

UNIVERSIDAD MIGUEL HERNÁNDEZ DE
ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



"HERRAMIENTAS PARA EL
MODELADO DE INDUCTORES DE
POTENCIA"

TRABAJO FIN DE GRADO

Septiembre-2021

AUTOR: Pablo Miguel Puche Azorín

DIRECTOR: David Marroquí Sempere

ÍNDICE

1. RESUMEN.....	1
2. OBJETIVOS	3
3. INTRODUCCIÓN	4
3.1 Inductor	4
3.1.1 Propiedades magnéticas	5
3.1.2 Tipos de Inductores	7
3.1.3 Materiales magnéticos del núcleo	8
3.2 Estado del Arte.....	11
4. MODELADO DE INDUCTOR BÁSICO	12
4.1 Explicación teórica modelo básico.....	12
4.2 Obtención de datos	13
4.2.1 Bode 100	13
4.2.2 Obtención de datos mediante Bode100.....	14
4.3 Cálculo de componentes del modelo equivalente	16
4.4 Implementación en App Designer.....	18
4.4.1 App Designer	18
4.4.2 Interfaz gráfica del modelo equivalente del inductor de potencia.....	20
4.5 Obtención modelo LTspice a partir del modelo equivalente.....	23
4.5.1 LTspice.....	24
4.5.2 Obtención modelo para usar en LTspice.....	25
5. MODELADO INDUCTOR AVANZADO.....	29
5.1 Explicación teórica del modelo Chan.....	30
5.2 Obtención de variables para nuestro modelo	30
5.2.1 Obtención variables mediante Ciclo Histéresis.....	30
5.2.2 Obtención variables mediante método Kg	32
5.3 Obtención datos experimentales.....	37
5.4 Implementación en App Designer.....	37
5.4 Obtención del modelo en LTspice	42
6. VALIDACIÓN EXPERIMENTAL – Convertidor Buck	44
6.1 Convertidor Reductor – <i>Buck</i>	45
6.1.1 Componentes seleccionados para el DC-DC <i>Buck</i>	46
6.2 Diseño del circuito Buck	48
6.2.1 EAGLE.....	48
6.2.2 Diseño del circuito	50
7. RESULTADOS EXPERIMENTALES.....	52
7.1. Pruebas de validación con el modelo básico	52
7.1.1 DC1012-563L.....	55

7.1.2 RFS1317-224KL	59
7.1.3 RFS1317-473KL	63
7.1.4 RFB1010-101L.....	67
7.1.5 RFS1317-824KL	71
7.1.6 Primer inductor con forma de núcleo ETD	75
7.1.7 Segundo inductor con forma de núcleo ETD	79
7.1.8 Tercer inductor con forma de núcleo ETD.....	83
7.2. Pruebas de validación con el modelo avanzado	87
7.2.1 Segundo inductor con forma de núcleo ETD	88
8. CONCLUSIONES Y LÍNEAS FUTURAS	92
9. BIBLIOGRAFÍA.....	94
10. ANEXOS.....	95
10.1 FORMAS GEOMÉTRICAS DEL NÚCLEO MAGNÉTICO	95
10.2 ESQUEMATICO DEL CONVERTIDOR DC-DC BUCK	99
10.3 CAPAS DEL DISEÑO DEL CIRCUITO <i>BUCK</i>	100
10.4 CÓDIGO APLICACIÓN INDUCTOR BÁSICO.....	102
10.5 CÓDIGO APLICACIÓN INDUCTOR AVANZADO	108



INDICE DE ILUSTRACIONES

Fig. 1 Inductores de núcleo de aire y material ferroso. [11].....	4
Fig. 2 Gráfica L vs I de inductores RFS1317. [16].....	5
Fig. 3 Ciclo de Histéresis de un material magnético suave. [2].....	7
Fig. 4 Inductor toroidal [11].....	8
Fig. 5 Inductores de núcleo de aire [11].....	9
Fig. 6 Inductor con núcleo de ferrita y forma de E [11].....	10
Fig. 7 Modelo equivalente Inductor de Potencia. [3].....	12
Fig. 8 Gráfica de magnitud y fase de impedancia de un inductor.....	12
Fig. 9 Esquema conexiones Bode100 [4].....	13
Fig. 10 Parámetros Configuración Barrido de Frecuencia Bode100.....	14
Fig. 11 Configuración Parámetros a medir en Bode100.....	15
Fig. 12 Barrido en frecuencia de la Impedancia de un Inductor.....	16
Fig. 13 Entorno de trabajo App Designer "Design View".....	19
Fig. 14 Entorno de trabajo App Designer "Code View".....	20
Fig. 15 Figura del buscador de la aplicación.....	21
Fig. 16 Figura del buscador de la aplicación.....	21
Fig. 17 Vista Inicial de la aplicación.....	23
Fig. 18 Vista de la aplicación una vez introducido los datos.....	23
Fig. 19 Circuito del modelo equivalente en LTspice.....	25
Fig. 20 Circuito del modelo equivalente en LTspice.....	27
Fig. 21 Símbolo del modelo equivalente.....	28
Fig. 22 Gráfica B-H del material 3C96 [21].....	31
Fig. 23 Inductor a) Estructura; b) Circuito magnético equivalente [2].....	33
Fig. 24 Representación donde el devanado debe encajar con el área del núcleo magnético [2].....	35
Fig. 25 Vista aplicación modelo avanzado.....	38
Fig. 26 Sección de la aplicación de modelado avanzado.....	38
Fig. 27 Sección de la aplicación de modelado avanzado.....	39
Fig. 28 Esquema circuital de un Convertidor Buck. [5].....	45
Fig. 29 Grafica representa el voltaje y la intensidad a la salida del inductor en un circuito Buck [5].....	46
Fig. 30 Shield PWM [9].....	47
Fig. 31 Interfaz gráfica de la aplicación asociada al PWM [9].....	47
Fig. 32 Entorno de trabajo del Esquemático en EAGLE.....	49
Fig. 33 Entorno de trabajo del "Board" en EAGLE.....	49
Fig. 34 Diseño de circuito Buck en Eagle.....	50
Fig. 35 Diseño de la distribución de los componentes de la placa en Eagle.....	50
Fig. 36 Circuito de Validación.....	51
Fig. 37 Circuito de validación con inductor ETD.....	51
Fig. 38 Circuito Buck implementado en LTspice.....	54

ÍNDICE DE FIGURAS

<i>Tabla 1</i>	<i>Tabla general con datos de los inductores</i>	<i>52</i>
<i>Tabla 2</i>	<i>Comparativa resultados simulación</i>	<i>55</i>
<i>Tabla 3</i>	<i>Condiciones Iniciales con la 1ª prueba. Modelo: DC1012-563L</i>	<i>55</i>
<i>Tabla 4</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: DC1012-563L</i>	<i>56</i>
<i>Tabla 5</i>	<i>Condiciones Iniciales con la 2ª prueba. Modelo: DC1012-563L</i>	<i>56</i>
<i>Tabla 6</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: DC1012-563L</i>	<i>57</i>
<i>Tabla 7</i>	<i>Condiciones Iniciales con la 3ª prueba. Modelo: DC1012-563L</i>	<i>57</i>
<i>Tabla 8</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: DC1012-563L</i>	<i>58</i>
<i>Tabla 9</i>	<i>Comparativa resultados simulación</i>	<i>59</i>
<i>Tabla 10</i>	<i>Condiciones Iniciales con la 1ª prueba. Modelo: RFS1317-224KL</i>	<i>59</i>
<i>Tabla 11</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFS1317-224KL</i>	<i>60</i>
<i>Tabla 12</i>	<i>Condiciones Iniciales con la 2ª prueba. Modelo: RFS1317-224KL</i>	<i>60</i>
<i>Tabla 13</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFS1317-224KL</i>	<i>61</i>
<i>Tabla 14</i>	<i>Condiciones Iniciales con la 3ª prueba. Modelo: RFS1317-224KL</i>	<i>61</i>
<i>Tabla 15</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFS1317-224KL</i>	<i>62</i>
<i>Tabla 16</i>	<i>Comparativa resultados simulación</i>	<i>63</i>
<i>Tabla 17</i>	<i>Condiciones Iniciales con la 1ª prueba. Modelo: RFS1317-473KL</i>	<i>63</i>
<i>Tabla 18</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFS1317-473KL</i>	<i>64</i>
<i>Tabla 19</i>	<i>Condiciones Iniciales con la 2ª prueba. Modelo: RFS1317-473KL</i>	<i>64</i>
<i>Tabla 20</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFS1317-473KL</i>	<i>65</i>
<i>Tabla 21</i>	<i>Condiciones Iniciales con la 3ª prueba. Modelo: RFS1317-473KL</i>	<i>65</i>
<i>Tabla 22</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFS1317-473KL</i>	<i>66</i>
<i>Tabla 23</i>	<i>Comparativa resultados simulación</i>	<i>67</i>
<i>Tabla 24</i>	<i>Condiciones Iniciales con la 1ª prueba. Modelo: RFB1010-101L</i>	<i>67</i>
<i>Tabla 25</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFB1010-101L</i>	<i>68</i>
<i>Tabla 26</i>	<i>Condiciones Iniciales con la 2ª prueba. Modelo: RFB1010-101L</i>	<i>68</i>
<i>Tabla 27</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFB1010-101L</i>	<i>69</i>
<i>Tabla 28</i>	<i>Condiciones Iniciales con la 3ª prueba. Modelo: RFB1010-101L</i>	<i>69</i>
<i>Tabla 29</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFB1010-101L</i>	<i>70</i>
<i>Tabla 30</i>	<i>Comparativa resultados simulación</i>	<i>71</i>
<i>Tabla 31</i>	<i>Condiciones Iniciales con la 1ª prueba. Modelo: RFS1317-824KL</i>	<i>71</i>
<i>Tabla 32</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFS1317-824KL</i>	<i>72</i>
<i>Tabla 33</i>	<i>Condiciones Iniciales con la 2ª prueba. Modelo: RFS1317-824KL</i>	<i>72</i>
<i>Tabla 34</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFS1317-824KL</i>	<i>73</i>
<i>Tabla 35</i>	<i>Condiciones Iniciales con la 3ª prueba. Modelo: RFS1317-824KL</i>	<i>73</i>
<i>Tabla 36</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFS1317-824KL</i>	<i>74</i>
<i>Tabla 37</i>	<i>Condiciones Iniciales con la 1ª prueba. Modelo: 1r Inductor ETD</i>	<i>75</i>
<i>Tabla 38</i>	<i>Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: 2n Inductor ETD</i>	<i>76</i>

<i>Tabla 39 Condiciones Iniciales con la 2ª prueba. Modelo: 1r Inductor ETD</i>	76
<i>Tabla 40 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: 1r Inductor ETD</i>	77
<i>Tabla 41 Condiciones Iniciales con la 3ª prueba. Modelo: 1r Inductor ETD</i>	77
<i>Tabla 42 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: 1r Inductor ETD</i>	78
<i>Tabla 43 Condiciones Iniciales con la 1ª prueba. Modelo: 2n Inductor ETD</i>	79
<i>Tabla 44 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: 2n Inductor ETD</i>	80
<i>Tabla 45 Condiciones Iniciales con la 2ª prueba. Modelo: 2n Inductor ETD</i>	80
<i>Tabla 46 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: 2n Inductor ETD</i>	81
<i>Tabla 47 Condiciones Iniciales con la 3ª prueba. Modelo: 2n Inductor ETD</i>	81
<i>Tabla 48 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: 2n Inductor ETD</i>	82
<i>Tabla 49 Condiciones Iniciales con la 1ª prueba. Modelo: 3r Inductor ETD</i>	83
<i>Tabla 50 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: 3r Inductor ETD</i>	84
<i>Tabla 51 Condiciones Iniciales con la 2ª prueba. Modelo: 3r Inductor ETD</i>	84
<i>Tabla 52 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: 3r Inductor ETD</i>	85
<i>Tabla 53 Condiciones Iniciales con la 3ª prueba. Modelo: 3r Inductor ETD</i>	85
<i>Tabla 54 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: 3r Inductor ETD</i>	86
<i>Tabla 55 Condiciones Iniciales con la 1ª prueba. Modelo: 2n Inductor ETD</i>	88
<i>Tabla 56 Comparativa Intensidad entre Modelo Avanzado y Circuito Real en la 1ª prueba. Modelo: 2n Inductor ETD</i>	89
<i>Tabla 57 Condiciones Iniciales con la 2ª prueba. Modelo: 2n Inductor ETD</i>	89
<i>Tabla 58 Comparativa Intensidad entre Modelo Avanzado y Circuito Real en la 2ª prueba. Modelo: 2n Inductor ETD</i>	90
<i>Tabla 59 Condiciones Iniciales con la 3ª prueba. Modelo: 2n Inductor ETD</i>	90
<i>Tabla 60 Comparativa Intensidad entre Modelo Avanzado y Circuito Real en la 3ª prueba. Modelo: 2n Inductor ETD</i>	91

AGRADECIMIENTOS

A mis padres por apoyarme siempre y por estar en las buenas y en las no tan buenas.

A mi tutor David por haber confiado en mí a pesar de todo y por la ayuda que me ha brindado tanto para el trabajo como para mi vida.

A mis compañeros de clases y amigos por hacer estos años de universidad mucho más amenos y divertidos.



1. RESUMEN

En los últimos años, la necesidad de obtener modelos equivalentes más precisos se ha vuelto indispensable. Esto es debido al aumento de complejidad de los distintos elementos que componen los sistemas de electrónica de potencia como son las bobinas y, en su defecto, los transformadores. Estos elementos forman la base de un circuito electrónico de potencia y, por ende, es necesario obtener modelos informáticos sobre dichos elementos para así poder realizar simulaciones que se asemejen a la realidad. Esta necesidad surge de la necesidad de optimizar los circuitos para poder obtener mejores resultados a raíz de optimizar las pérdidas generadas en circuitos de potencia donde se trabaja con unas intensidades y voltajes elevados. Para la obtención de estos modelos, tenemos dos formas de conseguirlos. El primero sería a través del diseño del elemento, donde se parte de unas condiciones iniciales, las cuales dicho elemento debe cumplir de forma teórica. La segunda sería tratar el elemento como una caja negra de la que no se conoce nada y, a través de distintas mediciones, se obtendría un modelo equivalente con el cual podríamos trabajar en simulaciones. Con este método obtendríamos una mejora considerable a la hora de diseñar circuitos electrónicos ya que podríamos hacer simulaciones mucho más acordes a la realidad.

En este proyecto, se ha hecho uso del segundo método comentado. Se ha diseñado y desarrollado una interfaz gráfica, mediante el programa Matlab, que permite obtener modelos equivalentes de inductores a partir de ciertas medidas obtenidas de forma experimental del inductor a modelar. De esta forma, se tiene una interfaz que permite obtener modelos equivalentes de cualquier inductor posible. Esto permitirá obtener mejores resultados para futuras simulaciones de circuitos de electrónica de potencia.

Por un lado, se realizará un modelado de un inductor en frecuencia. Para ello, se usará el dispositivo Bode100, un equipo analizador de impedancias cuyas funciones se explican en el apartado 4.2.1, para obtener el comportamiento del inductor en un rango de frecuencias determinado. Una vez adquiridos dichos datos, se procederá a obtener un modelo equivalente del inductor haciendo uso de diferentes conceptos teóricos. Una vez realizado, se obtendrá un modelo equivalente que se comportará de la misma forma que

el inductor en alta frecuencia. A continuación, se procedió a desarrollar la interfaz gráfica con el software App Designer, perteneciente a MATLAB y explicado en el punto 4.4.1, al que se le introducirán los datos obtenidos por Bode100 y la aplicación generará un archivo con el modelo equivalente del inductor. Finalmente, se hará uso del software de simulación LTspice, software de simulación explicado en el punto 4.5.1, para analizar y comparar el comportamiento del modelo obtenido.

Por otro lado, se realizará otra interfaz gráfica capaz de crear modelos de inductores a partir de unas condiciones previas de diseño. Para esto, se hará uso de un método de diseño de inductores llamado “método kg”. En este caso, se procederá a programar la interfaz acorde al método nombrado anteriormente y, una vez finalizado, se obtendrá un modelo de inductor más avanzado que será utilizado en el software LTspice para analizar y comparar el comportamiento del modelo obtenido.



2. OBJETIVOS

El objetivo principal de este trabajo es desarrollar dos aplicaciones que, ante unas entradas, sean capaces de generar un modelo SPICE de simulación de un inductor. La primera aplicación generará un modelo a partir del circuito equivalente de un inductor en frecuencia. La segunda aplicación generará un modelo más avanzado que será capaz de simular los efectos de la saturación del inductor. Los objetivos secundarios y específicos de este proyecto son los siguientes:

- Definición de un método de caracterización de inductores basado en el uso de un analizador de redes vectoriales (Bode100)
- Generación de un entorno amigable, basado en Matlab, que permita la generación de los ficheros finales para simulación en un entorno de simulación SPICE, concretamente el software LTSpice.
- Generación de un convertidor DC-DC Buck capaz de aceptar todos los encapsulados de inductor que se pretenden validar experimentalmente.
- Evaluar los diferentes modelos de inductores creados por medio de comparativas de los resultados experimentales y de simulación.
- Redacción de la presente memoria científica.

3. INTRODUCCIÓN

3.1 Inductor

El inductor es un componente electrónico pasivo que está formado por un cable con la forma de una bobina que enrolla un núcleo cuya función es almacenar y liberar energía al transformarla en campo magnético a través de la inducción. Cuando se introduce una corriente eléctrica al inductor, este produce un voltaje variable en función de la corriente, esta propiedad se llama inductancia y se mide en henrios, H. En la figura 1 podemos observar tanto un inductor con núcleo de aire como un inductor con núcleo de material magnético como podría ser la ferrita.



Fig. 1 Inductores de núcleo de aire y material ferroso. [11]

Normalmente, los inductores suelen clasificarse según el tipo de núcleo sobre el cual se enrolla el cable de cobre. Los materiales más comunes que forman el núcleo de los inductores suelen ser aire o cualquier material no magnético, hierro o ferrita. En función del tipo de núcleo que posea el inductor, este será más adecuado para trabajar en aplicaciones de alta frecuencia o en electrónica de potencia, esto hace que la variedad de usos de la bobina sea prácticamente ilimitada debido a su gran versatilidad para trabajar en distintos ámbitos.

En este trabajo nos centraremos en inductores que trabajen en rangos de frecuencia alrededor de los 100 KHz para electrónica de potencia, esto quiere decir que necesitaremos inductores de elevada inductancia capaces de trabajar en altos rangos de corriente. Debido a las altas corrientes en las que trabajan estos inductores necesitamos

que no se produzca una saturación del inductor ya que esto provocaría la disminución de la inductancia del inductor. Este dato suele venir dado por el fabricante en una gráfica como la siguiente figura donde la inductancia varía en función de la corriente que atraviesa el inductor.

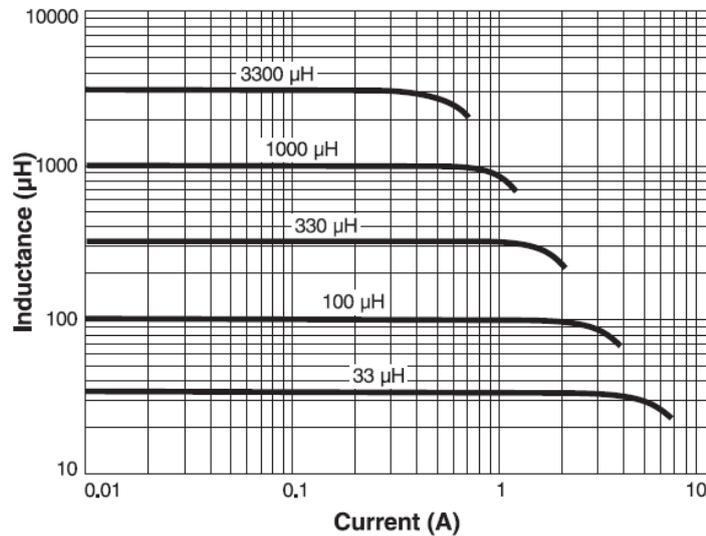


Fig. 2 Gráfica L vs I de inductores RFS1317. [16]

3.1.1 Propiedades magnéticas

Cuando una corriente fluye a través de un cable enrollado se produce una fuerza magnética circulatoria, H , y un campo magnético, B , alrededor del conductor. La magnitud del flujo viene determinada por la corriente, I , y el número de vueltas del bobinado, N . La relación entre el campo magnético y la fuerza magnética viene dada por la fórmula 1, donde nos encontramos la permeabilidad magnética.

$$H = \mu_0 * B \quad (1)$$

- **Permeabilidad Magnética**

La permeabilidad es la capacidad de un material de conducir un flujo magnético y nos permite saber cómo de bien se puede magnetizar el material. La mayoría de los materiales suelen ser malos conductores de flujo magnético debido a la baja permeabilidad que poseen. La permeabilidad en el vacío, se usa como referencia y está definida como una constante magnética. Muchos materiales como el aire, papel o el cobre poseen una permeabilidad similar a la del vacío y se les consideran como materiales no magnéticos.

- **Ciclo de histéresis (B-H)**

El ciclo de histéresis representa el ciclo completo de magnetización y desmagnetización de un material ferromagnético como podemos observar en la figura 3. Este ciclo comienza en el origen, en el punto donde la densidad de flujo, B , es cero y el campo magnético tiene un valor H_c , cuyo valor corresponde a la fuerza coercitiva explicada más adelante. . Conforme el campo magnético, H , aumenta, la densidad de flujo magnético, B , también crece hasta llegar al punto de saturación, B_s . A continuación, H comienza a disminuir hasta llegar a cero, pero B no ha llegado a cero, eso significa que el material aun esta magnetizado, es decir, que todavía existe un flujo magnético en el material. Este flujo que existe en ese momento se llama flujo remanente y tiene una densidad de flujo B_r .

En este momento, el campo magnético, H , revierte su polaridad hacia un valor negativo y se repite el mismo proceso realizado anteriormente. En este caso, nos centraremos en la fuerza coercitiva, H_c , La fuerza coercitiva representa la intensidad de campo magnético, H , para reducir el flujo remanente a cero y desmagnetizar el material completamente.

El ciclo de histéresis nos permite conocer la pérdida de energía en el núcleo del material ferromagnético ya que el área encerrada que vemos en la figura 3 representa la medida de pérdida de energía en el material del núcleo durante el ciclo. Esta pérdida de energía viene compuesta por dos componentes: las pérdidas producidas por la histéresis y las pérdidas por las corrientes de Foucault. Estas últimas se producen por el paso del flujo a través del núcleo y esto hace que se produzca un aumento de temperatura en el núcleo.

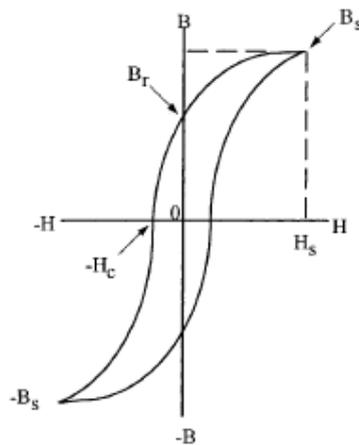


Fig. 3 Ciclo de Histéresis de un material magnético suave. [2]

3.1.2 Tipos de Inductores

Actualmente existen multitud de tipos de inductores basados tanto en la forma que tienen como en los materiales con los que se ha fabricado. Cada uno tienen distintos comportamientos y tienen una serie de ventajas y desventajas en función del uso para el que se requiera. Para diferenciarlos, por un lado, vamos a separarlos en función del material y en función de la forma que tienen.

- **Inductor con núcleo axial**

Una bobina está formada por un cable que rodea a un núcleo y da varias vueltas. Cuando circula una corriente eléctrica por el cable, se genera el flujo magnético con la capacidad de inducir el voltaje que circula por el inductor. Este tipo de forma puede ser usada en una gran variedad de inductores que tienen núcleos hechos por distintos materiales.

- **Inductor con núcleo toroidal**

Los núcleos toroidales presentan una simetría de su circuito magnético que provoca un flujo de fuga bajo. Esto hace que sea un circuito cerrado que tenga un mayor campo magnético y, por tanto, mayor inductancia que el resto. Además, los inductores toroidales son capaces de conducir mayor intensidad y producir una menor interferencia electromagnética.[11]

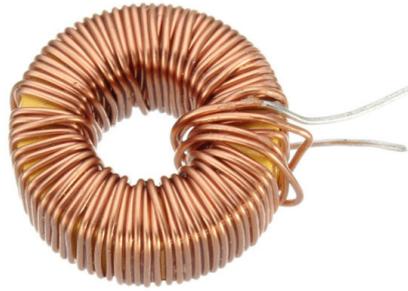


Fig. 4 Inductor toroidal [11]

- **Inductor con núcleo hueco**

Para prevenir la saturación del núcleo en inductores de potencia, huecos de aire pueden ser usados en el núcleo. Esto se produce debido a que el introducir un hueco de aire en un núcleo genera un gran efecto desmagnetizante, además de un decrecimiento considerable en la permeabilidad del material del núcleo. Sin embargo, es importante tener en cuenta que genera un campo de fuga considerable por lo que es recomendable no colocar cables cerca del núcleo del inductor.

3.1.3 Materiales magnéticos del núcleo

El material magnético que se utiliza para construir los inductores es uno de los factores a tener en cuenta para poder determinar su comportamiento ya que, en función del material usado por el fabricante, el inductor será capaz de trabajar en un rango de frecuencia y corriente determinados. Actualmente, los inductores son fabricados principalmente a partir de materiales magnéticos como acero eléctrico, “*permalloy*” (aleación de hierro y níquel), “*permendur*” (aleación de hierro y cobalto), ferrita y aleaciones metálicas amorfas. Además, también existen materiales derivados como polvo de hierro, “*permalloy*” o de “*sendust*”, que sirven como materiales magnéticos para fabricar núcleos magnéticos.

- **Acero eléctrico**

El acero eléctrico fue una de las primeras aleaciones en ser utilizadas para fabricar inductores y, actualmente, es uno de los materiales magnéticos más empleados. En el pasado solo se usaba acero, a priori no existía ningún problema para usarlo, pero, con el

tiempo, se dieron cuenta que conforme pasaban los años, las pérdidas aumentaban. Debido a esto, decidieron realizar una aleación con silicona, esto hizo que mejorara la estabilidad del material con el paso del tiempo. El acero eléctrico ofrece una alta saturación de densidad de flujo magnético, una permeabilidad relativamente buena con un elevado flujo magnético y unas pérdidas aceptables en un rango de frecuencia de audio.

- **Núcleo de aire**

Con este tipo de núcleo, el aire se usa como medio para almacenar energía magnética. Esta falta de un núcleo ferromagnético le proporciona muchas ventajas para aplicaciones de conmutación de alta frecuencia como no tener pérdidas en el núcleo o ser capaz de no saturarse. Sin embargo, el no tener un núcleo ferromagnético le limita a la hora de soportar altas intensidades y voltajes. Es por esto por lo que este tipo de inductor es recomendable para aplicaciones de baja potencia.



Fig. 5 Inductores de núcleo de aire [11]

- **Hierro**

Utilizar inductores con núcleo de hierro nos permite tener una mayor capacidad para almacenar energía electromagnética. Además, el núcleo de hierro tiene la capacidad de aumentar la inductancia del inductor. A pesar de ser capaz de aumentar la inductancia del inductor, el hierro produce pérdidas muy altas en el núcleo magnético en altas frecuencias. Es por esto por lo que este tipo de núcleos son utilizados en aplicaciones donde se necesite elevados valores de intensidad y voltaje a bajas frecuencias.

- **Ferrita blanda**

La ferrita es un material cerámico y homogéneo compuesto de óxidos, de los cuales, el óxido de hierro es el que se presenta en mayor cantidad. La ferrita blanda se

puede dividirse en dos categorías; manganeso-zinc y níquel-zinc. Dentro de cada categoría existen diferentes materiales que se obtienen a partir del grado de manganeso-zinc o níquel-zinc que poseen dichos materiales. La mayor diferencia entre el manganeso-zinc y el níquel-zinc es que los materiales de manganeso-zinc poseen una mayor permeabilidad y el níquel-zinc posee una mayor resistividad.

Cuando hablamos de su uso, podemos diferenciar el uso de ambas categorías, previamente mencionadas, en función de la frecuencia de trabajo a la que está sometido el inductor. Esto es debido a que los materiales de manganeso-zinc son usados en frecuencias de trabajo menores de 2MHz mientras que los materiales de níquel-zinc son capaces de trabajar en un rango de frecuencia que va desde 1-2MHz hasta los cientos de MHz.



Fig. 6 Inductor con núcleo de ferrita y forma de E.

- **Núcleo de polvo de hierro**

El desarrollo de núcleos de polvo de hierro comprimido se debió en gran parte a la necesidad de encontrar un sustituto para núcleos de hierro-cable. Para obtener este material, se utilizaba polvo de hierro electrolítico, el cual era templado e insolado para así, obtener una fina capa de granos de hierro insolados. Estos no se quebraban después de comprimirlos. Gracias a esto, se consiguió un material con una permeabilidad de menos de 100 y una gran capacidad de almacenamiento de energía.

Este material apenas genera pérdidas en el núcleo cuando se le aplica una corriente en DC. Sin embargo, al aplicarle una corriente AC de alta frecuencia, no se recomienda usar este tipo de núcleo ya que las pérdidas que genera en el núcleo serán un factor muy limitante en ciertas aplicaciones.

Actualmente existen cuatro estándares de materiales de polvo disponibles para componentes magnéticos: núcleo de polvo de “Mollypermalloy” (MPP), núcleo de polvo de alto flujo magnético, núcleo de polvo de “Sendust” y núcleo de polvo de hierro.

3.2 Estado del Arte

La necesidad de tener simulaciones del comportamiento de circuitos electrónicos lo más acordes a la realidad ha hecho que la investigación de componentes electrónicos derive en el estudio del comportamiento bajo diferentes condiciones de trabajo de componentes electrónicos. Al tener un modelo acorde a la realidad, el diseño de circuitos se vuelve óptimo a la hora de tener en cuenta posibles pérdidas generadas tanto por el aumento de temperatura, por la posible saturación de componentes debido a la intensidad que recorre por el circuito, posibles efectos electromagnéticos...

En este tipo de investigaciones, encontramos principalmente a los fabricantes de componentes electrónicos como podrían ser “Murata”, “Coilcraft”, “Farnell” ... Todos ellos ofrecen modelos de simulación que están asociados a cada componente electrónico. Estos modelos están disponibles desde transistores hasta condensadores, inductores y el resto de los componentes que ofertan. Cuando nos adentramos en el apartado de inductores de una compañía como “Coilcraft”, vemos que nos ofrece distintos modelos de simulación para distintos programas entre los que se encuentran los basados en SPICE. Muchos de estos fabricantes suelen ofrecer un modelo simple y otro modelo más avanzado que suele tener en cuenta más factores como la saturación o la temperatura para ofrecer un modelo más realista.

4. MODELADO DE INDUCTOR BÁSICO

4.1 Explicación teórica modelo básico

Para la realización de este TFG, comenzamos con el modelo de inductor de potencia sencillo que se basa en el comportamiento del inductor en frecuencia. Esto es debido a que existe un circuito equivalente el cual podemos ver en la figura 4. Básicamente, el circuito consiste en un condensador, C_p , el cual representa la capacidad parásita del circuito. Este condensador está conectado en paralelo tanto a la resistencia R_p , que representa las pérdidas del núcleo, como a L y R_s , las cuales están en serie entre ellos y representan la inductancia ideal del inductor y la resistencia total del bobinado.

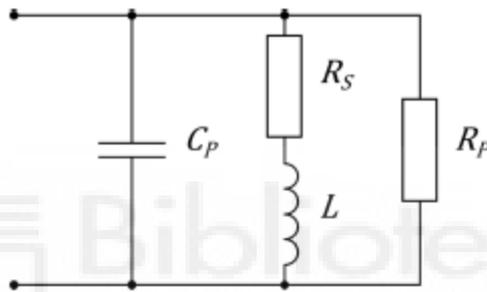


Fig. 7 Modelo equivalente Inductor de Potencia. [3]

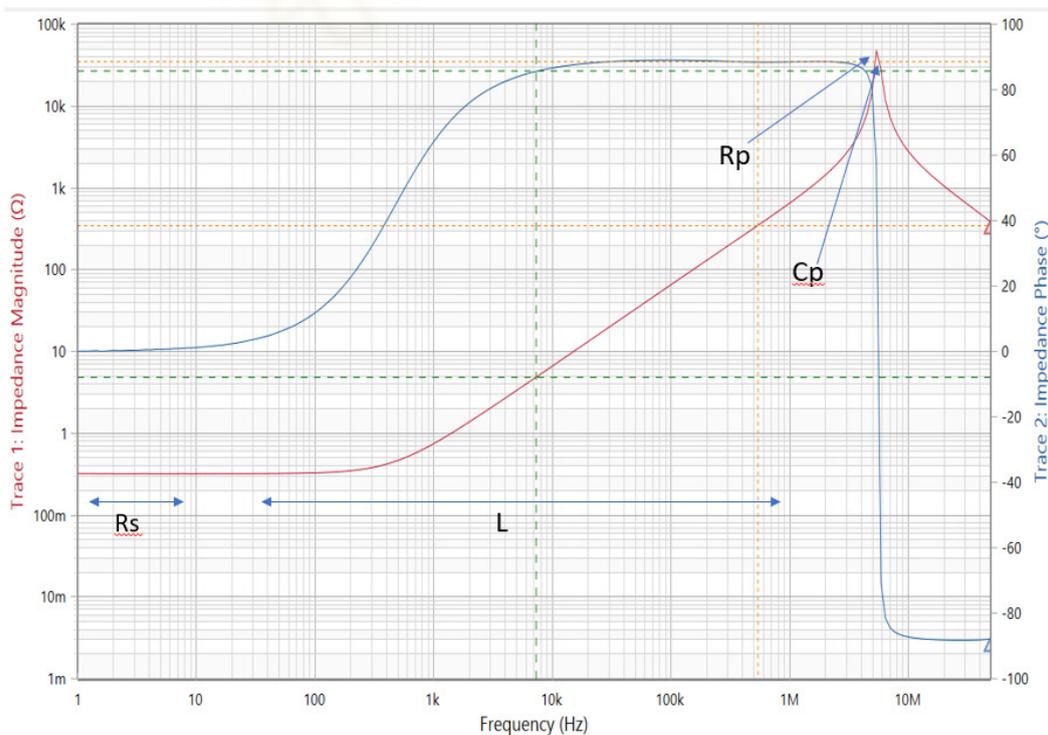


Fig. 8 Gráfica de magnitud y fase de impedancia de un inductor

4.2 Obtención de datos

El siguiente paso que realizaremos una vez entendido como funciona el modelo equivalente de inductor presentado, será la obtención de datos experimentales que nos servirán para obtener el modelo equivalente de simulación. En este modelo, haremos uso de herramientas que nos permitan realizar un barrido en alta frecuencia. Con los datos obtenidos por el barrido de frecuencia seremos capaces de obtener el modelo que perseguimos en este punto.

4.2.1 Bode 100

Como hemos dicho previamente, el modelado de los inductores se obtiene tras hacer un barrido en alta frecuencia sobre el inductor que queremos modelar. Para ello, usaremos el dispositivo “Bode 100”. Este dispositivo se trata de un analizador de red vectorial que también funciona como analizador de respuesta en frecuencia, medidor de ganancia y fase, así como analizador de impedancia. Con el “Bode 100” seremos capaces de obtener un barrido de frecuencia en un rango desde 1Hz hasta 50MHz.



Fig. 9 Esquema conexiones Bode100 [4]

Además, “Bode 100” nos proporciona una interfaz gráfica muy intuitiva, la cual es muy fácil de entender y usar en nuestro ordenador. Otra ventaja que nos ofrece dicho dispositivo es la posibilidad de exportar los resultados del barrido de frecuencia en “.csv”, lo cual usaremos para introducir los datos que obtengamos en nuestra aplicación de modelado de inductores de potencia de alta frecuencia.

4.2.2 Obtención de datos mediante Bode100

Cuando queramos obtener una medición usando el “Bode 100”, lo primero de todo será determinar qué es lo que queremos medir, ya que en función de nuestras necesidades optaremos entre el análisis de red vectorial o el análisis de impedancia. En nuestro caso, seleccionaremos el análisis de impedancia. A continuación, nos darán a elegir entre diferentes tipos de medidas. Nosotros optaremos por la mediación de “*Impedance Adapter*”.

Una vez seleccionado nuestro tipo de medida, nos dispondremos a ajustar los parámetros que nos ofrece la interfaz gráfica de “Bode 100” para que podamos obtener las mediciones que necesitamos. Estos parámetros serán el rango del barrido de frecuencia, el modo de barrido, el número de puntos que tendrá la gráfica, el nivel y el ancho de banda del receptor. Los valores de los parámetros serán los siguientes:



Fig. 10 Parámetros Configuración Barrido de Frecuencia Bode100

El barrido de frecuencia comenzará en 1Hz ya que para obtener la resistencia del bobinado necesitamos estar lo más cerca posible de DC. Por otro lado, el barrido acabará en 50MHz ya que no sabemos dónde podría estar la frecuencia de resonancia del inductor medido. Una vez ajustados los valores de los parámetros, podremos comenzar con la calibración de rango completo del “Bode 100”, la cual es necesaria para que no haya ningún tipo de error no deseado en nuestra medida y así obtener la mayor precisión posible.

