

# **Centro Nacional de Investigación y Desarrollo Tecnológico**

**Subdirección Académica**

**Departamento de Ingeniería Electrónica**

## **TESIS DE MESTRÍA EN CIENCIAS**

**Asistente para el Análisis y Diseño de Topologías Comunes de  
Convertidores CD-CD**

presentada por  
**Ing. Eider Miguel Amores Campos**

como requisito para la obtención del grado de  
**Maestro en Ciencias en Ingeniería Electrónica**

Director de tesis  
**Dr. Carlos Aguilar Castillo**

Cuernavaca, Mor., 04 de noviembre de 2016.

OFICIO No. DIE/227/2016

**Asunto:** Aceptación de documento de tesis

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**  
**PRESENTE**

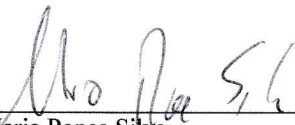
Por este conducto, los integrantes de Comité Tutorial del **C. Eider Miguel Amores Campos** con número de control **M14CE072** de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado **“ASISTENTE PARA EL ANÁLISIS Y DISEÑO DE TOPOLOGÍAS COMUNES DE CONVERTIDORES CD-CD”** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS



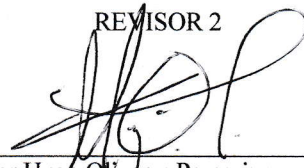
Dr. Carlos Aguilar Castillo  
Doctor en Ciencias en Ingeniería Electrónica  
Cédula profesional 3581040

REVISOR 1



Dr. Mario Ponce Silva  
Doctor en Ciencias en Ingeniería Electrónica  
Cédula profesional 3516427

REVISOR 2



Dr. Víctor Hugo Olivares Peregrino  
Doctor en Ciencias en Ingeniería Electrónica  
Cédula profesional 5999002

C.p. M.T.I. María Elena Gómez Torres- Jefa del Departamento de Servicios Escolares.  
Estudiante  
Expediente

CMAZ/lrr



Cuernavaca, Mor., 9 de noviembre de 2016

**OFICIO No. SAC/321/2016**

**Asunto:** Autorización de impresión de tesis

**ING. EIDER MIGUEL AMORES CAMPOS**  
**CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS**  
**EN INGENIERÍA ELECTRÓNICA**  
**P R E S E N T E**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **“ASISTENTE PARA EL ANÁLISIS Y DISEÑO DE TOPOLOGÍAS COMUNES DE CONVERTIDORES CD-CD”**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

**ATENTAMENTE**

“CONOCIMIENTO Y TECNOLOGIA AL SERVICIO DE MEXICO”

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**



S.E.P. - TecNM  
CENTRO NACIONAL  
DE INVESTIGACIÓN  
Y DESARROLLO  
TECNOLÓGICO  
DIRECCIÓN

C.p. M.T.I. María Elena Gómez Torres- Jefa del Departamento de Servicios Escolares.  
Expediente

GVGR/mcr

## DEDICADO A

*A mi esposa y mis hijos, por ser la motivación y gran apoyo de todo en mi vida, por el sacrificio, esfuerzo, comprensión y amor.*

*Mis padres y hermano, por el apoyo incondicional, el amor y el ejemplo que siempre me han dado.*

*A dios por guiarme y darme fuerzas para seguir adelante y no desmayar en los problemas que he vivido, enseñándome a encarar las adversidades sin perder nunca la dignidad ni fallecer en el intento.*

## **AGRADECIMIENTOS**

**A mis asesores**, el Dr. Marco Antonio Oliver Salazar y el Dr. Carlos Aguilar Castillo por el apoyo de su tiempo, paciencia y conocimiento invertido durante el desarrollo de este trabajo de tesis

**A mis revisores de tesis**, el Dr. Víctor Hugo Olivares Peregrino y el Dr. Mario Ponce Silva, por sus valiosos consejos, apoyo y orientación.

**A mis profesores**, los doctores Jesús Aguayo Alquicira, Abraham Claudio Sánchez, Jaime Eugenio Arau Roffiel, Jorge Hugo Calleja Gjumlich, Jesús Darío Mina Antonio, por el gran apoyo y por haber contribuido en mi desarrollo académico

**Al Tecnológico Nacional de México (TecNM)**, por ser parte del compromiso de la educación de calidad en México.

**Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)**, por darme la oportunidad de realizar mi formación de posgrado en el área de maestría en ciencias.

**Al Consejo Nacional de Ciencia y Tecnología (CONACYT)**, por solventar económicamente el desarrollo de esta investigación.

**Al personal administrativo** del departamento de electrónica, así como del departamento de servicios escolares, por darme las facilidades para realizar todos los trámites de titulación.

**A todos mis amigos y compañeros** de CENIDET por los gratos momentos compartidos y porque gracias a su apoyo ha sido una etapa inolvidable de mi vida.

**A mi compañero y amigo de generación**, por los buenos momentos y aventuras dentro y fuera de la escuela, y sobre todo por tu amistad.

## **Resumen.**

En esta tesis se presenta el “Asistente para el Análisis y Diseño de Topologías Comunes de Convertidores CD-CD”, el cual es una herramienta que apoya al desarrollador de convertidores CD-CD en el proceso de análisis del funcionamiento y comportamiento de operación en lazo abierto, así como en realizar el modelado del lazo de control de tensión utilizando un controlador PID, ver la respuesta del mismo y validarlo ante diferentes perturbaciones.

Se consideraba integrar al asistente con las topologías básicas Buck, Boost, Buck-Boost y Cuk. Sin embargo, se define establecer prioridad a integrar un esquema de control dentro del asistente, acotándose a un solo convertidor, con la finalidad de proporcionar al usuario una solución más completa en la implementación de convertidores y su control. Se podrán integrar más topologías, en trabajos futuros ya que el asistente se encuentra tanto en programación como en estructura de manera modular, esto es, con archivos y rutinas de ejecución de tal manera que se pueden agregar o modificar diferentes topologías y esquemas de control.

De igual manera se presenta la manera en que se realizó el asistente, la interfaz gráfica, el proceso en la manera de ejecutarlo, los resultados obtenidos del asistente, los resultados obtenidos en la implementación propuesta del asistente y la comparación de dichos resultados.

El Asistente permite obtener los valores de los elementos de la topología Buck, el comportamiento del convertidor en lazo abierto, la técnica de modulación PWM SWITCH y el comportamiento del convertidor con un controlador PID, con la ventaja de tener un despliegue de toda la información antes mencionada, siendo de gran utilidad para el diseñador de convertidores CD-CD.

Se establecieron los alcances de los resultados a mostrar del asistente, lo cuales incluyen, los valores de los elementos, los esfuerzos de corrientes y tensiones que integran la topología Buck, así como el comportamiento en lazo abierto del convertidor. También los resultados del comportamiento en pequeña señal, utilizando el método PWM SWITCH y por último, el comportamiento en lazo cerrado del convertidor con un controlador PID.

El asistente es el inicio de una herramienta que contemplará diferentes topologías, esquemas de control y diferentes opciones que sean de gran apoyo en el análisis desarrollo e implementación de convertidores CD-CD tanto en lazo abierto como en lazo cerrado, está enfocado inicialmente al apoyo de primera mano en laboratorio, para la evaluación de los convertidores en diferentes procesos en dónde se utilicen. Es por ello que el usuario de ésta herramienta debe de tener conocimientos básicos en el ámbito de los convertidores CD-CD, en

los datos básicos de diseño, en los resultados esperados, en la interpretación de las gráficas obtenidas así como en lo que refiere al esquema de control.

## **Abstract**

In this thesis the "Wizard Analysis and Design Topologies Common DC-DC converters", is presented, which is a tool that supports the developer of DC-DC converters in the analysis process of performance and open loop operating behavior, perform the model of voltage-mode control using a PID controller, see the response to it and validate different disturbances.

It was considered integrating the assistant with basic topologies Buck, Boost, Buck-Boost and Cuk. However, it defined set priority to integrate a control scheme within the wizard, narrowed to a single converter, in order to provide the user with a more complete solution in the implementation of converters and their control. They can be integrated more topologies, in future work as the assistant is both programming and structure in a modular manner, that is, with files and runtime routines so that you can add or modify different topologies and control schemes.

Similarly it is presented the way the wizard were performed, the GUI, the process on how to run it, the results of the wizard, the results obtained in implementing proposed the wizard and comparison of these results was performed.

Assistant allows to obtain the values of the elements of the Buck topology, the behavior of the converter in open loop modulation technique PWM SWITCH and behavior of the converter with a PID controller, with the advantage of having a display of all the information before mentioned, being very useful for the designer of DC-DC converters.

The scope of the results to show by the wizard were established, which include the values of the elements, the efforts of voltages and currents that make up the Buck topology and behavior in open loop converter. Also the results of behavior in small signal, using the PWM SWITCH method and finally closed loop behavior of the converter with a PID controller.

The wizard is the beginning of a tool that will consider different topologies, control schemes and different options that are of great support in the analysis development and implementation of DC-DC converters in both open and closed-loop, it is initially focused support first hand in the laboratory for evaluation of converters in different processes where used. That is why the user of this tool must have basic knowledge in the field of DC-DC converters, the basic design data, the expected results in the interpretation of the graphs obtained and when it comes to control scheme.



## CONTENIDO

LISTA DE FIGURAS .....	iii
LISTA DE TABLAS.....	vi
SIMBOLOGÍA.....	vii
ACRÓNIMOS .....	ix
<b>CAPÍTULO I. INTRODUCCIÓN.....</b>	<b>1</b>
<b>1.1 Antecedentes.....</b>	<b>2</b>
<b>1.2 Conceptos de convertidores CD-CD.....</b>	<b>2</b>
<b>1.2.1 Modos de Conducción.....</b>	<b>4</b>
<b>1.2.2 Ciclo de trabajo.....</b>	<b>5</b>
<b>1.3 Lazo de Control.....</b>	<b>6</b>
<b>1.3.1 Lazo Abierto.....</b>	<b>6</b>
<b>1.3.2 Lazo Cerrado.....</b>	<b>7</b>
<b>1.4 Planteamiento del Problema.....</b>	<b>8</b>
<b>1.5 Objetivos.....</b>	<b>9</b>
1.5.1 Objetivo general.....	10
1.5.2 Objetivos particulares.....	10
<b>1.6 Justificación.....</b>	<b>10</b>
<b>1.7 Organización de la Tesis.....</b>	<b>11</b>
<b>CAPÍTULO II. ESTADO DEL ARTE.....</b>	<b>12</b>
<b>2.1 Introducción.....</b>	<b>13</b>
<b>2.2 Consideraciones de diseño y dimensionamiento de convertidores CD-CD.....</b>	<b>13</b>
<b>2.3 Identificación de Factores Críticos en el diseño de convertidores CD-CD.....</b>	<b>13</b>
<b>2.4 Selección de dispositivos óptimos para la implementación de convertidores CD-CD.....</b>	<b>14</b>
<b>2.5 Herramientas y software de simulación para diseño de convertidores CD-CD.....</b>	<b>14</b>
<b>CAPÍTULO III. ANÁLISIS DEL CONVERTIDOR BUCK.....</b>	<b>21</b>
<b>3.1 Convertidor Reductor (Buck).....</b>	<b>22</b>
<b>3.2 Ecuaciones características del convertidor.....</b>	<b>22</b>
<b>3.3 Análisis PWM SWITCH.....</b>	<b>23</b>
<b>CAPÍTULO IV. DESARROLLO DEL ASISTENTE.....</b>	<b>27</b>

4.1	Descripción general del Asistente. ....	28
4.3	Diseño y Modelado de la Topología Buck. ....	32
4.4	Implementación PWM-switch.....	32
4.5	Control PID.....	33
4.6	Programación de Rutinas de Diseño.....	34
4.7	Interfaz de Usuario.....	34
4.8	Uso del Asistente en un convertidor Buck.....	41
4.14	Comparativa de resultados gráficos del Asistente y los obtenidos en laboratorio, en lazo abierto.....	60
4.15	Comparativa de resultados gráficos del Asistente y los obtenidos en laboratorio, en lazo cerrado. ....	62
4.16	Perturbaciones.....	63
4.17	Análisis concordancia del Asistente.....	65
CAPÍTULO V. CONCLUSIONES Y TRABAJOS FUTUROS.....		71
5.1	Conclusiones. ....	72
5.2	Trabajos Futuros.....	74
<b>Bibliografía.</b> ....		75
ANEXOS.....		77
Anexo A. Código de programación del Asistente.....		78

## LISTA DE FIGURAS

Figura 1.1. Operación de un convertidor en Electrónica de Potencia.....	2
Figura 1.2. Modos de operación en un convertidor Buck, Boost o Buck-Boost.....	4
Figura 1.3. Diagrama de Lazo Abierto.....	6
Figura 1.4. Diagrama de Lazo Cerrado.....	7
Figura 1.5. Pasos para el diseño e implementación de un convertidor cd-cd.....	9
Figura 2.1. Interfaz gráfica propuesta en [16].....	15
Figura 2.2. Esquema propuesto en [17].....	15
Figura 2.3. Modelo propuesto en [18].....	16
Figura 2.4. Interfaz gráfica propuesta en [19].....	17
Figura 2.5. Interfaz gráfica propuesta en [20].....	18
Figura 2.6. Interfaz gráfica propuesta en [21].....	19
Figura 3.1. Topología convertidor Buck.....	22
Figura 3.2. Circuito y formas de onda del convertidor Buck.....	22
Figura 3.3. a) Circuito Buck b) Circuito Buck con PWM switch.....	24
Figura 3.4. Modelo de entrada-salida en pequeña señal.....	24
Figura 3.5. Modelo de impedancia de salida en pequeña señal.....	25
Figura 3.6. Modelo de control-salida en pequeña señal.....	26
Figura 4.1. Diagrama de flujo del Asistente.....	29
Figura 4.2. Estructura de funcionamiento y programación del asistente.....	30
Figura 4.3. Cuadrantes del Asistente.....	31
Figura 4.4. Controlador PID continuo.....	33
Figura 4.5. Interfaz de usuario inicial del Asistente.....	35
Figura 4.6. Interfaz de usuario de parámetros de entrada del convertidor.....	36
Figura 4.7. Interfaz de usuario de parámetros del convertidor con resultados en el segundo cuadrante. .....	37
Figura 4.8. Implementación en Simulink de las ecuaciones características del Convertidor Buck.....	38
Figura 4.9. Resultados gráficos en lazo abierto del convertidor Buck.....	38
Figura 4.10. Configuración de componentes parásitos y obtención de diagramas de Bode.....	39
Figura 4.11. Selección del control y despliegue de configuración de parámetros del control PID.....	40
Figura 4.12. Configuración de parámetros del control PID.....	40
Figura 4.13. Respuesta del control PID ante perturbaciones.....	41

Figura 4.14. Resultados de los elementos y consideraciones para la implementación del convertidor Buck. ....	42
Figura 4.15. Pulso de disparo del dispositivo de conmutación. ....	43
Figura 4.16. Forma de onda de la corriente en el inductor. ....	43
Figura 4.17. Forma de onda de la corriente en el interruptor. ....	44
Figura 4.18. Forma de onda de la corriente en el diodo. ....	44
Figura 4.19. Forma de onda de la tensión de salida. ....	45
Figura 4.20. Configuración de componentes parásitos. ....	46
Figura 4.21. Diagrama de Bode de la función de transferencia entrada-salida. ....	46
Figura 4.22. Diagrama de Bode de la función de transferencia impedancia de salida. ....	47
Figura 4.23. Diagrama de Bode de la función de transferencia control-salida. ....	47
Figura 4.24. Selección del esquema de control PID. ....	48
Figura 4.25. Configuración del esquema de control PID. ....	49
Figura 4.26. Respuesta del control PID ante perturbaciones. ....	49
Figura 4.27. Circuito esquemático del convertidor Buck. ....	50
Figura 4.28. Placa del convertidor Buck. ....	50
Figura 4.29. Convertidor Buck implementado en el laboratorio. ....	51
Figura 4.30. Pulso de disparo del dispositivo de conmutación. ....	52
Figura 4.31. Forma de onda de la corriente en el inductor. ....	52
Figura 4.32. Forma de onda de la corriente en el interruptor. ....	53
Figura 4.33. Forma de onda de la corriente en el diodo. ....	53
Figura 4.34. Forma de onda de la tensión de salida. ....	54
Figura 4.35. Diagrama de bloques de control del convertidor. ....	54
Figura 4.36. Circuito de control PID. ....	55
Figura 4.37. Configuración del control PID en el Asistente. ....	56
Figura 4.38. Placa del circuito de control PID e impulsor. ....	57
Figura 4.39. Convertidor Buck con lazo cerrado de control. ....	57
Figura 4.40. Respuesta del esquema de control PID (color cian). ....	58
Figura 4.41. Comportamiento de la tensión de salida (color cian) ante incremento de la tensión de entrada (color verde). ....	58
Figura 4.42. Comportamiento de la tensión de salida (color cian) ante decremento de la tensión de entrada (color verde). ....	59
Figura 4.43. Comportamiento de la tensión de salida (color cian) ante decremento en la carga (color azul). ....	59
Figura 4.44. Comparación de pulsos de disparo del dispositivo de conmutación (a) y (b). ....	60
Figura 4.45. Comparación de la corriente en el inductor (a) y (b). ....	60
Figura 4.46. Comparación de la corriente en el dispositivo de conmutación (a) y (b). ....	61
Figura 4.47. Comparación de la corriente en el diodo (a) y (b). ....	61
Figura 4.48. Comparación de la tensión de salida del convertidor (a) y (b). ....	62
Figura 4.49. Comparación de respuesta transitoria del control PID (a) y (b). ....	63

Figura 4.50. Comparación del comportamiento de la tensión de salida ante incremento de la tensión de entrada (a) y (b).....	64
Figura 4.51. Comparación del comportamiento de la tensión de salida ante decremento de la tensión de entrada (a) y (b).....	64
Figura 4.52. Comparación del comportamiento de la tensión de salida ante decremento en la carga (a) y (b).....	65
Figura 4.53. Comparación de magnitud de corriente en el dispositivo de conmutación.....	66
Figura 4.54. Comparación de magnitud de corriente en el dispositivo de conmutación.....	67
Figura 4.55. Comparación de magnitud de corriente ascendente en el inductor.....	68
Figura 4.56. Comparación de magnitud de corriente descendente en el inductor.....	69
Figura 4.57. Comparación del control PID. ....	70

## LISTA DE TABLAS

Tabla 2.1. Comparativa de Asistentes para el análisis y diseño de convertidores CD-CD. ....	20
Tabla 3.1. Ecuaciones Básicas del Convertidor Buck. ....	23
Tabla 4.1. Especificaciones Técnicas del Convertidor Buck. ....	41
Tabla 4.2. Porcentaje de error promedio en la corriente del dispositivo de conmutación. ....	67
Tabla 4.3. Porcentaje de error en la corriente del diodo. ....	67
Tabla 4.4. Porcentaje de error en la corriente ascendente del inductor. ....	68
Tabla 4.5. Porcentaje de error en la corriente ascendente del inductor. ....	69
Tabla 4.6. Porcentaje de error en la corriente ascendente del inductor. ....	70

## SIMBOLOGÍA

A	Ampere
C	Capacitor de salida
D	Ciclo de trabajo
I	Corriente
V	Tensión
$V_i$	Tensión de entrada
$V_o$	Tensión de salida
L	Inductor
Q	Interruptor controlado
R	Resistencia
$T_{off}$	Tiempo de corte
$T_{on}$	Tiempo de conmutación
$T$	Periodo de conmutación
MHz	Mega Hertz
kHz	Kilo Hertz
$f_s$	Frecuencia de conmutación
$\Delta i_L$	Rizo de corriente de fase
$\Delta V_o$	Rizo de tensión de salida
$I_Q$	Corriente promedio en el dispositivo de conmutación
$i_{pkQ}$	Corriente pico en el dispositivo de conmutación
$V_C$	Tensión capacitor
$I_o$	Corriente de carga
$V_{DSOFF}$	Tensión máxima en el dispositivo de conmutación
$I_D$	Corriente promedio en el diodo
$i_{pkD}$	Corriente pico en el diodo
$V_{AKSOFF}$	Tensión máxima en el diodo
M	Ganancia
$R_L$	Resistencia parásita en el inductor
$R_C$	Resistencia parásita en el capacitor
$K_P$	Ganancia proporcional
$K_I$	Ganancia integral
$K_D$	Ganancia derivativa

W	Watts
$V_{GS}$	Voltaje compuerta – fuente



## ACRÓNIMOS

CD	Corriente directa.
CA	Corriente alterna.
PID	Esquema de control proporcional, integral y derivativo
$P_E$	Potencia de entrada.
$P_S$	Potencia de salida.
$P_P$	Pérdidas de potencia.
ON	Estado de conmutación.
OFF	Estado de corte.
MCC	Modo de conducción continua.
MCD	Modo de conducción discontinua.
PWM	Modulador por ancho de pulso
GUI	Interfaz gráfica de usuario
MOSFET	Transistor de efecto de campo

# **CAPÍTULO I. INTRODUCCIÓN**

## 1.1 Antecedentes.

La electrónica de potencia es parte de la electrónica que estudia los dispositivos y circuitos electrónicos usados para modificar características de la energía eléctrica (tensión, frecuencia, forma de onda, etc.). Un convertidor, en general, se encarga de transformar la potencia eléctrica de corriente y tensión determinada, a otra forma de corriente y tensión, con el objetivo de minimizar las pérdidas de energía derivadas por los elementos que integran el convertidor y el calor que se obtiene en el proceso de conversión de energía, ver Figura 1.1.

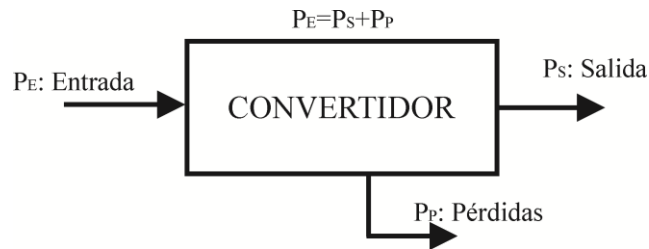


Figura 1.1. Operación de un convertidor en Electrónica de Potencia.

Un convertidor CD-CD convierte en forma directa de CD a CD y es equivalente a la función de un transformador CA, solo que en CD, con una relación de vueltas que varía en forma continua. Al igual que un transformador, se puede usar para subir o bajar la tensión de una fuente, dependiendo del diseño del mismo. Los convertidores CD se usan en el control de motores de tracción de automóviles eléctricos, tranvías, grúas marinas, montacargas y computadoras.

## 1.2 Conceptos de convertidores CD-CD.

Una de las partes más importantes de la electrónica es la denominada electrónica de potencia, que estudia la conversión y control de la energía eléctrica y sus aplicaciones tratando de maximizar la eficiencia, o lo que es lo mismo, minimizar la disipación de energía. Para aumentar la eficiencia y reducir el peso, el tamaño y el costo de los sistemas electrónicos de potencia, los convertidores CD-CD son unos de los circuitos más utilizados. Se utilizan en todas las situaciones donde es necesario mantener una tensión a un nivel deseado.

El principio de funcionamiento es el mismo para todos los convertidores conmutados, es el principio de almacenamiento y transferencia de energía en ciclos de conmutación. Durante el primer intervalo del ciclo de trabajo, el convertidor almacena la energía en la bobina, transfiriendo en el segundo intervalo de trabajo esta energía al condensador. El control determina los estados de conducción y de bloqueo de los dispositivos de conmutación.

Con el objetivo de convertir energía eléctrica con la máxima eficiencia, los convertidores de potencia ideales poseen únicamente elementos que no presentan pérdidas, es decir, que no disipan potencia. Se pueden establecer dos grupos básicos de componentes reales que presentan pérdidas mínimas:

- **Componentes reactivos:** Son los elementos que almacenan energía, como los capacitores e inductores. Estos componentes absorben energía del circuito, la almacenan y finalmente la devuelven al circuito.
- **Componentes de conmutación:** Son dispositivos que se comportan idealmente como interruptores, es decir, sin pérdidas de conmutación; en estado de conmutación ('ON') presentan una tensión en terminales nula ( $v=0$ ), y en estado de corte ('OFF') tienen una corriente de paso nula ( $i=0$ ). De esta manera el producto  $V \cdot I$  será siempre cero y no disiparan potencia. Otro punto importante para que los interruptores conmutados disipen la menor potencia posible, es procurar que los tiempos de conmutación sean muy pequeños, es decir que la conmutación sea lo más rápida posible para evitar estar mucho tiempo en la zona lineal.

Estos convertidores tienen múltiples aplicaciones: fuentes de poder en computadoras, sistemas distribuidos de potencia, sistemas de potencia en vehículos eléctricos, etc. Las configuraciones básicas son tres: Buck (reductora), Boost (elevadora) y Buck-Boost (elevadora- reductora). Estas configuraciones permiten elevar, reducir o elevar/reducir la tensión de alimentación ( $V_i$ ) en la salida ( $V_o$ ). Cada configuración a su vez contiene cuatro elementos básicos: bobina (L), capacitor (C), diodo y un interruptor controlado (Q); así las propiedades de cada topología dependen de la ubicación de estos cuatro elementos. Normalmente se considera que la carga para los convertidores es de tipo resistiva (R), aunque esto puede variar.

A parte de estos convertidores CD-CD existen otros tipos, como los convertidores con aislamiento eléctrico o convertidores resonantes.

### 1.2.1 Modos de Conducción.

Todos los convertidores pueden presentar dos modos de conducción, los cuales se deben a la relación entre el tiempo en el que el interruptor se encuentra cerrado, y el tiempo necesario para que la bobina descargue totalmente la energía almacenada previamente. Los modos de conducción posibles son:

**Modo de conducción continua (MCC):** La corriente que fluye por el inductor oscila entre unos valores máximo y mínimo, pero nunca llega a anularse. Esto se debe a que el dispositivo de conmutación deberá estar bloqueado un intervalo de tiempo que permita a la corriente en la carga no hacerse cero. De este modo, al comenzar el siguiente periodo la corriente podrá partir de un valor inicial.

**Modo de conducción discontinua (MCD):** La corriente en inductor se hace nula en un momento determinado a lo largo de un intervalo de tiempo  $T_{off}$  durante el cual el interruptor esta abierto. El tiempo que permanece abierto el interruptor es mayor que el tiempo que puede estar el inductor cediendo energía, con lo que al iniciarse el siguiente periodo la corriente en el inductor partirá de cero.

Ambos modos de operación se pueden observar en la Figura 1.2. Existe un modo de operación que en realidad es una transición entre los modos descritos y se denomina “frontera”.

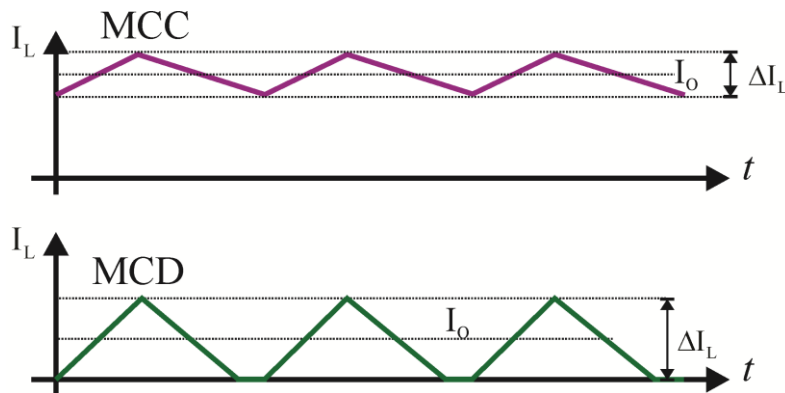


Figura 1.2. Modos de operación en un convertidor Buck, Boost o Buck-Boost.

### 1.2.2 Ciclo de trabajo.

En un convertidor el interruptor se abre y se cierra siguiendo una señal de periodo  $T$ . El tiempo durante el cual el interruptor se encuentra cerrado, se denomina tiempo de conducción ( $T_{on}$ ). Por otro lado el tiempo que el interruptor se encuentra abierto, se llama tiempo de bloqueo,  $T_{off}$ . La suma de  $T_{on}$  y  $T_{off}$  forman el periodo del convertidor  $T$ .

En un convertidor CD-CD la potencia entregada a la carga está en función de  $D$ , cociente entre  $T_{on}$  y  $T$ . A este cociente se le denomina *ciclo de trabajo*. Y se define como la fracción del periodo del convertidor en el cual el interruptor se encuentra cerrado.

$$D = \frac{T_{on}}{T} \quad (1.1)$$

Observando la ecuación (1.1), se deduce que se presentan tres formas diferentes de modificar el ciclo de trabajo, y por tanto la tensión de salida:

1. Variando el tiempo de conducción  $T_{on}$ , al mismo tiempo que se mantiene  $T$  fijo. Llamado también Modulación por Ancho de Pulso (PWM), debido a que la frecuencia de la señal del convertidor se mantiene constante, mientras que no ocurre así con la anchura del pulso que define el tiempo de conducción del convertidor.
2. Variando  $T$  y conservando  $T_{on}$  constante. Denominado Modulación de Frecuencia, ya que es la frecuencia del convertidor la que varía. El inconveniente más representativo de este método de control, se encuentra en la generación indeseada de armónicos a frecuencias variables, por lo que el diseño de un filtro se convertirá en un trabajo adicional.
3. Modificando ambos, el tiempo de conducción  $T_{on}$  y  $T$

### 1.3 Lazo de Control.

Los controles automáticos o sistemas de control constituyen una parte muy importante en los procesos industriales modernos, donde se les usa principalmente para regular diferentes variables (Tensión, presión, temperatura, nivel, flujo, etc.). Una definición de un sistema de control puede ser, “Es un arreglo de componentes físicos conectados de tal manera, que el arreglo pueda comandar, dirigir o regular a sí mismo o a otro sistema o proceso”.

Los convertidores pueden sufrir variaciones tanto de sus parámetros externos (tensión de entrada, variaciones de la potencia requerida por la carga), como de los parámetros internos (derivas de capacidades, autoinducciones por envejecimiento, y calentamiento). Estas variaciones influyen en el comportamiento según el sistema de control que presente el convertidor. Los convertidores pueden funcionar tanto en *Lazo abierto* como en *Lazo cerrado*.

A continuación se describen los sistemas de control según se encuentre trabajando el convertidor.

#### 1.3.1 Lazo Abierto.

El convertidor trabajando en lazo abierto no cuenta con ningún tipo de realimentación, por lo que la señal de conmutación (señal que rige el estado del interruptor), vendrá determinada al efectuar la comparación entre una señal de referencia, con otra señal de frecuencia fija procedente de un oscilador que determina la frecuencia de funcionamiento del sistema. En la Figura 1.3 se muestra el diagrama a bloques de un sistema de lazo abierto.

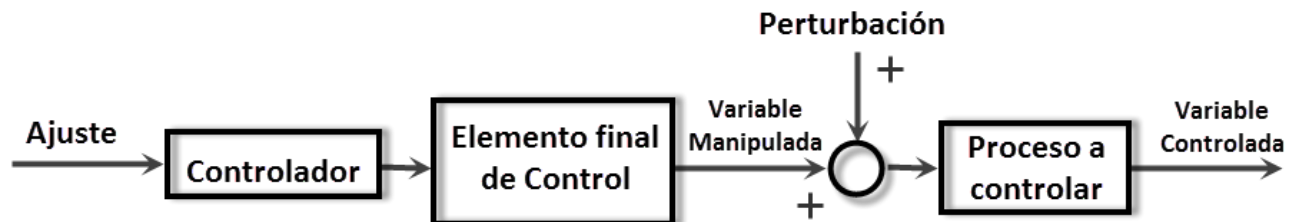


Figura 1.3. Diagrama de Lazo Abierto.

