



DISEÑO E IMPLEMENTACIÓN DEL CONTROL INTELIGENTE A UN PROTOTIPO D-STATCOM.

Mauricio José Triana Figueroa

Universidad de Pamplona
Ingenierías y arquitectura, Norte de Santander
Cúcuta, Colombia

2023

DISEÑO E IMPLEMENTACIÓN DE UN D-STATCOM UTILIZANDO TÉCNICAS INTELIGENTE.

Mauricio José Triana Figueroa

Trabajo de grado presentado como requisito para optar al título de:
Magíster en Controles Industriales

Director: MSc. Jorge Luís Díaz R
Codirector: PhD. Aldo Pardo García

Universidad de Pamplona
Ingenierías y arquitectura, Norte de Santander
Cúcuta, Colombia
2023

A Dios por darme la vida y las fortalezas para vencer todas las adversidades y salir adelante en los momentos difíciles. A mis padres y hermanos, por el inmenso apoyo que me han brindado durante todos mis estudios y durante toda mi vida.

Nunca mires tu origen como un impedimento para ser alguien en la vida, míralo como una motivación para nunca rendirte, ya que muchas veces de atrás venimos los que adelante vamos a estar.

Mauricio José Triana Figueroa.

Agradecimientos

Agradezco principalmente a Dios por darme frutos producto de mi esfuerzo, por permitirme llegar a este momento importante en mi vida, por esta oportunidad de mejorar a nivel profesional, por esta hermosa oportunidad en la docencia permitiéndome transferir mis conocimientos a los futuros profesionales.

A mi familia, a mis padres y hermanos que siempre están cuando los necesito, que gracias a Dios todavía disfruto de su compañía, que por más inconvenientes que se nos presente siempre encontramos la forma de solucionarlo y salir adelante.

A mi director de tesis M.Sc. Jorge Luis Días que me coordino mi proyecto de grado, que, gracias a sus enseñanzas, consejos y sobre todo ánimos no me dejaron rendirme, a mis docentes de maestría por sus conocimientos y asesorías compartidas, pero sobre todo a la Universidad de Pamplona, mi segundo hogar, mi alma máter, que gracias a ella he tenido la oportunidad de mejorar mi calidad de vida, no me alcanza la vida para agradecerle.

A mis estudiantes, que son mi motivación para estar aprendiendo constantemente, permitiéndome ser cada día un poco más digno de ser parte de su formación profesional.

Contenido

LISTA DE FIGURAS	6
RESUMEN	8
PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	9
INTRODUCCIÓN.....	11
CAPÍTULO 1. TIPOS DE CARGA ELÉCTRICA.	12
1.1 Factor de Potencia.	13
1.2 Corrección del Factor de Potencia.	15
CAPÍTULO 2. SISTEMAS FLEXIBLES DE TRANSMISIÓN DE CA.	16
2.1 Compensador de VAR estático.....	17
2.2 Compensador avanzado de VAR estático.	18
2.3 Capacitor serie conmutado por tiristor (TSSC).	19
2.4 Capacitor serie controlado por tiristor (TCSC).	20
2.5 Compensador SSSVC avanzado.....	21
CAPÍTULO 3. CARACTERÍSTICAS BÁSICAS DE UN D-STATCOM.	22
3.1 Caracterización del D-STATCOM.	22
3.1.1 Inversor de voltaje SPWM.....	23
3.1.2 Convertidor de voltaje DC-DC BOOST.	24
CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN DEL D-STATCOM.....	25
4.1 Puente H.....	26
4.2 Detector de secuencia de fase.....	26
4.3 Convertidor DC-DC de tipo Boost.	27
4.4 Entrada para el sensor PZEM-004T.....	28
4.5 Control del D-STATCOM.....	29
CAPÍTULO 5. ESTRATEGIAS DE CONTROL, PRUEBAS Y RESULTADOS.....	29
5.1 Inversor SPWM	30
5.2 Convertidor DC-DC de tipo elevador	33
5.3 Sincronización del inversor con la red eléctrica.....	35
5.4 Control y sincronización del voltaje RMS de la etapa inversora con la red AC.....	38
5.5 Efecto y corrección de la carga sobre la etapa inversora	49
5.6 Código y resultados del entrenamiento de la Red Neuronal para la sincronización.	51
5.7 Conversión de los datos obtenidos en la Red Neuronal al lenguaje Arduino.....	56
5.8 Sincronización del prototipo D-STATCOM con la red eléctrica.....	58

5.8.1 Sincronización del prototipo D-STATCOM con la red eléctrica a plena carga igualando voltaje RMS.	60
5.8.2 Sincronización del prototipo D-STATCOM con la red eléctrica a plena carga aplicando voltaje RMS de compensación.	61
5.8.3 Sincronización del prototipo D-STATCOM con la red eléctrica a plena carga aplicando voltaje RMS menor al de la red eléctrica.	62
5.9 Control del FP de potencia en la etapa de sincronización.	63
5.9.1 Control P&O del FP de potencia en la etapa de sincronización.	65
Conclusiones	68
Referencias.	69
Anexos.	73

LISTA DE FIGURAS

Figura. 1 Sistema de Generación, transmisión y distribución de la energía eléctrica.	12
Figura. 2 Representación de un dispositivo eléctrico como carga eléctrica.	13
Figura. 3 Desfase de la corriente respecto al voltaje.	14
Figura. 4 Relación entre la potencia reactiva con el ángulo del factor de potencia.	14
Figura. 5 Compensación de Factor de potencia usando un capacitor en paralelo.	15
Figura. 6 Compensador de VAR estático usando TSR y TSC.	17
Figura. 7 Compensador avanzado de VAR estático.	19
Figura. 8 Capacitor en serie conmutado por tiristores	20
Figura. 9 Capacitor en serie controlado por tiristor.	20
Figura. 10 Compensador SSSVC avanzado.	21
Figura. 11 Circuito compensador por derivación.	23
Figura. 12 Puente H para la etapa inversora tipo SPWM	24
Figura. 13 Convertidor de voltaje DC-DC Boost.	25
Figura. 14 Prototipo D-STATCOM.	25
Figura. 15 Puente H	26
Figura. 16 Circuito detector de secuencia de fase.	27
Figura. 17 Convertidor DC-DC de tipo Boost.	28
Figura. 18 Entrada del Sensor PZEM-004T.	28
Figura. 19 Entrada de los Arduino nano.	29
Figura. 20 Entrada del Arduino Control-SPWM.	30
Figura. 21 Diseño esquemático del circuito generador SPWM	31
Figura. 22 Señales del Arduino en el puente H	32
Figura. 23 Señales generadas en el transformador.	33
Figura. 24 Circuito esquemático de control TL494	33

Figura. 25	Circuito esquemático del convertidor DC-DC con señales PWM.....	34
Figura. 26	Pulsos PWM del convertidor DC-DC Boost.	35
Figura. 27	Lista de pulsos PWM.....	36
Figura. 28	Generación de la onda senoidal de la etapa inversora	36
Figura. 29	Activación de la función verificar	36
Figura. 30	Generación de la onda senoidal de la etapa inversora	37
Figura. 31	Representación del control proporcional	38
Figura. 32	Sincronización de la etapa inversora con la red eléctrica.	38
Figura. 33	Monitoreo del voltaje del inversor y de la red eléctrica.	39
Figura. 34	Toma de datos para calcular el voltaje RMS.	40
Figura. 35	Cálculo de los voltajes RMS en Arduino.	40
Figura. 36	Módulo Bluetooth y aplicación móvil.	41
Figura. 37	Módulo Bluetooth y aplicación móvil.	41
Figura. 38	Módulo Bluetooth y aplicación móvil.	42
Figura. 39	Funciones de membresía de entrada.	43
Figura. 40	Control del ciclo de trabajo del TL-494.	44
Figura. 41	Funciones de membresía de salida.	44
Figura. 42	Reglas de control.	45
Figura. 43	Conexión del control de voltaje difuso.....	46
Figura. 44	Librería eFLL (Embedded Fuzzy Logic Library).....	46
Figura. 45	Funciones de membresía de entrada.....	47
Figura. 46	Funciones de membresía de salida	47
Figura. 47	Reglas de control	48
Figura. 48	Sincronización con el DAC MCP4725.....	49
Figura. 49	Simulador de cargas.	50
Figura. 50	Carga de datos en la Red Neuronal.	51
Figura. 51	Entorno de Trabajo	52
Figura. 52	Librerías en Python para el análisis de datos.....	52
Figura. 53	Arreglo de datos para el procesamiento	53
Figura. 54	Construcción de la Red Neuronal	54
Figura. 55	Magnitud de pérdidas en el entrenamiento.....	55
Figura. 56	Valores calculados de los pesos	56
Figura. 57	Implementación de la red neuronal en Arduino lgt8f328p.....	57
Figura. 58	Control difuso en el voltaje del inversor	57
Figura. 59	Control difuso en el voltaje del inversor	58
Figura. 60	Configuración de acople.....	59
Figura. 61	Relé de estado sólido para la sincronización en AC.....	60
Figura. 62	Sincronización igualando voltaje RMS	61
Figura. 63	Sincronización con voltaje RMS de compensación	62
Figura. 64	Sincronización con voltaje RMS más bajo.....	63
Figura. 65	Fuente DC extraída de la red eléctrica para el D-STATCOM.....	64
Figura. 66	Relé de estado sólido para el control de la fuente DC.	65
Figura. 67	Algoritmo P&O	66
Figura. 68	Control del Factor de Potencia mediante el control P&O	67

RESUMEN

Este proyecto tiene como objetivo la realización del control necesario para un dispositivo D-STATCOM. Para cumplir con tal fin, se empieza con la caracterización de los componentes que lleva el dispositivo y su funcionamiento. Se realiza una revisión del estado del arte de cómo debe funcionar tal dispositivo, y cuales deben ser el funcionamiento de cada etapa. Posteriormente se realizó un prototipo que permita tanto la conexión en serie como en paralelo del dispositivo. Por último, se realizará el control del prototipo construido utilizando dos Arduino nano LGT8F328P con diferentes estrategias de control.

PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

Actualmente y debido a la creciente demanda de energía eléctrica, producto del aumento de la población a nivel mundial, en las cuales tenemos viviendas unifamiliares y multifamiliares, pequeñas y medianas industrias, centros comerciales, etc. Esto produce un aumento significativo en cargas reactivas provocando un bajo factor de potencia y problemas en la calidad de energía. El aumento de la carga eléctrica no útil como la reactiva disminuyen la capacidad de distribución tanto de los transformadores como de los conductores eléctricos.

¿Qué efectos nocivos genera el aumento de la potencia reactiva?

El problema del aumento de la potencia reactiva es que genera un desfase entre el voltaje de alimentación con respecto a la corriente eléctrica que circula por la carga, este desfase se conoce como factor de potencia y va de cero a uno. Actualmente y con gran velocidad las personas aumentan el consumo de energía eléctrica, esto debido al aumento de productos tecnológicos en el hogar con alta carga inductiva como lo son ventiladores, neveras, lavadoras, aires acondicionados entre otros, no obstante, otra razón es debido a que muchas personas están modificando sus viviendas convirtiéndolas en edificios de apartamento o negocios, incluyendo otros artefactos de gran consumo como lo serían motobombas y motores eléctricos. Este aumento eléctrico de por sí ya disminuye la capacidad de los transformadores, conductores eléctricos y protecciones que en su momento no se diseñaron teniendo en cuenta estas tendencias futuras, a este problema de la velocidad con la que aumenta la carga en los sistemas eléctricos, debemos sumarle la problemática de la potencia reactiva, que aunque no sea una potencia útil, es una potencia que provoca un aumento en la corriente de carga tanto en los transformadores como en los conductores eléctricos disminuyendo la capacidad de estos aún más provocando fallas de capacidad, protecciones y calentamiento.

¿Porque construir un prototipo D-STATCOM?

El problema de la disminución de la capacidad de los transformadores y conductores eléctricos de distribución debido al aumento de la potencia reactiva se puede amortiguar con algunos compensadores existentes, uno de estos son los bancos capacitivos que hacen el efecto contrario a las cargas reactivas permitiéndonos recuperar capacidad de distribución perdida por el efecto de la potencia reactiva, no obstante, la limitación que tiene los bancos capacitivos es que solo maneja un rango de compensación, lo que significa que para poder utilizarlo hay que esperar hasta que se encuentre en un punto específico limitando su uso. El D-STATCOM es un dispositivo cuya función es comportarse como un capacitor variable permitiendo recuperar la capacidad máxima de suministro eléctrico. Estos dispositivos, aunque ya existen en el mercado, y se tiene conocimiento básico de su funcionamiento es una tecnología reservada por sus fabricantes, por lo tanto, es de gran ayuda contar con un prototipo que nos permita mejorar los conocimientos en el campo de la calidad de energía como en las estrategias de control.

¿Vale la pena construir un prototipo para mejorar el conocimiento en las estrategias de control?

Este prototipo nos permite tener no solo un gran laboratorio de calidad de energía ya que se tiene pensado realizar de tal forma que su funcionamiento sea tanto en serie como derivativo, sino que también nos permite investigar en un gran número de estrategias de control, como lo puede ser la Lógica difusa, redes neuronales, etc.

Actualmente y fundamentado en una creciente demanda de energía eléctrica ha sido necesario mejorar los sistemas de generación, transmisión y distribución. Este aumento de la demanda genera problemas de calidad de energía como bajo factor de potencia, armónicos, caída de tensión y aumento en la corriente de distribución.

El aumento de la carga eléctrica no útil, como la potencia reactiva, disminuyen la capacidad de distribución tanto de los transformadores como de los conductores. Una de las alternativas para disminuir el efecto nocivo del aumento de las cargas reactivas es el uso de compensadores electrónicos conocidos como FACTS.

Estos dispositivos acompañados con técnicas de control permiten disminuir los efectos negativos del aumento de la carga reactiva recuperando capacidad de funcionamiento de equipos importantes como transformadores, conductores eléctricos entre otros.

INTRODUCCIÓN

Debido a la creciente demanda de energía eléctrica, producto del aumento de la población mundial, en las cuales tenemos viviendas unifamiliares y multifamiliares, pequeñas y medianas industrias, centros comerciales, etc. Esto ha producido un aumento significativo en cargas reactivas provocando un bajo factor de potencia y problemas en la calidad de energía, por esto ha sido necesario mejorar los sistemas de generación, transmisión y distribución. Actualmente el avance de la electrónica de potencia nos ha brindado una alternativa en el mejoramiento de cargas de compensación, dándonos un amplio margen de controlabilidad disminuyendo muchos de los problemas que nos genera el aumento de la demanda[1].

El aumento de la carga eléctrica no útil como la reactiva disminuyen la capacidad de distribución tanto de los transformadores como de los conductores. Una de las alternativas que existen para disminuir el efecto nocivo del aumento de las cargas reactivas son los compensadores electrónicos conocidos como FACTS. Estos dispositivos acompañados con técnicas de control permiten disminuir los efectos negativos del aumento de la carga reactiva recuperando capacidad de funcionamiento de equipos importantes como transformadores, conductores eléctricos entre otros[2]–[4].

El compensador estático de distribución (D-STATCOM) es un dispositivo que pertenece a los sistemas flexibles de transmisión de AC (FACTS), y en este documento proponemos un prototipo que nos permita aumentar nuestros conocimientos sobre esta tecnología, además de proporcionar una herramienta de fácil uso a la hora de crear diferentes estrategias de control[4], [5].

CAPÍTULO 1. TIPOS DE CARGA ELÉCTRICA.

Los dispositivos eléctricos se pueden representar mediante 3 componentes, en los que tenemos la fuente generadora, los conductores eléctricos (líneas de transmisión) y la carga final (**Figura 1**), esta última son aquellos dispositivos eléctricos que usamos en el día a día ya sea en nuestros hogares o sitios de trabajo.

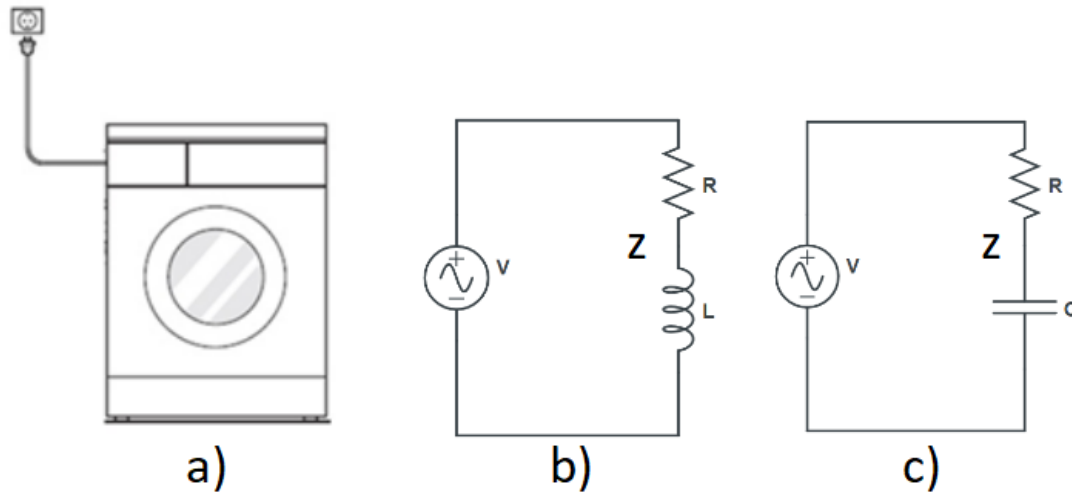
Figura. 1 Sistema de Generación, transmisión y distribución de la energía eléctrica.



(ingelcom, 2022)

Sin importar la cantidad de componentes o que tan complejos sean los dispositivos eléctricos, estos se pueden representar en 3 tipos de carga (cargas resistivas, cargas capacitivas, cargas inductivas) (**Figura 2**), las cargas inductivas son aquellas conformadas por bobinas eléctricas y las cargas capacitivas son aquellas conformadas por capacitores, si la carga internamente tiene más componentes inductivos el circuito general se puede representar como se observa en la (**Figura 2.b**), de lo contrario, si la carga internamente tiene más componentes capacitivos el circuito general se puede representar como se observa en la (**Figura 2.c**).

Figura. 2 Representación de un dispositivo eléctrico como carga eléctrica.



En los sistemas de distribución se usa el mismo principio de análisis de circuitos, donde podemos obtener el circuito equivalente $Z = R + jX$ dependiendo de la cantidad de dispositivos eléctricos, ya sea una vivienda, pequeño comercio, o una gran industria.

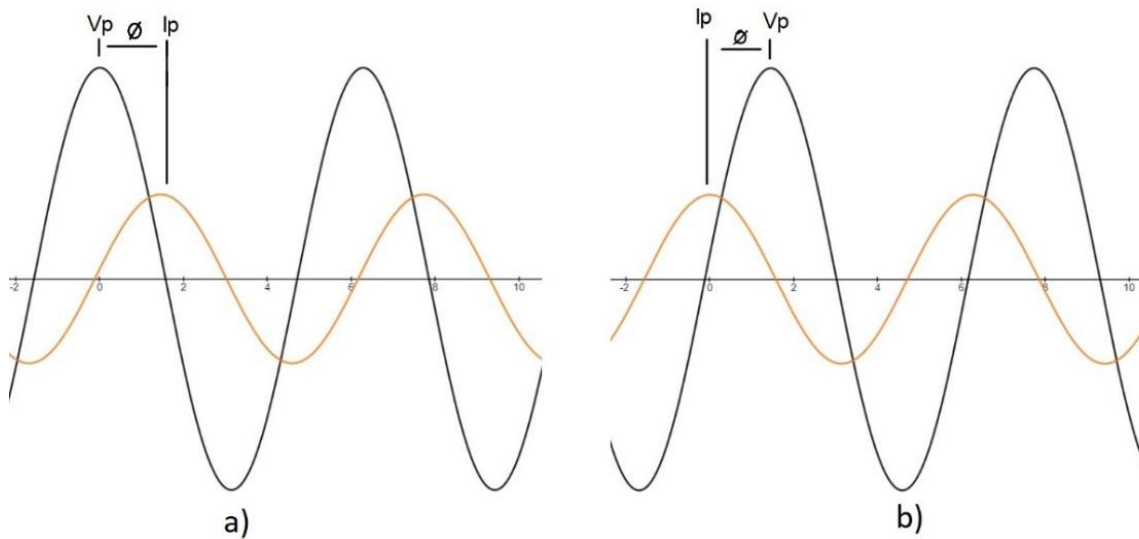
$$Z = R + jX \quad (1)$$

R = Representa la cantidad parcial o total de la carga resistiva conectada a la fuente.

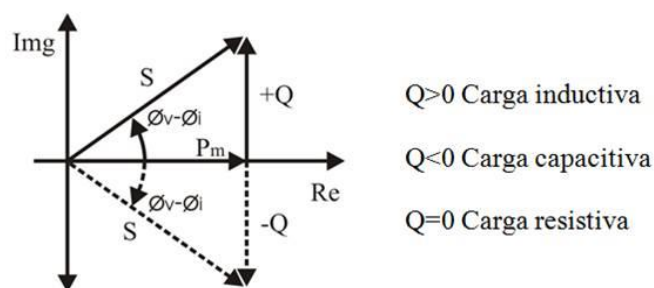
jX = Representa la cantidad parcial o total de la carga reactiva conectada a la fuente.

1.1 Factor de Potencia.

Las cargas reactivas tienen un efecto sobre la corriente que circula a través del dispositivo, este efecto es desfasar la corriente con el voltaje de alimentación haciendo que la corriente se atrase o adelante (**Figura 3**), si la carga eléctrica tiene una mayor cantidad de dispositivos inductivos que dispositivos capacitivos, el efecto que produce sobre la corriente es atrasarlo con respecto al voltaje de alimentación, este efecto lo podemos apreciar en la (**Figura 3.a**), en cambio si la reactancia capacitiva predomina sobre la inductiva, el efecto que produce es adelantar la corriente respecto al voltaje de alimentación (**Figura 3.b**).

Figura. 3 Desfase de la corriente respecto al voltaje.

Este efecto de adelantar o atrasar la corriente con respecto al voltaje es provocada por la cantidad de potencia reactiva que consume la carga, entre más grande sea la carga reactiva, más grande es el desfase, en la **(Figura 4)** podemos ver la relación entre la potencia reactiva Q y el ángulo de desfase.

Figura. 4 Relación entre la potencia reactiva con el ángulo del factor de potencia.

Esta potencia reactiva no es buena ya que no produce trabajo útil y le exige más corriente a la fuente, para tratar de mantener la potencia activa requerida, lo ideal es mantener un ángulo de desfase lo más cercano a cero ya que el cálculo de la potencia útil se calcula como:

$$P = |V_{rms}| * |I_{rms}| * \cos(\theta_v - \theta_i) \quad (2)$$

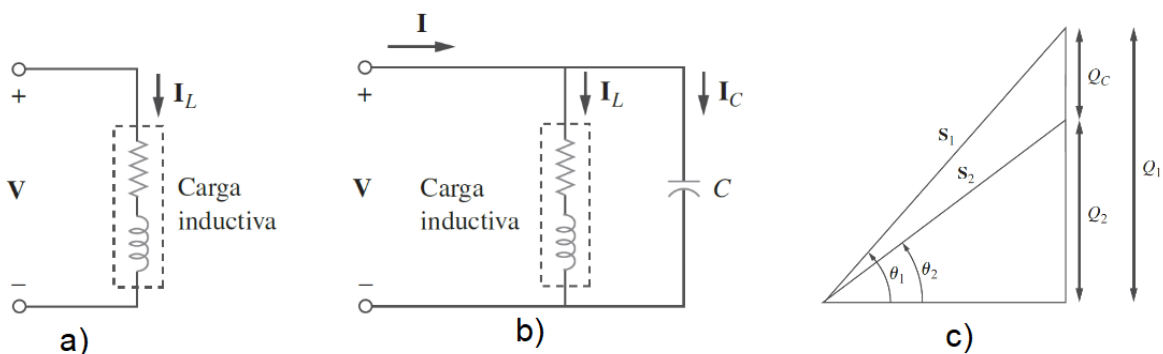
El $\cos(\theta_v - \theta_i)$ es lo que conocemos como Factor de Potencia (**FP**), y su valor varía entre 0 y 1, cuando el ángulo de desfase entre $\theta_v - \theta_i$ es igual a cero el $\cos(\theta_v - \theta_i)$ es igual a 1 indicando que la potencia suministrada a la carga es solamente activa, esto ocurre cuando la carga es solamente resistiva (**R**) o cuando la cantidad total de inductancia en la carga es igual a la cantidad total de capacitancia.

