

DISEÑO Y SIMULACIÓN DE UN CONVERTIDOR BUCK

”La motivación es lo que te ayuda a empezar. El hábito es lo que te mantiene ahí.”

SEBASTIÁN ADOLFO ALBORNOZ VILLAMIL

Código: 2205607

Ing. Electrónica

Universidad Industrial De Santander

JUAN ESTEBAN PINTO OROZCO

Código: 2215585

Ing. Electrónica

Universidad Industrial De Santander

Abstract—The development of this project focused on the creation of a DC-DC Buck converter, using both idealized models and models that reflect real operating conditions. To ensure the feasibility of its implementation, commercial components were selected, taking into account availability and design specifications. A comparison was made between an ideal model, where lossless components were assumed, and a realistic model, where voltage drops and power losses inherent to components such as the MOSFET and diodes were considered. This comparison was crucial to identify differences between theory and practice and allowed key adjustments to the design. Converter control is implemented using a PWM signal generated by an Arduino UNO board. Simulations played a fundamental role in validating the design, helping to adjust parameters such as duty cycle and switching frequency to optimize efficiency and keep voltage ripple within established limits.

Index Terms—DC-DC Buck Converter, Idealized and realistic models, Commercial components, Component losses, PWM signal, Arduino UNO, Simulations, Duty cycle, Switching frequency, Efficiency, Voltage ripple

I. INTRODUCCIÓN

En la industria moderna, los convertidores DC-DC son componentes esenciales y de alta eficiencia para ajustar niveles de tensión, facilitando el funcionamiento óptimo de diversos sistemas electrónicos. Su principio de operación se basa en la modulación de la forma de onda de la tensión o corriente de entrada para obtener una salida con características específicas de tensión y corriente. Existen distintas topologías de convertidores DC-DC, como los tipos Buck, Boost y Buck-Boost, cada uno con propiedades y aplicaciones específicas. En este proyecto, se llevó a cabo la investigación y diseño de un convertidor tipo Buck, encargado de reducir el nivel de tensión en la salida. El diseño requiere una fuente de alimentación, MOSFETs como interruptores de alta velocidad, diodos, capacitores, resistencias, un controlador PWM para regular el ciclo de trabajo y un circuito de retroalimentación para estabilizar el voltaje de salida. Mediante el ajuste del ciclo de trabajo a través del controlador PWM, se define el tiempo en que el interruptor permanece abierto o cerrado, lo que permite regular la tensión de salida según las especificaciones requeridas.

II. OBJETIVOS

A. Objetivo general

Diseñar, simular e implementar (opcionalmente) un convertidor de corriente continua tipo Buck, que suministre una tensión de salida de 12V a una resistencia de carga con una corriente de hasta 2,5 A, manteniendo una eficiencia superior al 90 % y una tensión de rizado pico-pico inferior a 50 mV, considerando un rango de tensión de entrada de 13 V a 20 V.

Objetivos específicos

- Realizar los cálculos teóricos necesarios para el diseño del convertidor Buck, considerando los parámetros de carga, el rango de tensión de entrada, y la eficiencia esperada.
- Validar el modelo teórico mediante simulaciones del comportamiento del convertidor Buck en un software especializado, asegurando que los parámetros de diseño establecidos se cumplen.
- Identificar y seleccionar los componentes reales adecuados para un análisis práctico del convertidor Buck.
- Realizar un análisis detallado del convertidor Buck en software, utilizando los componentes seleccionados, y considerando pérdidas y consumo de los elementos para aproximarse al comportamiento real.
- Optimizar el diseño del convertidor, realizando mejoras que permitan alcanzar la mejor versión posible, respetando los parámetros establecidos.
- Verificar que el nuevo diseño, considerando las pérdidas, cumpla con los parámetros inicialmente planteados.
- Automatizar el proceso de selección del ciclo de trabajo, permitiendo que el convertidor opere de manera autónoma.
- (Opcional) Implementar físicamente el convertidor, si se demuestra que el diseño es viable para su implementación, asegurando la capacidad de ajuste automático de la tensión de salida.
- (Opcional) Evaluar el rendimiento del convertidor implementado bajo distintas condiciones.

III. MARCO TEÓRICO

• **Principios de Operación de un Convertidor Buck:** El convertidor buck utiliza un interruptor (transistor MOSFET o IGBT), un diodo, un inductor y un capacitor para convertir una tensión DC de entrada a una tensión DC menor en su salida. Este proceso se realiza a través de la conmutación rápida del interruptor, que alterna entre los estados de encendido y apagado, controlando la energía transferida a la carga.

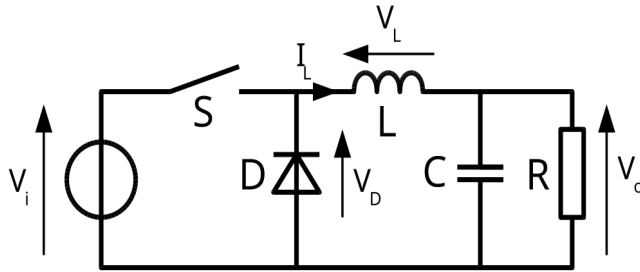


Fig. 1: Esquema básico de un convertidor Buck. El interruptor suele ser un MOSFET, IGBT o BJT.

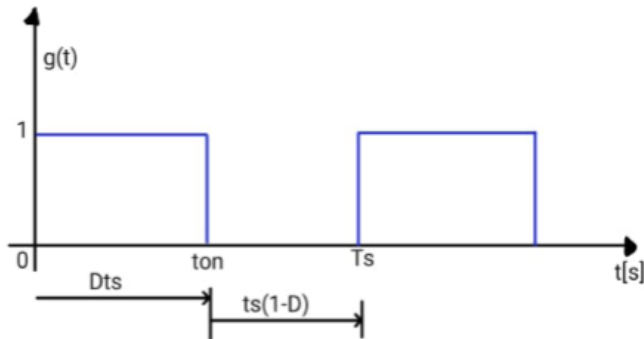


Fig. 2: señal de conmutación.

Donde T_s es periodo de conmutación, F_s frecuencia de conmutación, D es la relación entre el tiempo que el interruptor está encendido respecto al período total de conmutación y se definen así:

$$F_s = \frac{1}{T_s} \quad D = \frac{T_{on}}{T_s}$$

El valor de la tensión de salida V_{out} en un convertidor Buck ideal se relaciona con la tensión de entrada V_{in} , y el ciclo de trabajo D en un convertidor Buck ideal se expresa mediante la siguiente fórmula:

$$V_{out} = D \cdot V_{in}$$

En un convertidor real, las pérdidas de energía, como las pérdidas en el MOSFET y el diodo, afectan la eficiencia total.

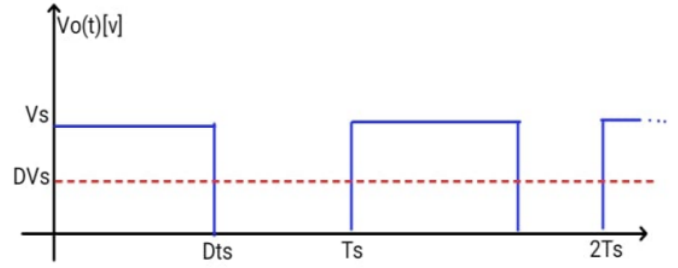


Fig. 3: voltaje de salida

En la imagen anterior se puede visualizar el voltaje de salida el cual se define mediante las siguiente expresión:

$$V_{oDI} = \frac{1}{T_s} \int_0^{DT_s} V_s dt = \frac{V_s \cdot DT_s}{T_s} = D \cdot V_s$$

• **Rizado en la Tensión:** El rizado en la tensión es causado por la carga y descarga del capacitor de salida y la corriente en el inductor. La magnitud del rizado puede aproximarse mediante la siguiente ecuación:

$$V_{rizado} = \frac{I_{out} \cdot (1 - D)}{f_s \cdot C_{out}}$$

Donde I_{out} es la corriente de salida f_s es la frecuencia de conmutación y C_{out} es el valor del capacitor de la salida. para mantener el rizado bajo los los 50mV, es necesario un capacitor con alto valor y baja ESR (Resistencia serie equivalente).

Para analizar este circuito debemos tener en cuenta las siguientes suposiciones:

- El circuito está en estado estable.
- Se considera un periodo de conmutación T . El interruptor está cerrado un tiempo DT y abierto un tiempo $(1 - D)T$ en cada periodo de conmutación.
- La corriente por el inductor L no cae a cero y siempre es positiva.
- El condensador de salida C es lo suficientemente grande tal que la tensión se considera constante de valor V_o .
- Los componentes son ideales.

