2);
bar((0:200),fftio.signals.values(1:201,:,2),'b')
xlim([0 200])
ylabel ('Amperios (A)')
xlabel ('N° Armónico')

set(handles.axes1,'visible','on');
set(handles.axes2,'visible','on');
set(handles.axes3,'visible','off');
set(handles.axes4,'visible','off');

set(handles.titulo1,'string','FFT vO(t)');
set(handles.titulo2,'string','FFT iO(t)');
set(handles.titulo3,'visible','off');
set(handles.titulo4,'visible','off');

linkaxes([handles.axes1 handles.axes2], 'x');

elseif slide==6

if modulacion==1||modulacion==2||modulacion==3

axes(handles.axes1)
plot(tatb.time,tatb.signals.values(:,1),'r')
legend('TA+')

```

```

ylabel ('ON/OFF')
ylim([-0.2 1.2])
set(handles.axes1, 'ytick', [])

axes(handles.axes2);
plot(tatb.time,tatb.signals.values(:,3), 'b')
legend('TA-')
ylabel ('ON/OFF')
ylim([-0.2 1.2])
set(handles.axes2, 'ytick', [])

axes(handles.axes3);
plot(tatb.time,tatb.signals.values(:,2), 'g')
legend('TB+')
ylabel ('ON/OFF')
ylim([-0.2 1.2])
set(handles.axes3, 'ytick', [])

axes(handles.axes4);
plot(tatb.time,tatb.signals.values(:,4), 'm')
legend('TB-')
ylabel ('ON/OFF')
xlabel ('Tiempo (s)')
ylim([-0.2 1.2])
set(handles.axes4, 'ytick', [])

set(handles.axes3, 'visible', 'on');
set(handles.axes1, 'visible', 'on');
set(handles.axes2, 'visible', 'on');
set(handles.axes3, 'visible', 'on');
set(handles.axes4, 'visible', 'on');

set(handles.titulo3, 'visible', 'on');
set(handles.titulo4, 'visible', 'on');

set(handles.titulo1, 'string', 'TA+');
set(handles.titulo2, 'string', 'TA-');
set(handles.titulo3, 'string', 'TB+');
set(handles.titulo4, 'string', 'TB-');

linkaxes([handles.axes1 handles.axes2 handles.axes3
handles.axes4], 'x');

elseif modulacion==4

axes(handles.axes3);
plot(0.0)

axes(handles.axes4);
plot(0.0)

axes(handles.axes1)
plot(tatb.time,tatb.signals.values(:,1), 'r')
legend('TA+')
ylabel ('ON/OFF')
ylim([-0.2 1.2])
set(handles.axes1, 'ytick', [])

axes(handles.axes2);

```

```

plot(tatb.time,tatb.signals.values(:,2), 'g')
legend('TB+')
ylabel ('ON/OFF')
xlabel ('Tiempo (s)')
ylim([-0.2 1.2])
set(handles.axes2, 'ytick', [])

set(handles.axes3, 'visible', 'on');
set(handles.axes1, 'visible', 'on');
set(handles.axes2, 'visible', 'on');
set(handles.axes3, 'visible', 'off');
set(handles.axes4, 'visible', 'off');

set(handles.titulo3, 'visible', 'off');
set(handles.titulo4, 'visible', 'off');

set(handles.titulo1, 'string', 'TA+');
set(handles.titulo2, 'string', 'TB+');

linkaxes([handles.axes1 handles.axes2], 'x');
end
end

```

---

### Botón “Calcular” en pantalla Inversores Trifásicos

```

function Calcular_Callback(hObject, eventdata, handles)

Vi = str2double (get(handles.inputvi, 'string'));
assignin('base', 'Vi', Vi);
RI = str2double (get(handles.inputri, 'string'));
assignin('base', 'RI', RI);
CI = str2double (get(handles.inputci, 'string'));
assignin('base', 'CI', CI);
R = str2double (get(handles.inputr, 'string'));
assignin('base', 'R', R);
L = str2double (get(handles.inputl, 'string'));
assignin('base', 'L', L);
ma = str2double (get(handles.inputma, 'string'));
assignin('base', 'ma', ma);
mf = str2double (get(handles.inputmf, 'string'));
assignin('base', 'mf', mf);
frec = str2double (get(handles.inputf, 'string'));
assignin('base', 'frec', frec);
Ron = str2double (get(handles.inputron, 'string'));
assignin('base', 'Ron', Ron);
td = str2double (get(handles.td, 'string'));
assignin('base', 'td', td);

pwm = get(handles.PWM, 'Value');
terc = get(handles.Tercer, 'Value');
esp = get(handles.Vespaciales, 'Value');
disc= get(handles.Discontinua, 'Value');
vect= get(handles.Vectorial, 'Value');