1.2 Corrección del Factor de Potencia.

Dependiendo de los dispositivos eléctricos, aumentar la carga reactiva aumenta el desfase de la corriente con referencia al voltaje de alimentación, la corriente se atrasa o se adelanta dependiendo de cual reactancia predomine, se atrasa si es inductiva (**jXL**) y se adelanta si es capacitiva (**jXC**), en la mayoría de los casos, la corriente se atrasa predominando la reactancia inductiva, esto como efecto a las cargas domésticas e industriales como (lavadoras, aire acondicionado, refrigeradores, motores de inducción etc.) estos operan con un factor de potencia bajo y atrasado.

Una de las soluciones más usadas para corregir el bajo factor de potencia producido por las reactancias inductivas, es usar su contraparte capacitiva donde ayude a corregir el factor de potencia y no permita que el $\cos(\theta_v - \theta_i)$ se aleje mucho de 1, en la (Figura 5) se ve un ejemplo de la corrección usando un capacitor, no obstante y aunque no sea muy común, si la carga es mayormente capacitiva produciendo un adelanto de la corriente en vez de poner un capacitor en paralelo a la carga, se puede poner un inductor obteniendo el efecto contrario, la idea es compensar una reactancia con su contraparte tratando de mantener el factor de potencia lo más cercano a uno[6].

Figura. 5 Compensación de Factor de potencia usando un capacitor en paralelo.



(Fundamentos de Circuitos Eléctricos Sadiku, 2013)

Volviendo al ejemplo anterior donde debemos compensar la potencia inductiva con un capacitor que nos permite restar la cantidad deseada mediante $Q_c = Q_1 - Q_2$, para el caso del capacitor lo podemos calcular aplicando la siguiente formula:

$$C = \frac{Q_c}{\omega * |V_{rms}|^2} = \frac{P(\tan(\theta_1) - \tan(\theta_2))}{\omega * |V_{rms}|^2} \quad (3)$$

Este método simple es eficiente para compensar cargas cuya variación del factor de potencia es muy poca, puede funcionar bien en pequeñas industrias, pero para sistemas de distribución no es muy eficiente hoy en día, ya que las cargas conectadas a la red son variables, sin embargo aunque se usan actualmente, solo se puede corregir una cantidad fija del factor de potencia en ciertos rangos de tiempo, esto se debe a que ciertas horas del día entran más dispositivos eléctricos aumentando la carga inductiva provocando un bajo factor de potencia, es ahí cuando entra el compensador capacitivo, pero cuando las cargas empiezan a desconectarse de la red eléctrica, esta compensación puede ser contraproducente ya que si no se desconecta en el instante ideal, puede provocar un adelanto de la corriente respecto al voltaje produciendo el mismo efecto negativo que una carga inductiva con la diferencia que la corriente estaría en adelanto.

CAPÍTULO 2. SISTEMAS FLEXIBLES DE TRANSMISIÓN DE CA.

Actualmente el gran crecimiento de la demanda eléctrica producida por viviendas, comercios, industrias, etc., lleva al mejoramiento de las líneas de transmisión, esto debido a las limitaciones en el suministro de potencia en CA[8], [9].

Las cargas industriales afectan negativamente la calidad de la energía entregada a los consumidores ya que uno de los efectos más comunes es el deterioro de la calidad de la energía causando fluctuación de tensión generando el parpadeo de iluminaciones domésticas, a su vez influyendo negativamente en los aparatos del sistema de distribución y transmisión[10]–[12].

Otro problema actual es el de la calidad de la energía, incluye principalmente el desequilibrio de las cargas, los armónicos, caída de tensión, el aumento de la corriente, etc. Estos problemas de calidad de la energía pueden provocar operaciones anormales en las instalaciones o incluso desconectar los dispositivos de protección. Por lo tanto, el mantenimiento y la mejora de la calidad de la eléctrica se han convertido en un escenario importante en la actualidad[12], [13].

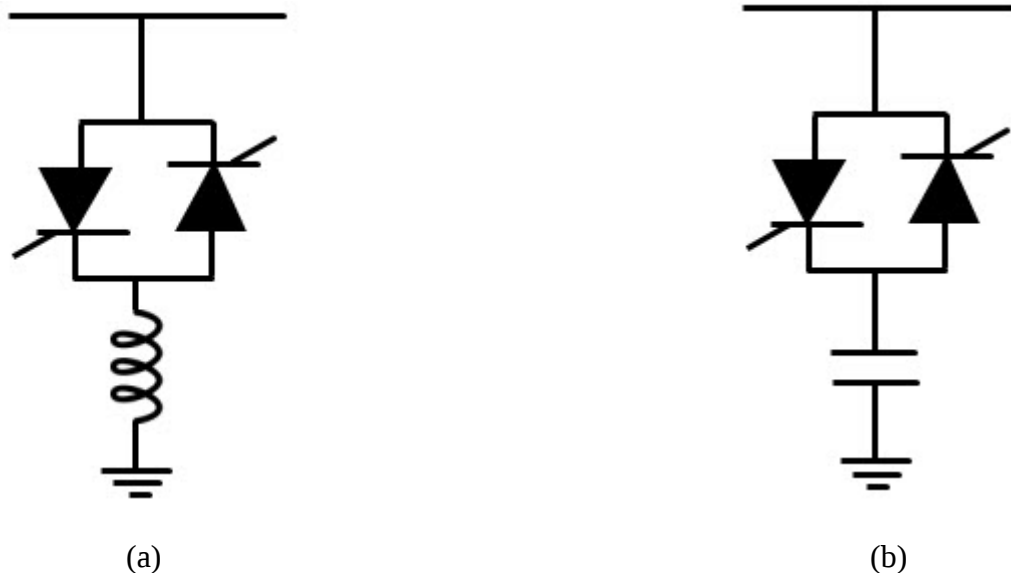
En este capítulo hablaremos de los sistemas flexibles de transmisión de CA (FACTS, flexible ac transmission systems) una nueva tecnología emergente, y su papel principal es aumentar la capacidad de control y de transferencia de potencia. Esta tecnología usa electrónica de alta capacidad y con una estrategia de control se puede controlar los flujos de potencia permitiendo usar con mayor eficiencia las líneas de transmisión[9], [14], [15].

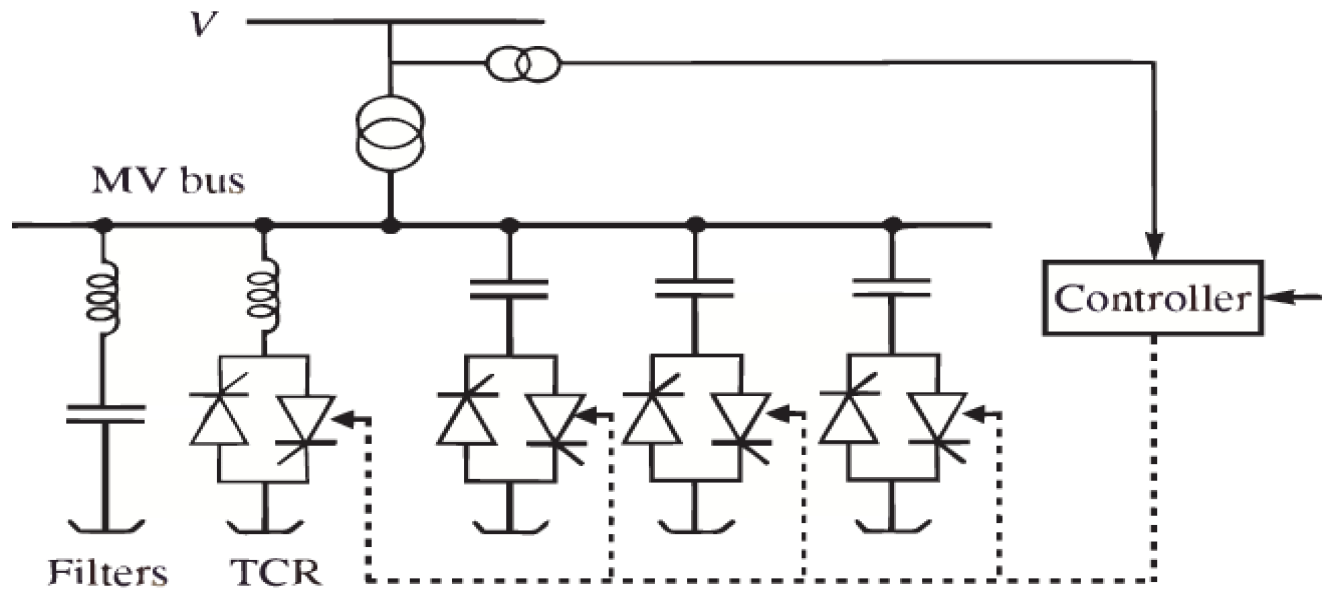
2.1 Compensador de VAR estático.

El compensador de VAR estático (SVC, static VAR compensator, **Figura 6.c**) consiste en uno o varios TCR (Thyristor-switched reactor, **Figura 6.a**) en paralelo con uno o varios TSC (Thyristor-switched capacitor, **Figura 6.b**), este arreglo permite la compensación tanto capacitiva como inductiva, no obstante y aunque no sea muy común, la compensación reactiva nos permite compensar la corriente en adelanto del sistema, pero también nos permite limitar el rango de potencia capacitiva teniendo un mayor rango de compensación, la cantidad de combinaciones o rangos que podemos obtener, depende de la cantidad de elemento en paralelos conectados, esto lo podemos calcular mediante $C = 2^n$, siendo n la cantidad de elementos y C el número de combinaciones posible[16], [17].

Con la estrategia de control adecuada podemos ir activando o desactivando cada carga en el instante deseado, el objetivo es mantener el voltaje y el factor de potencia en un valor fijo, para eso se debe tener una señal en tiempo real usando sensores de potencia determinando los instantes exactos de disparo de los reactores[18].

Figura. 6 Compensador de VAR estático usando TSR y TSC.





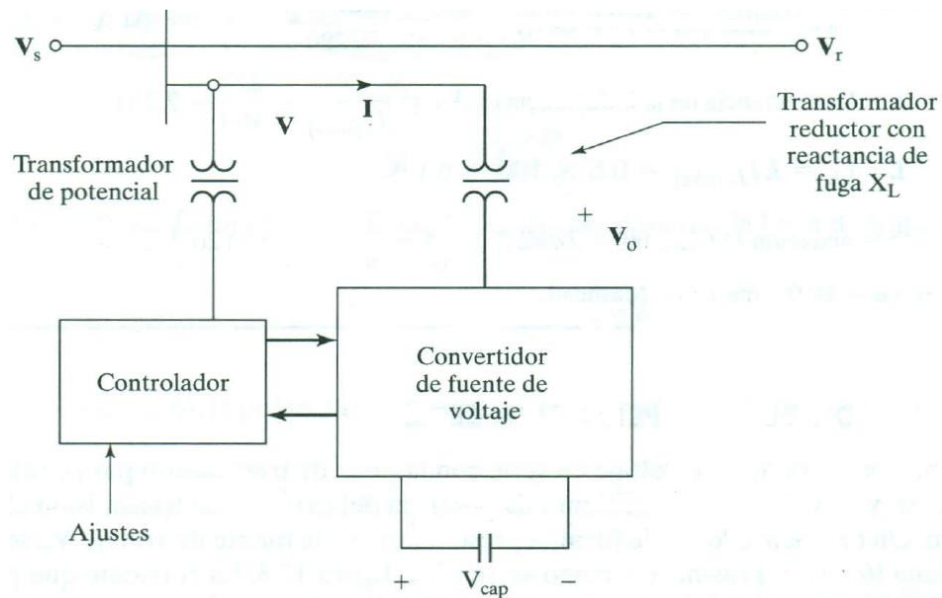
(c)

2.2 Compensador avanzado de VAR estático.

El compensador avanzado de VAR estático (STATCOM) es, en esencia, un convertidor con fuente de voltaje **Figura 7**[19], [20]. También puede sustituirse por un inversor con fuente de corriente, la función de este dispositivo es sincronizarse con la red eléctrica de modo que $V \angle \theta = V_o \angle 0$, si la tensión del dispositivo es mayor al de la red eléctrica como $V_o > V$, el dispositivo suministraría una corriente en adelanto comportándose como un capacitor, ahora si la tensión del dispositivo es menor al de la red eléctrica, este consumiría una corriente en atraso comportándose como un inductor. La idea es utilizar una estrategia de control adecuada para sincronizar el voltaje y el ángulo de fase del dispositivo con la red e ir controlando la amplitud dependiendo de la necesidad[21].

Este tipo de dispositivos nos permite 1) amplio intervalo de operación para proporcionar reactancia capacitiva aun con bajo voltaje, 2) menor especificación nominal que el SVC contraparte convencional, para lograr la misma estabilidad y 3) aumento de especificación de transitorios y mejor capacidad de manejo de perturbaciones dinámicas del sistema[19], [21].

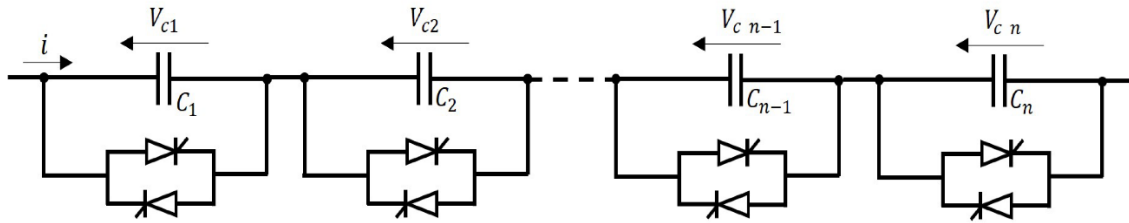
Figura. 7 Compensador avanzado de VAR estático.



(Rashid, 2004)

2.3 Capacitor serie conmutado por tiristor (TSSC).

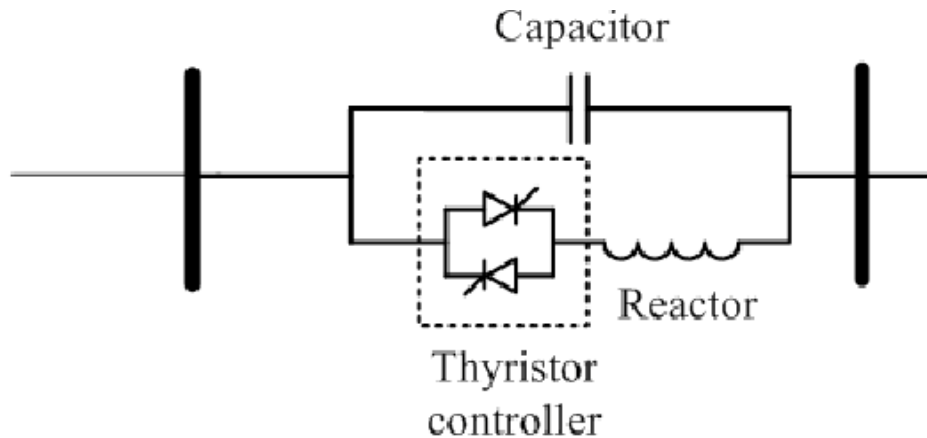
Este tipo de compensador consta de un grupo de capacitores en serie donde cada capacitor está en paralelo con un tiristor (TSSC, thyristor-switched series capacitor), esta configuración permite activar de forma escalonada (**Figura 8.**) la cantidad necesaria de capacitores como sea conveniente, permitiendo un amplio rango de compensación, no obstante los estados del capacitor son solo dos, la cantidad de combinaciones depende de $C = 2^n$, siendo n la cantidad de elementos y C el número de combinaciones posible[22]–[24].

Figura. 8 Capacitor en serie conmutado por tiristores

2.4 Capacitor serie controlado por tiristor (TCSC).

Este dispositivo es similar al TSSC (thyristor-switched series capacitor), y consiste en un capacitor en serie con la red eléctrica **Figura 9**[25], [26]., este capacitor a su vez está en paralelo con un reactor controlado por tiristor, la idea de este dispositivo es que el reactor con impedancia XL sea suficientemente pequeña, más que el capacitor XC . Esto permite operar en forma de encendido y apagado como el TSSC, ya que al variar el ángulo de retardo se puede variar la impedancia inductiva el TCR (Thyristor-switched reactor), de esta manera el TCSC puede tener un capacitor variable disminuyendo el rango capacitivo producto de la cancelación parcial que produce el TCR (Thyristor-switched reactor) en el capacitor[27].

Con una estrategia de control adecuada podemos controlar la reactancia total del sistema con la siguiente ecuación[27], [28]:

Figura. 9 Capacitor en serie controlado por tiristor.

$$X_T = \frac{X_C * X_L(\alpha)}{X_L(\alpha) - X_C} \quad (4)$$

Donde X_L lo podemos calcular como:

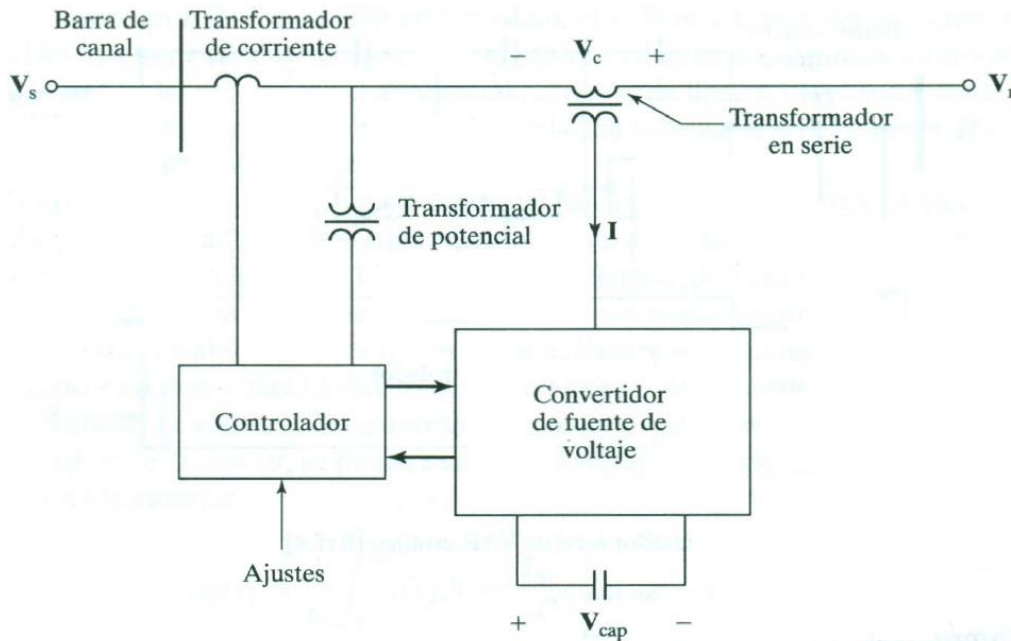
$$X_L(\alpha) = X_L * \frac{\pi}{\pi - 2\alpha - \sin 2\alpha} \quad (5)$$

La reactancia inductiva es $X_L = \omega * L$ y α es el ángulo de retardo medido desde el valor pico del voltaje del capacitor o del cruce de la corriente por cero.

2.5 Compensador SSVC avanzado.

Este dispositivo es la versión en serie del Compensador avanzado de VAR estático de la **Figura 7**, se conecta en el lado de baja a un inversor de fuente de voltaje (VSI) y el lado de alta está conectada en serie con la línea de transmisión **Figura 10**. Este tipo de compensador aparte de hacer las funciones del compensador avanzado de VAR estático (STATCOM) también se puede usar para aumentar la estabilidad contra transitorios, para amortiguar la resonancia subsíncrona donde se usan otros capacitores fijos, y para aumentar la capacidad de potencia de la línea.

Figura. 10 Compensador SSVC avanzado.



(Rashid, 2004)

CAPÍTULO 3. CARACTERÍSTICAS BÁSICAS DE UN D-STATCOM.

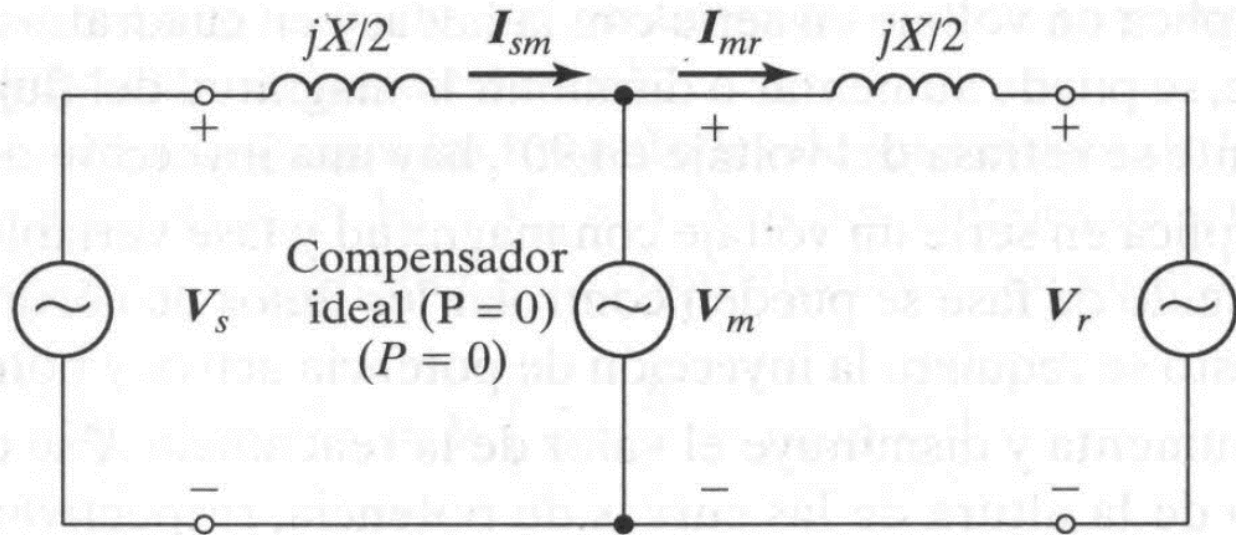
El D-STATCOM es el hermano menor del compensador avanzado de VAR estático (STATCOM) de la **Figura 7**, ya que este se utiliza en sistemas de distribución por los problemas con respecto a las variaciones de voltaje y como compensador de potencia reactiva[29]. Este problema es crítico en las líneas de distribución rurales largas, ya que los sistemas de regulación de voltaje generalmente están ubicados en la salida de la subestación o en puntos intermedios del alimentador[29], [30].