### 1.3.2 Lazo Cerrado.

Cualquiera que sea el convertidor CD-CD, el nivel de la señal de salida depende directamente del ciclo de trabajo, por lo tanto a fin de mantener constante el nivel de dicha señal, el valor del ciclo de trabajo varía según las condiciones instantáneas de potencia requerida por la carga.

Para que el convertidor dependa poco de la variación de los parámetros se debe regular con un circuito de control (controlador) que suministre una consigna (tensión o corriente) y que intente mantener el estado del convertidor próximo a ella. Por lo que la regulación se efectúa en lazo cerrado mediante lazos de realimentación que traducen las desviaciones en una desviación de signo contrario de la potencia entregada a la carga modificando el ciclo de trabajo de la señal de conmutación [15].

La variación del ciclo de trabajo o margen de control tiene siempre unos límites que dependen, en general, de las tensiones de entrada y de salida, de la frecuencia de conmutación y de la rapidez con que el sistema deba retornar a sus condiciones estables. La Figura 1.4 muestra el diagrama de bloques de un sistema en lazo cerrado.

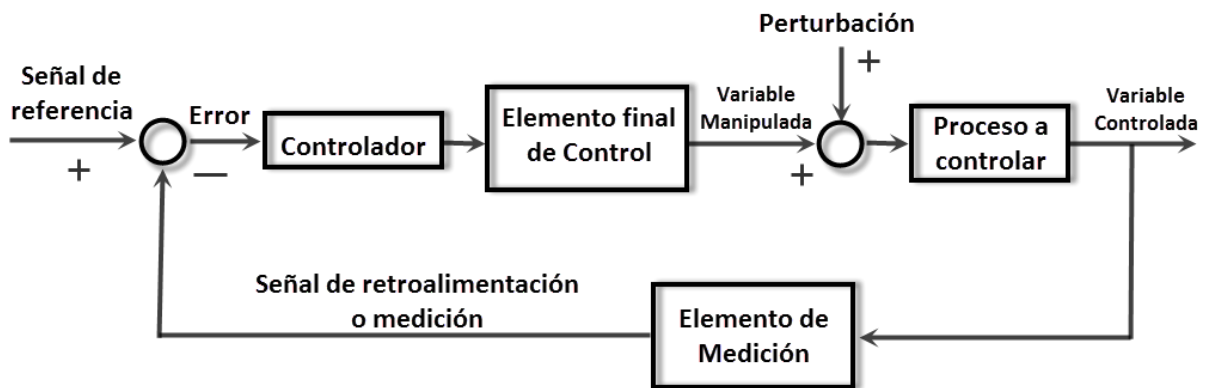


Figura 1.4. Diagrama de Lazo Cerrado.



#### **1.4 Planteamiento del Problema.**

El diseño de convertidores tipo CD-CD, es un procedimiento bien definido, que el diseñador de convertidores debe repetirlo una y otra vez, hasta lograr un diseño que después se pueda llevar a una etapa de armado para validar la conveniencia del diseño. Si existe algún parámetro físico que no sea satisfactorio, el diseñador deberá repetir todo el procedimiento hasta la etapa de pruebas con el nuevo diseño.

Lo que se busca con esta tesis, es desarrollar una herramienta en simulación, que contemple la topología Buck y que le permita al usuario diseñar y rediseñar un convertidor de manera fácil, así como adquirir capacitación en convertidores, con la ventaja de que el diseño final del convertidor le permita implementar sólo una versión final, sin el uso de paquetes de simulación muy sofisticados.

Con esto se busca que la herramienta tenga la ventaja de realizar de manera repetitiva el proceso de diseño de modo que se reduzca el tiempo desde la etapa de diseño hasta la de pruebas funcionales, y permitir interacción con la información de diseño, mediante la manipulación de los datos de especificación de cada convertidor. Además el diseñador puede validar para el convertidor, el efecto que tiene en la señal regulada al hacer cambios en los valores de los componentes.

La Figura 1.5 muestra el proceso para realizar el diseño e implementación de un convertidor CD-CD.

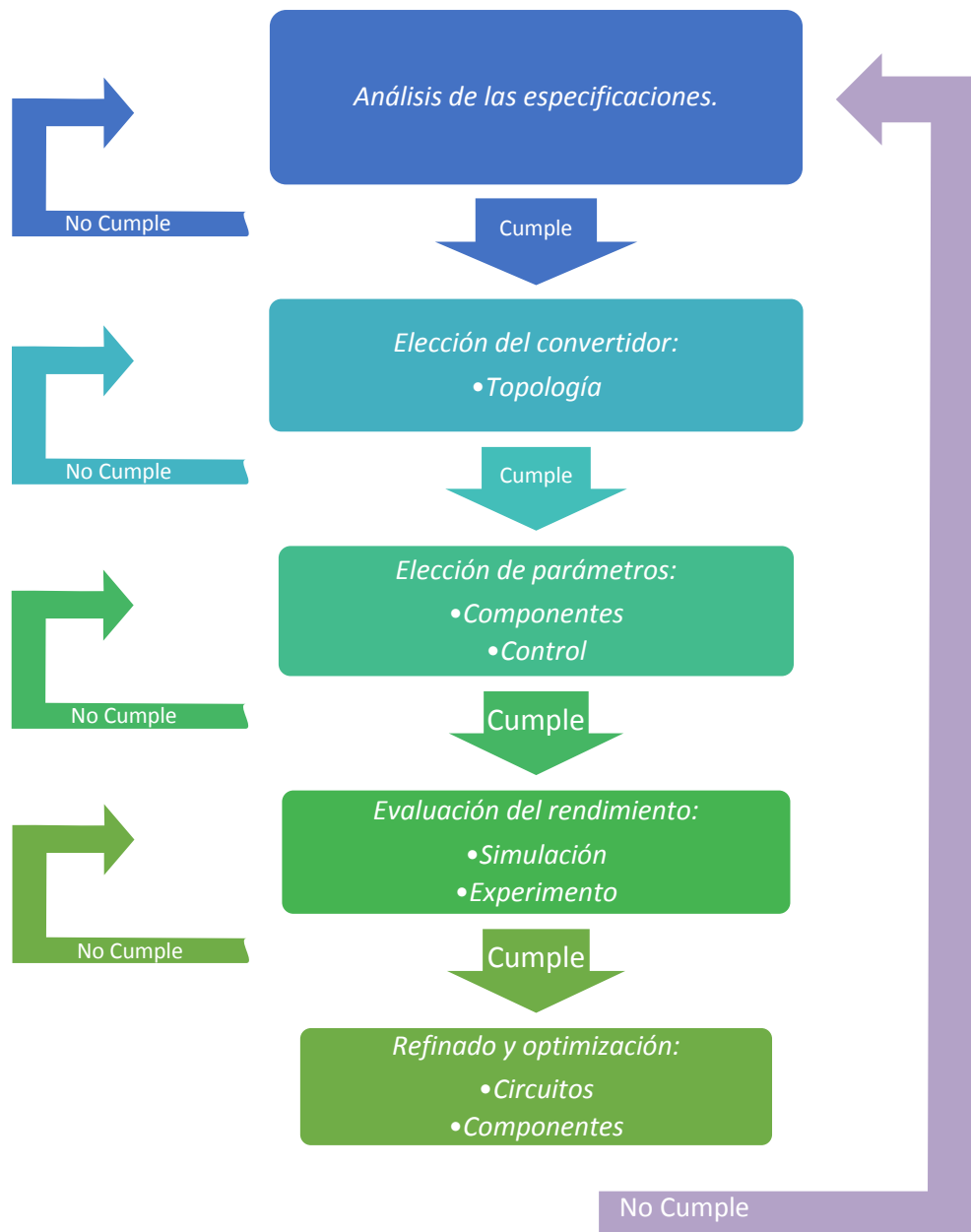


Figura 1.5. Pasos para el diseño e implementación de un convertidor cd-cd.

## 1.5 Objetivos.

A continuación se presenta el objetivo general y los objetivos específicos desglosados de la tesis.

### **1.5.1 Objetivo general.**

Desarrollar una herramienta que asista al desarrollador de convertidores CD-CD tipo Buck en su diseño, así como también asista en establecer un esquema de control en lazo cerrado.

### **1.5.2 Objetivos particulares.**

- Modelado del convertidor CD-CD tipo Buck.
- Estudio de los métodos de diseño y caracterización del convertidor mencionado.
- Programación de rutinas de diseño.
- Desarrollo de una interfaz amigable para incorporar las especificaciones de diseño.
- Implementación de un prototipo del convertidor, para validar la utilidad de la herramienta desarrollada (Corroborar los resultados del asistente con las mediciones experimentales obtenidas en laboratorio).

## **1.6 Justificación.**

La sistematización del diseño de los convertidores CD-CD más comúnmente utilizados en aplicaciones de electrónica de potencia, le permitirá al diseñador un ahorro de tiempo y componentes electrónicos al contar con una herramienta interactiva de diseño.

Permite contar con una herramienta de simulación de convertidores CD-CD no sofisticada. De igual manera, proporciona al usuario experiencia en el comportamiento dinámico debido a algunos cambios en los parámetros de los convertidores CD-CD de manera interactiva.

Dada la importancia y la complejidad en establecer un sistema de control en lazo cerrado, siguiendo el procedimiento convencional, el asistente facilitará el desarrollo de la etapa de control, ya que debe mostrar los resultados de manera inmediata y proporcionar los parámetros de control del convertidor a implementar. Con la ventaja de que en caso de que no se cumpla el comportamiento requerido, permitir recalcularlos de manera rápida, hasta satisfacer las necesidades del usuario, optimizando con esto el tiempo requerido para visualizar el

comportamiento y posteriormente realizar la implementación de un lazo cerrado de control del convertidor en cuestión.

## **1.7 Organización de la Tesis.**

La distribución de los capítulos en este trabajo se realiza de la siguiente manera:

En el capítulo 2 se presenta la revisión del estado del arte sobre los trabajos relacionados con el tema de tesis. Se muestran las herramientas existentes para la simulación e implementación de convertidores CD-CD.

En el capítulo 3 se describe el análisis que se realizó del convertidor Buck, en donde se obtienen las ecuaciones características del convertidor, el análisis en pequeña señal para la obtención de las funciones de transferencia que rigen al convertidor Buck.

En el capítulo 4 se presenta el proceso de desarrollo, la configuración y la forma de ejecución del asistente. Así mismo, se describe el proceso de implementación, en donde a partir de un diseño propuesto, los resultados ofrecidos por el asistente, sirven como base para implementación práctica del prototipo, se muestran los resultados obtenidos del proceso, una comparativa entre los resultados, y por último, un análisis de validación de los resultados proporcionados por el asistente y los obtenidos en el laboratorio, para corroborar el margen de error de la herramienta.

En el capítulo 5 se muestran las conclusiones generales de la tesis y visualización de los trabajos futuros para que sean complementados dentro del asistente

Finalmente se incluye la bibliografía que se utilizó para la elaboración de la tesis.

## **CAPÍTULO II. ESTADO DEL ARTE**

## **2.1 Introducción.**

En la búsqueda bibliográfica se han encontrado estudios y temas relacionados para la elaboración de esta tesis. Dichos estudios y temas, de acuerdo al grado de impacto que se tiene dentro de la propuesta, se clasifican en:

- Consideraciones de diseño y dimensionamiento de convertidores CD-CD.
- Identificación de Factores Críticos en el diseño de convertidores CD-CD.
- Selección de dispositivos óptimos para la implementación de convertidores CD-CD.
- Herramientas y software de simulación para diseño de convertidores CD-CD.

## **2.2 Consideraciones de diseño y dimensionamiento de convertidores CD-CD.**

Para la realización de convertidores CD-CD, es necesario conocer los elementos, la topología y el principio básico de operación, para cada una de las configuraciones básicas de modelado que contempla esta propuesta de tesis. La necesidad de conocer cada una de las configuraciones básicas la podemos encontrar en la literatura [1] [2] [3] [4] [5] [6] [7].

Los estudios reportados en la literatura [8], [9] y [10], son considerados esenciales para el diseño y dimensionamiento de convertidores CD-CD, en los cuales, se presentan las principales etapas y los diferentes medios que pueden utilizarse para ayudar al diseñador a implementar un convertidor; describen aspectos relacionados en el diseño de convertidores a muy altas frecuencias (30-300 MHz); se aborda la importancia del modelado y la simulación de los convertidores, se presentan los métodos de diseño sencillos de modelado de convertidores CD-CD con ayuda del software MATLAB / Simulink.

## **2.3 Identificación de Factores Críticos en el diseño de convertidores CD-CD.**

Una vez determinadas las configuraciones básicas de convertidores CD-CD, es necesario identificar los factores críticos de diseño. En [8] y [11], se presenta la importancia en la elección de la topología del convertidor en función del nivel de potencia a convertir; la dificultad de diseñar y operar un convertidor CD-CD en niveles de alta potencia y tensión; la manera en que interruptores (semiconductores) se dañan en alta potencia y en diferentes

niveles de frecuencia de conmutación, así como la saturación del circuito magnético; los efectos de la inductancia de fuga y la saturación de diferentes cables y técnicas de bobinado.

#### **2.4 Selección de dispositivos óptimos para la implementación de convertidores CD-CD.**

Una vez determinada la configuración que se va a diseñar, simular e implementar es necesario establecer los elementos físicos que integran la topología deseada. Para ello en [12], [13], [14] y [15] se presenta la tendencia en el diseño de convertidores en construirlos en miniatura, hacerlos más ligeros y de alta potencia, y el impacto que tiene en los componentes que integran la topología; se menciona un método para la determinación de componentes bajo estrés térmico y estrés de vibración, así como una evaluación de vida útil en los convertidores CD-CD; la degradación en los componentes de los convertidores y un pronóstico basado en predicción de la tendencia de fallas por degradación en el rendimiento de los convertidores; se menciona la determinación de los elementos del convertidor considerando un diseño estático, que consiste en la determinación de los elementos del convertidor a partir de los requerimientos o aplicaciones deseadas, y un diseño dinámico que consiste en un lazo de control que permite al convertidor en estado estable mantener la potencia máxima requerida en la carga ante pequeñas perturbaciones.

#### **2.5 Herramientas y software de simulación para diseño de convertidores CD-CD.**

Se analizaron los artículos [16], [17], [18], [19], [20] y [21] que muestran diferentes desarrollos para el diseño y modelado de convertidores CD-CD.

En [16], se presenta una herramienta en Matlab con el fin de ayudar a los estudiantes a comprender a detalle la funcionalidad de los convertidores de potencia básicos y la limitación impuesta por sus diseños. Se grafica la corriente, la tensión de salida y la potencia de la carga en régimen de estado transitorio o estable, así como en modo MCC y MCD. En la Figura 2.1 se muestra la interfaz gráfica propuesta en [16].

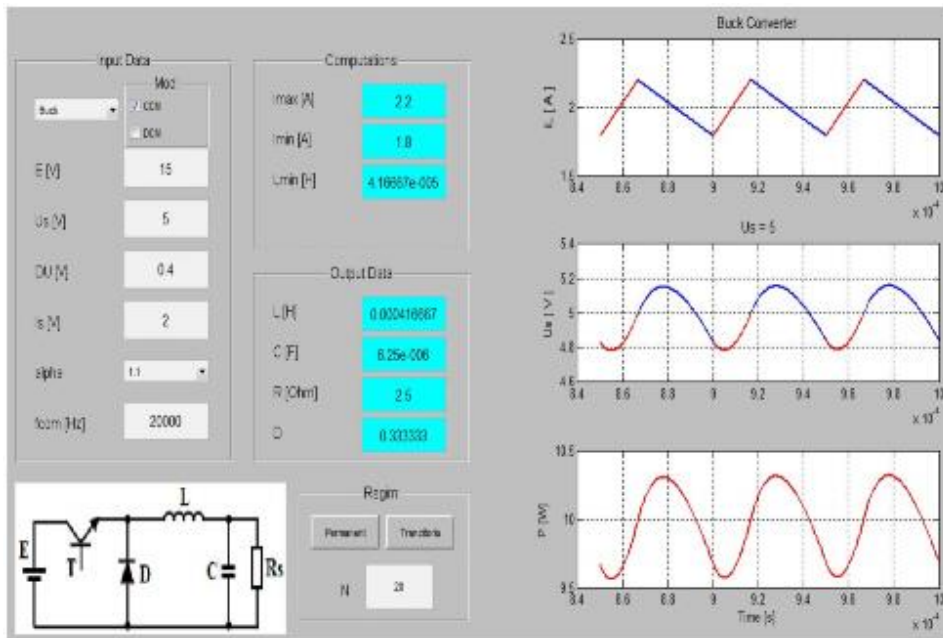


Figura 2.1. Interfaz gráfica propuesta en [16].

En [17], se muestra un nuevo enfoque para el diseño de convertidores CD-CD. El enfoque incluye la programación en PSPICE de las ecuaciones de la etapa de potencia y el circuito de control, utilizando el modelado del comportamiento analógico. La Figura 2.2 se muestra la propuesta de la referencia [17].

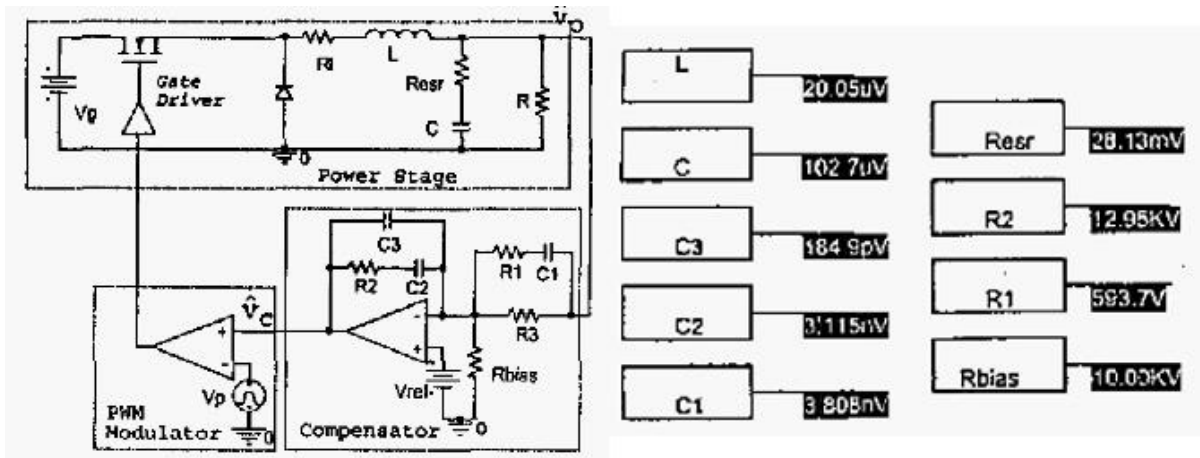


Figura 2.2. Esquema propuesto en [17].



También en [18], se contemplan las simulaciones y las condiciones de funcionamiento de convertidores de potencia utilizando el software Matlab/Simulink. Los resultados ilustran que las características dinámicas y las propiedades de los convertidores siguen la teoría básica de los mismos. La Figura 2.3 se muestra el modelo propuesto en la referencia [18].

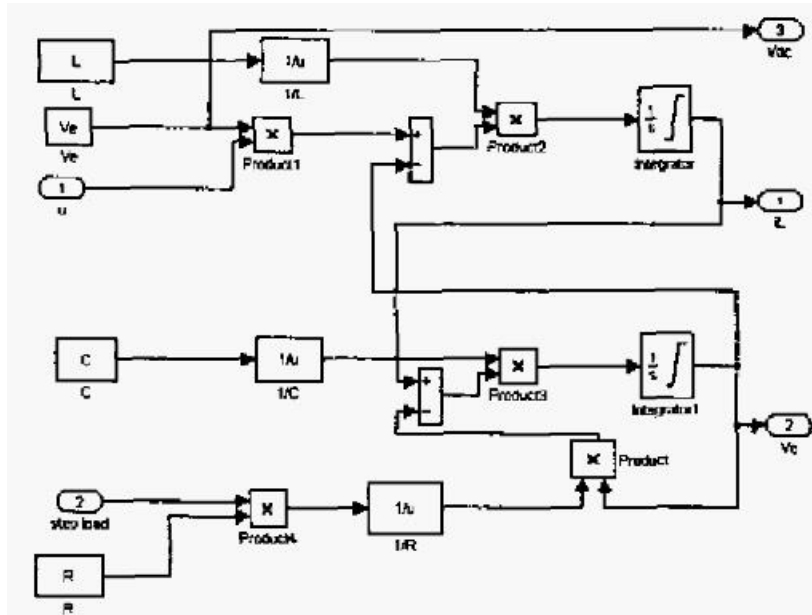
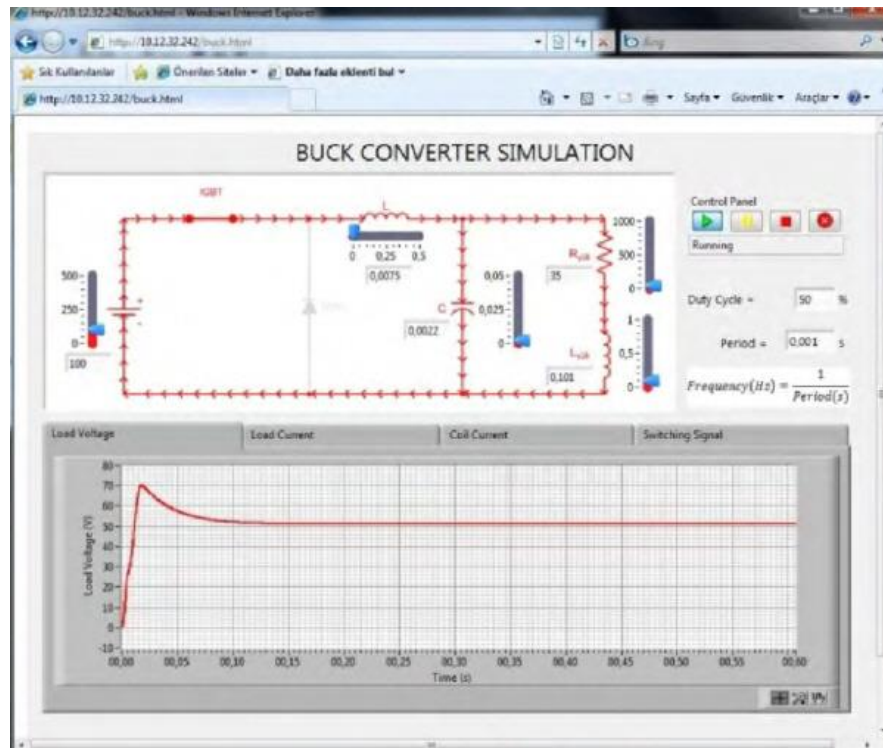


Figura 2.3. Modelo propuesto en [18].

En [19], se menciona la utilidad y la ayuda que ofrece internet para la simulación de convertidores CD-CD. Se desarrolló un conjunto de herramientas de simulación para las configuraciones básicas de los convertidores, en dónde los usuarios se pueden conectar al sistema, a través de una conexión de internet y poder realizar simulaciones. El conjunto de herramientas para la simulación incluye el software Matlab/Simulink, que se utiliza para la simulación del sistema y el software LabVIEW, que se utiliza para realizar la interfaz de usuario. La Figura 2.4 se presenta la herramienta de la referencia [19].



**Figura 2.4. Interfaz gráfica propuesta en [19].**

Los trabajos de investigación presentados anteriormente, muestran la importancia que toma el desarrollo de las herramientas de apoyo para el diseño y el modelado de convertidores CD-CD.

En [20], se aporta un desarrollo de herramientas interactivas y de simulación, para servir como una herramienta de diseño orientado a la enseñanza y a las prácticas de laboratorio para cursos de electrónica de potencia. Al contar con una herramienta gráfica e interactiva, permite a los estudiantes manipular el diseño de los convertidores y comprender mejor el funcionamiento de los mismos. De esta manera, se complementan los conocimientos adquiridos en clase y la herramienta de simulación. Cabe destacar, que la herramienta está enfocada al ámbito de aprendizaje, con un desarrollo y aplicación externa, la cual no requiere de alguna licencia de software para su funcionamiento. La Figura 2.5 muestra la interfaz de la herramienta propuesta en la referencia [20].

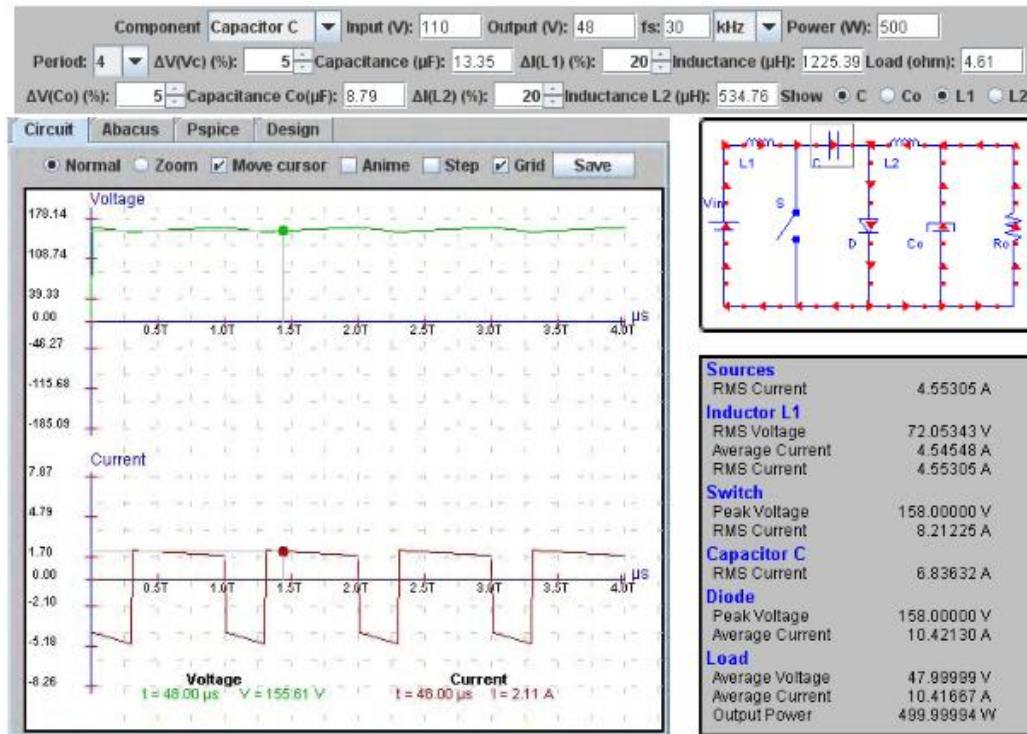


Figura 2.5. Interfaz gráfica propuesta en [20].

En [21] se propone un software reconfigurable on-line con el objetivo de realizar una plataforma capaz de construir las seis topologías básicas, que permite a los diseñadores o estudiantes tener una mejor comprensión sobre el comportamiento de los convertidores de potencia a través de experimentos prácticos.

El diseño de esta plataforma consta de tres elementos: tiene una mesa de pruebas reconfigurable con elementos de electrónica de potencia y diferentes tipos de cargas, un laboratorio virtual a distancia cuyo acceso es mediante internet (Web-based) y un software interactivo on-line (e-Learning).

La plataforma e-Learning contiene la pantalla donde se despliegan las gráficas correspondientes a las mediciones que se realiza, además contiene el panel del laboratorio virtual así como botones que permiten modificar los valores de la tensión de entrada y el ciclo de trabajo.

Las desventajas que tiene esta plataforma que solo es para simulación de convertidores, no se proponen valores para implementación, sino que los únicos elementos que se disponen son los que están en el tablero de pruebas. Las ventajas que tiene es que permite ajustar los valores de la fuente de alimentación, del generador de señales y ganancias del compensador vía internet y está disponible las 24 horas los 7 días de la semana. La Figura 2.6 presenta la propuesta de la referencia [21].

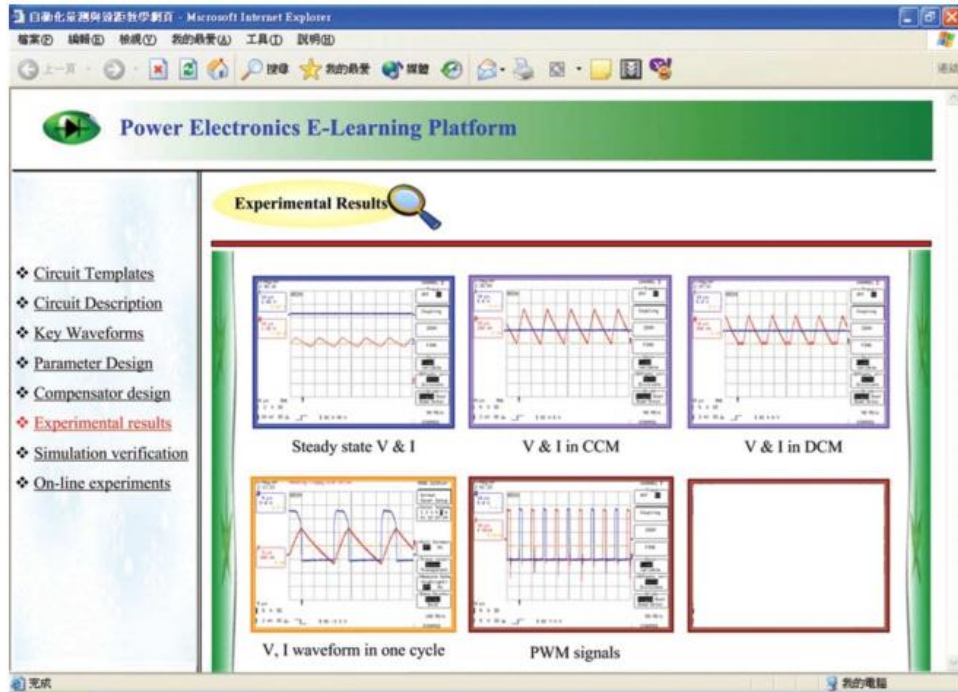


Figura 2.6. Interfaz gráfica propuesta en [21].

A continuación, la Tabla 2.1 muestra la comparativa de las herramientas encontradas que ofrecen apoyo en el análisis y diseño de convertidores CD-CD.

**Tabla 2.1. Comparativa de Asistentes para el análisis y diseño de convertidores CD-CD.**

Asistentes Para el Diseño de Convertidores							
Nombre	Plataforma de desarrollo	Convertidores cd-cd		Características			Diseño de esquema de control
		Buck	Simulación	Diseño	MCC	MCD	
A Matlab Tool for Determining the Parameters of Power DC–DC Non Isolated Converters [16]	Matlab	SI	SI	SI	SI	SI	NO
A Pspice-based Design of DC-DC Converter Systems [17]	Pspice	SI	SI	NO	SI	SI	NO
A Matlab/Simulink Tool for Enhancing Efficient Education of Power Electronics Corresponding to the ETH Power Converter Laboratory [18]	Matlab	SI	SI	SI	SI	SI	NO
Internet Based Toolkit For Simulation Of DC/DC Converters [19]	Matlab/ Simulink LabView	SI	SI	NO	SI	SI	NO
Interactive DC-DC Converters Tools For Instant Design And Education [20]	JAVA	SI	SI	NO	SI	SI	NO
Software-Reconfigurable e-Learning Platform for Power Electronics Courses [21]	JAVA	SI	SI	NO	SI	NO	NO
Asistente Para El Diseño De Topologías Básicas De Convertidores CD-CD	Matlab/ Simulink	SI	SI	SI	SI	NO	SI (PID)

# **CAPÍTULO III. ANÁLISIS DEL CONVERTIDOR BUCK**

### 3.1 Convertidor Reductor (Buck).

El circuito del convertidor Buck es representado en la Figura 3.1,  $V_i$  es la tensión de corriente directa de entrada,  $V_o$  es la tensión de salida, que se supone constante. En este convertidor siempre se cumple que  $V_o$  es menor o igual que  $V_i$ , razón por la que se le llama *Reductor*. El interruptor  $Q$  conmuta a una frecuencia  $f_s = 1/T$ . Se distingue dos modos de operación, según la corriente por el inductor  $L$  se anule en el período de operación  $T$ : MCC, cuando la mencionada corriente no se anula, y MCD, cuando la corriente por  $L$  se anula durante un intervalo.