Dadas las suposiciones mencionadas podemos analizar el circuito en cuestion en dos estados:

1) **circuito equivalente interruptor cerrado:**

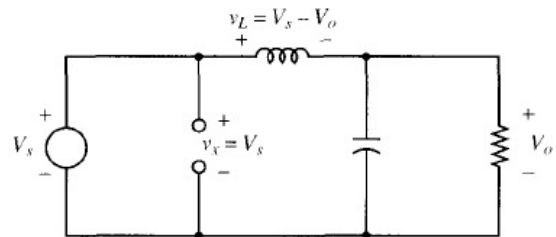


Fig. 4: circuito equivalente interruptor cerrado

Este estado se da para: $0 \leq t < DT$, obteniendo las siguientes ecuaciones:

$$V_L = V_s - V_o = L \frac{di_L}{dt}$$

$$\frac{di_L}{dt} = \frac{V_s - V_o}{L}$$

$$\frac{\Delta I_L}{dt} = \frac{V_s - V_o}{L}$$

$$\Delta I_L = \frac{(V_s - V_o)D \cdot T}{L}$$

2) Análisis circuito interruptor abierto:

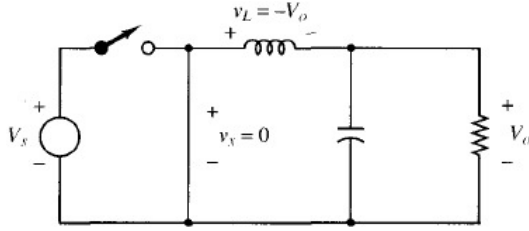


Fig. 5: circuito equivalente interruptor abierto

Este analisis se da en $DT \leq t < T$ obteniendo las siguientes ecuaciones:

$$V_L = -V_o = L \frac{di_L}{dt}$$

$$\frac{di_L}{dt} = \frac{-V_o}{L}$$

$$\frac{-\Delta I_L}{T - DT} = \frac{-V_o}{L}$$

$$\Delta I_L = \frac{V_o \cdot T(1 - D)}{L}$$

Igualando la ecuaciones de este y el equivalente anterior obtenemos:

$$\frac{(V_s - V_o)DT}{L} = \frac{V_o \cdot T(1 - D)}{L}$$

$$V_s D - V_o D = V_o - V_o D$$

$$V_o = V_s \cdot D \quad 0 \leq D \leq 1 \quad 0 \leq V_o \leq V_s$$

Partiendo de las ecuaciones anteriores, se puede deducir que los valores de inductancias y capacitancias corresponden a:

$$L_{min} \geq \frac{R(1 - D)}{2F_s}$$

Esta expresi3n indica la condici3n de operaci3n en modo continuo, el valor m3nimo que L puede tomar para que no pase a operaci3n discontinua, no se recomienda usar ese valor exacto.

$$L = \frac{V_o \cdot (1 - D)}{F_s \cdot \Delta_i}$$

En cambio el valor de L determinador por la ecuacion es el recomendado para un dise1o.

$$C_{min} \geq \frac{1 - D}{16_{min} \cdot F_s}$$

Esta expresion igual que la de la inductancia, determina el valor minimo de la capacitancia para que el circuito opere en estado continuo, tampoco es recomendable implementarlo especificamente con este valor.

$$C = \frac{V_o \cdot (1 - D)}{8 \cdot L \cdot f^2 \cdot \Delta V_c}$$

Esta ecuaci3n define el valor de la capacitancia en funci3n de los dem3s valores antes obtenidos, este es uno de los 3ltimos pasos de la definici3n de valores que determinan el rendimiento del dise1o.

IV. COMPONENTES DEL CIRCUITO

• **Componentes Comerciales:** Para elegir los componentes que se usar3n en el dise1o, se debe tener en cuenta lo siguiente:

- Deben soportar altas frecuencias, ya que la frecuencia de este dise1o rondar3 los 150 kHz.
- Deben funcionar de manera estable en el rango de temperaturas mencionado en los objetivos.

Para la se1al PWM, se consideraron las siguientes opciones: Arduino Uno, ESP32 y Raspberry Pi Pico. Se opt3 por el Arduino Uno por las siguientes razones:

- Facilidad de programaci3n: al tener familiaridad con el software, se convierte en una opci3n conveniente.
- Precio: esta tarjeta de desarrollo es considerablemente m3s econ3mica que la Raspberry Pi Pico, estando aproximadamente entre cinco mil y diez mil pesos colombianos por encima del Arduino.
- La tarjeta ESP32 se descart3 debido a que no representa una mejora respecto a lo que se necesita en este dise1o.

• **Driver de MOSFET:** Un driver es un circuito utilizado para controlar la conmutaci3n eficiente de un MOSFET. Este driver puede estar compuesto por transistores o componentes integrados, y su funci3n principal es gestionar el paso de corte a saturaci3n del MOSFET de manera 3ptima, lo que evita el sobrecalentamiento y permite manejar grandes corrientes de manera m3s segura y eficiente. En muchas aplicaciones de potencia, se utiliza para controlar tanto el lado alto como el bajo del puente MOSFET. En configuraciones de medio puente (un medio puente es una configuraci3n que utiliza dos transistores para controlar la corriente en una carga, como un motor o una etapa de potencia, permitiendo controlar tanto la magnitud como la direcci3n de la corriente a trav3s de se1ales PWM (Modulaci3n por Ancho de Pulso)), el driver puede incorporar una funci3n de bootstrap que eleva el voltaje en la puerta del MOSFET de canal N en el lado

alto del circuito, permitiendo que el transistor se encienda correctamente. Esto se logra a través del voltaje almacenado en un condensador de bootstrap, sumado al voltaje de la puerta. Adicionalmente, el driver ofrece protección y maneja mayores voltajes, superando las limitaciones de los MOSFETs de nivel lógico que no se abren completamente con los 5V de un microcontrolador como el Arduino.

El driver “IR2184” está diseñado específicamente para la conducción de medios puentes en aplicaciones electrónicas. Cuando la señal de entrada es alta, el IR2184 enciende el MOSFET superior y apaga el inferior. Por el contrario, cuando la señal de entrada es baja, enciende el MOSFET inferior y apaga el superior. Esta capacidad permite controlar el medio puente con una sola señal PWM, simplificando el diseño del circuito. Una de las ventajas destacadas del IR2184 es su control integrado de conducción cruzada, que evita que ambos MOSFETs se activen simultáneamente, lo cual podría causar un cortocircuito. Además, el IR2184 es valorado por su relación costo-rendimiento, siendo capaz de operar a frecuencias de hasta 300 kHz. También cuenta con un tiempo muerto interno fijo de 500 ns, lo que contribuye a su eficiencia y seguridad operativa. En cuanto a su aplicación, el IR2184 es ideal para aplicaciones de alta frecuencia debido a su rapidez en el encendido y apagado de los MOSFETs. Esto lo hace un componente valioso en desarrollos futuros que requieran un control eficiente y fiable. Su facilidad de uso y rendimiento notable lo hacen útil en una variedad de aplicaciones electrónicas avanzadas.

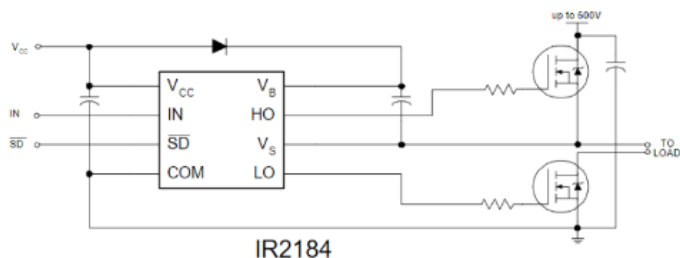


Fig. 6: Esquemático IR2184.

• Conexiones del driver:

- **VCC:** Este pin va a la fuente de alimentación, que provee el voltaje necesario para el funcionamiento del IR2184 (normalmente entre 10V y 20V).
- **IN:** Es la entrada de la señal PWM que controlará la conmutación de los MOSFETs. Esta señal puede provenir de un microcontrolador como el Arduino.
- **SD (Shutdown):** Este pin es para habilitar o deshabilitar el driver. Si se conecta a un nivel lógico bajo, el driver desactiva ambos MOSFETs. Para habilitarlo, conecta este pin a un nivel alto o a VCC.
- **COM:** Este es el pin de tierra común (Ground) del driver. Debe conectarse al potencial de tierra del sistema.
- **HO (High-side Output):** Este pin controla el MOSFET de "alto lado" (el MOSFET superior). La salida va conectada

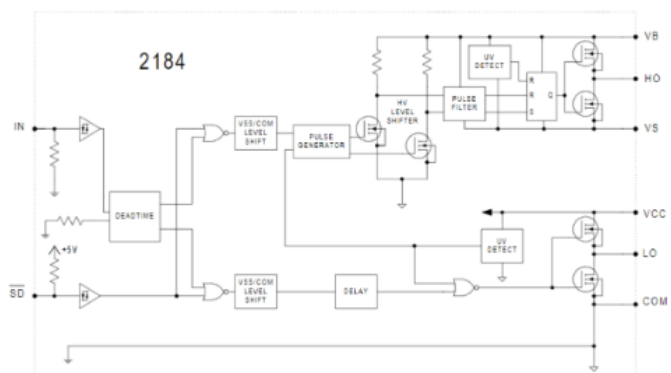


Diagrama de bloques del IR2184

Fig. 7: Diagrama de conexiones IR2184.

a la compuerta (gate) de este MOSFET a través de una resistencia.

- **LO (Low-side Output):** Este pin controla el MOSFET de "bajo lado" (el MOSFET inferior). La salida va conectada a la compuerta (gate) de este MOSFET a través de una resistencia.
- **VB:** Este pin es la fuente bootstrap, utilizada para incrementar el voltaje necesario para encender el MOSFET de alto lado. Se conecta a un capacitor bootstrap que carga a través de un diodo desde VCC.
- **VS:** Este pin se conecta al nodo común entre los dos MOSFETs (drain del MOSFET inferior y source del MOSFET superior). Este nodo es clave para la función bootstrap, ya que determina el voltaje de referencia del MOSFET de alto lado.
- **Diodo de conmutación rápida:** en este punto del diseño se obtuvieron dos candidatos que fueron el MBR 1645 y el MBR 1060 donde se decidió escoger el MBR 1060 debido a su capacidad de conmutación superior además el 1645 es complicado de adquirir en este país.
- **Mosfet de conmutación:** Para la elección de este componente se debió sacrificar rendimiento por economía, ya que el RD15 era una gran opción, pero tiene un costo elevado en comparación con el IRFZ44N, cuyas características son similares al primero mencionado y logra satisfacer las necesidades del diseño.