Finalmente, después de realizar la calibración, sólo faltará definir en que formato queremos obtener nuestras mediciones de impedancia, las cuales serán magnitud y fase. Cuando ya tengamos la gráfica de magnitud y fase de impedancia de nuestro inductor de potencia en el rango de frecuencia 1Hz-50MHz podremos importar los resultados en formato “.csv”, el cual usaremos en nuestra aplicación de modelado en el entorno de trabajo Matlab.

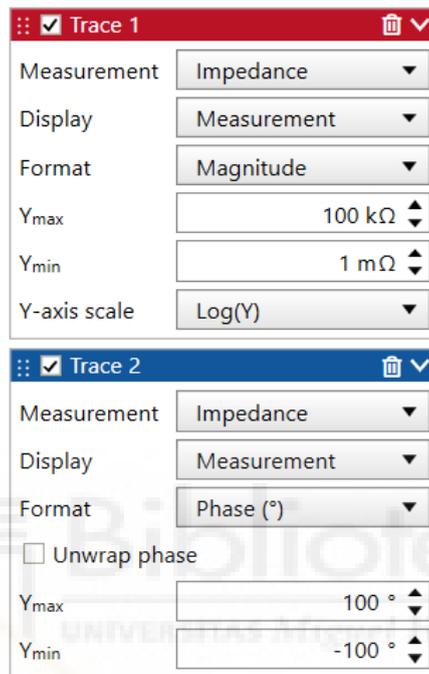


Fig. 11 Configuración Parámetros a medir en Bode100

Una vez introducidos los parámetros de impedancia a medir, en este caso la magnitud y fase de la impedancia, solo nos quedaría iniciar la medición para que el Bode100 realice el barrido de frecuencia que le hemos indicado anteriormente. Al finalizar la medición, Bode100 nos ofrecerá una gráfica como la que vemos en la figura 11, donde nos muestra la magnitud y fase de la impedancia del inductor medido. Con esta gráfica, seremos capaces de extraer estos datos en archivo “.csv” para su posterior uso en nuestra aplicación.

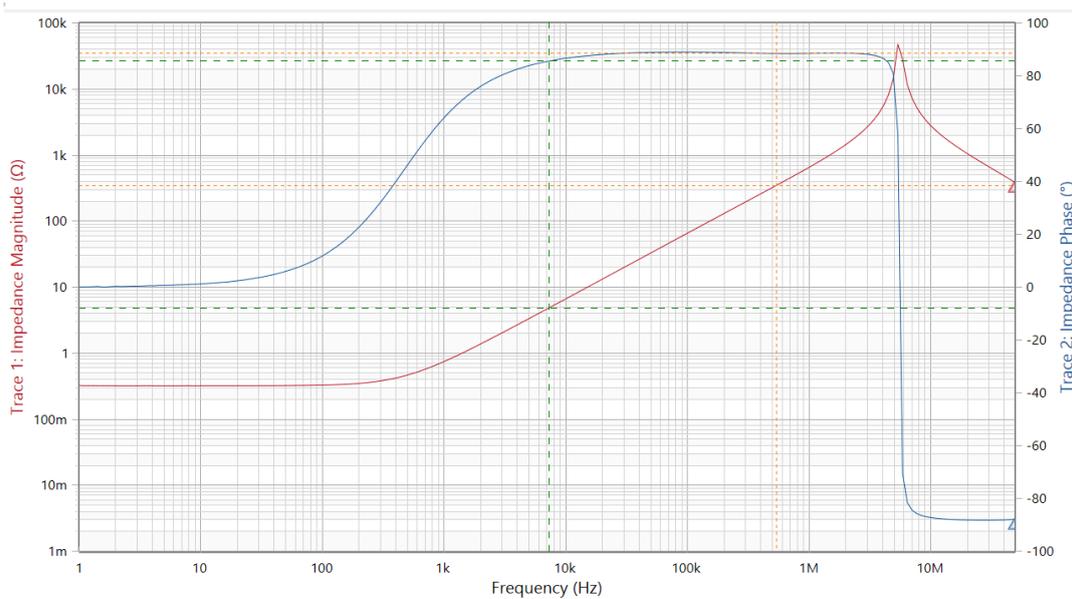


Fig. 12 Barrido en frecuencia de la Impedancia de un Inductor

4.3 Cálculo de componentes del modelo equivalente

Como se explicó en el punto 4.1, debemos obtener los valores del modelo del circuito equivalente del inductor. Esto quiere decir que debemos obtener los valores de la resistencia de bobinado, R_s , la inductancia, L , las pérdidas magnéticas, R_p , y la capacidad parásita C_p .

Primero de todo, comenzaremos obteniendo la resistencia de bobinado, R_s . Para ello tendremos que fijarnos en el primer valor de magnitud de impedancia ya que está asociado a la frecuencia de comienzo en 1Hz. Aunque R_s se mide en DC, con una frecuencia tan baja como 1Hz, la inductancia y el efecto “*Skin*” apenas perturban el valor que obtendríamos con una medición en DC.

A continuación, vamos a calcular el valor de R_p , el cual representa las pérdidas magnéticas del inductor. Para ello, tenemos que encontrar la frecuencia de resonancia del inductor y su magnitud de impedancia correspondiente será el valor de R_p . Esta frecuencia corresponde al valor máximo de impedancia que se obtiene al realizar la medición con “Bode100”, además, la fase correspondiente a dicha frecuencia es cero ya que esta frecuencia tiene una impedancia puramente resistiva.

Por otro lado, procederemos a calcular la inductancia, L , del inductor. Para ello debemos tener en cuenta el rango de frecuencia en el que la inductancia tiene un valor predominante sobre el resto de los componentes y nos permite obtener un valor fiable. Este rango comprende entre las decenas de hercios, debido a la influencia de la resistencia del bobinado, y el megahercio. Este último valor es variable ya que depende del valor de fase correspondiente a la frecuencia y esto se debe a que mientras la inductancia es predominante, la fase se mantiene alrededor de 90° . Por la tanto, cuando este valor comienza a decaer quiere decir que el efecto de la inductancia comienza a desaparecer. Una vez conocemos el rango de frecuencia, podemos aplicar la fórmula (2), donde X_L es la reactancia inductiva, la cual es directamente proporcional a la frecuencia de trabajo, para así obtener la inductancia asociada a cada frecuencia y, finalmente haremos una media entre todos los valores para tener una inductancia lo más fiable posible.

$$L = \frac{X_L}{2 * \pi * f_o} \quad (2)$$

Finalmente, sólo nos queda calcular la capacitancia parásita, C_p . Para ello, no podremos medir directamente el valor de la capacitancia por lo que tendremos que desarrollar matemáticamente ciertas ecuaciones para obtener dicho valor. Para ello, sabemos que la admitancia de nuestro inductor es:

$$Y = \frac{1}{R_p} + \frac{R_s}{R_s^2 + \omega^2 L^2} + j\omega * (C_p - \frac{L}{R_s^2 + \omega^2 L^2}) \quad (3)$$

Si observamos la parte imaginaria, vemos que existe una frecuencia, frecuencia de resonancia, que fuerza a la parte imaginaria a ser cero. Para esta frecuencia cero, ω_o , respectivamente, f_o , la ecuación queda de la siguiente forma:

$$Im\{Y\} = 0; \quad C_p = \frac{L}{R_s^2 + (2\pi f_o)^2 L^2} \quad (4)$$

Como ya hemos explicado anteriormente, R_s , tiene influencia a frecuencias bajas y sabemos que la frecuencia de resonancia tiene un valor muchísimo mayor por lo que podemos retirar a R_s de la ecuación y quedaría de la siguiente manera:

$$Im\{Y\} = 0; \quad C_p = \frac{L}{(2\pi f_o)^2 L^2} \quad (5)$$

Finalmente, con esta fórmula seremos capaces de obtener la capacitancia parásita del inductor sustituyendo los valores previamente ya obtenidos.

4.4 Implementación en App Designer

En este punto, decidimos hacer uso de App Designer, una herramienta del software matemático MATLAB, ya que nos permitía realizar el desarrollo de aplicaciones en un entorno de desarrollo interactivo. Optamos por el uso de esta herramienta ya que MATLAB es una herramienta usada por la mayoría de las universidades en las carreras de ingeniería y durante toda la carrera hemos obtenido los conocimientos necesarios para poder hacer uso de esta.

4.4.1 App Designer

App Designer es un entorno de desarrollo para diseñar y programar aplicaciones de todo tipo. Esta herramienta tiene un vínculo entre el código y las vistas del diseño, esto quiere decir que las variaciones que se realizan en un lado afectan directamente al otro.

Como podemos ver en la figura 12, la interfaz gráfica de App Designer es muy intuitiva y fácil de entender. En el centro, veríamos un espacio el cual sería nuestra aplicación a simple vista, es decir, en ese cuadrado podremos añadir diferentes elementos gráficos para que la aplicación sea más intuitiva para el usuario. En la parte izquierda tenemos distintos elementos que podremos añadir a nuestra aplicación para hacerla más interactiva y vistosa como pueden ser tablas, gráficas, imágenes... En la parte derecha vemos la opción de editar los elementos que hemos añadido a nuestra aplicación. Con esta opción podremos editar las propiedades gráficas de los componentes como puede ser la fuente de la letra, el alineado y demás funciones.

Hasta ahora sólo hemos explicado lo que podría ser la interfaz que vería el usuario, pero aún faltaría saber dónde podríamos programar el código de nuestra aplicación. Para ello, en la parte superior derecha tenemos la opción de “*Design View*” y “*Code View*”. Como bien se puede entender, “*Design View*” sería el entorno de trabajo donde configuraríamos la apariencia de nuestra aplicación mientras que “*Code View*” sería el entorno de trabajo donde programaríamos el código de nuestra aplicación.

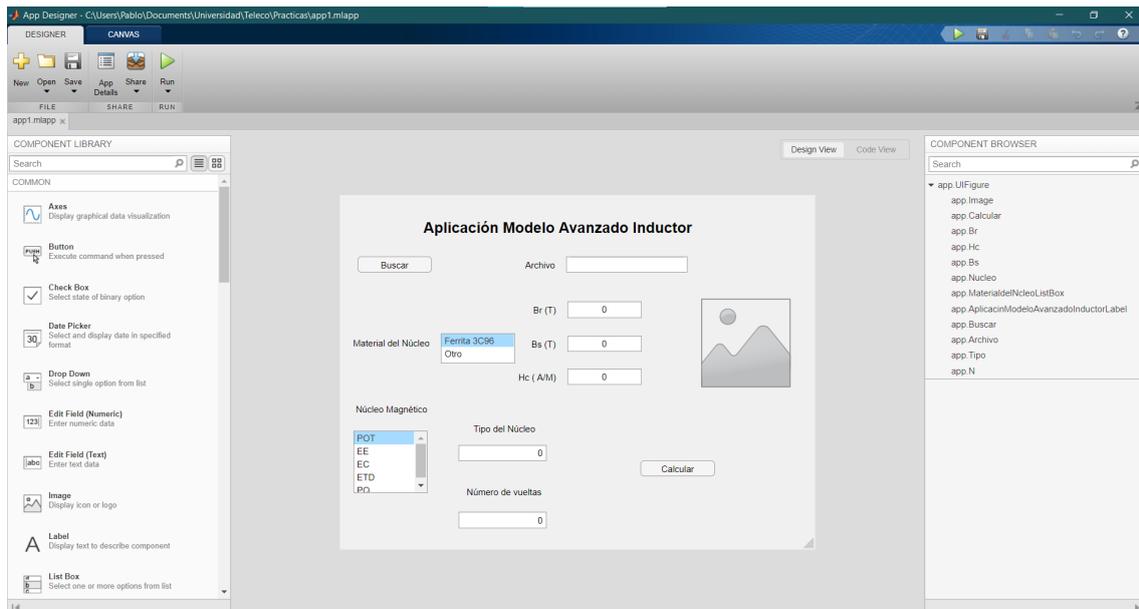


Fig. 13 Entorno de trabajo App Designer "Design View"

Si seleccionamos la opción de “Code View”, el entorno de trabajo cambiaría y se vería como en la imagen 13. En la parte izquierda, tendríamos la opción de crear y configurar funciones “callbacks”, esta una de las mayores ventajas que nos ofrece App Designer ya que nos da la posibilidad de asignar funciones a ciertos elementos como serían los botones. Estas funciones se denominan “callbacks” y con ellas podemos asignar una serie de acciones a realizar cuando, en el caso del botón, lo presionamos. Un ejemplo sería poner la opción de escribir dos números y un botón al que le haremos una función “callback” para que cuando presionen el botón, automáticamente este hará la suma de los números escritos anteriormente. En la parte central estaría el entorno donde podríamos programar y compilar nuestro código y en la parte superior tendríamos la opción de “Run” para compilar y ejecutar nuestra aplicación.

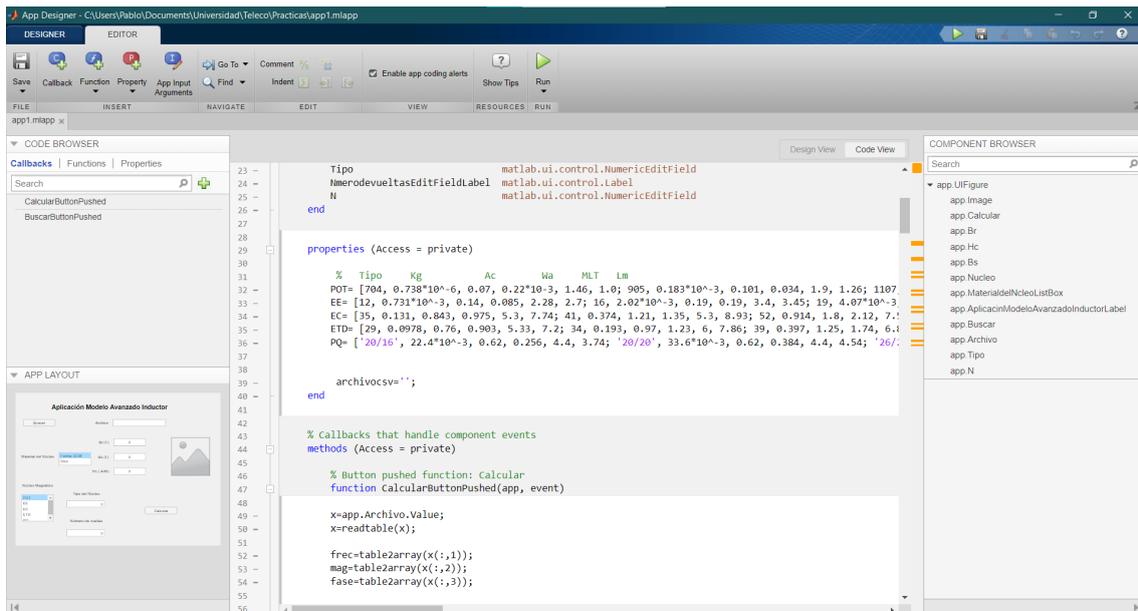


Fig. 14 Entorno de trabajo App Designer "Code View"

4.4.2 Interfaz gráfica del modelo equivalente del inductor de potencia

Una vez explicado cómo funciona App Designer y los recursos que nos ofrece para realizar una aplicación. Procederemos a realizar nuestra interfaz gráfica de nuestro modelo equivalente del inductor de potencia. Para ello, deberemos tener en cuenta qué parámetros debemos pedir al usuario, qué componentes nos harán falta para tener una aplicación completa y qué queremos que muestre nuestra interfaz, una vez implementado el código.

Primero de todo comenzaremos por elegir qué parámetros debemos pedir al usuario para poder modelar nuestro inductor de potencia. Como hemos explicado en el punto 4.2, el usuario realizará una medición con el Bode100 con el que obtendrá un archivo .CSV. Para ello generaremos una función "callback" llamada "Buscar". Este archivo será el que la aplicación requerirá para poder realizar el cálculo. Esto lo haremos con un botón el cual tendrá la función de obtener archivos y otro que mostrará la dirección del archivo obtenido como vemos en la figura 14 y cómo se mostrará una vez seleccionado el archivo.

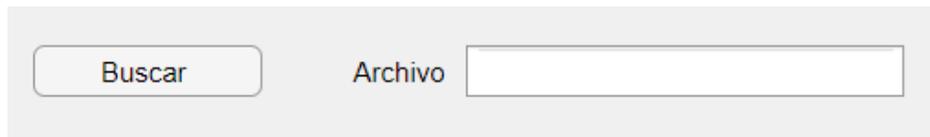


Fig. 15 Figura del buscador de la aplicación

Para obtener el archivo haremos uso de la función “*uigetfile*” la cual nos permite obtener el archivo deseado y la ubicación del archivo en nuestro ordenador. El código de “Buscar” sería el siguiente:

```
function Buscar(app, event)
    [app.archivocsv, direccion] = uigetfile({'*.csv'});
    dircompleta=strcat(direccion,app.archivocsv);
    app.Archivo.Value=dircompleta;
end
```

El resultado después de implementar el código y buscar y seleccionar un archivo sería el siguiente:



Fig. 16 Figura del buscador de la aplicación

Una vez tenemos el archivo .CSV, creamos una función “*callback*” llamada “Calcular” que calculará el valor numérico los componentes del modelo equivalente del inductor de potencia. Para ello, deberemos implementar un código en función a lo explicado en el punto 4.3. El código resultante sería el siguiente:

```
function Calcular(app, event)
    x=app.Archivo.Value;
    x=readtable(x);

    frec=table2array(x(:,1));
    mag=table2array(x(:,2));
    fase=table2array(x(:,3));

    %Cálculo Rs
    r=find(frec<10);
    frsmax=max(r);
    Rs=mean(mag(1:frsmax));
    app.Componentes.Data(1,1)=Rs;
```

```

%Cálculo Rp
Rp=max(mag(:,1));
app.Componentes.Data(1,2)=Rp;

%Cálculo L
Lfase=find(fase(:,1)>88.5);
Lfasemin=Lfase(1);
Lfasemax=max(Lfase);
Zl=mag(Lfasemin:Lfasemax,1);
Fs=frec(Lfasemin:Lfasemax,1);
Lmatr=(1/(2*pi))*(Zl./Fs);
L=mean(Lmatr);
app.Componentes.Data(1,3)=L;

%Cálculo Cp
%Cálculo frec fase=0
Neg=find(fase(:,1)<0);
x1=frec((Neg(1)-1),1);
y1=fase((Neg(1)-1),1);
x2=frec((Neg(1)),1);
y2=fase((Neg(1)),1);
Pend=(y1-y2)/(x1-x2);
f0=- (y1/Pend)+x1;

Cp=1/(4*pi^2*f0^2*L);
app.Componentes.Data(1,4)=Cp;

```

Cuando ya tenemos los valores de los componentes del modelo de inductor decidimos usar una tabla en la que mostraríamos los valores obtenidos, además de las gráficas de magnitud y fase para tener una comprobación rápida de que los datos usados coinciden con los medidos en Bode100. Por último, decidimos insertar una imagen del circuito equivalente del inductor para que el usuario tenga una idea de cómo es.

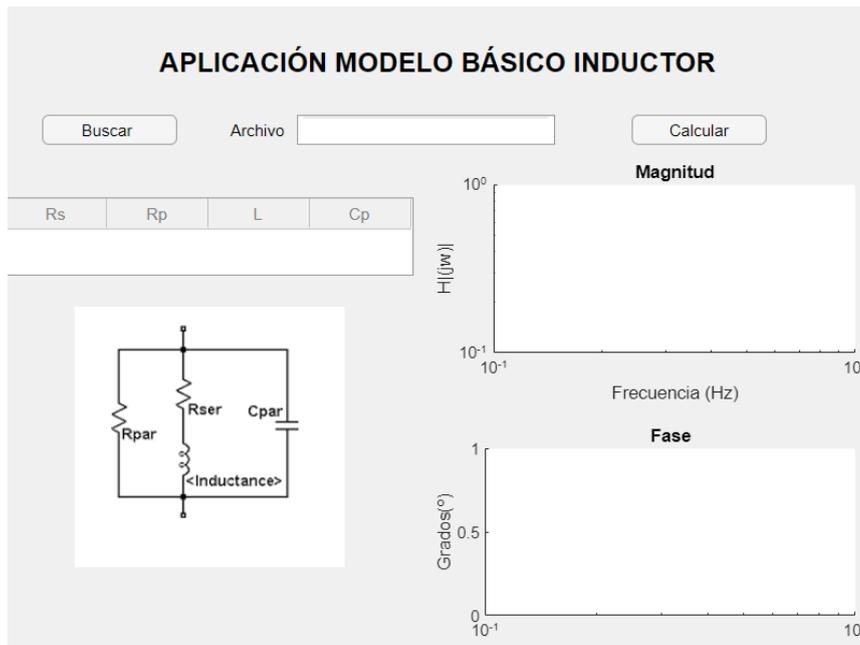


Fig. 17 Vista Inicial de la aplicación

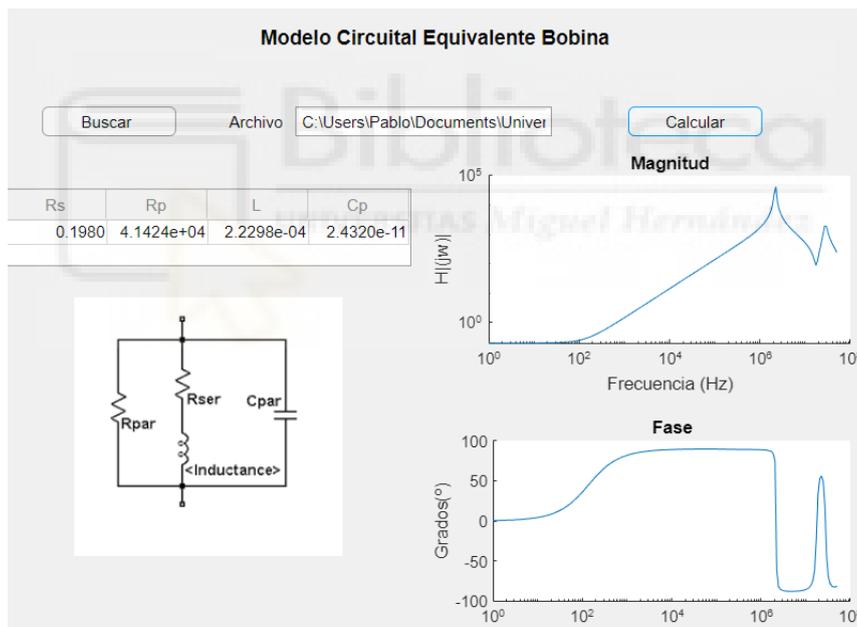


Fig. 18 Vista de la aplicación una vez introducido los datos

4.5 Obtención modelo LTspice a partir del modelo equivalente

En este momento, tenemos la capacidad de conocer el modelo equivalente de cualquier inductor de potencia que queramos. Sin embargo, no resultaría muy práctico tener que elaborar el circuito equivalente cada vez que queramos usarlo en simulaciones.

Es por ello por lo que el siguiente paso en nuestro código será conseguir que este genere un archivo con el circuito y los valores de los componentes correspondientes y así ser capaz de simularlo sin tener que perder el tiempo.

Antes de generar esta parte de código, decidimos que software de simulación utilizaríamos para realizar nuestras simulaciones. El software que decidimos usar fue LTspice, este software de simulación es gratuito, por lo tanto accesible, y fácil de usar como veremos a continuación.

4.5.1 LTspice

LTspice es una herramienta software de simulación SPICE (abreviatura de “*Simulation Program with Integrated Circuit Emphasis*”) de alto rendimiento gratuito creado por la compañía “*Analog Devices*”.

Entre las muchas ventajas que ofrece LTspice, resalta que sea accesible a todo el mundo ya que es gratuito. Por otro lado, es un simulador SPICE que ofrece un número ilimitado de nodos para construir circuitos analógicos, un editor de esquemático sencillo de entender, un visor de forma de onda y una librería de componentes pasivos como condensadores, amplificadores o inductores en los que podemos atribuirles diferentes características para que sean componentes más afines a los usados en la vida real y así no ofrecer un comportamiento ideal que, en muchas situaciones obtenemos simulaciones ideales que se distinguen mucho de la realidad.

Una de las características más importantes que nos ofrece LTspice es la librería de componentes pasivos. Ya que además de ofrecernos distintos modelos de componentes pasivos, también existe la posibilidad de importar tus propias librerías al programa y así poder realizar simulaciones con tus propios modelos y es por esto por lo que LTspice es una herramienta indispensable en nuestro trabajo para así poder realizar simulaciones con nuestros modelos equivalentes de inductor.

Como podemos ver en la imagen 18, LTspice nos ofrece un entorno de trabajo bastante minimalista donde tenemos todas las opciones en la parte superior de la interfaz

gráfica. En esta, tendríamos un acceso rápido a los componentes electrónicos más usados como son resistencias, bobinas o condensadores. Además de poder seleccionar los cables para unir los componentes y la tierra. Esta herramienta te ofrece infinidad de recursos para diseñar y simular cualquier tipo de circuito electrónico ya que, dentro de cada componente, LTspice tiene la opción de ir a la librería mencionada anteriormente para poder escoger el componente que querrías usar para el circuito.

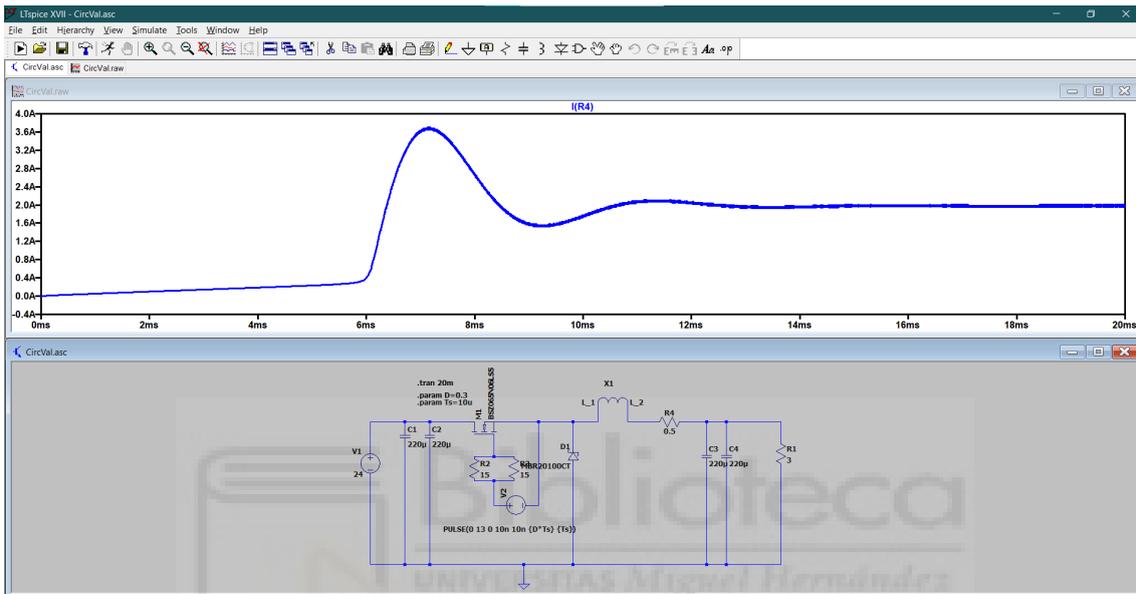


Fig. 19 Circuito del modelo equivalente en LTspice

4.5.2 Obtención modelo para usar en LTspice

Dentro de LTspice existen múltiples tipos de archivos que desempeñan diferentes funciones y representan distintos tipos de datos. Entre los diferentes archivos utilizables, nosotros usaremos dos tipos en concreto, el archivo “.asc” y el archivo “.asy”. El archivo “.asc” es un archivo de texto que representa el dibujo de tu circuito, es decir, en este archivo vienen explicadas las posiciones de cada componente junto con sus conexiones para que el software se haga a la idea de cómo está organizado nuestro circuito. Por otro lado, el archivo “.asy” es un archivo de texto que describe el símbolo que se verá en el esquemático.

Con estos dos archivos seremos capaces de crear nuestros propios modelos de inductores y podremos añadirlos a la librería de LTspice de forma permanente para poder

realizar simulaciones, aunque no conozcamos nada sobre el inductor que usamos.

Ahora, desarrollaremos una función que nos permita generar dos archivos, uno “.asc” y otro “.asy”, que representen el modelo equivalente del inductor medido a partir de los valores obtenidos con el programa explicado en el punto anterior. Esta función vendrá incluida dentro de la aplicación para que la misma aplicación genere los archivos que utilizaremos en LTspice. Como seguimos usando MATLAB, tendremos que crear dos archivos y editarlos usando los siguientes comandos:

%Crear Inductor LTSpice

```
ind=app.archivocsv;
ind=strsplit(ind, '.'); %Como el nombre es x.csv, separa este
en x y nombre y usa como delimitador el punto

indref=sprintf("%s",string(ind(1,1))); %Ind es un conjunto de
celdas por lo que hay que convertir la primera, que es la que tiene el
nombre del inductor, en un vector

%Definición de los nombres de los ficheros

inductor_name_asc=strcat(indref, '.asc');

inductor_name_asy=strcat(indref, '.asy');

app.LTSpice.Value=inductor_name_asc;
```

Como hemos explicado anteriormente, el modelo equivalente de nuestro inductor es un circuito formado por un condensador, conectado en paralelo tanto a la resistencia como a un inductor, los cuales están en serie entre ellos. Este circuito será el que tengamos que describir en el archivo .ASC acompañado de dos pines que actuarán como las patas del inductor por donde pasaría la corriente. Para ello usaremos el siguiente código:

```
%Creación del fichero del modelo
fileID = fopen(inductor_name_asc, 'w');

fprintf(fileID, 'Version 4\r\n');
fprintf(fileID, 'SHEET 1 880 680\r\n');
fprintf(fileID, 'WIRE 256 0 256 -32\r\n');
fprintf(fileID, 'WIRE 256 0 96 0\r\n');
fprintf(fileID, 'WIRE 496 0 256 0\r\n');
fprintf(fileID, 'WIRE 256 48 256 0\r\n');
fprintf(fileID, 'WIRE 96 112 96 0\r\n');
fprintf(fileID, 'WIRE 496 128 496 0\r\n');
fprintf(fileID, 'WIRE 256 160 256 128\r\n');
fprintf(fileID, 'WIRE 96 288 96 192\r\n');
```

```

fprintf(fileID, 'WIRE 256 288 256 240\r\n');
fprintf(fileID, 'WIRE 256 288 96 288\r\n');
fprintf(fileID, 'WIRE 496 288 496 192\r\n');
fprintf(fileID, 'WIRE 496 288 256 288\r\n');
fprintf(fileID, 'WIRE 256 320 256 288\r\n');
fprintf(fileID, 'FLAG 256 -32 L_1\r\n');
fprintf(fileID, 'IOPIN 256 -32 BiDir\r\n');
fprintf(fileID, 'FLAG 256 320 L_2\r\n');
fprintf(fileID, 'IOPIN 256 320 BiDir\r\n');
fprintf(fileID, 'SYMBOL res 80 96 R0\r\n');
fprintf(fileID, 'SYMATTR InstName R1\r\n');
fprintf(fileID, 'SYMATTR Value {Rpar}\r\n');
fprintf(fileID, 'SYMBOL res 240 32 R0\r\n');
fprintf(fileID, 'SYMATTR InstName R2\r\n');
fprintf(fileID, 'SYMATTR Value {Rser}\r\n');
fprintf(fileID, 'SYMBOL ind 240 144 R0\r\n');
fprintf(fileID, 'SYMATTR InstName L1\r\n');
fprintf(fileID, 'SYMATTR Value {Inductance}\r\n');
fprintf(fileID, 'SYMBOL cap 480 128 R0\r\n');
fprintf(fileID, 'SYMATTR InstName C1\r\n');
fprintf(fileID, 'SYMATTR Value {Cpar}\r\n');
fprintf(fileID, 'TEXT -500 144 Left 2 !.param
Rpar=%E\\n.param          Rser=%E\\n.param          Inductance=%E\\n.param
Cpar=%E\r\n', Rp, Rs, L, Cp);
fclose(fileID);

```

Con este código tendríamos un archivo “.asc” que, si lo abrimos en LTSpice, veríamos el circuito de nuestro modelo equivalente y los valores de los componentes que lo forman:

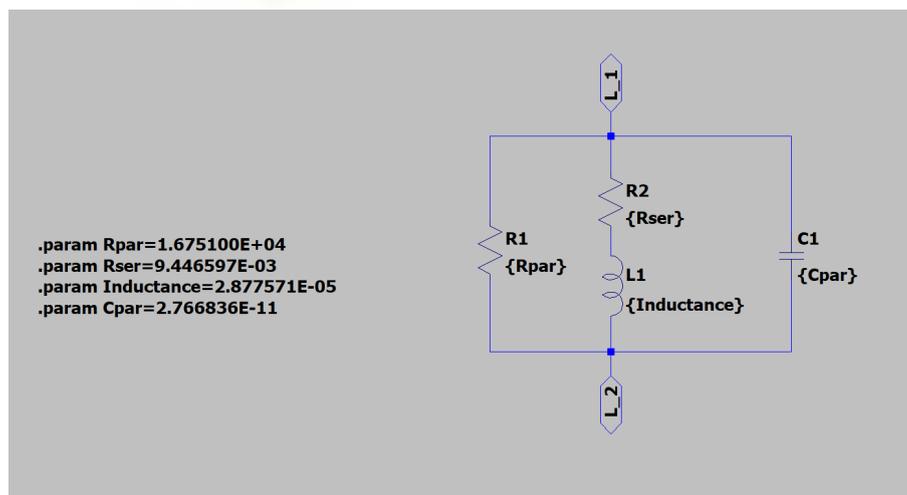


Fig. 20 Circuito del modelo equivalente en LTSpice

Una vez tenemos definido el código del archivo “.asc”, pararemos a definir el código del “.asy”. Recordemos que el archivo “.asy” actúa como el símbolo de nuestro

inductor. Es decir, en el “. asy” definiremos un inductor, el cual estará asociado al “. asy” y que nos simplificará el circuito equivalente de nuestro modelo en un inductor. Esto será de mucha utilidad para futuras simulaciones usando los modelos equivalentes de inductores obtenidos. Para ello usaremos el siguiente código:

```

%Creación del modelo del símbolo
fileID2 = fopen(inductor_name_asy,'w');

fprintf(fileID2,'Version 4\r\n');
fprintf(fileID2,'SymbolType BLOCK\r\n');
fprintf(fileID2,'ARC Normal -80 1 -47 33 -43 17 -85 15\r\n');
fprintf(fileID2,'ARC Normal -47 1 -16 33 -10 16 -55 16\r\n');
fprintf(fileID2,'ARC Normal -16 1 17 33 22 15 -25 18\r\n');
fprintf(fileID2,'WINDOW 0 -44 -33 Bottom 2\r\n');
fprintf(fileID2,'PIN -80 16 RIGHT 8\r\n');
fprintf(fileID2,'PINATTR PinName L_1\r\n');
fprintf(fileID2,'PINATTR SpiceOrder 1\r\n');
fprintf(fileID2,'PIN 16 16 LEFT 8\r\n');
fprintf(fileID2,'PINATTR PinName L_2\r\n');
fprintf(fileID2,'PINATTR SpiceOrder 2\r\n');
fclose(fileID2);

```

Con este código tendríamos un archivo “.asc” que, si lo abrimos en LTspice, veríamos el circuito de nuestro modelo equivalente y los valores de los componentes que lo forman:

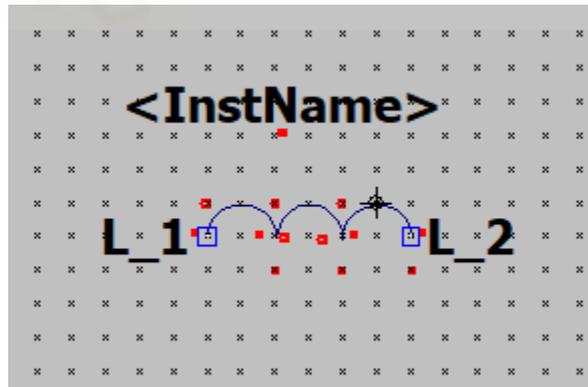


Fig. 21 Símbolo del modelo equivalente

Ahora que ya se ha explicado tanto la teoría en la que se basa el modelado como el desarrollo de la aplicación para generar el archivo que se usará en LTspice. Tan solo faltará validar la semejanza del modelo equivalente con un inductor real. Esto lo veremos en el punto 7, donde realizaremos diferentes pruebas para validar el modelo simulado.

5. MODELADO DE INDUCTOR AVANZADO

Con el modelo equivalente obtenido en la sección anterior, conseguimos obtener un comportamiento básico del inductor donde tenemos en cuenta una serie de factores generales que, aunque conseguir obtener unos valores simulados muy similares al real, no conseguimos tener una completa equivalencia entre el modelo equivalente y el inductor. Es por esto por lo que ahora debemos centrarnos en distintos métodos que, además de proporcionarnos unos valores similares de comportamiento general, también sea capaz de proporcionar un comportamiento ajustado cuando se producen ciertos cambios como puede ser el cambio de temperatura, cómo afecta el tipo de núcleo que tiene el inductor a ciertos cambios como acercarse a la saturación de la propia bobina y los efectos que puede producir.