En los sistemas de distribución, los problemas de calidad de la energía surgen principalmente debido a las diferentes cargas no lineales y la expansión no planificada de los sistemas de distribución. Varios de estos problemas comprenden corrientes armónicas, carga desequilibrada, alta carga de potencia reactiva. El D-STATCOM es un dispositivo de potencia personalizado conectado en derivación que se utiliza para solucionar este tipo de problemas en sistemas de distribución[31], [32].

3.1 Caracterización del D-STATCOM.

Teóricamente el funcionamiento del D-STATCOM es bastante sencillo de entender, ya que consta de una fuente DC que alimenta una etapa inversora, este inversor por medio de una correcta estrategia de control se sincroniza su frecuencia, secuencia de fase y amplitud. En el momento calculado se conecta el inversor a la red eléctrica cuya función es tener una amplitud y ángulo de fase deseada, ya que el objetivo de este dispositivo es sincronizarse con la red eléctrica de modo que $V \angle 0 = V_o \angle 0$ **Figura 10**, si la tensión del dispositivo es mayor al de la red eléctrica como $V_o > V$ el dispositivo suministraría una corriente en adelanto comportándose como un capacitor, ahora si la tensión del dispositivo es menor al de la red eléctrica $V_o < V$, este consumiría una corriente en atraso comportándose como un inductor[33], [34].

El D-STATCOM usa el principio de la compensación por derivación **Figura 11**, donde esta debe ser capaz de suministrar potencia reactiva permitiendo minimizar el sobrevoltaje de línea bajo condiciones de carga ligera, y mantener los niveles de voltaje bajo condiciones de carga pesada. El dispositivo debería estar aproximadamente en un punto medio de la línea de distribución.

Figura. 11 Circuito compensador por derivación.

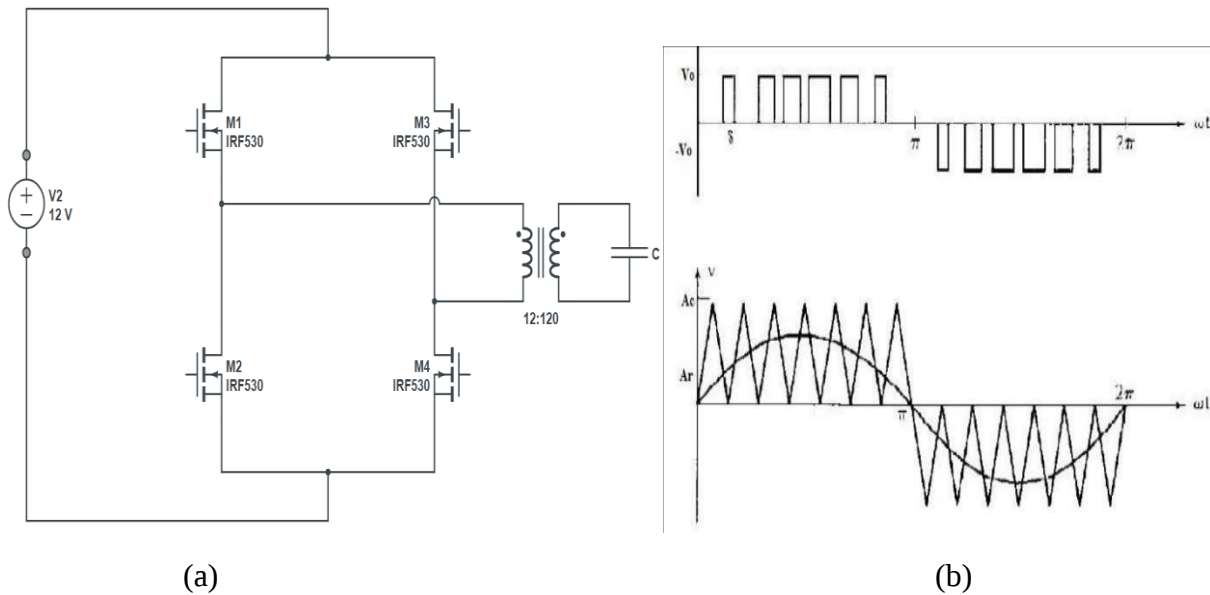
(Rashid, 2004)

Los sistemas eléctricos a nivel mundial y dependiendo de las regiones suelen tener un voltaje RMS de 120V y 220V, con una frecuencia de 50Hz o 60Hz, bajo estos datos necesitamos un inversor capaz de modificar su frecuencia a las especificaciones mencionadas, también necesitamos poder controlar la amplitud del voltaje a nuestra necesidad, estos serían dos de los pilares más importantes del dispositivo, a esto se le debe sumar sensores, controladores y funcionalidades complementarias que nos ayuden a realizar maniobras en tiempo real de acuerdo a las necesidades presentadas[35], [36].

3.1.1 Inversor de voltaje SPWM

El inversor de voltaje **Figura 12a**, tiene como finalidad convertir un nivel de tensión en DC a un voltaje en AC, con el control adecuado se alterna la activación de los mosfet M1, M2, M3 y M4. Cuando se activan los mosfet M1 y M4 se deben permanecer desactivados M2 y M3, con esta configuración la corriente circula por el transformador por el lado de baja en sentido horario, en el tiempo calculado se invierte la configuración activando los mosfet M2 y M3 desactivando M1 y M4 para que la corriente que circula por el lado de baja cambie de sentido, las conmutaciones se deben realizar con la finalidad de igualar las alternancias de un sistema en AC, esta frecuencia es de 50Hz o 60Hz a nivel comercial[37], [38].

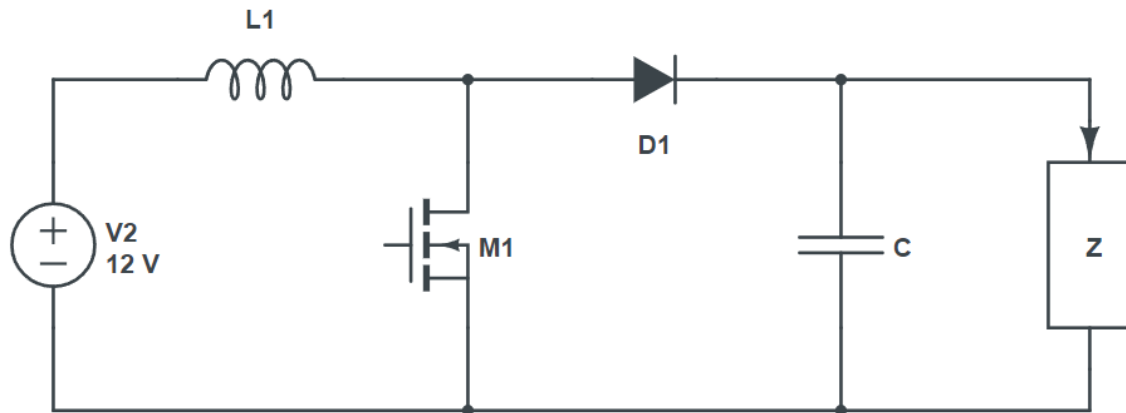
El inversor spwm (sine pulse width modulator) por sus siglas en inglés, significa modulador de ancho de pulso sinusoidal y su funcionamiento es generar ancho de pulsos variando su ciclo de trabajo **Figura 12b**, con la finalidad de cargar el capacitor conectado en el lado de alta a una velocidad controlada para igualar una forma de onda al de la red eléctrica[39], [40].

Figura. 12 Puente H para la etapa inversora tipo SPWM

3.1.2 Convertidor de voltaje DC-DC BOOST.

El convertidor de voltaje DC-DC Boost **Figura 13**, es una configuración de dispositivos electrónicos que tiene la finalidad de convertir un nivel de voltaje en DC a otro nivel de voltaje en DC, en el caso del convertidor elevador (Boost), este se encarga de convertir en nivel de voltaje a uno más elevado, su funcionamiento es simple, primero se activa y desactiva el mosfet M1 a una frecuencia determinada permitiendo la circulación de una corriente por el inductor L1 hacia tierra, en el momento de desactivar el mosfet M1 el inductor trata de mantener la misma corriente que está circulando al colapsar el campo magnético almacenado aumentando su nivel de tensión, estos picos de tensión se almacenan en el capacitor C, a una frecuencia ideal y conectado a la carga Z se puede mantener un nivel de voltaje deseado con un nivel de rizado despreciable.

Figura. 13 Convertidor de voltaje DC-DC Boost.



CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN DEL D-STATCOM

En este capítulo se explicará el diseño e implementación de un prototipo D-STATCOM basado en el funcionamiento básico que este tiene, partiendo de la unión de diseños eléctricos y electrónicos que nos permita abordar la problemática con una estrategia eficiente y lo menos complicada posible.

El diseño del prototipo D-STATCOM consta de múltiples funciones que nos permite mayor facilidad en el momento de aplicar el control necesario, en la **Figura 14**, se puede observar la tarjeta que incluye el puente H, un convertidor DC-DC, conexión para 2 Arduino nano para múltiples tareas, conexiones para diferentes sensores, detectores de secuencia de fase tanto para la red eléctrica como para el inversor mismo. Todas estas funcionalidades serán mencionadas con detalle una por una.

Figura. 14 Prototipo D-STATCOM.



4.1 Puente H

El puente H (**Figura 15.a**) está compuesto por 8 IGBT irgp50b60pd1 en paralelo por cada etapa, esta configuración se usó para dividir la carga de corriente por IGBT y disminuir su calentamiento, la conexión TRA1 y TRA2 se usa para conectar el transformador en el lado de baja.

La capacidad de corriente de un circuito depende del elemento más débil, aunque los IGBT puedan soportar una corriente de hasta 60A, las pistas de la tarjeta solo pueden soportar una fracción de esta corriente, por lo tanto, se deben añadir conexiones de refuerzo externa para poder soportar mayor cantidad de corriente, una de estas conexiones son las pistas de aluminio (**Figura 15.b**) en la que se puede agregar un fragmento de alambre calibre 10 u 8 que pueda soportar desde 40A hasta 70A.

Figura. 15 Puente H



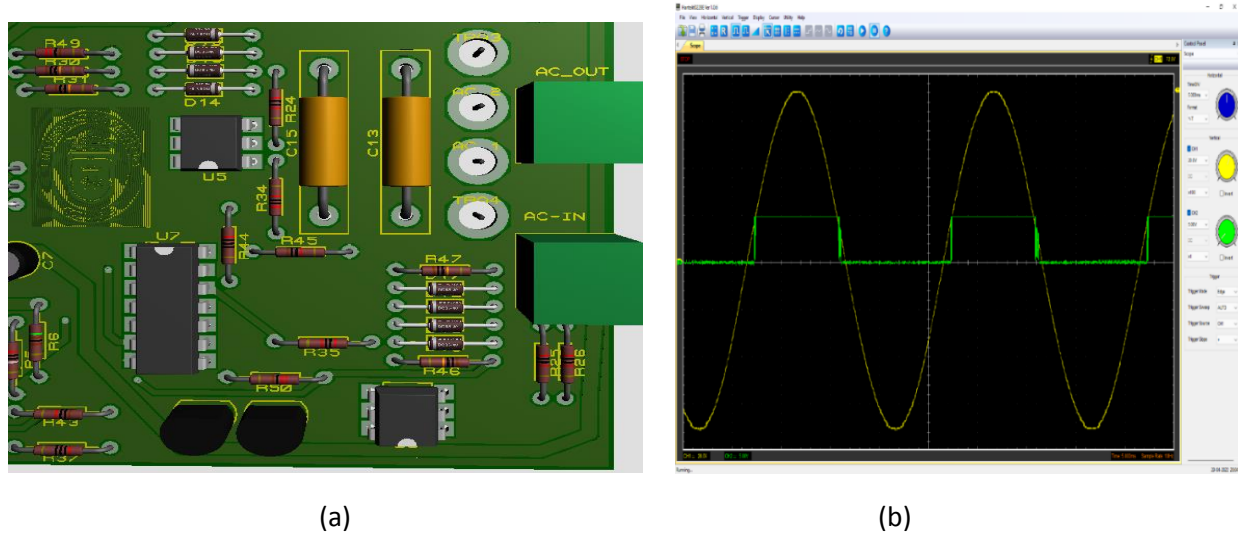
4.2 Detector de secuencia de fase.

La tarjeta cuenta con un circuito detector de secuencia de fase (**Figura 16**), este circuito está conformado por un optoacoplador 4N25 y 2 resistencias de 20K Ω para limitar la corriente de entrada, su funcionamiento es simple, en la conexión AC-IN se conecta la red eléctrica de referencia a la cual se pretende sincronizar, al igual que la entrada TRA3 y TRA4 donde se conecta el lado de alta del transformador. Por la configuración del optoacoplador la corriente solo circulara en secuencia positiva, esta señal se pasa por una compuerta lógica 74LS08 para convertirla en pulsos cuadrados. El pulso cuadrado generado por la señal de referencia y la señal del inversor se pasa por otra compuerta multiplicadora 74LS08 que sirve de referencia para ver que tanto se encuentra sincronizado las dos señales.

En la (**Figura 16.a**) se observa las entradas AC-IN y TRA3-TRA4, en la (**Figura 16.b**) se observa el pulso generado por la secuencia positiva tanto del inversor como de la señal de referencia AC-IN, estas señales

llegan al Arduino nano en las entradas digitales D2, D3 y D4 que esta incorporado en la tarjeta con el fin de usarlas para la sincronización del inversor con la red eléctrica.

Figura. 16 Circuito detector de secuencia de fase.

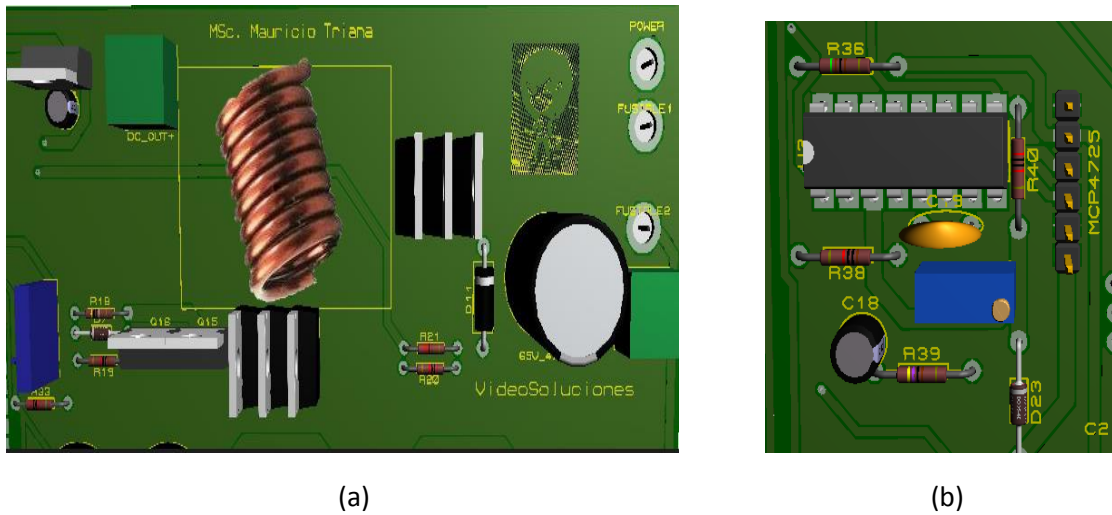


4.3 Convertidor DC-DC de tipo Boost.

En la figura (**Figura 17**) se observa la etapa del convertidor DC-DC de tipo elevador (Boost), la función del convertidor es controlar el voltaje DC en el puente H, ya que al aumentar o disminuir el voltaje en el puente H, modifica el voltaje en AC del lado de baja del transformador modificando el lado de alta, esta etapa es muy importante ya que el funcionamiento del D-STATCOM requiere estar modificando el voltaje dependiendo del factor de potencia medido.

En la (**Figura 17.b**) se encuentra el generador PWM que utilizamos para controlar el voltaje del convertidor DC-DC, este circuito está conformado por el TL494 y un convertidor MCP4725 que nos permite modificar el ancho del pulso para subir o bajar el voltaje DC.

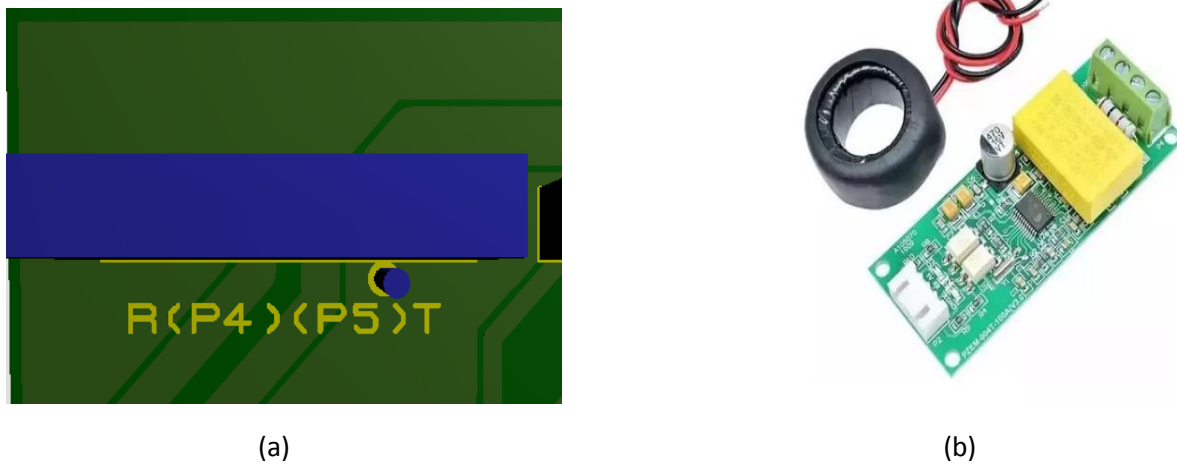
Figura. 17 Convertidor DC-DC de tipo Boost.



4.4 Entrada para el sensor PZEM-004T.

La tarjeta cuenta con una entrada (**Figura 18.a**) que permite conectar el sensor comercial PZEM-004T (**Figura 18.b**), este sensor nos sirve para obtener datos en tiempo real como el Voltaje, La corriente, La potencia, La frecuencia y el Factor de Potencia, este sensor va directamente conectado al Arduino denominado Master.

Figura. 18 Entrada del Sensor PZEM-004T.

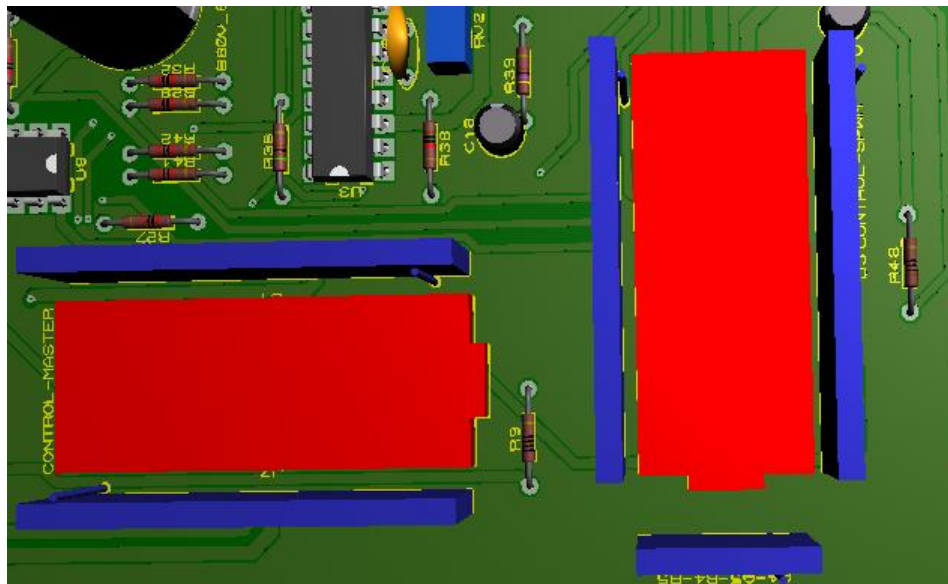


4.5 Control del D-STATCOM.

La tarjeta cuenta con 2 entradas para Arduino nano (**Figura 19**), una de las entradas se encarga de generar las señales SPWM que controlan el puente H (**Figura 14**), estas señales son las encargadas de crear la onda senoidal del inversor, esta tarjeta también recibe las señales provenientes del detector de secuencia de fase (**Figura 15**) permitiéndonos aplicar estrategias de control para poder sincronizar la señal del inversor con la señal de referencia de la red eléctrica.

El otro Arduino se encarga de recibir las señales provenientes del sensor PZEM-004T que nos indica datos esenciales como el Voltaje, la Corriente, la Potencia y el Factor de potencia, con estos datos obtenidos podemos utilizar diferentes estrategias de control para modificar la amplitud del inversor a través del TL494.

Figura. 19 Entrada de los Arduino nano.



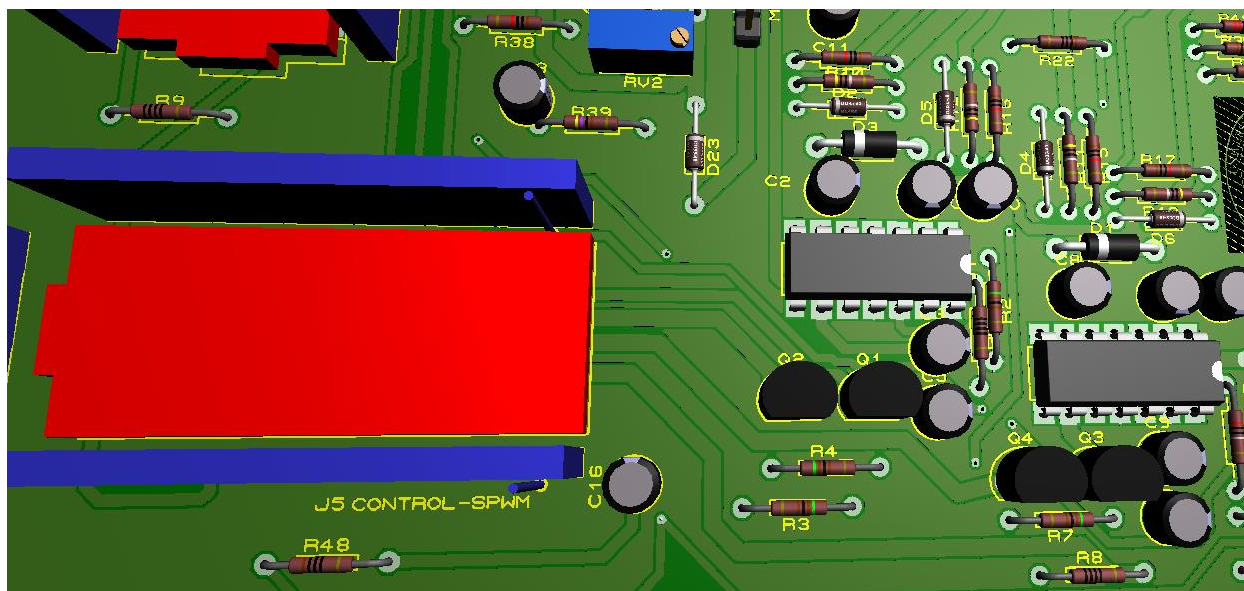
CAPÍTULO 5. ESTRATEGIAS DE CONTROL, PRUEBAS Y RESULTADOS

En este capítulo se aborda las diferentes estrategias de control aplicada a la tarjeta a nivel de código, haciendo énfasis en el hardware diseñado con el fin de facilitar el control. Se mencionará las diferentes funciones agregadas y como a través del código podemos darle la funcionalidad que requiere un D-STATCOM.