Características - IR2184

- Floating channel designed for bootstrap operation Fully operational to +600V Tolerant to negative transient voltage dV/dt immune
- Gate drive supply range from 10 to 20V
- Undervoltage lockout for both channels
- 3.3V and 5V input logic compatible
- Matched propagation delay for both channels
- Logic and power ground +/- 5V offset
- Lower dI/dt gate driver for better noise immunity
- Output source/sink current capability 1.4A/1.8A
- Also available LEAD-FREE (PbF)

Características - IR2104

- Floating channel designed for bootstrap operation Fully operational to +600V Tolerant to negative transient voltage dV/dt immune
- Gate drive supply range from 10 to 20V
- Undervoltage lockout
- 3.3V, 5V and 15V input logic compatible
- Cross-conduction prevention logic
- Internally set deadtime
- High side output in phase with input
- Shut down input turns off both channels
- Matched propagation delay for both channels
- Also available LEAD-FREE (PbF)

• **Inductores:** A partir de los cálculos teóricos, se ha definido que la inductancia necesaria para el diseño es de $100 \mu\text{H}$. En el mercado actual, un inductor de estas características, tiene un costo de \$6,800 COP. Este elemento es fundamental para el correcto funcionamiento del diseño y su precio se encuentra dentro de un rango razonable, sin superar los \$10,000 COP.



Fig. 8: Bobina $100\mu\text{H}$.

• **Capacitores:** A partir de los cálculos teóricos, se ha definido que la capacitancia necesaria para que el equipo funcione de manera óptima es de $25 \mu\text{F}$, pero por accesibilidad se tomara el valor de $47 \mu\text{F}$. En el mercado actual, un condensador electrolítico de estas características, con un voltaje de 160 V, tiene un costo aproximado de \$1,100 COP. Este componente es fundamental para garantizar el rendimiento del sistema y su adecuado funcionamiento bajo las especificaciones establecidas.

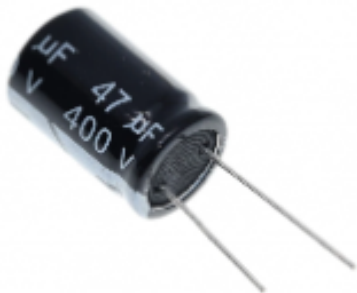


Fig. 9: Capacitor $47\mu\text{F}$.

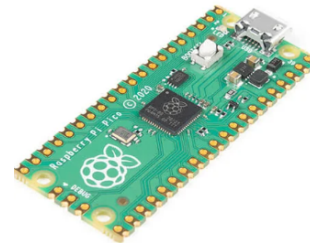


(a) RD15



(b) IRFZ44N

Fig. 10: Comparación RD15 y IRFZ44N.



(a) Raspberry Pi Pico



(b) Arduino Uno

Fig. 11: Microcontroladores.

• **Costo del diseño:** El diseño que se ha propuesto tendría un costo aproximado de entre sesenta mil y ochenta mil pesos colombianos, sin incluir los costos de envío. Este rango de precios varía en función de la calidad de fabricación de la placa de circuito impreso (PCB) y de otros factores relacionados, como la complejidad del diseño y los materiales utilizados. Cabe destacar que el precio final podría ser más alto, ya que se consultaron varios proveedores y cada uno de ellos tiene sus propias tarifas de envío. Además, es importante considerar que algunos fabricantes ofrecen opciones adicionales, como el ensamblaje de componentes o pruebas de funcionalidad, lo que también puede influir en el costo total del proyecto.

Por lo tanto, aunque la estimación inicial se sitúa en ese rango, es recomendable tener en cuenta todas las variables involucradas antes de tomar una decisión definitiva sobre el proveedor. La elección del fabricante no solo impactará el precio, sino también la calidad y el tiempo de entrega, factores

clave para el éxito de cualquier proyecto que requiera precisión en la construcción de la PCB.

los precios de los artículos necesarios para este proyecto se encuentran en el siguiente enlace:// https://drive.google.com/drive/folders/1BoTY2SPOy4LQH_UAk24OI_CQDTqPV7lo?usp=sharing

V. CONSIDERACIONES DE DISEÑO

El presente diseño del convertidor Buck se rige por una serie de parámetros definidos previamente, cuyo objetivo es garantizar que el dispositivo cumpla con las especificaciones establecidas. A continuación, se detallan los parámetros asignados que guían el desarrollo y la configuración del diseño:

Parámetro	Valor
Tensión de Salida	12 [V]
Corriente de Salida Máxima	2.5 [A]
Eficiencia	> 90%
Tensión de Rizado Pico a Pico	< 50 [mV]
Rango de la Tensión a la Entrada	13 [V] - 20 [V]

TABLE I: Parámetros del diseño del troceador tipo 'BUCK'.

Consideración	Valor
Ajuste de Tensión de Salida	Manual o Automático
Rango de la Temperatura Ambiente	-10°C a 50°C

TABLE II: Consideraciones en caso de implementación.

VI. CALCULOS TEÓRICOS CORRESPONDIENTES AL DISEÑO:

El planteamiento inicial para diseñar un convertidor buck se consideró a partir de un modelo ideal, esto con el objetivo de simplificar los cálculos básicos y proporcionar una base sólida. Este enfoque asume componentes sin pérdidas ni imperfecciones, lo que facilita la determinación de parámetros clave como el ciclo de trabajo, el valor del inductor y del condensador. Aunque este punto de partida es más fácil, es necesario realizar simulaciones para ajustar el diseño, incorporando pérdidas reales y otras imperfecciones del circuito, análisis que veremos más adelante en esta sección.

• **Ciclo de trabajo:** El primer parámetro que se determinará para comenzar con el diseño del convertidor será el ciclo de trabajo (D) con el que operará. En este caso, es necesario analizar los dos límites del diseño, ya que la fuente de entrada puede variar en un rango de valores, fluctuando entre un mínimo de 13 V y un máximo de 20 V. El porcentaje del ciclo de trabajo en cada uno de estos escenarios corresponden al siguiente cálculo.

a) *Caso 1: Tensión de entrada mínima (13 V)*

$$D_{\min} = \frac{V_{\text{out}}}{V_{\text{in}}} = \frac{12 \text{ V}}{13 \text{ V}} \approx 0.923$$

b) *Caso 2: Tensión de entrada máxima (20 V)*

$$D_{\max} = \frac{V_{\text{out}}}{V_{\text{in}}} = \frac{12 \text{ V}}{20 \text{ V}} \approx 0.6$$

El cálculo del ciclo de trabajo muestra que, a medida que aumenta la tensión de entrada, el convertidor Buck necesita operar menos tiempo para mantener una salida constante, lo que sugiere un uso más eficiente de energía bajo condiciones de mayor voltaje.

• **Frecuencia de conmutación:** Para determinar este parámetro debemos tener en claro sus afectaciones sobre el sistema en general, ya que impacta directamente en el rendimiento, la eficiencia y el diseño físico del circuito, a continuación, se detallan algunas de sus implicaciones.

- **Tamaño de componentes:** A frecuencias más altas, se pueden usar inductores y condensadores más pequeños, lo que permite un diseño más compacto, pero aumenta las pérdidas por conmutación.
- **Eficiencia:** Frecuencias más bajas disminuyen las pérdidas por conmutación y mejoran la eficiencia, aunque requieren componentes pasivos más grandes, lo que aumenta el tamaño del circuito.
- **Rendimiento térmico:** Las frecuencias más altas generan más calor debido a las mayores pérdidas por conmutación, lo que demanda una gestión térmica más efectiva.
- **Tipo de componentes:** MOSFETs rápidos permiten trabajar a frecuencias más altas, reduciendo el tamaño de los componentes pasivos sin sacrificar el rendimiento.

Para el diseño de este convertidor reductor se optó por trabajar con una frecuencia de conmutación de 150 KHz, valor que corresponde a un rango adecuado para aplicaciones de baja potencia como esta. Si bien frecuencias más altas pueden reducir aún más el tamaño de estos componentes, también incrementan las pérdidas por conmutación, afectando la eficiencia del sistema.

$$F_s = 150 \text{ kHz}$$

• **Cálculo de los Componentes L y C:** El cálculo del inductor y el condensador en un convertidor Buck es esencial para controlar la ondulación de la corriente y del voltaje. El valor del inductor se selecciona para mantener la corriente de ondulación dentro de un rango aceptable, lo cual depende del voltaje de entrada, el voltaje de salida, el ciclo de trabajo y la frecuencia de conmutación. Por otro lado, el condensador se dimensiona para asegurar que la ondulación del voltaje en la salida sea mínima y esté dentro de los límites deseados. Ambos componentes deben ser calculados cuidadosamente, ya que influyen directamente en la eficiencia y el rendimiento global del convertidor.

Primero, determinaremos el valor crítico del inductor y del condensador, que corresponde al valor mínimo que pueden tener estos componentes para garantizar que el sistema

permanezca en modo de operación continuo.

c) *Caso 1: Tensión de entrada mínima (13 V)*

$$L_c = \frac{(V_{in} - V_{out}) \cdot D \cdot T_s}{2 \cdot I_{o_{max}}} = \frac{(13 - 12) \cdot 0.923 \cdot \frac{1}{150000}}{2 \cdot 2.5} \approx 1.23 \mu H \quad (1)$$

$$C_c = \frac{(1 - D)}{16 \cdot L_c \cdot F_s^2} = \frac{(1 - 0.923)}{16 \cdot 1.23 \times 10^{-6} \cdot 150000^2} \approx 0.1739 \mu F \quad (2)$$

d) *Caso 2: Tensión de entrada máxima (20 V)*

$$L_c = \frac{(V_{in} - V_{out}) \cdot D \cdot T_s}{2 \cdot I_{o_{max}}} = \frac{(20 - 12) \cdot 0.6 \cdot \frac{1}{150000}}{2 \cdot 2.5} \approx 6.4 \mu H \quad (3)$$

$$C_c = \frac{(1 - D)}{16 \cdot L_c \cdot F_s^2} = \frac{(1 - 0.6)}{16 \cdot 6.4 \times 10^{-6} \cdot 150000^2} \approx 0.1736 \mu F \quad (4)$$

Una vez determinado el valor mínimo que pueden tener estos componentes en cada uno de los extremos correspondientes, podemos determinar su valor estimado bajo ciertas condiciones. Para los valores correspondientes a ΔI_o y ΔV_o , se tendrán en cuenta los parámetros de diseño y la teoría. Por ende, ΔI_o se trabajará como el 20% (0.5 [A]) de la corriente de salida. Se recomienda utilizar un rango de variación entre el 20%-60%. En cuanto a ΔV_o , tal como dictamina el parámetro, debe ser menor a 50 mV; por lo cual tomaremos arbitrariamente un valor de 30 mV.