En este punto, nos centraremos en crear un modelo equivalente capaz de tener un comportamiento lo más parecido al inductor cuando este está cerca de llegar a la saturación. Esto es debido a que esta situación se produce en innumerables ocasiones en circuitos de potencia como los convertidores DC-DC conmutados. La saturación del inductor se produce por un exceso de magnetización debido a una corriente elevada atraviese el inductor. Esta corriente de saturación hace que el núcleo del inductor no sea capaz de aumentar la cantidad de flujo magnético. Sabiendo el material del núcleo que forma el inductor, podemos estimar cuándo ocurrirá la saturación gracias al ciclo de histéresis explicado anteriormente.

Este efecto producido por la corriente de saturación no se suele tener en cuenta en el modelado de inductores y, por lo tanto, existen una serie de pérdidas generadas en ese momento que no se tienen en cuenta y, por lo tanto, la simulación que hacemos no es tan acorde a la realidad como nos gustaría. Por eso decidimos hacer uso de un modelo que ofrece LTSpice llamado “*The Chan Model*”.

Este modelo está basado en el comportamiento de la histéresis del núcleo del inductor y con la geometría de este. Nosotros desarrollaremos una aplicación que creará un modelo equivalente del inductor en base a los distintos parámetros en los que se apoya en “*The Chan Model*” para así poder obtener un modelo de inductor con un

comportamiento que se ajuste a la realidad.

5.1 Explicación teórica del modelo Chan

El modelo Chan es un modelo de simulación para software SPICE con el que se definía un modelo de transformador, el cual consistía en un núcleo no lineal con histéresis. Para definir el transformador, se requería especificar información sobre la geometría del núcleo y su nombre, además de información sobre la histéresis del núcleo magnético usado e información sobre el devanado del transformador [6]. En este modelo, el autor presenta una forma de modelar los núcleos de los inductores a través de funciones hiperbólicas continuas relacionadas con tres parámetros. Estos parámetros se utilizan para caracterizar el ciclo de histéresis y son la fuerza coercitiva H_c , la densidad de flujo B_r , y el punto de saturación, B_s . Estos valores suelen encontrarse publicados en catálogos de materiales magnéticos. Con estos tres parámetros tendremos definido el ciclo de histéresis del material del núcleo de nuestro inductor y, por tanto, tendremos definidas sus propiedades magnéticas.

Una vez que ya tenemos definido el ciclo de histéresis del núcleo de nuestro inductor, pasaremos a definir la geometría del núcleo de nuestro inductor para poder establecer la inductancia no lineal que simularemos en nuestros circuitos de simulación. Para definir la geometría necesitaremos tres parámetros geométricos y otro parámetro que tendrá en cuenta el número de vueltas del devanado. Para obtener estos parámetros, usaremos un método de diseño de inductores, el método kg, explicado en el apartado 5.2.2, el cual reutilizaremos para introducir parámetros obtenidos de nuestro inductor y crear el modelo equivalente.

5.2 Obtención de variables para nuestro modelo

5.2.1 Obtención variables mediante Ciclo Histéresis

Como hemos explicado anteriormente, para poder utilizar el modelo que nos ofrece LTSpice, debemos introducirle tres parámetros que definirán el ciclo de histéresis del núcleo de nuestro inductor, los cuales son la fuerza coercitiva H_c , la densidad de flujo remanente B_r , y la densidad de flujo en el punto de saturación, B_s . Ya que estos valores

se pueden encontrar en los catálogos de materiales magnéticos, lo que haremos será guardar los valores de diferentes materiales magnéticos que sean utilizados con más frecuencia para crear los núcleos de inductores.

Para este proyecto, nuestra aplicación se usará principalmente para núcleos magnéticos de ferrita. Para ello, definiremos los valores de la fuerza coercitiva H_c , la densidad de flujo remanente B_r , y el punto de saturación de la densidad de flujo, B_s , de la ferrita ya que el modelo en el que nos hemos basado para programar la aplicación nos ofrece resultados fiables con ferrita.

Por otro lado, al tener una gran variedad tanto de fabricantes como de diferentes aleaciones de ferrita, es muy difícil conseguir las tres variables necesarias para reproducir el ciclo de histéresis del núcleo, En nuestro caso, los inductores del laboratorio usados poseen un núcleo de ferrita denominado “3C96” perteneciente a “*Ferroxcube*” el cual no nos proporciona todas las variables. Sin embargo, sí que nos proporciona la gráfica B-H del material como se ve en la figura 21, tal y como se explica en el apartado de Ciclo de Histéresis del punto 3.1.1. A partir de esta imagen conseguimos obtener todos los valores necesarios para nuestra aplicación.

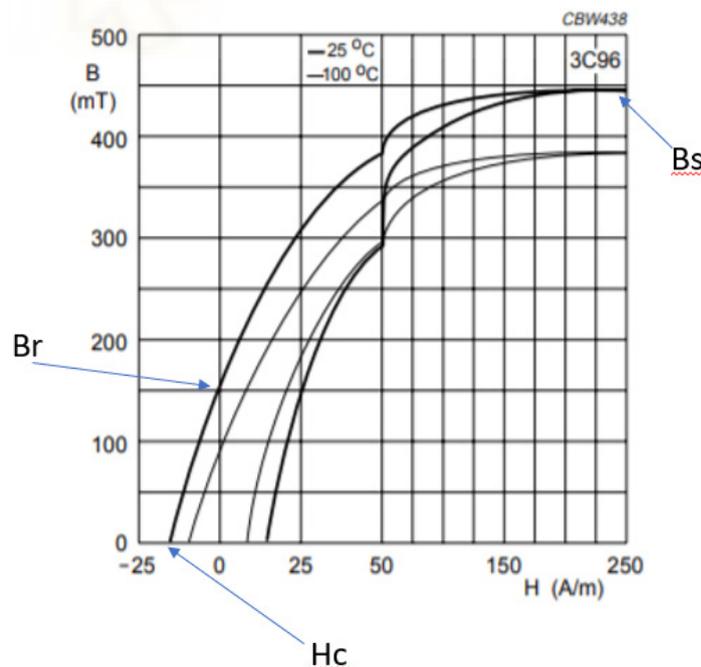


Fig. 22 Gráfica B-H del material 3C96 [21]

De aquí obtenemos los valores de las tres variables asociadas con el núcleo de ferrita 3C96 que resultarían ser:

- H_c : 17.95A/m
- B_r : 0.244T
- B_s : 0.445T

5.2.2 Obtención variables mediante método K_g

El método k_g se usa para el diseño de elementos magnéticos como los inductores. Este método está definido para conseguir que reproduzca las pérdidas de cobre de la forma más exacta posible. Normalmente, este método se usa definiendo las características que queremos que tenga el inductor diseñado. Sin embargo, nosotros realizaremos la medición de estas características sobre el inductor a modelar para así tener un modelo equivalente del inductor para poder usarlo en simulaciones.

En principio, para usar este método es necesario especificar varias variables por adelantado. Por un lado, tenemos que especificar la forma y el tipo de núcleo del inductor, la inductancia general del inductor L , la resistencia del bobinado R_s y el número de vueltas del cable. Primero explicaremos el desarrollo teórico del diseño a través del método K_g para conocer de donde viene este método y, en el procedimiento de cálculo explicaremos que hemos necesitado para completar nuestra aplicación.

Como hemos comentado anteriormente, con este método, seremos capaces de modelar inductores con hueco de aire y que tengan unas ciertas formas definidas esto se debe a que, al tener como base un modelo que se basa en el diseño de inductor, la capacidad de análisis a diferentes tipos de inductores se ve reducido. Como vemos en el anexo 10.1, seremos capaces de modelar hasta 5 tipos de inductores.

Para llegar a las ecuaciones que utilizaremos para modelar el inductor, comenzaremos definiendo el diseño del inductor presentado en la figura 22. Primero de todo, asumiremos que las pérdidas del núcleo y las pérdidas de proximidad son despreciables para que así predominen las pérdidas del bobinado como veíamos en el punto 5, las cuales estaban representadas por R_s .

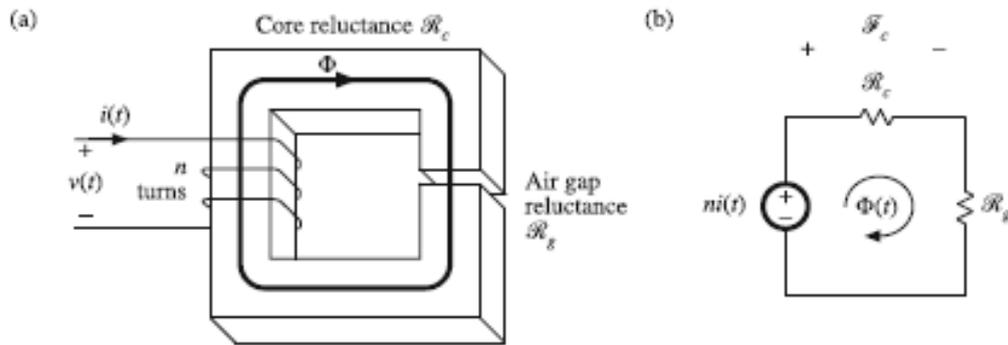


Fig. 23 Inductor a) Estructura; b) Circuito magnético equivalente [2]

Por otra parte, se asume que la geometría del inductor es la representada en la figura 22.a, y que su circuito magnético equivalente es el representado en la figura 22.b. La reluctancia del núcleo R_c y la del hueco del aire R_g son:

$$R_c = \frac{l_c}{\mu_c A_c} \quad (6)$$

$$R_g = \frac{l_g}{\mu_0 A_c} \quad (7)$$

Donde l_c es la longitud del camino magnético, A_c es el área de la sección transversal del núcleo, μ_c es la permeabilidad del núcleo y l_g es la longitud del hueco de aire. Asumimos que el núcleo y el hueco de aire ocupan la misma área de sección transversal para poder hacer uso del modelo del inductor que tenemos en la figura 22.b, la cual quedaría de la siguiente manera:

$$ni = \Phi * (R_c + R_g) \quad (8)$$

Además, $R_c \ll R_g$ y, por tanto, la ecuación 8 se puede aproximar de la siguiente forma:

$$ni = \Phi * R_g \quad (9)$$

Esto hace que el hueco del aire predomine sobre las propiedades del inductor y, por tanto, podemos identificar cuatro nuevas restricciones.

- **Densidad Máxima de flujo**

Una vez dada una corriente máxima I_{max} , queremos que la densidad de flujo del núcleo opere en el valor máximo B_{max} . Si sustituimos la ecuación (10) en la ecuación (9) obtendremos la ecuación (11):

$$\Phi = B * A_c \quad (10)$$

$$ni = B * A_c * R_g \quad (11)$$

Y suponiendo que trabajamos con una corriente máxima y una densidad de flujo máxima obtenemos:

$$n * I_{max} = B_{max} * A_c * R_g = B_{max} * \frac{l_g}{\mu_0} \quad (12)$$

Esta sería la primera restricción. El número de vueltas n y la longitud del hueco son desconocidas.

- **Inductancia**

La inductancia dada con valor L tiene que ser obtenida. La inductancia es equivalente a:

$$L = \frac{n^2}{R_g} = \frac{A_c * n^2 * \mu_0}{l_g} \quad (13)$$

Esta sería la segunda restricción. Donde la permeabilidad magnética en el vacío, μ_0 , es $4\pi * 10^{-7}$ H/m y el número de vueltas n , el área del núcleo, A_c , y la longitud del hueco, l_g , son desconocidas.

- **Área del devanado**

Como podemos ver en la figura 23, el devanado debe encajar sobre la ventana, por ejemplo, el agujero en el centro del núcleo: El área de sección transversal del conductor es A_w . Entonces, si el devanado tiene n vueltas, el área del conductor en la ventana será:

$$n * A_w \quad (14)$$

Si el núcleo tiene una ventana de área W_A , podemos expresar el área disponible para el devanado como:

$$K_u * W_A \quad (15)$$

Donde K_u es el factor de utilización de la ventana o factor de relleno. Por tanto, la tercera restricción se puede definir como:

$$K_u * W_A \gg n * A_W \quad (16)$$

El factor de relleno, K_u , es la fracción del área de la ventana del núcleo que está relleno de cobre. Además, K_u debe comprender entre cero y uno. Normalmente, los valores típicos para el factor de relleno son: 0.5 para un inductor de bajo voltaje, 0.25 a 0.3 para un transformador, 0.05 a 0.2 para un transformador de alto voltaje que suministra bastantes kV. Para nuestro modelo, decidimos usar el valor de $K_u=0.4$. [1]

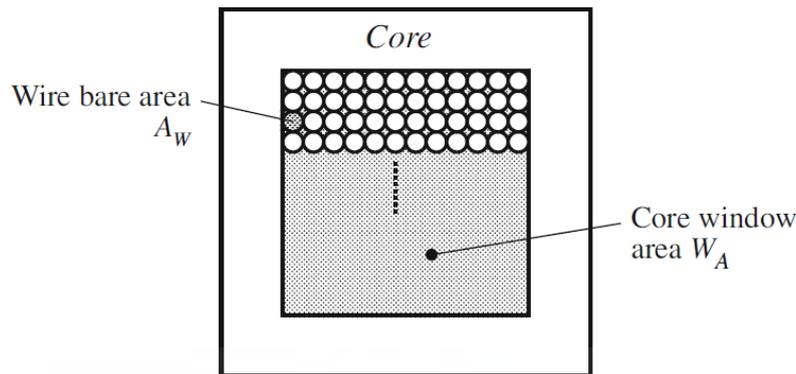


Fig. 24 Representación donde el devanado debe encajar con el área del núcleo magnético [2]

- **Resistencia del bobinado**

La resistencia del bobinado es

$$R = \rho * \frac{l_b}{A_W} \quad (17)$$

Donde ρ es la resistividad del material conductor, l_b es la longitud del cable y A_W es el área de sección transversal. La resistividad del cobre a temperatura ambiente es $1.724 * 10^{-6} \Omega/\text{cm}$. Por otro lado, la longitud del cable teniendo el número de vueltas se puede expresar de la siguiente manera:

$$l_b = n * (MLT) \quad (18)$$

Donde MLT es longitud media por vuelta (mean-length-per-turn) del devanado. La longitud media por vuelta es una función geométrica del núcleo. Y, sustituyendo la ecuación (18) en (17) obtenemos:

$$R = \rho * \frac{n * (MLT)}{A_W} \quad (19)$$

Esta sería la cuarta restricción.

- **Constante geométrica del núcleo K_g**

Las cuatro restricciones tienen en cuenta las funciones geométricas del núcleo A_c , W_A y MLT , los valores I_{max} , B_{max} , L , K_u , R_s , ρ y μ_0 que ya sabemos y los valores n , l_g y A_w los cuales no sabemos. Eliminando los valores que no sabemos llegamos a la siguiente ecuación:

$$\frac{A_c^2 * W_A}{MLT} \geq \frac{\rho * L^2 * I_{max}^2}{B_{max}^2 * R * K_u} \quad (20)$$

Mientras que los valores que están al lado derecho de la ecuación son valores especificados o previamente conocidos, los valores situados en la parte izquierda de la ecuación tan solo hacen referencia a funciones geométricas del núcleo. Es por ello que se necesita elegir un núcleo que satisfaga la ecuación anterior.

Para simplificar este proceso, se creo un nuevo parámetro denominado K_g . K_g se define como la constante geométrica del núcleo. Este parámetro describe el tamaño eléctrico efectivo de núcleos magnéticos en aplicaciones donde las pérdidas del devanado y la densidad máxima de flujo se pueden definir. Además existen una serie de tablas que podemos encontrar en el Anexo 10.1 con la lista de valores asociados a K_g para distintos estándares de núcleos de ferrita.

$$K_g = \frac{A_c^2 * W_A}{MLT} \quad (21)$$

- **Procedimiento de cálculo**

A continuación, haremos uso de las ecuaciones obtenidas previamente para llegar a modelar los inductores que queramos medir. Este procedimiento no llega a tener en cuenta todas las variables posibles que pueden afectar al comportamiento de un inductor como el aumento de temperatura, el aislamiento requerido para el devanado... Sin embargo, nos puede dar un resultado preciso que podremos utilizar para futuras simulaciones.

Primero de todo, partimos de la base de que ya conocemos los valores de inductancia, resistencia del devanado, el número de vueltas, n , y el valor de K_g junto con el tipo de núcleo del inductor se obtienen los valores de A_c y l_m a partir de las tablas que se encuentran en el anexo 10.1. A partir de aquí, tan solo necesitaríamos el valor de l_g .

Para ello, usaremos la ecuación que resulta de simplificar la ecuación (13) y obtendremos la siguiente ecuación:

$$l_g = \frac{A_c * n^2 * \mu_0}{L} * 10^{-4} (m) \quad (22)$$

Con todos estos cálculos, junto con los valores de histéresis explicados anteriormente, ya tendríamos los valores necesarios que nos pide LTSpice para usar el “modelo *Chan*” con el que podremos realizar simulaciones con el modelo equivalente obtenido.

5.3 Obtención datos experimentales

Como hemos explicado anteriormente, este modelo nos exige ciertos valores de los inductores a modelar. Estos datos los obtendremos del inductor de forma experimental en el laboratorio y siempre será necesario obtenerlos. Las variables que debemos medir del inductor serán la inductancia general del inductor L y la resistencia del bobinado R_s .

Para la obtención de la resistencia del bobinado y la inductancia general del inductor podemos utilizar el dispositivo Bode100, explicado en el punto 4.2, ya que con el barrido en alta frecuencia no es necesario realizar otras medidas. Respecto a la forma de núcleo y su tipo, al ser todo valores estándares, normalmente nos vendrán indicado por los fabricantes y finalmente, el número de vueltas será el que tenga el inductor enrollado sobre el núcleo.

5.4 Implementación en App Designer

Una vez conocido el proceso matemático de cómo obtener los valores que requiere el modelo “*Chan*” de LTSpice, comenzaremos a explicar cómo hemos programado nuestra aplicación en App Designer para que se quede una aplicación fácil de entender para el usuario y que nos cree los modelos de los futuros inductores que queramos utilizar.

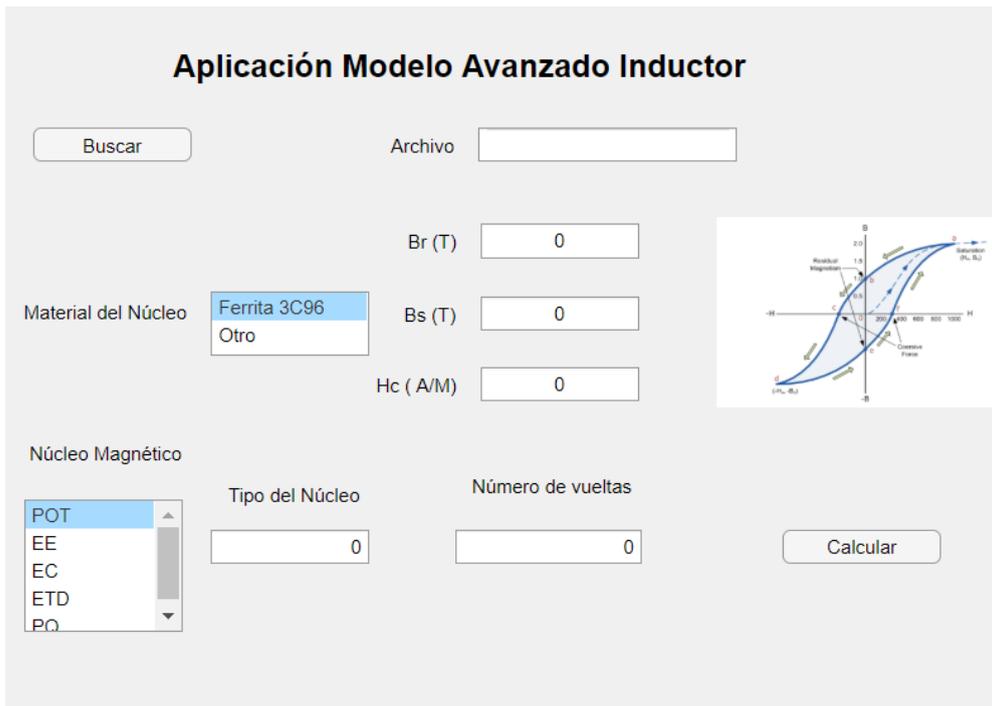


Fig. 25 Vista aplicación modelo avanzado

Esta aplicación estará dividida en tres partes. Por un lado, tendremos la parte relacionada con la petición de buscar el archivo “.csv” que se necesita para calcular los valores de inductancia y resistencia del devanado. Esta parte es completamente igual a la vista en el punto 4.4.2.

La segunda parte tratará sobre el ciclo de histéresis, donde en principio tendremos unos valores de la fuerza coercitiva H_c , la densidad de flujo remanente B_r , y la densidad de flujo en su punto de saturación, B_s , donde se dará la opción de usar los valores del núcleo de ferrita 3C96 o, en caso de que no sea ese núcleo, daremos la opción al usuario de que presionando “Otro”, como vemos en la figura 26, pueda introducir unos valores de su material de ferrita.

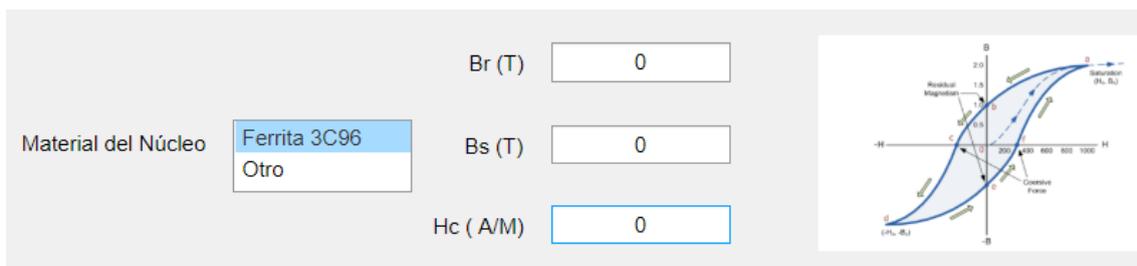


Fig. 26 Sección de la aplicación de modelado avanzado

Por otro lado, tendremos la parte que haga referencia al modelo Kg en donde pediremos al usuario introducir la forma de núcleo magnético y la referencia del tipo dentro de la tabla como vemos en la figura 27. Como ejemplo, si tenemos un núcleo “EE20”, en “Núcleo Magnético” seleccionaremos “EE” y en tipo de núcleo pondremos “20” y automáticamente, el programa ya determinará los valores relacionados con el núcleo “EE20”.

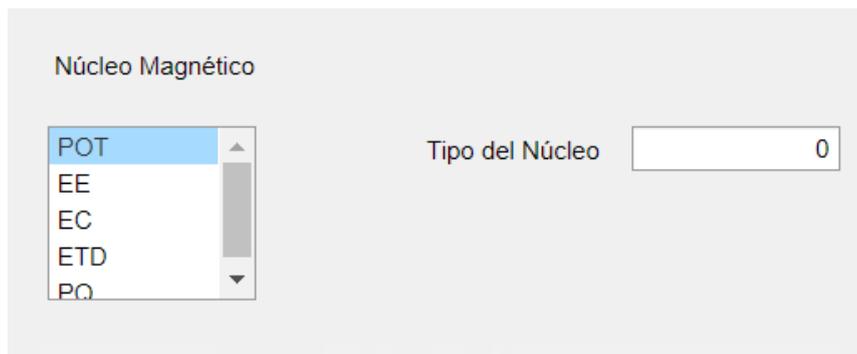


Fig. 27 Sección de la aplicación de modelado avanzado

Entonces, cuando todos los valores estén introducidos, ya estará listo para darle al botón de “Calcular”, el cual están definidos todos los pasos explicados en el punto anterior. Además, hemos introducido a mano las tablas de las formas de núcleo magnético que aparecen en el Anexo 10.1 para que, en función del tipo de núcleo escogido, la aplicación utilice los valores correspondientes. Una vez hecho todo esto, la aplicación obtendrá el valor de l_g que necesitamos y así crearemos el modelo para LTspice con el resto de variables previamente obtenidas.

```
function CalcularButtonPushed(app, event)
```

```
    x=app.Archivo.Value;
    x=readtable(x);
```

```
    frec=table2array(x(:,1));
    mag=table2array(x(:,2));
    fase=table2array(x(:,3));
```

```
    %Cálculo Rs
```

```
        r=find(frec<10);
        frsmax=max(r);
```

```
    matriz para hacer la media
```

```
    % Matriz valores de frec <10Hz
    % Último valor menos que 10 de la
```

```

Rs=mean(mag(1:frsmax)); % Media de la columna de magnitud
hasta el ultimo valor <10

%Cálculo L
Lfase=find(fase(:,1)>88.5); % Matriz con posiciones de
valores de fase >88,5º
Lfasemin=Lfase(1);          % Posiciones max y min
Lfasemax=max(Lfase);
Zl=mag(Lfasemin:Lfasemax,1); % Valores de impedancia de la
bobina desde 10Hz a ...
Fs=frec(Lfasemin:Lfasemax,1); % Valores de frecuencia
correspondientes a impedancias
Lmatr=(1/(2*pi))*(Zl./Fs); % Matriz con su inductancia
correspondiente
L=mean(Lmatr);             % Valor de L desde 10Hz

p=1.724*10^-6; % Ω/cm Resistividad del cobre a T. ambiente

Ku=0.4; %Se usa como valor estandar a la hora de diseñar bobinas

u0=4*pi*10^-7; %H/m Permeabilidad

if app.Nucleo.Value=="POT"

    r=find(app.POT(:,1)==app.Tipo.Value);

    % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
    if numel(r)>1
        kg= app.POT(r(1),2);          % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
        Ac=app.POT(r(1),3);
        Lm=app.POT(r(1),6);
    else
        kg= app.POT(r,2);             % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
        Ac=app.POT(r,3);
        Lm=app.POT(r,6);
    end

elseif app.Nucleo.Value=="EE"

    r=find(app.EE(:,1)==app.Tipo.Value);

    % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
    if numel(r)>1
        kg= app.EE(r(1),2);          % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
        Ac=app.EE(r(1),3);
        Lm=app.EE(r(1),6);
    else

```

```

        kg= app.EE(r,2);           % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
        Ac=app.EE(r,3);
        Lm=app.EE(r,6);
    end

    elseif app.Nucleo.Value=="EC"

        r=find(app.EC(:,1)==app.Tipo.Value);

        % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
        if numel(r)>1
            kg= app.EC(r(1),2);           % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
            Ac=app.EC(r(1),3);
            Lm=app.EC(r(1),6);
        else
            kg= app.EC(r,2);           % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
            Ac=app.EC(r,3);
            Lm=app.EC(r,6);
        end

    elseif app.Nucleo.Value=="ETD"

        r=find(app.ETD(:,1)==app.Tipo.Value);

        % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
        if numel(r)>1
            kg= app.ETD(r(1),2);           % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
            Ac=app.ETD(r(1),3);
            Lm=app.ETD(r(1),6);
        else
            kg= app.ETD(r,2);           % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
            Ac=app.ETD(r,3);
            Lm=app.ETD(r,6);
        end

    elseif app.Nucleo.Value=="PQ"

        r=find(app.PQ(:,1)==app.Tipo.Value);

        % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
        if numel(r)>1
            kg= app.PQ(r(1),2);           % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
            Ac=app.PQ(r(1),3);
            Lm=app.PQ(r(1),6);
        else

```

```

kg= app.PQ(r,2); % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
Ac=app.PQ(r,3);
Lm=app.PQ(r,6);
end
end

lg=u0*Ac*app.N.Value^2*10^-4/L;

if app.MaterialdelNcleoListBox.Value== "Otro"

app.Hc.Value=app.Hc.Value;
app.Bs.Value=app.Bs.Value;
app.Br.Value=app.Br.Value;

elseif app.MaterialdelNcleoListBox.Value== "Ferrita 3C96"

app.Hc.Value=17.95;
app.Br.Value=0.244;
app.Bs.Value=0.445;

end

%Conversión de cm a metros

Ac=Ac*10^-4;
Lm=Lm*10^-2;

```

5.4 Obtención del modelo en LTspice

Al igual que explicamos en el punto 4.5.2, además de tener un programa que nos permita obtener los parámetros necesarios para poder hacer simulaciones en LTspice, necesitamos que genere un archivo con las variables que pide LTspice calculadas. Es por ello por lo que también haremos un archivo “.asc” y otro “.asy” que se generarán con cada inductor equivalente que queramos obtener. Para ello, se seguirá el mismo proceso que en el punto 4.5.2 y lo que variará será el nombre definido para diferenciarlos y el código del “.asc” ya que la única diferencia que hay entre ambos modelos es que el primero usa un circuito con varios componentes electrónicos mientras que este solo usa un inductor, el cual vendrá definido por los valores previamente obtenidos. Por tanto, una vez definido el código, este sería así:

```

%Crear Inductor LTSpice
ind=app.archivocsv;
ind=strsplit(ind, '.'); %Como el nombre es x.csv, separa este en
"x" y "csv" y usa como delimitador el .

```

indref=sprintf("%s",string(ind(1,1))); %Ind es un conjunto de celdas por lo que hay que convertir la primera, que es la que tiene el nombre del inductor, en un vector

%Definición de los nombres de los ficheros

```
inductor_CHAN_name_asc=strcat(indref,'_CHAN.asc');  
inductor_CHAN_name_asy=strcat(indref,'_CHAN.asy');
```

%Creación del fichero del modelo

```
fileID = fopen(inductor_CHAN_name_asc,'w');
```

```
fprintf(fileID,'Version 4\r\n');  
fprintf(fileID,'SHEET 1 880 680\r\n');  
fprintf(fileID,'WIRE 240 208 80 208\r\n');  
fprintf(fileID,'WIRE 496 208 320 208\r\n');  
fprintf(fileID,'FLAG 80 208 L_1\r\n');  
fprintf(fileID,'IOPIN 80 208 BiDir\r\n');  
fprintf(fileID,'FLAG 496 208 L_2\r\n');  
fprintf(fileID,'IOPIN 496 208 BiDir\r\n');  
fprintf(fileID,'SYMBOL ind 224 224 R270\r\n');  
fprintf(fileID,'WINDOW 0 32 56 VTop 2\r\n');  
fprintf(fileID,'WINDOW 3 5 56 VBottom 2\r\n');  
fprintf(fileID,'SYMATTR InstName L1\r\n');  
fprintf(fileID,'SYMATTR Value Hc=%E Bs=%E Br=%E A=%E Lm=%E  
Lg=%E N=%E\r\n',app.Hc.Value,app.Bs.Value,app.Br.Value,Ac,Lm,lg,n);
```

```
fclose(fileID);
```

%Creación del modelo del símbolo

```
fileID2 = fopen(inductor_CHAN_name_asy,'w');
```

```
fprintf(fileID2,'Version 4\r\n');  
fprintf(fileID2,'SymbolType BLOCK\r\n');  
fprintf(fileID2,'ARC Normal -80 1 -47 33 -43 17 -85 15\r\n');  
fprintf(fileID2,'ARC Normal -47 1 -16 33 -10 16 -55 16\r\n');  
fprintf(fileID2,'ARC Normal -16 1 17 33 22 15 -25 18\r\n');  
fprintf(fileID2,'WINDOW 0 -44 -33 Bottom 2\r\n');  
fprintf(fileID2,'PIN -80 16 RIGHT 8\r\n');  
fprintf(fileID2,'PINATTR PinName L_1\r\n');  
fprintf(fileID2,'PINATTR SpiceOrder 1\r\n');  
fprintf(fileID2,'PIN 16 16 LEFT 8\r\n');  
fprintf(fileID2,'PINATTR PinName L_2\r\n');  
fprintf(fileID2,'PINATTR SpiceOrder 2\r\n');
```

```
fclose(fileID2);
```

```
end
```

6. VALIDACIÓN EXPERIMENTAL – Convertidor Buck

Una de las fases más importantes a la hora de realizar un modelo equivalente de cualquier tipo de componente es que el comportamiento del modelo sea lo más similar al comportamiento del componente real, en nuestro caso el inductor de potencia. Comparando los resultados obtenidos podremos determinar cómo de precisa y fiable es nuestra simulación.

Para esta fase, se ha diseñado una fuente de alimentación conmutada o convertidor DC-DC. Este tipo de circuito electrónico de potencia es capaz de convertir una tensión continua en otro nivel de continua mediante el empleo de semiconductores (transistores, por ejemplo) que trabajan en conmutación.

Dentro de las fuentes de alimentación conmutadas, tenemos tres tipologías de convertidores DC-DC conmutados. Tenemos el circuito *Buck* (Reductor), *Boost* (Elevador) y *Buck-Boost* (Reductor-Elevador). Como podemos intuir, el circuito *Buck* tiene la capacidad de reducir el voltaje de salida respecto al de entrada, el circuito *Boost* tiene la capacidad de elevar el voltaje de salida respecto al de entrada y el circuito *Buck-Boost* tiene la capacidad tanto de elevar como de reducir el voltaje de salida respecto al de entrada. Los tres tipos de convertidores usan dos interruptores, uno suele ser controlado, normalmente suele ser un MOSFET, y otro que no se puede controlar, normalmente es un diodo, para controlar el flujo de potencia de la entrada a la salida. Además, también hacen uso de un inductor y un condensador para acumular y transferir energía de la entrada a la salida.

Para poder determinar qué tipología debíamos elegir qué parámetro medir para validar nuestro modelo equivalente obtenido. De los posibles parámetros decidimos elegir la corriente que atraviesa el inductor de potencia ya que podríamos tener unos resultados que reflejen el comportamiento de la bobina.

De las distintas tipologías a elegir para la medición de la corriente del inductor, llegamos a la conclusión de que la mejor opción es utilizar el convertidor *Buck* debido a

su simplicidad y a la facilidad para medir la corriente que atraviesa el inductor, ya que esa zona coincide con el voltaje de salida y tendríamos una respuesta menos variable que si midiésemos la corriente antes de que llegara al inductor. Esto es debido a que la conmutación del convertidor no afectaría tanto en nuestra zona de medición.

6.1 Convertidor Reductor – *Buck*

Como hemos explicado anteriormente, el convertidor *Buck* consta de dos interruptores, uno controlado que es el transistor y otro no controlado que es el diodo, un inductor de potencia, un condensador y una carga., generalmente representada por una resistencia.

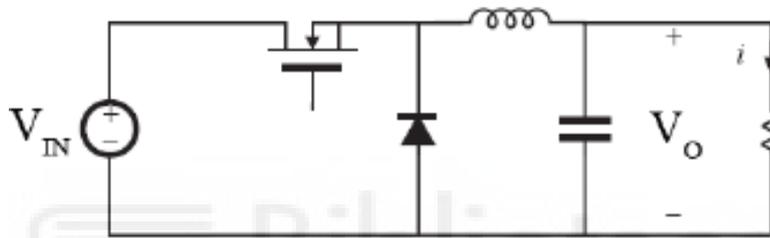


Fig. 28 Esquema circuital de un Convertidor Buck. [5]

Además, el convertidor *Buck* puede tener dos modos de operación: Modo continuo y modo discontinuo. En el modo continuo la corriente que fluye por el inductor nunca llega a ser cero mientras que el modo discontinuo se caracteriza por no conducir corriente sobre el inductor durante una parte del periodo de conmutación. Esta corriente que fluye por el inductor va variando de valor de forma periódica en función de si está “abierto” o “cerrado” el MOSFET, que funciona en región óhmica para “encendido” y en corte para “cerrado”. Esta intensidad toma una forma triangular, la cual será la que midamos como vemos en la figura 28.

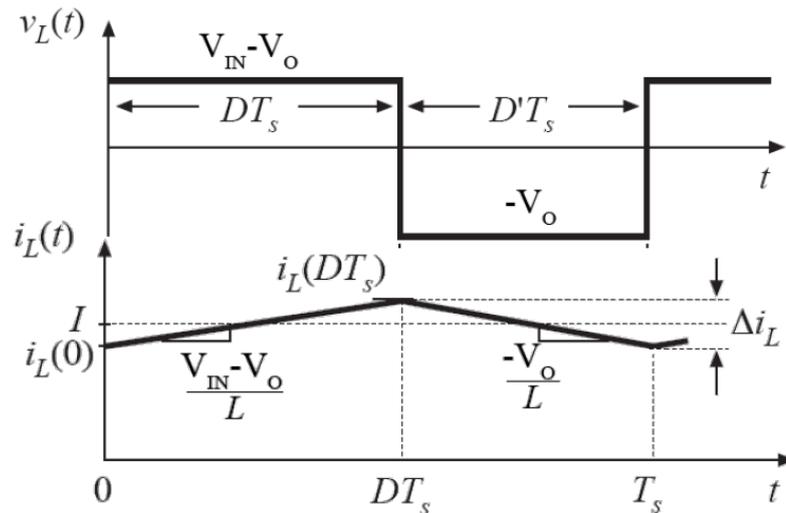


Fig. 29 Grafica representa el voltaje y la intensidad a la salida del inductor en un circuito Buck [5]

6.1.1 Componentes seleccionados para el DC-DC *Buck*

- 1x Transistor MOSFET “STP80NF06”
- 1x Diodo MUR1520
- 1x Resistencia *Shunt*
- 4x Condensadores de acoplo y desacoplo 220 μ F
- 1x Carga electrónica
- 1x *Shield* PWM
- 6x Bornas
- 2x Codos
- 2x Resistencias 15 Ω \pm 5% 1/4W

Como podemos observar, hay componentes que, teóricamente no estarán en el circuito Buck como tal pero sí que serán de gran importancia para el funcionamiento de este. Estos elementos serían:

- ***Shield* PWM:** Este *shield* PWM es un circuito que va unido junto a un PSoC como vemos en la figura 31, el cual es un circuito integrado que cuenta con un microcontrolador, una pequeña FPGA (*Field Programmable Gate Array*) y un sistema de señales mixtas. Esto nos ayudará a controlar el ciclo de trabajo de nuestro circuito gracias a una aplicación desarrollada y asociada al *Shield* [9].