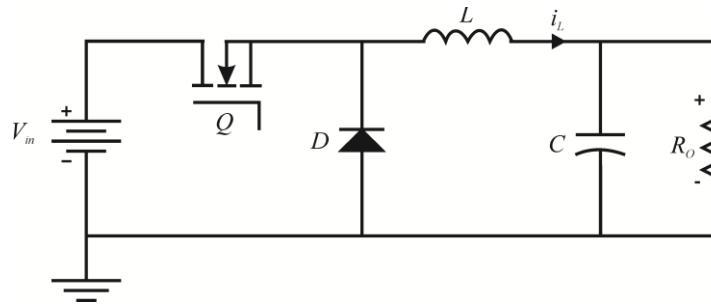


Figura 3.1. Topología convertidor Buck.

### 3.2 Ecuaciones características del convertidor.

Con la topología del convertidor Buck, se realizan los cálculos para determinar las ecuaciones características y establecer cada uno de los componentes que integran el convertidor, para ello es necesario analizar el circuito de la topología Buck, así como las formas de onda que el convertidor presenta, ver Figura 3.2.

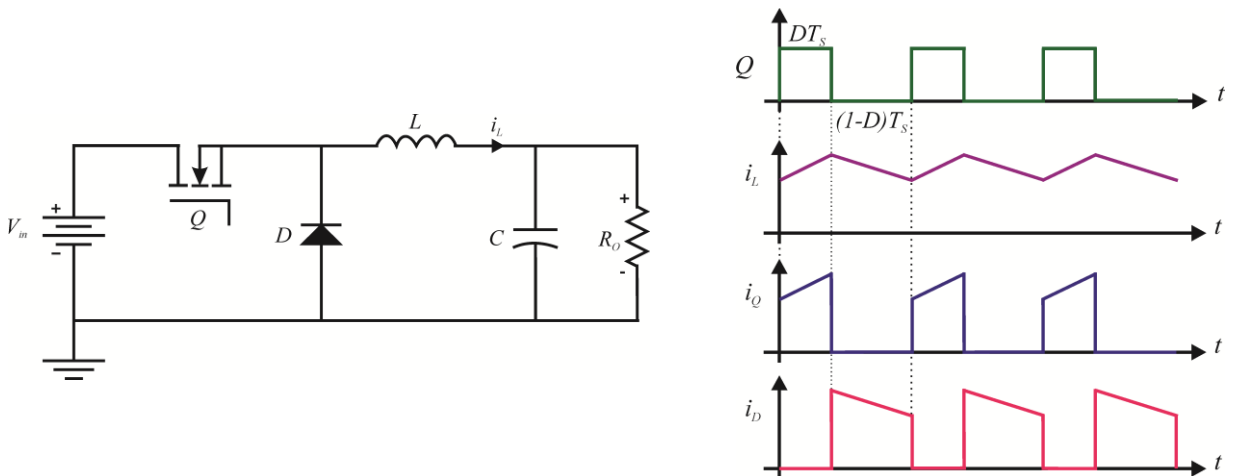


Figura 3.2. Circuito y formas de onda del convertidor Buck.

Las ecuaciones características obtenidas del análisis de este convertidor se muestran en Tabla 3.1.

**Tabla 3.1. Ecuaciones Básicas del Convertidor Buck.**

<b>Ganancia de CD</b>	$M = \frac{V_o}{V_{in}} = D$	(1.2)
<b>Inductor</b>	$L = \frac{(V_{in} - V_o)D}{\Delta i_{L(ON)} f_s}$	(1.3)
<b>Capacitor</b>	$C = \frac{(V_{in} - V_o)D}{8\Delta V_o L f_s^2}$	(1.4)
<b>Tensión en el capacitor</b>	$V_C = V_o$	(1.5)
<b>Corriente promedio en el interruptor</b>	$I_Q = I_o D = \frac{V_o}{R} D$	(1.6)
<b>Corriente pico en el interruptor</b>	$i_{pkQ} = I_o + \frac{(V_{in} - V_o)D}{2L f_s}$	(1.7)
<b>Tensión máxima en el interruptor</b>	$V_{DSOFF} = V_{in}$	(1.8)
<b>Corriente promedio en el diodo.</b>	$I_D = I_o (1 - D)$	(1.9)
<b>Corriente pico en el diodo.</b>	$i_{pkD} = I_o + \frac{(V_{in} - V_o)D}{2L f_s}$	(1.10)
<b>Tensión máxima en el diodo.</b>	$V_{AKSOFF} = V_{in}$	(1.11)

### 3.3 Análisis PWM SWITCH.

Con la necesidad de establecer un lazo de control cerrado dentro del asistente, es necesario obtener matemáticamente las funciones de transferencia del convertidor Buck, para ello se utiliza el método de análisis PWM-switch. El convertidor Buck es una de las topologías básicas que tiene un interruptor activo y un interruptor pasivo, que realizan la conmutación en el convertidor. El interruptor activo es controlado directamente por una señal de control externa. Usualmente es llevado a cabo con un transistor bipolar o de efecto de campo. El interruptor pasivo es controlado indirectamente por el estado del interruptor activo y la condición del circuito. Estos dos interruptores pueden combinarse en una sola red con tres terminales *a*, *p* y *c*, que es sinónimo de activo, pasivo, y común, ver Figura 3.3. Esta red de tres terminales se llama el PWM-switch. Puesto que se supone que todos los demás elementos



de los convertidores son lineales, el PWM-switch es el único elemento no lineal y, por tanto, responsable por el comportamiento no lineal de los convertidores.

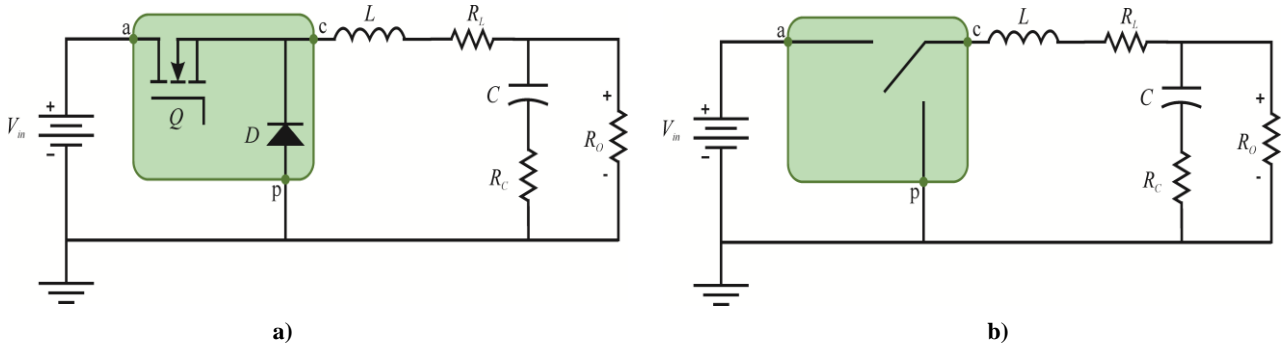


Figura 3.3. a) Circuito Buck b) Circuito Buck con PWM switch.

Hay ventajas en el uso del modelo PWM SWITCH, una de ellas es que permite que muchos convertidores PWM puedan ser analizados con el uso de programas sencillos lineales de circuitos electrónicos, sin recurrir a programas especializados de manipulación de ecuaciones de espacio de estado.

La Figura 3.4 muestra el convertidor Buck linealizado con el modelo de *entrada-salida* en pequeña señal, en donde a partir del análisis correspondiente, se obtiene la función de transferencia establecida en la ecuación (1.12).

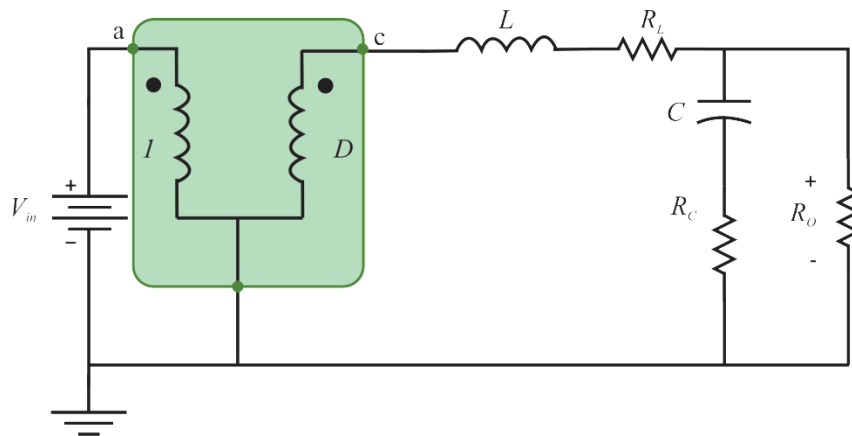


Figura 3.4. Modelo de entrada-salida en pequeña señal.

$$\frac{\hat{v}_{o(s)}}{\hat{v}_{i(s)}} = M \frac{1 + \frac{s}{s_{z1}}}{1 + \frac{s}{w_o Q} + \frac{s^2}{w_o^2}} \quad (1.12)$$

En donde:

$$s_{z1} = \frac{1}{R_c C} \quad (1.13)$$

$$w_o = \sqrt{\frac{1}{LC} \frac{1 + \frac{R_L}{R}}{1 + \frac{R_c}{R}}} \quad (1.14)$$

$$Q = \frac{1}{w_o} \frac{1}{\frac{L}{R_L + R} + C(R_c + R_L \parallel R)} \quad (1.15)$$

La Figura 3.5 muestra el convertidor Buck linealizado con el modelo de *impedancia de salida* en pequeña señal, en donde a partir del análisis correspondiente, de igual forma, se obtiene la función de transferencia establecida en la ecuación (1.16).

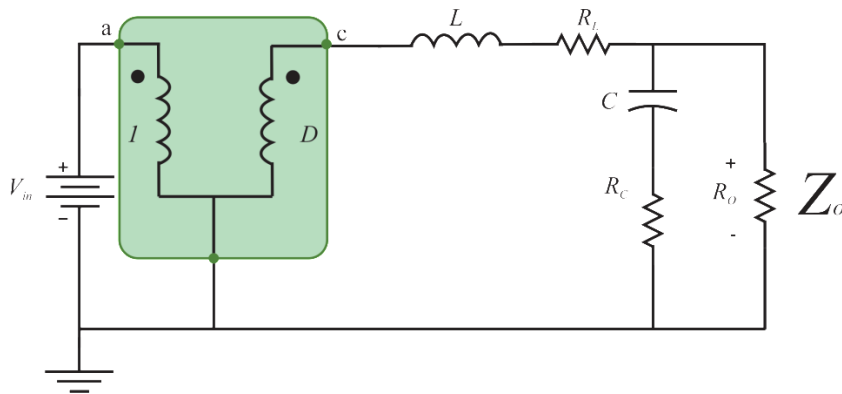


Figura 3.5. Modelo de impedancia de salida en pequeña señal.

$$Z_o = R_L \parallel R \frac{\left(1 + \frac{s}{s_{z1}}\right) \left(1 + \frac{s}{s_{z2}}\right)}{1 + \frac{s}{\omega_o Q} + \frac{s^2}{\omega_o^2}} \quad (1.16)$$

En donde:

$$s_{z1} = \frac{R_L}{L} \quad (1.17)$$

$$s_{z2} = \frac{1}{R_c C} \quad (1.18)$$

Por último la Figura 3.6, presenta el modelo en pequeña señal de *control-salida* correspondiente al convertidor Buck, y la ecuación (1.19) muestra la función de transferencia obtenida para el lazo cerrado para el convertidor Buck. Cabe destacar que dicha ecuación se utiliza para implementar el esquema de control del convertidor Buck dentro del asistente.

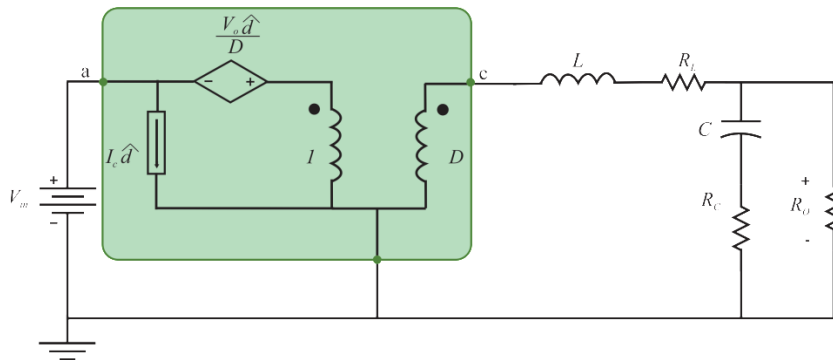


Figura 3.6. Modelo de control-salida en pequeña señal.

$$\frac{\widehat{v}_o(s)}{\widehat{d}(s)} = V_{in} \frac{1 + \frac{s}{s_{z1}}}{1 + \frac{s}{\omega_o Q} + \frac{s^2}{\omega_o^2}} \quad (1.19)$$

## **CAPÍTULO IV. DESARROLLO DEL ASISTENTE**

#### **4.1 Descripción general del Asistente.**

Primeramente se establece la plataforma de diseño del asistente, la cual está desarrollada sobre MATLAB y SIMULINK, considerando el conocimiento previo que se tiene sobre ésta plataforma y sobre todo, la modularidad para realizar trabajos futuros sobre el asistente.

En la Figura 4.1, se presenta el diagrama de flujo del asistente, que permite comprender mejor el funcionamiento general de la herramienta, así como las validaciones que presenta y el proceso que realiza para la obtención de diferentes resultados que se muestran dentro del asistente.

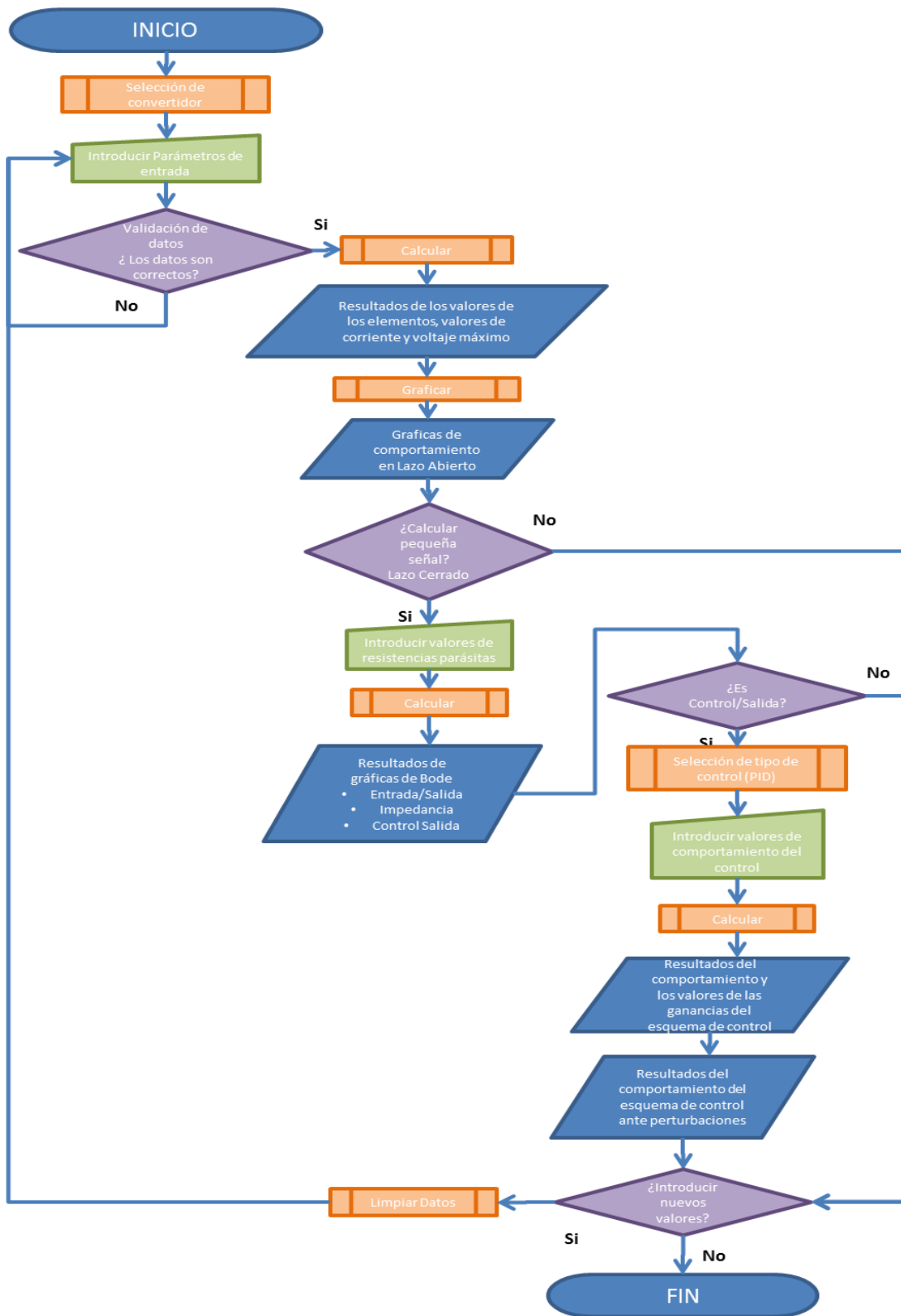


Figura 4.1. Diagrama de flujo del Asistente.

## 4.2 Estructura del Asistente.

La estructura de ejecución del asistente se presenta en la Figura 4.2, está primeramente conformado, por el menú de inicio, posteriormente el menú que establece los datos de entrada y de ésta manera presentar los resultados en lazo abierto, en caso de que el usuario lo requiera, permite continuar con los datos de entrada para establecer las funciones de transferencia del lazo cerrado para el convertidor Buck, y por último, se encuentra la configuración para establecer el control PID que satisfaga las necesidades del usuario, así como la validación del mismo control ante diferentes perturbaciones.

Cabe destacar que cada uno de estos procesos se encuentra de manera separada y con llamados de las rutinas de ejecución, dependiendo de las necesidades del usuario, con la intención de contar una modularidad para agregar más topologías en el asistente y complementarlo con trabajos futuros. La Figura 4.2 se muestra la estructura de funcionamiento y programación con la que se encuentra integrado el asistente, con el objetivo de visualizar la modularidad en que se configura para integrar nuevas topología y funcionalidades.

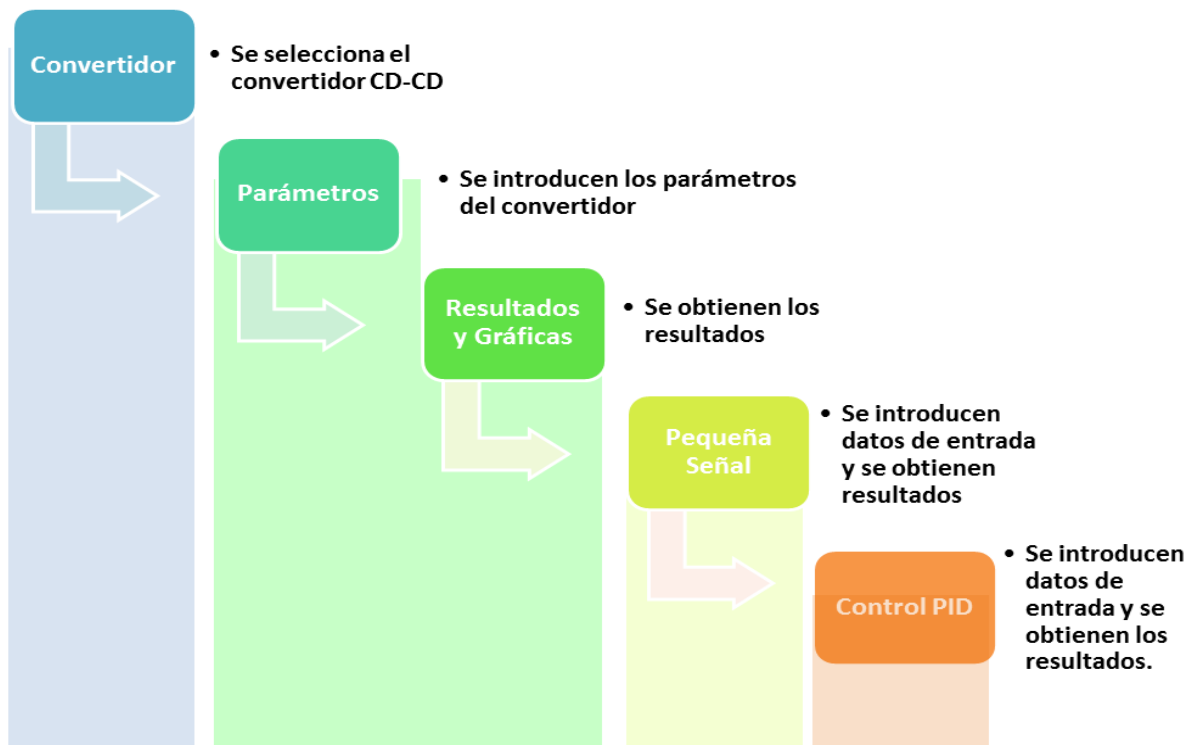


Figura 4.2. Estructura de funcionamiento y programación del asistente.

El Asistente para el Análisis y Diseño de Topologías comunes de Convertidores CD-CD, permite al usuario de forma interactiva simular un sistema propuesto, para posteriormente apoyar en la implementación del mismo de manera cómoda y sencilla. Considerando realizar un sinnúmero de correcciones en el diseño en el asistente antes de su implementación física, validando el comportamiento del diseño propuesto dentro de la herramienta. Para la realización del asistente se ha utilizado la herramienta GUIDE (Graphical User Interface Development Environment).

El asistente se encuentra estructurado en cuatro cuadrantes, el primero corresponde a los datos de entrada del convertidor, el segundo cuadrante, ofrece los valores de los elementos calculados que deberán integrar al convertidor propuesto, el tercer cuadrante, se muestra el comportamiento del convertidor en lazo abierto, y el cuarto cuadrante, se configuran y muestran los resultados en lazo cerrado, en conjunto con el comportamiento del control PID asociado al convertidor, ver Figura 4.3.

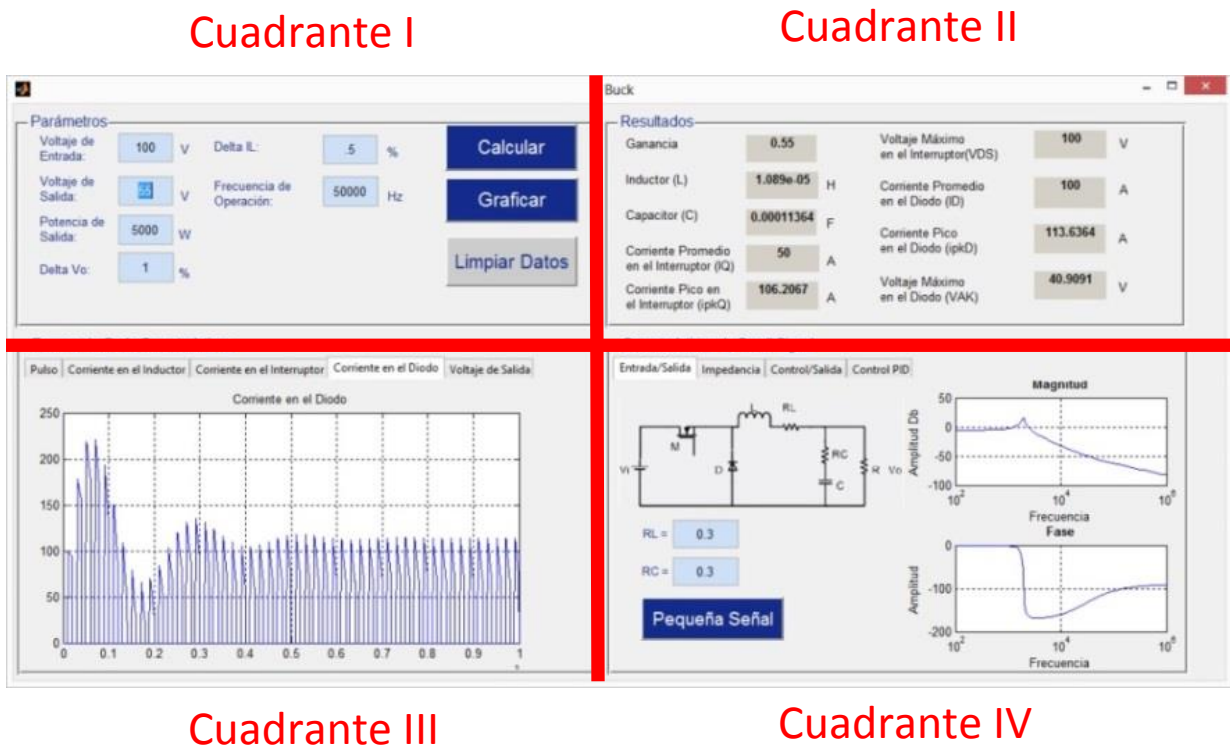


Figura 4.3. Cuadrantes del Asistente.



### **4.3 Diseño y Modelado de la Topología Buck.**

Cuando se diseña una topología de algún convertidor en específico, es necesario comprender el funcionamiento de cada uno de los elementos que integran al convertidor, así como la respuesta de tensiones y corrientes dentro del mismo. Con las ecuaciones características obtenidas del convertidor Buck, se procede a integrar un esquema de modelado para configurar el diseño dinámico de los elementos que integran la topología del convertidor. Para ello se adecúan las ecuaciones características dentro de SIMULINK para validar sus resultados de manera aislada.

Para modelar un convertidor de CD-CD en SIMULINK, lo podemos realizar de diferentes maneras, mediante la realización del modelo esquemático del circuito en el programa SIMULINK, mediante ecuaciones diferenciales características y diagrama a bloques. Se analizan y validan cada uno de los modelados antes mencionados, considerando los tiempos de respuesta en los resultados, con el objetivo de que el asistente tenga una respuesta eficiente al momento de desplegar los resultados.

Una vez realizada la evaluación, se determina la metodología correspondiente a emplear para el modelado del convertidor Buck, la cual corresponde al modelado mediante ecuaciones diferenciales características que integran a la topología.

### **4.4 Implementación PWM-switch.**

Con las funciones de transferencia obtenidas del análisis de la técnica de modelado PWM-switch se programa, configura e implementan en el asistente, como base de diseño para realizar el controlador PID del convertidor Buck que el usuario configure. Es de destacar que las ecuaciones obtenidas se integran en SIMULINK y de igual manera se valida la respuesta en la obtención de los resultados. En el asistente, se determina presentar los diagramas de Bode para cada una de las funciones de transferencia; *entrada-salida*, *impedancia de salida* y *control salida*.

## 4.5 Control PID.

Considerando los resultados obtenidos a partir de la técnica PWM SWITCH y la información que presenta el asistente, se implementa un controlador proporcional-integral-derivativo (PID). Dicha implementación, se realiza como una acción prioritaria, considerando el valor agregado y el impacto en el asistente, con la obtención del controlador para el convertidor y realizando su validación con el circuito en lazo cerrado.

La Figura 4.4 presenta el diagrama a bloques de un controlador PID continuo actuando sobre una señal de error  $e(t)$ .

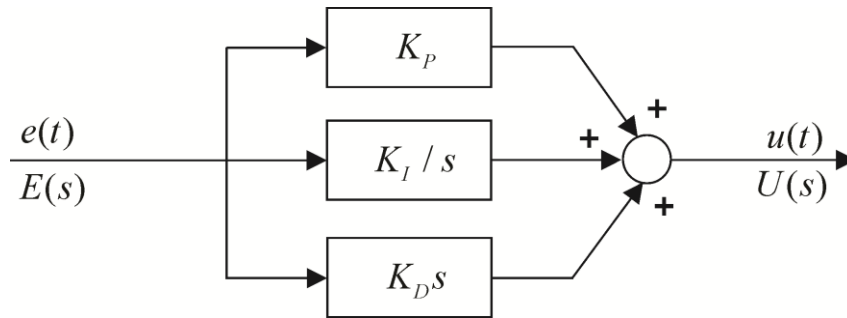


Figura 4.4. Controlador PID continuo.

Se estudian y analizan tres tipos de técnicas de control PID (método Analítico, Ziegler-Nichols y Cancelación de polos y ceros), y se establece implementar el método analítico, buscando satisfacer tanto las especificaciones en régimen transitorio (amortiguamiento), como en régimen permanente (error en estado estacionario), de igual forma el método analítico permite determinar, de manera no iterativa los valores del controlador,  $K_P$ ,  $K_I$  y  $K_D$ , misma que se comprueba en su funcionalidad y se programa el algoritmo de cálculo dentro de SIMULINK, por último se configura e integra su funcionamiento para ser aplicado dentro del asistente.

#### **4.6 Programación de Rutinas de Diseño.**

Una vez definida la funcionalidad del asistente, se realiza la programación de los resultados obtenidos de los cálculos de los elementos, corrientes y tensiones que integran la topología del convertidor Buck, de igual forma la interacción entre SIMULINK y el modelado de los convertidores, con la intención de obtener las gráficas de comportamiento del convertidor que se muestran al usuario, también se realiza la programación de la técnica de modelado PWM SWITCH y las gráficas de los bodes para cada una de las funciones de transferencia del convertidor Buck, así como la programación del algoritmo de control PID mediante el método analítico (método que considera tanto las especificaciones en estado estacionario como el estado transitorio). En el anexo A, se agrega toda la estructura de funcionamiento y programación con la que se encuentra realizado.

#### **4.7 Interfaz de Usuario.**

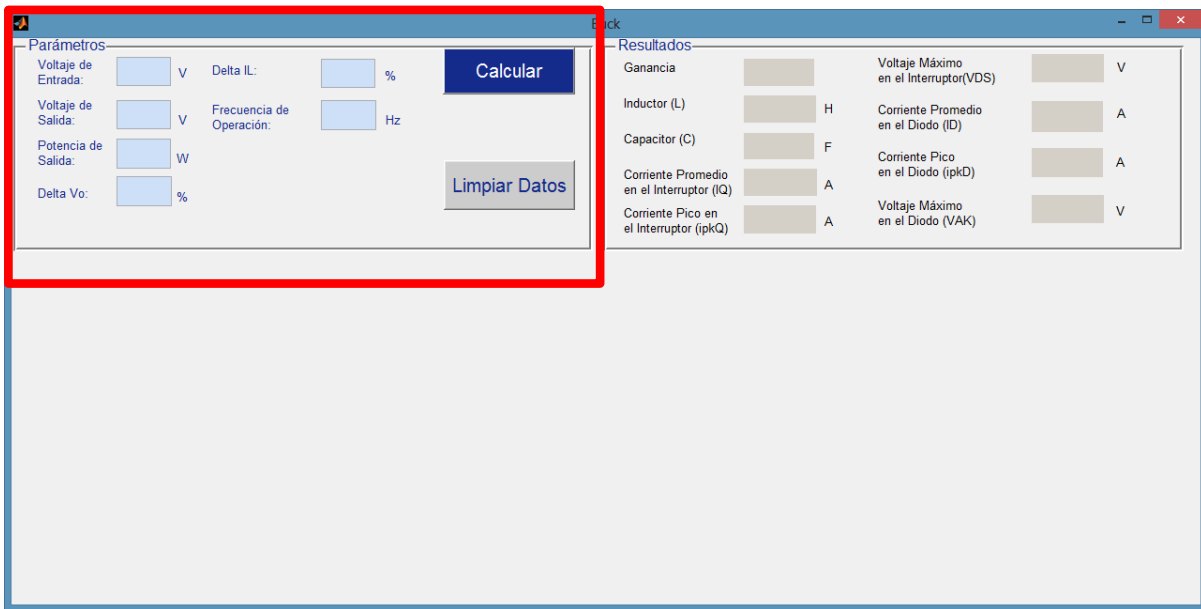
Interfaz gráfica de usuario (En inglés Graphic User Interface, también conocido con su acrónimo GUI en MATLAB), es un método para facilitar la interacción del usuario con la computadora a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas, etc.) además de texto. Surge como evolución de la línea de comandos de los primeros sistemas operativos y es pieza fundamental para un entorno gráfico.