A. *Caso 1: Tensión de entrada mínima (13 [V])*

$$L = \frac{V_{out} \cdot (1 - D)}{F_s \cdot \Delta I_o} = \frac{12 \cdot (1 - 0.923)}{150000 \cdot 0.5} \approx 12.32 [\mu H] \quad (5)$$

$$C = \frac{V_{out} \cdot (1 - D)}{8 \cdot L \cdot F_s^2 \cdot \Delta V_o} = \frac{12 \cdot (1 - 0.923)}{8 \cdot (12.32 \times 10^{-6}) \cdot (150000^2) \cdot (30 \times 10^{-3})} \approx 13.89 \mu F \quad (6)$$

Caso 2: Tensión de entrada máxima (20 V)

$$L = \frac{V_{out} \cdot (1 - D)}{F_s \cdot \Delta I_o} = \frac{12 \cdot (1 - 0.6)}{150000 \cdot 0.5} \approx 64 \mu H \quad (7)$$

$$C = \frac{V_{out} \cdot (1 - D)}{8 \cdot L \cdot F_s^2 \cdot \Delta V_o} = \frac{12 \cdot (1 - 0.6)}{8 \cdot 64 \cdot 10^{-6} \cdot 150000^2 \cdot 30 \cdot 10^{-3}} \approx 13.89 \mu F \quad (8)$$

Basados en los cálculos anteriores podemos determinar el valor de la capacitancia e inductancia que se debería implementar en el convertidor reductor.

- **Inductor:** Valor mínimo de 64 μH
- **Capacitor:** Valor mínimo de 13.89 μF

Los valores obtenidos representan el mínimo necesario para cumplir con las condiciones establecidas en el diseño y asegurar la operación en modo continuo. No obstante, se decidió ajustar dichos valores a una escala comercial que sea más fácil de adquirir en el mercado.

- **Inductor:** 100 μH
- **Capacitor:** 25 μF o 47 μF según accesibilidad

Selección de Diodo y MOSFETs Para seleccionar diodos y MOSFETs para un convertidor buck, es esencial considerar los siguientes factores:

- 1) **Voltaje de Bloqueo:** Ambos componentes deben tener un voltaje de bloqueo superior al máximo voltaje de entrada, con un margen de seguridad adecuado.
- 2) **Corriente de Conducción:** Los dispositivos deben soportar la corriente pico esperada en el circuito, idealmente con un margen adicional para mayor seguridad.
- 3) **Pérdidas:** Se deben elegir componentes con bajas pérdidas en conmutación y conducción para maximizar la eficiencia, especialmente en frecuencias de conmutación altas.
- 4) **Frecuencia de Conmutación:** Asegúrate de que el MOSFET y el Diodo seleccionado puedan conmutar eficientemente a la frecuencia deseada.

VII. SIMULACION DEL CONVERTIDOR REDUCTOR (BUCK)

A continuación, analizaremos el comportamiento del Buck según los valores determinados anteriormente, con el fin de corroborar la exactitud de los cálculos realizados. Para la simulación de este escenario, se implementó el software LTspice. A continuación, se ilustra la configuración utilizada:

Montaje implementado para el análisis correspondiente del buck, con un voltaje de entrada de 13 voltios.

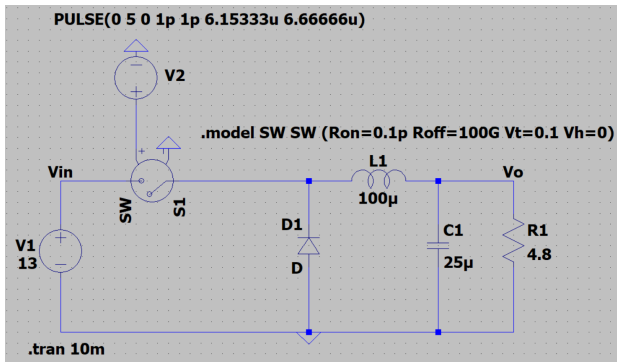


Fig. 12: Primer análisis del Buck Fuente de 13V

Montaje implementado para el analisis correspondiente del buck, con un voltaje de entrada de 20 voltios.

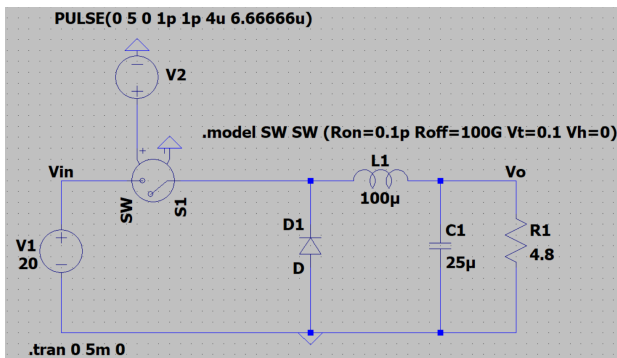


Fig. 13: Primer análisis del Buck Fuente de 20V

Configuracion correspondiente a los parametros de la fuente de pulso, tanto para una entrada de 13 voltios como para una entrada de 20 voltios.

(a) Configuración 20V

PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)

SINE(Voffset Vamp Freq Td Theta Phi Ncycles)

EXP(V1 V2 Td1 Tau1 Td2 Tau2)

SFFM(Voff Vamp Fcar MDI Fsig)

PWL#1 v1 t2 v2...)

PWL FILE: Browse

Vinitial[V]: 0

Von[V]: 5

Tdelay[s]: 0

Trise[s]: 1p

Tfall[s]: 1p

Ton[s]: 4u

Tperiod[s]: 6.6666u

Ncycles:

(b) Configuración 13V

PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)

SINE(Voffset Vamp Freq Td Theta Phi Ncycles)

EXP(V1 V2 Td1 Tau1 Td2 Tau2)

SFFM(Voff Vamp Fcar MDI Fsig)

PWL#1 v1 t2 v2...)

PWL FILE: Browse

Vinitial[V]: 0

Von[V]: 5

Tdelay[s]: 0

Trise[s]: 1p

Tfall[s]: 1p

Ton[s]: 6.15333u

Tperiod[s]: 6.6666u

Ncycles:

Fig. 14: Señal de Pulso.

Voltaje y corriente de salida del convertidor tipo Buck con una entrada de 13 y 20 voltios:

Entrada 13V: $I_o = 2.472[A]$ y $V_o = 11.869[V]$.

Entrada 20V: $I_o = 2.419[A]$ y $V_o = 11.610[V]$.

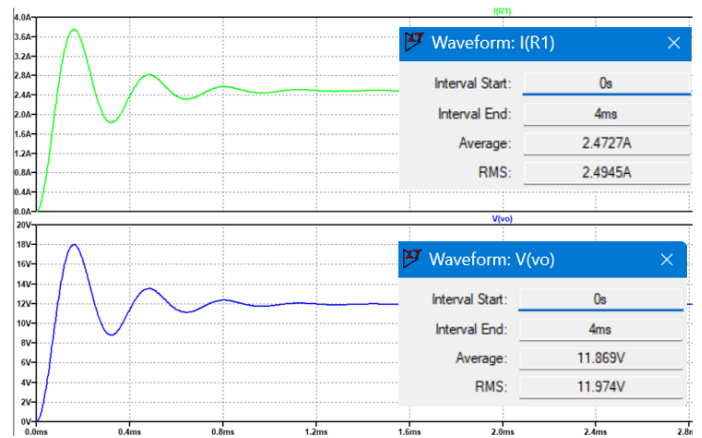


Fig. 15: Voltaje y corriente en la carga con fuente de 13V

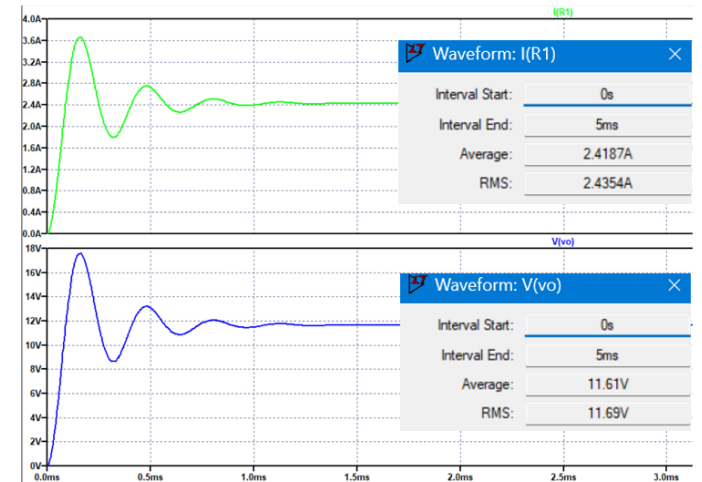


Fig. 16: Voltaje y corriente en la carga con fuente de 20V

El Buck en cuestión muestra un buen desempeño, ya que, en ambos casos planteados, la corriente de salida se aproxima a los 2.5 amperios y el voltaje está cerca de los 12 V esperados, aunque ligeramente por debajo de este valor. Estas pequeñas variaciones se deben principalmente a factores inherentes al modelo ideal utilizado, como la conmutación del interruptor. Dado que estamos trabajando con un modelo ideal, no hemos considerado las pérdidas asociadas a los dispositivos del circuito. Si se utilizara un modelo no ideal que incluyera estas pérdidas, se obtendría una representación más precisa del comportamiento del sistema. No obstante, al contrastar con el modelo teórico planteado, se puede afirmar que los cálculos realizados con base en el diseño del convertidor son correctos dentro del contexto de un análisis inicial centrado en un modelo ideal.