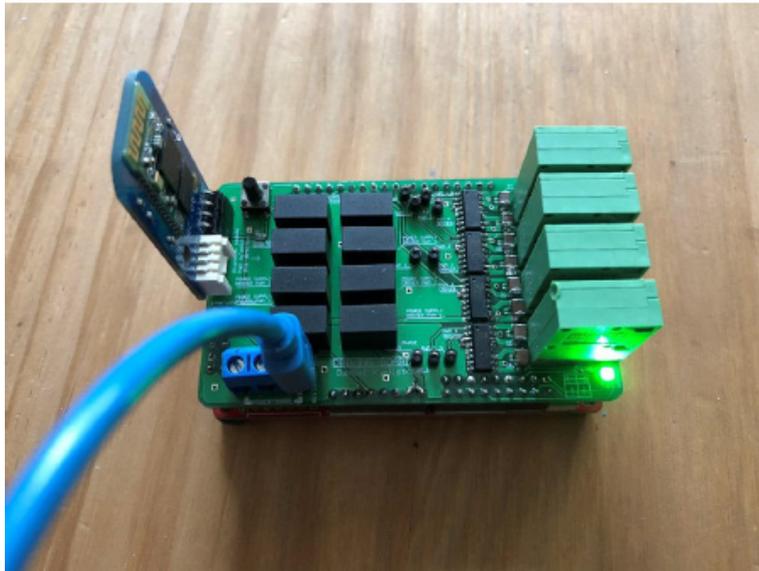


Fig. 30 Shield PWM [9]

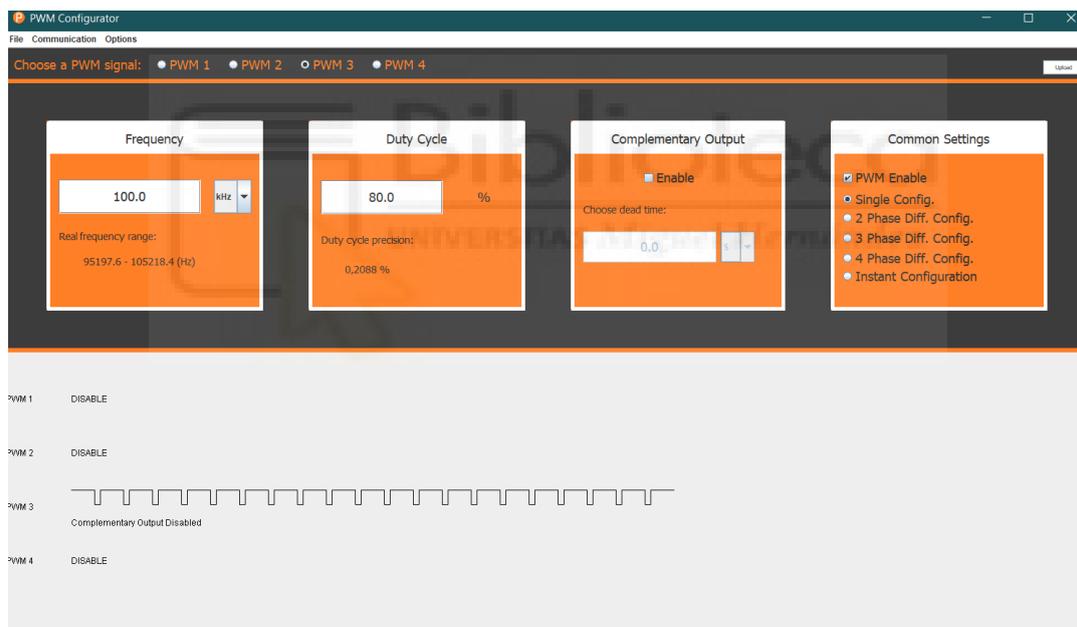


Fig. 31 Interfaz gráfica de la aplicación asociada al PWM [9]

- **Carga electrónica variable:** Como hemos explicado anteriormente, esta resistencia es un equipo electrónico, *RIGOL DL3021A*, que genera la resistencia de carga que nosotros le indiquemos.
- **Resistencias 15 Ω :** Estas dos resistencias se colocan entre el voltaje de control proveniente del PSoC y la puerta del MOSFET. Su función comprende en proporcionar una mayor seguridad al circuito evitando posibles picos de intensidad o voltaje del PSoC.

- **Cocodrilo:** En este caso, los cocodrilos estarán situados donde debería estar el inductor para así conectarlo al circuito y darnos la posibilidad de intercambiar los inductores de forma sencilla sin tener que soldar y desoldar los inductores a medir en el circuito.
- **Bornas:** En este caso, usaremos 6 bornas, de las cuales 4 de ellas servirán para conectar tanto la alimentación principal como la alimentación del voltaje de control que irá entre la puerta y el surtidor del MOSFET. Por otro lado, dejaremos las dos bornas restantes en paralelo junto con la resistencia de carga en caso de que, en el futuro se necesite conectar algún elemento más al circuito.
- **Resistencia *Shunt*:** Esta resistencia apenas presenta resistencia, valga la redundancia, al paso de corriente y tiene una forma que nos permite usar una sonda de corriente para verla representada en el osciloscopio. Esta intensidad será la que usaremos para validar la veracidad de nuestros modelos.

6.2 Diseño del circuito Buck

Ahora que conocemos los componentes del circuito *Buck* procederemos a diseñar y a dar forma al circuito mediante el programa EAGLE. EAGLE es un software de diseño electrónico que nos permitirá diseñar nuestro circuito de validación y definir los componentes que utilizaremos en nuestro circuito.

6.2.1 EAGLE

EAGLE es un software de automatización de diseño electrónico (EDA) que permite diseñar placas de circuito impreso (PCB) de forma sencilla. Con EAGLE podemos tanto realizar el esquemático donde haremos nuestro circuito *Buck* y conectaremos los distintos componentes como diseñar la PCB en la que más tarde soldaremos los componentes para tener nuestro circuito.

Como podemos ver en la imagen, el entorno de trabajo de EAGLE se divide en el esquemático y en lo que sería la placa, “*Board*”, en el entorno del esquemático vemos que es muy accesible ya que en la parte derecha tenemos todas las opciones necesarias para poder hacer nuestro circuito, desde añadir las partes de nuestro circuito a las conexiones, y distintas opciones de edición que nos facilitarán el proceso de creación del circuito.

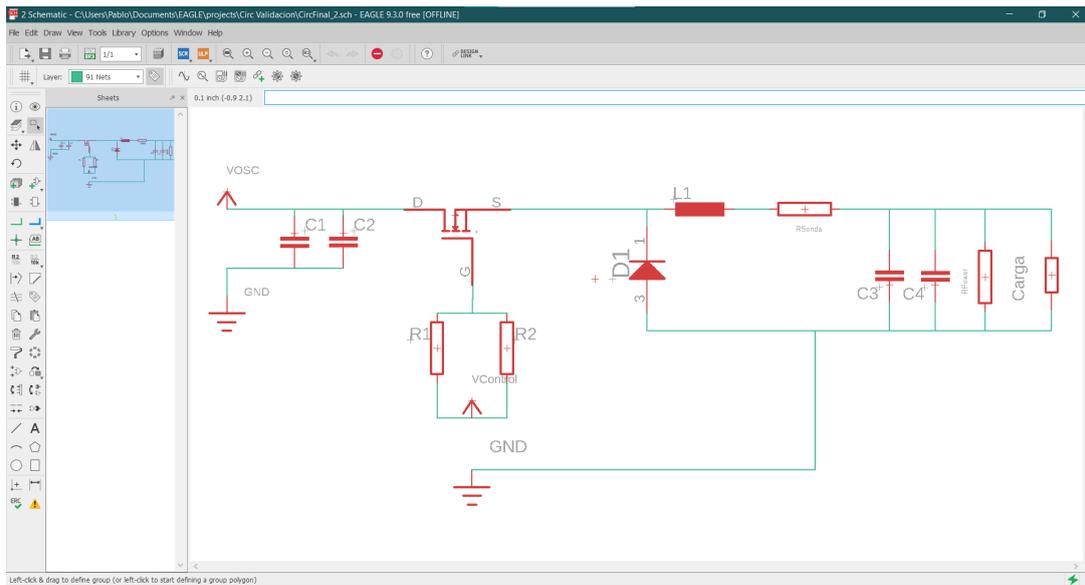


Fig. 32 Entorno de trabajo del Esquemático en EAGLE

Por otro lado, tendríamos la parte del *layout*, en esta deberemos organizar y distribuir los componentes electrónicos que usaremos en el circuito de la mejor forma posible para que entren en el tamaño del PCB disponible y no se produzcan interferencia electromagnéticas y demás efectos que pueden estropear tanto el funcionamiento del circuito como las mediciones que se hagan a posteriori. Como podemos ver en la imagen, el entorno de trabajo es similar al del esquemático con todas las opciones de edición situadas a la derecha.

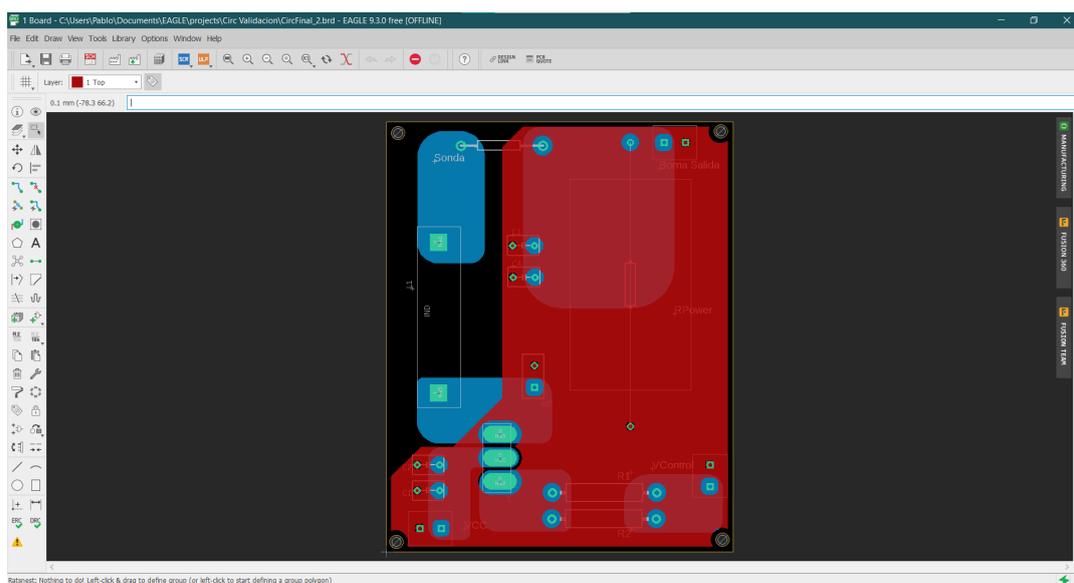


Fig. 33 Entorno de trabajo del "Board" en EAGLE.

6.2.2 Diseño del circuito

Para diseñar el esquemático de nuestro circuito, hemos seguido el modelo básico del convertido *Buck*. Por otro lado, al usar el dispositivo PWM que nos proporcionará un voltaje controlado de 13V y un rango de frecuencia desde 10KHz hasta 2.6MHz [9], a la puerta del transistor para controlarlo, pondremos dos resistencias en paralelo para minimizar riesgos.

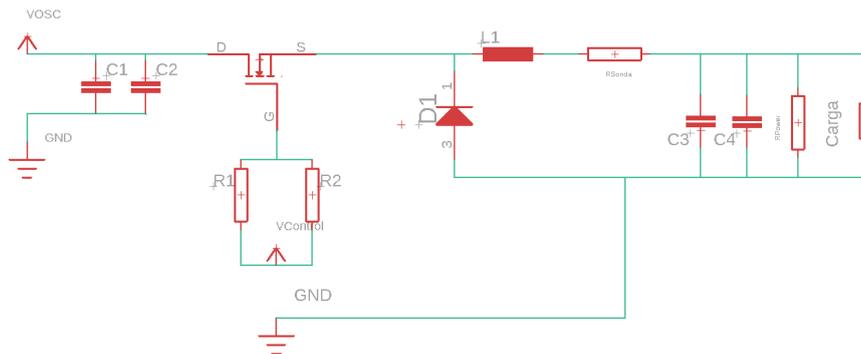


Fig. 34 Diseño de circuito Buck en Eagle

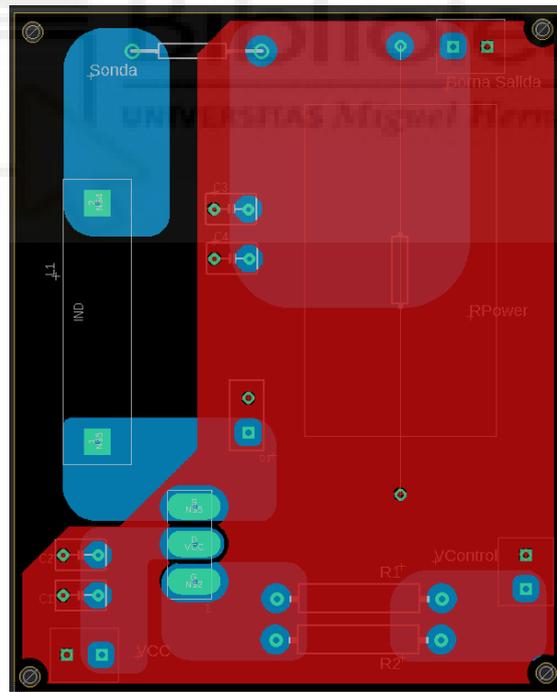


Fig. 35 Diseño de la distribución de los componentes de la placa en Eagle

Para diseñar la placa que posteriormente fabricaremos y usaremos para las validaciones de los modelos de los inductores, decidimos maximizar el área de las conexiones para poder minimizar posibles efectos generados en la placa como la

inductancia. Además de utilizar condensadores de acoplo y desacoplo para eliminar posibles señales de corriente alterna.

Por otro lado, para el inductor, decimos hacer un montaje, que nos permitirá medir inductores de diferentes tamaños para poder tener un mayor rango de inductores a medir como se puede ver en la imagen del circuito final montado. Finalmente, una vez construido, montado y soldado los componentes, nos queda el siguiente circuito:



Fig. 36 Circuito de Validación

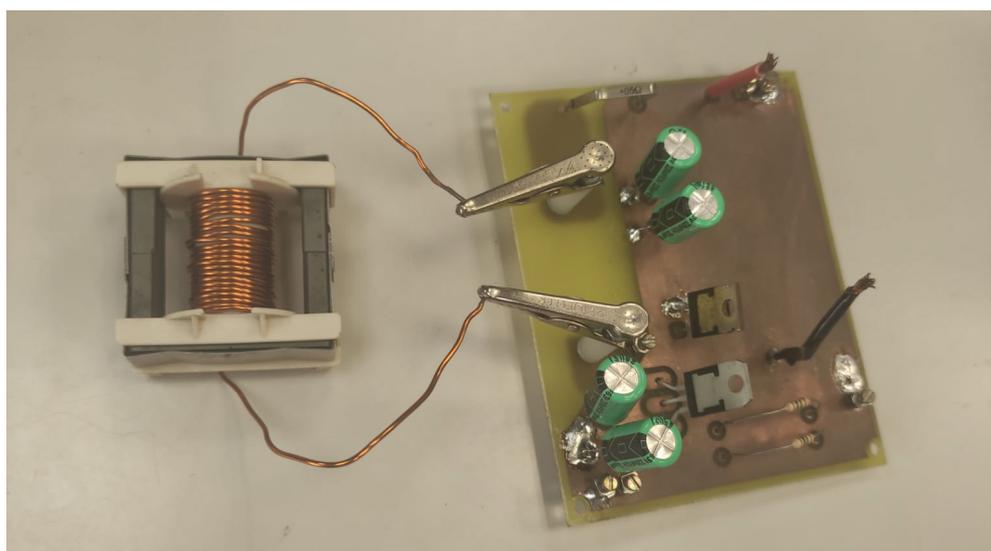


Fig. 37 Circuito de validación con inductor ETD

7. RESULTADOS EXPERIMENTALES

Una vez tenemos nuestro circuito de validación listo para realizar pruebas, comenzaremos por establecer las condiciones de trabajo en las que se realizarán las pruebas, además de los inductores a medir. Para este trabajo, al tener dos aplicaciones que pueden modelar inductores para ciertas condiciones de trabajo, primero estableceremos unas condiciones para la aplicación de modelo básico del inductor y otras condiciones para el modelo avanzado. Además, como ya se ha explicado anteriormente, el modelo avanzado se tiene que utilizar con inductores cuyos núcleos sean de ferrita y tengan una forma de núcleo específica. Esto hace que habrá inductores que se modelarán con ambas aplicaciones mientras que otros inductores se modelarán con el modelo básico.

7.1. Pruebas de validación con el modelo básico

Como hemos comentado anteriormente, el primer paso será elegir los inductores a medir. Por un lado, tendremos cinco inductores de potencia de *Coilcraft*, los cuales modelaremos con el modelo básico. Por otro lado, usaremos tres inductores de potencia cuya forma de núcleo es una de las que aparece en el Anexo 10.1, previamente diseñados y utilizados para aplicaciones de electrónica de potencia en el laboratorio de Electrónica Industrial de la Universidad Miguel Hernández de Elche.

REFERENCIA	FABRICANTE	<I> DATAHEET	TIPO NÚCLEO
DC1012-563L	COILCRAFT	2.06A	-
RFS1317-224KL	COILCRAFT	1.80A	-
RFS1317-473KL	COILCRAFT	3.90A	-
RFB1010-101L	COILCRAFT	1.90A	-
RFS1317-824KL	COILCRAFT	890mA	-
1° INDUCTOR ETD	-	-	ETD39
2° INDUCTOR ETD	-	-	ETD39
3° INDUCTOR ETD	-	-	ETD39

Tabla 1 Tabla general con datos de los inductores

Para las condiciones de trabajo de los inductores, hemos decidido tomar como referencia el valor de la intensidad que aparecen en sus *datasheet*, en el caso de los inductores pertenecientes a *Coilcraft*. A partir de este valor, decidimos hacer tres pruebas para cada inductor. Esta intensidad es la que tomaremos como referencia para validar nuestras aplicaciones y es la que nosotros mediremos en la resistencia *Shunt* como hemos explicado anteriormente. Cabe destacar que la frecuencia de trabajo del PWM será de 100KHz ya que en los *datasheet* todas las medidas se hicieron con esa frecuencia y esto hará que los resultados sean más fiables. Por lo tanto, las condiciones de trabajo fijas serán:

1. Intensidad RMS o de saturación indicada, al 80% y ciclo de trabajo, $D=0.3$
2. Intensidad RMS o de saturación indicada, y ciclo de trabajo, $D=0.5$
3. Intensidad RMS o de saturación indicada, al 120% y ciclo de trabajo, $D=0.8$

Con estas tres pruebas seremos capaces de comprobar si el comportamiento del modelo del inductor se asemeja al del inductor real en condiciones tan diferentes como las presentadas.

Una vez determinadas las condiciones de trabajo más importantes, continuaremos con la creación de un circuito *Buck* en LTspice con las mismas características y componentes que el real para poder comparar el comportamiento simulado y real. Con esto haremos diversas pruebas en el circuito simulado para conseguir las condiciones de trabajo mencionadas previamente. Para ello, modificaremos los valores de tensión de entrada, V_{in} , y la resistencia de carga, cuyos valores se pueden modificar fácilmente.

A continuación, cada inductor tendrá un circuito asociado en LTspice con sus condiciones y características específicas como vemos en la figura 33, además de la medida de corriente tanto simulada como real.

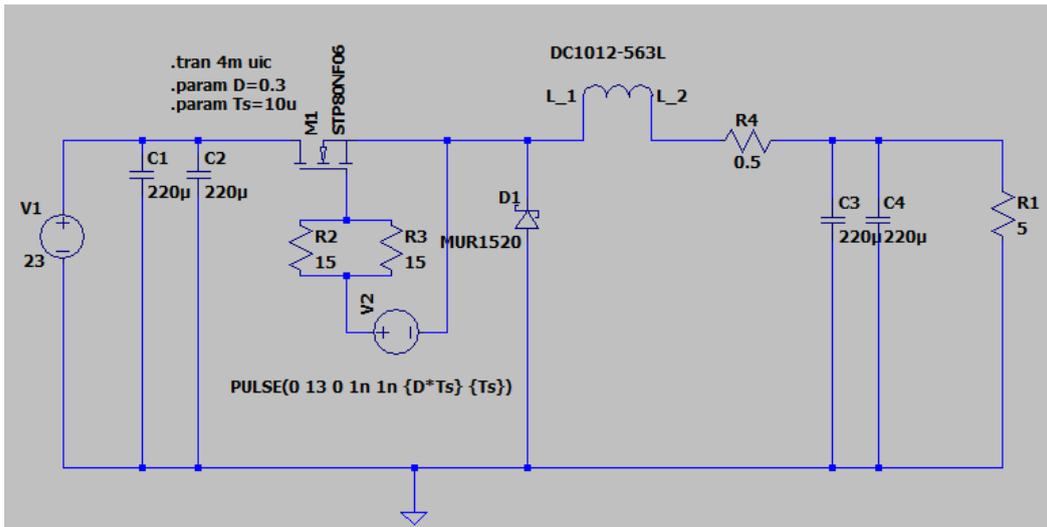


Fig. 38 Circuito Buck implementado en LTspice



7.1.1 DC1012-563L

<i>Comparación Componentes</i>	<i>Datasheet</i>	<i>Aplicación</i>
L	56 μH	28 μH
R_s	0.15 Ω	9.44 m Ω
R_p	-	16 k Ω
C_p	-	27pF

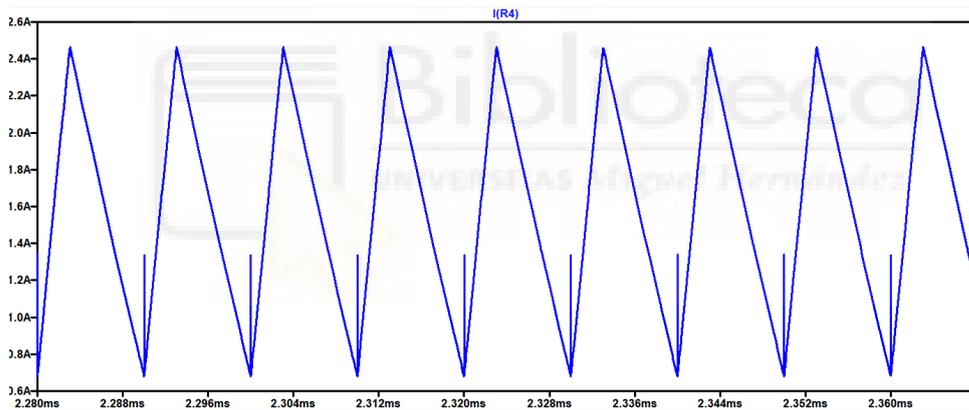
Tabla 2 Comparativa resultados simulación

1ª Prueba:

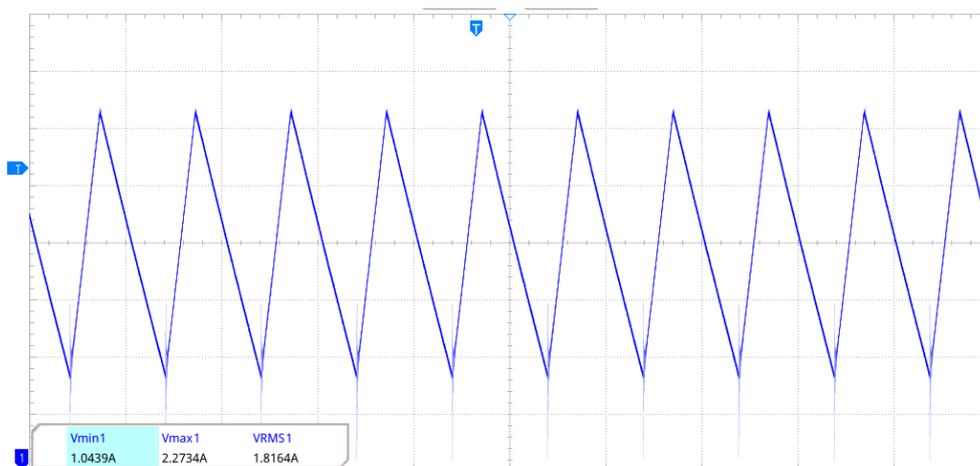
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	1.65A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	24V
Resistencia de Carga	4 Ω

Tabla 3 Condiciones Iniciales con la 1ª prueba. Modelo: DC1012-563L



Captura 1



Captura 2

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	1.6452A	1.8164A
I_{MIN}	700.0034mA	1.0439A
I_{MAX}	2.4542A	2.2734A

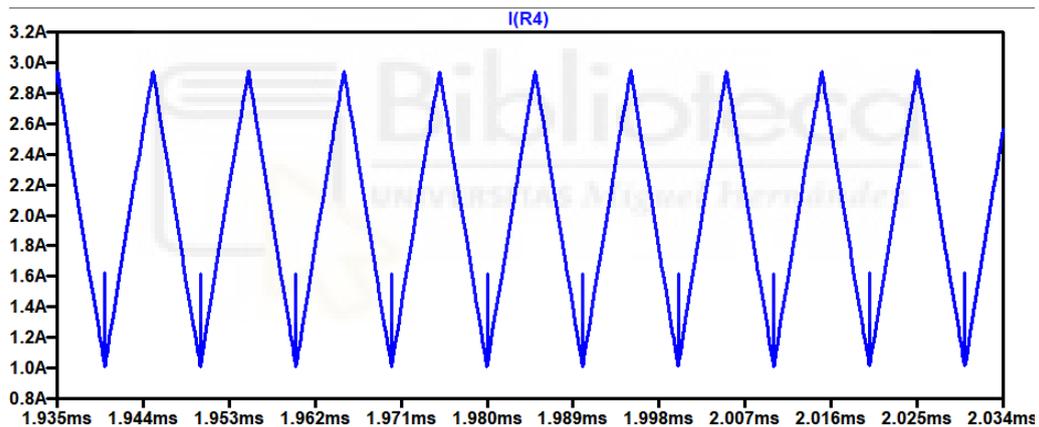
Tabla 4 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: DC1012-563L

2. Prueba:

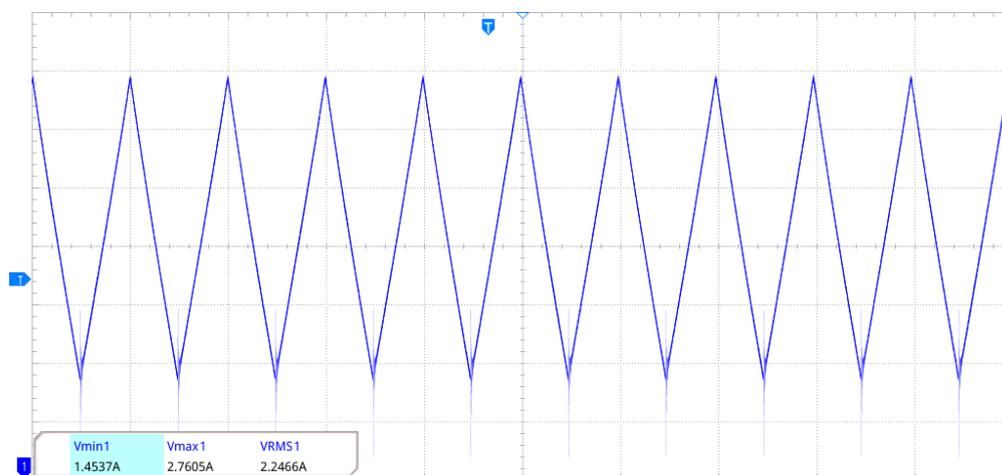
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.06A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	22V
Resistencia de Carga	5Ω

Tabla 5 Condiciones Iniciales con la 2ª prueba. Modelo: DC1012-563L



Captura 3



Captura 4

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	2.0458A	2.2466A
I_{MIN}	1.0108A	1.4538A
I_{MAX}	2.9403A	2.7605A

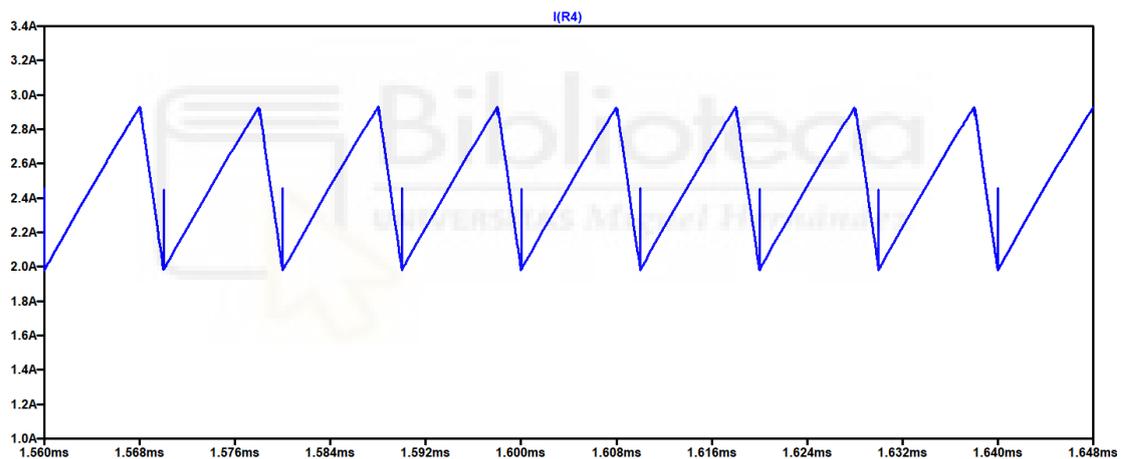
Tabla 6 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: DC1012-563L

3.Prueba

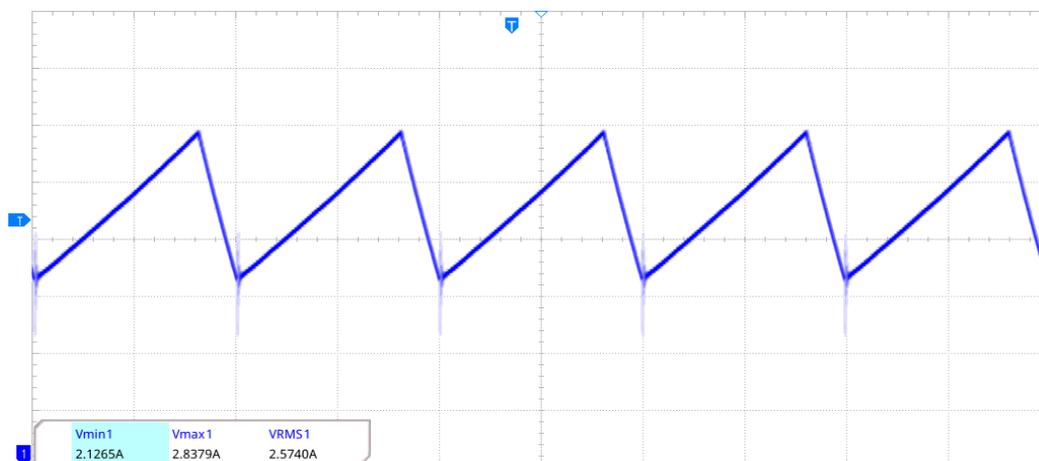
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.48A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	16V
Resistencia de Carga	5Ω

Tabla 7 Condiciones Iniciales con la 3ª prueba. Modelo: DC1012-563L



Captura 5



Captura 6

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	2.4635A	2.5740A
I_{MIN}	2.0040A	2.1265A
I_{MAX}	2.9216A	2.8379A

Tabla 8 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: DC1012-563L

Observaciones:

Una vez realizadas las tres pruebas indicadas anteriormente al inductor DC1012-563L, podemos comprobar que, en general, la forma de onda y las intensidades que obtenemos tienen un valor bastante similar, si bien en la mayoría de las pruebas hay una diferencia de unos 0.2A tanto en la primera prueba donde buscamos un comportamiento a intensidades no tan exigentes para el inductor que hemos elegido como a intensidades más elevadas en las que comenzamos a entrar en el rango de saturación según el *datasheet* de los fabricantes. Sin embargo, no vemos ningún rastro de una posible saturación ni en LTspice ni en las medidas del circuito de validación. Además, observamos algunos picos al cambiar de estado OFF-ON, el cual creemos que es debido a la capacidad parásita que tiene el inductor que también se refleja en varias medidas del circuito.