Se realiza la interfaz, la cual contiene la información más importante para analizar y diseñar un convertidor Buck. La interfaz se desarrolla con el GUI de MATLAB, es una interfaz dinámica en cuanto a los datos, ya que permite actualizar los campos de los resultados del asistente, las veces que el usuario lo requiera, de igual manera es en donde se concentran los llamados para realizar las ejecuciones de las rutinas de diseño del asistente. La Figura 4.5 muestra la interfaz de usuario de inicio del asistente. En esta ventana se muestra la descripción de la herramienta, y es en donde el usuario determina la topología para analizar y diseñar dentro del asistente. Es de resaltar que en esta primera ventana se encuentran como topologías opcionales la del Boost, Buck-Boost y Cuk, con la intención de dejar estructurada dicha ventana para agregar las topologías antes descritas en trabajos futuros. Sin embargo la única que permite continuar en la interfaz gráfica es con la topología Buck.



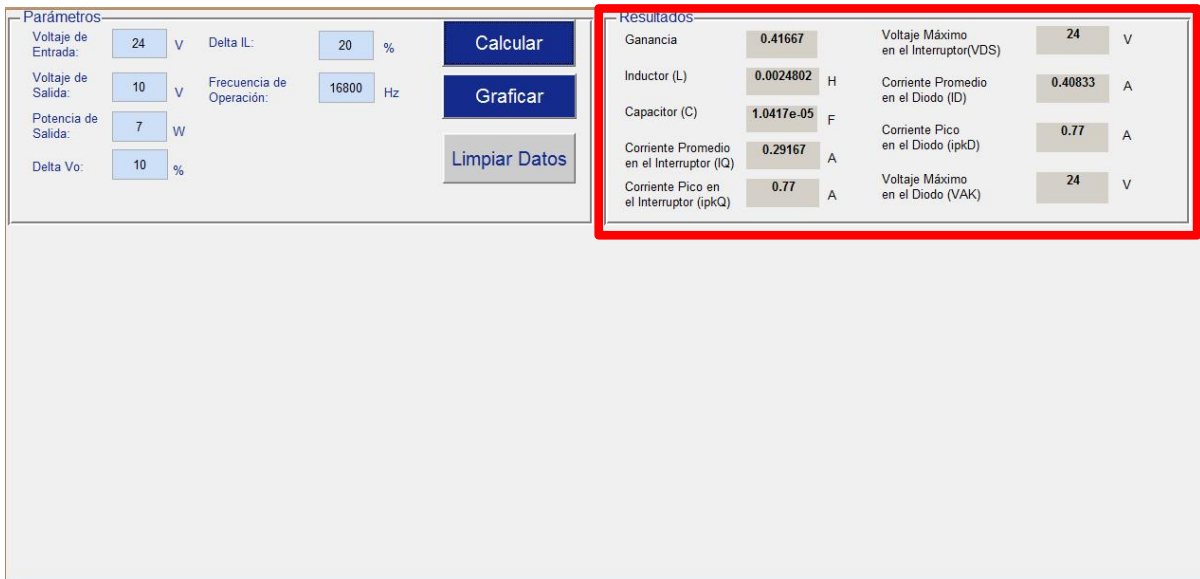
Figura 4.5. Interfaz de usuario inicial del Asistente.

Posteriormente, la Figura 4.6 presenta la interfaz en donde se establecen los datos de entrada del convertidor Buck. Aquí es donde el usuario ingresa en el primer cuadrante, los datos de diseño correspondientes del convertidor Buck que desea analizar y diseñar. Los datos necesarios son: tensión de entrada, tensión de salida, potencia de salida, rizado de corriente deseado en el inductor ( $\Delta i_L$ ), rizado de tensión de salida ( $\Delta V_o$ ) y por último la frecuencia de operación. que integran la topología y los esfuerzos de corriente y tensión del convertidor.



**Figura 4.6. Interfaz de usuario de parámetros de entrada del convertidor.**

Con los datos de entrada solicitados en el primer cuadrante, en la Figura 4.7 se observa en el segundo cuadrante, el despliegue de los resultados y parámetros de los elementos que integran el convertidor requerido por el usuario. Aquí permite visualizar los esfuerzos de tensión y corriente para el diodo y el dispositivo de conmutación, para posteriormente examinar los elementos indicados para la topología propuesta, así mismo muestra el valor tanto del inductor y del capacitor que debe incluir el convertidor configurado por el usuario.



**Figura 4.7. Interfaz de usuario de parámetros del convertidor con resultados en el segundo cuadrante.**

Para la obtención de los resultados de comportamiento en el tercer cuadrante, se utiliza el modelado del convertidor con sus ecuaciones características implementado en SIMULINK, en donde la programación del asistente realiza la simulación correspondiente y se obtiene el comportamiento del convertidor Buck propuesto por el usuario. Cabe señalar que éste proceso es transparente para el usuario, ya que es un cálculo que se realiza internamente con programación y permite guardar los resultados en SIMULINK, para posteriormente ser visualizados directamente en el asistente. La Figura 4.8 muestra el modelado en SIMULINK de las ecuaciones características del convertidor Buck, en donde se realiza la simulación correspondiente y se establece el comportamiento de la topología en lazo abierto.

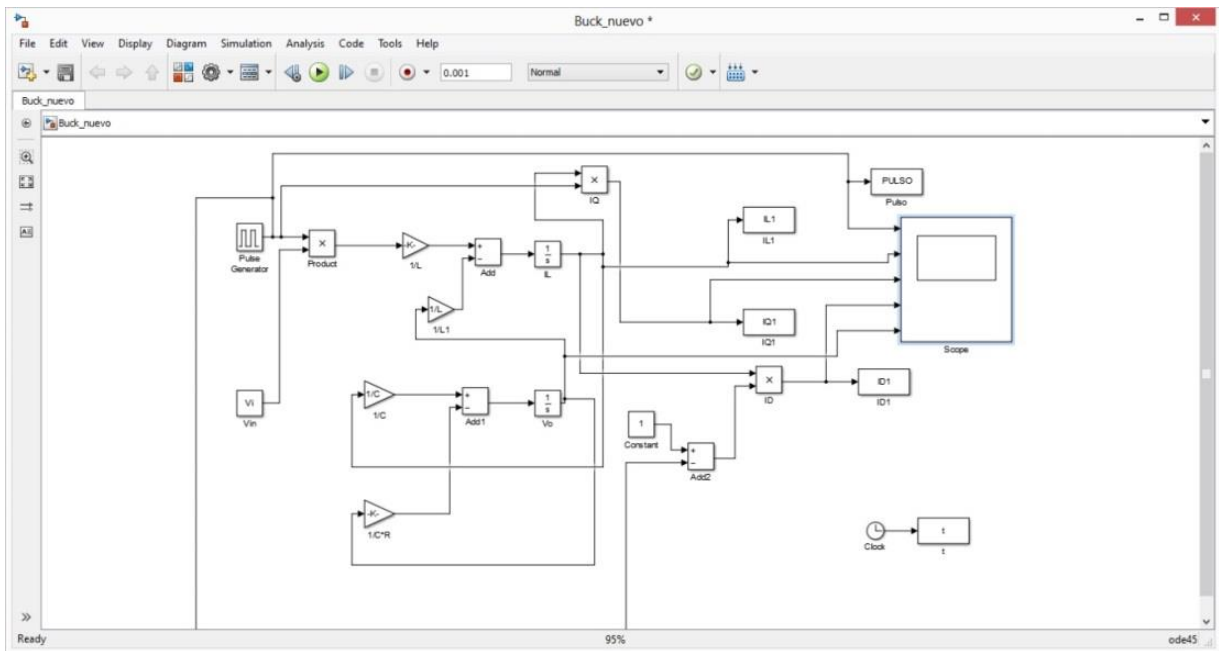
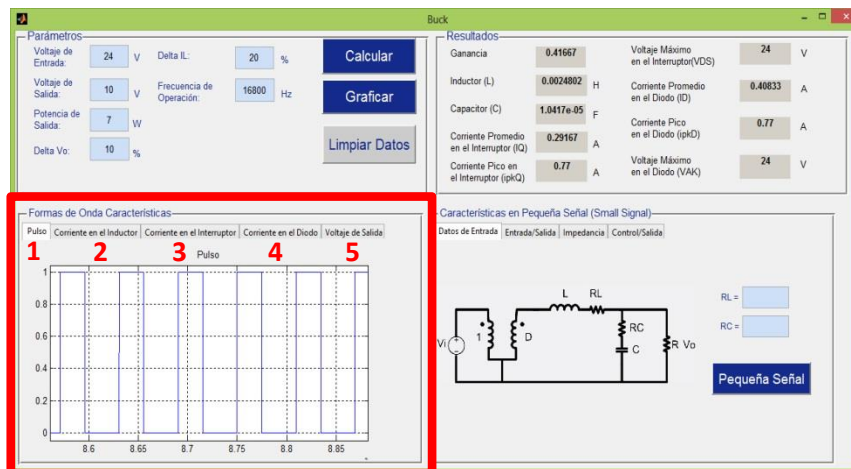


Figura 4.8. Implementación en Simulink de las ecuaciones características del Convertidor Buck

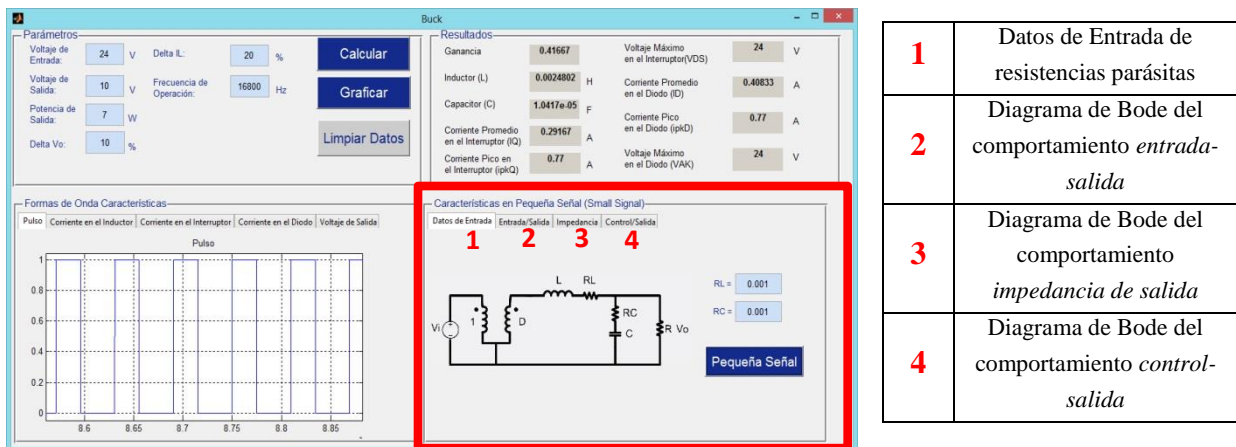
Siguiendo el proceso dentro del asistente, en la Figura 4.9 se muestran las gráficas de comportamiento en lazo abierto del convertidor propuesto por el usuario en el tercer cuadrante, el asistente ofrece cuatro pestañas de visualización con resultados gráficos; la primera corresponde al pulso de disparo del dispositivo de conmutación, la segunda al comportamiento de la corriente en el inductor, la tercera al comportamiento de la corriente en el diodo y la última el comportamiento de la tensión de salida.



1	Pulso de disparo
2	Comportamiento de corriente en el inductor
3	Comportamiento de corriente dispositivo de conmutación
4	Comportamiento de corriente en diodo
5	Comportamiento del tensión de salida

Figura 4.9. Resultados gráficos en lazo abierto del convertidor Buck.

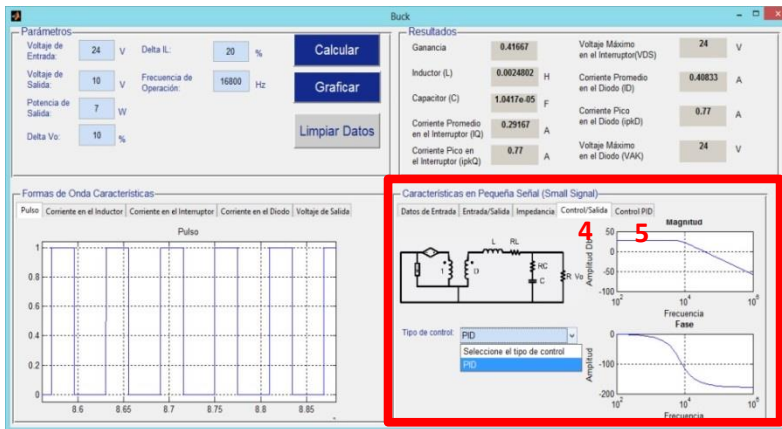
En el cuarto cuadrante la Figura 4.10 muestra la configuración de las características en pequeña señal, en donde el usuario a partir de los valores de las resistencias parásitas tanto en el inductor como en el capacitor, obtiene el comportamiento de los diagramas de bode de las funciones de transferencia de entrada-salida, impedancia de salida y control-salida, en cada una de las pestañas correspondientes.



**Figura 4.10.** Configuración de componentes parásitos y obtención de diagramas de Bode.

De igual forma en el cuarto cuadrante, el asistente permite seleccionar el esquema de control PID del convertidor establecido por el usuario, una vez que se obtiene diagrama de Bode de la función de transferencia *control-salida*. Una vez seleccionado el control PID, se despliega una pestaña adicional, en donde permite configurar el comportamiento del control PID que el usuario desee, ver Figura 4.11. Es importante mencionar que para realizar la configuración del control, el usuario lo puede visualizar y rehacer tantas veces como sea necesario, hasta satisfacer el criterio de comportamiento deseado. Así mismo, el asistente presenta los valores de las ganancias  $K_P$ ,  $K_I$  y  $K_D$ , del controlador. Con esto se pretende, que el usuario implemente, de manera más rápida, dichas ganancias ya sea analógica o digitalmente, para establecer el lazo de control del convertidor.

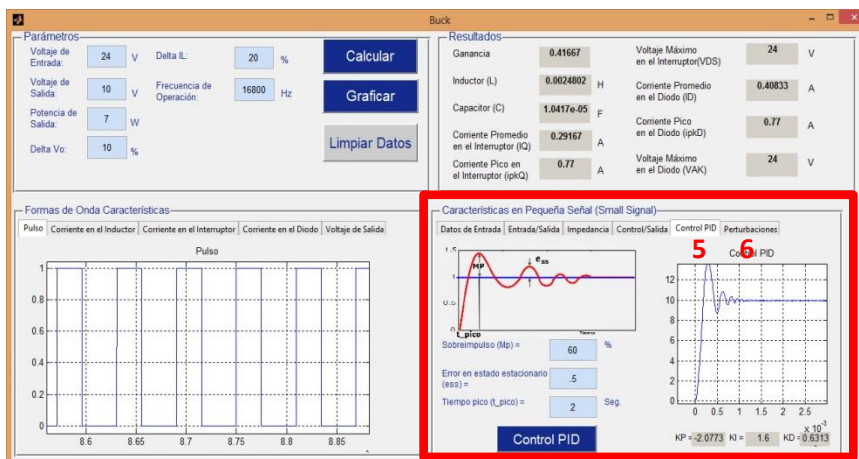




<b>4</b>	Diagrama de Bode del comportamiento <i>control-salida</i>
<b>5</b>	Configuración del esquema de control PID

Figura 4.11. Selección del control y despliegue de configuración de parámetros del control PID.

Una vez que se ha cumplido con el criterio de comportamiento de control establecido por el usuario, como se muestra en la Figura 4.12, la herramienta permite visualizar en una pestaña adicional, el comportamiento del esquema de control ante variaciones de carga y perturbaciones de incremento o decremento en la tensión de entrada, como se muestra en la Figura 4.13.



<b>5</b>	Configuración del esquema de control PID
<b>6</b>	Comportamiento del control PID ante diferentes perturbaciones

Figura 4.12. Configuración de parámetros del control PID

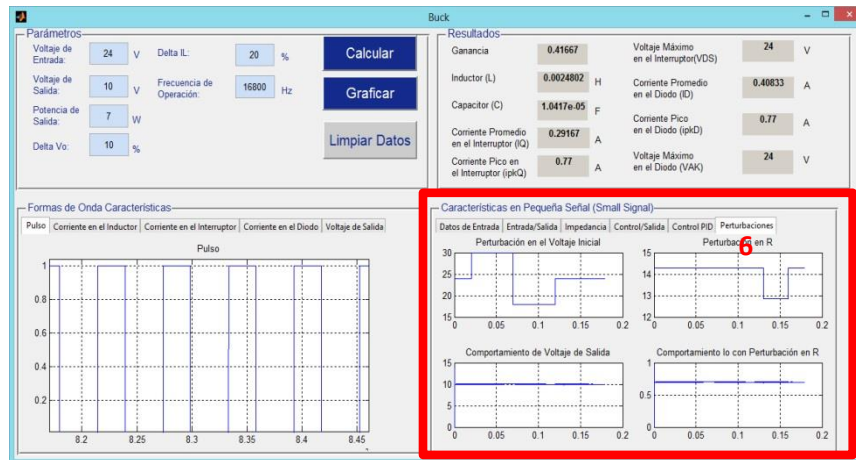


Figura 4.13. Respuesta del control PID ante perturbaciones

#### 4.8 Uso del Asistente en un convertidor Buck.

Para realizar la implementación del convertidor Buck, primeramente es necesario establecer los requerimientos de diseño, la Tabla 4.1 muestra las especificaciones del convertidor para analizar su comportamiento en lazo abierto y lazo cerrado dentro del Asistente:

Tabla 4.1. Especificaciones Técnicas del Convertidor Buck.

Parámetro	Valor
$V_i$	24V
$V_o$	10 V
$P_o$	7 W
$\Delta V_c$	10 %
$\Delta I_L$	20%
$f_s$	16.8kHz

#### 4.9 Resultados obtenidos en el Asistente en lazo abierto.

Con base en las especificaciones de diseño se procede a introducirlos, en el primer cuadrante del asistente, para obtener los valores y consideraciones, de los elementos que integran al convertidor. La Figura 4.14 muestra los resultados obtenidos en el segundo cuadrante del asistente.

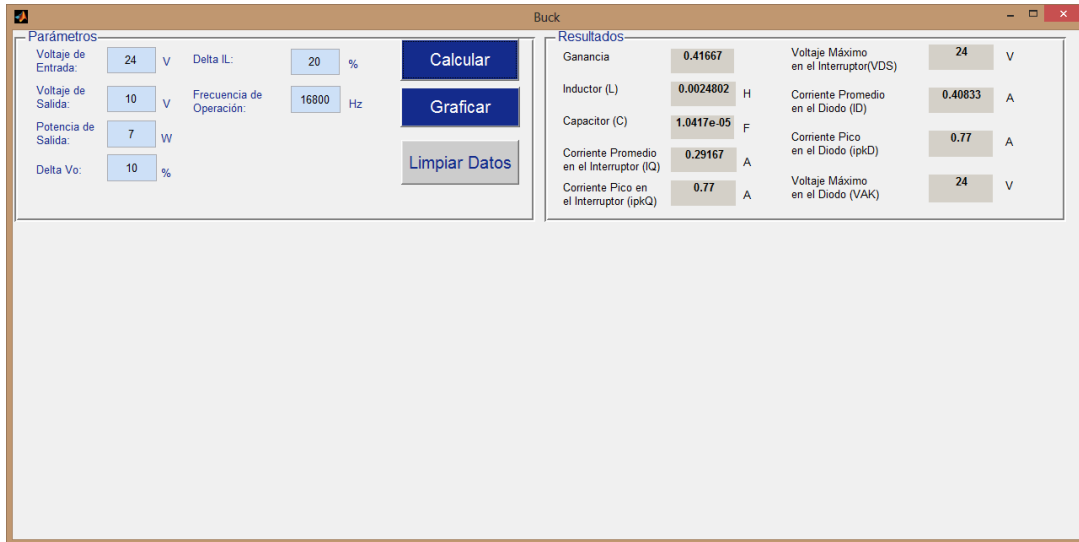


Figura 4.14. Resultados de los elementos y consideraciones para la implementación del convertidor Buck.

A continuación, en el tercer cuadrante, se gráfica y analiza el comportamiento del pulso de disparo del dispositivo de conmutación (Figura 4.15), la corriente que transita por el inductor (Figura 4.16), la corriente en el dispositivo de conmutación (Figura 4.17), la corriente en el diodo (Figura 4.18) y por último la tensión de salida del convertidor propuesto (Figura 4.19).

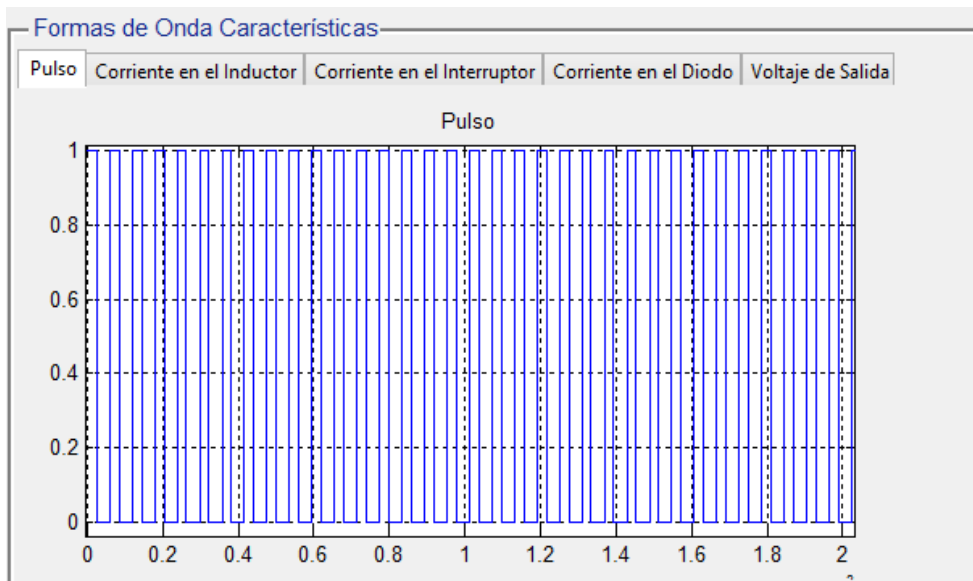


Figura 4.15. Pulso de disparo del dispositivo de conmutación.

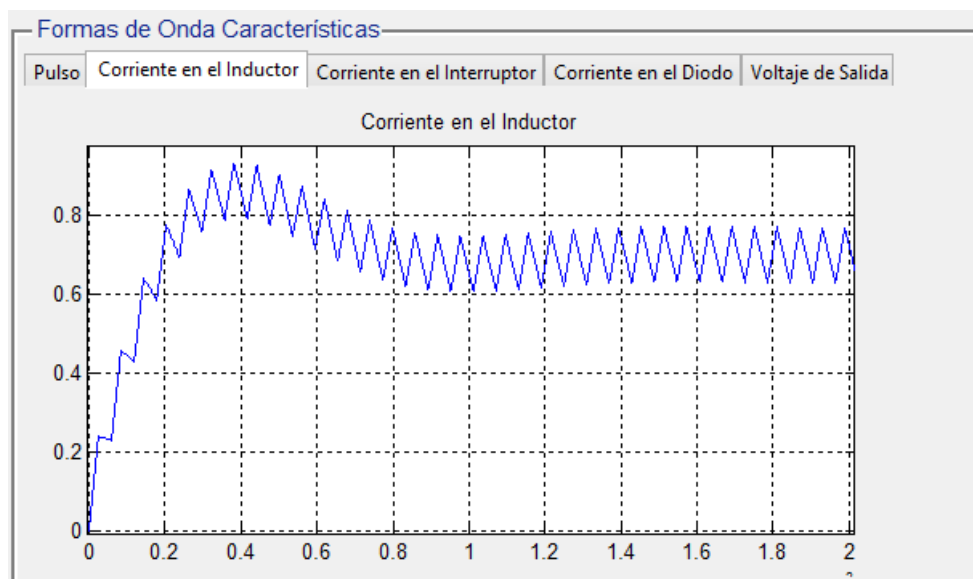


Figura 4.16. Forma de onda de la corriente en el inductor.

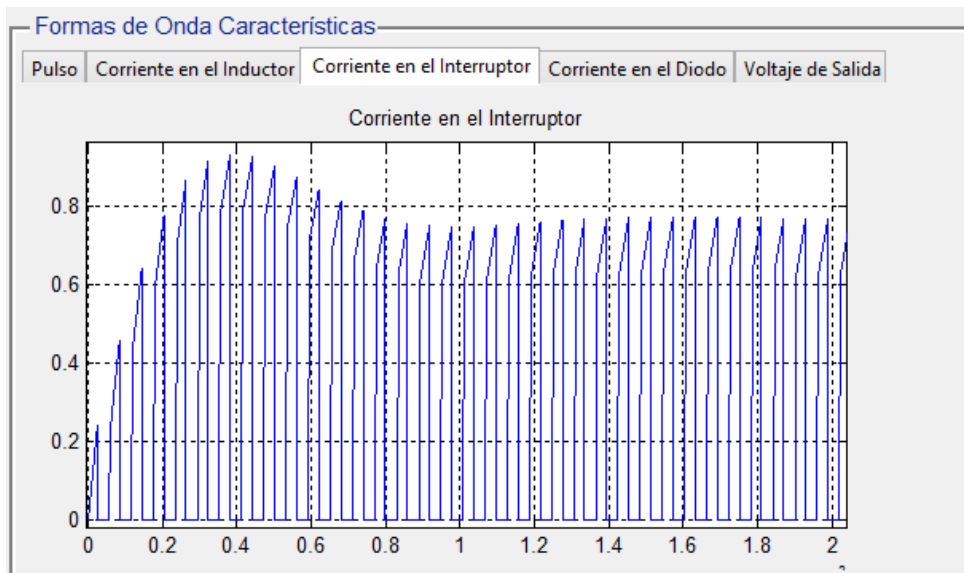


Figura 4.17. Forma de onda de la corriente en el interruptor.

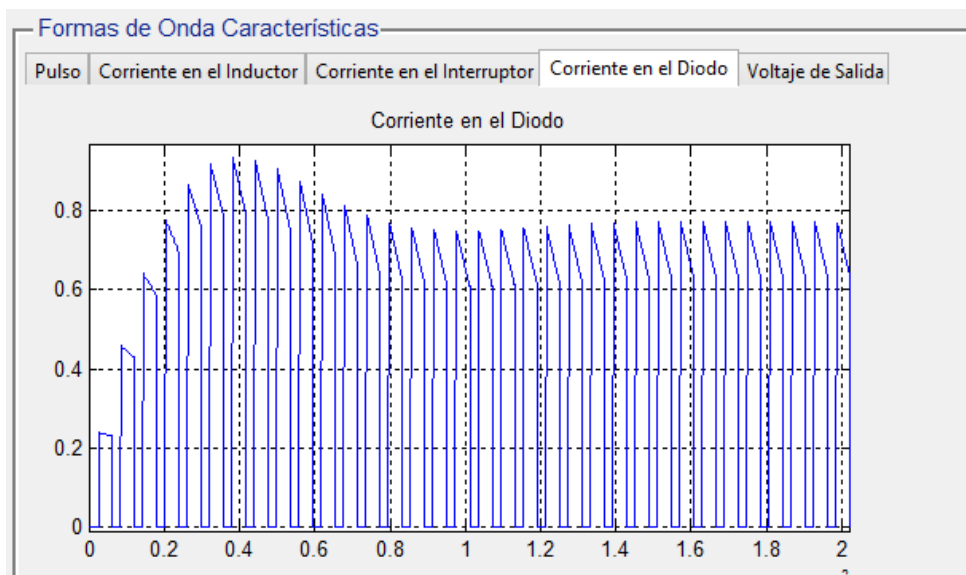


Figura 4.18. Forma de onda de la corriente en el diodo.

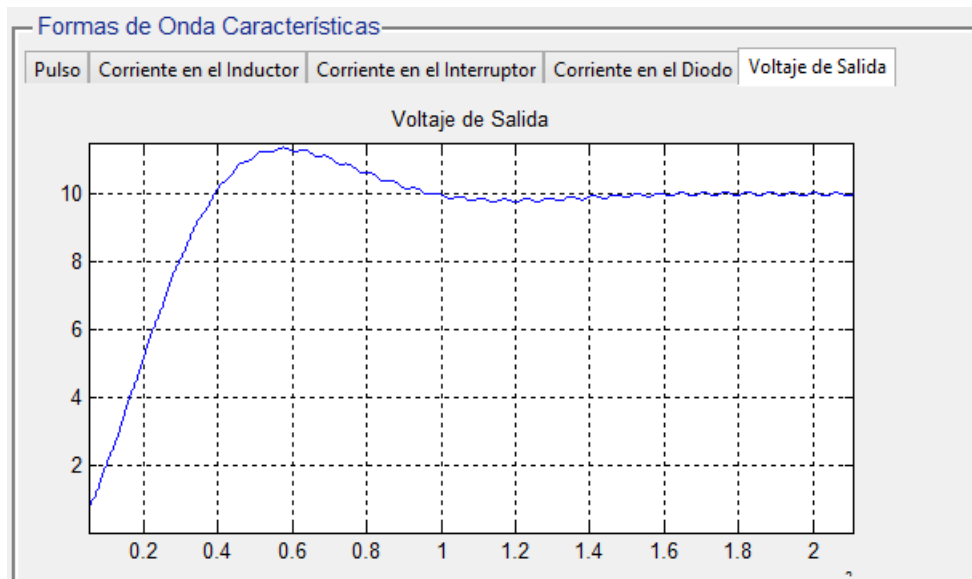


Figura 4.19. Forma de onda de la tensión de salida.

#### 4.10 Resultados obtenidos en el Asistente en lazo cerrado.

Una vez que se analiza y valida el comportamiento del convertidor propuesto, en el cuarto cuadrante, se establecen los valores de los componentes parásitos del inductor y el capacitor, para que el asistente realice el análisis en pequeña señal, y de esta manera, obtener los diagramas de Bode de las funciones de transferencia respectivamente. La Figura 4.20 muestra la configuración de las resistencias parásitas del convertidor propuesto, así como los diagramas de Bode de las funciones de transferencia de *entrada-salida* (Figura 4.21), *impedancia de salida* (Figura 4.22) y *control-salida* (Figura 4.23).