El rizado en la tension de salida para una entrada de 13 y 20 voltios corresponde aproximadamente:

Entrada 13V: 1.92mV.

Entrada 20V: 8.46mV.

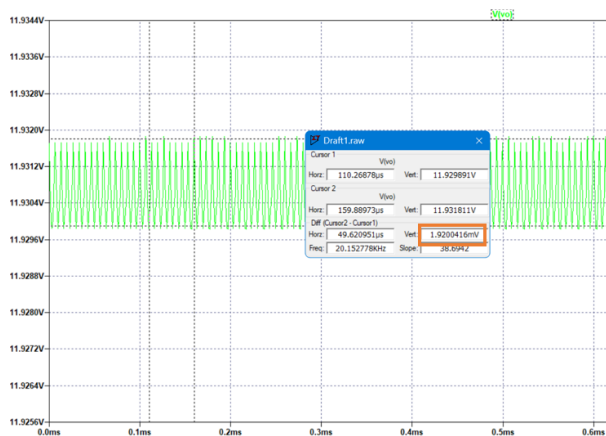


Fig. 17: Rizado en la tensión de salida 13V.

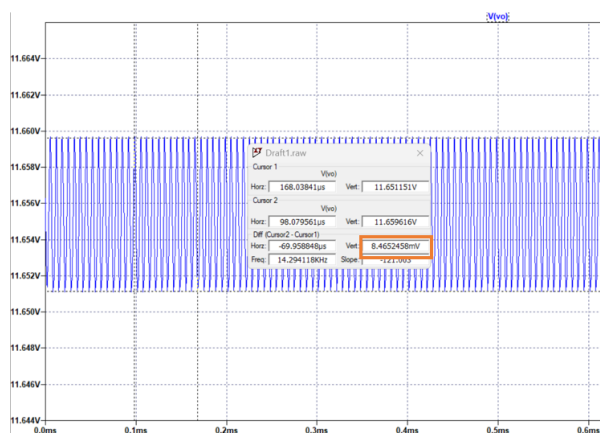


Fig. 18: Rizado en la tensión de salida 20V.

La eficiencia cuando el circuito trabaja con una fuente de 13 Voltios corresponde a:

$$\eta = \frac{P_{Out}}{P_{Int}} = \frac{29.825 \text{ W}}{30.315 \text{ W}} \times 100\% \approx 98.38\%$$

La eficiencia cuando el circuito trabaja con una fuente de 20 Voltios corresponde a:

$$\eta = \frac{P_{Out}}{P_{Int}} = \frac{28.47 \text{ W}}{29.721 \text{ W}} \times 100\% \approx 95.79\%$$

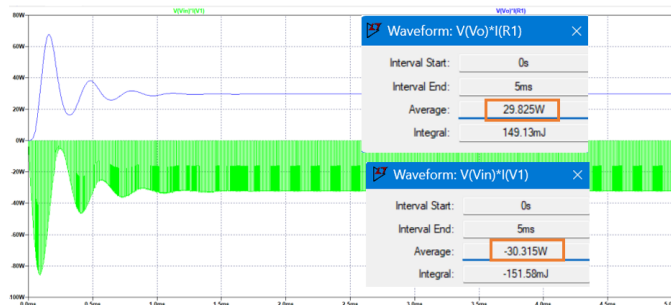


Fig. 19: Eficiencia del Buck con Fuente de Entrada de 13V



Fig. 20: Eficiencia del Buck con Fuente de Entrada de 20V

Basado en el comportamiento ilustrado en las gráficas anteriores, podemos afirmar que los parámetros planteados inicialmente se cumplen correctamente. Se esperaba poder reducir el rizado pico a pico del sistema por debajo de los 50 mV; en este caso, se obtuvo un rizado mucho mas reducido llegando aproximadamente a los 10mV, siendo mucho mejor que el esperado (30 mV). Por otro lado, la eficiencia del sistema superó las expectativas iniciales, alcanzando un 95.79%, cuando se había establecido como meta superar el 90%. Estos resultados reflejan un diseño óptimo, donde se ha logrado un equilibrio eficiente entre la estabilidad de la señal y la conversión de energía, factores esenciales para el correcto funcionamiento de un convertidor Buck.

VIII. APROXIMACION DEL MODELO A LA REALIDAD

Una vez verificado que el modelo teórico es correcto, tanto a través de los cálculos como mediante simulaciones iniciales, se procederá a simular el convertidor reductor en el software Proteus. En esta etapa, se incorporarán componentes no ideales, pérdidas y otras variables inherentes a las condiciones prácticas, cabe aclarar que a partir de este punto el capacitor simulado anteriormente con un valor de 25 μF pasara a uno de 47 μF , por su accesibilidad comercial.

Realizar esta simulación con componentes no ideales es esencial, ya que los circuitos reales presentan características que los modelos teóricos no contemplan, como las caídas de voltaje en diodos y MOSFETs. Incluir estas variaciones permite obtener una representación más precisa del comportamiento del circuito, lo que facilita ajustar el diseño para garantizar un funcionamiento confiable y eficiente en condiciones reales, antes de llevarlo a la fase de hardware físico.

IX. DISEÑO DEL CIRCUITO EN PROTEUS

El primer paso en la simulación consiste en generar una señal PWM a través de un Arduino en Proteus. Esta señal es fundamental para controlar el comportamiento de componentes electrónicos, en este caso, la conmutación del MOSFET. En el entorno de simulación de Proteus, se programa el Arduino con el código necesario para emitir la señal PWM a través de uno de sus pines digitales. El código en cuestión se encuentra en el repositorio de Github cuyo link esta al final del documento, sin embargo se puede apreciar acontinuacion.

```

1#include <PWM.h> // Se incluye la librería PWM.h, la
  ↳ cual permite controlar el ancho de pulso de
  ↳ una señal PWM y cambiar la frecuencia de los
  ↳ pines PWM. Esta librería ofrece funciones
  ↳ avanzadas para el control de PWM.
2int32_t frequency = 150000; // Se define la
  ↳ frecuencia de la señal PWM en 150 kHz. Esto
  ↳ significa que la señal generada en el pin
  ↳ tendrá esa frecuencia.
3const int pwmPin = 9; // Se define el pin del
  ↳ Arduino que se utilizará para generar la señal
  ↳ al PWM
4El Void Setup es una función que se ejecuta
  ↳ una sola vez al inicio para configurar el
  ↳ entorno.
5void setup() {
6  InitTimersSafe(); // Inicializa los temporizadores
  ↳ del Arduino de manera segura, asegurándose
  ↳ de que no entren en conflicto con otras
  ↳ funciones de hardware.
7  bool success = SetPinFrequencySafe(pwmPin,
  ↳ frequency); //Establece la frecuencia del pin
  ↳ pwmPin a 150 kHz. Esta función devuelve true
  ↳ si la configuración es exitosa. En caso de é
  ↳ xito, se ejecutan las instrucciones dentro
  ↳ del bloque if.
8  if (success) {
9    pinMode(13, OUTPUT); //Esto sirve como una
  ↳ indicación de que la configuración de PWM ha
  ↳ sido exitosa. (enciende el LED conectado al
  ↳ pin 13, que es el LED integrado en muchas
  ↳ placas Arduino)
10   digitalWrite(13, HIGH);
11  }
12}
13El Void loop es una función que se ejecuta
  ↳ continuamente una vez por ciclo de programa.
14void loop() {
15  pwmWrite(pwmPin,153); //Esta función genera una se
  ↳ ñal PWM en el pin 9 con un valor de 153 sobre
  ↳ un rango de 0 a 255. El valor de 153
  ↳ corresponde a un ciclo de trabajo del 60% (
  ↳ aproximadamente), lo que significa que el pin
  ↳ estará en alto el 60% del tiempo y en bajo
  ↳ el 40% del tiempo en cada ciclo de la señal
  ↳ PWM.
16}

```

A continuación, se puede evidenciar la señal capturada por medio del software Proteus una vez configurado el Arduino. El canal A se empleará para medir el período y voltaje de la señal (donde se puede apreciar correctamente la frecuencia y periodo empleado), mientras que el canal B permitirá observar su comportamiento general. Una vez confirmado

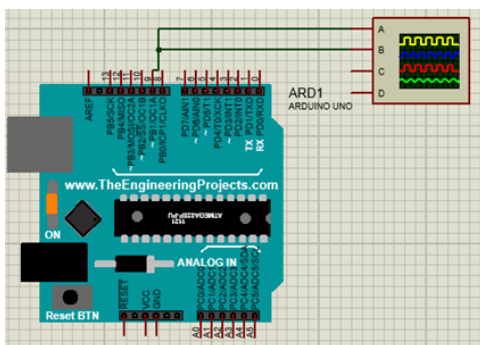


Fig. 21: Conexión Arduino.

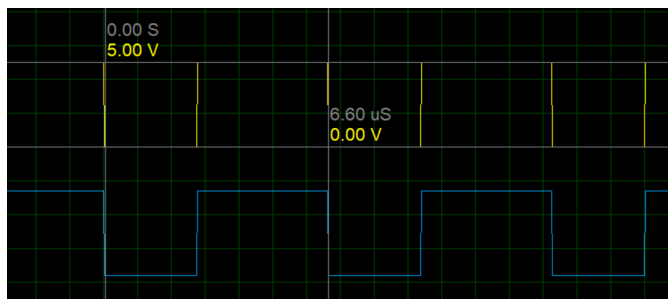


Fig. 22: Señal PWM con $F = 150$ kHz.

el correcto funcionamiento de la señal PWM generada, se procede a analizar su interacción con el driver, verificando si la configuración establecida permite que la señal se transmita de manera adecuada.