5.1 Inversor SPWM

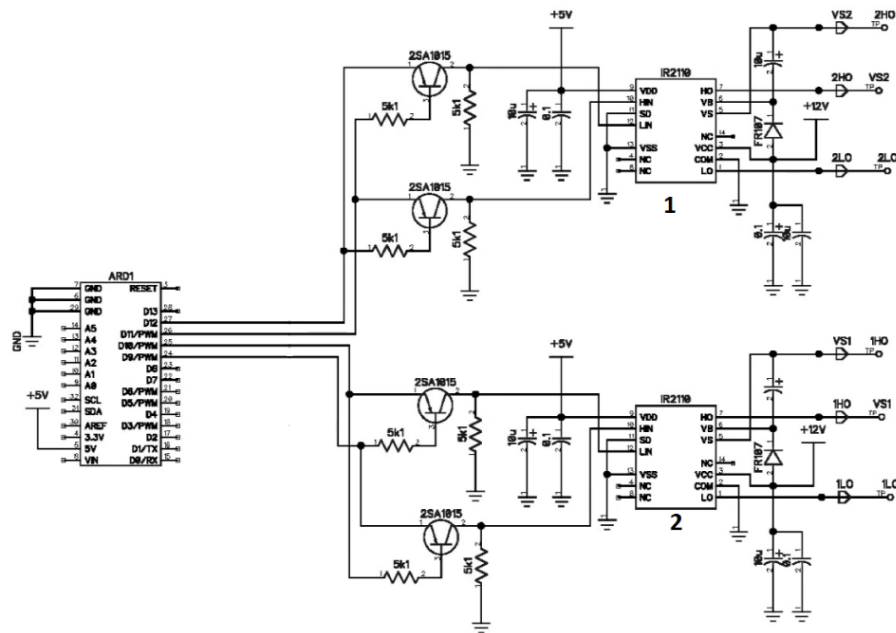
El funcionamiento del D-STATCOM es un convertidor de voltaje (Inversor) sincronizado con la red eléctrica, para eso se diseñó la etapa inversora compuesta por el puente H (**Figura 14**), esta es la encargada de crear la onda senoidal a partir de una fuente DC, el control de los pulsos de tipo SPWM lo realizamos a través del Arduino Control-SPWM (**Figura 20**), este Arduino tiene la funcionalidad de generar los pulsos necesarios en el puente H, pero para hacer la conexión del Arduino con los IGBT necesitamos de la intermediación de 2 controladores ir2110, ya que el Arduino solo genera señales de hasta 5V en DC no es suficiente para activar los IGBT, ya que estos requieren de al menos 12V en DC, con la ayuda del ir2110 podemos convertir los pulsos de 5V a los 12V que necesitan los IGBT generando la señal en alterna del lado de baja del transformador de acople.

Figura. 20 Entrada del Arduino Control-SPWM.

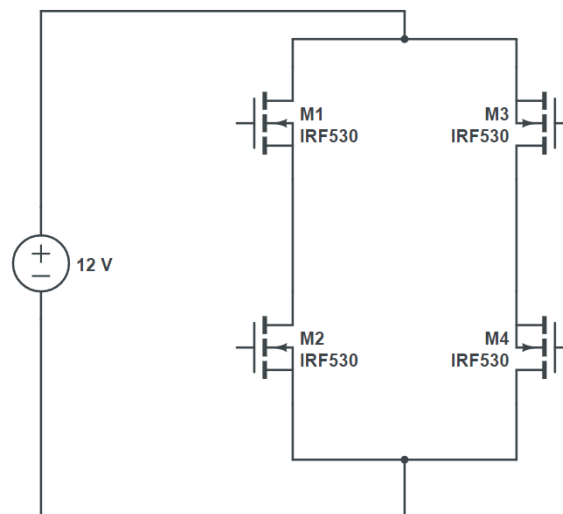


El diseño esquemático (**Figura 21.a**) encargado de la generación de los pulsos SPWM está compuesto por 2 controladores ir2110 encargados de convertir los pulsos de 5V en DC generado por el Arduino a pulsos de 12V en DC que necesita los IGBT, el ir2110 1 se encarga de activar los IGBT M1 y M2 (**Figura 21.b**), mientras que el ir2110 2 se encarga de activar los IGBT M3 y M4.

Figura. 21 Diseño esquemático del circuito generador SPWM



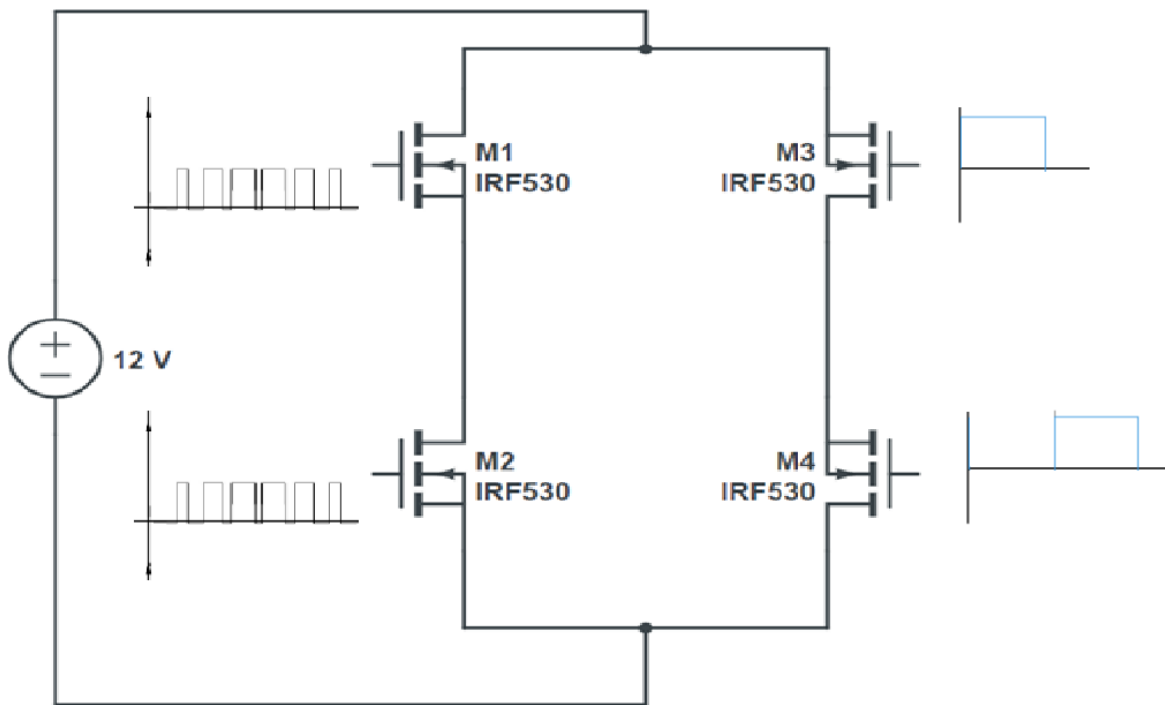
(a)



(b)

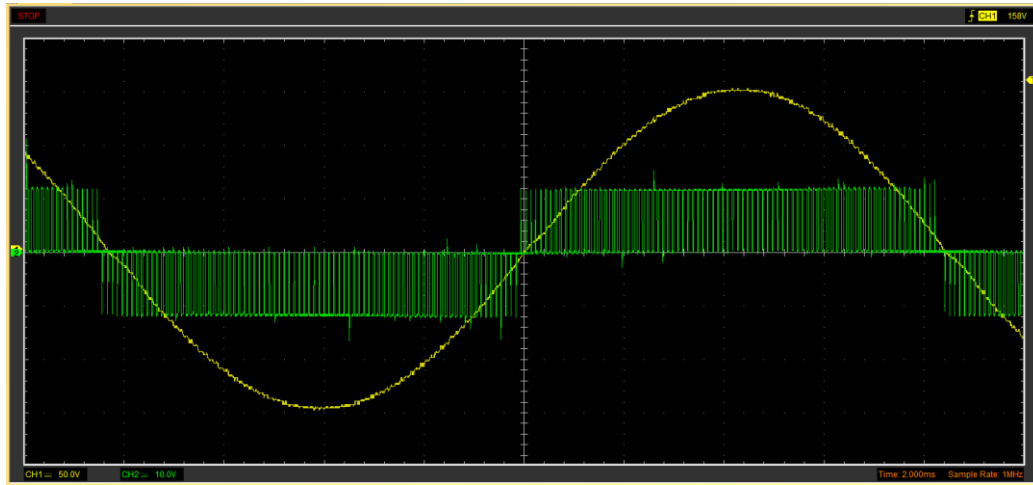
Los pulsos generados por el Arduino pasan a través de 2 transistores 2SA1015 en conexión antiparalelo con el fin de prevenir algún error a nivel de programación, ya que si por equivocación se activan 2 IGBT seguidos este generaría un corto en el puente H provocando posibles daños en la tarjeta. El ir2110 1 (**Figura 22.a**) recibe los pulsos SPWM mientras que el ir2110 2 recibe un pulso cuadrado con la duración del semiciclo de la señal en alterna que sería aproximadamente de 8,33ms teniendo los 60Hz de frecuencia, esta ilustración la podemos ver en la **Figura 22**.

Figura. 22 Señales del Arduino en el puente H



En la **Figura 22** podemos ver 2 tipos de señales, la señal de color verde son las generadas en el lado de baja del transformador, son señales SPWM con un ciclo de trabajo variable y un pico de voltaje en DC de 12V, estas señales se generan también en el lado de alta, pero gracias al capacitor que está conectado en paralelo podemos controlar la carga generando una señal AC de color amarillo similar al de la red eléctrica.

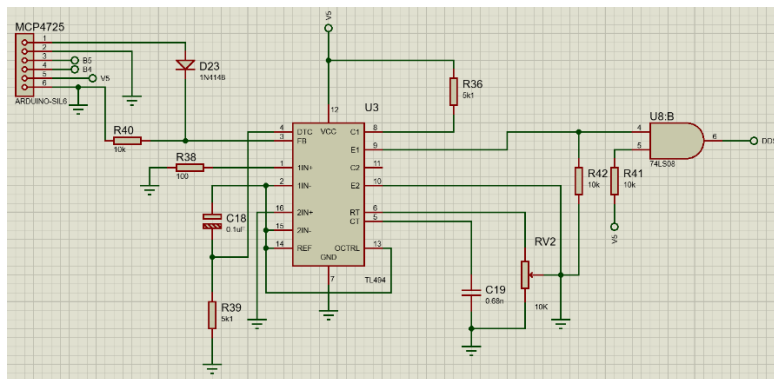
Figura. 23 Señales generadas en el transformador



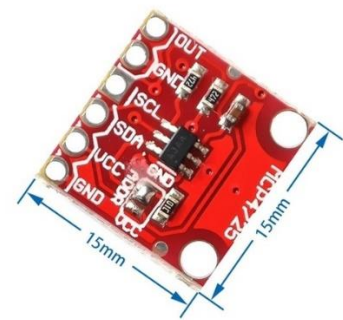
5.2 Convertidor DC-DC de tipo elevador

La funcionalidad del convertidor DC-DC elevador (**Figura 16**) es controlar el voltaje DC en el puente H (**Figura 21**), para esto se utilizó el generador PWM TL494 (**Figura 16.b**), el circuito diseñado se puede ver en la **Figura 23**.

Figura. 24 Circuito esquemático de control TL494



(a)

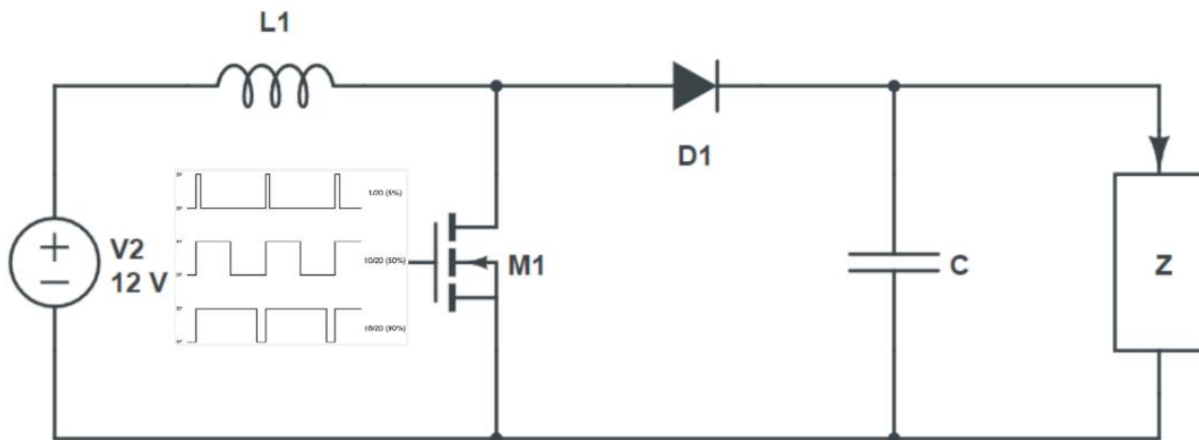


(b)

El circuito genera una señal SPWM cuya frecuencia depende del capacitor C19 y el potenciómetro RV2, con la variación del potenciómetro se puede modificar la frecuencia, el circuito cuenta con una entrada para el convertidor digital-analógico MCP4725 (**Figura 23.b**) que a través del diodo D23, nos permite generar una señal analógica de 0V a 5V en la entrada 3, esta entrada es de retroalimentación y la

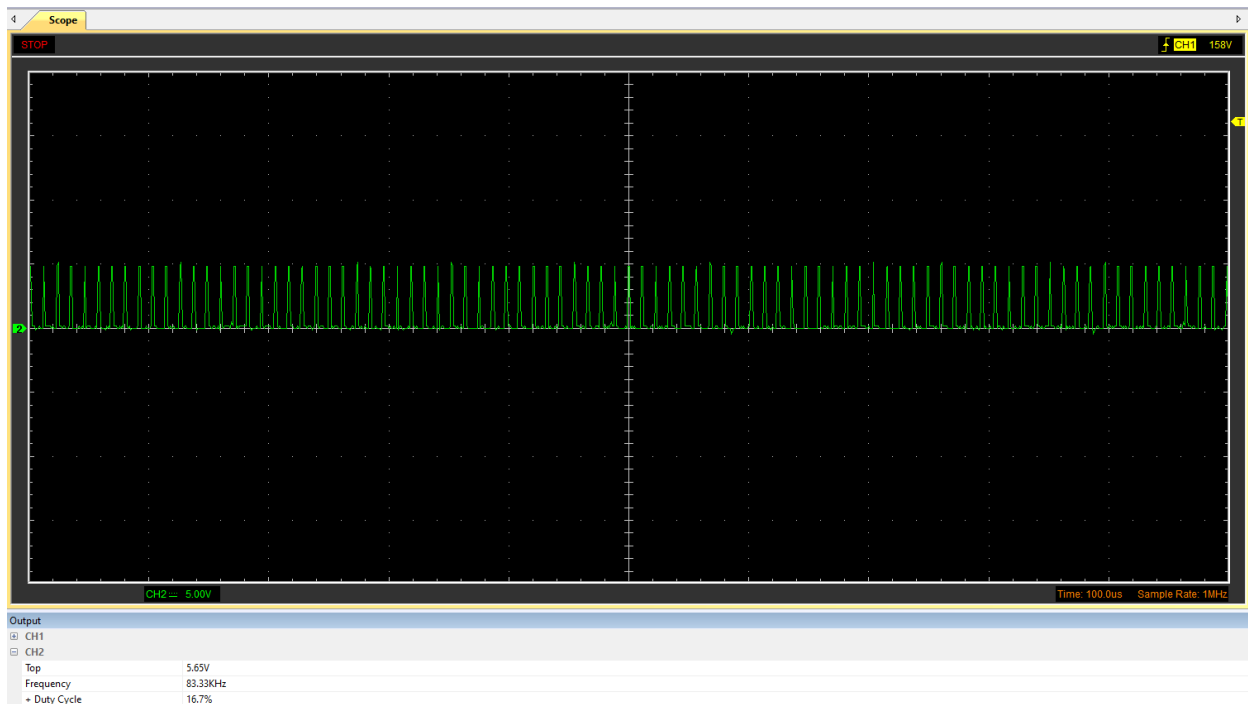
utilizamos para modificar el porcentaje del ciclo de trabajo de los pulsos PWM que llegan al convertidor DC-DC (**Figura 25**).

Figura. 25 Circuito esquemático del convertidor DC-DC con señales PWM



En el osciloscopio (**Figura 26**) podemos ver que el circuito funciona correctamente con una frecuencia de 83KHz aproximadamente, esta frecuencia se puede variar hasta 400KHz, pero se decidió usar una frecuencia menos a los 100KHz debido a que el Mosfet IRLZ44N que se usaron de interruptores en el convertidor DC-DC solo llega a los 100KHz según las especificaciones del fabricante.

Figura. 26 Pulsos PWM del convertidor DC-DC Boost.



5.3 Sincronización del inversor con la red eléctrica.

La tarjeta cuenta con un circuito detector de secuencia de fase (**Figura 15**) tanto para la red eléctrica a la que deseamos sincronizarnos, como a la etapa inversora incorporada en la tarjeta, este detector como se mencionó antes, detecta la secuencia de fase positiva y genera un pulso cuadrado que nos sirve de referencia.

Se diseñó 3 pulsos cuadrados con el fin de facilitar la sincronización utilizando diferentes técnicas de control ya que la tarjeta tiene fines investigativos, el pulso cuadrado generado por la secuencia de fase positiva de la red eléctrica llega a la entrada D2 del Arduino nano denominado CONTROL-SPWM, el segundo pulso cuadrado llega a la entrada D3 del mismo Arduino y es generado cuando se detecta la secuencia de fase positiva de la etapa inversora, y un tercer pulso cuadrado es generado por una compuerta lógica 74LS08 a la que llegan los dos pulsos anteriores, este tercer pulso se adicionó para indicarnos que porcentaje de sincronización tiene ambas señales de referencia, no obstante, se utilizó solo la señal de referencia de la red eléctrica a sincronizar en este trabajo.

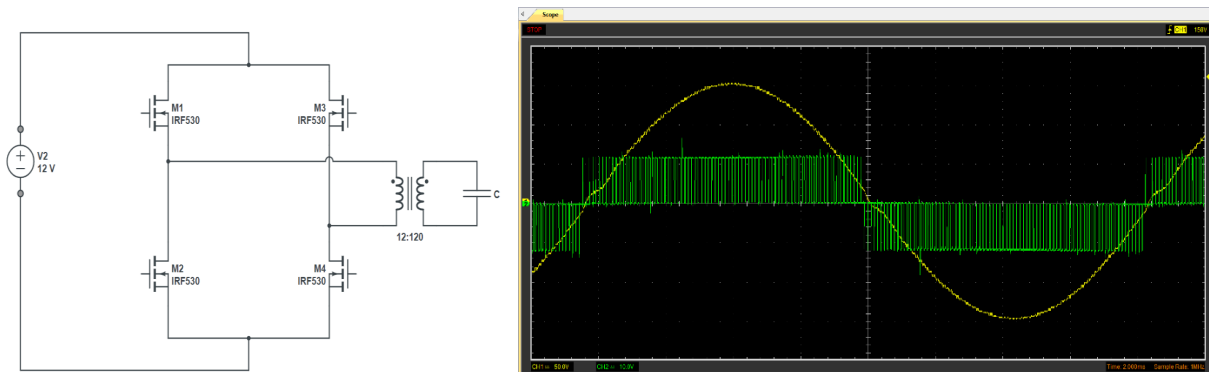
La generación SPWM se realizó por medio de una lista de pulsos (**Figura 27**) denominada **lookUp1**, cada dato de la lista representa un porcentaje del ciclo de trabajo PWM, estos pulsos de color verde (**Figura 28.b**) se generan en el lado de baja del transformador de acople. En el lado de alta los pulsos tiene como

función cargar y descargar el capacitor (**Figura 28.a**) de forma parcial creando la forma de onda senoidal de color amarillo (**Figura 28.b**)

Figura. 27 Lista de pulsos PWM

```
int lookUp1[] = {
  0, 30, 60, 90, 120, 150, 179, 208, 237, 266, 294, 322, 349, 376, 402, 428, 453, 478, 501, 524,
  547, 568, 589, 609, 628, 646, 664, 680, 695, 710, 723, 735, 747, 757, 766, 774, 781, 787, 792, 796,
  798, 800, 800, 799, 797, 794, 790, 784, 778, 770, 762, 752, 741, 729, 717, 703, 688, 672, 655, 637,
  619, 599, 579, 558, 536, 513, 489, 465, 441, 415, 389, 363, 335, 308, 280, 252, 223, 194, 164, 135,
  105, 75, 45, 15
};
```

Figura. 28 Generación de la onda senoidal de la etapa inversora



Después de haber observado el funcionamiento de la etapa inversora de la tarjeta D-STATCOM se estableció la siguiente estrategia de control, haciendo uso de la señal de secuencia positiva (**Figura 15**) que llega al Arduino denominado CONTROL-MASTER en la entrada D2, con la siguiente línea de código (**Figura 29**) se establece una sentencia hacia una función denominada verificar, esta función se activa cuando se detecta un flanco positivo denominado **RISING**.

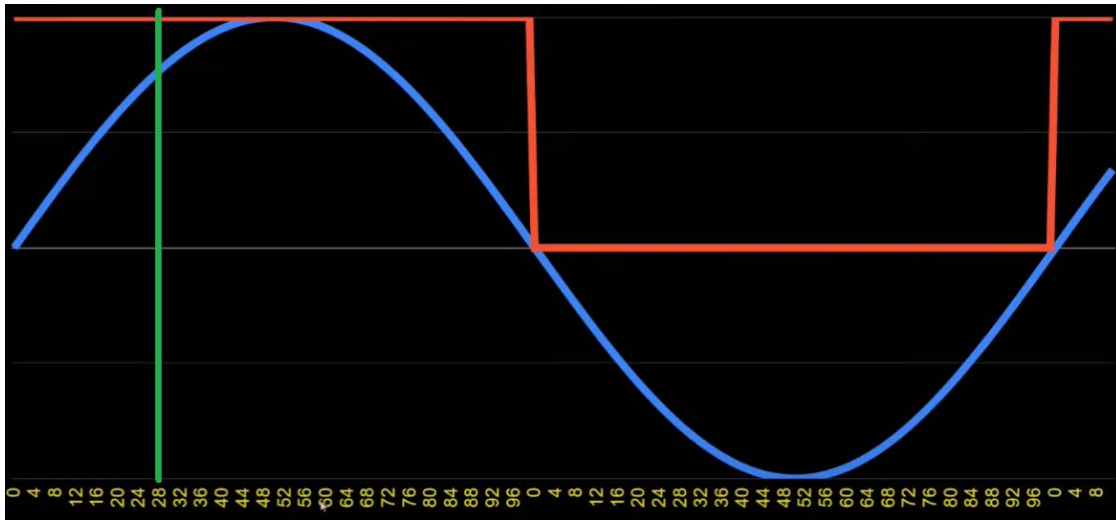
Figura. 29 Activación de la función verificar

```
attachInterrupt(digitalPinToInterrupt(2), verificar, RISING);
```

El diseño de la tarjeta permite diferentes tipos de control, para este proyecto y con la facilidad del hardware se utilizó un control proporcional haciendo uso de la señal de secuencia positiva (**Figura 15**) que nos activa la función **verificar**, la función verifica una variable denominada **num** que recorre los valores de la tabla **lookUp1[num]** (**Figura 26**), estos valores son los encargados de generar medio ciclo de la onda senoidal y se repite tanto para secuencia positiva como negativa completando la el ciclo. La estrategia de

control radica en el momento de que se activa la función **verificar** la variable **num** contiene un valor determinado, y recordando que la función se activa justo en el momento que se detecta el flanco de subida del detector de secuencia positiva (**Figura 15**) de la red eléctrica indicando que inicia la señal, este pulso de la red lo podemos ver en color verde (**Figura 30**). Si observamos la señal detectada de la red eléctrica en color verde (**Figura 30**) y el valor de la variable **num** podemos ver que tan desfasada se encuentra la etapa inversora con la red eléctrica, permitiéndonos ajustar el valor de la variable **num** ajustando la señal de la etapa inversora a la señal proporcionada por la red eléctrica.