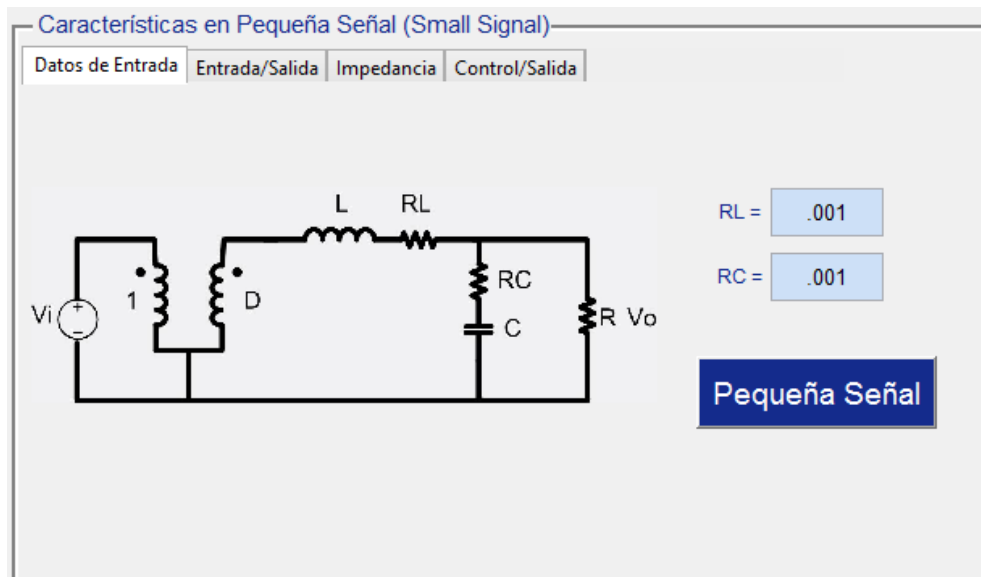


Figura 4.20. Configuración de componentes parásitos.

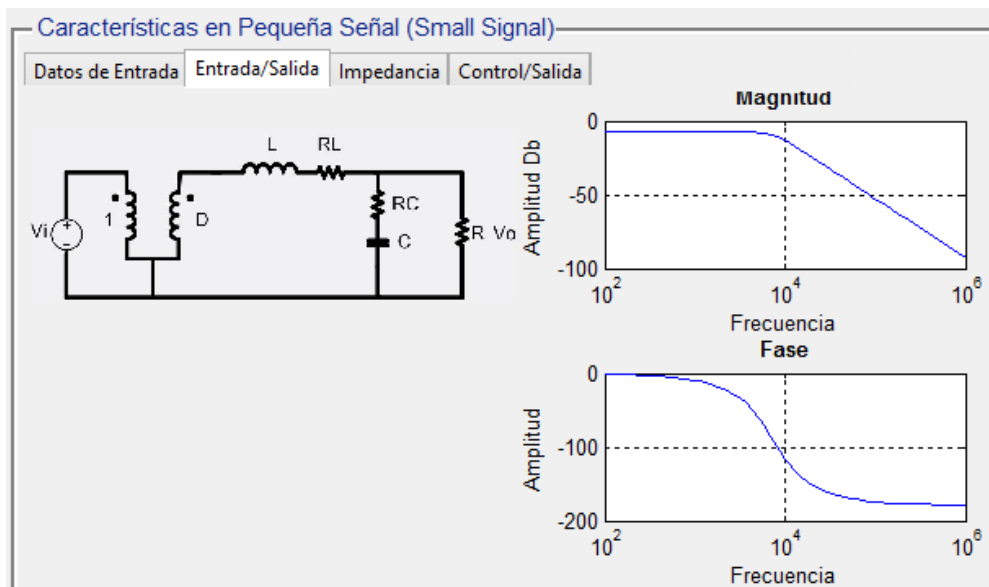


Figura 4.21. Diagrama de Bode de la función de transferencia entrada-salida.

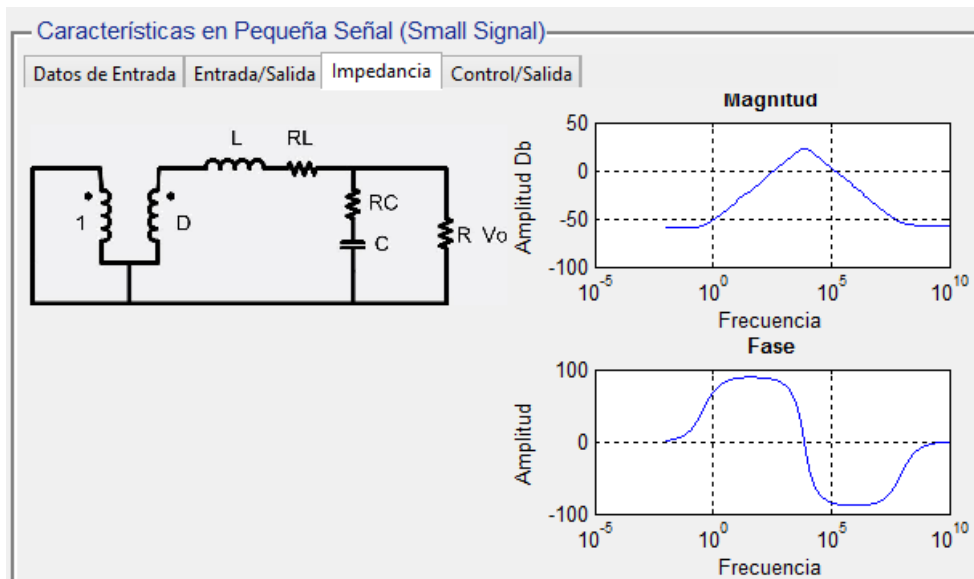


Figura 4.22. Diagrama de Bode de la función de transferencia impedancia de salida.

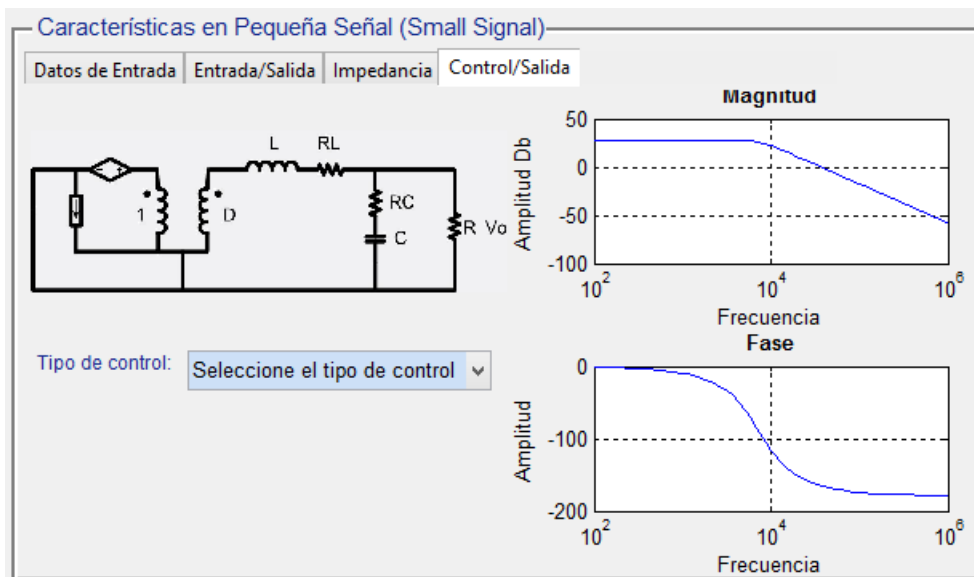


Figura 4.23. Diagrama de Bode de la función de transferencia control-salida.



En el cuarto cuadrante, después de analizar los diagramas de Bode descritos con anterioridad, en la pestaña de *control-salida*, se selecciona el esquema de control PID que ofrece el asistente, para posteriormente desplegar la pestaña en donde se configura el controlador del convertidor propuesto. Es en esta pestaña donde se configura el comportamiento del control PID hasta que se satisface la respuesta del mismo. La Figura 4.24 presenta la selección del control PID, la Figura 4.25 muestra la configuración del esquema de control y por último, la Figura 4.26 muestra la respuesta del controlador ante perturbaciones en la tensión de entrada y en la carga.

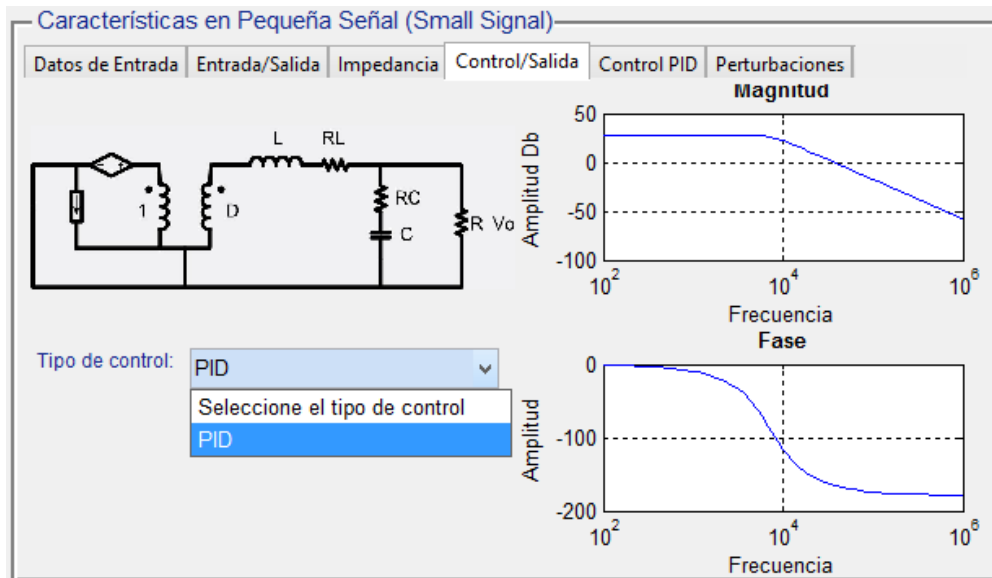


Figura 4.24. Selección del esquema de control PID.

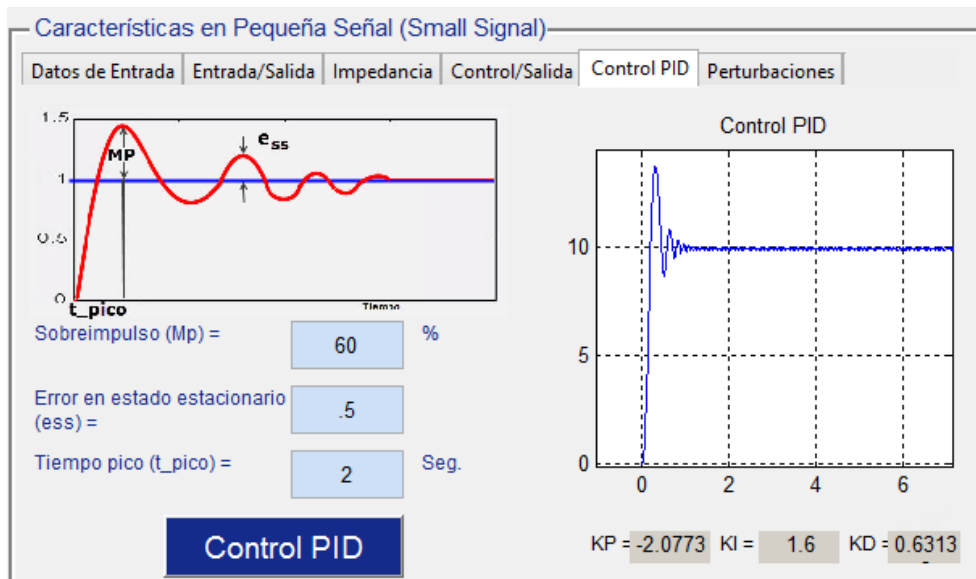


Figura 4.25. Configuración del esquema de control PID.

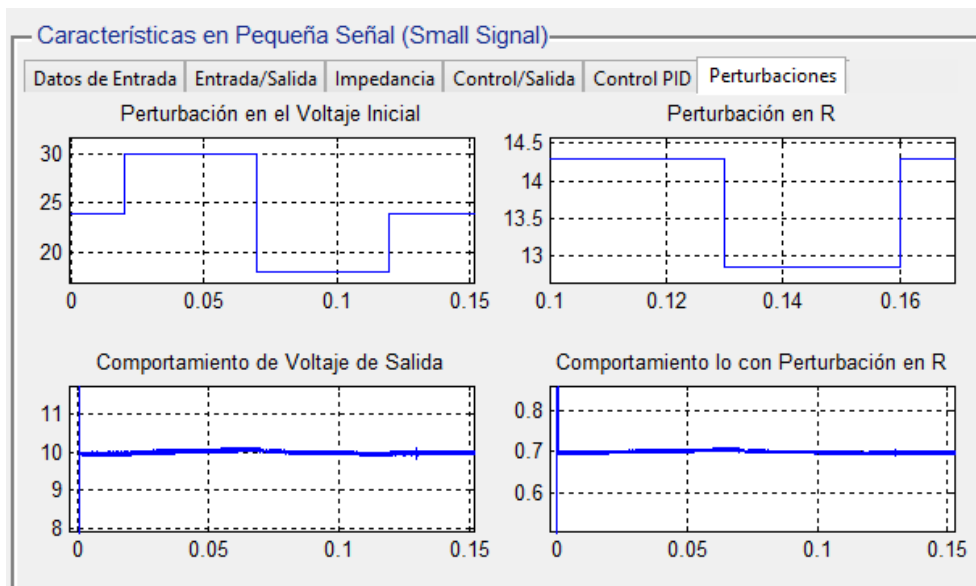


Figura 4.26. Respuesta del control PID ante perturbaciones.

### 4.11 Implementación práctica en laboratorio.

Con los resultados obtenidos de los elementos reactivos a partir del asistente, se establecen los valores para realizar la implementación práctica de la topología del convertidor Buck, para ello se utiliza el software EAGLE 7.3.0 Profesional, en donde se establece tanto el circuito esquemático, ver Figura 4.27, como la placa impresa a realizar, ver Figura 4.28.

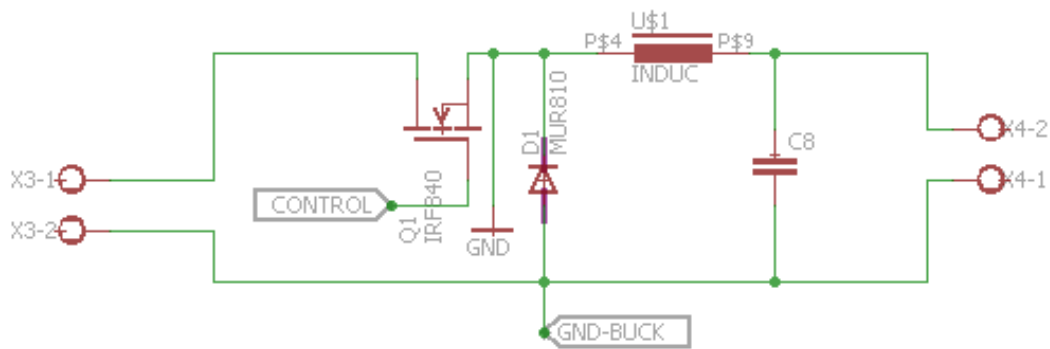


Figura 4.27. Circuito esquemático del convertidor Buck

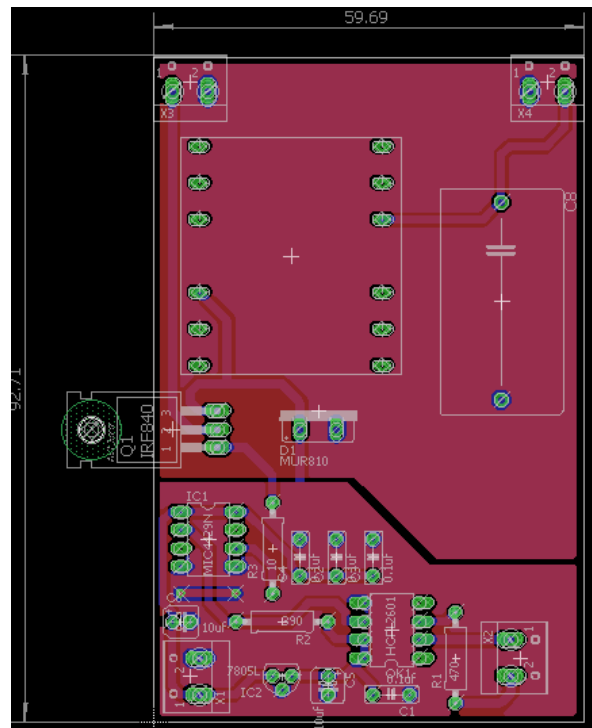
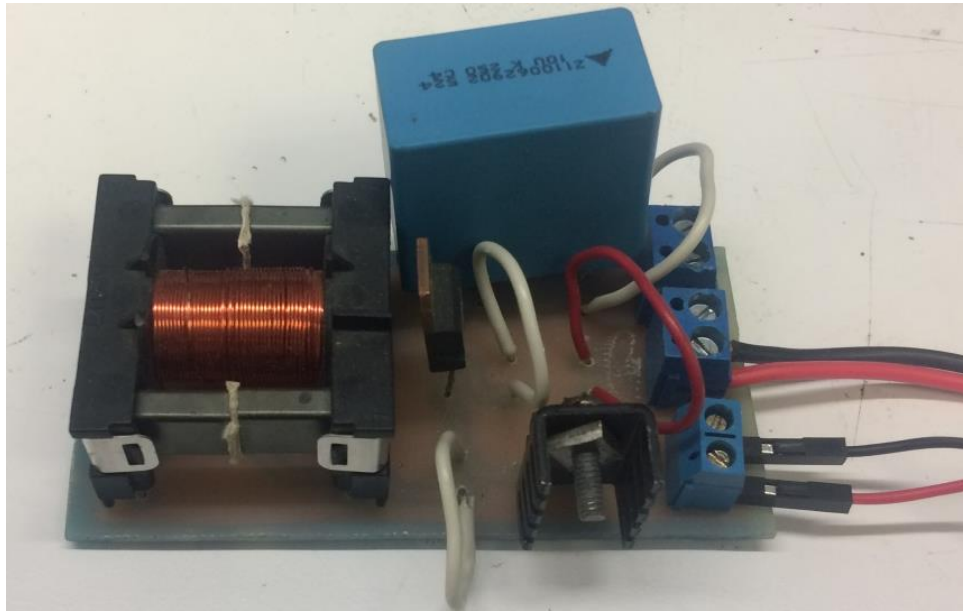


Figura 4.28. Placa del convertidor Buck.

Con respecto a los valores tanto para el MOSFET como para el diodo, se toman en cuenta los rangos de corriente y tensión para cada uno de éstos elementos, determinando el MOSFET IRF840 y el diodo MUR840. Los cuales cumplen con los rangos de tensión y corriente mostrados por el asistente, y de igual manera son componentes que se encuentran en disponibilidad en el laboratorio.

Posteriormente se incorporan todos los elementos que integran al convertidor Buck sobre la placa realizada. La Figura 4.29, presenta el prototipo final implementado en el laboratorio.



**Figura 4.29. Convertidor Buck implementado en el laboratorio.**

#### **4.12 Resultados experimentales en lazo abierto.**

Una vez realizada la placa correspondiente a la topología Buck, se procede a probar y validar la funcionalidad del convertidor en el laboratorio. Se obtienen los resultados en el osciloscopio del comportamiento del convertidor implementado en lazo abierto, los resultados obtenidos corresponden pulso de disparo del dispositivo de conmutación (Figura 4.30), la corriente que transita por el inductor (Figura 4.31), la corriente en el dispositivo de conmutación (Figura 4.32), la corriente en el diodo (Figura 4.33) y por último la tensión de salida del convertidor implementado físicamente (Figura 4.34).

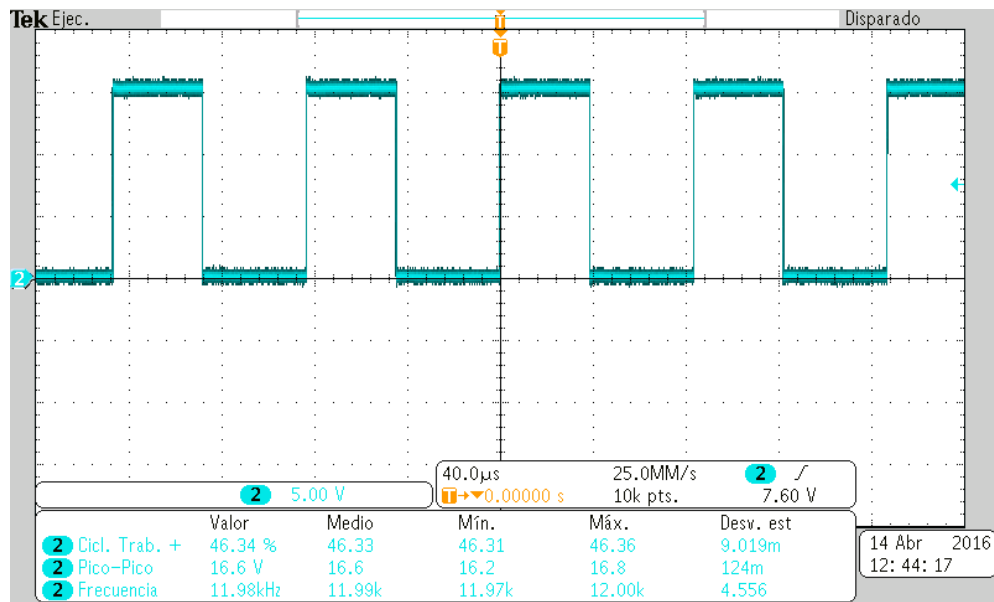


Figura 4.30. Pulso de disparo del dispositivo de conmutación.

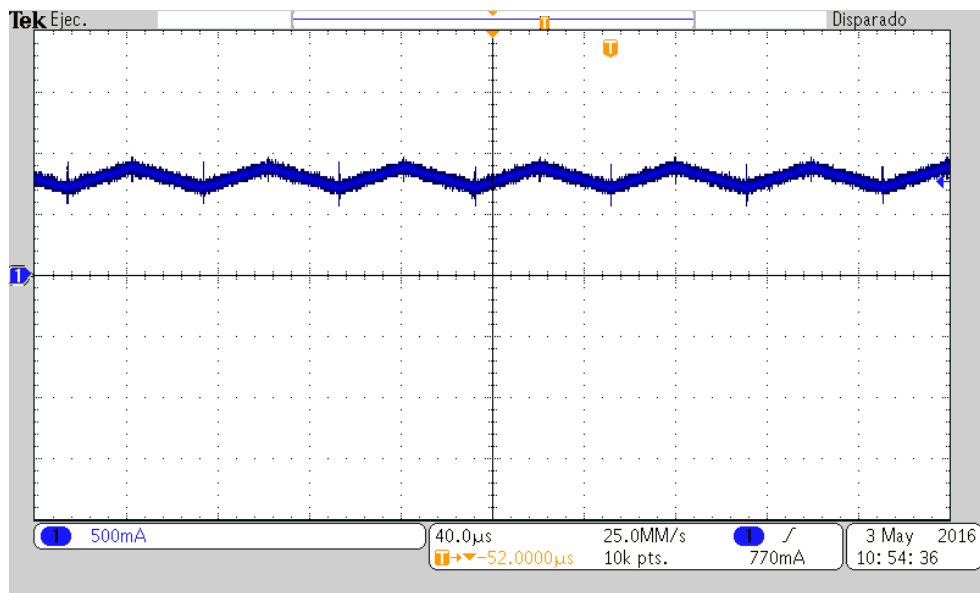


Figura 4.31. Forma de onda de la corriente en el inductor.

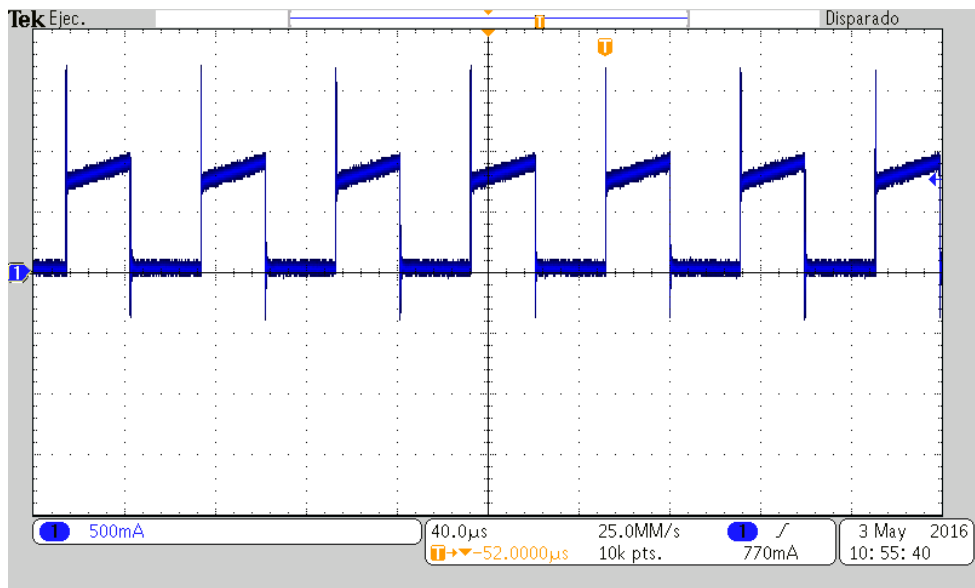


Figura 4.32. Forma de onda de la corriente en el interruptor.

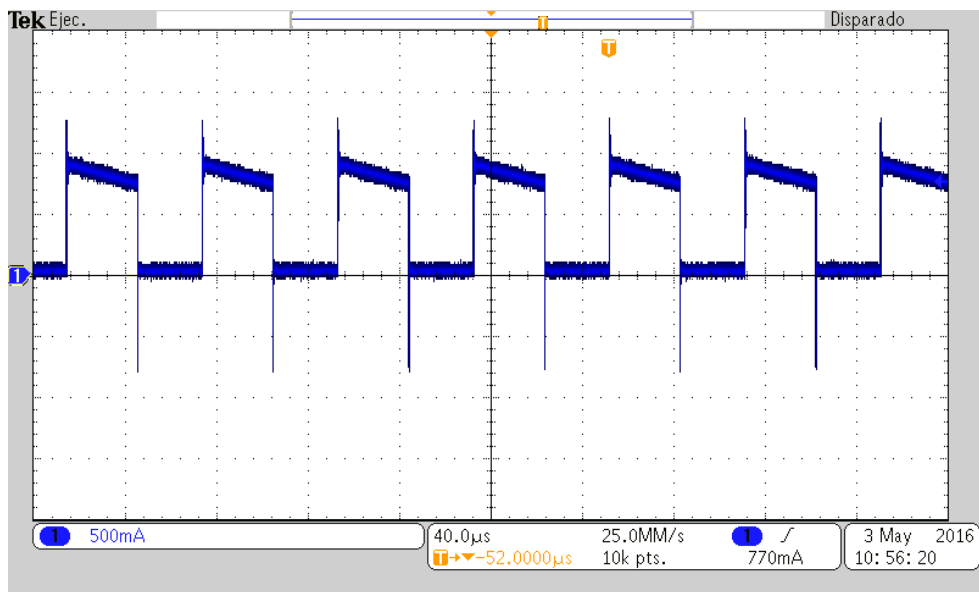


Figura 4.33. Forma de onda de la corriente en el diodo.

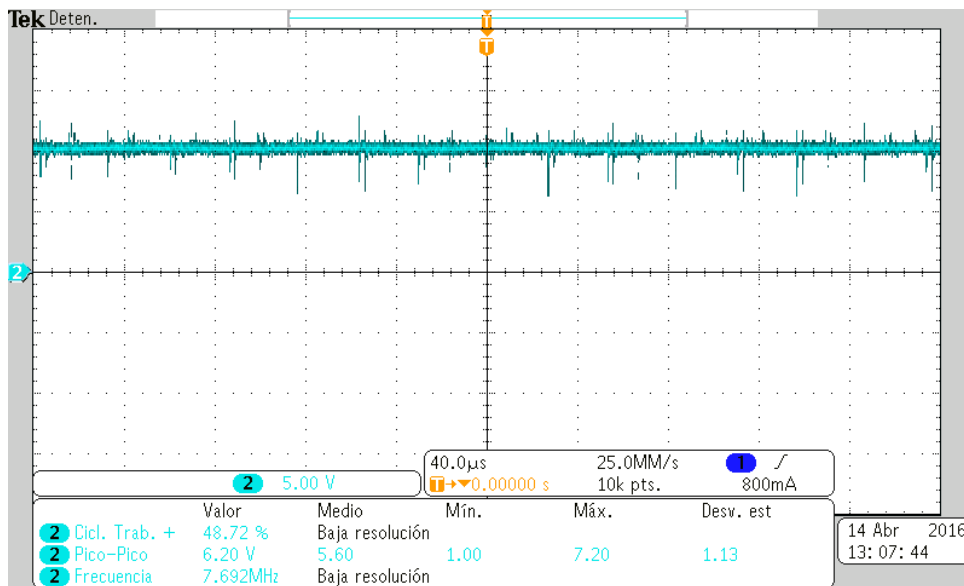


Figura 4.34. Forma de onda de la tensión de salida.

### 4.13 Resultados experimentales en lazo cerrado.

El sistema de control que se implementa para operar al convertidor Buck es un controlador PID, la Figura 4.35 muestra el sistema de control que será integrado al convertidor implementado en el laboratorio.

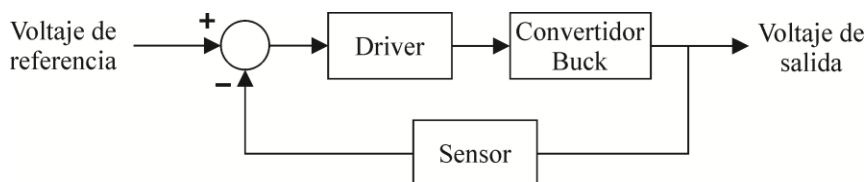


Figura 4.35. Diagrama de bloques de control del convertidor.

El sistema en lazo cerrado está conformado por un control analógico PID, integrado por las ganancias  $K_P$ ,  $K_I$  y  $K_D$ , que fueron obtenidas del asistente de diseño, posteriormente la señal de control entra a un comparador para generar una señal PWM.

Para acoplar el sistema de control con el sistema de potencia, fue necesario hacer utilizar un impulsor, el cual se encarga de variar la conmutación del transistor MOSFET para mantener un tensión constante en la salida del convertidor Buck.

La tensión de salida es la retro alimentación del sistema el cual ingresa nuevamente para ser comparado con la tensión de referencia del controlador PID.

En la Figura 4.36 se muestra el modelo del circuito de control PID.