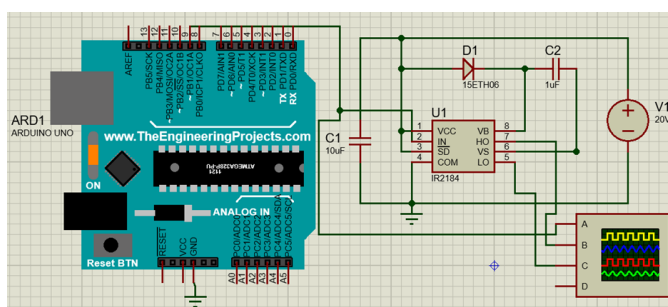


Fig. 23: Driver (controlador de la conmutación del mosfet)

La selección de los componentes en el circuito driver IR2184 se basa en asegurar un funcionamiento eficiente y estable de los MOSFETs en una configuración de medio puente. A continuación se detallan los componentes y las razones de su elección:

Capacitor de bootstrap (C2 - 1μF): Este componente es crucial para generar un voltaje superior al de la alimentación (Vcc), necesario para activar el MOSFET de lado alto. Este capacitor se carga cuando el MOSFET de lado bajo está encendido y luego se descarga para encender el MOSFET de lado alto. Un valor de 1μF es adecuado para frecuencias de conmutación cercanas a 150 KHz, ya que permite que el capacitor se cargue y descargue rápidamente en cada ciclo, mejorando así la respuesta del driver y asegurando un funcionamiento eficiente en aplicaciones de alta frecuencia. El valor del capacitor para estas frecuencias generalmente se encuentra en el rango de 0.1μF a 1μF.

Capacitor (C1 - 10μF): El capacitor C1 de 10 μF se utiliza como filtro para estabilizar la tensión de alimentación (Vcc) del driver, eliminando picos de ruido y oscilaciones no deseadas en el suministro de energía. Esto asegura una operación más estable y confiable del circuito, lo que es fundamental para el correcto funcionamiento del IR2184 en aplicaciones de control de potencia. El valor de 10 μF es comúnmente utilizado para filtrar ruidos y transitorios en circuitos de control de potencia.

Diodo D1 (15ETH06): Es un diodo Schottky de recuperación rápida que se utiliza en la configuración de bootstrap para cargar el capacitor C2 (1 μ F). El diodo permite que el capacitor se cargue rápidamente cuando la señal en el pin HO (puerta alta) es baja. Su rápida velocidad de conmutación es crítica para garantizar que el capacitor de bootstrap (C2) pueda cargarse adecuadamente en cada ciclo, y su voltaje inverso máximo de 600 V lo hace adecuado para trabajar con tensiones elevadas en la etapa de potencia del circuito.

A continuación, se analiza el comportamiento del driver ante la entrada PWM y sus dos salidas correspondientes para un MOSFET en alto y uno en bajo. A partir de la gráfica, se puede observar que las señales de salida del driver (HO y LO) muestran una conmutación adecuada y sincronizada con respecto a la señal de entrada PWM. Esto indica que el driver está operando correctamente, controlando las salidas de alto y bajo lado según la lógica de la señal de entrada. Las transiciones son nítidas, lo que sugiere una conmutación eficiente de los MOSFETs en el circuito. La simetría entre HO y LO también es un indicio de que ambos transistores están funcionando correctamente en un sistema de puente H o medio puente. Con todos los componentes configurados, procederemos a realizar la conexión del circuito para analizar su comportamiento.

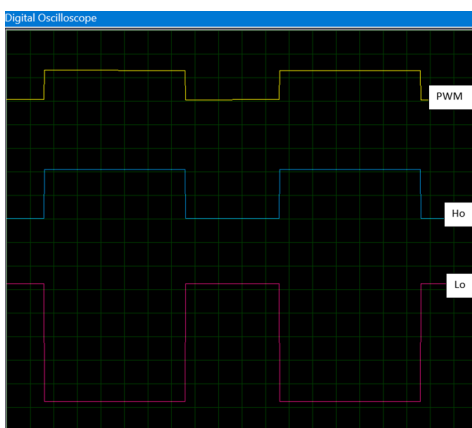


Fig. 24: gráfica driver en el osciloscopio

X. DISEÑO #1 DEL BUCK:

Inicialmente utilizaremos solamente la salida en alto del Driver, es decir, que solo controlaremos la conmutación de un MOSFET.

Cabe resaltar que, para este análisis, primero se utilizó un diodo con características similares al MBR10100 en la configuración del buck. Esto se debe a que Proteus no incluye este componente de manera predeterminada, y no fue posible descargarlo como librería. A continuación, se presentan los análisis correspondientes a los casos de voltaje de entrada mínimo y máximo.

Caso 1: Fuente de 20V

Para una fuente de 20V, se necesitó un ciclo de trabajo de 187 en PWM, lo que corresponde a un 73.333%.

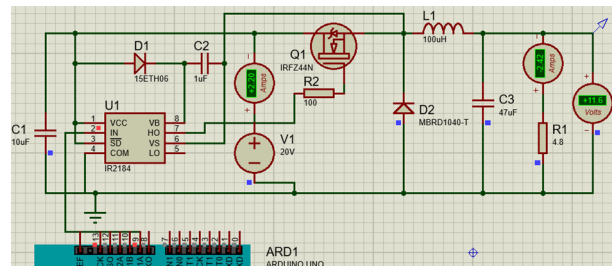


Fig. 25: Esquemático primer diseño

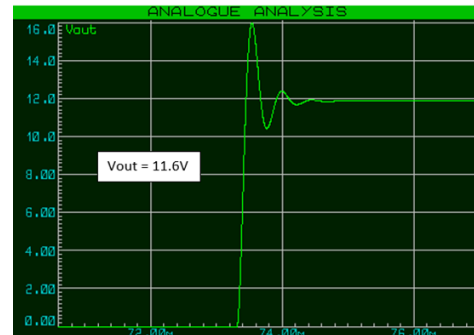


Fig. 26: Voltaje de salida

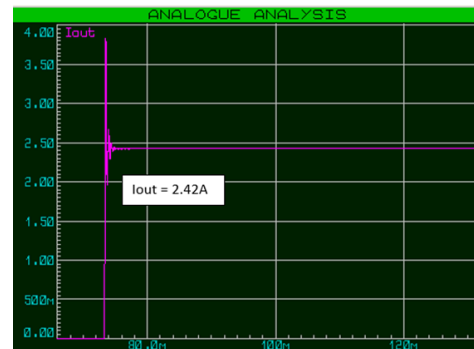


Fig. 27: Corriente de salida

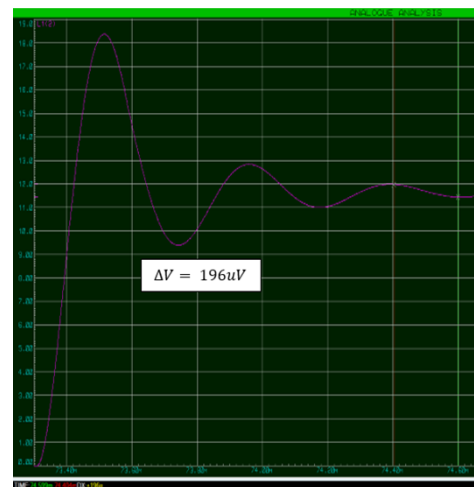


Fig. 28: Rizado pico pico

a) *Cálculo de la Eficiencia del Sistema (Fuente de 20V):*

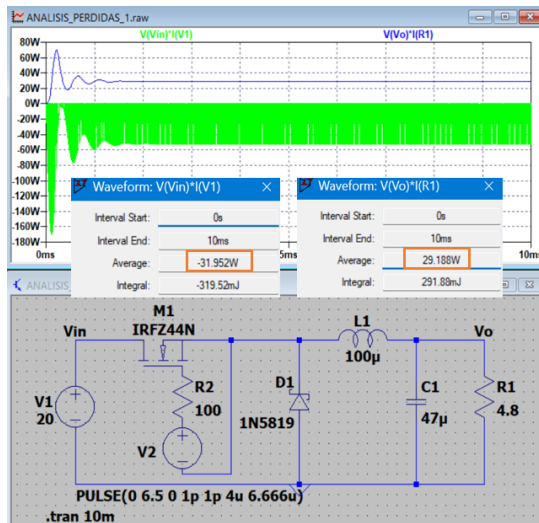


Fig. 29: Rizado pico pico

$$\eta = \frac{P_{out}}{P_{in}} = \frac{29.188W}{31.952W} \times 100\% \approx 91.35\%$$

Caso 2: Fuente 13V

Para una fuente de 13V se necesitó un ciclo de trabajo de 230 en PWM, es decir, 90.2%.