Figura. 30 Generación de la onda senoidal de la etapa inversora



El control proporcional (**Figura 31**) esta compuesto por la referencia $r(t)$, la salida $y(t)$, el error $e(t)$, la ganancia proporcional K_p , la entrada al proceso $u(t)$, y el proceso P , el diseño del hardware y las características mencionadas nos permite a simplificar el control proporcional, donde:

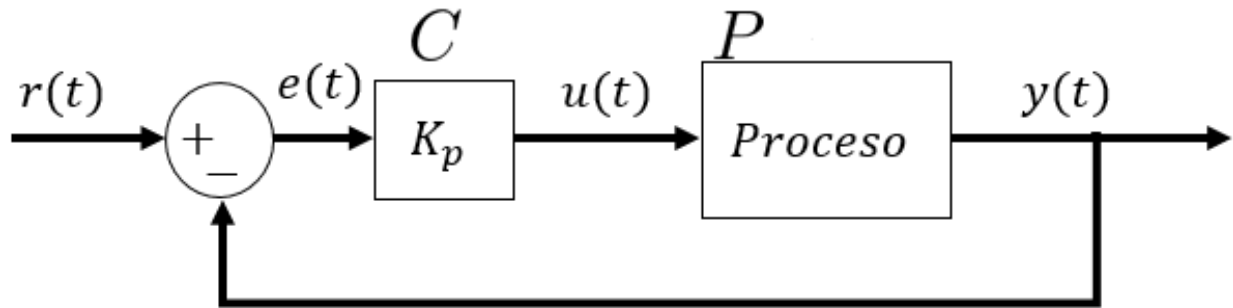
$$e(t) = y(t) - r(t) \quad (6)$$

$$P = \text{num} - e(t) \quad (7)$$

Para determinar el error $e(t)$ debemos recordar que lo ideal es que la señal de color verde (**Figura 29**) que surge en el momento que empieza la secuencia positiva de la red eléctrica coincida con la variable **num** cuando su valor sea cero en el recorrido de los datos **lookUp1[0]**, ya que ese sería el inicio de la secuencia positiva de la etapa inversora, por lo tanto nuestra referencia es **num = 0**, y nuestra salida sería el valor actual de la variable **num** en el momento que se activa la función **verificar**. En el momento de calcular el error tomando el ejemplo de la **Figura 29** observaríamos que el valor actual de la variable de salida $y(t)$ sería **num=8**, y como nuestra referencia es un valor de **num=0** el error sería de 8 pasos en la tabla **lookUp1[8]**, por lo tanto al proceso $P = \text{num} - e(t)$ sería restarle esos 8 pasos para que en la próxima activación de la función **verificar** coincida el inicio de la secuencia positiva de la red eléctrica con el inicio de la generación de la secuencia positiva de la etapa inversora, observemos que en este caso la ganancia

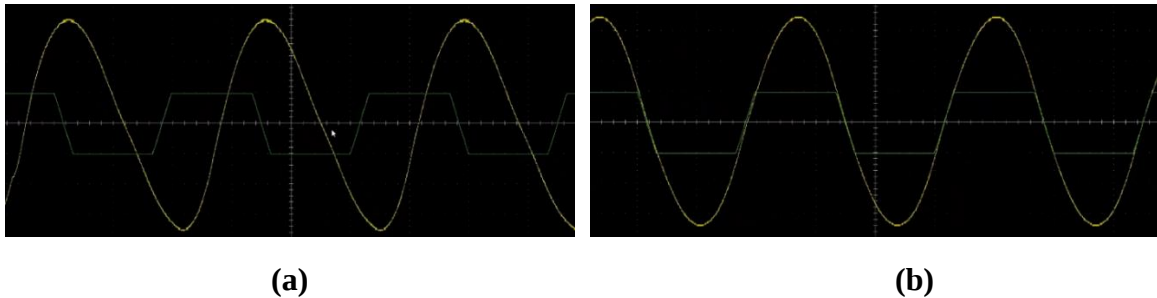
proporcional se omitiría o simplemente sería de valor **1** ya que los valores a corregir serían de tipo discreto.

Figura. 31 Representación del control proporcional



En el momento de aplicar el control a la sincronización de la etapa inversora de color amarillo con la red eléctrica de color verde observamos que inicialmente se encuentran desfasadas (**Figura 32.a**), casi inmediatamente la señal de la etapa inversora se ajusta a la señal de referencia manteniendo una sincronización bastante precisa demostrando que solamente utilizando un control proporcional simple se obtiene buenos resultados, no obstante, el diseño y la construcción del prototipo D-STATCOM es fundamental para tal simplicidad en el control mencionado.

Figura. 32 Sincronización de la etapa inversora con la red eléctrica.

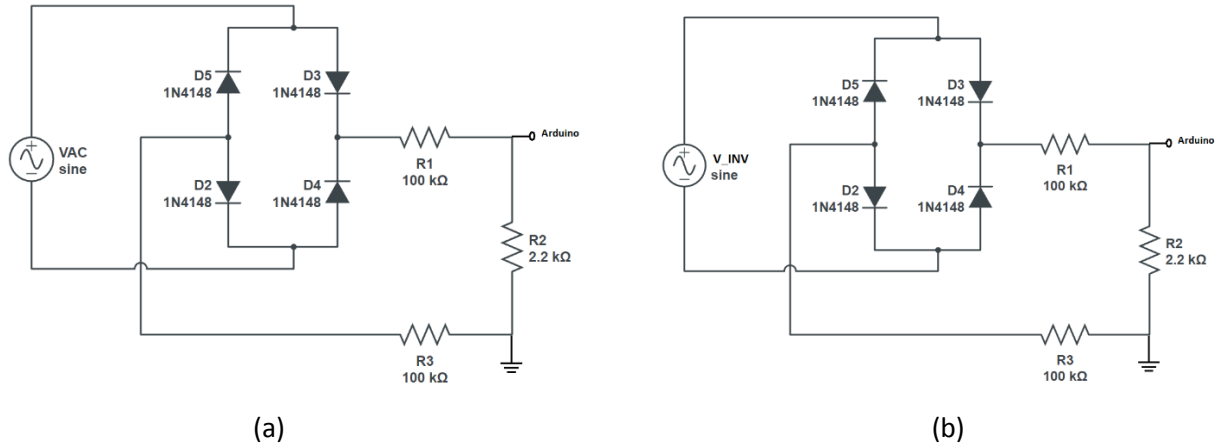


5.4 Control y sincronización del voltaje RMS de la etapa inversora con la red AC.

El control de la amplitud de la etapa inversora se hace a través del convertidor DC-DC (**Figura 16**). Para eso se diseñó dos circuitos (**Figura 33**) con los mismos valores de los componentes eléctricos que nos da

una muestra del voltaje instantáneo que tiene tanto la red eléctrica de referencia a la que se desea sincronizar como la etapa inversora.

Figura. 33 Monitoreo del voltaje del inversor y de la red eléctrica.

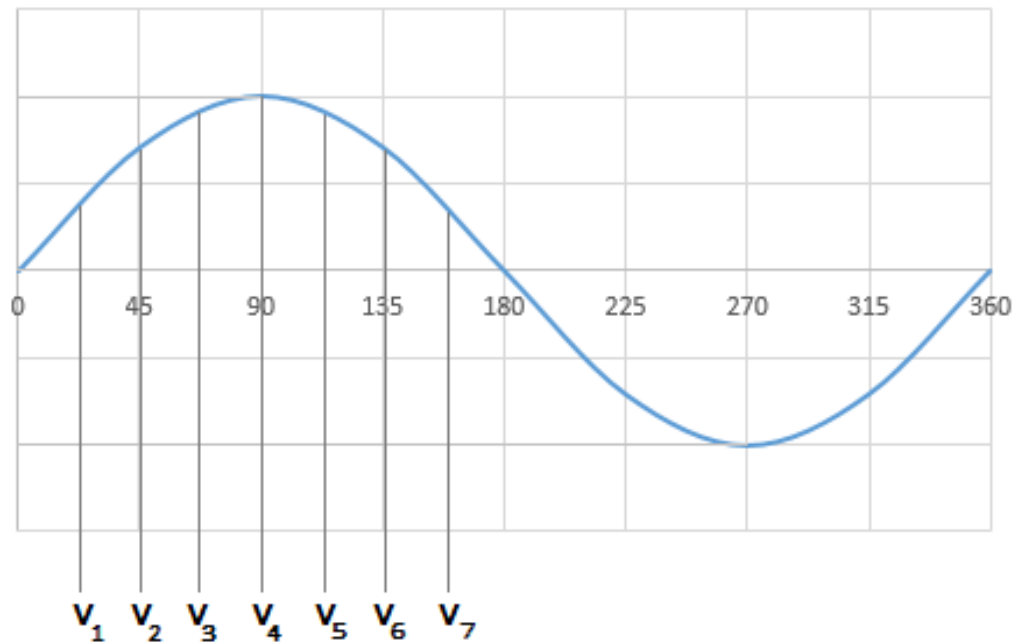


El circuito (**Figura 33**) permite una pequeña circulación de corriente que para el caso de la red eléctrica suponiendo un voltaje pico de 170V, la corriente máxima sería de tan solo 0.84mA, donde al obtener la diferencia de potencial en la resistencia de 2.2KΩ referente a tierra obtendríamos un voltaje de 1.84V máximo en DC, lo cual es un valor tolerable para el Arduino ya que en una de sus entradas analógicas no superaría los 5V que puede soportar.

Para poder monitorear el voltaje RMS tanto de la etapa inversora como de la red eléctrica AC de referencia, se utilizaron las entradas analógicas A0 y A3 del Arduino, ambas de 10 bits, debido a que el Arduino detecta un voltaje instantáneo se realizó varias mediciones (**Figura 33**) y se aplicó la fórmula de la ecuación 8 para calcular el voltaje RMS.

$$V_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N V_i^2} = \sqrt{\frac{V_1^2 + V_2^2 + \dots + V_N^2}{N}} \quad (8)$$

Figura. 34 Toma de datos para calcular el voltaje RMS.



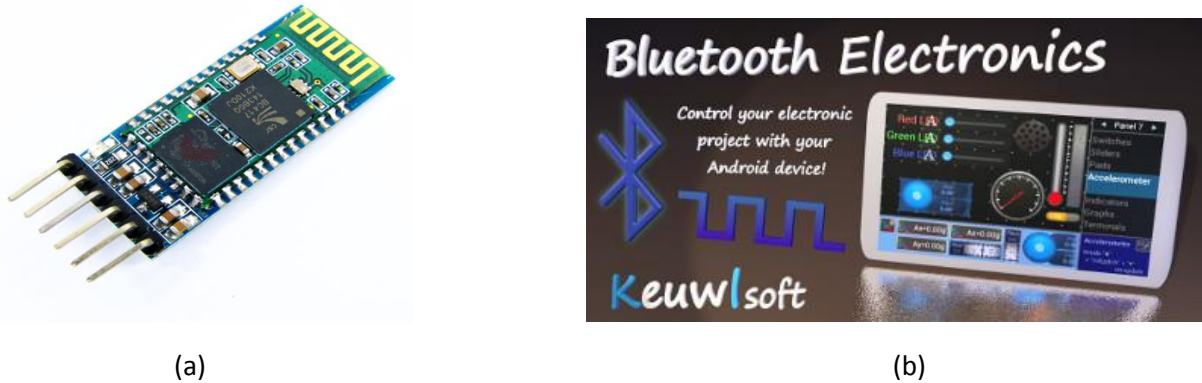
A nivel de código (**Figura 34**) se realizó alrededor de 8400 mediciones con un intervalo de 1 microsegundo ya que la frecuencia de trabajo de la zona es de 60Hz, lo cual el tiempo que demora medio ciclo de trabajo son de alrededor 8,33ms o 8333 microsegundos, este código se repite tanto para el voltaje de red como el voltaje del inversor.

Figura. 35 Cálculo de los voltajes RMS en Arduino.

```
for(int i=0;i<8400;i++){
  lectura2 = analogRead(entrada2);
  Vref = lectura2*lectura2+Vref;
  delayMicroseconds(1);
}
Vref = ((sqrt(Vref/8400))/205)*(202000/2000);
for(int i=0;i<8400;i++){
  lectura = analogRead(entrada);
  vinversor = lectura*lectura+vinversor;
  delayMicroseconds(1);
}
vinversor = ((sqrt(vinversor/8400))/205)*(202000/2000);
```

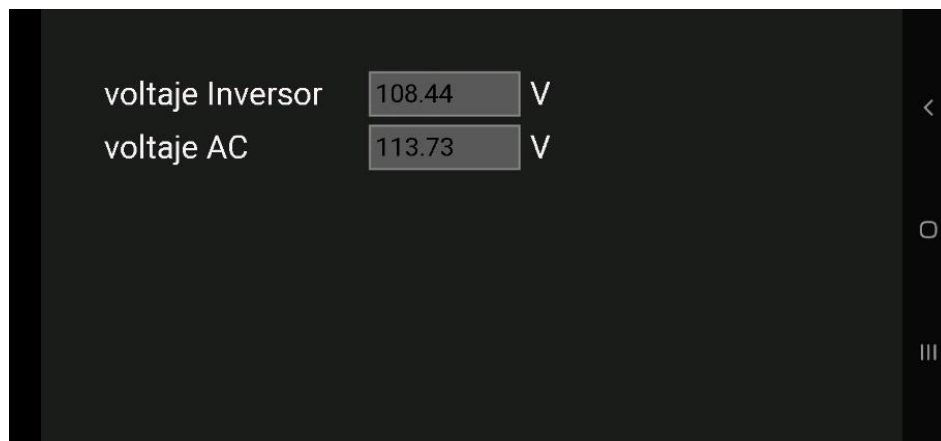
El monitoreo de los voltajes RMS y de los demás datos se está realizando con el módulo **Bluetooth HC-05** y la aplicación **Bluetooth Electronics (Figura 35)** que se instaló en el celular.

Figura. 36 Módulo Bluetooth y aplicación móvil.



Con ayuda de este módulo bluetooth, se evitó utilizar pantallas de diferente tipo en el monitoreo de datos, una muestra de la eficacia del módulo son los valores monitoreados (**Figura 36**) de la red eléctrica y la etapa inversora.

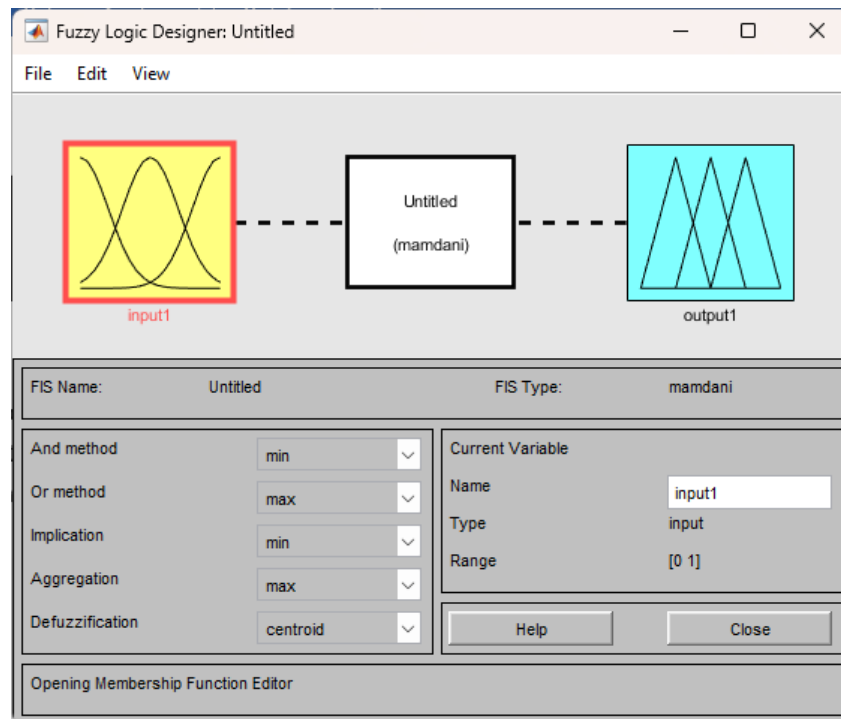
Figura. 37 Módulo Bluetooth y aplicación móvil.



Observamos que los valores de los voltajes RMS tanto de la etapa inversora como la red eléctrica de referencia se muestran en el dispositivo móvil, con estos valores podemos crear la estrategia de control que nos permita sincronizar ambos voltajes.

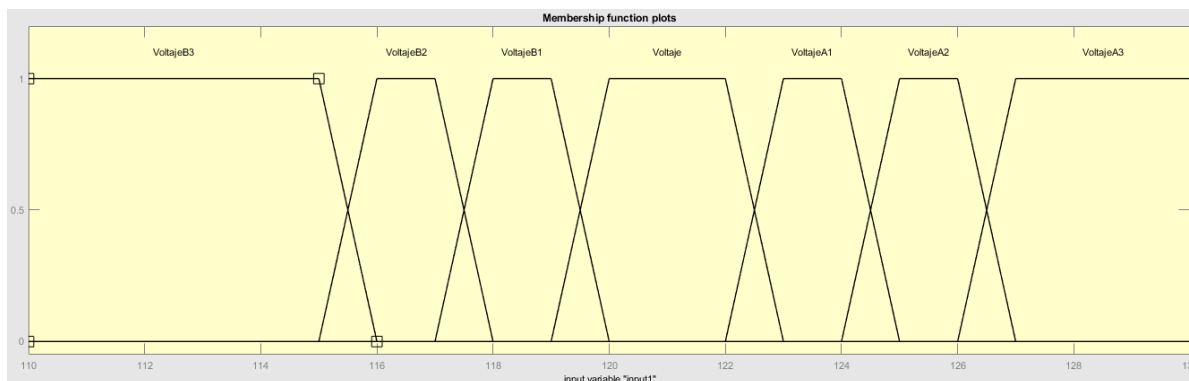
El control de voltaje se realizó utilizando Lógica difusa, para ello se utilizó la herramienta de Fuzzy Logic Designer en Matlab (**Figura 37**), con esta herramienta se realizó la simulación del comportamiento que debe tener el ciclo de trabajo del convertidor DC-DC de tipo elevador con el voltaje medido de la etapa inversora.

Figura. 38 Módulo Bluetooth y aplicación móvil.



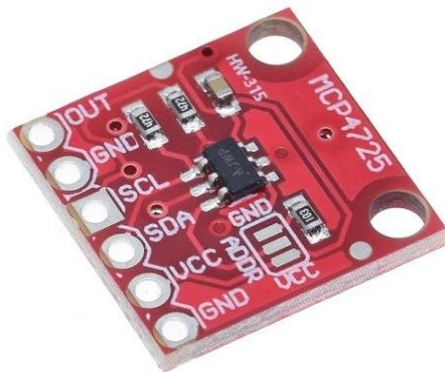
Para realizar la simulación establecemos un valor hipotético de referencia de unos 120Vrms, no obstante en la práctica este valor se mide de la red eléctrica, pero para la simulación solo necesitamos ver que las reglas de control funcione de forma adecuada con la variación del voltaje de la etapa inversora, en la (**Figura 38**) vemos los posibles valores medidos de la etapa inversora como entrada en el sistema fuzzy, nombrando **Voltaje** como el valor deseado, **voltajeB** como voltaje bajo donde tenemos 3 funciones de membresía (B1,B2,B3), **VoltajeA** como voltaje alto donde también tenemos 3 funciones de membresía (A1,A2,A3).

Figura. 39 Funciones de membresía de entrada.

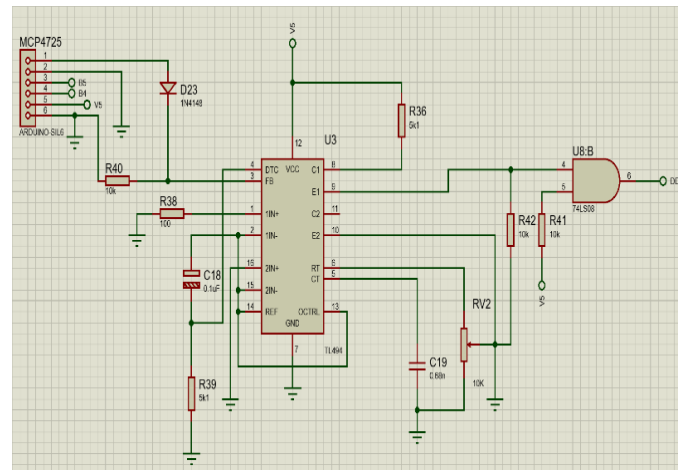


Con las funciones de membresía establecidas de entrada, se realizó las funciones de membresía de salida, que están asociadas a un convertidor digital analógico DAC MCP4725 (**Figura 39.a**) conectado a la entrada Feedback del TL-494 (**Figura 39.b**) en el pin 3, con este convertidor de 12 bits tenemos una variación de voltaje de 0 a 5V con unos 4096 pasos, lo que significa que cada paso equivale a 1,22mV en DC. El objetivo de controlar el voltaje en la entrada 3 del TL-494 es controlar el ciclo de trabajo en su salida que va directamente al convertidor elevador DC-DC pudiendo controlar el voltaje DC en el puente H, si en la entrada 3 el voltaje es cero, el ciclo de trabajo es máximo, pero si el voltaje es de aproximadamente 3,5V en DC, el ciclo de trabajo es de cero voltios manteniendo el voltaje en el convertidor DC-DC sin variación con únicamente el voltaje de entrada, que en nuestro caso serían los 12V que le llegan al puente H, esto significa que en el momento de encender el prototipo **D-STATCOM** el voltaje que debería llegarle al TL-494 deben ser de al menos 3,5V para empezar con un valor fijo en la entrada del puente H de la etapa inversora, para obtener este valor se programa el **DAC MCP4725** con un número de referencia poco más de 2870, pero para prevenir cualquier caída de voltaje se decidió un valor de 3000 pasos, lo que significa un voltaje alrededor de 3,66V en DC, con esto en el momento de realizar la medición y si se necesita aumentar el voltaje DC del puente H, se disminuye los pasos del **DAC MCP4725**, esto ocasionará que bajara el voltaje en la entrada 3 del TL-494 generando un aumento en el ciclo de trabajo que va al convertidor DC-DC provocando un aumento de voltaje en el puente H que a su vez aumenta en el voltaje en AC tanto del lado de baja como del lado de alta en el transformador.

Figura. 40 Control del ciclo de trabajo del TL-494.