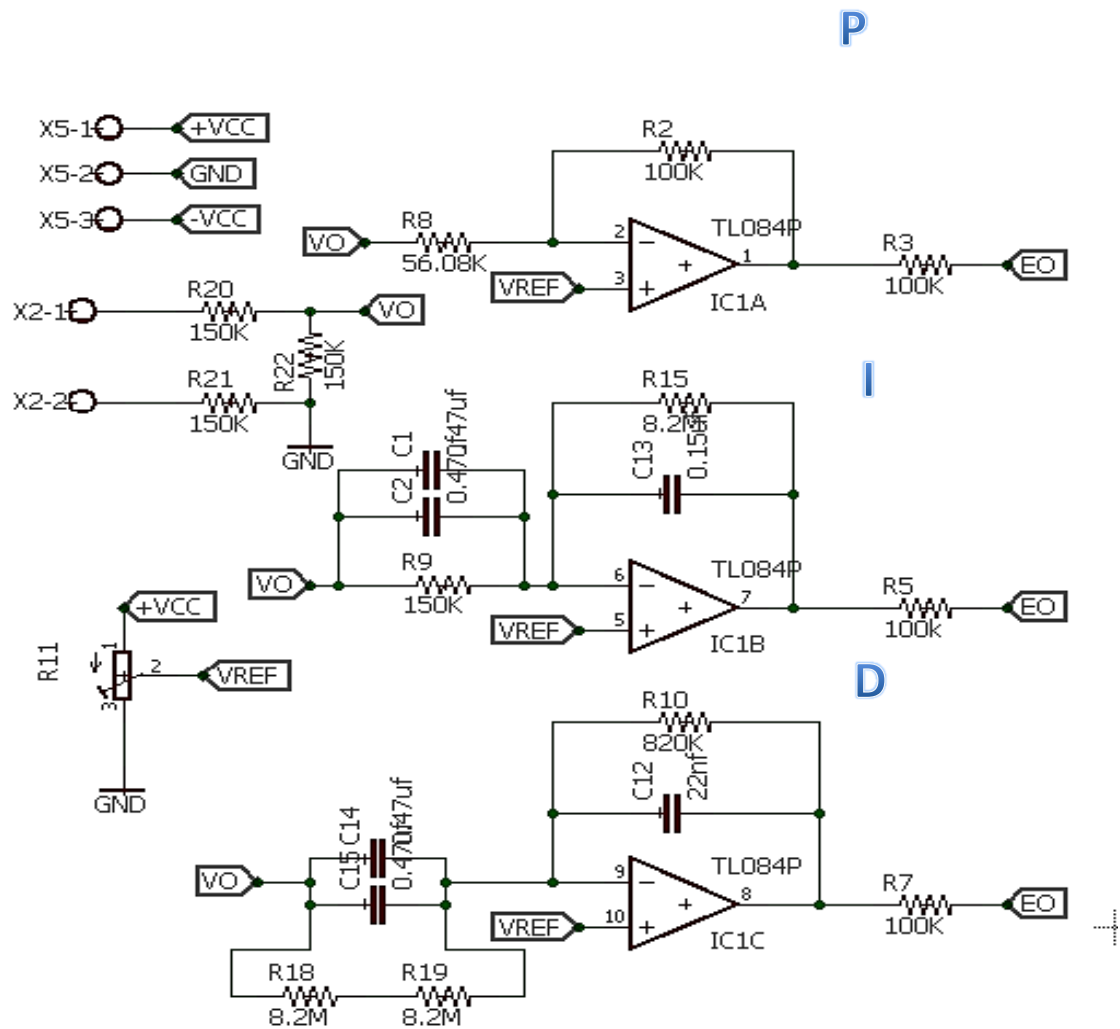


Figura 4.36. Circuito de control PID.



Después de realizar las pruebas del convertidor en lazo abierto se procede a establecer, con las ganancias que se obtienen en el Asistente, el control PID que satisface el control de la tensión de salida establecido. La Figura 4.37, muestra la configuración final obtenida en el Asistente, que permite, al convertidor implementado físicamente, mantener estable la tensión de salida.

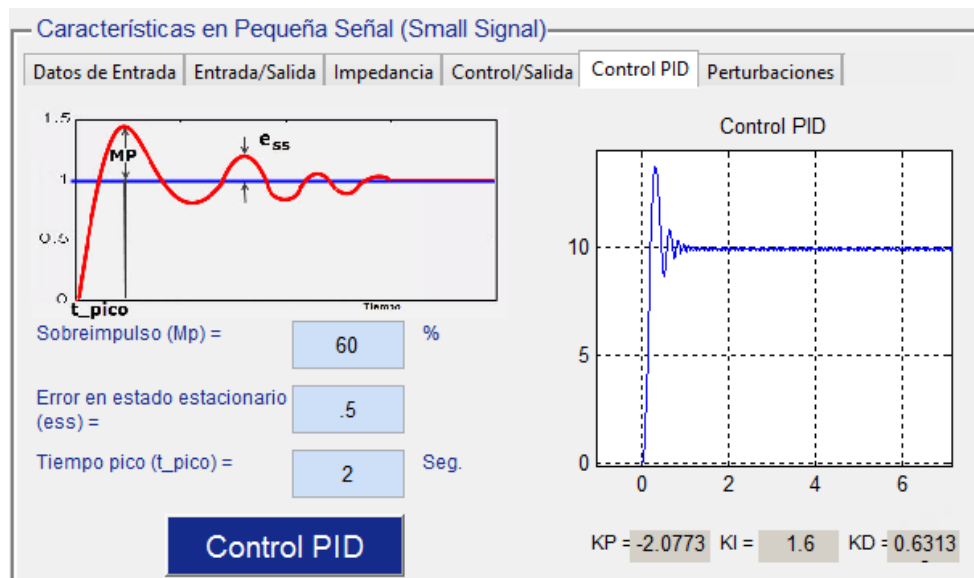


Figura 4.37. Configuración del control PID en el Asistente.

Con los valores obtenidos en la herramienta, se realiza el circuito impreso para validar los parámetros obtenidos del asistente, la Figura 4.38, muestra el circuito impreso del control PID con el impulsor.

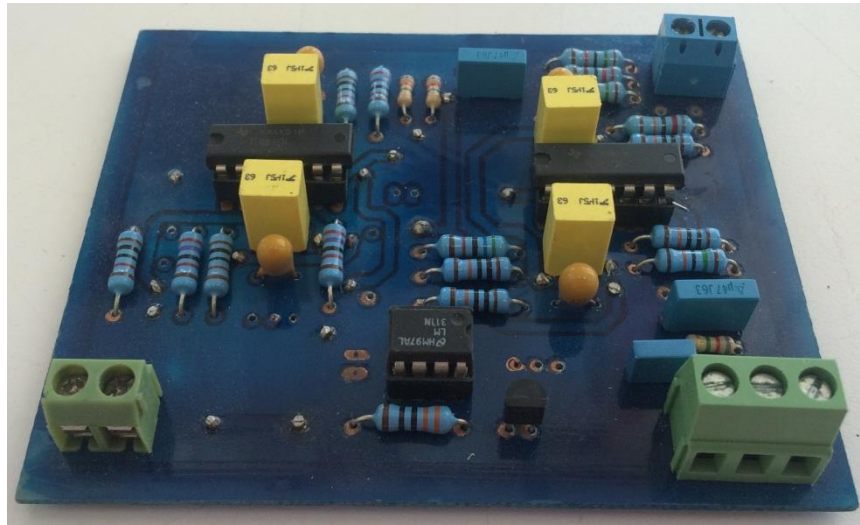
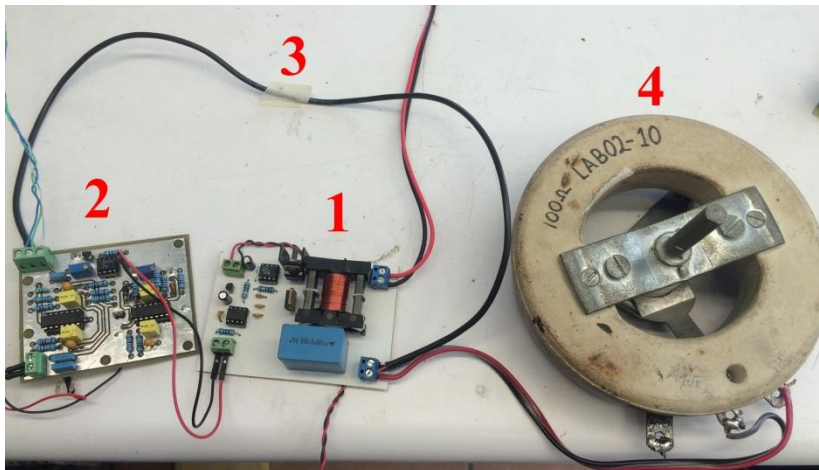


Figura 4.38. Placa del circuito de control PID e impulsor.

Posteriormente, se realizan las conexiones correspondientes para conformar el convertidor en lazo cerrado con el control PID, la Figura 4.39, muestra la configuración del convertidor implementado con un lazo cerrado de control tipo PID.



<b>1</b>	Convertidor Buck
<b>2</b>	Control PID e impulsor
<b>3</b>	Lazo cerrado de control
<b>4</b>	Carga

Figura 4.39. Convertidor Buck con lazo cerrado de control.

A continuación se presentan los resultados obtenidos en el osciloscopio de la respuesta transitoria del esquema de control PID ( Figura 4.40), así como la respuesta ante perturbaciones de incremento en la tensión de entrada ( Figura 4.41) y decremento en la misma (Figura 4.42), y por último el comportamiento de la tensión de salida ante variación en la carga del convertidor (Figura 4.43).

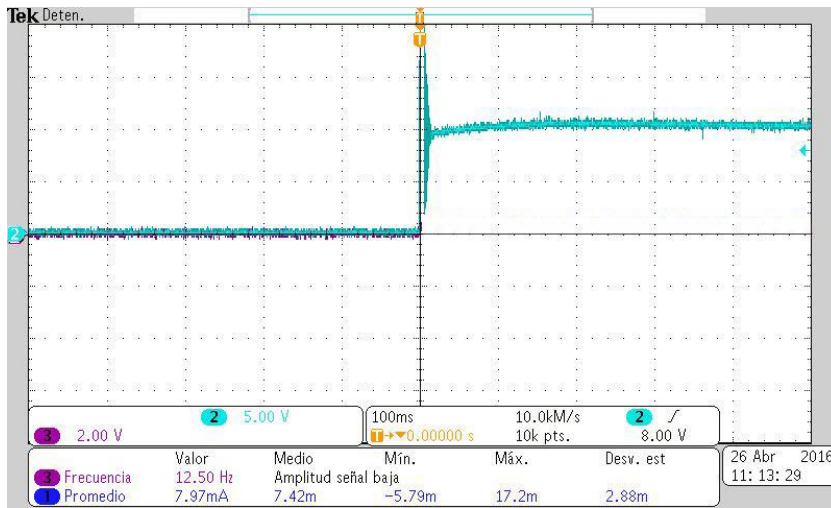


Figura 4.40. Respuesta del esquema de control PID (color cian).

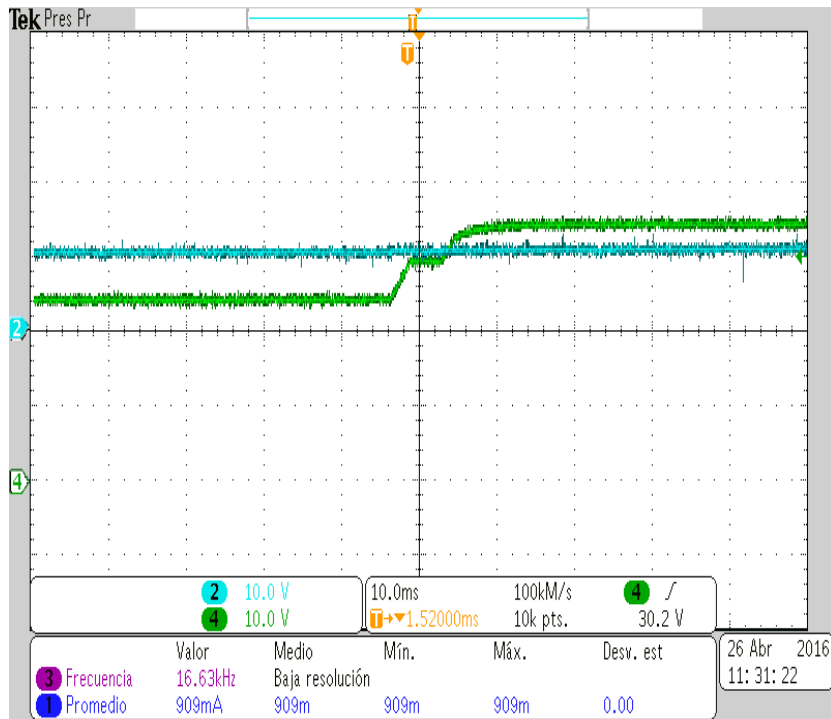


Figura 4.41. Comportamiento de la tensión de salida (color cian) ante incremento de la tensión de entrada (color verde).

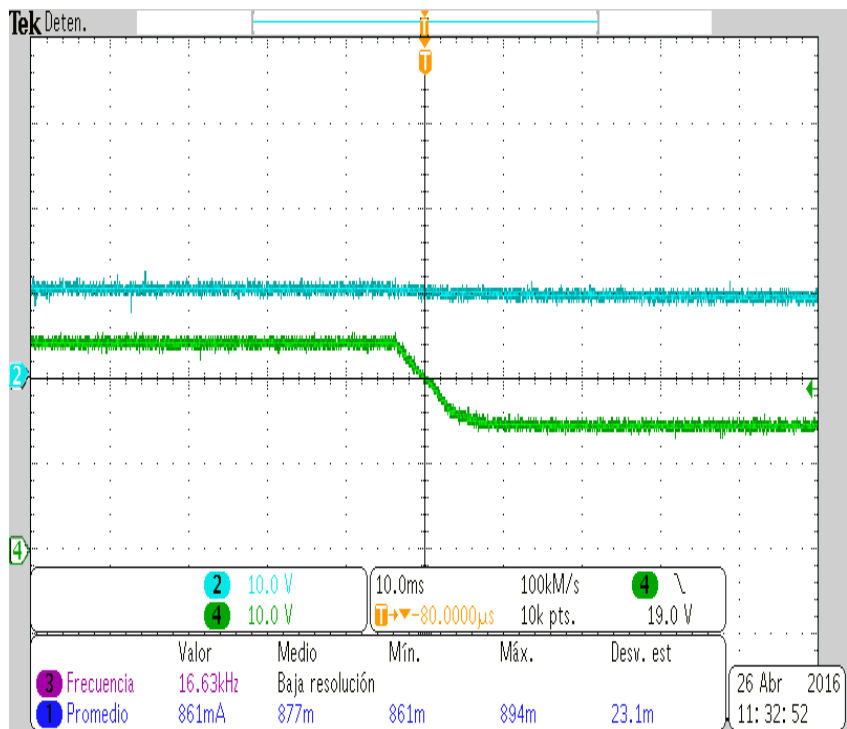


Figura 4.42. Comportamiento de la tensión de salida (color cian) ante decremento de la tensión de entrada (color verde).

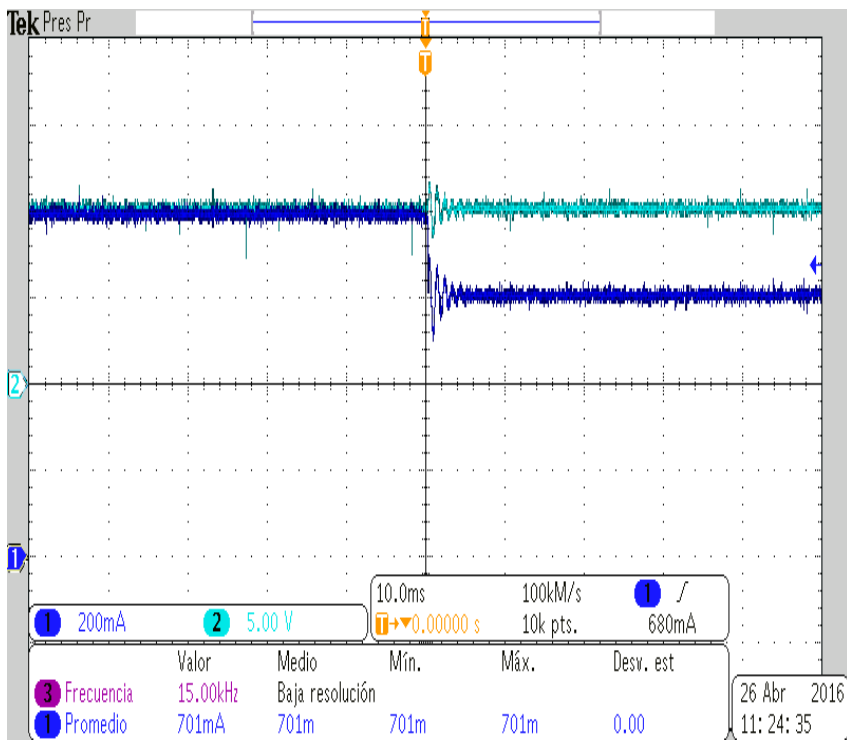


Figura 4.43. Comportamiento de la tensión de salida (color cian) ante decremento en la carga (color azul).

#### 4.14 Comparativa de resultados gráficos del Asistente y los obtenidos en laboratorio, en lazo abierto.

La Figura 4.44 muestra los pulsos de disparo de los dispositivos de conmutación, obtenidos en el osciloscopio (a) y los que presenta el Asistente.

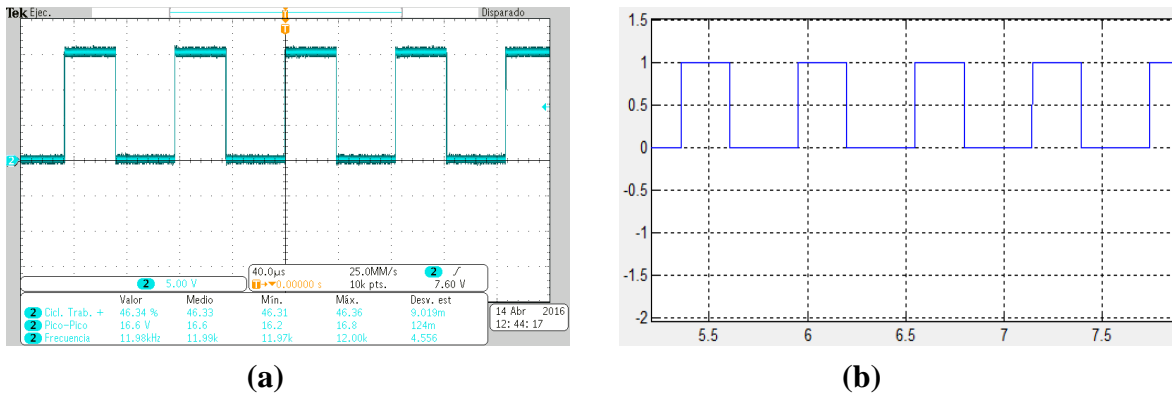


Figura 4.44. Comparación de pulsos de disparo del dispositivo de conmutación (a) y (b).

En la Figura 4.45 se presentan los gráficos de la corriente en el inductor obtenidos en el osciloscopio (a) y los del Asistente.

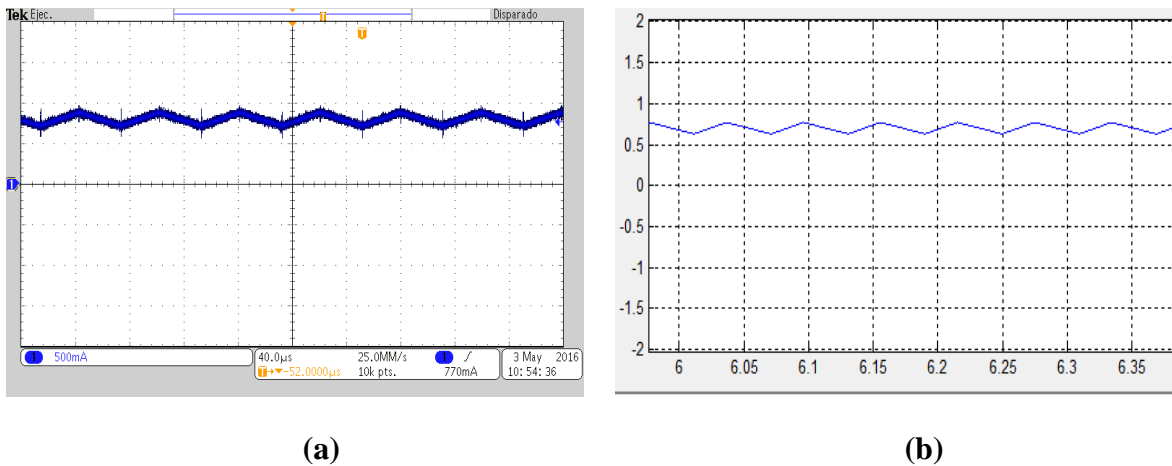
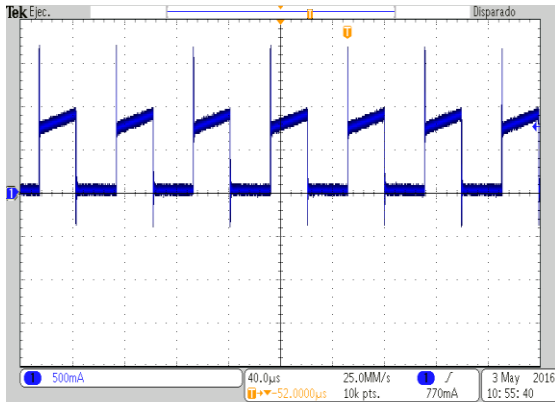
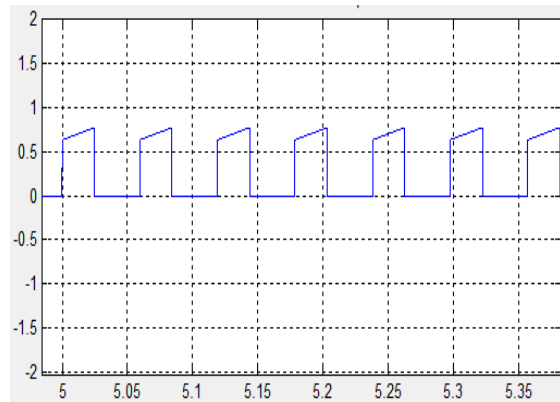


Figura 4.45. Comparación de la corriente en el inductor (a) y (b).

La Figura 4.46 despliega los gráficos de la corriente en el dispositivo de conmutación obtenidos en el laboratorio (a) y los que ofrece el asistente (b).



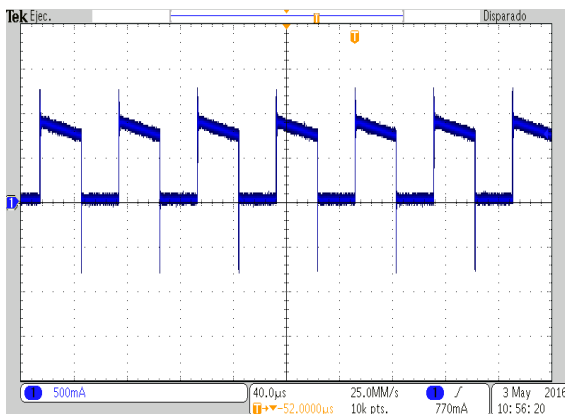
(a)



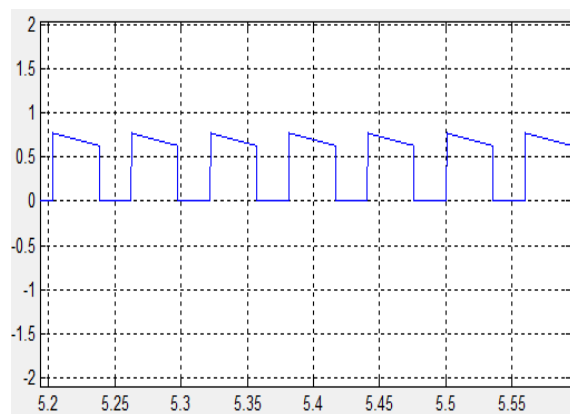
(b)

**Figura 4.46. Comparación de la corriente en el dispositivo de conmutación (a) y (b).**

También en la Figura 4.47 se presenta la corriente en el diodo que se obtiene a partir de las pruebas experimentales (a) y la correspondiente en la herramienta realizada.



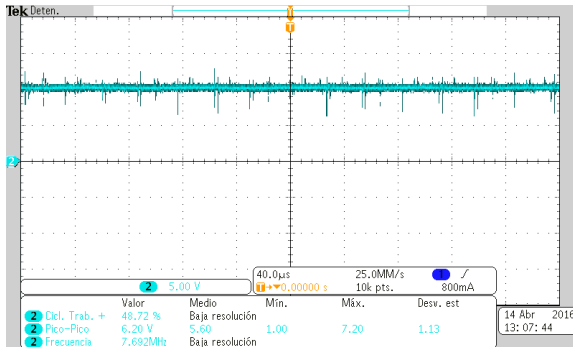
(a)



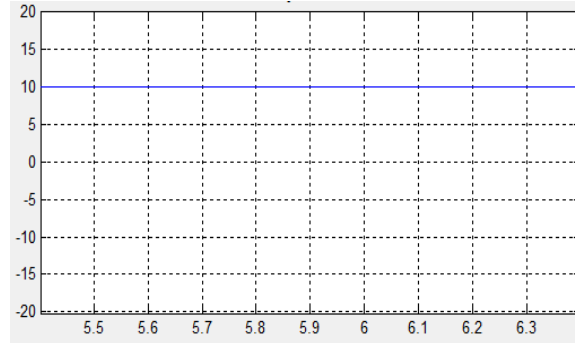
(b)

**Figura 4.47. Comparación de la corriente en el diodo (a) y (b).**

Por último, en la Figura 4.48 se observan los resultados de la tensión de salida del convertidor, obtenidos tanto en laboratorio (a) como en el asistente (b).



(a)



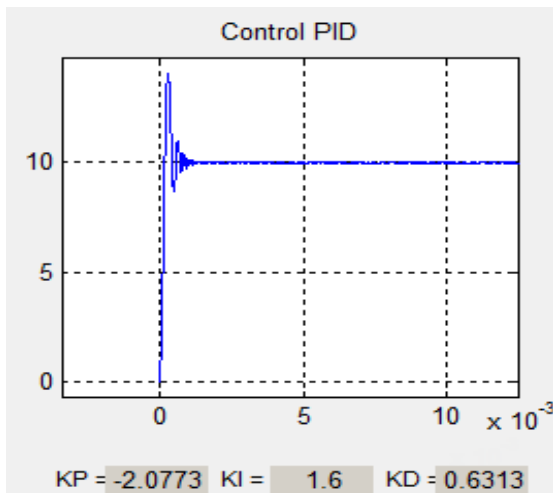
(b)

Figura 4.48. Comparación de la tensión de salida del convertidor (a) y (b).

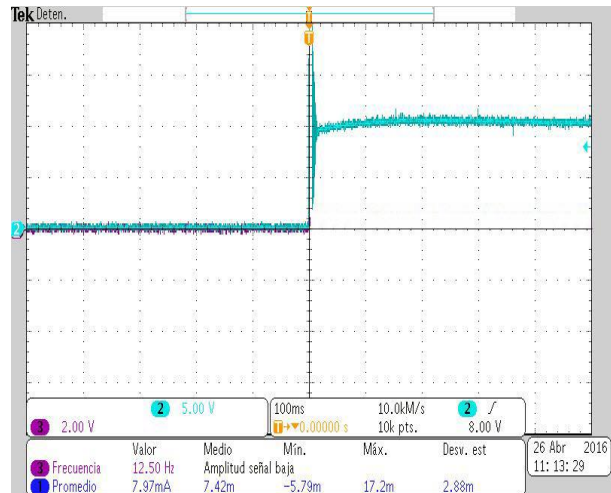
Como se puede observar los resultados gráficos que presenta el Asistente en lazo abierto, son consistentes con respecto a lo obtenido en laboratorio. Sin embargo, se procede a realizar la comparativa para el lazo cerrado del convertidor Buck.

#### 4.15 Comparativa de resultados gráficos del Asistente y los obtenidos en laboratorio, en lazo cerrado.

Posteriormente, se realizan las pruebas correspondientes del convertidor acoplado al lazo cerrado de control, así mismo se comparan los resultados ofrecidos por el asistente (a) y los obtenidos en el osciloscopio, ver Figura 4.49.



(a)



(b)

Figura 4.49. Comparación de respuesta transitoria del control PID (a) y (b).

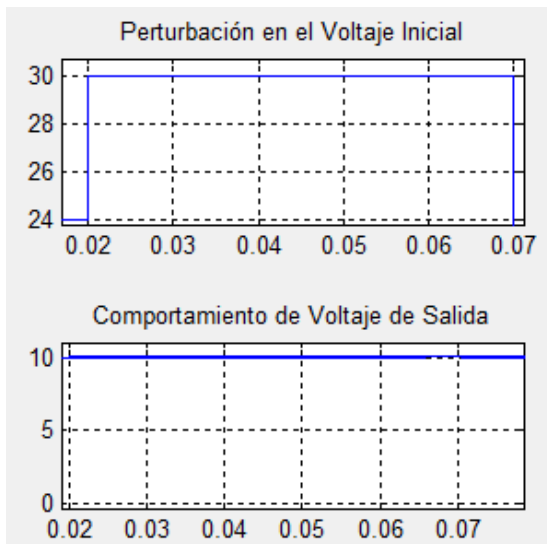
Cómo se puede apreciar de igual manera, los resultados gráficos de la implementación, son consistentes con respecto a los resultados mostrados en el asistente, en relación a la respuesta transitoria.

#### 4.16 Perturbaciones.

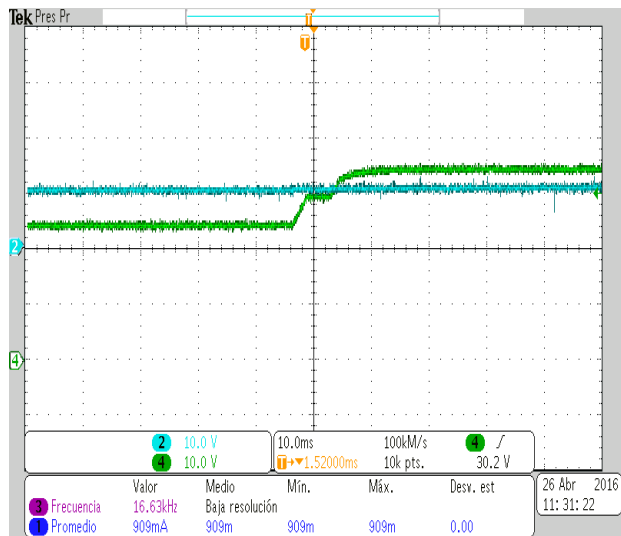
Con la finalidad de validar el funcionamiento del control, el asistente permite visualizar la respuesta del control ante perturbaciones en  $V_i$  y en la carga. De igual forma en la implementación, se realiza ésta validación para comparar los resultados ofrecidos por el asistente (a) y los resultados obtenidos en el laboratorio (b).

La Figura 4.50 muestra la respuesta del control que presenta el asistente (a) y la obtenida en laboratorio (b), ante un incremento del 20% en la tensión de entrada.





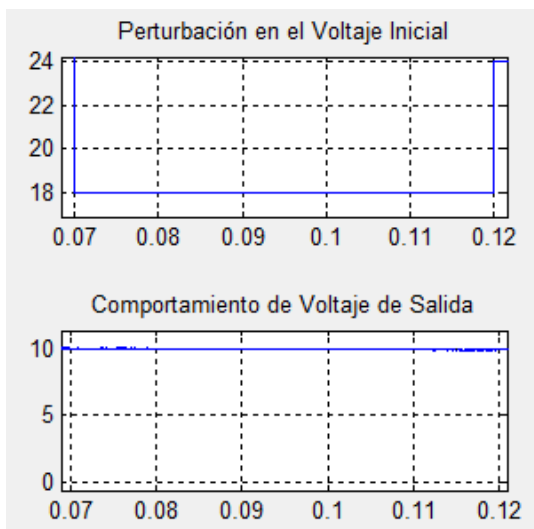
(a)



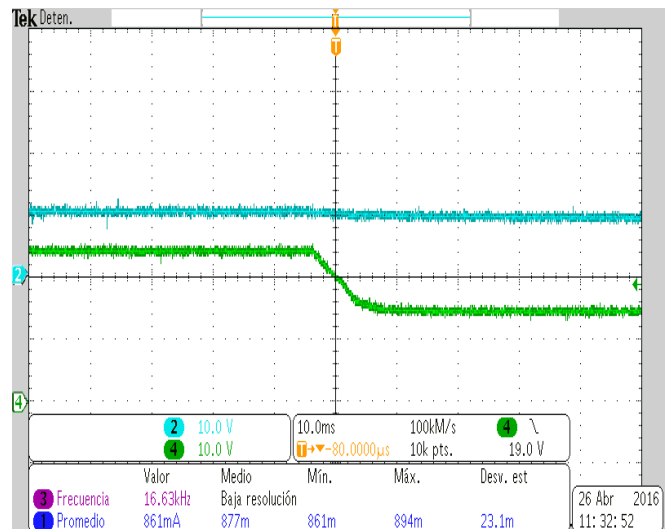
(b) Color cian tensión de salida y color verde tensión de entrada

Figura 4.50. Comparación del comportamiento de la tensión de salida ante incremento de la tensión de entrada (a) y (b).