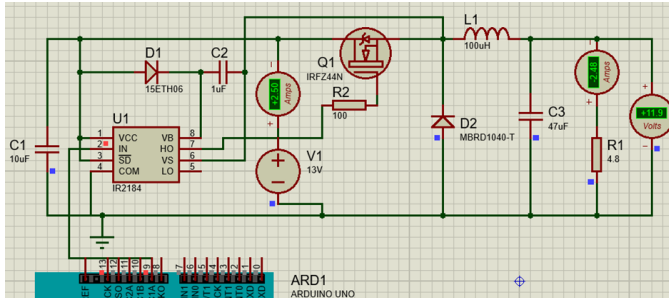


Fig. 30: esquemático con entrada a 13v

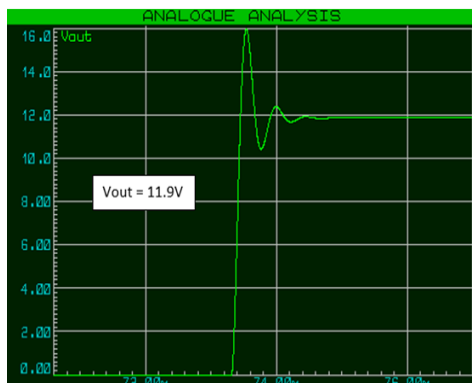


Fig. 31: voltaje de salida

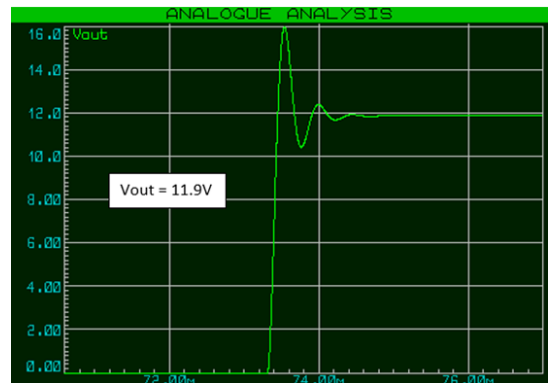


Fig. 32: corriente de salida

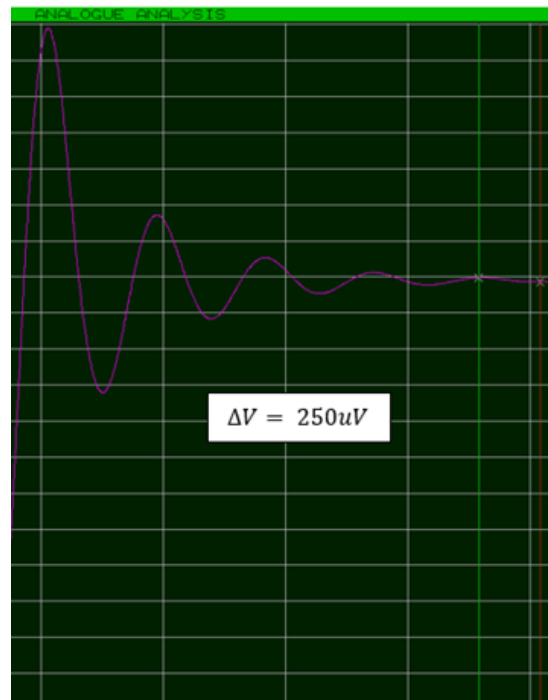


Fig. 33: voltaje pico a pico

b) *Cálculo de la Eficiencia del sistema (Fuente de 13V):*

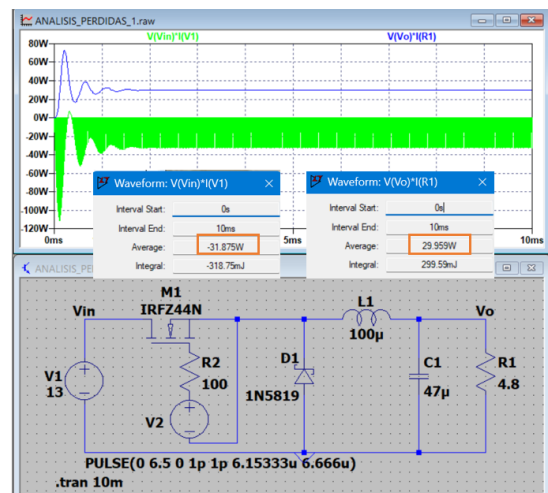


Fig. 34: Rizado pico pico

$$\eta = \frac{P_{out}}{P_{in}} = \frac{29.959W}{31.875W} \times 100\% \approx 93.98\%$$

A partir del procedimiento realizado se puede concluir que, al utilizar componentes no ideales en el circuito, fue necesario ajustar el ciclo de trabajo, ya que el voltaje y la corriente de salida no coincidían con los valores esperados. Esto se debe a las pérdidas inherentes en el sistema, que afectan su rendimiento y lo alejan de las condiciones ideales. Además, el ciclo de trabajo varía en función del voltaje de entrada, lo que requiere un sistema de control que ajuste automáticamente este ciclo a medida que cambian las condiciones de entrada.

Para determinar la resistencia del MOSFET en el convertidor buck, se utilizó un método de tanteo, iniciando con una fuente de 13V, donde el ciclo de trabajo es mayor. La razón de esta elección es que, cuando el voltaje de entrada aumenta, el ciclo de trabajo puede reducirse, lo que facilita el ajuste. Es más sencillo ajustar el ciclo de trabajo desde un valor mayor (cuando el voltaje es menor) que realizarlo desde el valor mínimo asociado al voltaje máximo de la fuente.

En resumen, el ajuste del ciclo de trabajo es esencial para compensar las pérdidas del sistema y mantener un rendimiento adecuado. Un sistema de control dinámico basado en las variaciones del voltaje de entrada permitiría optimizar el funcionamiento del convertidor buck, garantizando una salida estable frente a las fluctuaciones de la fuente.

XI. DISEÑO #2 DEL BUCK:

A partir de los resultados obtenidos anteriormente, se puede observar que en un modelo real, tanto la eficiencia como los valores de salida varían en función de las pérdidas y otros estímulos que afectan al sistema. En vista de la reducción en la eficiencia observada, se propone sustituir el diodo de conmutación por un MOSFET. Esta modificación permitiría aprovechar las capacidades del driver para mejorar el rendimiento general del sistema.

El uso de un MOSFET en lugar de un diodo en el buck converter puede incrementar significativamente la eficiencia debido a su baja resistencia interna (R_{on}), lo que minimiza las pérdidas por conducción que suelen producir los diodos. Además, los MOSFETs ofrecen una conmutación más rápida, lo cual es crucial en aplicaciones de alta frecuencia, ya que reduce las pérdidas por conmutación. Un MOSFET proporciona un mayor control sobre el sistema, al ser gestionable mediante señales de control, lo que aumenta la flexibilidad y permite optimizar aún más la eficiencia en comparación con un diodo convencional.

a) Caso 1: Fuente 20V

Para una fuente de 20V se necesitó un ciclo de trabajo de 155 en PWM, es decir, 60.78%.

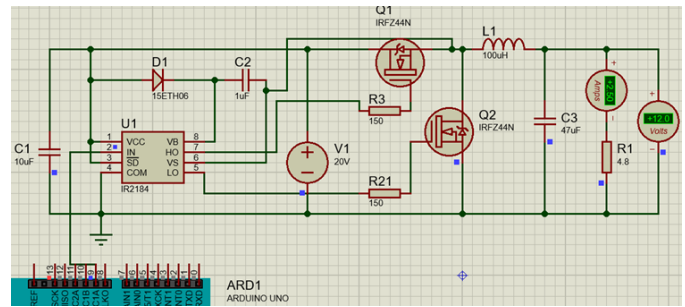


Fig. 35: diseño #2 20V

b) Caso 2: Fuente 13V

Para una fuente de 13V se necesitó un ciclo de trabajo de 230 en PWM, es decir, 90.2%.

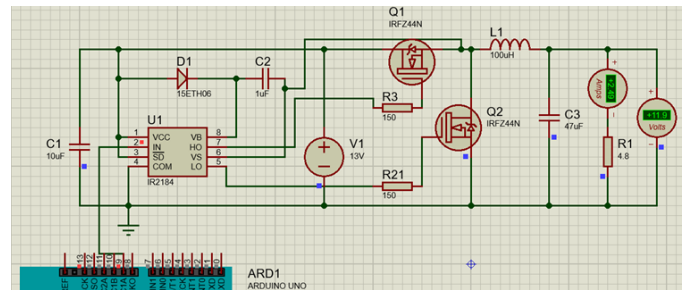


Fig. 36: diseño #2 13V

Después de implementar el segundo mosfet en el diseño del buck, no se apreciaron cambios relevantes en su rizado pico a pico como se ve en la siguiente gráfica:



Fig. 37: salida diseño #2

Sin embargo la implementación de un segundo mosfet permitió que el circuito se acercara más al valor deseado, por ende se presentó una mejora en la eficiencia del mismo, sin embargo aún presenta un problema, y es que el diseño propuesto no puede variar su ciclo de trabajo de forma independiente por lo cual se hace necesario la implementación de un código en arduino que lleve a cabo esta función, a continuación se presenta la explicación y el código implementado para cumplir con este propósito.

```
1#include <PWM.h> //Liberia correspondiente a la seña
   ↳ al PWM
2
3int32_t frequency = 150000; //Frecuencia asignada
   ↳ al Pin
```



```

4const float Vout = 12.0; //Voltaje deseado en la
  ↳ salida
5const int pwmPin = 9; // Pin asignado para la señal
  ↳ PWM
6const int VinPin1 = A4; //Pin de Lectura del Divisor
  ↳ de tension 1 (Salida)
7const int VinPin2 = A3; //Pin de Lectura del Divisor
  ↳ de tension 2 (Entrada)
8
9int PWM = 0; //Valor inicial de PWM correspondiente a
  ↳ la fuente de 20V
10int maxPWM = 254; // Valor máximo de PWM (100% ciclo
  ↳ de trabajo)
11int minPWM = 1; // Valor minimos de PWM (0% ciclo de
  ↳ trabajo)
12
13float voltaje_Out = 0; //Variable donde
  ↳ almacenaremos el valor censado (Salida)
14float voltaje_Int = 0; //Variable donde
  ↳ almacenaremos el valor censado (Entrada)
15
16void setup() {
17  InitTimersSafe(); // Inicializa los temporizadores
  ↳ del Arduino
18  bool success = SetPinFrequencySafe(pwmPin,
  ↳ frequency); //Establece la frecuencia del pin
  ↳ pwmPin a 150 kHz.
19  // Si se activa enciende el led #13 para indicar
  ↳ que fue correcto el funcionamiento
20  if (success) {
21    pinMode(13, OUTPUT);
22    digitalWrite(13, HIGH);
23  }
24}
25void loop() {
26  int Valor_Sensado_Out = analogRead(VinPin1); //
  ↳ Analiza la señal recibida por el divisor de
  ↳ tension (Salida)
27  int Valor_Sensado_Int = analogRead(VinPin2); //
  ↳ Analiza la señal recibida por el divisor de
  ↳ tension (Entrada)
28
29  // Convertir el valor analógico (0-1023) a voltaje
  ↳ real (suponiendo referencia de 5V)
30  voltaje_Out = (Valor_Sensado_Out*(5.0 /1023.0))
  ↳ *3.0; //Valor sensado y Ajustado (Salida)
31  voltaje_Int = (Valor_Sensado_Int*(5.0 /1023.0))
  ↳ *5.0; //Valor sensado y Ajustado (Entrada)
32
33//Valores fijos de PWM basados en el analisis
  ↳ realizado
34if (abs(voltaje_Int - 13) < 0.1) {
35  PWM = 230; // Si voltaje_Int es muy cercano a
  ↳ 13, el PWM será igual a 230
36} else if (abs(voltaje_Int - 20) < 0.1) {
37  PWM = 155; // Si voltaje_Int es muy cercano a
  ↳ 20, el PWM será igual a 155
38} else if (voltaje_Int != 13) {
39  PWM = (Vout / voltaje_Int) * 254.0; //Valor
  ↳ inicial del PWM
40
41  if (voltaje_Out < 12) {
42    // Si el voltaje es menor que 12V (12V en la
  ↳ salida), aumentar el ciclo de trabajo
43    PWM = constrain(PWM + 1, minPWM, maxPWM);
44  } else if (voltaje_Out >= 12) {
45    // Si el voltaje es mayor que 12V, disminuir
  ↳ el ciclo de trabajo
46    PWM = constrain(PWM - 1, minPWM, maxPWM);
47  }
48}
49pwmWrite(pwmPin, PWM); //Manda la señal PWM
  ↳ determinada por el pin seleccionado
50}