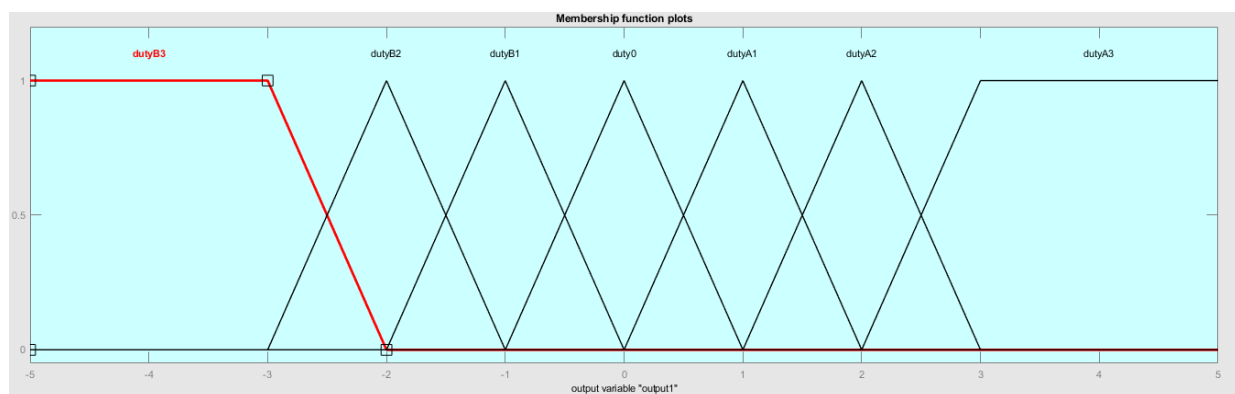
(a)



(b)

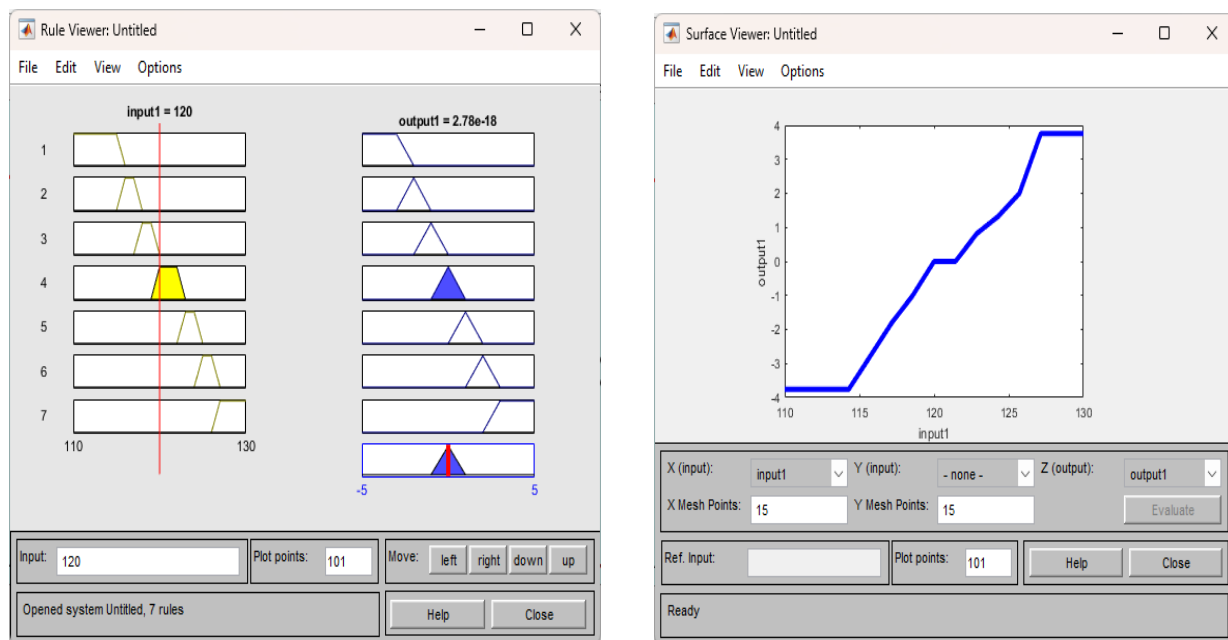
Como pudimos observar, el aumento o la disminución de la amplitud que permite igualar el voltaje de la etapa inversora con el voltaje de la red eléctrica va directamente asociado al control de los pasos del **DAC MCP4725**, si disminuyo los pasos aumenta el voltaje y si aumento los pasos el voltaje disminuye, con esta información se establecieron las funciones de membresía de salida (**Figura 40**), donde el rango va de -5 hasta 5, al igual que la entrada, la salida tiene 7 funciones de membresía que nos dicen si debemos aumentar o disminuir los pasos del **DAC** conforme vamos monitoreando el voltaje de entrada de la etapa inversora.

Figura. 41 Funciones de membresía de salida.



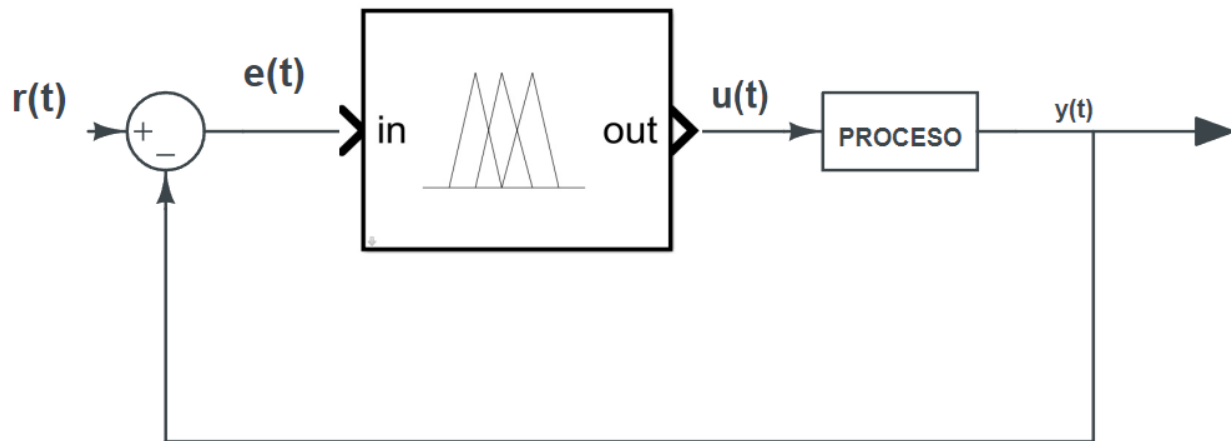
Cada entrada va directamente asociada a la salida, como podemos ver cuando el voltaje medido de la etapa inversora es igual a la entrada de la red eléctrica, la salida indica que los pasos que debemos modificar en el **DAC MCP4725** serian cero, pero si el voltaje es bajo los pasos del **DAC MCP4725** disminuyen para aumentar la amplitud de la etapa inversora, en el caso contrario donde el voltaje este por encima, los pasos aumentan disminuyendo la amplitud de la etapa inversora, en la (Figura 41) podemos ver las reglas de control, para esto se utilizó el método de mandaní.

Figura. 42 Reglas de control.



El control se realiza con la entrada de referencia, en el cual se observa el error entre el valor deseado y la salida monitoreada, cuando el error entra al control difuso este nos da un valor del DAC que va al proceso modificando el voltaje de salida.

Figura. 43 Conexión del control de voltaje difuso



Para la implementación en el Arduino se utilizó la librería eFLL (Embedded Fuzzy Logic Library), esta librería de código abierto nos permite aplicar las condiciones de entrada y salida previamente analizadas en el programa Fuzzy Logic Designer de Matlab (**Figura 37**).

Figura. 44 Librería eFLL (Embedded Fuzzy Logic Library)

eFLL de AJ Alves <aj.alves@zerokol.com>, Dr. Ricardo Lira...

1.4.1 instalado

eFLL (Embedded Fuzzy Logic Library). eFLL is a standard library for Embedded Systems to implement easy and efficient Fuzzy Systems.

[Más información](#)

1.4.1 ▼

ELIMINAR

Para la implementación de la librería se empieza declarando las funciones de membresía de entrada **Figura**, Se declararon 7 funciones de membresía, la llamada voltaje0 es el punto deseado de voltaje, que sería el voltaje monitoreado de la re3d eléctrica, los voltajes B1, B2, B3 serían funciones de membresía por debajo del deseado, y los voltajes A1, A2 y A3 serían voltajes de membresía por encima del deseado.

Figura. 45 Funciones de membresía de entrada.

```

FuzzyInput *voltaje = new FuzzyInput(1);
FuzzySet *voltajeB3 = new FuzzySet(Vref-6, Vref-6, Vref-5, Vref-4);
voltaje->addFuzzySet(voltajeB3);
FuzzySet *voltajeB2 = new FuzzySet(Vref-5, Vref-4, Vref-3, Vref-2);
voltaje->addFuzzySet(voltajeB2);
FuzzySet *voltajeB1 = new FuzzySet(Vref-3, Vref-2, Vref-1, Vref);
voltaje->addFuzzySet(voltajeB1);
FuzzySet *voltaje0 = new FuzzySet(Vref-1, Vref, Vref+2, Vref+3);
voltaje->addFuzzySet(voltaje0);
FuzzySet *voltajeA1 = new FuzzySet(Vref+2, Vref+3, Vref+4, Vref+5);
voltaje->addFuzzySet(voltajeA1);
FuzzySet *voltajeA2 = new FuzzySet(Vref+4, Vref+5, Vref+6, Vref+7);
voltaje->addFuzzySet(voltajeA2);
FuzzySet *voltajeA3 = new FuzzySet(Vref+6, Vref+7, Vref+8, Vref+8);
voltaje->addFuzzySet(voltajeA3);
fuzzy->addFuzzyInput(voltaje);

```

Las funciones de membresía de salida **Figura 46** también son 7, en el cual sería la modificación del duty que se aplicaría en el DAC. Cuando el valor del duty sea cero, significa que el voltaje monitoreado es el deseado, de lo contrario sería un valor positivo o negativo.

Figura. 46 Funciones de membresía de salida

```

FuzzyOutput *duty = new FuzzyOutput(1);
FuzzySet *dutyN3 = new FuzzySet(-4, -3, -3, -2);
duty->addFuzzySet(dutyN3);
FuzzySet *dutyN2 = new FuzzySet(-3, -2, -2, -1);
duty->addFuzzySet(dutyN2);
FuzzySet *dutyN1 = new FuzzySet(-2, -1, -1, 0);
duty->addFuzzySet(dutyN1);
FuzzySet *duty0 = new FuzzySet(-1, 0, 0, 1);
duty->addFuzzySet(duty0);
FuzzySet *dutyP1 = new FuzzySet(0, 1, 1, 2);
duty->addFuzzySet(dutyP1);
FuzzySet *dutyP2 = new FuzzySet(1, 2, 2, 3);
duty->addFuzzySet(dutyP2);
FuzzySet *dutyP3 = new FuzzySet(2, 3, 3, 4);
duty->addFuzzySet(dutyP3);
fuzzy->addFuzzyOutput(duty);

```

Por último, se realiza las reglas de control **Figura**, donde observamos que cada entrada está asociada con una salida en específica, estas reglas permiten al controlador difuso generar una salida que sería la que vamos aplicar al DAC.

Figura. 47 Reglas de control

```
//Reglas de control
FuzzyRuleAntecedent *ifvoltajeB3 = new FuzzyRuleAntecedent();
ifvoltajeB3->joinSingle(voltajeB3);
FuzzyRuleConsequent *tamdutyN3 = new FuzzyRuleConsequent();
tamdutyN3->addOutput(dutyN3);
FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifvoltajeB3, tamdutyN3);
fuzzy->addFuzzyRule(fuzzyRule01);

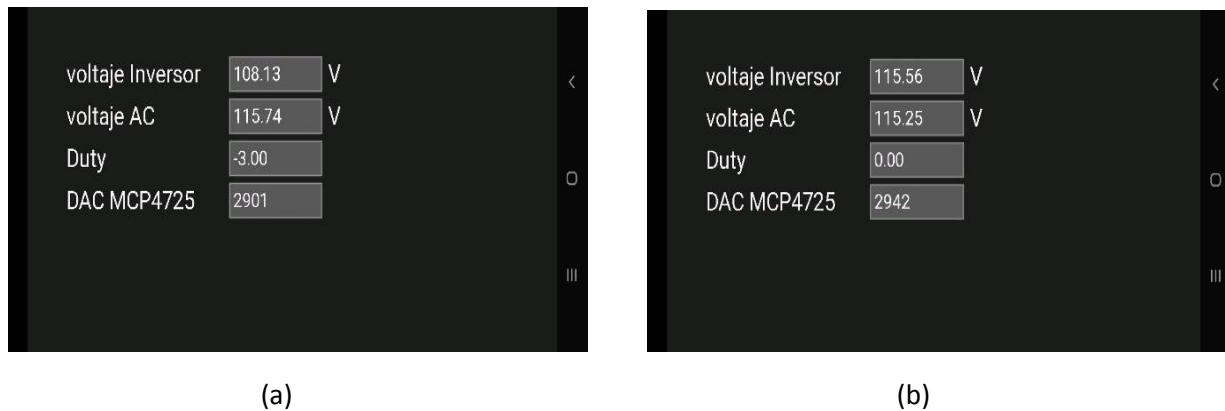
FuzzyRuleAntecedent *ifvoltajeB2 = new FuzzyRuleAntecedent();
ifvoltajeB2->joinSingle(voltajeB2);
FuzzyRuleConsequent *tamdutyN2 = new FuzzyRuleConsequent();
tamdutyN2->addOutput(dutyN2);
FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifvoltajeB2, tamdutyN2);
fuzzy->addFuzzyRule(fuzzyRule02);

FuzzyRuleAntecedent *ifvoltajeB1 = new FuzzyRuleAntecedent();
ifvoltajeB1->joinSingle(voltajeB1);
FuzzyRuleConsequent *tamdutyN1 = new FuzzyRuleConsequent();
tamdutyN1->addOutput(dutyN1);
FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifvoltajeB1, tamdutyN1);
fuzzy->addFuzzyRule(fuzzyRule03);

FuzzyRuleAntecedent *ifvoltaje0 = new FuzzyRuleAntecedent();
ifvoltaje0->joinSingle(voltaje0);
FuzzyRuleConsequent *tamduty0 = new FuzzyRuleConsequent();
tamduty0->addOutput(duty0);
FuzzyRule *fuzzyRule04 = new FuzzyRule(4, ifvoltaje0, tamduty0);
fuzzy->addFuzzyRule(fuzzyRule04);
```

En el momento de encender el prototipo observamos que el voltaje de la etapa inversora inicia con un voltaje por debajo de la red eléctrica (**Figura 43.a**), en ese momento vemos que el duty tiene un valor de -3, ese valor se le resta al valor general del **DAC MCP4725**, en el momento que se iguala al voltaje de la red eléctrica, vemos que el valor del **DAC MCP472** permanece fijo (**Figura 43.b**) y el valor de duty se queda en cero manteniendo el valor del **DAC MCP472** estable.

Figura. 48 Sincronización con el DAC MCP4725.



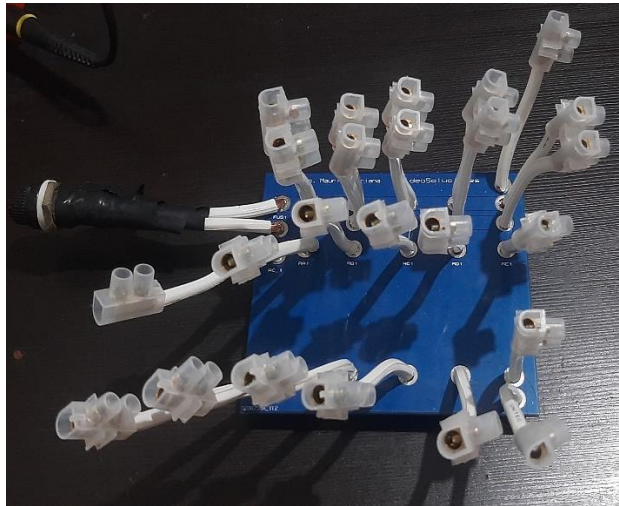
5.5 Efecto y corrección de la carga sobre la etapa inversora

Uno de los problemas que tiene los inversores, es que, en el momento de conectar la carga, se genera un bajonazo de la amplitud del mismo, si el prototipo funcionara simplemente como inversor, este efecto se corrige solamente aumentando el voltaje perdido. Para esto se puede utilizar cualquier método de control, sin embargo, la función de un D-STATCOM es sincronizarse con la red eléctrica justo en el momento que hay carga conectada, por lo tanto para prevenir efectos no deseados en el momento de la sincronización se tomara registros de diferentes tipos de cargas y ver que tanto se debe corregir por cada variación, la idea es que al monitorear el voltaje, la corriente y la potencia que está suministrando la red eléctrica a la carga, se aumente una cantidad estimada del voltaje de la etapa inversora por encima de la red eléctrica con el fin de no poner en riesgo el prototipo y compensar posibles bajonazos que tendría la etapa inversora en una sincronización directa con la carga.

Para poder tomar registro del comportamiento de la etapa inversora bajo diferentes tipos de carga, se diseñó una tarjeta (**Figura 44**) en la que se pueden conectar cargas resistivas de potencia, cargas inductivas y capacitivas.

Figura. 49 Simulador de cargas.

(a)



(b)

La estrategia consiste en conectar la mayor cantidad de configuraciones de carga posible y tomar registro del efecto que ocurre en la etapa inversora, después usar como referencia el voltaje de la red y corregir la caída de voltaje del inversor igualándola a la de la red eléctrica, en ese momento se toma registro del voltaje de la red, el voltaje del inversor, la corriente, la potencia activa **P**, la potencia reactiva **Q**, el factor de potencia **FP** y el valor del **DAC** registrado (**Tabla 1**).

Tabla 1

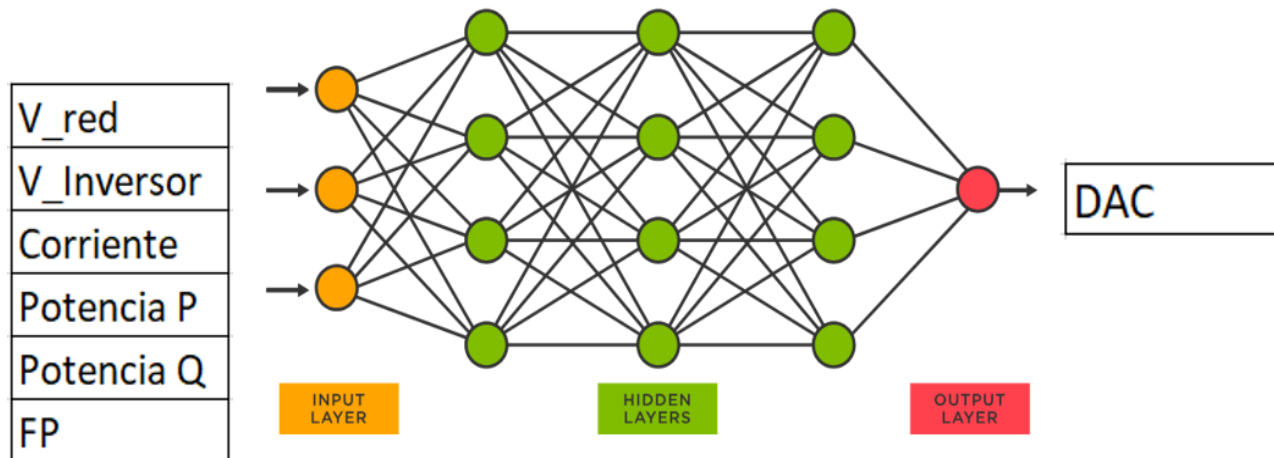
Valores registrados de diferentes cargas ejemplo

Datos	Carga 1	Carga 2	Carga 3	Carga 4	Carga 5	Carga 6	Carga 7	Carga 8	Carga 9	Carga 10
V_red	117,87	117,64	118,16	117,39	116,61	117,11	117,87	116,91	117,23	117,13
V_Inversor	117,66	117,16	117,79	116,92	116,13	116,9	117,48	116,49	116,78	116,67
Corriente	0,02	0,17	0,06	0,3	0,06	0,06	0,17	0,02	0,23	0,24
Potencia P	1,9	20	3,9	34,8	3,7	3,8	20	1,9	26,6	27,8
Potencia Q	2,1	2,84	6,29	0	5,92	6,35	2,84	2,05	0	0
FP	0,67	0,99	0,53	1	0,53	0,51	0,99	0,68	1	1
V_DC	11,37	12,24	11,32	12	11,27	11,27	12,2	11,07	11,95	11,61
DAC	2930	2709	2915	2411	2926	2921	2709	2942	2585	2558

La estrategia de control que se desea aplicar consiste en usar redes neuronales, el principio fundamental de la red neuronal (**Figura 45**) consiste en entrenarla con los datos registrados en la **Tabla 1**, para eso se

usan como entradas los valores del voltaje de red, el voltaje de la etapa inversora, la corriente, la potencia activa **P**, la potencia reactiva **Q**, el factor de potencia **FP**, y como dato de salida el valor del **DAC** que se registró cuando se igualaron los voltajes de red con la etapa inversora, esto se repite con todas las combinaciones de carga.

Figura. 50 Carga de datos en la Red Neuronal.



Una vez entrenada la red neuronal, se toman todas las neuronas, pesos y tipos de activación para calcular el valor de salida que en nuestro caso sería el valor del convertidor digital-analógico (DAC), con estos valores podemos estimar un voltaje de compensación de la etapa inversora antes de la sincronización y evitar comportamientos no deseados del prototipo.

5.6 Código y resultados del entrenamiento de la Red Neuronal para la sincronización.

Para el desarrollo de la Red Neuronal se utilizó la herramienta Jupiter (**Figura 46.a**) que viene en la interfaz de Anaconda Navigator (**Figura 46.b**), esta herramienta nos permite trabajar con librerías geoespaciales, además es gratuito y de código abierto compatible con diferentes sistemas operativos.

Figura. 51 Entorno de Trabajo



Lo primero que se realizó fue la importación de las librerías necesarias (**Figura 47**), entre estas tenemos la librería de TensorFlow que es una librería muy potente y de código libre para Machine Learning (ML), esta librería le pertenece a la empresa Google y fue liberada para uso libre en el 2015. Otra librería utilizada fue la de pandas, esta librería nos permite manipular y analizar de forma flexible los datos en el lenguaje Python, y por último la librería matplotlib que nos permite trabajar con diferentes tipos de grafica para la visualización de los resultados.

Figura. 52 Librerías en Python para el análisis de datos

```
import tensorflow as tf
import pandas as pd
import matplotlib.pyplot as plt
```

Después de importar las librerías necesarias para el análisis de datos creamos una variable llamada archivo (**Figura 48**), en esta variable se guardan los datos previamente tomados de una serie de ensayos con diferentes tipos de configuraciones de carga, la variable creada df con ayuda de la librería pandas nos muestra los datos que se encuentran guardados, como se puede observar tenemos 10 configuraciones distintas en las que tenemos el voltaje de red, voltaje del inversor, corriente, Potencia activa (P), Potencia reactiva (Q), Factor de Potencia (FP) y DAC. Estos datos son los mismos obtenidos con el simulador de carga (**Figura 44**) y se encuentran en la **Tabla 1**.

La estrategia que se utilizó fue conectar diferentes configuraciones de carga al inversor, estas configuraciones provocan un bajonazo en el voltaje del mismo, con la ayuda del DAC modificamos el ciclo de trabajo del TL-494 (**Figura 39**) que va al convertidor DC-DC modificando la amplitud del voltaje del inversor corrigiendo la caída, para eso se usó de referencia el voltaje de la red registrando los valores que vemos en la tabla, de esa forma cuando monitoreemos los valores del voltaje de red, corriente, Potencia activa (P), Potencia reactiva (Q), Factor de Potencia (FP) de una carga conectada a la red podríamos ajustar el DAC y generar un voltaje del inversor más alto para compensar las caídas que tendría en el momento de la sincronización.