Así mismo la Figura 4.51 muestra la respuesta del control que presenta el asistente (a) y la obtenida en laboratorio (b), ante un decremento del 20% en la tensión de entrada.



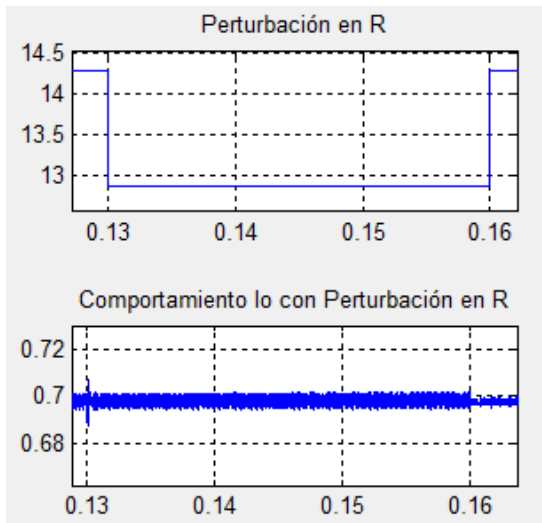
(a)



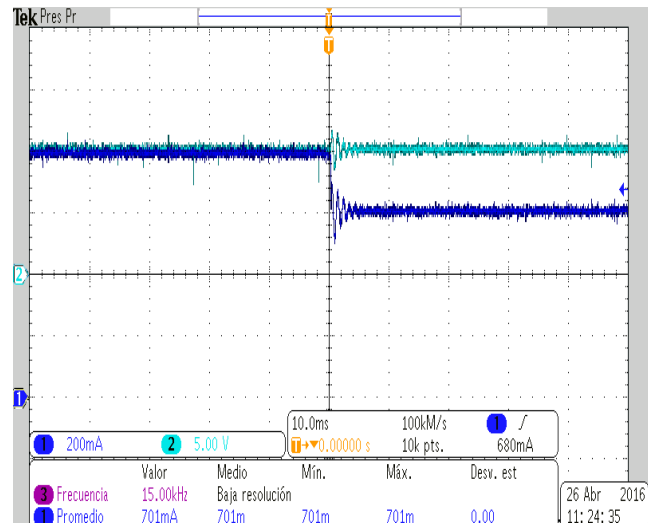
(b) Color cian tensión de salida y color verde tensión de entrada

Figura 4.51. Comparación del comportamiento de la tensión de salida ante decremento de la tensión de entrada (a) y (b).

Otro tipo de perturbación se presenta en la Figura 4.52, el cual se aplica un decremento en la carga, y de este modo, validar el comportamiento del control PID, y se presentan los resultados que ofrece el asistente (a) y los obtenidos en el osciloscopio (b).



(a)



(b) Color cian tensión de salida y color azul carga del convertidor

Figura 4.52. Comparación del comportamiento de la tensión de salida ante decremento en la carga (a) y (b).

#### 4.17 Análisis concordancia del Asistente.

Con los resultados gráficos obtenidos con anterioridad, se realiza un análisis de la concordancia del asistente, el cual consiste en obtener un margen de error entre los resultados que muestra el asistente y los datos obtenidos de la implementación, con el objetivo de establecer qué tan cercano o distante resulta la utilización de la herramienta.

Para realizar dicho análisis, se obtienen los datos de los valores del convertidor en lazo abierto ofrecidos por el asistente en un periodo de tiempo, y se extraen los valores directos del osciloscopio. Los valores se adecuan, organizan y se sintonizan para los mismos instantes de tiempo. Al evaluar las mediciones obtenidas, se realiza el cálculo del porcentaje de error, para estimar un porcentaje promedio de las muestras obtenidas en el laboratorio.

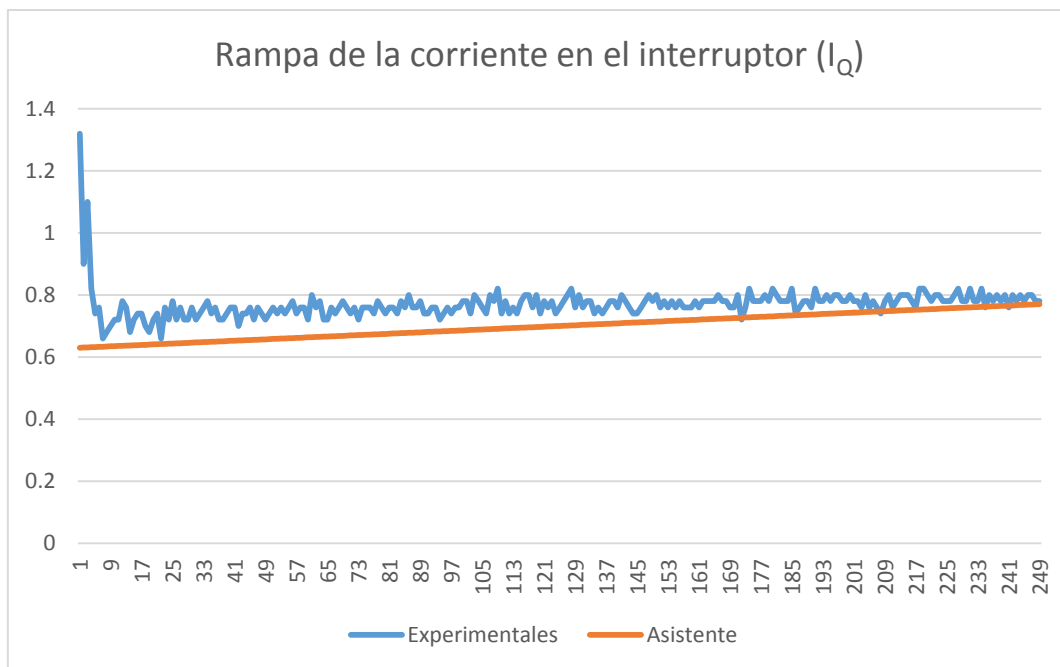
Para obtener el porcentaje de error se aplicó la siguiente fórmula.

$$\frac{|\text{Valor Laboratorio} - \text{Valor Asistente}|}{|\text{Valor Asistente}|} \times 100\%$$

Se promedian los valores obtenidos en las pruebas y los obtenidos en el asistente, consiguiendo así un promedio de los valores obtenidos en laboratorio y un promedio de los valores brindados por el Asistente, para posteriormente aplicar la fórmula de porcentaje de error y tener un porcentaje de error aproximado entre ambos resultados.

El análisis de los resultados de la concordancia para la corriente en el dispositivo de conmutación se muestra en la Figura 4.53.

Con los valores anteriores y aplicando la fórmula anteriormente mencionada se obtiene el porcentaje de error del asistente de  $I_Q$ , ver Tabla 4.2.

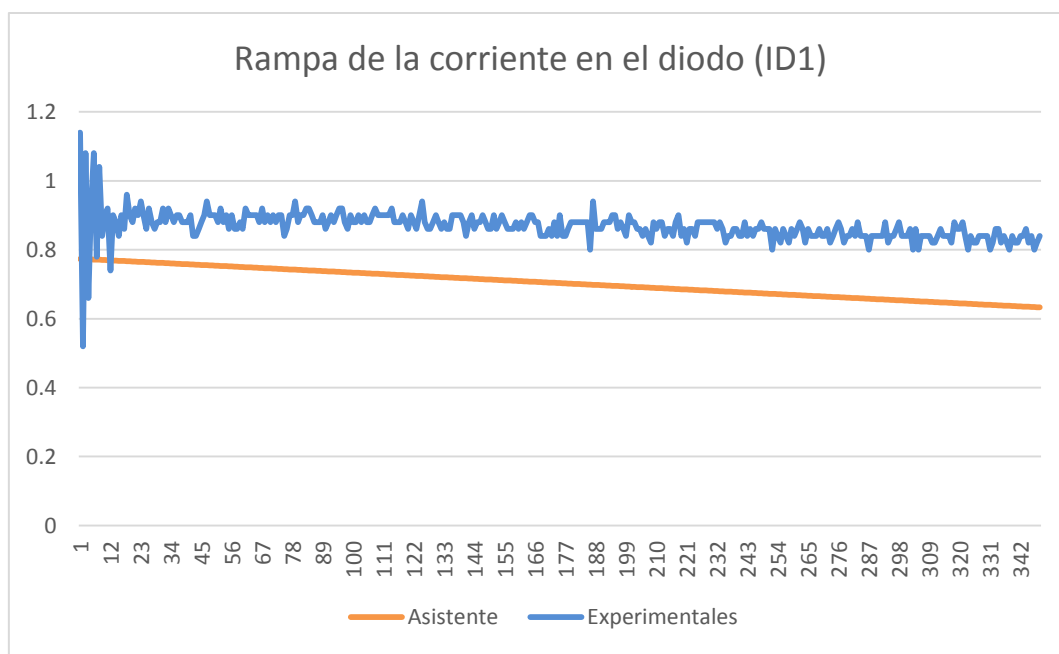


**Figura 4.53. Comparación de magnitud de corriente en el dispositivo de conmutación.**

**Tabla 4.2. Porcentaje de error promedio en la corriente del dispositivo de conmutación.**

Promedio Osciloscopio	Promedio Asistente
<b>0.769959839</b>	0.700004756
<b>Porcentaje de Error Promedio</b>	10%

De igual manera se realiza el análisis para la corriente que circula por el diodo, tal como se muestra en la Figura 4.54.



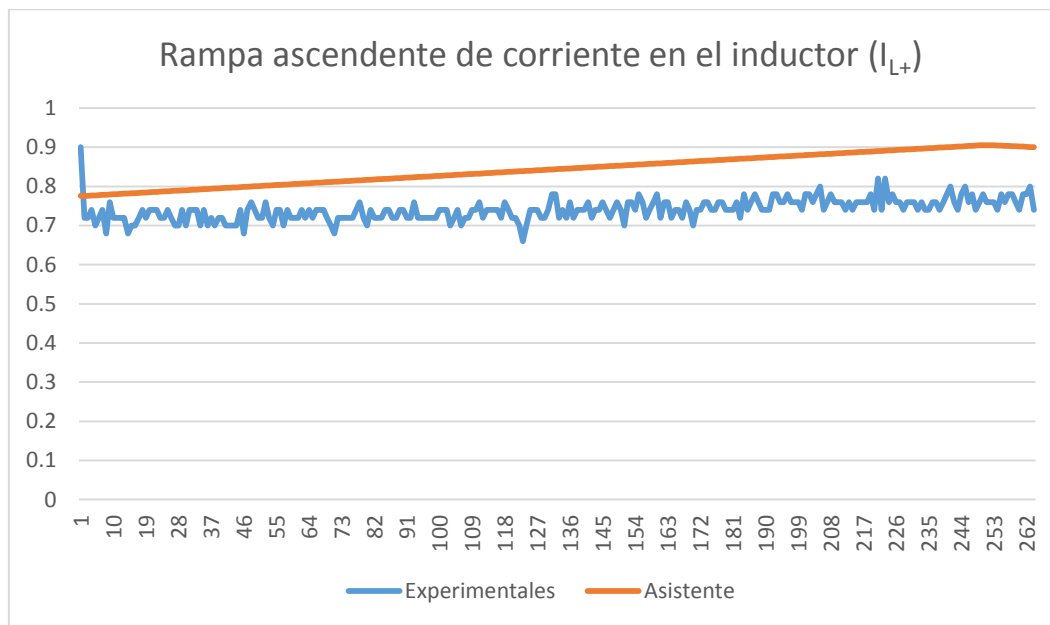
**Figura 4.54. Comparación de magnitud de corriente en el dispositivo de conmutación.**

El porcentaje de error de la corriente  $I_D$  se presenta en la Tabla 4.3.

**Tabla 4.3. Porcentaje de error en la corriente del diodo.**

Promedio Osciloscopio	Promedio Asistente
<b>0.725578065</b>	0.828740213
<b>Error relativo porcentual</b>	12%

En la Figura 4.55, se presenta el análisis de los datos obtenidos para la rampa de corriente ascendente en el inductor.



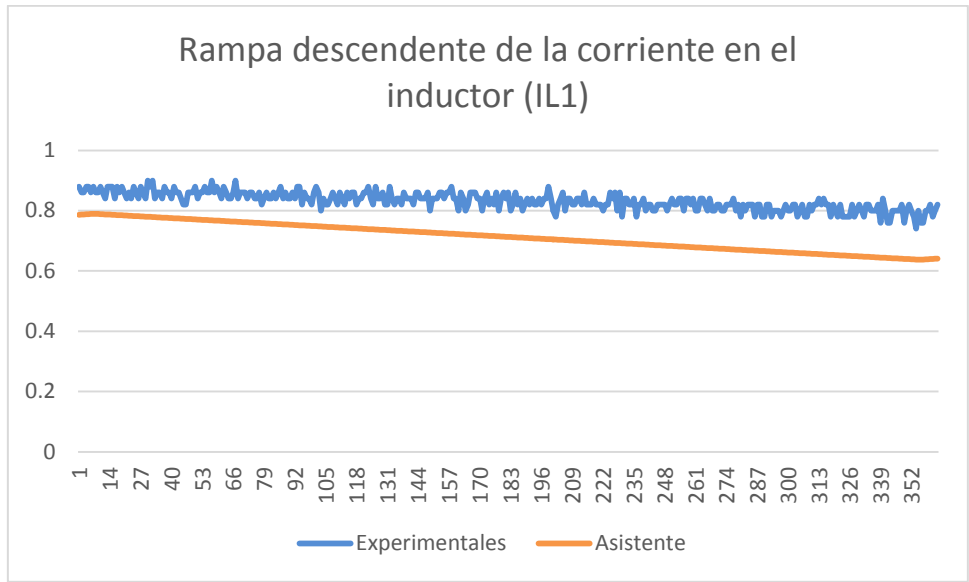
**Figura 4.55. Comparación de magnitud de corriente ascendente en el inductor.**

El porcentaje de error de la  $I_{L+}$  se presenta en la Tabla 4.4.

**Tabla 4.4. Porcentaje de error en la corriente ascendente del inductor.**

Promedio Osciloscopio	Promedio Asistente
<b>0.740757576</b>	0.843721584
<b>Error relativo porcentual</b>	12%

También se realiza el análisis para la rampa descendente de corriente en el inductor, ver Figura 4.56.



**Figura 4.56. Comparación de magnitud de corriente descendente en el inductor.**

La Tabla 4.5 muestra los resultados de la  $I_L$  en el inductor.

**Tabla 4.5. Porcentaje de error en la corriente ascendente del inductor.**

<b>Promedio Osciloscopio</b>	<b>Promedio Asistente</b>
<b>0.832011019</b>	0.742214926
<b>Error relativo porcentual</b>	12%

Para finalizar, se realiza el proceso para los datos obtenidos para el control PID, se adecúan para realizar el análisis correspondiente al porcentaje de error existente entre los datos obtenidos en laboratorio y los presentados por el asistente.

La Figura 4.57 muestra el análisis para determinar el porcentaje de error entre los datos que ofrece el asistente y los datos obtenidos en la implementación física.

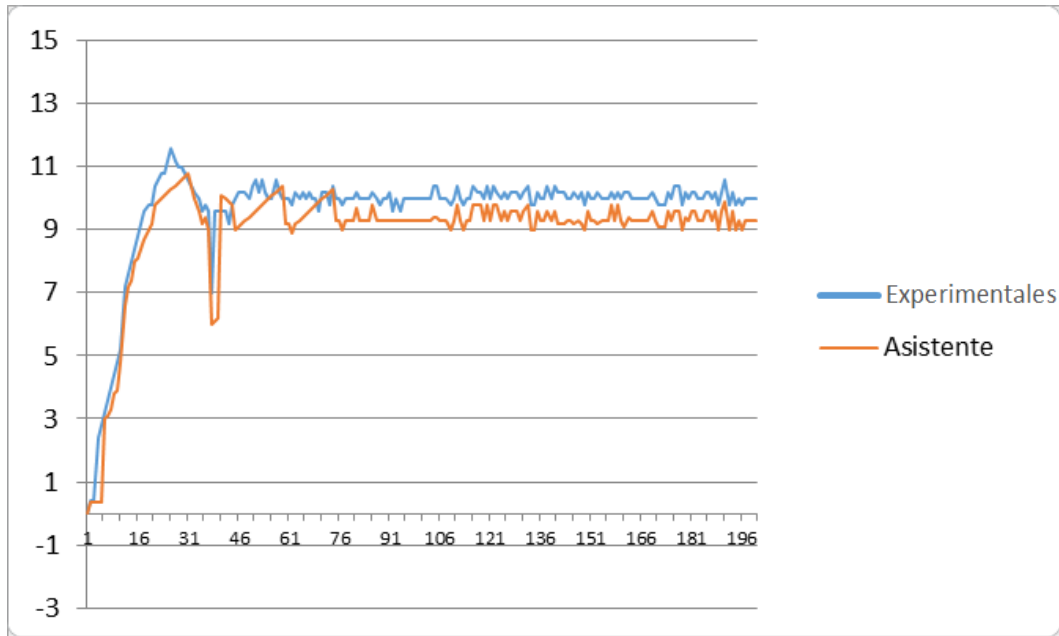


Figura 4.57. Comparación del control PID.

Con la comparación de datos anteriores, la Tabla 4.6 presenta los resultados del porcentaje de error del control PID.

Tabla 4.6. Porcentaje de error en la corriente ascendente del inductor.

Promedio Osciloscopio	Promedio Asistente
10.06043165	9.324523
<b>Error relativo porcentual</b>	7.3%

Con la comparación de los datos presentados por el análisis de utilidad del asistente, tanto en lazo abierto como en lazo cerrado de control del convertidor, se observa la congruencia de los resultados mostrados entre la herramienta y la implementación. Además se comprueba el pequeño margen de error que se observa en el laboratorio, lo que permite visualizar la confiabilidad en el apoyo de la herramienta para el usuario que analiza e implementa convertidores CD-CD.

# CAPÍTULO V. CONCLUSIONES Y TRABAJOS FUTUROS



## 5.1 Conclusiones.

El asistente de convertidores CD-CD, es una herramienta que permite al usuario apoyar en el proceso de análisis, diseño y comportamiento de operación en lazo abierto, así como apoyar en el modelado en lazo cerrado de control de voltaje del convertidor Buck. La herramienta surge de la necesidad de optimizar los tiempos para el análisis, diseño e implementación del convertidor Buck, establecido por el usuario, con la ventaja de rediseñar las veces que sea necesario hasta cumplir con el comportamiento esperado del convertidor. Existen actualmente herramientas que han sido diseñadas para apoyar tanto en la formación académica, así como para apoyar en la visualización del comportamiento solo en lazo abierto del convertidor, sin embargo la gran diferencia con el asistente, radica en el apoyo no solo en lazo abierto sino también en el esquema de control tipo PID, y validar el comportamiento del control ante perturbaciones.

Las conclusiones de la tesis se establecen a continuación:

1. El asistente es una herramienta de primera mano en la investigación, que tiene el enfoque para un usuario con perfil de conocimientos básicos en el ámbito de convertidores CD-CD, en los datos básicos de diseño y en la interpretación de resultados.
2. El Asistente no es tan sólo una herramienta indispensable, para introducir de forma didáctica al estudiante e investigador en el mundo de los Convertidores CD-CD, sino como un paso intermedio entre el trabajo analítico y la implementación física del circuito
3. A partir del análisis, validación y diseño tanto del modelado de convertidores como de las topologías y de las técnicas de modulación PWM, permite establecer la estructura final del asistente así como la presentación de los resultados que muestra la herramienta. También permite conocer las diferentes opciones de modelado, diseño, técnicas de modulación y controladores PID, para conocer a detalle el comportamiento de los convertidores.

4. Se cuenta con la estructura base del Asistente, de manera modular, tanto en programación como en su proceso interno para realizar las diferentes rutinas, con la finalidad de agregar más topologías y/o seguir complementando la herramienta con más funcionalidades.
5. La versión del asistente se realiza con programación dinámica en código de Matlab, lo que permite recalcular, cuántas veces requiera el usuario, los resultados que presenta la herramienta, hasta cumplir con el comportamiento deseado.
6. Se presenta una interfaz a través de la cual permite al usuario, de una manera cómoda y sencilla, configurar los diferentes parámetros que describen tanto el propio convertidor como el lazo de control, con el fin de poder así comprobar su comportamiento y estabilidad con un control PID, el cual es una función limitada para las herramientas actuales.
7. Los resultados de la implementación, tanto en lazo abierto como en lazo cerrado, muestran una consistencia gráfica con respecto a los resultados mostrados por el asistente, lo que permite establecer el gran apoyo que tiene la herramienta para el diseñador de convertidores CD-CD.
8. Para comprobar la consistencia gráfica de los resultados se realiza un análisis a detalle de los datos obtenidos por el asistente, con la finalidad de establecer la precisión de los resultados mostrados, mismos que permiten visualizarla existencia de un mínimo margen de diferencia en los resultados que ofrece el asistente, por lo que comprueba la utilidad del asistente como opción de primera mano para analizar el comportamiento de convertidores CD-CD, tanto en lazo abierto como con un esquema de control tipo PID.
9. Se debe considerar que los valores del inductor y el capacitor que muestra el asistente, no siempre son valores comerciales, razón por la cual a veces es necesario volver a calcular los valores, son embargo no es un impedimento para validar el comportamiento del convertidor.

10. En la realización de esta tesis han tenido en cuenta todas las necesidades requeridas por el usuario, por lo que se ha programado y diseñado al asistente, de tal forma que al acceder la pantalla de configuración del convertidor y llevar acabo todas las tareas que puede realizar el asistente sea de manera intuitiva y amigable.

## **5.2 Trabajos Futuros.**

Este trabajo de tesis ofrece una herramienta opcional para el análisis y diseño de convertidores CD-CD, está se puede visualizar tan amplia para ofrecer más resultados y funcionalidades, como se desee. La propuesta realizada muestra resultados confiables, los cuales pueden mejorar, visualizando los siguientes trabajos futuros:

- Ampliar la herramienta con más convertidores de potencia.
- Contar con el modo discontinuo para todos los convertidores que ofrezca el asistente.
- Complementar con más esquemas de control acorde a las necesidades del diseñador de convertidores CD-CD.
- Optimizar el cálculo de los elementos que integran el asistente, con una base de datos que contenga valores comerciales, con la finalidad de ofrecer al usuario el elemento indicado para implementar el convertidor.
- Contar con una funcionalidad en el asistente que permita presentar una propuesta del circuito impreso del convertidor a implementar.

## Bibliografía.

- [1] S. Ang y A. Oliva, *Power Switching Converter*, Estados Unidos: CRC Press Taylor & Francis group, 2005.
- [2] R. Erickson y D. Maksimovic, *Fundamental of Power Electronics*, Estados Unidos: Springer, 2001.
- [3] N. Mohan, T. M. Undeland y W. P. Robbins, *Power Electronics: Converters, Applications and Design*, Estados Unidos: Media Enhanced, 2003.
- [4] M. . H. Rashid, *Electrónica de Potencia: Circuitos, Dispositivos y Aplicaciones*, Estados Unidos: Prentice Hall, 1995.
- [5] G. C. Chryssis, *High-Frequency Switching Power Supplies: Theory and Design*, Estados Unidos: McGraw Hill, 1989.
- [6] M. Brown, *Power Supply Cookbook*, Estados Unidos: Newnes, 1994.
- [7] B. W. Williams, *Power Electronics Devices, Drivers, Applications and Passive Components*, Estados Unidos: Macmillan, 1992.
- [8] H. Piquet, T. Meynard, H. Foch y Y. Chéron , «Design tools and methods for Power Electronics,» de *2 rue Camichel31071 Toulouse CEDEX France 0-7803-2091-3/9/*, Toulouse, France, 1994.
- [9] J. M. Rivas, D. Jackson, . O. Leitermann, A. D. Sagneri, Y. Han y D. J. Perreault, «Power Electronics Specialists Conference, 2006. PESC '06. 37th IEEE,» de *Design Considerations for Very High Frequency dc-dc Converters*, Jeju, Corea del sur, 2006.
- [10] S. Chander , P. Agarwal y I. Gupta , «Design, Modeling and Simulation of DC-DC,» de *IPEC, 2010 Conference Proceedings* , Singapore, 2010.
- [11] M. Karadeniz, M. O. Gulbahce y D. A. Kocabas, «MECHATRONIKA, 2012 15th International Symposium,» de *A practical design and analysis of a high power DC to DC converter*, Prague, República Checa, 2012.
- [12] K. Bhattacharyya, «Trend towards the design of embedded DC–DC converters,» *IET Power Electronics*, vol. 6, nº 8, p. 1563–1574, 2012.
- [13] J. HUANG, P. ZHAO, C. GAO y S. DIA, «Key Components Determination Method for DC-DC,» de *Prognostics and System Health Management Conference (PHM-2014 Hunan)*, Zhangjiiaijie City, China, 2014.

- [14] Y. Wu , Y. Wang, Y. Jiang y Q. Sun, «Prognostics and Health Management (PHM), 2014 IEEE Conference on,» de *A Prognostic method for DC-DC converters under variable operating conditions*, Cheney, WA, 2014.
- [15] C. A. TORRES P., C. RESTREPO P. y A. ALZATE G., «CONSIDERACIONES DE DISEÑO ESTÁTICO Y DINÁMICO PARA CONVERTIDORES,» *Scientia et Technica Año XV*, nº 42, pp. 57-62, 2009.
- [16] A. Taut, O. Pop, I. Baciú y M. Daraban, «A Matlab Tool for Determining the Parameters of Power DC–DC Non Isolated Converters,» de *36th Int. Spring Seminar on Electronics Technology 978-1-4799-0036-7/13*, Cluj, Romania, 2013.
- [17] N. D. Muhamad, A. . H. Mohd Yatim y A. J. Shafie, «A Pspice-based Design of DC-DC Converter Systems,» de *National Power & Energy Conference (PECon) 2004 Proceedings 0-7803-8724-4/0*, Kuala Lumpur, Malaysia , 2004.
- [18] C. Tarasantisuk, S. Tunyasrirut y V. Tipsuwanpom, «A Matlab/Simulink Tool for Enhancing Efficient Education of Power Electronics Corresponding to the ETH Power Converter Laboratory,» de *The 30th Annual Conference of the IEEE Industrial Electronics Society*, Busan, Korea , 2004.
- [19] Ş. DEMİRBAŞ y G. MUTLU, «Internet Based Toolkit For Simulation Of DC/DC Converters,» de *Proceedings of the 2011 International Conference on Power Engineering, Energy and Electrical Drives*, Torremolinos (Málaga), Spain. May 2011, 2011.
- [20] F. A. S. Gonçalves, L. . P. Sampaio y C. A. Canesin, «INTERACTIVE DC-DC CONVERTERS TOOLS FOR INSTANT DESIGN AND EDUCATION,» de *INTERACTIVE DC-DC CONVERTERS TOOLS FOR INSTANT DESIGN AND EDUCATION*, Ilha Solteira - SP – Brazil, 2009.
- [21] W. Shun-Chung and L. Yi-Hwa, "Software-Reconfigurable e-Learning Platform for Power Electronics Courses," *In IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2416-2424, June 2008.

# ANEXOS

## Anexo A. Código de programación del Asistente.

### Inicio.

```
function varargout = Asistente2(varargin)
% ASISTENTE2 MATLAB code for Asistente2.fig
%   ASISTENTE2, by itself, creates a new
ASISTENTE2 or raises the existing
%   singleton*.
%
%   H = ASISTENTE2 returns the handle to
a new ASISTENTE2 or the handle to
%   the existing singleton*.
%
%
ASISTENTE2('CALLBACK',hObject,eventData,han
dles,...) calls the local
%   function named CALLBACK in
ASISTENTE2.M with the given input
arguments.
%
%   ASISTENTE2('Property','Value',...)
creates a new ASISTENTE2 or raises the
%   existing singleton*. Starting from
the left, property value pairs are
%   applied to the GUI before
Asistente2_OpeningFcn gets called. An
%   unrecognized property name or
invalid value makes property application
%   stop. All inputs are passed to
Asistente2_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools
menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the
response to help Asistente2

% Last Modified by GUIDE v2.5 04-Feb-2016
19:42:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
    'gui_Singleton',
gui_Singleton, ...
    'gui_OpeningFcn',
@Asistente2_OpeningFcn, ...
    'gui_OutputFcn',
@Asistente2_OutputFcn, ...
    'gui_LayoutFcn', [] ,
...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before Asistente2 is
made visible.
function Asistente2_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see
OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)
% varargin   command line arguments to
Asistente2 (see VARARGIN)

% Choose default command line output for
Asistente2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- LOGO PARA LA CABECERA DE LA PÁGINA
PRINCIPAL DEL ASISTENTE
axes(handles.Cabecera)
handles.imagen=imread('LOGO ESCUELA.png');
imagesc(handles.imagen)
axis off

% --- LOGO PARA LA IMAGEN ILUSTRATIVA DE LA
PÁGINA PRINCIPAL DEL ASISTENTE
axes(handles.imgIlustrativa)
handles.imagen=imread('Logo_DIE.jpg');
imagesc(handles.imagen)
axis off

% UIWAIT makes Asistente2 wait for user
response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are
returned to the command line.
function varargout =
Asistente2_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning
output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)

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

% --- Executes on button press in Generar.
```

```

function Generar_Callback(hObject,
eventdata, handles)
%Validación Tipo y Modo Conductor
TCStr = get(handles.MenuTipoConvertidor,
'String'); %Cadena selección tipo
convertidor
TC = get(handles.MenuTipoConvertidor,
'Value'); %Valor selección tipo convertidor
% MCStr = get(handles.MenuModoConvertidor,
'String'); %Cadena selección modo
convertidor
% MC = get(handles.MenuModoConvertidor,
'Value'); %Valor selección modo convertidor

    switch TCStr{TC};
    case 'Buck'
        run('Buck.m');
    case 'Boost'
    case 'Buck-Boost'
    case 'Cuk'
    otherwise
        beep;
        errordlg('Seleccione un Tipo de
Convertidor, intente nuevamente', 'Selección
Errónea', 'modal')
        return;
    end

% switch MCStr{MC};
%   case 'MCC'
%       switch TCStr{TC};
%       case 'Buck'
%           run('Buck.m');
%       case 'Boost'
%       case 'Buck-Boost'
%       case 'Cuk'
%       otherwise
%           beep;
%           errordlg('Seleccione un
Tipo de Convertidor, intente
nuevamente', 'Selección Errónea', 'modal')
%           return;
%       end
%   case 'MCD'
%       switch TCStr{TC};
%       case 'Buck'
%           run('BuckMCD.m');
%       case 'Boost'
%       case 'Buck-Boost'
%       case 'Cuk'
%       otherwise
%           beep;
%           errordlg('Seleccione un
Tipo de Convertidor, intente
nuevamente', 'Selección Errónea', 'modal')
%           return;
%       end
%   otherwise
%       beep;
%       errordlg('Seleccione un Modo de
Conducción, intente nuevamente', 'Selección
Errónea', 'modal')
%       return;
%   end

% --- Executes on selection change in
MenuModoConvertidor.