7.1.2 RFS1317-224KL

<i>Comparación Componentes</i>	<i>Datasheet</i>	<i>Aplicación</i>
L	220 μH	222 μH
R_s	0.239	0.198 Ω
R_p	-	41 $\text{k}\Omega$
C_p	-	2.43pF

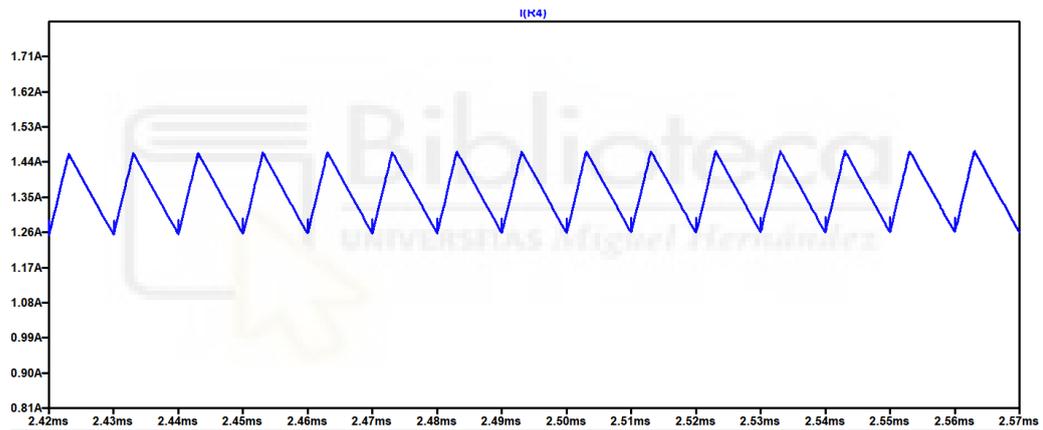
Tabla 9 Comparativa resultados simulación

1. Prueba

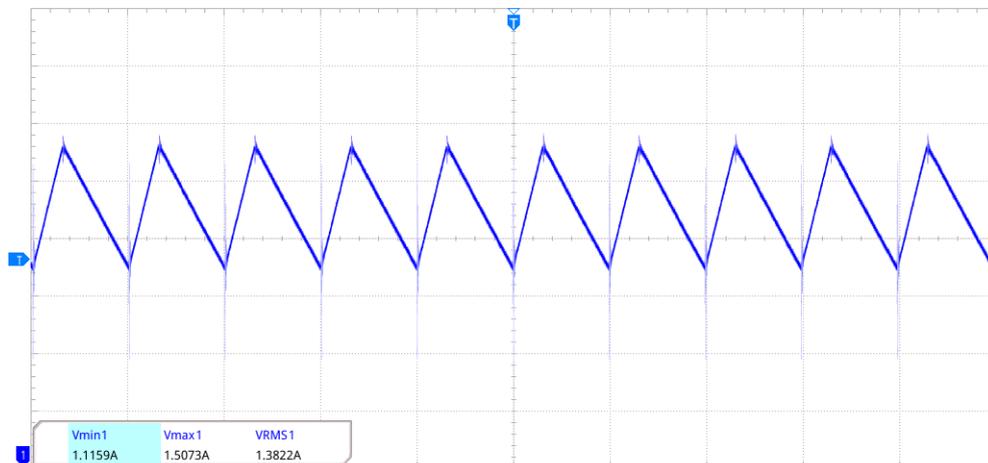
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	1.36A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	22V
Resistencia de Carga	4 Ω

Tabla 10 Condiciones Iniciales con la 1ª prueba. Modelo: RFS1317-224KL



Captura 7



Captura 8

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	1.3645A	1.3822A
I_{MIN}	1.2626A	1.1159A
I_{MAX}	1.4656A	1.5073A

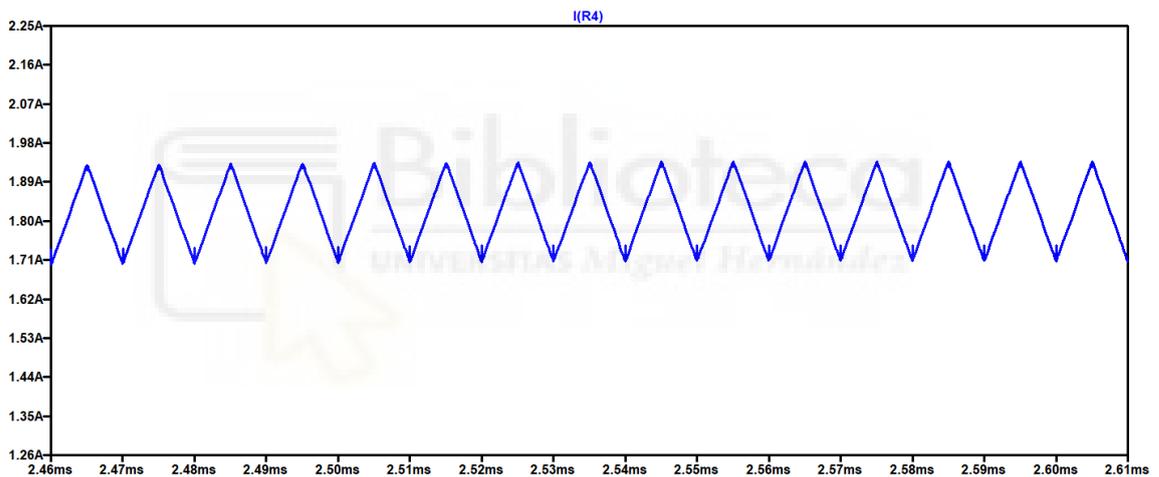
Tabla 11 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFS1317-224KL

2. Prueba

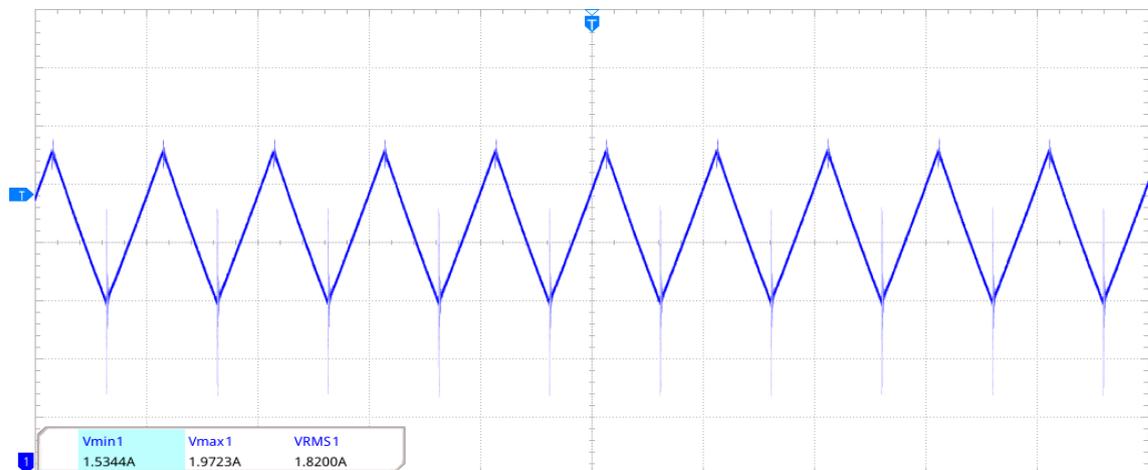
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	1.80A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	20V
Resistencia de Carga	5 Ω

Tabla 12 Condiciones Iniciales con la 2ª prueba. Modelo: RFS1317-224KL



Captura 9



Captura 10

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	1.8217A	1.8200A
I_{MIN}	1.7209A	1.5344A
I_{MAX}	1.9347A	1.9723A

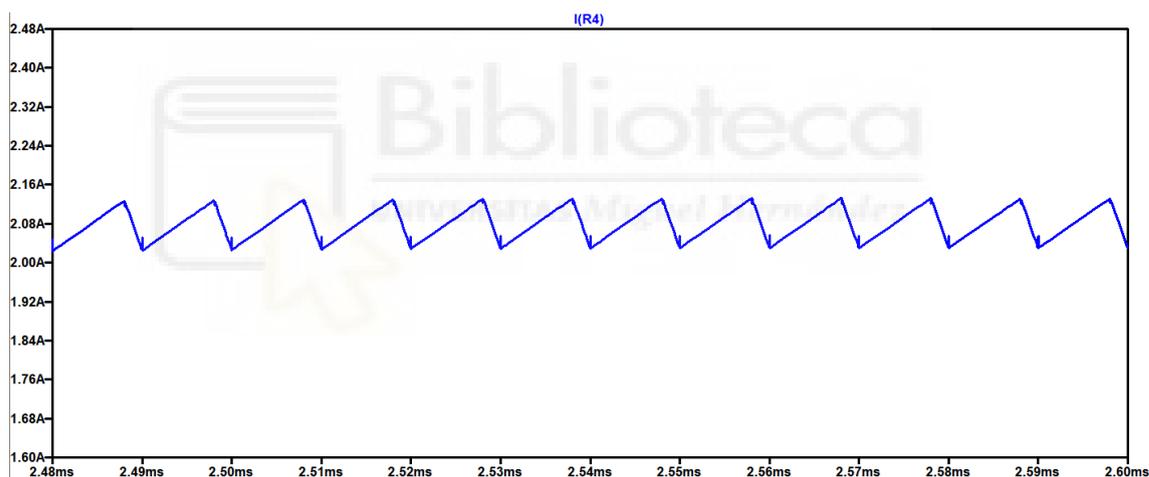
Tabla 13 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFS1317-224KL

3. Prueba

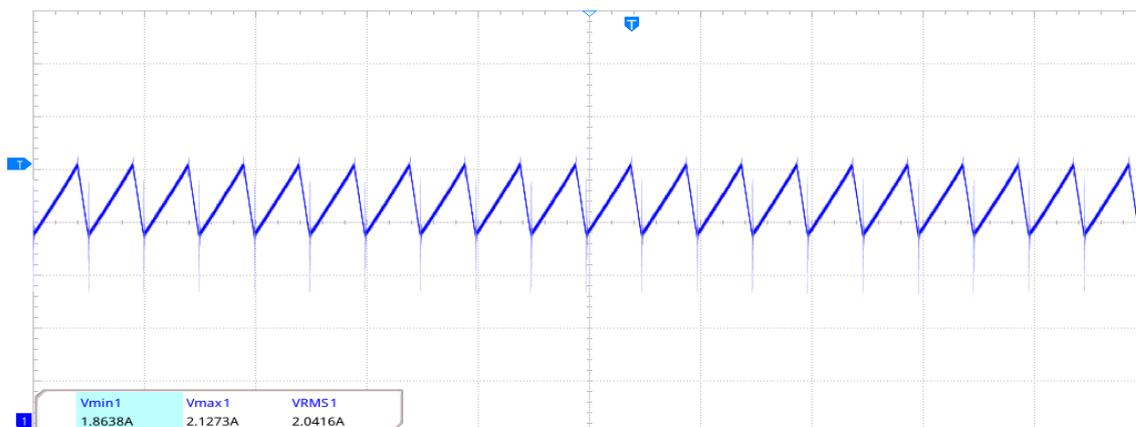
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.04A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	14V
Resistencia de Carga	5 Ω

Tabla 14 Condiciones Iniciales con la 3ª prueba. Modelo: RFS1317-224KL



Captura 11



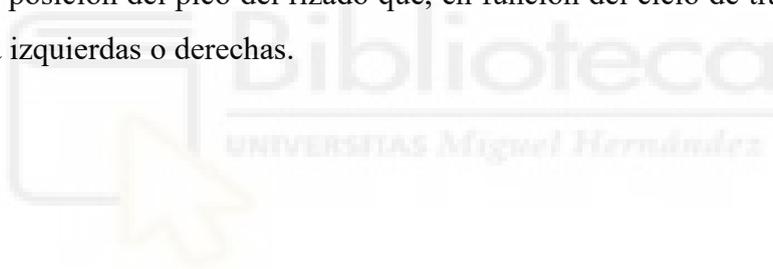
Captura 12

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	2.0799A	2.0416A
I_{MIN}	2.0316 A	1.8638A
I_{MAX}	2.1526A	2.1273A

Tabla 15 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFS1317-224KL

Observaciones:

Respecto a los resultados obtenidos con el inductor RFS1317-224KL, observamos que la intensidad RMS es prácticamente igual en todas las pruebas acometidas. Además, se observa que, aunque la intensidad RMS es igual, la amplitud pico-pico sí que difiere un poco más entre las pruebas en LTspice y las medidas del circuito de validación, aunque tampoco en exceso. Además, nos sigue apareciendo ese pico cuando el MOSFET cambia de estado OFF a ON en distintas pruebas aún siendo mucho menos visible que en las pruebas con el inductor DC1012-563L. Cabe destacar que las formas de onda comparten el mismo comportamiento dependiendo del ciclo de trabajo de la prueba, como se puede observar en la posición del pico del rizado que, en función del ciclo de trabajo, varía su posición hacia izquierdas o derechas.



7.1.3 RFS1317-473KL

Comparación Componentes

	Datasheet	Aplicación
L	470 μH	465 μH
R_s	0.409 Ω	31 $\text{m}\Omega$
R_p	-	333 $\text{k}\Omega$
C_p	-	3.13 pF

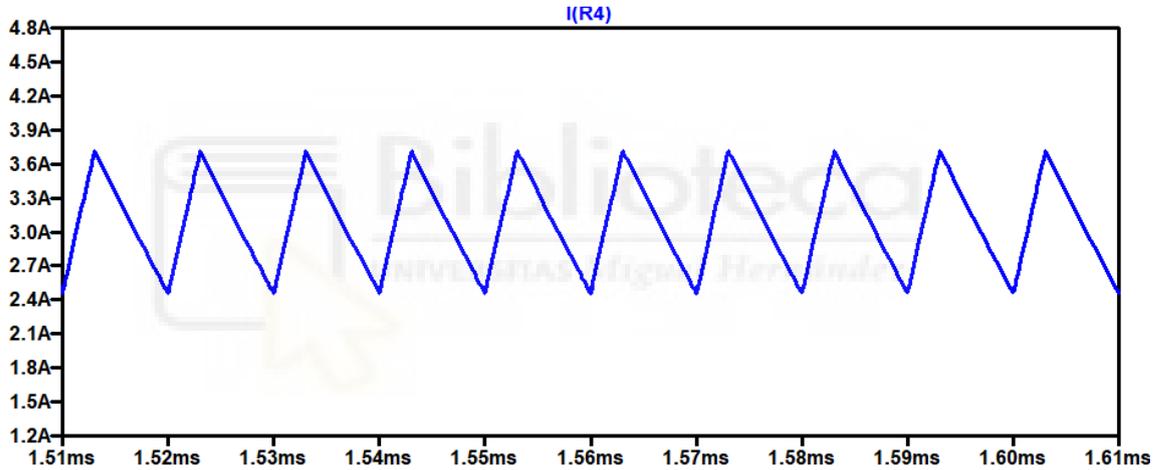
Tabla 16 Comparativa resultados simulación

1. Prueba

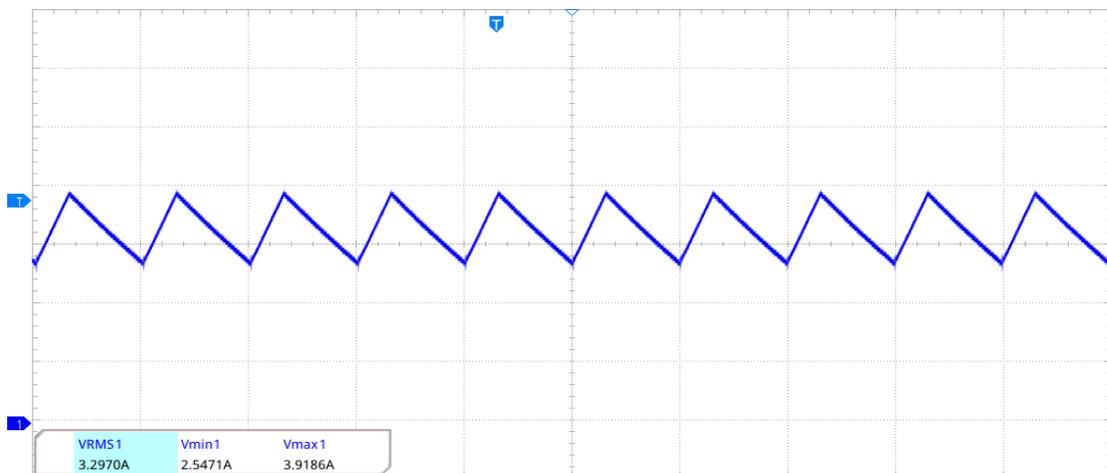
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	3.12A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	27V
Resistencia de Carga	2 Ω

Tabla 17 Condiciones Iniciales con la 1ª prueba. Modelo: RFS1317-473KL



Captura 13



Captura 14

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	3.1316A	3.2970A
I_{MIN}	2.4998A	2.5474A
I_{MAX}	3.7324A	3.9186A

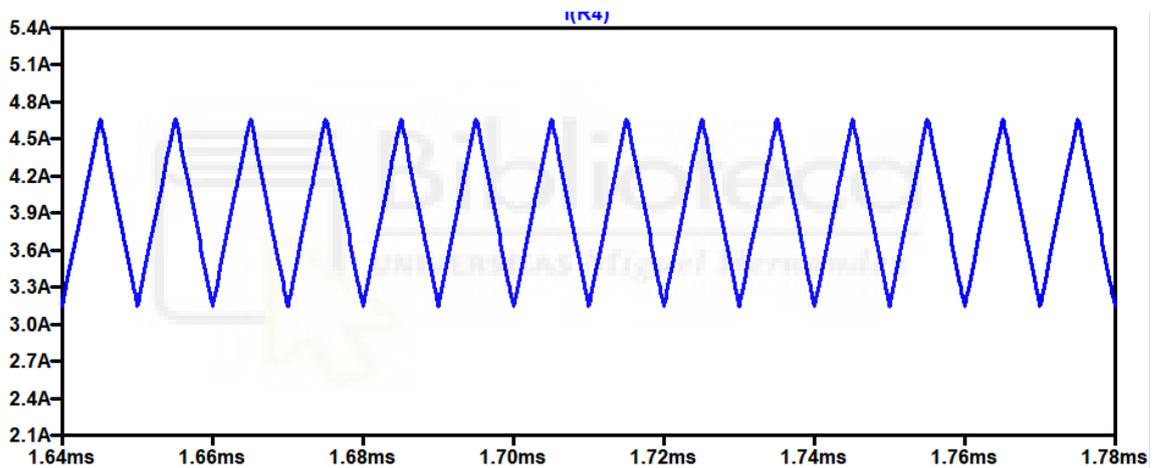
Tabla 18 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFS1317-473KL

2. Prueba

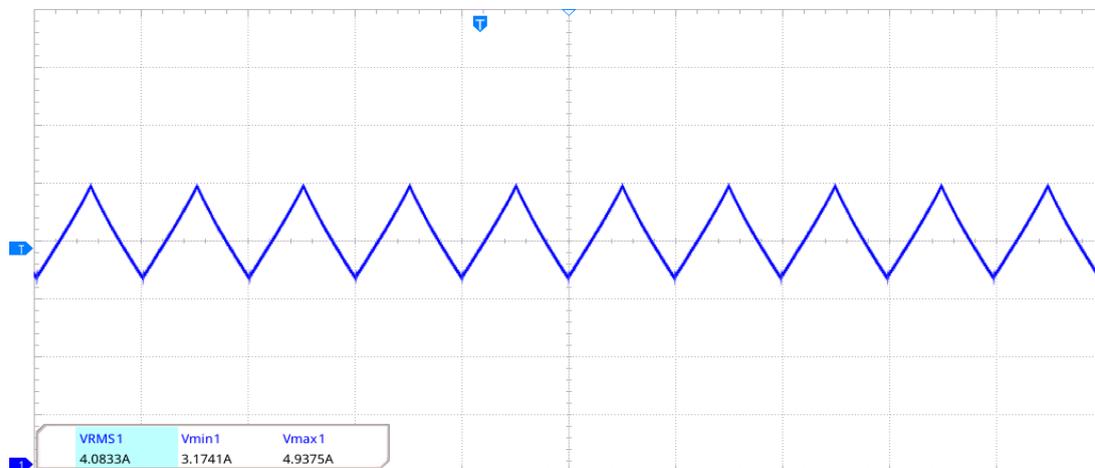
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	3.90A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	28V
Resistencia de Carga	3 Ω

Tabla 19 Condiciones Iniciales con la 2ª prueba. Modelo: RFS1317-473KL



Captura 15



Captura 16

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	3.8949A	4.0833A
I_{MIN}	3.1577A	3.1747A
I_{MAX}	4.7275A	4.9375A

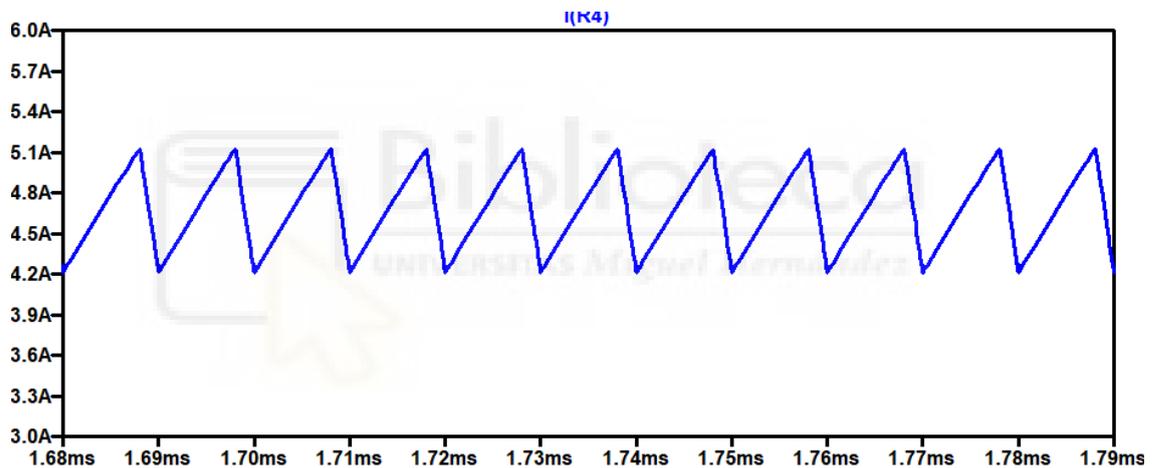
Tabla 20 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFS1317-473KL

3. Prueba

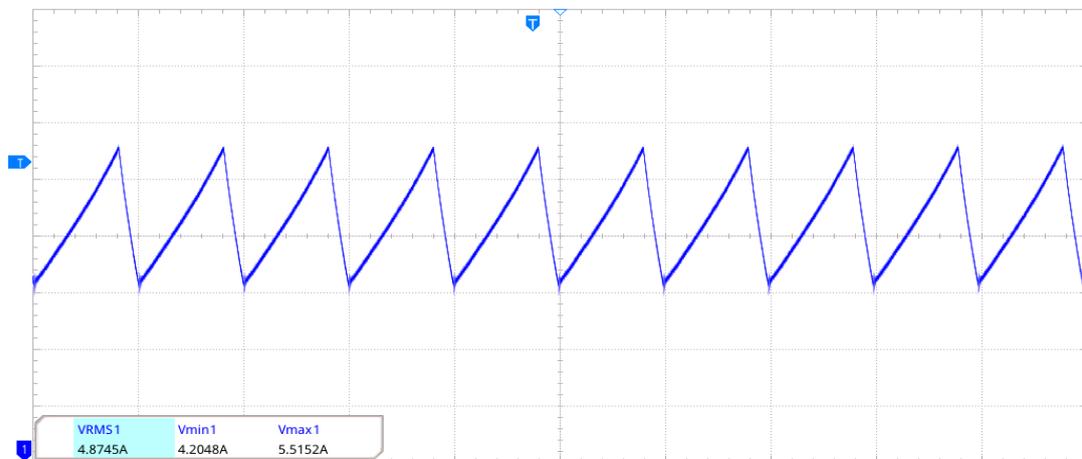
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	4.68A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	27V
Resistencia de Carga	4 Ω

Tabla 21 Condiciones Iniciales con la 3ª prueba. Modelo: RFS1317-473KL



Captura 17



Captura 18

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	4.6946A	4.8745A
I_{MIN}	4.2180A	4.2048A
I_{MAX}	5.1280A	5.5152A

Tabla 22 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFS1317-473KL

Observaciones:

Respecto a los resultados obtenidos con el inductor RFS1317-473KL, observamos que, como con el primer inductor, existe una diferencia de valor entre las intensidades RMSs del circuito LTspice y el circuito real de alrededor de 0.2A. Si hablamos sobre la forma de onda, en este caso también obtenemos resultados similares en su comportamiento dependiendo del ciclo de trabajo.

Por otro lado, en las dos primeras pruebas observamos una amplitud pico-pico semejante mientras que en la tercera prueba observamos una mayor diferencia de intensidad máxima de unos 0.4A de diferencia. Esto podría deberse a que el inductor podría estar en un estado cerca de la saturación y el comportamiento del inductor podría variar, pero este modelo no tiene en cuenta este tipo de sucesos.

7.1.4 RFB1010-101L

<i>Comparación Componentes</i>	<i>Datasheet</i>	<i>Aplicación</i>
L	100 μH	103 μH
R_s	0.15 Ω	0.32 Ω
R_p	-	47 k Ω
C_p	-	

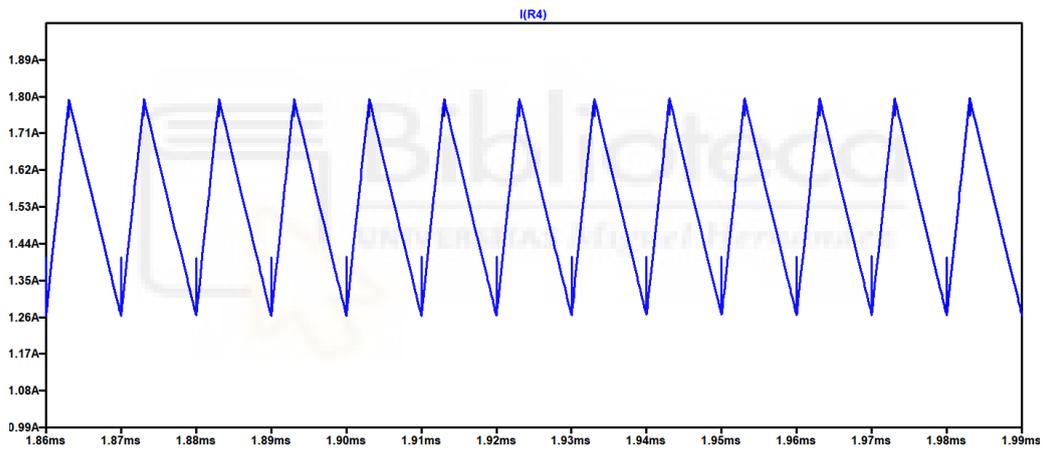
Tabla 23 Comparativa resultados simulación

1. Prueba

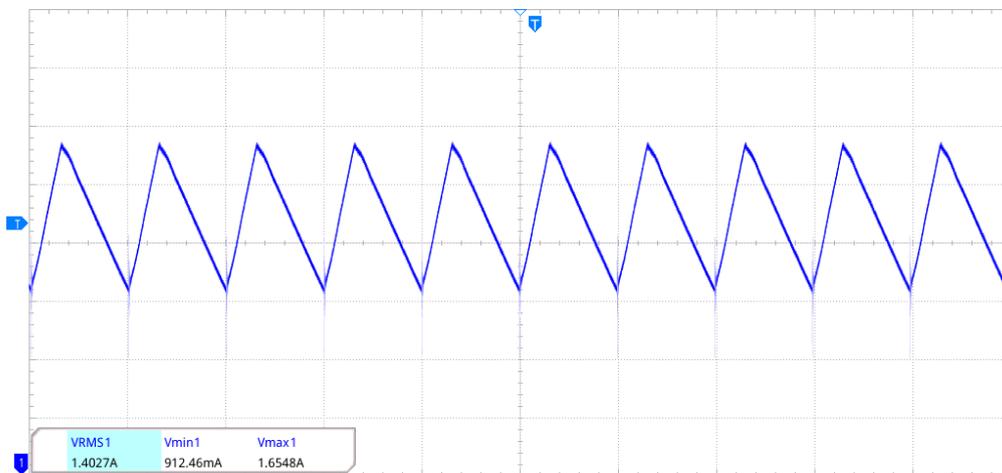
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	1.52A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	25.5V
Resistencia de Carga	4 Ω

Tabla 24 Condiciones Iniciales con la 1ª prueba. Modelo: RFB1010-101L



Captura 19



Captura 20

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	1.5358A	1.4072A
I_{MIN}	1.2625A	912.46mA
I_{MAX}	1.7617A	1.6548A

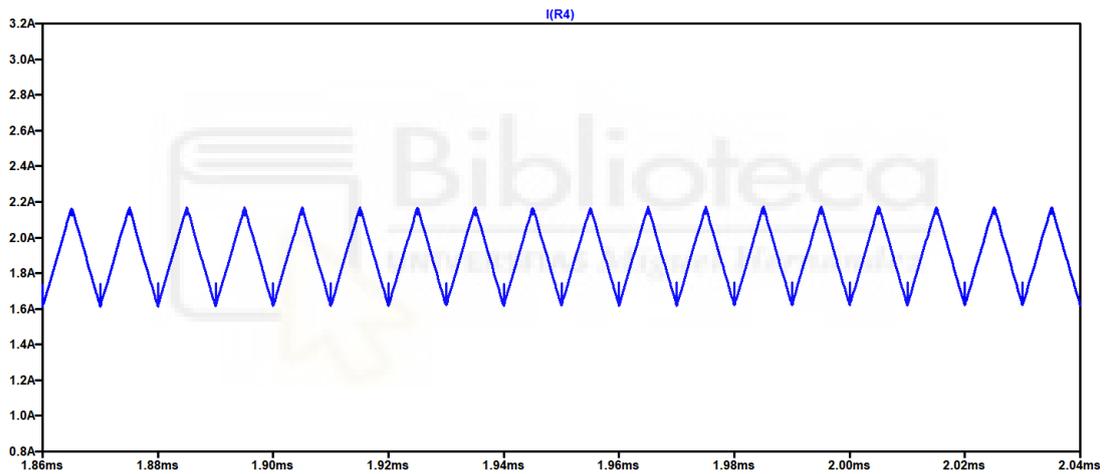
Tabla 25 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFB1010-101L

2. Prueba

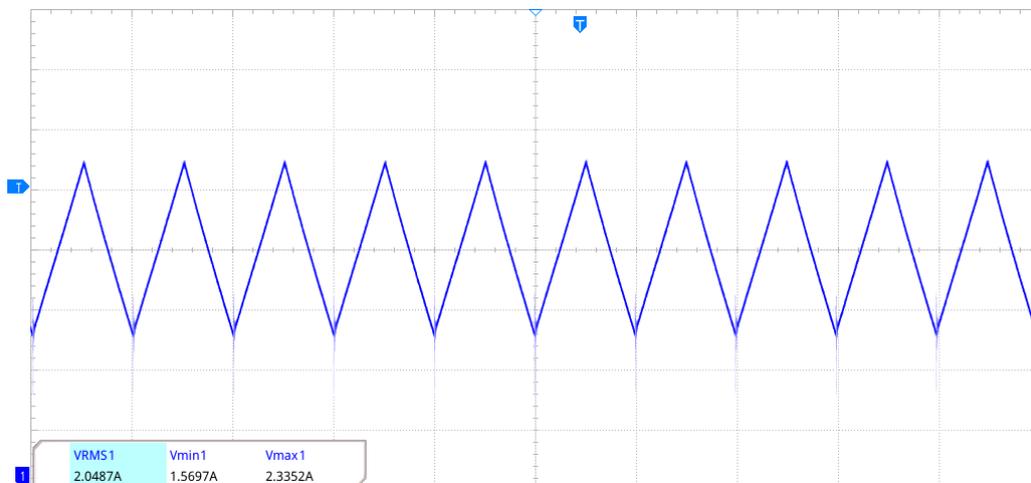
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	1.9A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	22.5V
Resistencia de Carga	5 Ω

Tabla 26 Condiciones Iniciales con la 2ª prueba. Modelo: RFB1010-101L



Captura 21



Captura 22

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	1.9006A	2.0487A
I_{MIN}	1.6244A	1.5697A
I_{MAX}	2.1668A	2.3352A

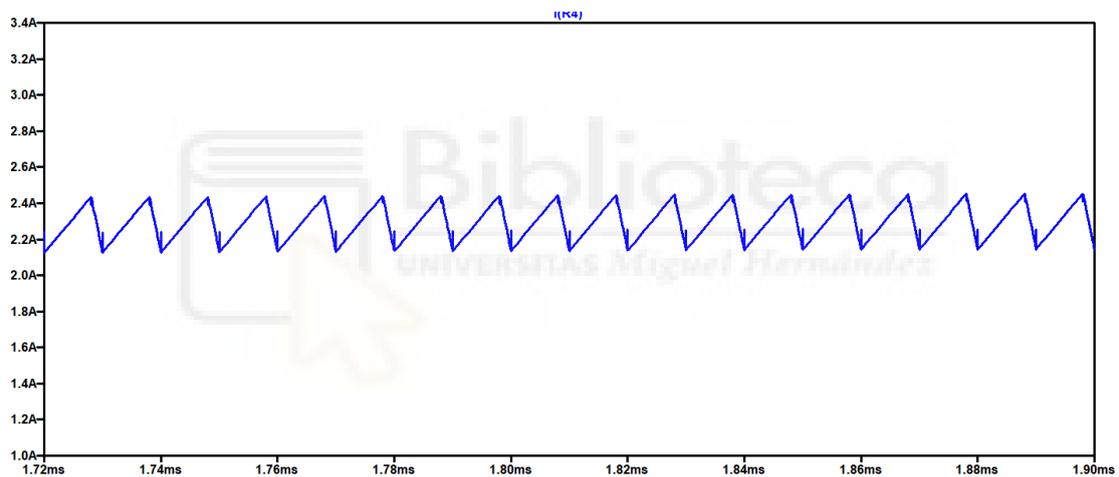
Tabla 27 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFB1010-101L

3. Prueba

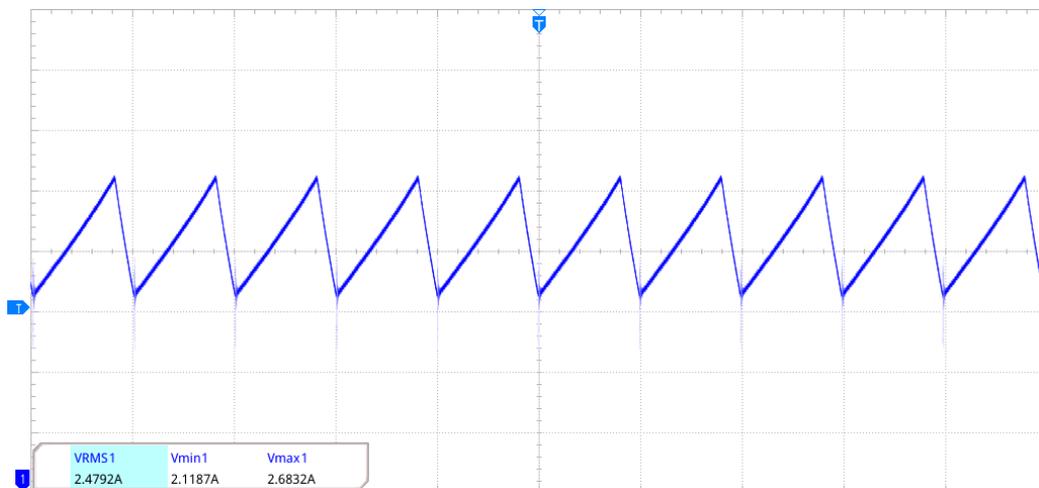
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.28A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	19.8V
Resistencia de Carga	6 Ω

Tabla 28 Condiciones Iniciales con la 3ª prueba. Modelo: RFB1010-101L



Captura 23



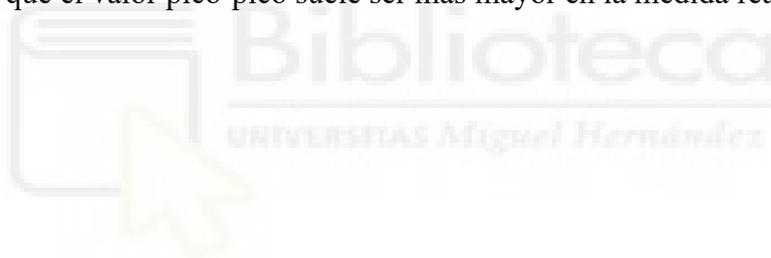
Captura 24

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	2.2922A	2.4792A
I_{MIN}	2.1439A	2.1187A
I_{MAX}	2.4414A	2.6832A

Tabla 29 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFB1010-101L

Observaciones:

Respecto a los resultados obtenidos con el inductor RFB1010-101L, vemos que sigue un poco la tendencia de las pruebas correspondientes a los inductores medidos donde la intensidad RMS es parecida, pero existe una ligera diferencia de unos 0.1-0.2A. Además, en las formas de onda vemos que se asemejan bastante, aunque en la primera y tercera prueba la amplitud pico-pico tienen una diferencia algo más notable que en la segunda. Cabe destacar que en la primera prueba vemos que la amplitud pico-pico del LTspice es mayor que la medida en el circuito real, cosa que en el resto de las pruebas no sucede ya que el valor pico-pico suele ser más mayor en la medida real.