Como podemos observar, tenemos 7 datos de 10 configuraciones diferentes lo cual usaremos las 6 primeras columnas como datos de entrada y la columna 7 como salida de la red neuronal, para eso se creó la variable `arreglox` como entrada y la variable `arregloy` como salida (**Figura 48**).

Figura. 53 Arreglo de datos para el procesamiento

```
archivo = "DatosPrueba2.csv"
df = pd.read_csv(archivo)
```

df

	V_red	V_Inversor	Corriente	Potencia P	Potencia Q	FP	DAC
0	117.87	117.66	0.02	1.9	2.10	0.67	2930
1	117.64	117.16	0.17	20.0	2.84	0.99	2709
2	118.16	117.79	0.06	3.9	6.29	0.53	2915
3	117.39	116.92	0.30	34.8	0.00	1.00	2411
4	116.61	116.13	0.06	3.7	5.92	0.53	2926
5	117.11	116.90	0.06	3.8	6.35	0.51	2921
6	117.87	117.48	0.17	20.0	2.84	0.99	2709
7	116.91	116.49	0.02	1.9	2.05	0.68	2942
8	117.23	116.78	0.23	26.6	0.00	1.00	2585
9	117.13	116.67	0.24	27.8	0.00	1.00	2558

```
arreglox = df[df.columns[:-1]].to_numpy()
arregloy = df[df.columns[-1]].to_numpy()
```

Después de tener las entradas y salidas definidas se construyó la red neuronal (**Figura 49**) con 6 entradas (`input_shape`), 2 capas ocultas (oculta 1, oculta 2) de 4 y 2 neuronas, la activación de tipo `relu`, el optimizador `Adam`, y con unas 15000 épocas de iteración.

Figura. 54 Construcción de la Red Neuronal

```

oculta1 = tf.keras.layers.Dense(units=4, activation=tf.nn.relu, input_shape=[6])
oculta2 = tf.keras.layers.Dense(units=2, activation=tf.nn.relu)
salida = tf.keras.layers.Dense(units=1)
modelo = tf.keras.Sequential([oculta1, oculta2, salida])

```

```

modelo.compile(
    optimizer=tf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)

```

```

print("Entrenando...")
historial = modelo.fit(arreglox, arregloy, epochs=15000, verbose=False)
print("Modelo entrenado!")

```

```

Entrenando...
Modelo entrenado!

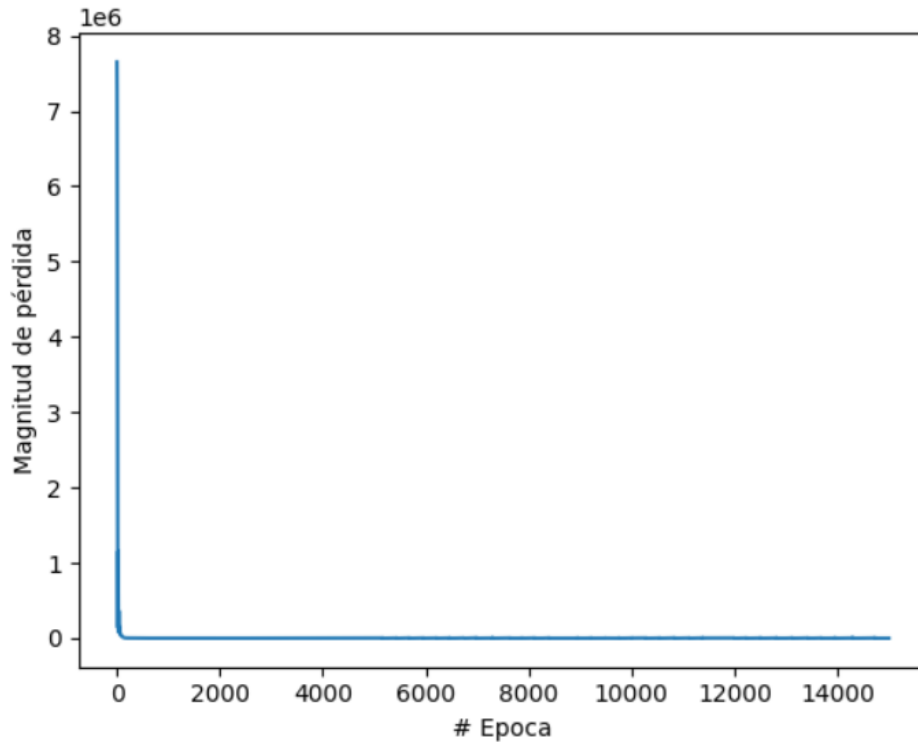
```

Una forma de obtener resultados bastantes precisos es adquirir una gran cantidad de datos de entrenamiento, pero el objetivo de esta estrategia es tener un valor que nos permita amortiguar posibles bajonazos del voltaje del inversor para mitigar efectos no deseados, no obstante, se debe realizar las correcciones necesarias después de la sincronización. En la gráfica mostrada se observa que con menos épocas se podría tener un error muy pequeño, pero se debió realizar el entrenamiento varias veces hasta obtener el resultado deseado y se decidió dejar los valores obtenidos con dichas iteraciones.

Figura. 55 Magnitud de perdidas en el entrenamiento

```
plt.xlabel("# Epoca")
plt.ylabel("Magnitud de pérdida")
plt.plot(historial.history["loss"])
```

[<matplotlib.lines.Line2D at 0x264c7b82650>]



Después de haberse realizado el entrenamiento se ingresaron unos valores de prueba para ver el cálculo estimado, observemos que se ingresó un valor de 115.44 que representa el voltaje medido de la red eléctrica, los valores de 0.02, 1.8, 1.94, 0.68 representan la medición de la corriente, la potencia activa, la potencia reactiva y el factor de potencia de una carga conectada en ese instante, con el control difuso se calibro el voltaje del inversor en un valor de 115.04 que sería más del 99% de similitud con el voltaje de referencia que sería la red eléctrica. Estos 6 valores se ingresan en la red neuronal previamente entrenada dándonos un valor de 2897.062, este valor se redondea y se ingresa al DAC dándonos un aumento estimado del voltaje inicialmente calibrado de 115.04 con el fin de compensar las caídas de tensión al momento de sincronizar.

Figura. 56 Valores calculados de los pesos

```

print("Hagamos una predicción!")
resultado = modelo.predict([[115.44, 115.04, 0.02, 1.8, 1.94, 0.68]])
resultado

Hagamos una predicción!
1/1 [=====] - 0s 128ms/step

array([[2897.062]], dtype=float32)

print("Variables capa 1 internas del modelo")
print(oculta1.get_weights())

Variables capa 1 internas del modelo
[array([[ 0.11443919, -1.2250757, -1.1219732,  1.5696936 ],
        [-0.5673916,  0.04570427, -0.03472929,  1.9919457 ],
        [-0.62376034, -0.53542125, -0.48469353, 10.798247 ],
        [ 0.05371702, -0.76149195, -0.641458 , -2.652646 ],
        [ 0.06428134, -0.7908198 , -1.1169902 ,  1.3960829 ],
        [ 0.7065898 , -0.8142374 , -0.52704865, 29.526333 ]],
        dtype=float32), array([ 0.          , -0.6005466 , -0.60054666,  8.310956 ], dtype=float32)]

print("Variables capa 2 internas del modelo")
print(oculta2.get_weights())

Variables capa 2 internas del modelo
[array([[ 0.46993327, -0.18515992],
        [-0.21944973, -0.81784177],
        [ 0.90769714,  0.3526613 ],
        [ 2.3445654 , -0.64331275]], dtype=float32), array([ 9.212305 , -1.1948252], dtype=float32)]

print("Variables salida internas del modelo")
print(salida.get_weights())

Variables salida internas del modelo
[array([[2.793823 ],
        [0.94930124]], dtype=float32), array([9.515261], dtype=float32)]

```

5.7 Conversión de los datos obtenidos en la Red Neuronal al lenguaje Arduino.

Después de haber entrenado la red neuronal en una PC con buen rendimiento, se tomaron los valores para implementarlo en el Arduino, para eso se creó 6 variables de entrada (p1, p2, p3, p4, p5, p6), estas variables representan en su orden el voltaje de la red eléctrica, el voltaje del inversor previamente igualada con la red, la corriente, la potencia activa, la potencia reactiva y el factor de potencia, estos valores son obtenidos con el módulo PZEM-004T (**Figura 17**), una vez obtenidos estos valores y aplicando los pesos y la función de activación obtenemos un nuevo valor x que sería el valor del DAC calculado para tener un voltaje de compensación antes de la sincronización.

Figura. 57 Implementación de la red neuronal en Arduino lgt8f328p

```

float p1, p2, p3, p4, p5, p6, n1, n2, n3, n4, n5, r1, r2, r3, r4, r5;
float voltage = pzem.voltage();
float current = pzem.current();
float power = pzem.power();
float frequency = pzem.frequency();
float pf = pzem.pf();
float qreactiva = sin((acos(pf)))*(voltage)*(current);

p1 = Vref;
p2 = voltage;
p3 = current;
p4 = power;
p5 = qreactiva;
p6 = pf;
n1 = p1*(1.9409899)+p2*(1.439289)+p3*(0.19086878)+p4*(-1.3682039)+p5*(1.712795)+p6*(0.52335304)+(1.1498305);
n2 = p1*(0.5543062)+p2*(-0.53611374)+p3*(-1.0040519)+p4*(-0.9338232)+p5*(-0.15242693)+p6*(-0.61381304)+(-0.2663546);
if (n1>=0) r1 = n1; else r1 = 0;
if (n2>=0) r2 = n2; else r2 = 0;
n3 = r1*(1.4657729)+r2*(0.9387726)+(0.9860326);
n4 = r1*(2.1178136)+r2*(-0.9397003)+(0.9193358);
if (n3>=0) r3 = n3; else r3 = 0;
if (n4>=0) r4 = n4; else r4 = 0;
n5 = r3*(1.8075663)+r4*(2.1481438)+(0.74887025);
if (n5>=0) r5 = n5; else r5 = 2700;
x = r5;
dac.setVoltage(x, false);

```

Como podemos observar en el rectángulo rojo (**Figura 53**) el valor del voltaje del inversor es menor al voltaje de la red eléctrica, al mismo tiempo observamos en el rectángulo azul los valores de corriente, potencia activa, potencia reactiva, y factor de potencia que está suministrando la red a una carga indefinida, una vez censado el voltaje de red, el Arduino denominado master comienza aplicando el control difuso mencionado en el **capítulo 5.4** modificando el valor del DAC mostrado en el rectángulo verde.

Figura. 58 Control difuso en el voltaje del inversor

En el momento de igualar el voltaje del inversor con el voltaje de la red eléctrica (**Figura 54.a**) se va a la red neuronal previamente establecida (**Figura 52**) ingresando los valores de entrada, estos valores generan un nuevo voltaje del inversor como compensación a los posibles bajonazos que tendría si se conectara solamente a la carga (**Figura 54.b**).

Figura. 59 Control difuso en el voltaje del inversor

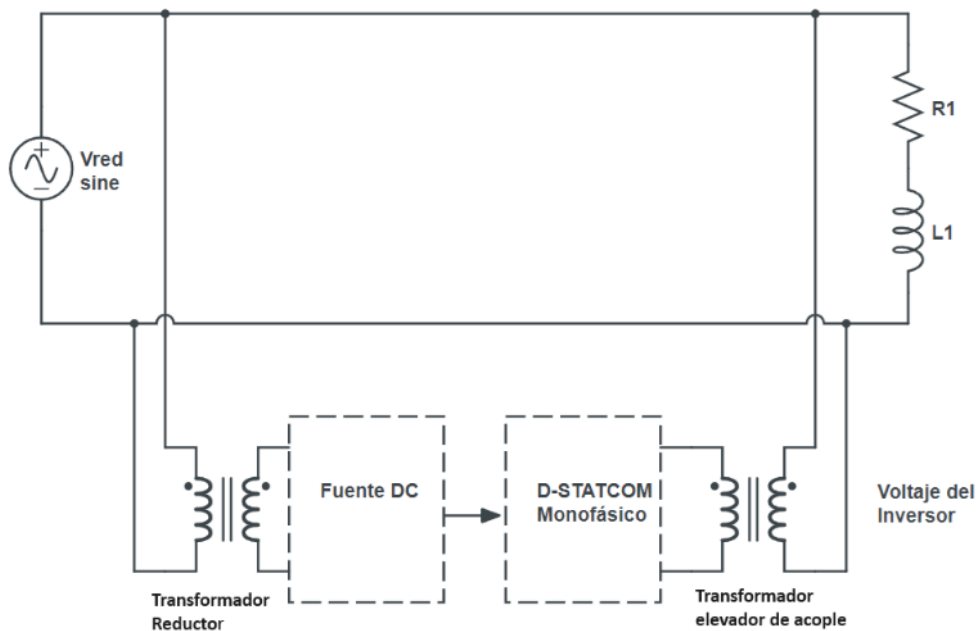


5.8 Sincronización del prototipo D-STATCOM con la red eléctrica.

En este capítulo se realizará un análisis de las diferentes conexiones y comportamiento del prototipo en sincronización con la red, sabiendo que el control del prototipo funciona correctamente se verificara su eficacia en la sincronización a plena carga y si su comportamiento es el mencionado por la teoría del D-STATCOM, no obstante, este proyecto se enfoca en las estrategias de control.

La configuración optada está compuesta por una fuente DC (**Figura 55**), esta fuente se toma directamente de la red que alimenta la carga a través del transformador reductor, después se hace pasar por el prototipo D-STATCOM, este se encarga de generar la parte inversora y del control necesario previo a su acople con el sistema. La teoría del D-STATCOM dice que la parte inversora del prototipo se sincroniza con la red eléctrica, si el factor de potencia es bajo, el voltaje del D-STATCOM aumenta comportándose como un capacitor, y si el factor de potencia es al alto, el voltaje del D-STATCOM disminuye comportándose como un inductor.

Figura. 60 Configuración de acople.



Para disminuir los riesgos de origen eléctrico se utilizó fusibles eléctricos tanto en la alimentación del D-STATCOM como en la carga conectada, también se agregó un relé de estado sólido de corriente alterna que permite la conexión de sincronización y la desconexión en caso de sobre corrientes, esto se hace para tener mecanismos de seguridad más robustos.

Figura. 61 Relé de estado sólido para la sincronización en AC.



5.8.1 Sincronización del prototipo D-STATCOM con la red eléctrica a plena carga igualando voltaje RMS.

En esta primera prueba se hará la sincronización del prototipo solamente igualando la frecuencia y el voltaje RMS del inversor, no obstante, y como se mencionó en el **capítulo 5.5** se había creado una red neuronal para el cálculo de un voltaje RMS que permita la compensación de cualquier bajonazo que genere la conexión en el prototipo a la red eléctrica, sin embargo y por fines investigativos se optó hacer la primera conexión sin esta compensación para tener valores de referencia y comprobar los efectos generados en su conexión.

Podemos observar en el cuadro rojo (**Figura 56.a**) que inicialmente el control del prototipo D-STATCOM cumple con su función sincronizando el voltaje de la etapa inversora con el voltaje de la red eléctrica, en el cuadro azul observamos los valores de la corriente, potencia activa, potencia reactiva y factor de potencia antes de la sincronización, una vez sincronizado el D-STATCOM observamos en el cuadro azul (**Figura 56.b**) que la sincronización está perfectamente equilibrada ya que no hay efectos abruptos en los parámetros como la corriente, potencia activa, potencia reactiva y factor de potencia.

Figura. 62 Sincronización igualando voltaje RMS

5.8.2 Sincronización del prototipo D-STATCOM con la red eléctrica a plena carga aplicando voltaje RMS de compensación.

En el **capítulo 5.5** se explicó el efecto que tiene un inversor al conectar una carga, donde se obtiene un bajonazo en el voltaje RMS del mismo. Para corregir este efecto solamente había que aumentar el voltaje RMS, pero como se mencionó en dicho capítulo, el D-STATCOM debe sincronizar su parte inversora con la red eléctrica conectada a plena carga y prevenir cualquier efecto nocivo, para eso se entrenó y provo una red neuronal que nos calcula un voltaje RMS de compensación que previene cualquier bajonazo si esa misma carga se conectara solamente a la etapa inversora del D-STATCOM, no obstante y como se vio en el capítulo 5.8.1 no hubo ningún efecto nocivo conectando el D-STATCOM simplemente igualando los voltajes RMS, sin embargo se verificara el efecto en la sincronización con el voltaje RMS de compensación.

Podemos observar en el cuadro rojo (**Figura 57.a**) que inicialmente el control del prototipo D-STATCOM cumple con su función sincronizando el voltaje de la etapa inversora con el voltaje de la red eléctrica, en el cuadro azul observamos los valores de la corriente, potencia activa, potencia reactiva y factor de potencia antes de la sincronización, una vez sincronizado el D-STATCOM observamos que el voltaje RMS aumento a 132.66V en el cuadro rojo (**Figura 57.b**), en el cuadro azul observamos que la sincronización está perfectamente equilibrada pero en este caso a diferente del capítulo anterior hay una pequeña ganancia en la potencia activa, potencia reactiva y corriente, sin embargo mejora un poco el factor de potencia. Otro dato que podemos observar es que antes de la sincronización del D-STATCOM el voltaje de la etapa inversora era de 132.66V, una vez sincronizados el voltaje que se mantiene es el de la red eléctrica.

Figura. 63 Sincronización con voltaje RMS de compensación

5.8.3 Sincronización del prototipo D-STATCOM con la red eléctrica a plena carga aplicando voltaje RMS menor al de la red eléctrica.

En los capítulos anteriores se observó que la sincronización está funcionando mejor de lo que se esperaba, ya que no hay cambios abruptos ni variaciones en los datos monitoreados, por eso se decidió observar el comportamiento en la sincronización con un voltaje RMS más bajo que el de la red eléctrica, ya que una de las funcionalidades del D-STATCOM es disminuir su voltaje en caso de haber un factor de potencia en adelanto comportándose como un inductor.

Observamos en el cuadro rojo (**Figura 58.a**) que el voltaje de la etapa inversora del D-STATCOM está por debajo al de la red eléctrica antes de la sincronización, en el momento que se sincronizan (**Figura 58.b**) hay una pequeña ganancia de corriente, potencia activa y potencia reactiva, pero el factor de potencia aparentemente obtiene una disminución en atraso como si se hubiera aumentado la potencia reactiva, lo que en teoría estaría bien.

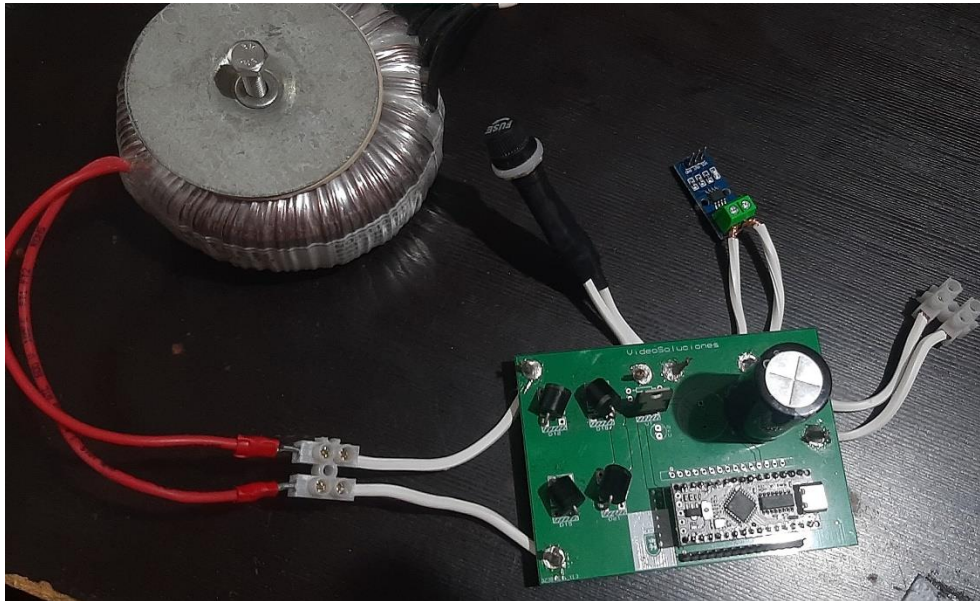
Figura. 64 Sincronización con voltaje RMS más bajo

5.9 Control del FP de potencia en la etapa de sincronización.

Hasta el momento se observó que el control en la sincronización del prototipo con la red eléctrica conectada a plena carga funciona de manera correcta, ya que no se dañó ningún fusible que se tenía de seguridad, y tampoco la programación de control desconecta el relé de estado sólido usado para la conexión.

En este capítulo se realizará el control del factor de potencia, sabiendo que la teoría indica que si el factor de potencia medido está en atraso, el prototipo D-STATCOM debe aumentar su voltaje RMS por encima de la red eléctrica comportándose como capacitor variable, caso contrario, si el factor de potencia está en adelanto, el prototipo D-STATCOM debe disminuir su voltaje RMS por debajo de la red eléctrica comportándose como un inductor, no obstante, ya que el D-STATCOM es un prototipo creado para aplicar control se decidió poner un factor de potencia deseado de 0.9 para la prueba controlada, esto significa que el control se aplicara a un factor de potencia en atraso.

Figura. 65 Fuente DC extraída de la red eléctrica para el D-STATCOM.



En la **Figura 65** se observa la conexión que se realizará, donde observamos que de la misma red se obtiene el voltaje de alimentación del D-STATCOM, esta tensión se disminuye con un transformador donde después se pasa por un puente de diodos que nos permite linealizar el voltaje obteniendo una fuente DC, de esta fuente se alimentara los circuitos encargados del control y el puente H que posteriormente se pasara por la etapa inversora. En la **Figura 56** se observa la conexión física del transformador y del circuito que la convierte en voltaje DC, el tal circuito se agregó un Arduino y un medidor de corriente de efecto hall para monitorear la corriente suministrada, esto con el fin de robustecer la seguridad, ya que se puede agregar un relé de estado sólido **Figura 62** en la fuente DC que nos permita realizar una desconexión en caso de tener una sobre corriente, no obstante y como se mencionó en anteriores capítulos, hay diferentes mecanismos de seguridad, entre estas los fusibles.

Figura. 66 Relé de estado sólido para el control de la fuente DC.



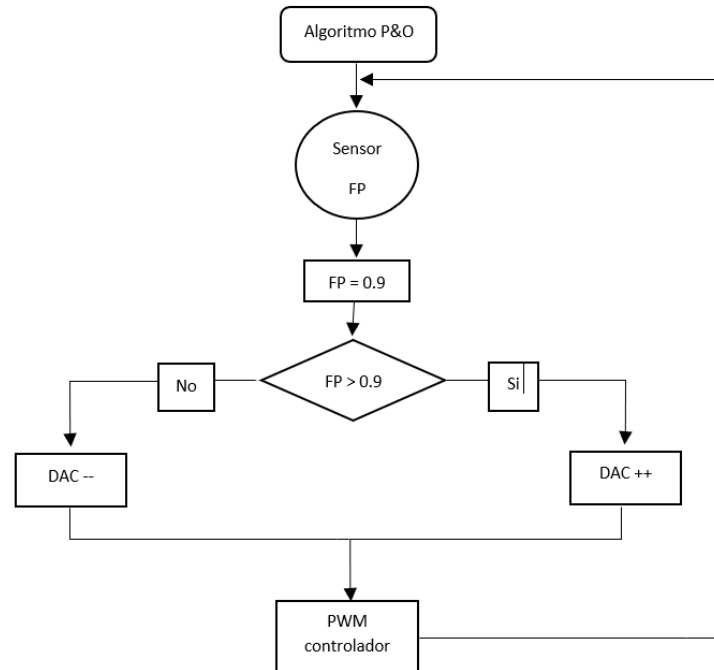
5.9.1 Control P&O del FP de potencia en la etapa de sincronización.