```

```

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

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

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

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

% --- Executes on selection change in
MenuTipoConvertidor.
function
MenuTipoConvertidor_Callback(hObject,
eventdata, handles)
% hObject    handle to MenuTipoConvertidor
(see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)

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

% --- Executes during object creation,
after setting all properties.
function
MenuTipoConvertidor_CreateFcn(hObject,
eventdata, handles)
% hObject    handle to MenuTipoConvertidor
(see GCBO)

```



```

% eventdata reserved - to be defined in a
future version of MATLAB
% handles empty - handles not created
until after all CreateFcns called

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

% -----
function Untitled_1_Callback(hObject,
eventdata, handles)
% hObject handle to Untitled_1 (see
GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and
user data (see GUIDATA)

% --- Executes during object creation,
after setting all properties.
function Cabecera_CreateFcn(hObject,
eventdata, handles)
% hObject handle to Cabecera (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles empty - handles not created
until after all CreateFcns called
% Hint: place code in OpeningFcn to
populate Cabecera

```

## Lazo Abierto.

```

%---CENIDET
%---TESIS "Asistente para el Análisis y
Diseño de Topologías Comunes de
Convertidores CD-CD"
%---AUTOR: EIDER MIGUEL AMORES CAMPOS
%---DESCRIPCIÓN: ARCHIVO DE CONVERTIDOR
"BUCK"
%----- En este apartado se hace el
cálculo de las variables con
% base en los datos
introducidos por el usuario. Derivado de
esto
% se muestran las gráficas
obtenidas y los resultados. Además se
% incluye la función para
limpiar datos y hacer nuevos cálculos,
% sin necesidad de volver a
abrir la aplicación.....
%---FECHA: ENERO 2016

function varargout = Buck(varargin)
% BUCK MATLAB code for Buck.fig
% BUCK, by itself, creates a new BUCK
or raises the existing
% singleton*.

```

```

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

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

% Last Modified by GUIDE v2.5 05-Apr-2016
22:33:48

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

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

% --- Executes just before Buck is made
visible.
function Buck_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see
OutputFcn.
% hObject handle to figure

```

```

% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and
user data (see GUIDATA)
% varargin command line arguments to Buck
(see VARARGIN)

```

```

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

```

```

% Update handles structure
guidata(hObject, handles);

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

% --- VALIDADOR DE LOS DATOS INTRODUCIDOS
POR EL USUARIO EN LOS PARÁMETROS
% --- Y CÁLCULO DE RESULTADOS
function Calcular_Callback(hObject,
eventdata, handles)
%Parámetros que el usuario teclea y
asignación de variables
X = get(handles.Vo,'string');
Vo = str2double(X);
Y = get(handles.Vi,'string');
Vi = str2double(Y);
Z = get(handles.PoMax,'string');
PoMax = str2double(Z);
A = get(handles.DeltaIL,'string');
DeltaIL = str2double(A)/100;
B = get(handles.FS,'string');
FS = str2double(B);
W = get(handles.DeltaVo,'string');
DeltaVo = str2double(W)/100;

set(handles.Vo,'string',X);
%Validación de datos
if (isnan(Vo))
    beep;
    errordlg('Falta asignar valor en
voltaje de salida, favor de
verificar.','Sin Valor.','modal')
    return;
else
    if (isnan(Vi))
        beep;
        errordlg('Falta asignar valor en
voltaje de entrada, favor de
verificar.','Sin Valor.','modal')
        return;
    else
        if (isnan(PoMax))
            beep;
            errordlg('Falta asignar valor
en potencia de salida, favor de
verificar.','Sin Valor.','modal')
            return;
        else
            if (isnan(DeltaIL))
                beep;
                errordlg('Falta asignar
valor en rizado de corriente (DeltaIL),
favor de verificar.','Sin Valor.','modal')
                return;
            else
                if (isnan(FS))
                    beep;
                    errordlg('Falta asignar
valor en frecuencia de operación, favor de
verificar.','Sin Valor.','modal')
                    return;
                else
                    if (isnan(DeltaVo))
                        beep;
                        errordlg('Falta
asignar valor en DeltaVo, favor de
verificar.','Sin Valor.','modal')
                        return;
                    else
                        %Obtención de la
Ganancia
                        D=Vo/Vi;
                        %Obtención de
Corriente para obtener R
                        I=PoMax/Vo;
                        %Obtención del
Inductor
                        L=((Vi-
Vo)*D)/((DeltaIL*I)*FS);
                        %Obtención de la
Capacitor
                        C=((Vi-
Vo)*D)/(8*DeltaVo*L*(FS^2));
                        %Obtención de la
Resistencia para diferentes cálculos
                        R=Vo/I;
                        %Obtener IQ
                        IQ=(Vo/R)*D;
                        %Obtener Corriente
Pico en el Interruptor
                        IPK=I+((Vi-
Vo)*D)/(2*L*FS));
                        %Obtener Voltaje
Máximo en el Interruptor
                        VDS=Vi;
                        %obtener corriente
promedio en el diodo
                        ID=I*(1-D);
                        %Obtener corriente
pico en el diodo
                        IDPK=I+((Vo*(1-
D))/(2*L*FS));
                        %Obtener voltaje
máximo en el diodo
                        VAK=Vi;

                        %Validar que el
Rizado de corriente sea menor a la
Corriente
                        if (Vo>=Vi)
                            beep;
                            errordlg('El
Voltaje de Entrada debe ser MAYOR al
Voltaje de Salida','Valor Erróneo','modal')
                            return;

```

```

        else
            if (DeltaIL >=
I)
                beep;
                errordlg('El
Rizado de Corriente Delta IL debe ser MENOR
a la Corriente','Valor Erróneo','modal')
                return;
            else

set(handles.Ganancia, 'string',
num2str(D));

set(handles.Inductor, 'string',
num2str(L));

set(handles.Capacitor, 'string',
num2str(C));

set(handles.IQ_Prom, 'string',
num2str(IQ));

set(handles.IPK_Q, 'string', num2str(IPK));

set(handles.VMaxInt, 'string',
num2str(VDS));

set(handles.ID, 'string', num2str(ID));

set(handles.IDPK, 'string', num2str(IDPK));

set(handles.VDMax, 'string', num2str(VAK));

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

```

```

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

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

```

```

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

```

```

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

```

```

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

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

% Hint: edit controls usually have a white
background on Windows.

```

```

%     See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

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

```

```

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

```

```

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

```

```

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

```

```

% Hint: edit controls usually have a white
background on Windows.

```

```

%     See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- FUNCIÓN PARA LIMPIAR LOS DATOS,
VALORES Y GRÁFICAS -REINICIAR-.

```

```

function LimpiarDatos_Callback(hObject,
eventdata, handles)
%Parámetros y Resultados
set(handles.Vo,'string','');
set(handles.Vi,'string','');
set(handles.PoMax,'string','');
set(handles.DeltaIL,'string','');
set(handles.FS,'string','');
set(handles.DeltaVo,'string','');
set(handles.Ganancia,'string','');
set(handles.Inductor,'string','');
set(handles.Capacitor,'string','');
set(handles.IQ_Prom,'string','');
set(handles.IPK_Q,'string','');
set(handles.VMaxInt,'string','');
set(handles.ID,'string','');
set(handles.IDPK,'string','');
set(handles.VDMax,'string','');
set(handles.Graficar,'visible','off');

```

```

%Paneles y control axes23
set(handles.axes23,'visible','off');
set(findobj('Title','Formas de Onda
Características'),'Visible','off');
set(findobj('Title','Características en
Pequeña Señal (Small Signal)'),'Visible',
'off');

```

```

% --- FUNCIÓN PARA MOSTRAR GRÁFICAS
DERIVADAS DE LOS DATOS QUE INTRODUZCA
% --- EL USUARIO.

```

```

function Graficar_Callback(hObject,
eventdata, handles)
%Gráficas con pestañas
pGraficas = uipanel('Title','Formas de Onda
Características','FontSize',11,'ForegrounColor',
[.078 .169 .549],...
    'BorderWidth', 2,...
    'Position',[.003 .01 .48
.59]);

```

```

pGroup = uitabgroup('Parent', pGraficas);

```

```

pest1 = uitab('Parent', pGroup, 'Title',
'Pulso');
pest2 = uitab('Parent', pGroup, 'Title',
'Corriente en el Inductor');
pest3 = uitab('Parent', pGroup, 'Title',
'Corriente en el Interruptor');
pest4 = uitab('Parent', pGroup, 'Title',
'Corriente en el Diodo');
pest5 = uitab('Parent', pGroup, 'Title',
'Voltaje de Salida');

```

```

%gráficas de parámetros del convertidor
Buck
X = get(handles.Vo,'string');
Vo = str2double(X);
Y = get(handles.Vi,'string');
Vi = str2double(Y);
A = get(handles.Inductor,'string');
L = str2double(A);
B = get(handles.Capacitor,'string');
C = str2double(B);
W = get(handles.PoMax,'string');
Z = get(handles.FS,'string');
FS = str2double(Z);
PoMax = str2double(W);
I=PoMax/Vo;
R=Vo/I;
D=(Vo/Vi)*100;

assignin ('base', 'Vo', Vo);
assignin ('base', 'Vi', Vi);
assignin ('base', 'L', L);
assignin ('base', 'C', C);
assignin ('base', 'R', R);
assignin ('base', 'D', D);
assignin ('base', 'FS', FS);

%BARRA DE ESPERA%
h = waitbar(0,'Ejecutando, espere por
favor...');
%ASIGNAR COLOR AZUL%
hPatch = findobj(h,'Type','Patch');
set(hPatch,'FaceColor',[0 0 1],
'EdgeColor',[0 0 1]);
%SEGUNDOS DE ESPERA%
steps = 2000;
for step = 1:steps
    waitbar(step / steps)
end
close(h)

sim ('Buck_nuevo');

%Pulso
axPulso =
axes('Parent',pest1,'Position',[.07 .09 .8
.8]);
plot(t,PULSO);
zoom on;
set(axPulso, 'visible', 'on');
grid on;
title ('Pulso');

%IL1
axIL1 = axes('Parent',pest2,'Position',[.07
.09 .8 .8]);
plot(t,IL1);
zoom on;
set(axIL1, 'visible', 'on');
grid on;
title ('Corriente en el Inductor');

%IQ1
axIQ1 = axes('Parent',pest3,'Position',[.07
.09 .8 .8]);
plot(t,IQ1);
zoom on;
set(axIQ1, 'visible', 'on');
grid on

title ('Corriente en el Interruptor');
axes(handles.axes23);
plot(1,1);
set(handles.axes23, 'visible', 'on');

%ID1
axID1 = axes('Parent',pest4,'Position',[.07
.09 .8 .8]);
plot(t,ID1);
set(axID1, 'visible', 'on');
grid on;
title ('Corriente en el Diodo');

%Voltaje de Salida
axVo = axes('Parent',pest5,'Position',[.07
.09 .8 .8]);
plot(t,Vo);
set(axVo, 'visible', 'on');
grid on;
title ('Voltaje de Salida');

%hold off
find_system('name','Buck_nuevo');
set_param(gcs,'SimulationCommand','Start');

%Pestañas Pequeña Señal
p = uipanel('Title','Características en
Pequeña Señal (Small
Signal)','FontSize',11,'ForegroundColor',[.
078 .169 .549],...
'BorderWidth', 2,...
'Position',[.49 .01 .48 .59]);
tgroup = uitabgroup('Parent', p);

tab0 = uitab('Parent', tgroup, 'Title',
'Datos de Entrada');
tab1 = uitab('Parent', tgroup, 'Title',
'Entrada/Salida');
tab2 = uitab('Parent', tgroup, 'Title',
'Impedancia');
tab3 = uitab('Parent', tgroup, 'Title',
'Control/Salida');
tab4 = '';
tab5 = '';
% tab4 = uitab('Parent', tgroup, 'Title',
'Control PID');
% tab5 = uitab('Parent', tgroup, 'Title',
'Convertidor LC');

%Imagen Datos de Entrada TAB0
ax0 = axes('Parent',tab0,'Position',[.01
.10 .65 .95]);
imshow('Buck2.jpg','Parent',ax0);
axis off
RLText =
uicontrol(tab0,'Style','text',...
'String','RL =',...
'ForegroundColor',[.078
.169 .549],'FontSize',9,...
'Position', [410 210 70
23]);
RL =
uicontrol(tab0,'Style','edit',...
'String','',...
'BackgroundColor',[.804
.878 .969],'FontSize',10,...
'ForegroundColor','black',...

```

```

        'Position', [465 210 70
30]);
    RCText =
uicontrol(tab0, 'Style', 'text', ...
        'String', 'RC =', ...
        'ForegroundColor', [.078
.169 .549], 'FontSize', 9, ...
        'Position', [410 170 70
23]);
    RC =
uicontrol(tab0, 'Style', 'edit', ...
        'String', '', ...
'ForegroundColor', 'black', ...
        'BackgroundColor', [.804
.878 .969], 'FontSize', 10, ...
        'Position', [465 170 70
30]);

    SmallSignal =
uicontrol(tab0, 'Style', 'pushbutton', 'String
', 'Pequeña Señal', 'FontSize', 14, ...
        'Units', 'normalized', ...
        'BackgroundColor', [.078
.169 .549], ...
'ForegroundColor', 'white', ...
        'Position', [.70 .30 .25
.15], ...
'Callback', {@Smallsignal, tab0, tab1, tab2, tab
3, tab4, tab5, RL, RC, tgroup});

```

```

function RL_Callback(hObject, eventdata,
handles)
% hObject    handle to RL (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)
function axes23_CreateFcn(hObject,
eventdata, handles)
% hObject    handle to axes23 (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called

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

```

```

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

% Hint: edit controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.

```

```

if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

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

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

```

```

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

```

```

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

```

```

% --- Función para calcular la "pequeña
señal" a través de la llamada con el botón
de SmallSignal.
function
Smallsignal(hObject, callbackdata, TAB0, TAB1,
TAB2, TAB3, tab4, tab5, rl, rc, tgroup)

```

```

archivoSmallsignal(hObject, callbackdata, TAB
0, TAB1, TAB2, TAB3, tab4, tab5, rl, rc, tgroup);

```

```

return;

```

## Análisis en pequeña señal.

```

%---CENIDET
%---TESIS "Asistente para el Análisis y
Diseño de Topologías Comunes de
Convertidores CD-CD"
%---AUTOR: EIDER MIGUEL AMORES CAMPOS
%---DESCRIPCIÓN: ARCHIVO PARA EL CÁLCULO
DE LA PEQUEÑA SEÑAL
%----- En este apartado se hace el
desarrollo de la función de
%           Pequeña señal, la cuál es
.....

```



```

%---FECHA: ENERO 2016

function
Smallsignal(hObject,callbackdata,TAB0,TAB1,
TAB2,TAB3,tab4,tab5,rl,rc,tgroup)
%BARRA DE ESPERA%
h = waitbar(0,'Ejecutando, espere por
favor...');
%ASIGNAR COLOR AZUL%
hPatch = findobj(h,'Type','Patch');
set(hPatch,'FaceColor',[0 0 1],
'EdgeColor',[0 0 1]);
%SEGUNDOS DE ESPERA%
steps = 1500;
for step = 1:steps
    waitbar(step / steps)
end
close(h)

% Buck en pequeña señal
strRL = get(rl,'string');
RL = str2double(strRL);
strRC = get(rc,'string');
RC = str2double(strRC);

strCapacitor =
get(findobj('Tag','Capacitor'),'String');
C = str2double(strCapacitor);
strInductor =
get(findobj('Tag','Inductor'),'String');
L = str2double(strInductor);
strGanancia =
get(findobj('Tag','Ganancia'),'String');
M = str2double(strGanancia);
strPoMax =
get(findobj('Tag','PoMax'),'String');
PoMax = str2double(strPoMax);
strVo =
get(findobj('Tag','Vo'),'String');
Vo = str2double(strVo);
strVi =
get(findobj('Tag','Vi'),'String');
Vi = str2double(strVi);

I=PoMax/Vo;
R=Vo/I;

SZ1=1/(RC*C);

W0=sqrt((1/(L*C))*(1+(RL/L))/(1+(RC/R)));

Q=(1/W0)*(1/((L/(RL+R))+C*(RC+((RL*L)/(RL+R)))));
numerador=[1/SZ1 1];
denominador=[1/(W0*W0) 1/(W0*Q) 1];

w=logspace(-1,2,100);

[mag,fase]=bode(numerador,denominador,w);

assignin('base','RL',RL);
assignin('base','RC',RC);
assignin('base','L',L);
assignin('base','C',C);
assignin('base','M',M);
assignin('base','PoMax',PoMax);
assignin('base','Vo',Vo);
assignin('base','Vi',Vi);

assignin('base','I',I);
assignin('base','R',R);
assignin('base','SZ1',SZ1);
assignin('base','W0',W0);
assignin('base','Q',Q);
assignin('base','numerador',
numerador);
assignin('base','denominador',
denominador);
assignin('base','w',w);

%Ploteo de pequeña señal
y=tf(numerador*M,denominador);
assignin('base','y',y);

assignin('base','YValor',y);

%función de transferencia
axFT =
axes('Parent',TAB1,'Position',[.03 .1 .4
.3]);
% plot(step(y));
% zoom on;
% %set(handles.fTransferencia,
'visible','on');
% set(axFT,'visible','on');
% grid on;
% hold on;
% title('Función de Transferencia');

%Ploteo de Bode
[mag, ph, w]=bode(y);
magg = ones(1,length(mag));
magg(1:length(mag)) =
mag(:, :, 1:length(mag));
phh = ones(1,length(ph));
phh(1:length(ph)) = ph(:, :, 1:length(ph));
clear mag ph
mag=20*log10(magg);
ph=phh;

%CONTENIDO TAB1
%Entrada/Salida
%Imagen TAB1
ax1 = axes('Parent',TAB1,'Position',[.01
.40 .5 .7]);
imshow('Buck2.jpg','Parent',ax1);
axis off

set(gca,'FontSize',10,
'fontName','Helvetica');
axMag = axes('Parent',TAB1,'Position',[.60
.64 .37 .3]);
semilogx(w,mag);
grid on
title('Magnitud','FontWeight','bold')
xlabel('Frecuencia')
ylabel('Amplitud Db')
set(axMag,'visible','on');
set(gca,'FontSize',10,
'fontName','Helvetica');
axFas = axes('Parent',TAB1,'Position',[.60
.13 .37 .3]);%[.52 .15 .47 .3]);
semilogx(w,ph)
grid on
title('Fase','FontWeight','bold')
xlabel('Frecuencia')
ylabel('Amplitud')

```

```

set(axFas, 'visible', 'on');

%CONTENIDO TAB2
%Impedancia salida
%Imagen Impedancia TAB2
ax2 = axes('Parent',TAB2,'Position',[.01
.40 .5 .7]);
imshow('Buck3.jpg','Parent',ax2);
axis off

SZ11=RL/L;
SZ2=1/(RC*C);
numerador2=[1/(SZ11*SZ2) 1/SZ11+1/SZ2 1];
denominador2=[1/(W0*W0) 1/(W0*Q) 1];
Rparalelo=(RL*R)/(RL+R);
z=tf(numerador2*Rparalelo,denominador2);

assignin('base','z',z);

%Ploteo de Bode2
[mag, ph, w]=bode(z);
magg = ones(1,length(mag));
magg(1:length(mag)) =
mag(:, :, 1:length(mag));
phh = ones(1,length(ph));
phh(1:length(ph)) = ph(:, :, 1:length(ph));
clear mag ph
mag=20*log10(magg);
ph=phh;

set(gca, 'FontSize', 10,
'fontName','Helvetica');
axMag2 =
axes('Parent',TAB2,'Position',[.60 .64 .37
.3]);
semilogx(w,mag);
grid on
title('Magnitud','FontWeight','bold')
xlabel('Frecuencia')
ylabel('Amplitud Db')
set(axMag2, 'visible', 'on');
set(gca, 'FontSize', 10,
'fontName','Helvetica');
axFas2 = axes('Parent',TAB2,'Position',[.60
.13 .37 .3]);
semilogx(w,ph)
grid on
title('Fase','FontWeight','bold')
xlabel('Frecuencia')
ylabel('Amplitud')
set(axFas2, 'visible', 'on');

%CONTENIDO TAB3
%Control salida
%Imagen Control/Salida TAB3
ax3 = axes('Parent',TAB3,'Position',[.01
.40 .5 .7]);
imshow('Buck4.jpg','Parent',ax3);
axis off

cs=tf(numerador*Vi,denominador);

assignin('base','cs',cs);

%Ploteo de Bode3
[mag, ph, w]=bode(cs);
magg = ones(1,length(mag));
magg(1:length(mag)) =
mag(:, :, 1:length(mag));
phh = ones(1,length(ph));
phh(1:length(ph)) = ph(:, :, 1:length(ph));
clear mag ph
mag=20*log10(magg);
ph=phh;

set(gca, 'FontSize', 10,
'fontName','Helvetica');
axMag3 =
axes('Parent',TAB3,'Position',[.60 .64 .37
.3]);%[.52 .63 .47 .3]);
semilogx(w,mag);
grid on
title('Magnitud','FontWeight','bold')
xlabel('Frecuencia')
ylabel('Amplitud Db')
set(axMag3, 'visible', 'on');
set(gca, 'FontSize', 10,
'fontName','Helvetica');
axFas3 = axes('Parent',TAB3,'Position',[.60
.13 .37 .3]);%[.52 .15 .47 .3]);
semilogx(w,ph)
grid on
title('Fase','FontWeight','bold')
xlabel('Frecuencia')
ylabel('Amplitud')
set(axFas3, 'visible', 'on');

% --- SELECCIÓN DE CONTROL
tagControlText =
uicontrol(TAB3,'Style','text',...
'String','Tipo de control:
',...
'ForegroundColor',[.078
.169 .549],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[10 110 100
30]);
comboSelectorControl =
uicontrol(TAB3,'Style','popupmenu',...
'String',{'Seleccione el
tipo de control','PID'},...
'BackgroundColor',[.804
.878 .969],'FontSize',10,...
'ForegroundColor','black',...
'Position',[105 110 190
30],...
'Callback',{@ejecutarControl,tab4,tab5,y,nu
merador,denominador,M,tgroup});
% ejecutarControl =
uicontrol(TAB3,'Style','pushbutton','String
','Ejecutar','FontSize',14,...
% 'Units','normalized',...
% 'BackgroundColor',[.078
.169 .549],...
%
'ForegroundColor','white',...
% 'Position',[.15 .15 .25
.13],...
%
'Callback',{@ejecutarControl});

```

```

function
ejecutarControl(source,eventdata,tab4,tab5,
y,numerador,denominador,M,tgroup)

    tipoControlStr = get(source,
'String');
    tipoControl = get(source,'Value');

        switch
tipoControlStr{tipoControl};
        case 'PID'
            %CONTENIDO TAB4
            %Control PID
            %Imagen PID

TAB4
            tab4 =
uitab('Parent', tgroup, 'Title', 'Control
PID');
            ax4 =
axes('Parent',tab4,'Position',[.01 .40 .5
.7]);
imshow('PID.jpg','Parent',ax4);
            axis off

            %Control PID

            tab4

                MpText =
uicontrol(tab4,'Style','text',...
'String','Sobreimpulso (Mp) = ',...
'ForegroundColor',[.078 .169
.549],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[10 130 160 30]);
                Mp =
uicontrol(tab4,'Style','edit',...
'String','',...
'BackgroundColor',[.804 .878
.969],'FontSize',10,...
'ForegroundColor','black',...
'Position',[170 130 70 30]);
                SignoPorcentaje =
uicontrol(tab4,'Style','text',...
'String','% ',...
'ForegroundColor',[.078 .169
.549],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[250 130 70 30]);
                KvelText =
uicontrol(tab4,'Style','text',...
'String','Error en estado estacionario
(ess) = ',...
'ForegroundColor',[.078 .169
.549],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[10 90 165 30]);
                Kvel =
uicontrol(tab4,'Style','edit',...
'String','',...
'BackgroundColor',[.804 .878
.969],'FontSize',10,...
'ForegroundColor','black',...
'Position',[170 90 70 30]);
                t_picoText =
uicontrol(tab4,'Style','text',...
'String','Tiempo pico (t_pico) = ',...
'ForegroundColor',[.078 .169
.549],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[10 50 165 30]);
                t_pico =
uicontrol(tab4,'Style','edit',...
'String','',...
'BackgroundColor',[.804 .878
.969],'FontSize',10,...
'ForegroundColor','black',...
'Position',[170 50 70 30]);
                TextoSeg =
uicontrol(tab4,'Style','text',...
'String','Seg.',...
'ForegroundColor',[.078 .169
.549],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[250 50 70 30]);
                ControlPID =
uicontrol(tab4,'Style','pushbutton','String
','Control PID','FontSize',14,...
'Units','normalized',...
'BackgroundColor',[.078 .169 .549],...
'ForegroundColor','white',...
'Position',[.15 .001 .25 .13],...
'Callback',{@ControlPID,tab4,tab5,Mp,Kvel,t
_pico,y,numerador,denominador,M,tgroup});
                case 'Otro'
                otherwise

```

```

        beep;
        errordlg('Seleccione un
tipo de convertidor, intente
nuevamente','Selección errónea','modal')
        return;
    end
    return;

%Archivo para desplegar las funciones
cálculadas del control PID
function
ControlPID(hObject,callbackdata,TAB4,TAB5,M
p,Kvel,t_pico,y,numerador,denominador,M,tgr
oup)

archivoControlPID(hObject,callbackdata,TAB4
,TAB5,Mp,Kvel,t_pico,y,numerador,denominao
r,M,tgroup);

return;

```

## Control PID y perturbaciones.

```

%---CENIDET
%---TESIS "...
%---AUTOR: EIDER MIGUEL AMORES CAMPOS
%---DESCRIPCIÓN: ARCHIVO PARA EL CÁLCULO
DEL CONTROL PID
%----- En este apartado se hace el
desarrollo de la función de
%          control PID, la cuál es
%          .....
%---FECHA: ENERO 2016

function
ControlPID(hObject,callbackdata,TAB4,TAB5,M
p,Kvel,t_pico,y,numerador,denominador,M,tgr
oup)
%Verificación PID
global nnum dden ess kvel ki zeta wn poly
raices ynum j i yden newy newy_1 Ki_s s eq1
eq2 reseq1 reseq2 S kp kd numPID denPID PID
sys sys_feedback
strMp = get(Mp,'string');
Mp = (str2double(strMp))/100;
strKvel = get(Kvel,'string');
Kvel=str2double(strKvel);
ess=Kvel;
Kvel = 1/(1+ess);
strt_pico = get(t_pico,'string');
t_pico = str2double(strt_pico);
syms ki s kp kd

w=logspace(-1,2,100);
y=tf(numerador*M,denominador);

nnum = y.num{1,1};
dden = y.den{1,1};
kvel=nnum(end)/dden(end);
ki=Kvel/kvel;
zeta = abs(log(Mp)/(sqrt(pi^2 +
(log(Mp))^2)));
wn = pi/(t_pico*sqrt(1- zeta^2));

```

```

poly = [1 2*zeta*wn wn^2];
raices = roots(poly);

ynum=0;
j=length(nnum)-1;

for i=1:length(nnum)
    ynum=(nnum(i)*s^j)+ynum;
    j=j-1;
end

yden=0;
j=length(dden)-1;

for i=1:length(dden)
    yden=(dden(i)*s^j)+yden;
    j=j-1;
end

newy=ynum/yden;
newy_1 = -1/newy;
s=raices(1);
newy=eval(newy_1);
Ki_s = -ki/s;
eq1 = kp + real(s)*kd;
reseq1 = real(newy + Ki_s);
eq2 = imag(s)*kd;
reseq2 = imag(newy + Ki_s);
S=solve(vpa(eq1==reseq1),vpa(eq2==reseq2),k
p,kd);
kp=double(S.kp);
kd=double(S.kd);
numPID = [kd kp ki];
denPID = [1 0];
PID=tf(numPID,denPID);
sys=PID*y;
sys_feedback = feedback(sys,-1,1);
lcerrado = step(sys_feedback);

    assignin ('base', 'numerador',
numerador);
    assignin ('base', 'denominador',
denominador);
    assignin ('base', 'w', w);
    assignin ('base', 'kp', kp);
    assignin ('base', 'kd', kd);
    assignin ('base', 'ki', ki);
    assignin ('base', 'sys_feedback',
sys_feedback);

    %BARRA DE ESPERA%
h = waitbar(0,'Ejecutando, espere por
favor...');
%ASIGNAR COLOR AZUL%
hPatch = findobj(h,'Type','Patch');
set(hPatch,'FaceColor',[0 0 1],
'EdgeColor',[0 0 1]);

%-EJECUTAR SIMULINK-
sim ('Buck_nuevo_LC');

%SEGUNDOS DE ESPERA%
steps = 2500;
for step1 = 1:steps
    waitbar(step1 / steps)
end
close(h)

```

```

%CONTENIDO TAB4
axMag4 = axes('Parent',TAB4,'Position',[.60
.23 .37 .65]);
axis off;
cla reset %(axMag4,'reset');
set(axMag4,'XTick',[],'YTick',[]);
plot(t,V0);
%plot(lcerrado);
axis on;
zoom on;
set(axMag4, 'visible', 'on');
grid on;
title ('Control PID');
%delete (axMag4);

%ETIQUETAS DE LA GRÁFICA DEL CONVERTIDOR

kpText =
uicontrol(TAB4,'Style','text',...
'String','KP = ',...
'ForegroundColor',[0.0 0.0
0.0],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[355 10
50 20]);
kiText =
uicontrol(TAB4,'Style','text',...
'String','KI = ',...
'ForegroundColor',[0.0 0.0
0.0],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[435 10
50 20]);
kdText =
uicontrol(TAB4,'Style','text',...
'String','KD =
',...
'ForegroundColor',[0.0 0.0
0.0],'FontSize',9,...
'HorizontalAlignment','left',...
'Position',[515 10
50 20]);
kpValue =
uicontrol(TAB4,'Style','text',...
'String','',...
'BackgroundColor',[.831 .816
.784],'FontSize',10,...
'ForegroundColor','black',...
'Position',[380 10
50 20]);
kiValue =
uicontrol(TAB4,'Style','text',...
'String','',...
'BackgroundColor',[.831 .816
.784],'FontSize',10,...
'ForegroundColor','black',...
'Position',[460 10
50 20]);
kdValue =
uicontrol(TAB4,'Style','text',...
'String','',...
'BackgroundColor',[.831 .816
.784],'FontSize',10,...
'ForegroundColor','black',...
'Position',[540 10
45 20]);
set(kpValue, 'string', num2str(kp));
set(kiValue, 'string', num2str(ki));
set(kdValue, 'string', num2str(kd));

%CONTENIDO TAB5
%Perturbaciones PID
sim
('Buck_nuevo_LC_ValidacionPID');

TAB5 = uitab('Parent', tgroup,
'Title', 'Perturbaciones');

axPerturbacion1 =
axes('Parent',TAB5,'Position',[.05 .11 .42
.3]);
plot(t,V0);
axis on;
zoom on;
set(axPerturbacion1, 'visible',
'on');
grid on;
title ('Comportamiento de
Voltaje de Salida');

axPerturbacion1 =
axes('Parent',TAB5,'Position',[.05 .62 .42
.3]);
plot(t,Vo);
axis on;
zoom on;
set(axPerturbacion1, 'visible',
'on');
grid on;
title ('Perturbación en el
Voltaje Inicial');

axPerturbacion2 =
axes('Parent',TAB5,'Position',[.55 .62 .42
.3]);
plot(t,R);
axis on;
zoom on;
set(axPerturbacion2, 'visible',
'on');
grid on;
title ('Perturbación en R');

axPerturbacion3 =
axes('Parent',TAB5,'Position',[.55 .11 .42
.3]);
plot(t,Io);
axis on;
zoom on;
set(axPerturbacion3, 'visible',
'on');
grid on;

```

```
title ('Comportamiento Io con  
Perturbación en R');
```

***cenidet***<sup>®</sup>  
*Centro Nacional de Investigación  
y Desarrollo Tecnológico*