7.1.5 RFS1317-824KL

<i>Comparación Componentes</i>	<i>Datasheet</i>	<i>Aplicación</i>
L	820 μ H	811 μ H
R_s	0.834 Ω	0.73 Ω
R_p	-	108 k Ω
C_p	-	2.37pF

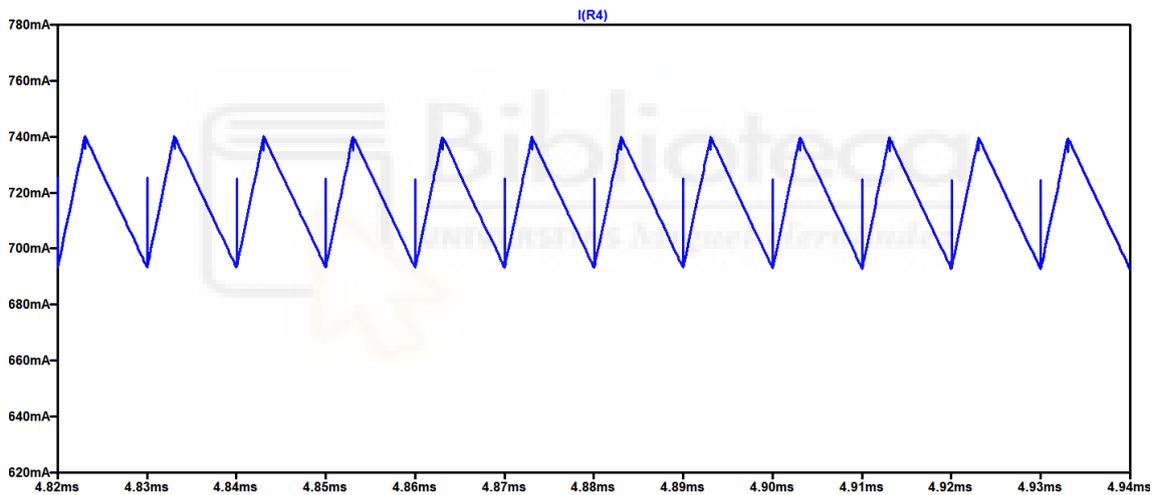
Tabla 30 Comparativa resultados simulación

1. Prueba

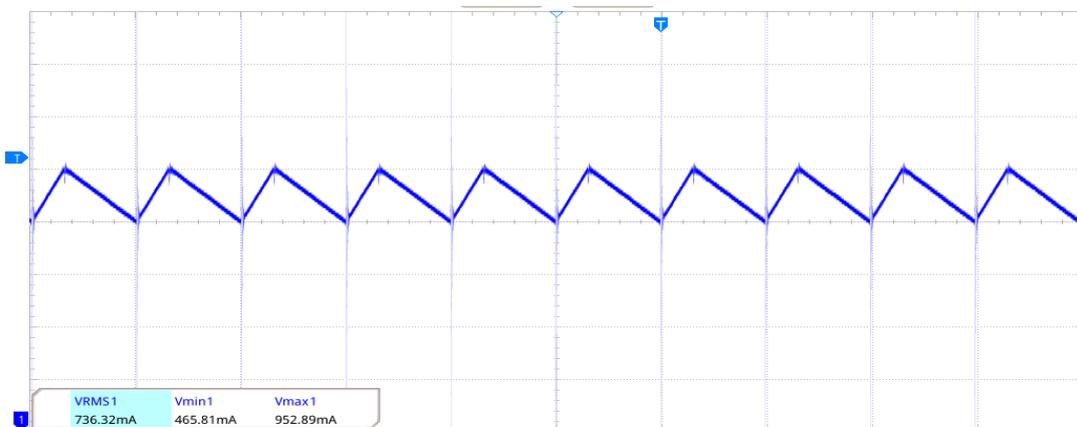
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	712mA
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	17.5V
Resistencia de Carga	6 Ω

Tabla 31 Condiciones Iniciales con la 1ª prueba. Modelo: RFS1317-824KL



Captura 25



Captura 26

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	716.52mA	736.32mA
I_{MIN}	693.44mA	465.81mA
I_{MAX}	738.46mA	952.89mA

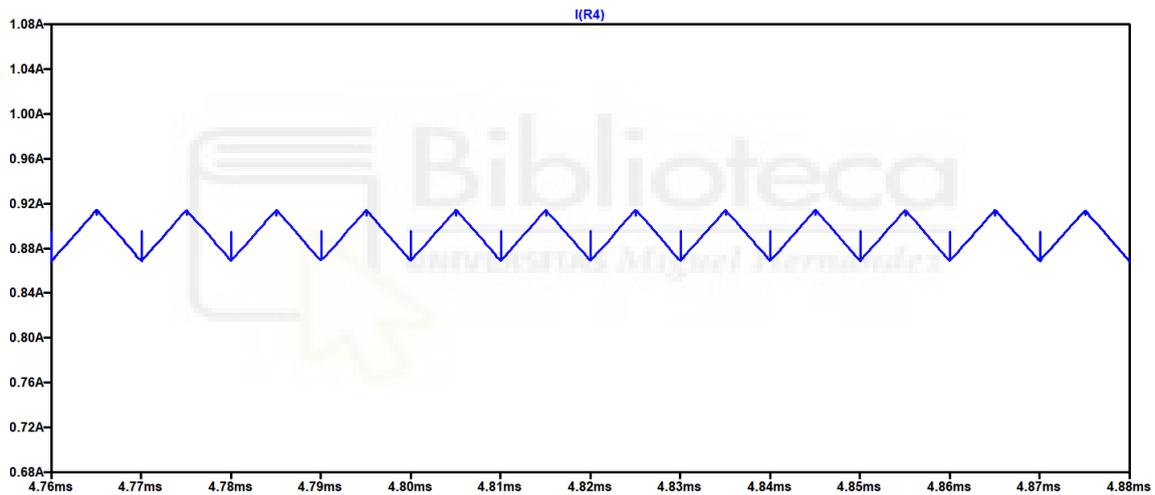
Tabla 32 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: RFS1317-824KL

2. Prueba

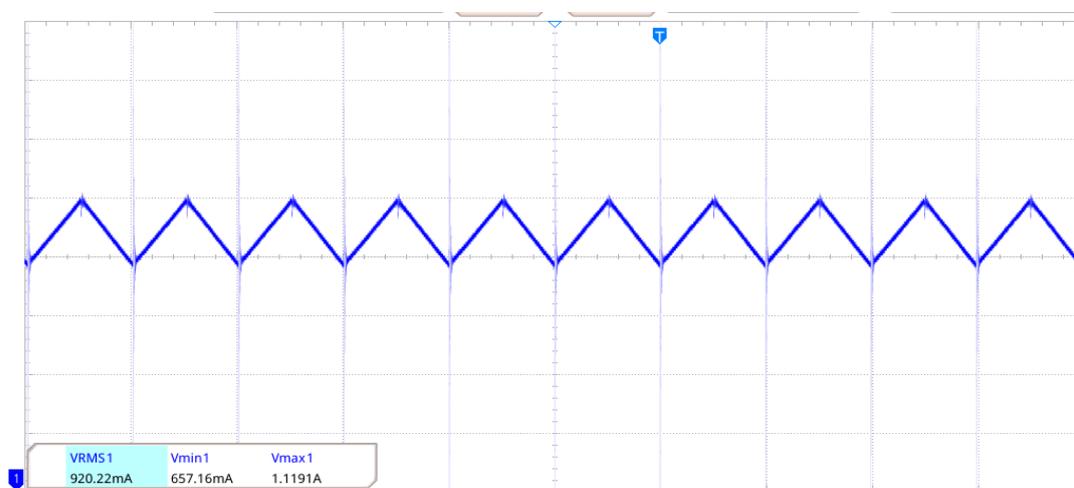
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	890mA
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	14.7V
Resistencia de Carga	7 Ω

Tabla 33 Condiciones Iniciales con la 2ª prueba. Modelo: RFS1317-824KL



Captura 27



Captura 28

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	890.72mA	920.22mA
I_{MIN}	868.68mA	657.16mA
I_{MAX}	914.05mA	1.1191A

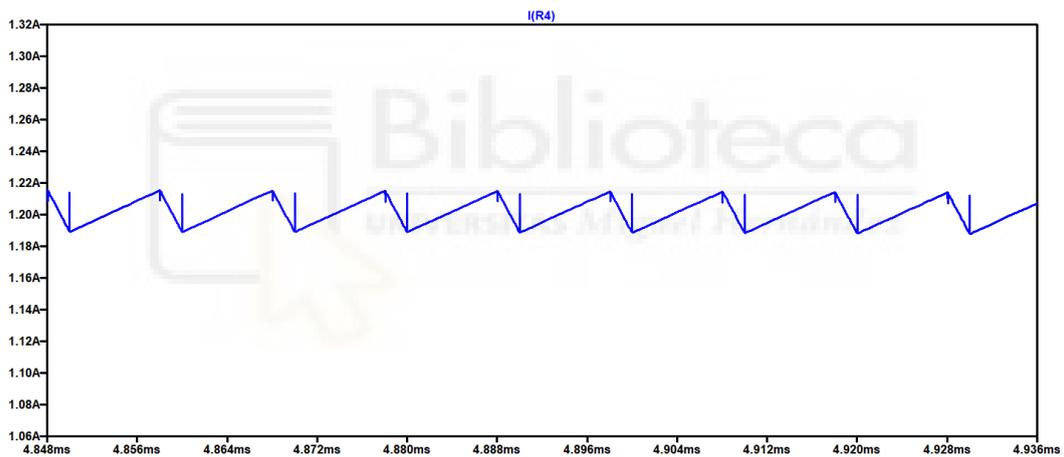
Tabla 34 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: RFS1317-824KL

3. Prueba

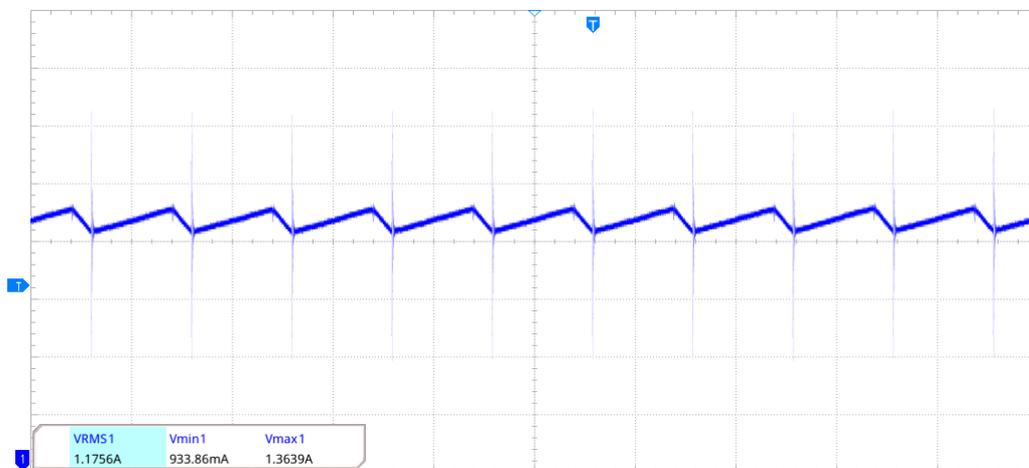
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	1.20A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	13V
Resistencia de Carga	8 Ω

Tabla 35 Condiciones Iniciales con la 3ª prueba. Modelo: RFS1317-824KL



Captura 29



Captura 30

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	1.2015A	1.1756A
I_{MIN}	1.1902A	933.86mA
I_{MAX}	1.2151A	1.3639A

Tabla 36 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: RFS1317-824KL

Observaciones:

Respecto a los resultados obtenidos con el inductor RFS1317-824KL observamos que la intensidad RMS en las tres pruebas es prácticamente la misma con una diferencia de tan solo 20-30mA. Sin embargo, en las tres pruebas tenemos una mayor diferencia en lo que se refiere a la amplitud pico-pico que obtenemos tanto en LTspice como en la medida real. Mientras que en LTspice, la amplitud pico-pico es de alrededor de 20mA, en las medidas reales, tenemos hasta una amplitud de 400mA. Este comportamiento puede ser debido a que haya entrado en saturación y por eso se produzca esta diferencia de amplitudes. Por otro lado, volvemos a observar los picos generados en el cambio OFF-ON tanto en la simulación como en el circuito.

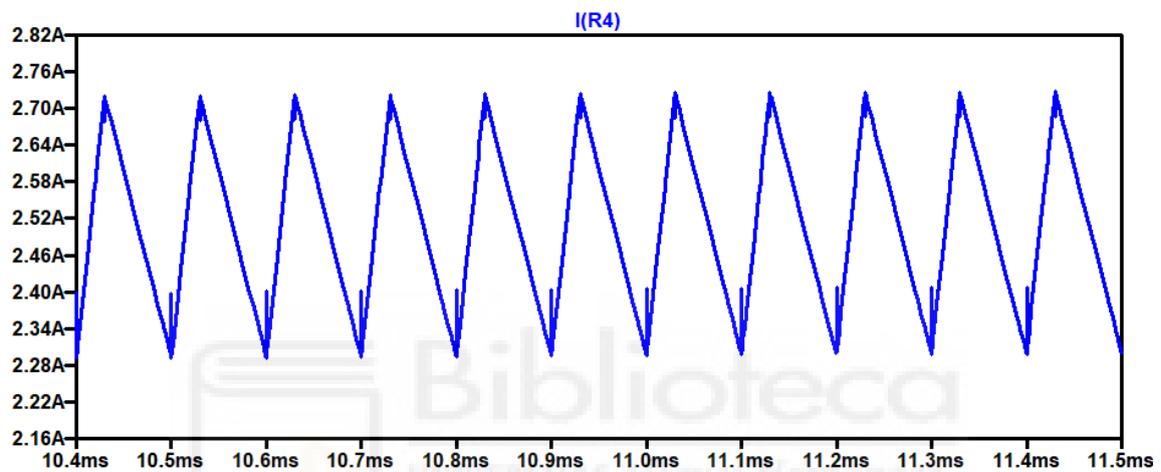
7.1.6 Primer inductor con forma de núcleo ETD

1. Prueba

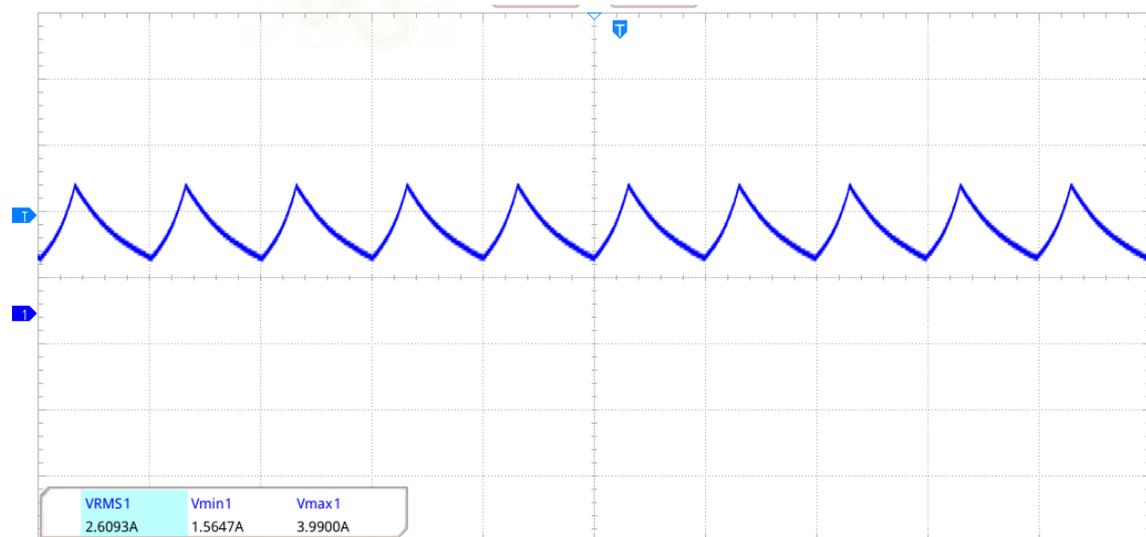
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.5A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	32V
Resistencia de Carga	3 Ω

Tabla 37 Condiciones Iniciales con la 1ª prueba. Modelo: 1r Inductor ETD



Captura 31



Captura 32

COMPARATIVA

LTSPICE

CIRCUITO REAL

I_{RMS}	2.5149A	2.6093A
I_{MIN}	2.2998A	1.5647A
I_{MAX}	2.7238A	3.9900A

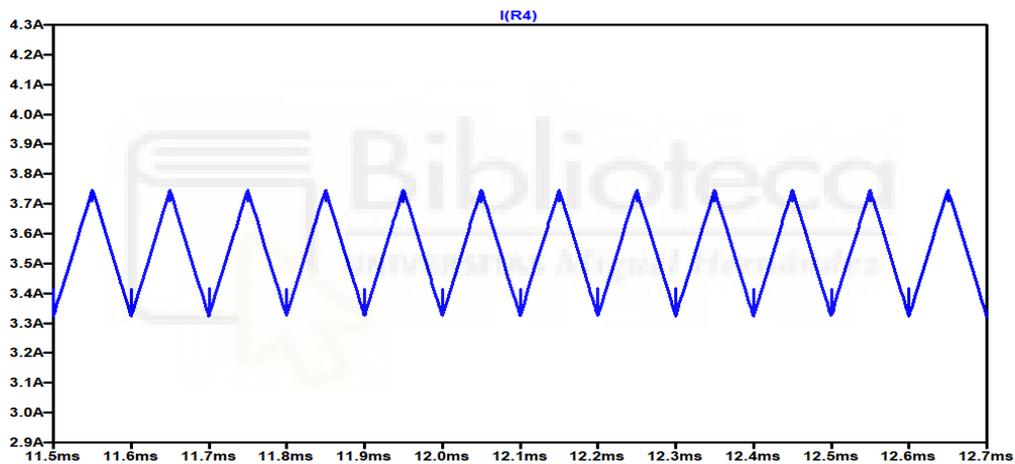
Tabla 38 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: 2n Inductor ETD

2. Prueba

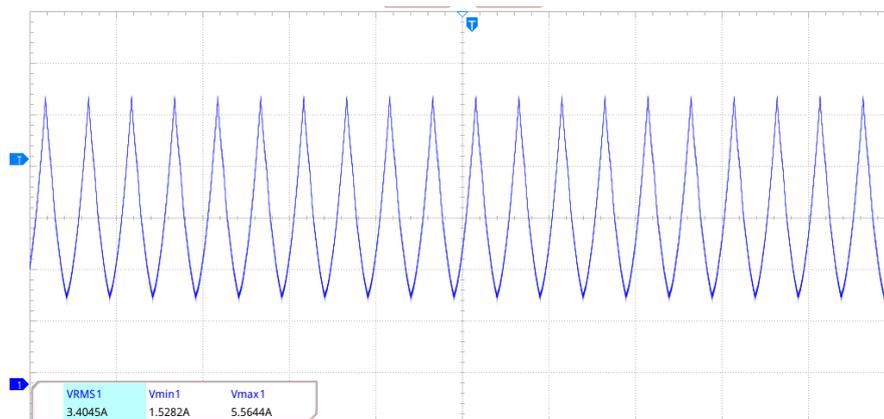
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	3.5A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	26.5V
Resistencia de Carga	3 Ω

Tabla 39 Condiciones Iniciales con la 2ª prueba. Modelo: 1r Inductor ETD



Captura 33



Captura 34

COMPARATIVA

LTSPICE

CIRCUITO REAL

I_{RMS}	3.5374A	3.4045A
I_{MIN}	3.3254A	1.5280A
I_{MAX}	3.7458A	5.5644A

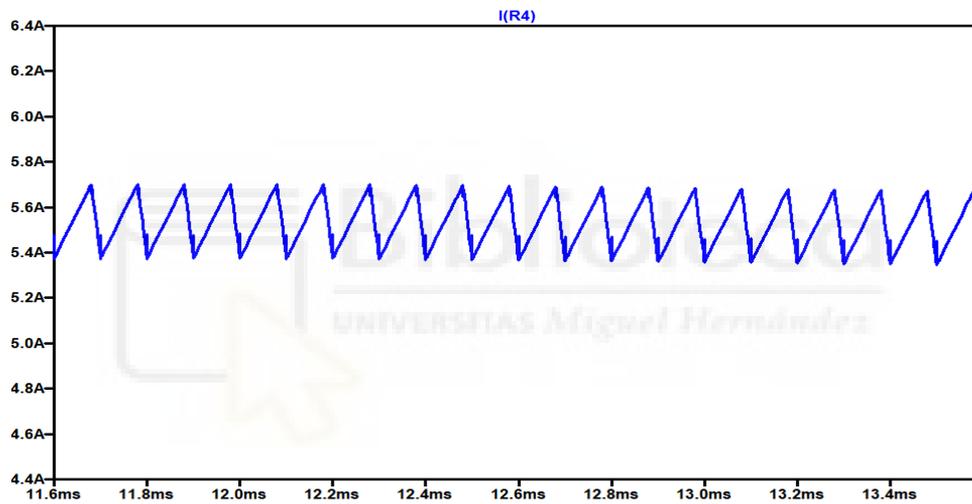
Tabla 40 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: 1r Inductor ETD

3.Prueba

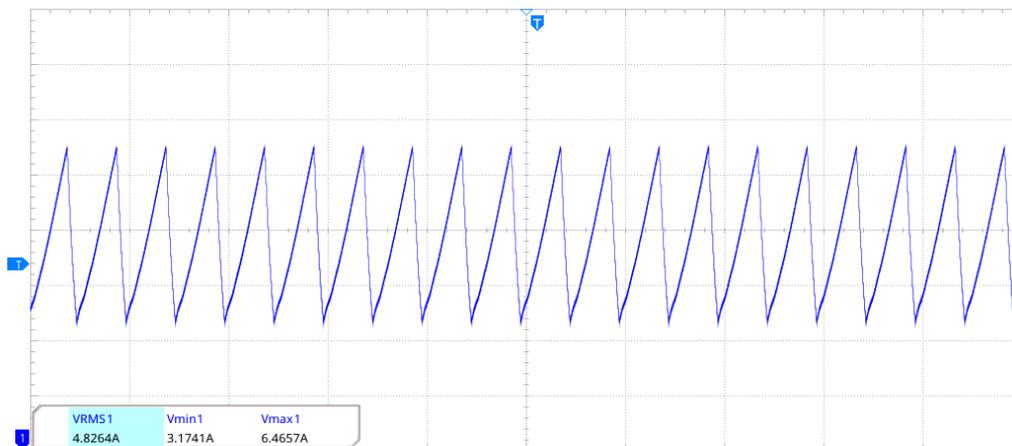
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	5.5A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	32V
Resistencia de Carga	4 Ω

Tabla 41 Condiciones Iniciales con la 3ª prueba. Modelo: 1r Inductor ETD



Captura 35



Captura 36

COMPARATIVA

LTSPICE

CIRCUITO REAL

I_{RMS}	5.5151A	4.8264A
I_{MIN}	5.3812A	3.1741A
I_{MAX}	5.6887A	6.4657A

Tabla 42 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: 1r Inductor ETD

Observaciones:

Respecto a los resultados obtenidos con el primer inductor con forma de núcleo ETD observamos que, como ya pasara con el inductor RFS1317-824KL, la diferencia entre la amplitud pico-pico en todas las pruebas es muy amplia añadiéndole que la intensidad RMS también es bastante dispar en la tercera prueba principalmente donde hay una diferencia de 0.7A, lo cual nos hace pensar que el inductor está en saturación y por eso obtenemos esas intensidades tan altas.



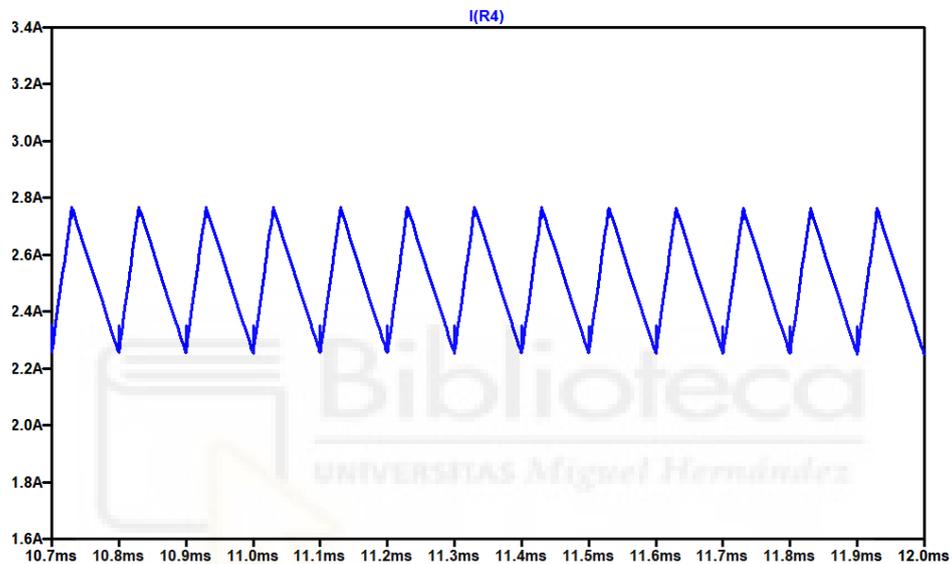
7.1.7 Segundo inductor con forma de núcleo ETD

1. Prueba

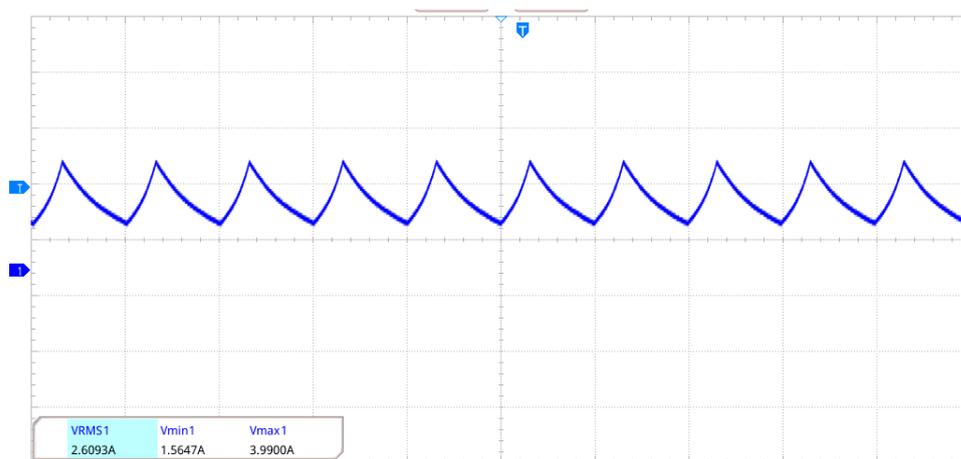
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.5A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	31V
Resistencia de Carga	3 Ω

Tabla 43 Condiciones Iniciales con la 1ª prueba. Modelo: 2n Inductor ETD



Captura 37



Captura 38

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	2.5126A	2.6093A
I_{MIN}	2.2575A	1.5647A
I_{MAX}	2.7515A	3.9900A

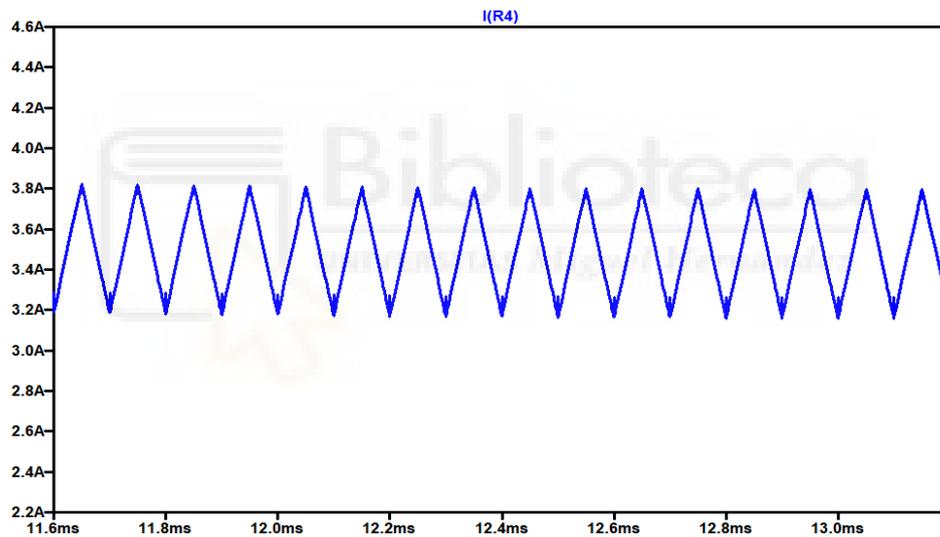
Tabla 44 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: 2n Inductor ETD

2. Prueba

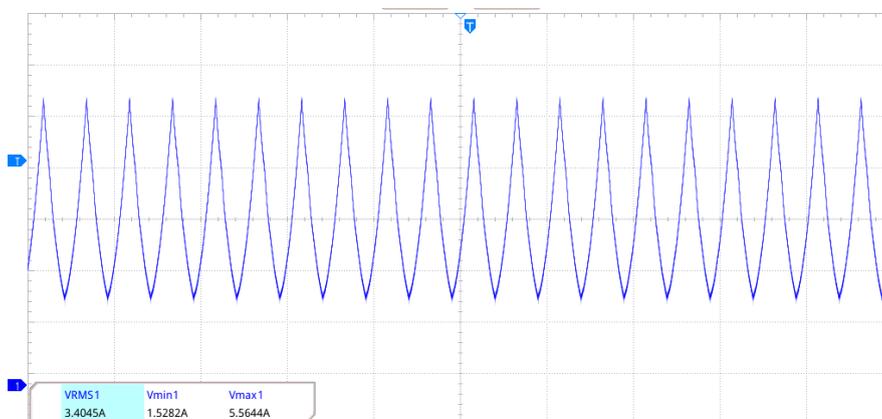
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	3.5A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	32.4V
Resistencia de Carga	4 Ω

Tabla 45 Condiciones Iniciales con la 2ª prueba. Modelo: 2n Inductor ETD



Captura 39



Captura 40

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	3.4911A	3.4045A
I_{MIN}	3.1751A	1.5280A
I_{MAX}	3.7896A	5.5644A

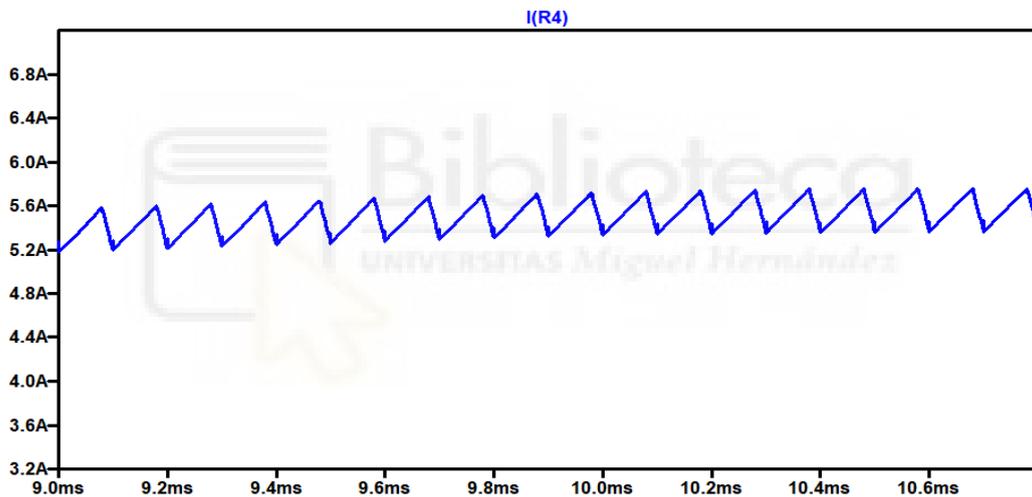
Tabla 46 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: 2n Inductor ETD

3. Prueba

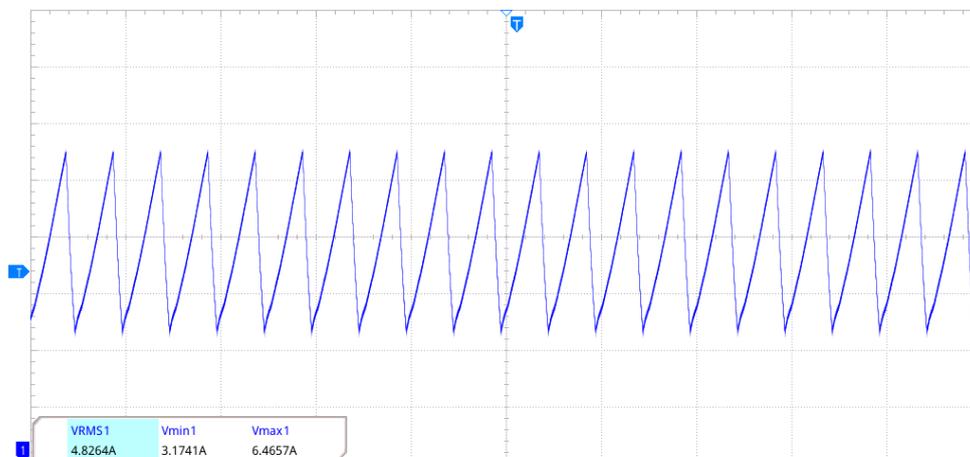
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	5.5A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	31.5V
Resistencia de Carga	4 Ω

Tabla 47 Condiciones Iniciales con la 3ª prueba. Modelo: 2n Inductor ETD



Captura 41



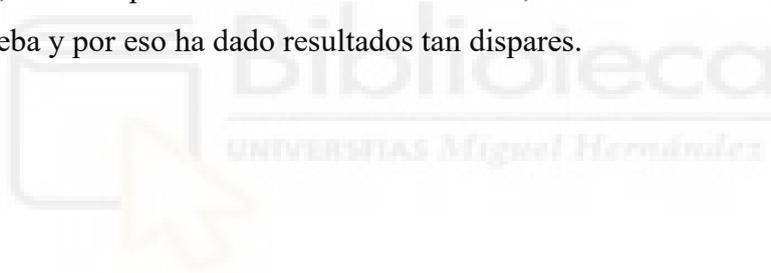
Captura 42

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	5.5136A	4.8264A
I_{MIN}	5.2914A	3.1741A
I_{MAX}	5.6792A	6.4657A

Tabla 48 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: 2n Inductor ETD

Observaciones:

Respecto a los resultados obtenidos con el segundo inductor con forma de núcleo ETD observamos un comportamiento similar al primer inductor con forma de núcleo ETD donde en las dos primeras pruebas tenemos una intensidad RMS que se asemeja bastante, pero con una amplitud pico-pico mucho mayor en el circuito real que en la simulación de LTspice. Respecto a la tercera prueba, observamos que, aun siendo la intensidad RMS medida en el circuito es menor que la medida en LTspice, la amplitud pico-pico medida en el circuito es de unos 3.2A mientras que en LTspice es de 0.4A. Creemos que, como ha pasado con el inductor anterior, este ha entrado en saturación con la última prueba y por eso ha dado resultados tan dispares.