```

El código anterior se encuentra ubicado en el repositorio de Github cuyo link esta al final del documento.

A continuación se evidencia el comportamiento del circuito bajo las condiciones establecidas del código, tanto para cambios en la fuente de entrada como en la carga en la salida.

A partir del análisis realizado, se concluye que el sistema es eficiente y cumple con los requisitos establecidos para un voltaje de salida de 12V y una corriente inferior a 2.5A, siempre que la carga en la salida sea de 4.8 ohmios o superior. Además, el rizado pico a pico del voltaje de salida se mantiene en el orden de los microvoltios, cumpliendo con los parámetros especificados. También se ha logrado una eficiencia superior al 90% dentro de los rangos de voltaje establecidos.

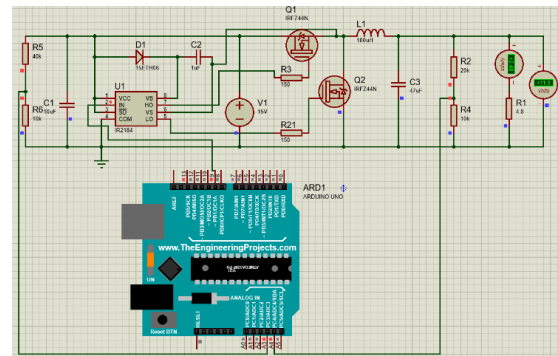


Fig. 38: 15 Voltios con una carga de 4.8 ohmios

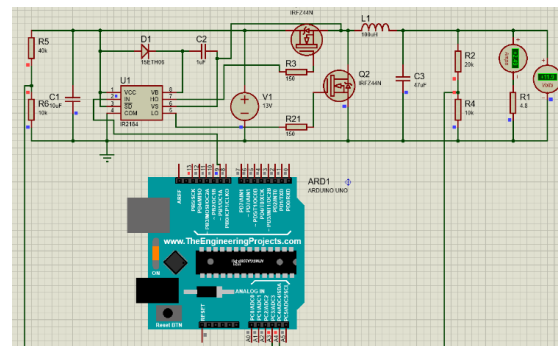


Fig. 39: 13 Voltios con una carga de 4.8 ohmios

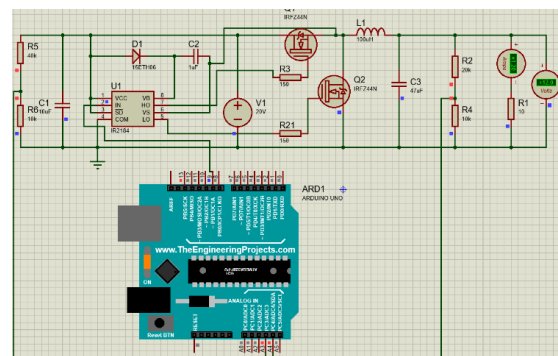


Fig. 40: 20 Voltios con una carga de 10 ohmios

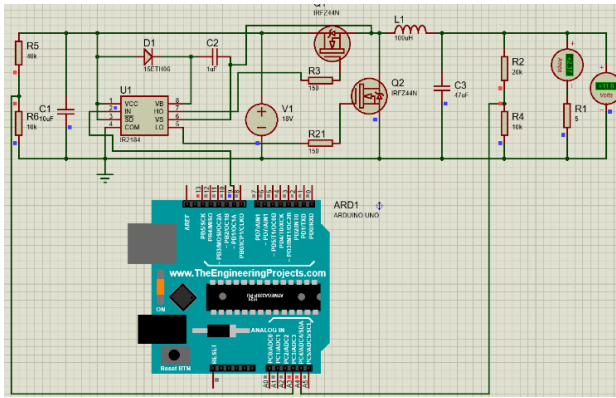


Fig. 41: 18 Voltios con una carga 5 ohmios.

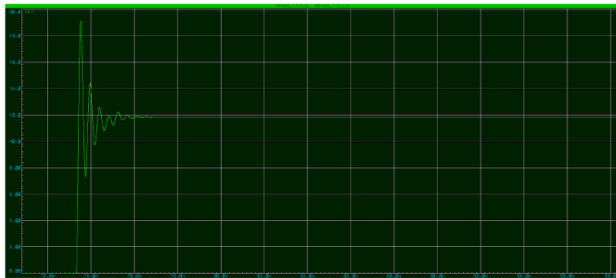


Fig. 42: Voltaje de Salida tras el control.

XII. CONCLUSIONES

- Se diseñó con éxito un convertidor Buck que cumple con los requisitos planteados inicialmente, proporcionando una salida estable de 12V y una corriente máxima de 2.5A, con un rizado de tensión de salida notablemente bajo. Además, el sistema alcanzó una eficiencia superior al 90% dentro de los rangos de voltaje estipulados y logró un ajuste automático de la tensión de salida frente a variaciones en la resistencia o la fuente, gracias al uso de un microcontrolador.
- Se comprobó que, para mantener la salida estable (V_{out}), se debe modificar el ciclo de trabajo en función del voltaje de entrada.
- La inclusión de pérdidas reales en el análisis de los componentes (MOSFET y diodos) permitió identificar las limitaciones del diseño teórico, mostrando que las simulaciones deben incluir estas variables para reflejar mejor las condiciones del mundo real y así obtener un mejor diseño.
- Se logró demostrar, por medio de la simulación y análisis, que los cálculos teóricos y la selección de componentes realizada fueron correctos, ya que se obtuvieron los resultados esperados por medio de la simulación (a excepción de la eficiencia en algunos casos específicos).
- Se logró aprovechar por completo el Driver de conmutación al modificar el sistema y reemplazar el diodo del convertidor por otro transistor, el cual asimila el comportamiento del diodo pero reduce las pérdidas de conmutación.

- Por medio de un Arduino Uno, se logró automatizar la variación del ciclo de trabajo del convertidor reductor, logrando así que este mantenga los parámetros establecidos ante cambios en la fuente de entrada o en la carga de salida (teniendo como excepciones valores de voltaje de entrada fuera del rango establecido y cargas menores a 4.8 ohmios).
- Aunque el diseño del convertidor aparenta ser viable para implementar, se requiere de una opinión más certera, además de una retroalimentación por parte del docente. Por lo cual se concluye que se debe esperar un análisis más detallado, además de una futura corrección para mejorar los parámetros faltantes en el diseño.

REFERENCES

- [1] YouTube. *Convertidor Buck - Parte 5*. Recuperado de: https://www.youtube.com/watch?v=E_urBmWlwk&list=PLwdnzlV3ogoWVgA9fHBV36L_bxWZlpa7X&index=5
- [2] EE Times. *Choosing the Optimum Switching Frequency of Your DC-DC Converter*. Recuperado de: <https://www.eetimes.com/choosing-the-optimum-switching-frequency-of-your-dc-dc-converter/>
- [3] Nomadas Electrónicos. *Convertidores DC-DC Buck*. Recuperado de: <https://nomadaselectronicos.wordpress.com/2015/04/12/convertidores-dcdc-buck/>
- [4] CVR Electrónica. *Descripción de productos*. Recuperado de: https://cvrelectronica.com/description/?product_id=687
- [5] YouTube. *Convertidor Buck - Parte 6*. Recuperado de: <https://www.youtube.com/watch?v=xUwp3TOe4m0&t=597s>
- [6] N. Mohan, T. M. Undeland, y W. P. Robbins, *Power Electronics*, 3rd ed. New York: John Wiley & Sons, 2003.

XIII. ANEXOS

A. Repositorio de Github Correspondiente al Trabajo Realizado

https://github.com/JuanEs-Pinto/BUCK_CONVERTER

En caso de no encontrar los códigos correspondientes estos también pueden ser visualizados en los siguientes enlaces

B. Código de la Señal PWM

El código utilizado para generar la señal PWM se encuentra disponible en el siguiente enlace:

<https://drive.google.com/file/d/13N02QU015oszOfsu89S4nGg4BmLxexOC/view?usp=sharing>

Este código fue programado en un Arduino UNO.

C. Código Control ciclo de trabajo BUCK

<https://drive.google.com/file/d/1xO1bYBKTVXevVw-6LrnfXSRVrsH8PUS/view?usp=sharing>

Este código fue programado en un Arduino UNO y se utiliza para controlar el ciclo de trabajo del convertidor Buck.