if pwm==1
    sim 'ModulacionPWM.mdl';
    close figure 1;

```

```

set(handles.outputpi, 'String', Pi)
set(handles.outputiR_rms, 'String', iR_rms)
set(handles.outputiC_rms, 'String', iC_rms)
set(handles.outputvo, 'String', vABlrms)
set(handles.outputiopp, 'String', iApicopico)
set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvAB)
set(handles.outputthdio, 'String', THDiA)
modulacion = str2double('1');
assignin('base', 'modulacion', modulacion);

save workspace

elseif terc==1
slide=get(handles.rel, 'Value');
if slide==1
    assignin('base', 'rel', 4);
elseif slide==2
    assignin('base', 'rel', 6);
end
sim 'ModulacionTercerArmonico.mdl';
close figure 1;

set(handles.outputpi, 'String', Pi)
set(handles.outputiR_rms, 'String', iR_rms)
set(handles.outputiC_rms, 'String', iC_rms)
set(handles.outputvo, 'String', vABlrms)
set(handles.outputiopp, 'String', iApicopico)
set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvAB)
set(handles.outputthdio, 'String', THDiA)
modulacion = str2double('2');
assignin('base', 'modulacion', modulacion);
save workspace

elseif esp==1
sim 'ModulacionSecuenciaCero.mdl';
close figure 1;

set(handles.outputpi, 'String', Pi)
set(handles.outputiR_rms, 'String', iR_rms)
set(handles.outputiC_rms, 'String', iC_rms)
set(handles.outputvo, 'String', vABlrms)
set(handles.outputiopp, 'String', iApicopico)
set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvAB)
set(handles.outputthdio, 'String', THDiA)
modulacion = str2double('3');
assignin('base', 'modulacion', modulacion);
save workspace

elseif disc==1
sim 'ModulacionDiscontinua.mdl';
close figure 1;

set(handles.outputpi, 'String', Pi)
set(handles.outputiR_rms, 'String', iR_rms)
set(handles.outputiC_rms, 'String', iC_rms)
set(handles.outputvo, 'String', vABlrms)

```

```

set(handles.outputiopp, 'String', iApicopico)
set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvAB)
set(handles.outputthdio, 'String', THDiA)
modulacion = str2double('4');
assignin('base', 'modulacion', modulacion);
save workspace

elseif vect==1
slide=get(handles.modovect, 'Value');
if slide==1
run 'ModulacionVectorial';
elseif slide==2
run 'ModulacionVectorialPares';
end
sim 'ModulacionVectorialE.mdl';
close figure 1;

set(handles.outputpi, 'String', Pi)
set(handles.outputiR_rms, 'String', iR_rms)
set(handles.outputiC_rms, 'String', iC_rms)
set(handles.outputvo, 'String', vABlrms)
set(handles.outputiopp, 'String', iApicopico)
set(handles.outputpo, 'String', Po)
set(handles.outputthdvo, 'String', THDvAB)
set(handles.outputthdio, 'String', THDiA)
modulacion = str2double('5');
assignin('base', 'modulacion', modulacion);
save workspace
end

```

---

### Botón “Borrar” en pantalla Inversores Trifásicos

```

function Borrar_Callback(hObject, eventdata, handles)

set(handles.inputvi, 'string', ' ');
set(handles.inputci, 'string', ' ');
set(handles.inputri, 'string', ' ');
set(handles.inputr, 'string', ' ');
set(handles.inputl, 'string', ' ');
set(handles.inputma, 'string', ' ');
set(handles.inputmf, 'string', ' ');
set(handles.inputf, 'string', ' ');
set(handles.inputron, 'string', ' ');
set(handles.td, 'string', ' ');
set(handles.outputpi, 'String', ' ')
set(handles.outputiR_rms, 'String', ' ')
set(handles.outputiC_rms, 'String', ' ')
set(handles.outputvo, 'String', ' ')
set(handles.outputiopp, 'String', ' ')
set(handles.outputpo, 'String', ' ')
set(handles.outputthdvo, 'String', ' ')
set(handles.outputthdio, 'String', ' ')

```

---

### Botón “Descargar en Placa” en pantalla Inversores Trifásicos

```

function DescPlaca_Callback(hObject, eventdata, handles)

```



```
helpdlg('Esta funcionalidad no está implementada', 'Aviso');
```

---

### Botón “Abrir” en pantalla Inversores Trifásicos

```
function abrir_Callback(hObject, eventdata, handles)

pwm = get(handles.PWM, 'Value');
terc = get(handles.Tercer, 'Value');
esp = get(handles.Vespaciales, 'Value');
disc= get(handles.Discontinua, 'Value');
vect= get(handles.Vectorial, 'Value');

if pwm==1
    open ('ModulacionPWM.mdl')
elseif terc==1
    open ('ModulacionTercerArmonico.mdl')
elseif esp==1
    open ('ModulacionSecuenciaCero.mdl')
elseif disc==1
    open ('ModulacionDiscontinua.mdl')
elseif vect==1
    open ('ModulacionVectorialE.mdl')
end
```

---

### Botón “Representación” en pantalla Inversores Trifásicos