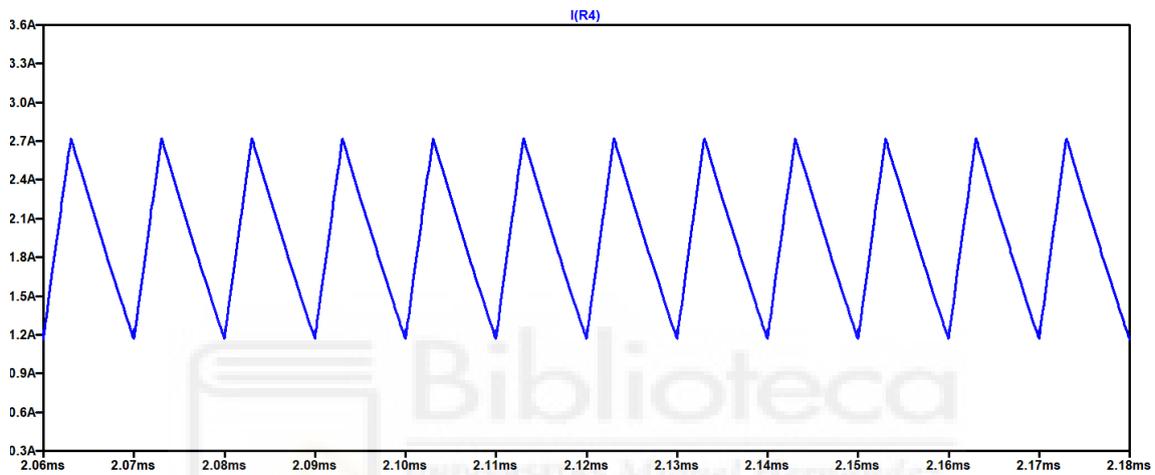
7.1.8 Tercer inductor con forma de núcleo ETD

1. Prueba

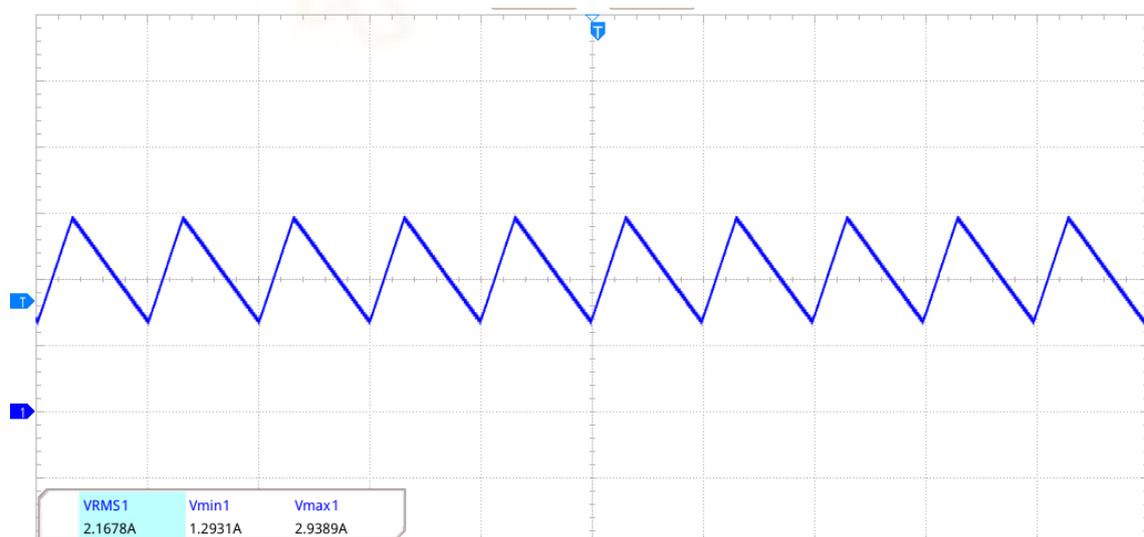
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.00A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	24V
Resistencia de Carga	3 Ω

Tabla 49 Condiciones Iniciales con la 1ª prueba. Modelo: 3r Inductor ETD



Captura 43



Captura 44

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	1.9869A	2.1678A
I_{MIN}	1.1662A	1.2931A
I_{MAX}	2.7180A	2.9389A

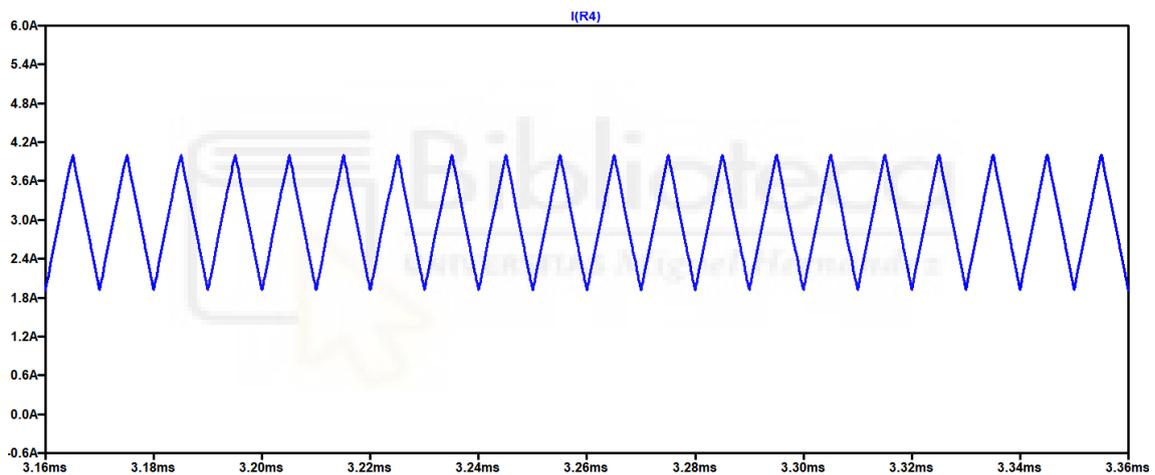
Tabla 50 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 1ª prueba. Modelo: 3r Inductor ETD

2. Prueba

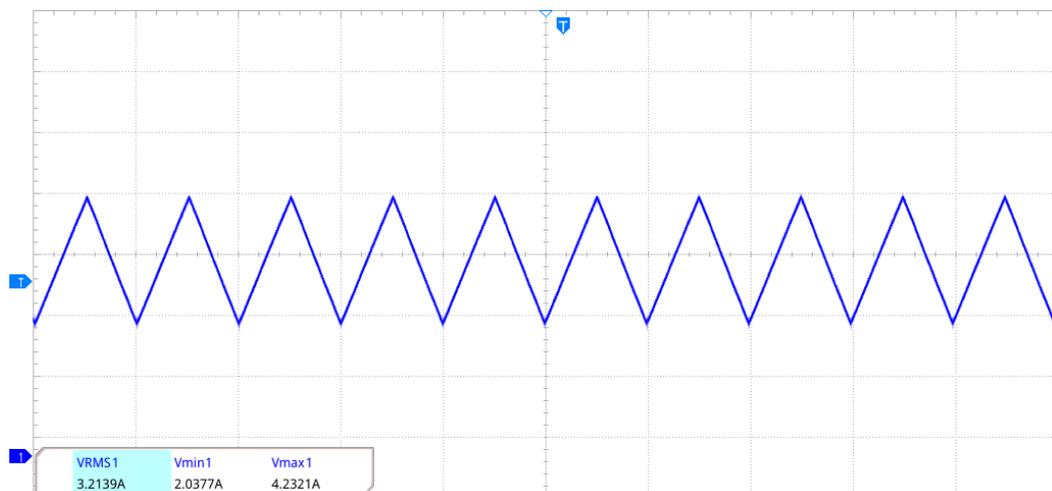
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	3.00A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	27.5V
Resistencia de Carga	4 Ω

Tabla 51 Condiciones Iniciales con la 2ª prueba. Modelo: 3r Inductor ETD



Captura 45



Captura 46

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	3.0274A	3.2139A
I_{MIN}	1.9105A	2.0377A
I_{MAX}	4.0212A	4.2321A

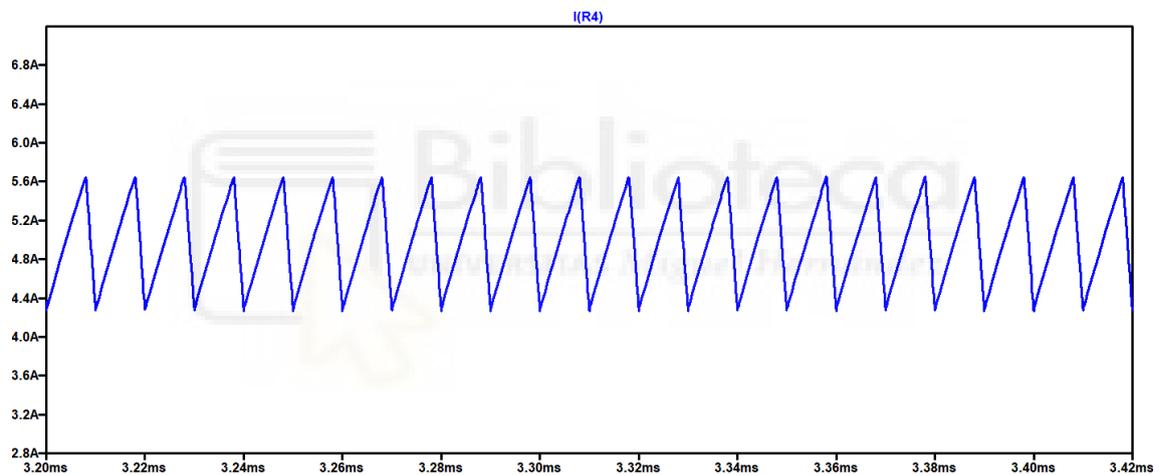
Tabla 52 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 2ª prueba. Modelo: 3r Inductor ETD

3. Prueba

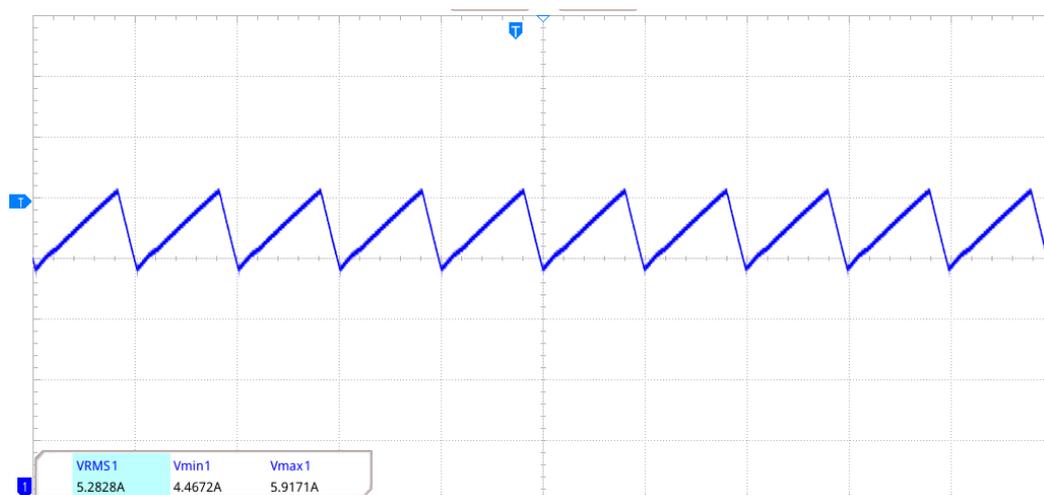
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	5.00A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	28.5V
Resistencia de Carga	4 Ω

Tabla 53 Condiciones Iniciales con la 3ª prueba. Modelo: 3r Inductor ETD



Captura 47



Captura 48

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	4.9937A	5.2828A
I_{MIN}	4.2835A	4.4672A
I_{MAX}	5.6521A	5.9171A

Tabla 54 Comparativa Intensidad entre Modelo Básico y Circuito Real con la 3ª prueba. Modelo: 3r Inductor ETD

Observaciones:

Respecto a los resultados obtenidos con el tercer y último inductor con forma de núcleo ETD observamos todo lo contrario que ocurría con los dos primeros inductores con forma de núcleo ETD. En este caso, tanto la intensidad RMS como la amplitud pico-pico de la señal de corriente obtenida es bastante pareja en LTspice como en la medida real y no presenta esa disparidad tan abultada que veíamos con intensidades más elevadas. Esto puede indicarnos que la intensidad de saturación de este inductor sea mucho más mayor que los dos anteriores y, por eso hemos tenido resultados tan diferentes anteriormente y resultados tan similares con este último inductor.



7.2. Pruebas de validación con el modelo avanzado

Para realizar las pruebas con el modelo avanzado hemos utilizado la segunda bobina con forma de núcleo ETD ya que es la única de la que teníamos la seguridad de que el núcleo del inductor era 3C96 y, por tanto, nos podría arrojar resultados congruentes con los que comparar con los resultados obtenidos en el circuito de validación.



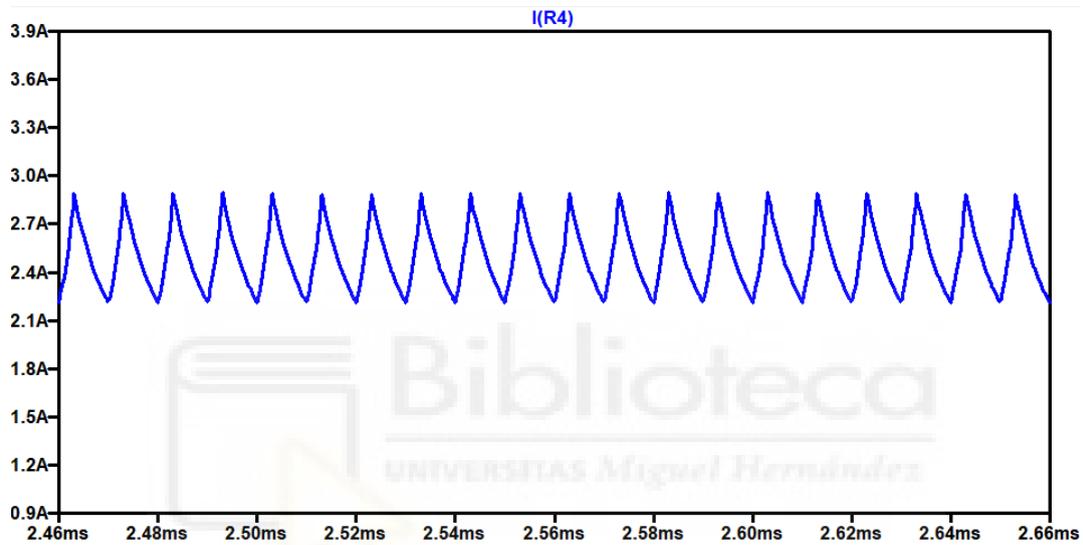
7.2.1 Segundo inductor con forma de núcleo ETD

1. Prueba

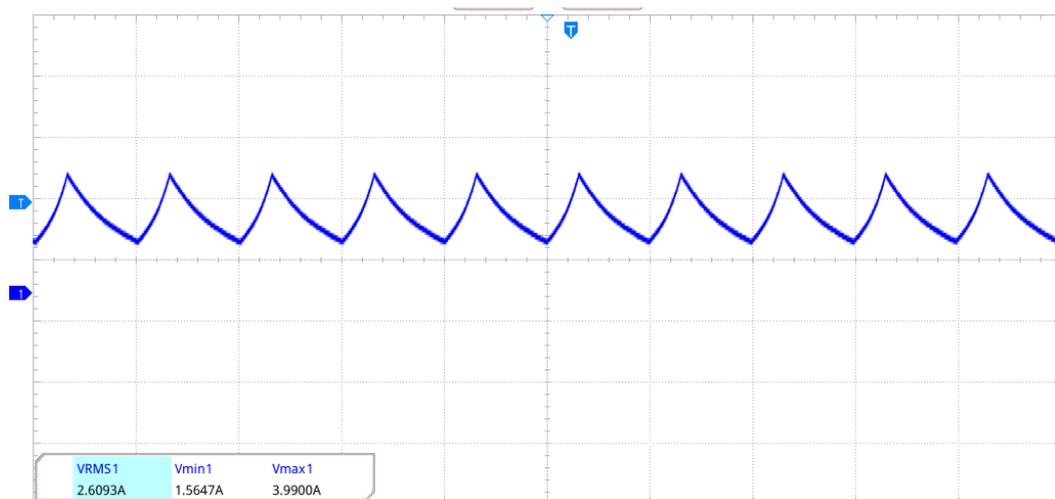
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	2.5A
Ciclo de trabajo (D)	0.3
Alimentación entrada (V_{in})	31V
Resistencia de Carga	3 Ω

Tabla 55 Condiciones Iniciales con la 1ª prueba. Modelo: 2n Inductor ETD



Captura 49



Captura 50

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	2.4906A	2.6093A
I_{MIN}	2.2351A	1.5647A
I_{MAX}	2.6245A	3.9900A

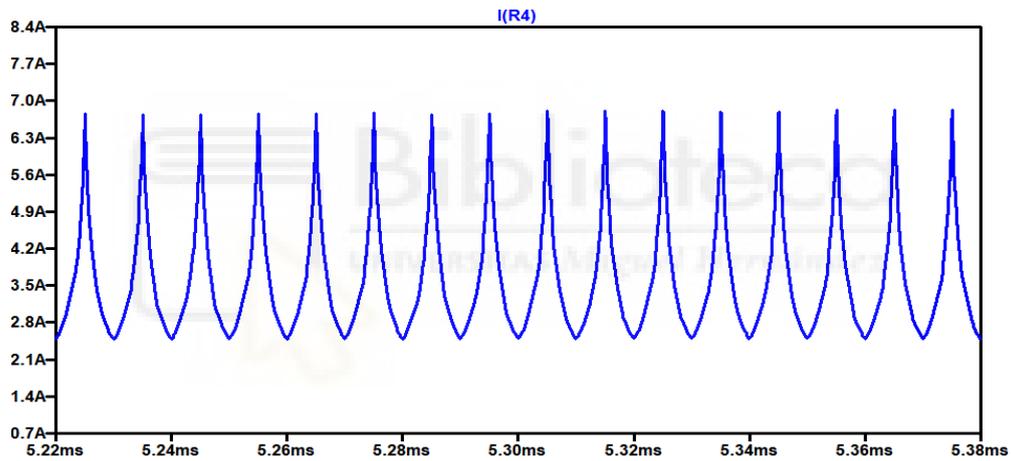
Tabla 56 Comparativa Intensidad entre Modelo Avanzado y Circuito Real en la 1ª prueba. Modelo: 2n Inductor ETD

2. Prueba

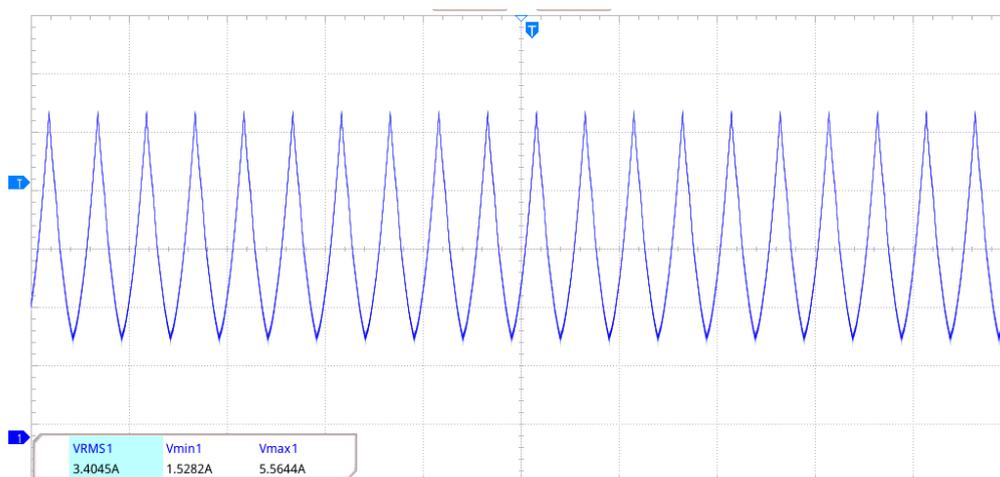
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	3.5A
Ciclo de trabajo (D)	0.5
Alimentación entrada (V_{in})	32.4V
Resistencia de Carga	4 Ω

Tabla 57 Condiciones Iniciales con la 2ª prueba. Modelo: 2n Inductor ETD



Captura 51



Captura 52

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	3.5332A	3.4045A
I_{MIN}	2.4976A	1.5280A
I_{MAX}	5.8114A	5.5644A

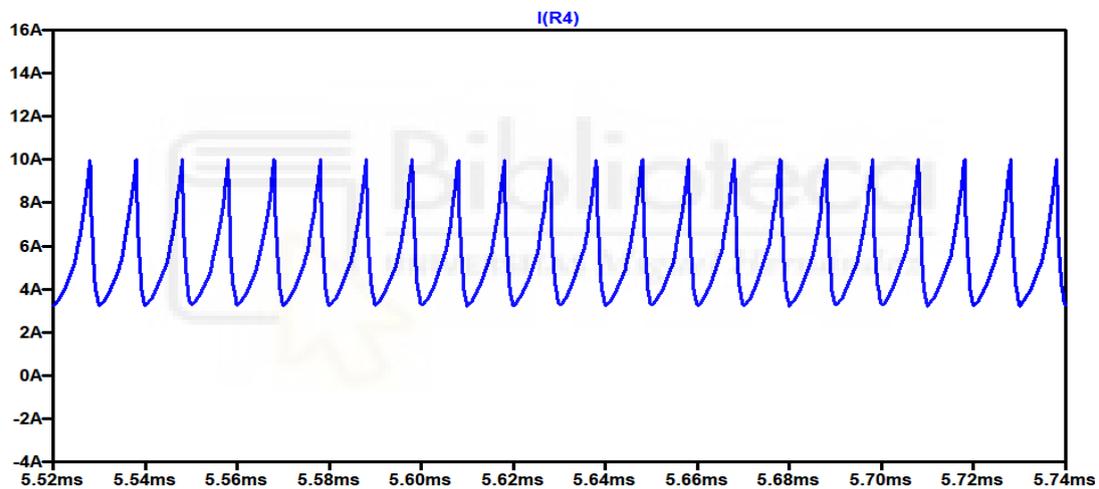
Tabla 58 Comparativa Intensidad entre Modelo Avanzado y Circuito Real en la 2ª prueba. Modelo: 2n Inductor ETD

3. Prueba

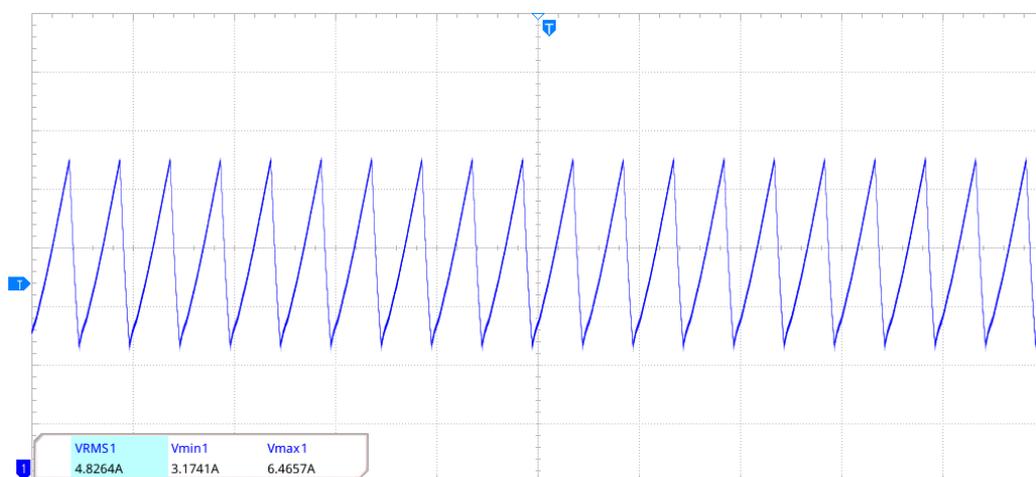
CONDICIONES DE TRABAJO

Intensidad (I_{RMS})	5.5A
Ciclo de trabajo (D)	0.8
Alimentación entrada (V_{in})	31.5V
Resistencia de Carga	4 Ω

Tabla 59 Condiciones Iniciales con la 3ª prueba. Modelo: 2n Inductor ETD



Captura 53



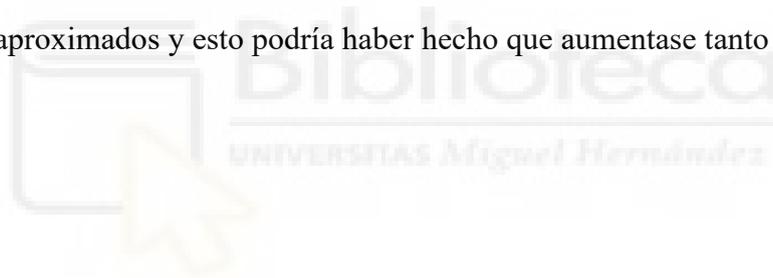
Captura 54

COMPARATIVA	LTSPICE	CIRCUITO REAL
I_{RMS}	5.6798A	4.8264A
I_{MIN}	3.2251A	3.1741A
I_{MAX}	9.9687A	6.4657A

Tabla 60 Comparativa Intensidad entre Modelo Avanzado y Circuito Real en la 3ª prueba. Modelo: 2n Inductor ETD

Observaciones:

Respecto a los resultados obtenidos con el segundo inductor con forma de núcleo ETD con el modelo avanzado observamos a primera vista una mayor similitud en las formas de onda en comparación con el modelo anterior. Por otro lado, en las dos primeras pruebas se puede observar que la intensidad RMS sí que se asemeja bastante y la amplitud pico-pico es bastante más aproximado, sobre todo en la segunda prueba. En cuanto a la tercera prueba sí que observamos como el modelo avanzado simula la saturación ya que la amplitud pico-pico es incluso mayor que la del propio inductor en el circuito real. Esta variación podría ser debido a los valores del ciclo de histéresis los cuales no son los exactos, sino aproximados y esto podría haber hecho que aumentase tanto la saturación.



8. CONCLUSIONES Y LÍNEAS FUTURAS

Con este trabajo se han logrado desarrollar dos aplicaciones funcionales con el software MATLAB con el que se han obtenido dos modelos equivalentes de inductores distintos. Con la primera aplicación, se genera un archivo que simula un modelo equivalente en frecuencia que tan sólo necesita un .csv de un barrido en frecuencia del inductor que se genera con el Bode100 para crearlo. Con la segunda aplicación, se genera un archivo que simula un modelo equivalente que tiene en cuenta la saturación del inductor. Esta aplicación también requiere de un archivo .csv del barrido de frecuencia del inductor a medir además del número de vueltas que tiene el inductor, la forma y tipo del núcleo y los valores de fuerza coercitiva, densidad de flujo de remanente y densidad de flujo en el punto de saturación del núcleo magnético del inductor.

Además, se ha desarrollado un circuito de validación, en este caso un circuito DC-DC reductor o *Buck*. Con este circuito se comparó la corriente a la salida del inductor obtenida tras medirlo con la intensidad simulada en LTspice. Los resultados obtenidos con el modelo básico nos han proporcionado una simulación acorde a la realidad para la mayoría de los inductores medidos en diferentes condiciones de trabajo y con distintas intensidades a las que se han ajustado dando resultados muy confiables con diferencias de 0.2 amperios en condiciones de trabajo en las que el inductor no llegaba a saturarse. Cuando este se producía, sí que veíamos una mayor disparidad de resultados entre LTspice y la medición del circuito, aunque era un suceso que cabía esperar ya que este modelo funciona principalmente para situaciones estables.

Por otro lado, los resultados con el modelo avanzado no han sido muy concluyentes ya que sí que teníamos una forma de onda más fiel a la que medíamos en el circuito, pero a la hora de buscar la saturación del inductor, la saturación sí que se ha producido como se veía en la prueba 3, pero llegaba a ser incluso superior a la medida en el circuito. Esto puede que haya sido posible a que la obtención de los valores del ciclo de histéresis no hayan sido los adecuados ya que el fabricante no nos lo proporcionaba. Para poder corroborar que la aplicación modelaba inductores con resultados que fuesen los esperables deberíamos haber hecho uso de inductores cuyos valores de densidad de flujo remanente, densidad de flujo en el punto de saturación y fuerza coercitiva fueran

datos.

Para este proyecto podríamos explorar distintas líneas de desarrollo a futuro ya que se podría implementar un programa que fuera capaz de leer las gráficas B-H de los datasheets de los inductores para no tener que depender de que los fabricantes especifiquen los valores de H_c , B_r y B_s . Por otro lado, se podría programar estas en un entorno como PYTHON que es más accesible y optimizado. Además, se podría realizar otros modelos que tuviesen en cuenta otros efectos que hagan variar el comportamiento del inductor como podría ser la temperatura.



9. BIBLIOGRAFÍA

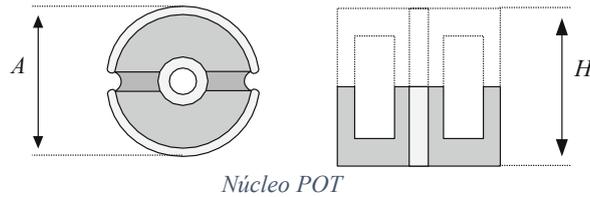
- [1] D. Erickson, Robert W.-Maksimović, *Fundamentals Of Power Electronics*, 3rd Editio. Springer Nature Switzerland AG, 2020.
- [2] C. W. T. McLyman, *Transformer and Inductor Design Handbook*, 4th Editio. CRC Press, 2017.
- [3] M. Bitschnau, “Power Inductor Modelling,” *Bode 100 Appl. Note*, 2017.
- [4] OMICRON, “Bode 100 User Manual,” vol. 1, p. 141, 2010.
- [5] “Apuntes Asignatura Electrónica de Potencia: Tema 5 – DC/DC;,” *UMH*.
- [6] J. Chan, John H. - Vladimirescu, Andrei - Gao, Xiao-Chun - Valainis, “Nonlinear Transformer Model for Circuit Simulation,” *IEEE Transactions on Computer-Aided Design*, 1991.
- [7] “The Chan model - LTwiki-Wiki for LTspice.” [Online]. Available: http://ltwiki.org/index.php?title=The_Chan_model. [Accessed: 10-Jun-2021].
- [8] “LTwiki-Wiki for LTspice.” [Online]. Available: http://ltwiki.org/index.php?title=Main_Page. [Accessed: 10-Jun-2021].
- [9] D. Kjellström Copete, “Diseño e implementación de un Shield PWM multifase y un software de control para PSOC 4,” 2018.
- [10] “MATLAB Documentation - MathWorks España.” [Online]. Available: <https://es.mathworks.com/help/matlab/>. [Accessed: 22-Jun-2021].
- [11] “Que son las bobinas y sus tipos - Ingeniería Mecafenix.” [Online]. Available: <https://www.ingmecafenix.com/electronica/bobinas/>. [Accessed: 17-Jul-2021].
- [12] N. Agnihotri, “What are the different types of buzzers?,” *Agil. Magn.*, 2020.
- [13] L. Technology, “LTspice 4 Getting Started Guide,” pp. 1–53, 2011.
- [14] “STP80NF06 - STB80NF06 STW80NF06,” *STMicroelectronics*, no. May, pp. 1–14, 2007.
- [15] C. Styles, “Mur1520 murb1520 murb1520-1,” *Int. Rectifier*, pp. 1–8, 2006.
- [16] “Shielded Power Inductors – RFS1317,” *Coilcraft*, pp. 10–12, 2017.
- [17] I. Guide, “Instructional Guide Inductors - Primer Inductors - Primer,” pp. 1–8, 2012.
- [18] “Inductor Current Measurement in Switched Power Supplies | Analog Devices.” [Online]. Available: <https://www.analog.com/en/analog-dialogue/raqs/raq-issue-170.html>. [Accessed: 05-Jun-2021].
- [19] “Power Inductors-DC1012 Series,” *Coilcraft*.
- [20] “Power Inductors-RFB17,” *Coilcraft*.
- [21] I. Q. Environmental et al., “Soft Ferrites and Accessories.”

10. ANEXOS

10.1 FORMAS GEOMÉTRICAS DEL NÚCLEO MAGNÉTICO

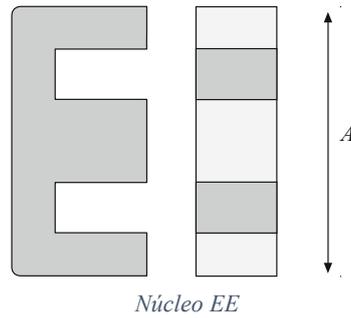
Anexo B, Fundamentals of Power Electronics [1]

10.1.1 Tabla Núcleo POT



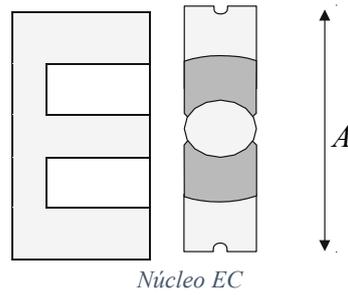
Core type	Geometrical constant	Geometrical constant	Cross-sectional area	Bobbin winding area	Mean length per turn	Magnetic path length	Thermal resistance	Core weight
(AH)	K_g	K_{gfe}	A_c	W_A	MLT	l_m	R_{th}	(g)
(mm)	cm^5	cm^x	(cm^2)	(cm^2)	(cm)	(cm)	($^{\circ}\text{C}/\text{W}$)	
704	$0.738 \cdot 10^{-6}$	$1.61 \cdot 10^{-6}$	0.070	$0.22 \cdot 10^{-3}$	1.46	1.0		0.5
905	$0.183 \cdot 10^{-3}$	$256 \cdot 10^{-6}$	0.101	0.034	1.90	1.26		1.0
1107	$0.667 \cdot 10^{-3}$	$554 \cdot 10^{-6}$	0.167	0.055	2.30	1.55		1.8
1408	$2.107 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	0.251	0.097	2.90	2.00	100	3.2
1811	$9.45 \cdot 10^{-3}$	$2.6 \cdot 10^{-3}$	0.433	0.187	3.71	2.60	60	7.3
2213	$27.1 \cdot 10^{-3}$	$4.9 \cdot 10^{-3}$	0.635	0.297	4.42	3.15	38	13
2616	$69.1 \cdot 10^{-3}$	$8.2 \cdot 10^{-3}$	0.948	0.406	5.28	3.75	30	20
3019	0.180	$14.2 \cdot 10^{-3}$	1.38	0.587	6.20	4.50	23	34
3622	0.411	$21.7 \cdot 10^{-3}$	2.02	0.748	7.42	5.30	19	57
4229	1.15	$41.1 \cdot 10^{-3}$	2.66	1.40	8.60	6.81	13.5	104

10.1.2 Tabla Núcleo EE



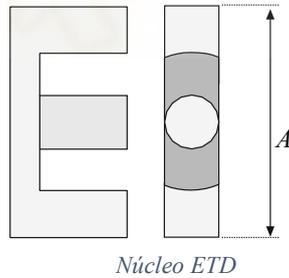
Core type	Geometrical constant	Geometrical constant	Cross-sectional area	Bobbin winding area	Mean length per turn	Magnetic path length	Core weight
(A) (mm)	K_g (cm ²)	K_{gfe} (cm ^x)	A_c (cm ²)	W_A (cm ²)	MLT (cm)	4_m (cm)	(g)
EE12	$0.731 \cdot 10^{-3}$	$0.458 \cdot 10^{-3}$	0.14	0.085	2.28	2.7	2.34
EE16	$2.02 \cdot 10^{-3}$	$0.842 \cdot 10^{-3}$	0.19	0.190	3.40	3.45	3.29
EE19	$4.07 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	0.23	0.284	3.69	3.94	4.83
EE22	$8.26 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$	0.41	0.196	3.99	3.96	8.81
EE30	$85.7 \cdot 10^{-3}$	$6.7 \cdot 10^{-3}$	1.09	0.476	6.60	5.77	32.4
EE40	0.209	$11.8 \cdot 10^{-3}$	1.27	1.10	8.50	7.70	50.3
EE50	0.909	$28.4 \cdot 10^{-3}$	2.26	1.78	10.0	9.58	116
EE60	1.38	$36.4 \cdot 10^{-3}$	2.47	2.89	12.8	11.0	135
EE70/68/19	5.06	$75.9 \cdot 10^{-3}$	3.24	6.75	14.0	18.0	280

10.1.3 Tabla Núcleo EC



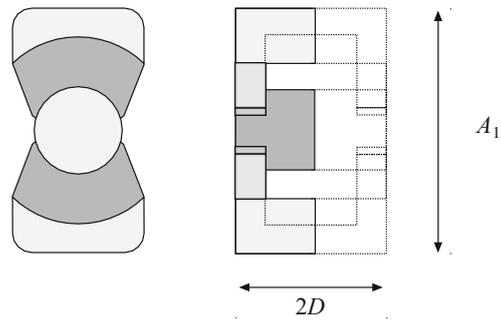
Core type	Geometrical constant	Geometrical constant	Cross-sectional area	Bobbin winding area	Mean length per turn	Magnetic path length	Thermal resistance	Core weight
(A) (mm)	K_g (cm ⁵)	K_{gfe} (cm ⁴)	A_c (cm ²)	W_A (cm ²)	MLT (cm)	4_m (cm)	R_{th} (°C/W)	(g)
EC35	0.131	$9.9 \cdot 10^{-3}$	0.843	0.975	5.30	7.74	18.5	35.5
EC41	0.374	$19.5 \cdot 10^{-3}$	1.21	1.35	5.30	8.93	16.5	57.0
EC52	0.914	$31.7 \cdot 10^{-3}$	1.80	2.12	7.50	10.5	11.0	111
EC70	2.84	$56.2 \cdot 10^{-3}$	2.79	4.71	12.9	14.4	7.5	256

10.1.4 Tabla Núcleo ETD



Core type	Geometrical constant	Geometrical constant	Cross-sectional area	Bobbin winding area	Mean length per turn	Magnetic path length	Thermal resistance	Core weight
(A) (mm)	K_g (cm ⁵)	K_{gfe} (cm ⁴)	A_c (cm ²)	W_A (cm ²)	MLT (cm)	4_m (cm)	R_{th} (°C/W)	(g)
ETD29	0.0978	$8.5 \cdot 10^{-3}$	0.76	0.903	5.33	7.20		30
ETD34	0.193	$13.1 \cdot 10^{-3}$	0.97	1.23	6.00	7.86	19	40
ETD39	0.397	$19.8 \cdot 10^{-3}$	1.25	1.74	6.86	9.21	15	60
ETD44	0.846	$30.4 \cdot 10^{-3}$	1.74	2.13	7.62	10.3	12	94
ETD49	1.42	$41.0 \cdot 10^{-3}$	2.11	2.71	8.51	11.4	11	124

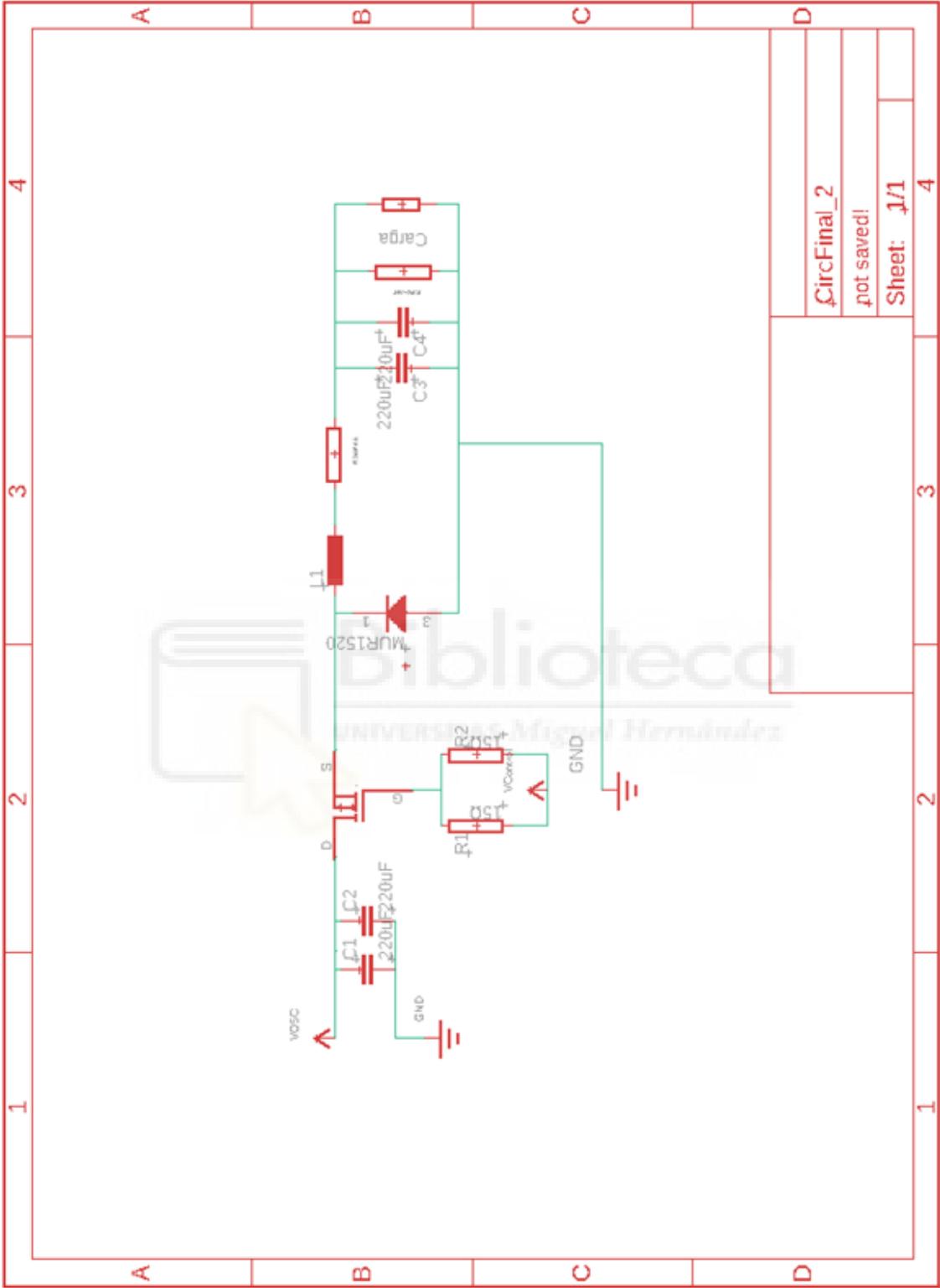
10.1.5 Tabla Núcleo PQ



Núcleo PQ

Core type	Geometrical constant	Geometrical constant	Cross-sectional area	Bobbin winding area	Mean length per turn	Magnetic path length	Core weight
$(A_1/2D)$ (mm)	K_g (cm^5)	K_{gfe} (cm^x)	A_c (cm^2)	W_A (cm^2)	MLT (cm)	ℓ_m (cm)	(g)
PQ20/16	$22.4 \cdot 10^{-3}$	$3.7 \cdot 10^{-3}$	0.62	0.256	4.4	3.74	13
PQ20/20	$33.6 \cdot 10^{-3}$	$4.8 \cdot 10^{-3}$	0.62	0.384	4.4	4.54	15
PQ26/20	$83.9 \cdot 10^{-3}$	$7.2 \cdot 10^{-3}$	1.19	0.333	5.62	4.63	31
PQ26/25	0.125	$9.4 \cdot 10^{-3}$	1.18	0.503	5.62	5.55	36
PQ32/20	0.203	$11.7 \cdot 10^{-3}$	1.70	0.471	6.71	5.55	42
PQ32/30	0.384	$18.6 \cdot 10^{-3}$	1.61	0.995	6.71	7.46	55
PQ35/35	0.820	$30.4 \cdot 10^{-3}$	1.96	1.61	7.52	8.79	73
PQ40/40	1.20	$39.1 \cdot 10^{-3}$	2.01	2.50	8.39	10.2	95

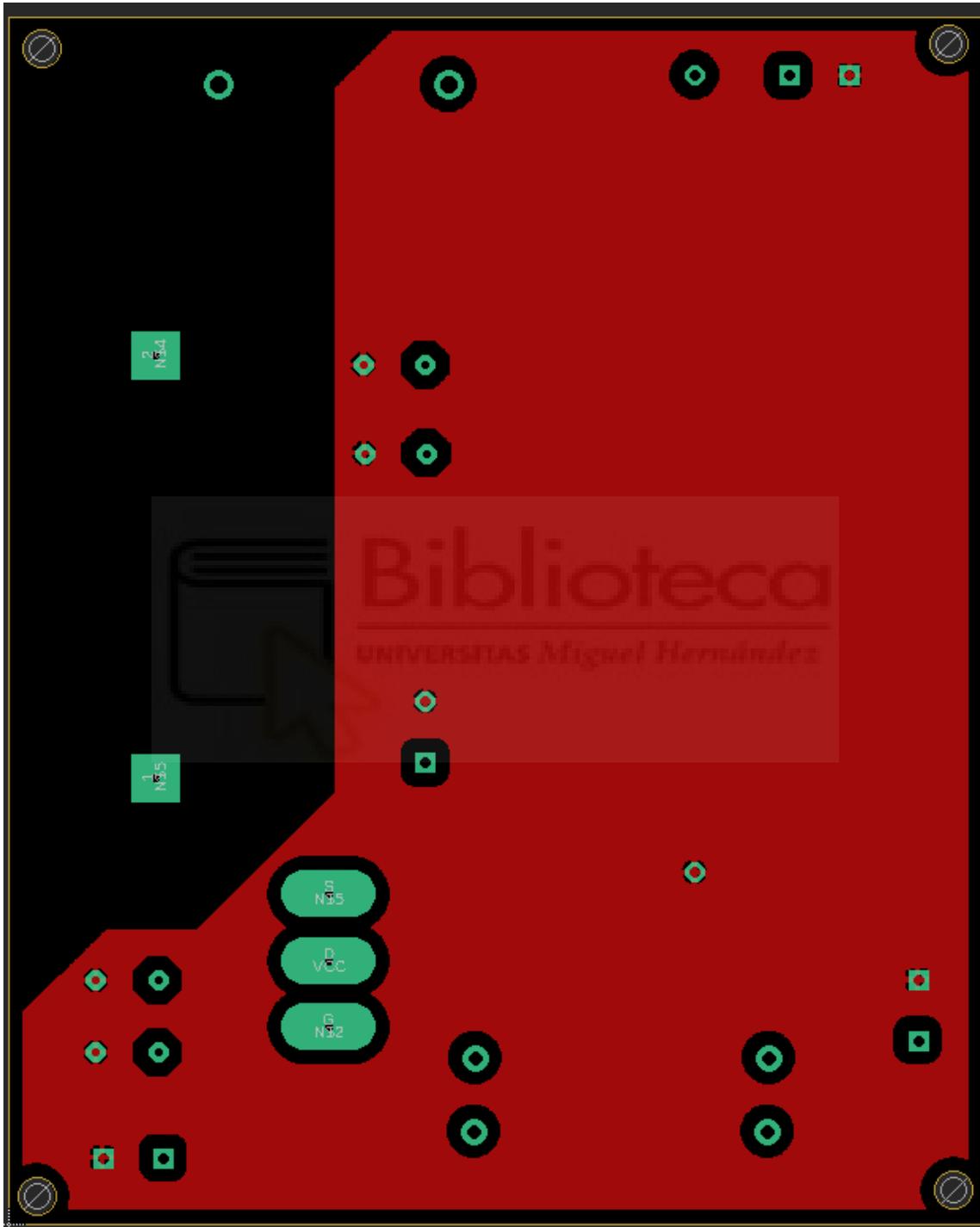
10.2 ESQUEMATICO DEL CONVERTIDOR DC-DC BUCK



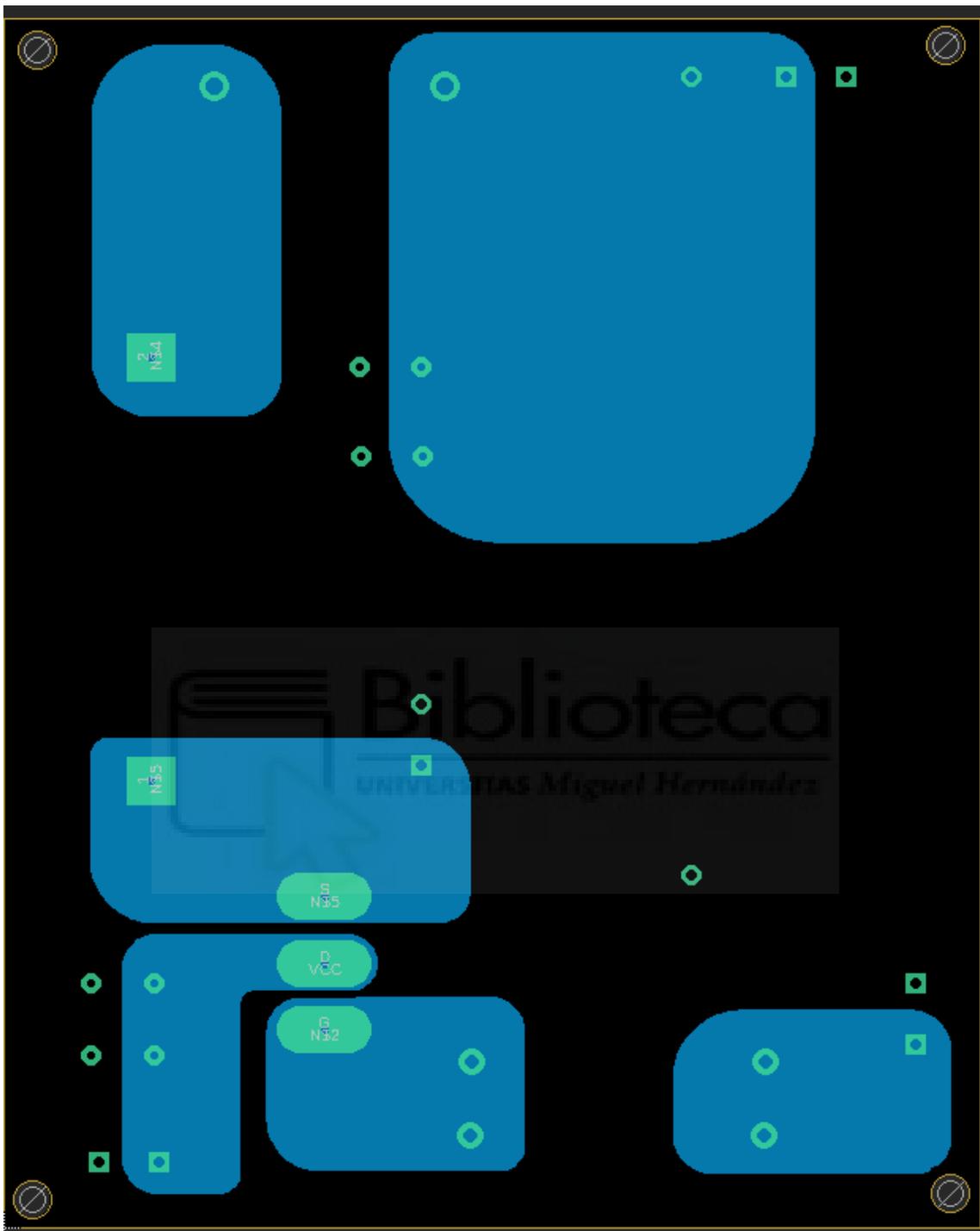
CircFinal_2
 .pot saved!
 Sheet: 1/1

10.3 CAPAS DEL DISEÑO DEL CIRCUITO *BUCK*

Cara TOP del circuito:



Cara BOTTOM del circuito:



10.4 CÓDIGO APLICACIÓN INDUCTOR BÁSICO

```
classdef AppModeloEquivBobina < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    UIFigure      matlab.ui.Figure
    Image         matlab.ui.control.Image
    Label         matlab.ui.control.Label
    Calculo       matlab.ui.control.Button
    ModeloCircuita1EquivalenteBobinaLabel  matlab.ui.control.Label
    Componentes  matlab.ui.control.Table
    ArchivoLabel  matlab.ui.control.Label
    Archivo       matlab.ui.control.EditField
    BuscarButton  matlab.ui.control.Button
    Magnitud     matlab.ui.control.UIAxes
    Fase         matlab.ui.control.UIAxes
end

properties (Access = private)
    archivocsv='';
end

% Callbacks that handle component events
methods (Access = private)

% Button pushed function: BuscarButton
function Buscar(app, event)

    [app.archivocsv, direccion] = uigetfile({'*.csv'});

    dircompleta=strcat(direccion,app.archivocsv);

    app.Archivo.Value=dircompleta;

end

% Button pushed function: Calculo
function Calcular(app, event)
    x=app.Archivo.Value;
    x=readtable(x);

    frec=table2array(x(:,1));
    mag=table2array(x(:,2));
    fase=table2array(x(:,3));

%Graficar
    plot(app.Magnitud, frec, mag);
```

```

plot(app.Fase, freq, fase);

%Cálculo Rs
r=find(freq<10); % Matriz valores de freq <10Hz
frsmax=max(r); % Último valor menos que 10 de la
matriz para hacer la media
Rs=mean(mag(1:frsmax)); % Media de la columna de magnitud
hasta el ultimo valor <10
app.Componentes.Data(1,1)=Rs;

%Cálculo Rp
Rp=max(mag(:,1)); % Valor maximo frecuencia resonancia no
hay imaginarios
app.Componentes.Data(1,2)=Rp;

%Cálculo L
Lfase=find(fase(:,1)>88.5); % Matriz con posiciones de
valores de fase >88,5º
Lfasemin=Lfase(1); % Posiciones max y min
Lfasemax=max(Lfase);
Zl=mag(Lfasemin:Lfasemax,1); % Valores de impedancia de la
bobina desde 10Hz a ...
Fs=freq(Lfasemin:Lfasemax,1); % Valores de frecuencia
correspondientes a impedancias
Lmatr=(1/(2*pi))*(Zl./Fs); % Matriz con su inductancia
correspondiente
L=mean(Lmatr); % Valor de L desde 10Hz
app.Componentes.Data(1,3)=L;

%Cálculo Cp
%Cálculo freq fase=0
Neg=find(fase(:,1)<0); % Primer valor negativo (el punto
anterior se encuentra antes de 0º)
x1=freq((Neg(1)-1),1); % Valores de freq y fase antes
de 0º

y1=fase((Neg(1)-1),1);
x2=freq((Neg(1)),1); % Valores de freq y fase despues de
0º

y2=fase((Neg(1)),1);
Pend=(y1-y2)/(x1-x2); %pendiente recta
f0=-y1/Pend)+x1; %valor de Frec para fase=0

Cp=1/(4*pi^2*f0^2*L);%Valor de C
app.Componentes.Data(1,4)=Cp;

%LTspice
% texto=sprintf('L=%g\nRser=%g\nRpar=%g\nCpar=%g',L,Rs,Rp,Cp);

%Crear Inductor LTSpice
ind=app.archivocsv;
ind=strsplit(ind, '.'); %Como el nombre es x.csv, separa este
en x y nombre y usa como delimitador el .

```

indref=sprintf("%s",string(ind(1,1))); %Ind es un conjunto de celdas por lo que hay que convertir la primera, que es la que tiene el nombre del inductor, en un vector

```
%Definición de los nombres de los ficheros
inductor_name_asc=strcat(indref, '.asc');
inductor_name_asy=strcat(indref, '.asy');
```

```
app.LTSpice.Value=inductor_name_asc;
```

```
%Creación del fichero del modelo
fileID = fopen(inductor_name_asc, 'w');
```

```
fprintf(fileID, 'Version 4\r\n');
fprintf(fileID, 'SHEET 1 880 680\r\n');
fprintf(fileID, 'WIRE 256 0 256 -32\r\n');
fprintf(fileID, 'WIRE 256 0 96 0\r\n');
fprintf(fileID, 'WIRE 496 0 256 0\r\n');
fprintf(fileID, 'WIRE 256 48 256 0\r\n');
fprintf(fileID, 'WIRE 96 112 96 0\r\n');
fprintf(fileID, 'WIRE 496 128 496 0\r\n');
fprintf(fileID, 'WIRE 256 160 256 128\r\n');
fprintf(fileID, 'WIRE 96 288 96 192\r\n');
fprintf(fileID, 'WIRE 256 288 256 240\r\n');
fprintf(fileID, 'WIRE 256 288 96 288\r\n');
fprintf(fileID, 'WIRE 496 288 496 192\r\n');
fprintf(fileID, 'WIRE 496 288 256 288\r\n');
fprintf(fileID, 'WIRE 256 320 256 288\r\n');
fprintf(fileID, 'FLAG 256 -32 L_1\r\n');
fprintf(fileID, 'IOPIN 256 -32 BiDir\r\n');
fprintf(fileID, 'FLAG 256 320 L_2\r\n');
fprintf(fileID, 'IOPIN 256 320 BiDir\r\n');
fprintf(fileID, 'SYMBOL res 80 96 R0\r\n');
fprintf(fileID, 'SYMATTR InstName R1\r\n');
fprintf(fileID, 'SYMATTR Value {Rpar}\r\n');
fprintf(fileID, 'SYMBOL res 240 32 R0\r\n');
fprintf(fileID, 'SYMATTR InstName R2\r\n');
fprintf(fileID, 'SYMATTR Value {Rser}\r\n');
fprintf(fileID, 'SYMBOL ind 240 144 R0\r\n');
fprintf(fileID, 'SYMATTR InstName L1\r\n');
fprintf(fileID, 'SYMATTR Value {Inductance}\r\n');
fprintf(fileID, 'SYMBOL cap 480 128 R0\r\n');
fprintf(fileID, 'SYMATTR InstName C1\r\n');
fprintf(fileID, 'SYMATTR Value {Cpar}\r\n');
fprintf(fileID, 'TEXT -500 144 Left 2 !.param
```

```
Rpar=%E\n.param Rser=%E\n.param Inductance=%E\n.param
Cpar=%E\r\n', Rp, Rs, L, Cp);
```

```
fclose(fileID);
```

```
%Creación del modelo del símbolo
fileID2 = fopen(inductor_name_asy, 'w');
```

```
fprintf(fileID2, 'Version 4\r\n');
fprintf(fileID2, 'SymbolType BLOCK\r\n');
```

```

        fprintf(fileID2, 'ARC Normal -80 1 -47 33 -43 17 -85
15\r\n');
        fprintf(fileID2, 'ARC Normal -47 1 -16 33 -10 16 -55
16\r\n');
        fprintf(fileID2, 'ARC Normal -16 1 17 33 22 15 -25
18\r\n');

        fprintf(fileID2, 'WINDOW 0 -44 -33 Bottom 2\r\n');
        fprintf(fileID2, 'PIN -80 16 RIGHT 8\r\n');
        fprintf(fileID2, 'PINATTR PinName L_1\r\n');
        fprintf(fileID2, 'PINATTR SpiceOrder 1\r\n');
        fprintf(fileID2, 'PIN 16 16 LEFT 8\r\n');
        fprintf(fileID2, 'PINATTR PinName L_2\r\n');
        fprintf(fileID2, 'PINATTR SpiceOrder 2\r\n');

        fclose(fileID2);

    return

end

end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 640 480];
app.UIFigure.Name = 'UI Figure';

% Create Image
app.Image = uiimage(app.UIFigure);
app.Image.Position = [32 63 241 194];
app.Image.ImageSource = 'inductormodel.png';

% Create Label
app.Label = uilabel(app.UIFigure);
app.Label.Position = [293 441 25 22];
app.Label.Text = '';

% Create Calculo
app.Calculo = uibutton(app.UIFigure, 'push');
app.Calculo.ButtonPushedFcn = createCallbackFcn(app,
@Calcular, true);
app.Calculo.Position = [465 378 100 22];

```

```

app.Calculo.Text = 'Calcular';

% Create ModeloCircuitalequivalenteBobinaLabel
app.ModeloCircuitalequivalenteBobinaLabel =
uilabel(app.UIFigure);
app.ModeloCircuitalequivalenteBobinaLabel.HorizontalAlignment
= 'center';
app.ModeloCircuitalequivalenteBobinaLabel.FontSize = 15;
app.ModeloCircuitalequivalenteBobinaLabel.FontWeight =
'bold';
app.ModeloCircuitalequivalenteBobinaLabel.Position = [1 423
640 40];
app.ModeloCircuitalequivalenteBobinaLabel.Text = {'Modelo
Circuitalequivalente Bobina'; ''; ''};

% Create Componentes
app.Componentes = uitable(app.UIFigure);
app.Componentes.ColumnName = {'Rs'; 'Rp'; 'L'; 'Cp'};
app.Componentes.RowName = {};
app.Componentes.Position = [1 280 302 59];

% Create ArchivoLabel
app.ArchivoLabel = uilabel(app.UIFigure);
app.ArchivoLabel.HorizontalAlignment = 'right';
app.ArchivoLabel.Position = [162 378 46 22];
app.ArchivoLabel.Text = {'Archivo'; ''};

% Create Archivo
app.Archivo = uieditfield(app.UIFigure, 'text');
app.Archivo.Position = [217 378 191 22];

% Create BuscarButton
app.BuscarButton = uibutton(app.UIFigure, 'push');
app.BuscarButton.ButtonPushedFcn = createCallbackFcn(app,
@Buscar, true);
app.BuscarButton.Position = [28 378 100 22];
app.BuscarButton.Text = {'Buscar '; ''};

% Create Magnitud
app.Magnitud = uiaxes(app.UIFigure);
title(app.Magnitud, 'Magnitud')
xlabel(app.Magnitud, 'Frecuencia (Hz)')
ylabel(app.Magnitud, 'H|(jw)|')
app.Magnitud.PlotBoxAspectRatio = [2.14285714285714 1 1];
app.Magnitud.XMinorTick = 'on';
app.Magnitud.YMinorTick = 'on';
app.Magnitud.XScale = 'log';
app.Magnitud.YScale = 'log';
app.Magnitud.Position = [317 184 324 185];

```

```

    % Create Fase
    app.Fase = uiaxes(app.UIFigure);
    title(app.Fase, 'Fase')
    xlabel(app.Fase, 'Frecuencia (Hz)')
    ylabel(app.Fase, 'Grados(°)')
    app.Fase.PlotBoxAspectRatio = [2.20779220779221 1 1];
    app.Fase.XMinorTick = 'on';
    app.Fase.XScale = 'log';
    app.Fase.Position = [317 -12 324 185];

    % Show the figure after all components are created
    app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = AppModeloEquivBobina

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargin == 0
            clear app
        end
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end

```

10.5 CÓDIGO APLICACIÓN INDUCTOR AVANZADO

```
classdef app1 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure
        Image              matlab.ui.control.Image
        Calcular           matlab.ui.control.Button
        BrTLabel           matlab.ui.control.Label
        Br                  matlab.ui.control.NumericEditField
        HcAMLabel          matlab.ui.control.Label
        Hc                  matlab.ui.control.NumericEditField
        BsTLabel           matlab.ui.control.Label
        Bs                  matlab.ui.control.NumericEditField
        NcleoMagnticoLabel matlab.ui.control.Label
        Nucleo              matlab.ui.control.ListBox
        MaterialdelNcleoListBoxLabel matlab.ui.control.Label
        MaterialdelNcleoListBox matlab.ui.control.ListBox
        AplicacinModeloAvanzadoInductorLabel matlab.ui.control.Label
        Buscar              matlab.ui.control.Button
        ArchivoEditFieldLabel matlab.ui.control.Label
        Archivo             matlab.ui.control.EditField
        TipodelNcleoLabel  matlab.ui.control.Label
        Tipo                matlab.ui.control.NumericEditField
    end

    properties (Access = private)

        % Tipo      Kg      Ac      Wa      MLT      Lm
        POT= [704, 0.738*10^-6, 0.07, 0.22*10^-3, 1.46, 1.0; 905,
0.183*10^-3, 0.101, 0.034, 1.9, 1.26; 1107, 0.667*10^-3, 0.167, 0.055,
2.3, 1.55; 1408, 2.107*10^-3, 0.251, 0.097, 2.9, 2; 1811, 9.45*10^-3,
0.433, 0.187, 3.71, 2.6; 2213, 27.1*10^-3, 0.635, 0.297, 4.42, 3.15;
2616, 69.1*10^-3, 0.948, 0.406, 5.28, 3.75; 3019, 0.18, 1.38, 0.587, 6.2,
4.5; 3622, 0.411, 2.02, 0.748, 7.42, 5.3; 4229, 1.15, 2.66, 1.4, 8.6,
6.81];
        EE= [12, 0.731*10^-3, 0.14, 0.085, 2.28, 2.7; 16, 2.02*10^-3,
0.19, 0.19, 3.4, 3.45; 19, 4.07*10^-3, 0.23, 0.284, 3.69, 3.94; 22,
8.26*10^-3, 0.41, 0.196, 3.99, 3.96; 30, 85.7*10^-3, 1.09, 0.476, 6.6,
5.77; 40, 0.209, 1.27, 1.1, 8.5, 7.7; 50, 0.909, 2.26, 1.78, 10, 9.58;
60, 1.38, 2.47, 2.89, 12.8, 11; 70, 5.06, 3.24, 6.75, 14, 18] ;
        EC= [35, 0.131, 0.843, 0.975, 5.3, 7.74; 41, 0.374, 1.21, 1.35,
5.3, 8.93; 52, 0.914, 1.8, 2.12, 7.5, 10.5; 70, 2.84, 2.79, 4.71, 12.9,
14.4];
        ETD= [29, 0.0978, 0.76, 0.903, 5.33, 7.2; 34, 0.193, 0.97, 1.23,
6, 7.86; 39, 0.397, 1.25, 1.74, 6.86, 9.21; 44, 0.846, 1.74, 2.13, 7.62,
10.3; 49, 1.42, 2.11, 2.71, 8.51, 11.4];
        PQ= ['20/16', 22.4*10^-3, 0.62, 0.256, 4.4, 3.74; '20/20',
33.6*10^-3, 0.62, 0.384, 4.4, 4.54; '26/20', 83.9*10^-3, 1.19, 0.333,
5.62, 4.63; '26/25', 0.125, 1.18, 0.503, 5.62, 5.55; '32/20', 0.203, 1.7,
```

```
0.471, 6.71, 5.55; '32/30', 0.384, 1.61, 0.995, 6.71, 7.46; '35/35',
0.820, 1.96, 1.61, 7.52, 8.79; '40/40', 1.2, 2.01, 2.5, 8.39, 10.2];
```

```
    archivocsv='';
end
```

```
% Callbacks that handle component events
methods (Access = private)
```

```
% Button pushed function: Calcular
function CalcularButtonPushed(app, event)
```

```
    x=app.Archivo.Value;
    x=readtable(x);
```

```
    frec=table2array(x(:,1));
    mag=table2array(x(:,2));
    fase=table2array(x(:,3));
```

```
%Cálculo Rs
```

```
    r=find(frec<10); % Matriz valores de frec <10Hz
    frsmax=max(r); % Último valor menos que 10 de la
matriz para hacer la media
    Rs=mean(mag(1:frsmax)); % Media de la columna de magnitud
hasta el ultimo valor <10
```

```
%Cálculo L
```

```
    Lfase=find(fase(:,1)>88.5); % Matriz con posiciones de
valores de fase >88,5º
    Lfasemin=Lfase(1); % Posiciones max y min
    Lfasemax=max(Lfase);
    Zl=mag(Lfasemin:Lfasemax,1); % Valores de impedancia de la
bobina desde 10Hz a ...
    Fs=frec(Lfasemin:Lfasemax,1); % Valores de frecuencia
correspondientes a impedancias
    Lmatr=(1/(2*pi))*(Zl./Fs); % Matriz con su inductancia
correspondiente
    L=mean(Lmatr); % Valor de L desde 10Hz
```

```
p=1.724*10^-6; % Ω/cm Resistividad del cobre a T. ambiente
```

```
Ku=0.4; %Se usa como valor estandar a la hora de diseñar bobinas
```

```
u0=4*pi*10^-7; %H/m Permeabilidad
```

```
if app.Nucleo.Value=="POT"
```

```
    r=find(app.POT(:,1)==app.Tipo.Value);
```

```

                                % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
    if numel(r)>1
        kg= app.POT(r(1),2);          % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
        Ac=app.POT(r(1),3);
        Lm=app.POT(r(1),6);
    else
        kg= app.POT(r,2);            % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
        Ac=app.POT(r,3);
        Lm=app.POT(r,6);
    end

elseif app.Nucleo.Value=="EE"

    r=find(app.EE(:,1)==app.Tipo.Value);

                                % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
    if numel(r)>1
        kg= app.EE(r(1),2);          % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
        Ac=app.EE(r(1),3);
        Lm=app.EE(r(1),6);
    else
        kg= app.EE(r,2);            % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
        Ac=app.EE(r,3);
        Lm=app.EE(r,6);
    end

elseif app.Nucleo.Value=="EC"

    r=find(app.EC(:,1)==app.Tipo.Value);

                                % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
    if numel(r)>1
        kg= app.EC(r(1),2);          % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
        Ac=app.EC(r(1),3);
        Lm=app.EC(r(1),6);
    else
        kg= app.EC(r,2);            % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
        Ac=app.EC(r,3);
        Lm=app.EC(r,6);
    end

elseif app.Nucleo.Value=="ETD"

    r=find(app.ETD(:,1)==app.Tipo.Value);

```

```

                                % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
    if numel(r)>1
        kg= app.ETD(r(1),2);          % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
        Ac=app.ETD(r(1),3);
        Lm=app.ETD(r(1),6);
    else
        kg= app.ETD(r,2);            % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
        Ac=app.ETD(r,3);
        Lm=app.ETD(r,6);
    end

elseif app.Nucleo.Value=="PQ"

    r=find(app.PQ(:,1)==app.Tipo.Value);

                                % Como tiene que ser >= Kg, obtenemos un
vector con los valores de kg mayores que el kg calculado
    if numel(r)>1
        kg= app.PQ(r(1),2);          % Obtenemos el valor de
kg de la tabla y los distintos datos que necesitamos
        Ac=app.PQ(r(1),3);
        Lm=app.PQ(r(1),6);
    else
        kg= app.PQ(r,2);            % Obtenemos el valor de kg
de la tabla y los distintos datos que necesitamos
        Ac=app.PQ(r,3);
        Lm=app.PQ(r,6);
    end
end

lg=u0*Ac*app.N.Value^2*10^-4/L;

if app.MaterialdeINcleoListBox.Value== "Otro"

    app.Hc.Value=app.Hc.Value;
    app.Bs.Value=app.Bs.Value;
    app.Br.Value=app.Br.Value;

elseif app.MaterialdeINcleoListBox.Value== "Ferrita 3C96"

    app.Hc.Value=17.95;
    app.Br.Value=0.244;
    app.Bs.Value=0.445;

end

%Conversión de cm a metros

Ac=Ac*10^-4;
Lm=Lm*10^-2;

```

```

%Crear Inductor LTSpice
ind=app.archivocsv;
ind=strsplit(ind, '.'); %Como el nombre es x.csv, separa este en
"x" y "csv" y usa como delimitador el .
indref=sprintf("%s",string(ind(1,1))); %Ind es un conjunto de
celdas por lo que hay que convertir la primera, que es la que tiene el
nombre del inductor, en un vector

```

```

%Definición de los nombres de los ficheros

```

```

inductor_CHAN_name_asc=strcat(indref, '_CHAN.asc');
inductor_CHAN_name_asy=strcat(indref, '_CHAN.asy');

```

```

%Creación del fichero del modelo

```

```

fileID = fopen(inductor_CHAN_name_asc, 'w');

```

```

fprintf(fileID, 'Version 4\r\n');
fprintf(fileID, 'SHEET 1 880 680\r\n');
fprintf(fileID, 'WIRE 240 208 80 208\r\n');
fprintf(fileID, 'WIRE 496 208 320 208\r\n');
fprintf(fileID, 'FLAG 80 208 L_1\r\n');
fprintf(fileID, 'IOPIN 80 208 BiDir\r\n');
fprintf(fileID, 'FLAG 496 208 L_2\r\n');
fprintf(fileID, 'IOPIN 496 208 BiDir\r\n');
fprintf(fileID, 'SYMBOL ind 224 224 R270\r\n');
fprintf(fileID, 'WINDOW 0 32 56 VTop 2\r\n');
fprintf(fileID, 'WINDOW 3 5 56 VBottom 2\r\n');
fprintf(fileID, 'SYMATTR InstName L1\r\n');
fprintf(fileID, 'SYMATTR Value Hc=%E Bs=%E Br=%E A=%E Lm=%E
Lg=%E N=%E\r\n', app.Hc.Value, app.Bs.Value, app.Br.Value, Ac, Lm, lg, n);

```

```

fclose(fileID);

```

```

%Creación del modelo del símbolo

```

```

fileID2 = fopen(inductor_CHAN_name_asy, 'w');

```

```

fprintf(fileID2, 'Version 4\r\n');
fprintf(fileID2, 'SymbolType BLOCK\r\n');
fprintf(fileID2, 'ARC Normal -80 1 -47 33 -43 17 -85 15\r\n');
fprintf(fileID2, 'ARC Normal -47 1 -16 33 -10 16 -55 16\r\n');
fprintf(fileID2, 'ARC Normal -16 1 17 33 22 15 -25 18\r\n');
fprintf(fileID2, 'WINDOW 0 -44 -33 Bottom 2\r\n');
fprintf(fileID2, 'PIN -80 16 RIGHT 8\r\n');
fprintf(fileID2, 'PINATTR PinName L_1\r\n');
fprintf(fileID2, 'PINATTR SpiceOrder 1\r\n');
fprintf(fileID2, 'PIN 16 16 LEFT 8\r\n');
fprintf(fileID2, 'PINATTR PinName L_2\r\n');
fprintf(fileID2, 'PINATTR SpiceOrder 2\r\n');

```

```

fclose(fileID2);

```

```

end

% Button pushed function: Buscar
function BuscarButtonPushed(app, event)
    [app.archivocsv, direccion] = uigetfile({'*.csv'});

    dircompleta=strcat(direccion,app.archivocsv);

    app.Archivo.Value=dircompleta;
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 640 480];
app.UIFigure.Name = 'UI Figure';

% Create Image
app.Image = uiimage(app.UIFigure);
app.Image.Position = [427 220 241 121];
app.Image.ImageSource = 'CicloHisteresis.png';

% Create Calcular
app.Calcular = uibutton(app.UIFigure, 'push');
app.Calcular.ButtonPushedFcn = createCallbackFcn(app,
@CalcularButtonPushed, true);
app.Calcular.Position = [481 137 100 22];
app.Calcular.Text = 'Calcular';

% Create BrTLabel
app.BrTLabel = uilabel(app.UIFigure);
app.BrTLabel.HorizontalAlignment = 'right';
app.BrTLabel.Position = [256 315 36 22];
app.BrTLabel.Text = 'Br (T)';

% Create Br
app.Br = uieditfield(app.UIFigure, 'numeric');
app.Br.HorizontalAlignment = 'center';
app.Br.Position = [307 315 100 22];

```

```

% Create HcAMLabel
app.HcAMLabel = uilabel(app.UIFigure);
app.HcAMLabel.HorizontalAlignment = 'right';
app.HcAMLabel.Position = [236 224 56 22];
app.HcAMLabel.Text = 'Hc ( A/M)';

% Create Hc
app.Hc = uieditfield(app.UIFigure, 'numeric');
app.Hc.HorizontalAlignment = 'center';
app.Hc.Position = [307 224 100 22];

% Create BsTLabel
app.BsTLabel = uilabel(app.UIFigure);
app.BsTLabel.HorizontalAlignment = 'right';
app.BsTLabel.Position = [254 269 38 22];
app.BsTLabel.Text = 'Bs (T)';

% Create Bs
app.Bs = uieditfield(app.UIFigure, 'numeric');
app.Bs.HorizontalAlignment = 'center';
app.Bs.Position = [307 269 100 22];

% Create NcleoMagnticoLabel
app.NcleoMagnticoLabel = uilabel(app.UIFigure);
app.NcleoMagnticoLabel.HorizontalAlignment = 'right';
app.NcleoMagnticoLabel.Position = [25 153 102 22];
app.NcleoMagnticoLabel.Text = 'Núcleo Magnético';

% Create Nucleo
app.Nucleo = uilistbox(app.UIFigure);
app.Nucleo.Items = {'POT', 'EE', 'EC', 'ETD', 'PQ'};
app.Nucleo.Position = [27 50 100 84];
app.Nucleo.Value = 'POT';

% Create MaterialdelNcleoListBoxLabel
app.MaterialdelNcleoListBoxLabel = uilabel(app.UIFigure);
app.MaterialdelNcleoListBoxLabel.HorizontalAlignment =
'right';
app.MaterialdelNcleoListBoxLabel.Position = [10 245 105 71];
app.MaterialdelNcleoListBoxLabel.Text = 'Material del
Núcleo';

% Create MaterialdelNcleoListBox
app.MaterialdelNcleoListBox = uilistbox(app.UIFigure);
app.MaterialdelNcleoListBox.Items = {'Ferrita 3C96', 'Otro'};
app.MaterialdelNcleoListBox.Position = [130 253 100 41];
app.MaterialdelNcleoListBox.Value = 'Ferrita 3C96';

```

```

        % Create AplicacinModeloAvanzadoInductorLabel
        app.AplicacinModeloAvanzadoInductorLabel =
uilabel(app.UIFigure);
        app.AplicacinModeloAvanzadoInductorLabel.HorizontalAlignment
= 'center';
        app.AplicacinModeloAvanzadoInductorLabel.FontSize = 20;
        app.AplicacinModeloAvanzadoInductorLabel.FontWeight = 'bold';
        app.AplicacinModeloAvanzadoInductorLabel.Position = [107.5
420 373 34];
        app.AplicacinModeloAvanzadoInductorLabel.Text = 'Aplicación
Modelo Avanzado Inductor';

        % Create Buscar
        app.Buscar = uibutton(app.UIFigure, 'push');
        app.Buscar.ButtonPushedFcn = createCallbackFcn(app,
@BuscarButtonPushed, true);
        app.Buscar.Position = [25 376 100 22];
        app.Buscar.Text = 'Buscar';

        % Create ArchivoEditFieldLabel
        app.ArchivoEditFieldLabel = uilabel(app.UIFigure);
        app.ArchivoEditFieldLabel.HorizontalAlignment = 'right';
        app.ArchivoEditFieldLabel.Position = [245 376 46 22];
        app.ArchivoEditFieldLabel.Text = 'Archivo';

        % Create Archivo
        app.Archivo = uieditfield(app.UIFigure, 'text');
        app.Archivo.Position = [306 376 163 22];

        % Create TipodelNcleoLabel
        app.TipodelNcleoLabel = uilabel(app.UIFigure);
        app.TipodelNcleoLabel.HorizontalAlignment = 'right';
        app.TipodelNcleoLabel.Position = [204 112 88 22];
        app.TipodelNcleoLabel.Text = 'Tipo del Núcleo';

        % Create Tipo
        app.Tipo = uieditfield(app.UIFigure, 'numeric');
        app.Tipo.Position = [307 112 100 22];

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

```

```

% Construct app
function app = app1

    % Create UIFigure and components
    createComponents(app)

    % Register the app with App Designer
    registerApp(app, app.UIFigure)

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end

```