```
function representacion_Callback(hObject, eventdata, handles)
pwm = get(handles.PWM, 'Value');
terc = get(handles.Tercer, 'Value');
esp = get(handles.Vespaciales, 'Value');
disc= get(handles.Discontinua, 'Value');
vect= get(handles.Vectorial, 'Value');

if pwm==1||terc==1||disc==1||esp==1
    representaciontrif
elseif vect==1
    representaciontrifvect
end
```

---

### Pantalla de botón “Representación VAB1=f(ma)” en pantalla Inversores Trifásicos

```
function ejecutar_Callback(hObject, eventdata, handles)

def = get(handles.defecto, 'Value');
cal = get(handles.calcular, 'Value');

save handles
load workspace.mat
load handles.mat

if def==1
    if modulacion==1
        open ('BarridoPWM.fig')
    elseif modulacion==2
```

```

        open ('BarridoTercArm.fig')
    elseif modulacion==3
        open ('BarridoSecCero.fig')
    elseif modulacion==4
        open ('BarridoDisc.fig')
    elseif modulacion==5
        open ('BarridoVectorial.fig')
    end

elseif cal==1
    ma = str2double (get(handles.inicial,'string'));
    assignin('base','ma', ma);

    inicial=str2double (get(handles.inicial,'string'));
    final= str2double (get(handles.final,'string'));
    paso=str2double (get(handles.paso,'string'));
    for i=1:(final-inicial)/paso+1
        if modulacion==1
            sim 'ModulacionPWM.mdl';
        elseif modulacion==2
            sim 'ModulacionTercerArmonico.mdl';
        elseif modulacion==3
            sim 'ModulacionSecuenciaCero.mdl';
        elseif modulacion==4
            sim 'ModulacionDiscontinua.mdl';
        elseif modulacion==5
            run 'ModulacionVectorial';
            sim 'ModulacionVectorialE.mdl';
        end
        armonico(i,1)=armonico;
        Ciclos(i,1)=ma;
        ma=ma+paso;
        assignin('base','ma', ma);
    end
    close figure 1;
    figure(1);
    plot(Ciclos,armonico)
    grid on;
    ylabel ('VLLrms/Vi')
    xlabel ('ma')
    title('Variación primer armónico VAB=f(ma)')
end

```

Parte VI

# Manual de usuario



# Manual de usuario

Como Manual de usuario de la herramienta se ha realizado un videotutorial explicativo de la misma.

El videotutorial consta de dos partes, la primera de ellas explica la parte de la interfaz gráfica de los inversores monofásicos, la segunda parte explica la interfaz gráfica asociada a los inversores trifásicos.

Ambos videos se encuentran disponibles en la carpeta *Videotutoriales* dentro del directorio raíz de la herramienta.

A su vez se puede acceder a ambos videotutoriales desde la herramienta pulsando el botón “**Videotutorial**” de cada una de sus ventanas correspondientes.



Parte VII

## Bibliografía





# Bibliografía

- [1] M. C. Pérez Rubio, J. Ureña Ureña , D. Ruíz Pereda y E. García Nuñez, «Descripción y análisis de una experiencia de innovación en la asignatura de convertidores avanzados de potencia,» *Revista digital de investigación en docencia Caracciolos*, vol. 1, nº 1, pp. 1-12, 2013.
- [2] «Manual Matlab Simulink,» [En línea]. Available: <http://es.mathworks.com/products/simulink/>. [Último acceso: Febrero 2015].
- [3] «Manual Matlab GUI,» [En línea]. Available: <http://es.mathworks.com/discovery/matlab-gui.html>. [Último acceso: Febrero 2015].
- [4] R. B. Galindo, "*Sistema HW de control PWM gestionado desde Matlab*", Universidad de Alcalá, 2014.
- [5] M. d. I. M. Ballesteros, «Uso de programas de simulación para promover la pedagogía activa en la docencia,» *Investigación e innovación educativa*, 2014.
- [6] J. Ureña Ureña, M. Sotelo Vázquez, F. Rodríguez Sanchez, R. Barea Navarro, M. Domínguez Herranz, E. Bueno Peña y P. Revenga de Toro, *Electrónica de Potencia*, Alcalá de Henares: Universidad de Alcalá, 1999.
- [7] G. Camacho, D. Lopez , J. Díaz y C. Gaviria, «Caracterización de las técnicas de modulación Pulse Width Modulation (PWM) aplicadas a inversores trifásicos,» *Revista Épsilon*, 2012.
- [8] B. Wu, «High-Power Converters and AC Drives,» *Wiley & IEEE Press*, 2006, ISBN-13 978-0-0471-73171-9.
- [9] N. Mohan, T. M. Undeland y W. P. Robbins, *Power Electronics: Converters, Applications, and Design*; 2nd edition, John Wiley & Sons, Inc., 1995.
- [10] D. O. B. Guerrero, «MANUAL DE INTERFAZ GRÁFICA DE USUARIO EN MATLAB PARTE I,» Ecuador, 2008.