En este capítulo se realizará el control final de este proyecto donde se tendrá en cuenta el estado de sincronización. En los capítulos anteriores realizamos la sincronización de la frecuencia y el voltaje del prototipo D-STATCOM antes de la sincronización, posteriormente se registró el comportamiento en el momento de la conexión a la red eléctrica conectada a plena carga, estos controles funcionaron de forma óptima ya que no reaccionaron de forma negativa manteniendo una buena estabilidad. Ya para finalizar hay que recordar que el funcionamiento del prototipo D-STATCOM depende del monitoreo del factor de potencia, ya que, si el factor de potencia está en atraso el voltaje de la parte inversora deberá aumentar por encima de la red eléctrica inyectando corriente en adelante comportándose como un capacitor variable, pero si está en adelanto deberá bajar su voltaje por debajo de la red eléctrica comportándose como un inductor.

El control que se realizara es el algoritmo MPPT de perturbación y Observación (P&O), este algoritmo es muy usado con turbinas eólicas y paneles solares para maximizar la extracción de energía en todas las condiciones, se decidió usar por su simplicidad y poder utilizar una nueva técnica de control con fin académico.

Como observamos en la **Figura 67** el algoritmo es simple, este se encarga de monitorear el factor de potencia en el instante, si el factor de potencia está por debajo del deseado en nuestro caso de prueba 0.9, el DAC disminuye 1 paso generando un aumento del ciclo de trabajo en el convertidor DC-DC elevador, caso contrario, si el valor está por encima del deseado, el DAC aumenta 1 paso disminuyendo el ciclo de trabajo del convertidor DC-DC, este control deberá verse reflejado al instante del monitoreo. Este control se decidió aplicar debido a que la velocidad del Arduino puede ser mayor que aplicar un control como el difuso, además que debe estar monitoreando en tiempo real todo el tiempo.

Figura. 67 Algoritmo P&O



Observamos que en la **Figura 68.a** antes de la sincronización con la red eléctrica y la carga, utilizaremos la red neuronal que nos da un voltaje de compensación, para este ejemplo el voltaje calculado fue de **163.45V_{rms}**. En el momento de la sincronización ambas fuentes de voltaje quedan iguales, eso se debe a que el voltaje proporcionado por la red tiene mayor potencia obligando a la etapa inversora a igualar el voltaje, en ese preciso instante empieza a trabajar el algoritmo P&O modificando el valor del DAC que a su vez modifica el porcentaje del duty de trabajo del convertidor DC-DC aumentando el voltaje RMS por encima del voltaje de la red, y como se mencionó antes el prototipo D-STATCOM esta es inyectando corriente en adelanto corrigiendo el factor de potencia, observemos en la **Figura 68.b** que el algoritmo mantiene el factor de potencia previamente establecido entre 0.90 y 0.93, aumentando o disminuyendo los pasos del DAC conforme se va monitoreando.

Figura. 68 Control del Factor de Potencia mediante el control P&O



Como pudimos observar en la **Figura 68**, el prototipo está funcionando correctamente como un capacitor variable, no obstante se observa que al modificar el factor de potencia aumenta el suministro de potencia activa, reactiva y corriente, esto se debe a que el transformador utilizado tienen una cierta cantidad de potencia, si se aumenta la capacidad del transformador se obtendría los mismos resultados del factor de potencia con menos aumento de la potencia activa, reactiva y corriente, pero al ser un prototipo académico confirma el buen diseño y comportamiento del proyecto junto con su control.

Conclusiones

- La estrategia de sincronización utilizada funciona de forma eficiente ya que no solo se sincronizo con el voltaje de compensación para prevenir comportamiento no deseado, sino que también se hicieron pruebas utilizando el mismo voltaje obteniendo casi el mismo resultado sin efecto negativo.
- El diseño de la detección de la secuencia de fase permite crear diferentes estrategias de control a parte de la proporcional. Debido a que su simplicidad permite utilizar otras estrategias de control como la lógica difusa o las redes neuronales en la sincronización de la frecuencia.
- El prototipo D-STATCOM también puede funcionar como laboratorio de potencia reactiva, ya que su diseño permite conexiones tanto en serie como derivativas, estas conexiones sirven para profundizar en el estudio de técnicas de control y en el comportamiento de dichas conexiones, además que su diseño también permite realizar un diseño trifásico aumentando las prácticas de estudio.
- El uso del Módulo Bluetooth permite un mejor monitoreo y ajuste del funcionamiento de la sincronización, ya que se convierte en una herramienta que facilita monitorear diferentes variables de interés.
- El control de las diferentes etapas funciona de forma correcta, pero el comportamiento del D-STATCOM no esta funcionando como se esperaba ya que teóricamente al mejorar el factor de potencia debería haber una disminución de la corriente total suministrada por la red eléctrica, pero está pasando todo lo contrario, no solo aumenta la corriente, sino que aumenta el consumo de la potencia activa como reactiva, esto puede ser ocasionado por diferentes factores que se pueden estudiar con más detalle en futuros trabajos de investigación, ya que este proyecto esta enfocado en las estrategias de control.

Referencias

- [1] B. D. Parameshachari, B. Reshma, GSSS Institute of Engineering and Technology for Women, Institute of Electrical and Electronics Engineers. Bangalore Section, and Institute of Electrical and Electronics Engineers, *International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques : ICEECOT-2017 : 15-16 December 2017*.
- [2] W.-N. Chang and K.-D. Yeh, "Design and Implementation of DSTATCOM with Symmetrical Components Method for Fast Load Compensation of Unbalanced Distribution Systems."
- [3] J. G. Agrawal and K. D. Joshi, "Experimental study of some variable impedance type FACTS devices," in *International Conference on Emerging Trends in Engineering and Technology, ICETET*, 2011, pp. 164–167. doi: 10.1109/ICETET.2011.31.
- [4] A. Ghosh and G. Ledwich, "Load compensating DSTATCOM in weak AC systems," *IEEE Transactions on Power Delivery*, vol. 18, no. 4, pp. 1302–1309, Oct. 2003, doi: 10.1109/TPWRD.2003.817743.
- [5] E. Ghahremani and I. Kamwa, "Optimal placement of multiple-type FACTS devices to maximize power system loadability using a generic graphical user interface," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 764–778, 2013, doi: 10.1109/TPWRS.2012.2210253.
- [6] "Circuitos eléctricos."
- [7] "Fundamentos de Circuitos Eléctricos," 2013.
- [8] Or.) IEEE Power & Energy Society. General Meeting (2018 : Portland and Institute of Electrical and Electronics Engineers, *2018 IEEE Power & Energy Society General Meeting (PESGM)*.
- [9] *2013 International Conference on New Concepts in Smart Cities : Fostering Public and Private Alliances (SmartMILE)*. IEEE, 2013.
- [10] O. Gomis-Bellmunt, J. Sau-Bassols, E. Prieto-Araujo, and M. Cheah-Mane, "Flexible Converters for Meshed HVDC Grids: From Flexible AC Transmission Systems (FACTS) to Flexible DC Grids," *IEEE Transactions on Power Delivery*, vol. 35, no. 1, pp. 2–15, Feb. 2020, doi: 10.1109/TPWRD.2019.2939588.
- [11] Adhiparasakthi Engineering College. Department of Electrical and Electronics Engineering, Adhiparasakthi Engineering College, and Institute of Electrical and Electronics Engineers, *2016 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*.
- [12] J. Alnasseir, R. Alcharea, and F. Almaghout, "Improving the Stability of Smart Grids by Using Flexible AlternatingCurrent Transmission Systems (FACTS)," in *2021 12th International Renewable Engineering Conference, IREC 2021*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021. doi: 10.1109/IREC51415.2021.9427827.
- [13] *2010 7th International Multi-Conference on Systems, Signals, and Devices : June 27th to 29th, 2010, Philadelphia University, Amman, Jordan*.

- [14] F. Ali El-Sheikhi, Y. Mabruk Saad, S. Omar Osman, and K. M. El-Arroudi, "Voltage Stability Assessment Using R4odal Analysis of Power Systems including Flexible A C Transmission System (FACTS) Abstr-ucr."
- [15] A. Kechroud, J. M. A. Myrzik, and W. Kling, "TAKING THE EXPERIENCE FROM FLEXIBLE AC TRANSMISSION SYSTEMS TO FLEXIBLE AC DISTRIBUTION SYSTEMS."
- [16] I. N. Syamsiana, R. A. Susilo, and F. Ronilaya, "Design Synergetic Control for Electric Power System with Static VAR Compensator," in *4th International Conference on Vocational Education and Training, ICOVET 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 228–234. doi: 10.1109/ICOVET50258.2020.9230086.
- [17] Y. Satake, T. Tanaka, A. Tokiwa, M. Fukuda, and H. Yamada, "New Control Strategy for Hybrid Static Var Compensator with Series Active Filters."
- [18] K. Ramadhan, K. Yonathan, I. M. Ardita, F. H. Jufri, and A. R. Utomo, "Voltage stability improvement using load shedding and static VAR compensator (SVC): Study case of senayan-sambas power system," in *2nd IEEE International Conference on Innovative Research and Development, ICIRD 2019*, Institute of Electrical and Electronics Engineers Inc., Jun. 2019. doi: 10.1109/ICIRD47319.2019.9074752.
- [19] W. Krysthre, A. Gonqalves, J. Carlos De Oliveira, V. Lucia, and S. Franco, "Advanced Static VAR Compensator Performance under Non-Ideal Electric Power Supply Conditions AV. JOHO Naves de Avila, 2 160-U b e r h 'a-MG-Brazil."
- [20] IEEE International Conference on Mechatronics and Automation 12 2015.08.02-05 Beijing and IEEE ICMA Conference 12 2015.08.02-05 Beijing, *2015 IEEE International Conference on Mechatronics and Automation (ICMA) 2 - 5 Aug. 2015, Beijing, China*. IEEE, 2015.
- [21] B. Xu, Institute of Electrical and Electronics Engineers. Beijing Section, and Institute of Electrical and Electronics Engineers, *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC 2016) : October 3-5, 2016, Xi'an China*.
- [22] E. and C. S. 2016 B. IEEE Students' Conference on Electrical, S. Saxena, Institute of Electrical and Electronics Engineers, E. and C. S. 2016. 03. 05-06 B. IEEE Student Conference on Electrical, E. and C. S. 2016. 03. 05-06 B. IEEE Students' Conference on Electrical, and SCEECs 2016.03.05-06 Bhopal, *2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECs) March 5-6, 2016, Maulana Azad National Institute of Technology, Bhopal*.
- [23] N. Johansson, L. Ängquist, and H. P. Nee, "Erratum: An adaptive controller for power system stability improvement and power flow control by means of a thyristor switched series capacitor (TSSC) (IEEE Transactions on Power Systems (2010) 25:1 (381-391))," *IEEE Transactions on Power Systems*, vol. 25, no. 2. p. 1200, May 2010. doi: 10.1109/TPWRS.2010.2047335.
- [24] N. Johansson, L. Ängquist, and H. P. Nee, "An adaptive controller for power system stability improvement and power flow control by means of a Thyristor Sitched Series Capacitor (TSSC),"

- IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 381–391, Feb. 2010, doi: 10.1109/TPWRS.2009.2036484.
- [25] K. Li, J. Zhao, C. Zhang, and W. J. Lee, "A study on mode-switching control of TCSC based on conditional firing of thyristor," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 1196–1202, Apr. 2011, doi: 10.1109/TPWRD.2010.2080290.
 - [26] S. Biswas, P. K. Nayak, and G. Pradhan, "An EMD-DT assisted intelligent protection scheme for TCSC compensated transmission lines," in *IEEE Power and Energy Society General Meeting*, IEEE Computer Society, Aug. 2020. doi: 10.1109/PESGM41954.2020.9281397.
 - [27] I. Motilal Nehru National Institute of Technology (Allahabad, I. D. of E. E. Motilal Nehru National Institute of Technology (Allahabad, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section. IE/PEL/CS Joint Chapter, and Institute of Electrical and Electronics Engineers, *ICPCES-2017 : the Role of Power, Control and Embedded Systems in Sustainable Society : March 9-11, 2017, Department of Electrical Engineering, Motilal Nehru National Institute of Technology Allahabad, India*.
 - [28] A. Akbar Emarloo, F. Mahmouditabar, and A. Shulaei, "Examination and Comparison of Thyristor and Gate-Controlled Series Capacitors Performance for the Voltage Stabilization of Sensitive Loads," in *2021 12th Power Electronics, Drive Systems, and Technologies Conference, PEDSTC 2021*, Institute of Electrical and Electronics Engineers Inc., Feb. 2021. doi: 10.1109/PEDSTC52094.2021.9405938.
 - [29] Institute of Electrical and Electronics Engineers. Madras Section and Institute of Electrical and Electronics Engineers, *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS) : 7-8 April 2016*.
 - [30] S. A. Chatterjee and K. D. Joshi, "Comparison of double loop, Direct-Output-Voltage and Fuzzy-PI control strategies for D-STATCOM," in *Proceedings - 3rd International Conference on Emerging Trends in Engineering and Technology, ICETET 2010*, 2010, pp. 322–327. doi: 10.1109/ICETET.2010.47.
 - [31] Institute of Electrical and Electronics Engineers. and Institute of Electrical and Electronics Engineers. Region 8., *EUROCON, 2007 : proceedings : The International Conference on "Computer as a Tool" : September 9-12, 2007, Warsaw, Poland*. IEEE, 2007.
 - [32] Ieee, *2012 International Conference on Advances in Engineering, Science and Management*. IEEE, 2012.
 - [33] Institute of Electrical and Electronics Engineers. Madras Section and Institute of Electrical and Electronics Engineers, *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS) : 7-8 April 2016*.
 - [34] Centurion University of Technology and Management, Institute of Electrical and Electronics Engineers. Bhubaneswar Subsection, and Institute of Electrical and Electronics Engineers, *International Conference on Signal Processing, Communication, Power and Embedded System : (SCOPES)-2016 : 3rd-5th, October 2016 : IEEE conference proceedings*.

- [35] H. Masdi, N. Mariun, S. M. Bashi, A. Mohamed, and S. Yusuf, "Design of a Prototype D-Statcom using DSP Controller for Voltage Sag Mitigation."
- [36] IEEE International Conference on Service-Oriented Computing and Applications, I. IEEE International Conference on Control, and K. D. ICCICCT 2014.07.10-11 Kumaracoil, *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 10-11 July 2014, Kumaracoil, Thuckalay, Kanyakumari Dist., Tamil Nadu, India.*
- [37] J. C. S. Rojas, D. A. Bautista López, D. M. Barrera Leguizamón, and F. R. Jiménez López, "Simulation of the Model, Design and Control of a Current Source Inverter with Unipolar SPWM Modulation."
- [38] S. Hannan, S. Aslam, and M. Ghayur, "Design and Real-Time Implementation of SPWM based Inverter."
- [39] I. M. Al-Adwan and Y. A. Al Shiboul, "Effect of the Modulation Index and the Carrier Frequency on the Output Vol Tage Waveform of the Spwm Voltage Source Inverter," in *Proceedings of the 17th International Multi-Conference on Systems, Signals and Devices, SSD 2020*, Institute of Electrical and Electronics Engineers Inc., Jul. 2020, pp. 960–968. doi: 10.1109/SSD49366.2020.9364104.
- [40] B. H. Yuditya, R. N. Hasanah, L. Ardhenta, and T. Taufik, "Performance Comparison of Single-Phase Multilevel Inverter with SPWM Unipolar and Bipolar Switching Techniques," in *Proceeding - 1st FORTEI-International Conference on Electrical Engineering, FORTEI-ICEE 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 102–107. doi: 10.1109/FORTEI-ICEE50915.2020.9249911.

Anexos

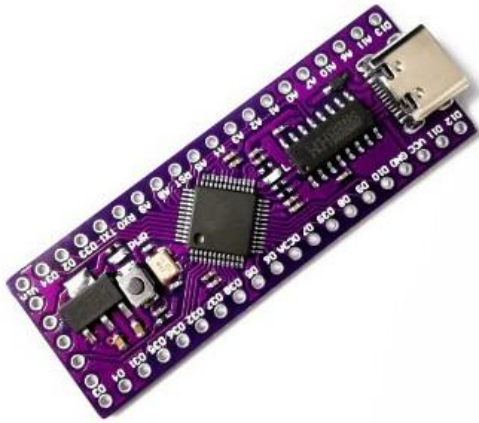
Anexo A : SELECCIÓN DE LOS EQUIPOS DE MEDICIÓN Y CONTROL	74
Anexo B: SOFTWARE Y PROGRAMACIÓN.	81

Anexo A : SELECCIÓN DE LOS EQUIPOS DE MEDICIÓN Y CONTROL

1. Tarjeta de control: Arduino LGT8F328P.

Esta es una tarjeta de programación y desarrollo de bajo costo basado en el Arduino nano, es compatible con el IDE Arduino y sus librerías. Esta tarjeta es mejor que el Arduino nano ya que tiene una frecuencia de reloj de hasta 32MHz, un DAC incorporad, un ADC de 12 bits y puerto de micro USB de tipo C.

Imagen 1 Arduino LGT8F328P



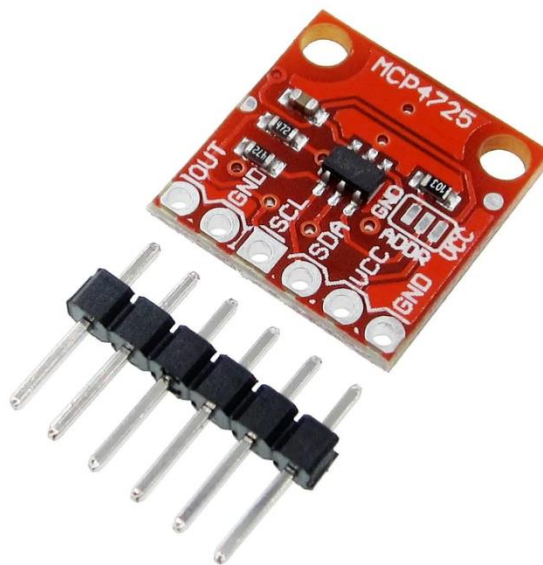
Especificaciones Técnicas.

- Microcontrolador: LGT8F328P (8-bit)
- Chip USB: CH340G
- Conector USB: USB-C
- Voltaje de alimentación: 7V - 12V DC
- Voltaje nivel lógico I/O: 5V DC
- Pines digitales I/O: 14
- Entradas analógicas: 8
- Corriente máx. por Pin I/O: 40mA
- Memoria FLASH: 32KB
- Memoria SRAM: 2KB
- Memoria EEPROM: 0KB
- Memoria E2PROM: 0K/1K/2K /4K /8K (compartido con FLASH)
- Frecuencia de Reloj: 32MHz
- Dimensiones: 18.5*43.2 mm
- Peso: 6.5 gramos

2. Módulo DAC MCP4725

El DAC es un convertidor digital a analógico que permite generar una señal analógica a partir de una señal digital, lo opuesto a un ADC. El módulo DAC MCP4725 permite agregar una salida analógica real a nuestro microcontrolador/Arduino/PIC/Raspberry y de esa forma poder generar señales tipo senoidal, triangular, etc. Posee interfaz de comunicación digital tipo I2C. Ideal para proyectos de control por voltaje analógico en reemplazo de un potenciómetro manual, otra aplicación es el campo del control automático como salida de un controlador PID.

Imagen 2 Módulo DAC MCP4725



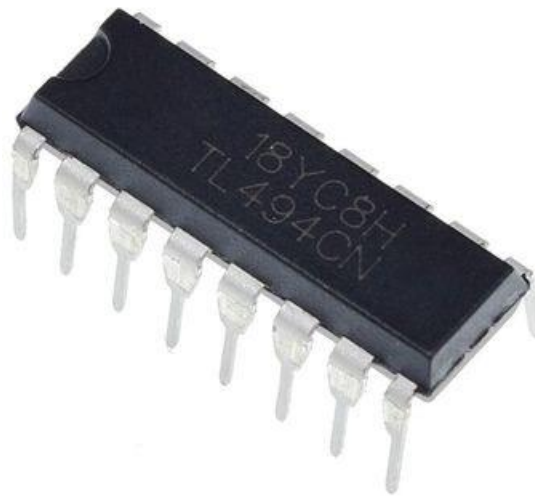
Especificaciones Técnicas.

- Voltaje de Alimentación: 2.7V a 5.5V
- CHIP: Microchip MCP4725
- Resolución de 12-bit (1.22mV si VCC=5V)
- Interfaz I2C (Standard, Fast, y High-Speed soportados)
- Dirección I2C: 0x60/0x61/0x62/0x63
- EEPROM interna para almacenar configuraciones
- Dimensiones: 19*19 mm
- Peso: 3 gramos

3. Modulador de ancho de pulso TL494

El TL494CD es un circuito de control PWM que incorpora todas las funciones necesarias en la construcción de un circuito de control de anchura de la modulación de impulsos (PWM) en un solo chip. Diseñado principalmente para el control de la fuente de alimentación, este dispositivo ofrece la flexibilidad para adaptar el circuito de control de fuente de alimentación para una aplicación específica.

Imagen 3 Modulador de ancho de pulso TL494



Especificaciones Técnicas.

- Voltaje de Alimentación: 2.7V a 5.5V
- Tensión de alimentación mínima: 7 V
- Tensión de alimentación máxima: 40 V
- Voltaje de salida: 40 V
- Corriente de salida: 500 mA
- Frecuencia: 40 kHz
- Temperatura de operación mínima: 0 °C
- Temperatura de operación máxima: 70 °C
- Tipo de caja: DIP
- Número de pines: 16
- Modelo: 33AH5119

4. Medidor PZEM-004T

El módulo de medidor multifunción PZEM-004T permite medir el voltaje RMS, corriente RMS, potencia activa y energía que toma una carga conectada a una línea monofásico de 110 / 220V como por ejemplo una estufa, Nevera, motor, electrodoméstico, etc..... esta información puede ser enviada a un microcontrolador (por ejemplo, Arduino o PIC), a la PC usando un adaptador USB a TTL, a un módulo WiFi ESP8266 para enviarla a internet (hay código disponible para esta aplicación) o a un PLC.

Imagen 4 Sensor PZEM-004T



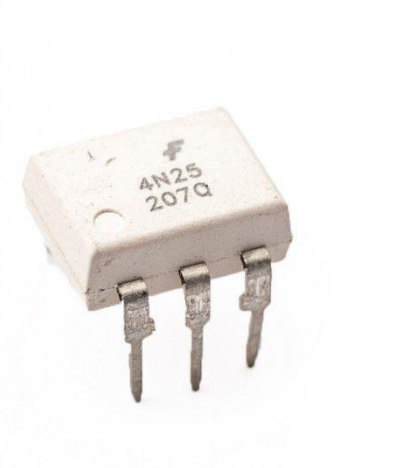
Especificaciones Técnicas.

- Voltaje de trabajo: 80~260VAC
- Voltaje de prueba: 80~260VAC
- Potencia nominal: 100A/22000W
- Frecuencia de trabajo: 45-65Hz
- Precisión de medición: 1.0

5. Optoacoplador 4N25

El 4N25 es un acoplador óptico de 1 canal salida del fototransistor con conexión a la base. Se compone de LED infrarrojo de arseniuro de galio y un fototransistor NPN de silicio.

Imagen 5 Optoacoplador 4N25



Especificaciones Técnicas.

- Tipo salida del optoacoplador: Phototransistor
- Número de canales: 1
- Voltaje de colector-emisor $V_{(br)ceo}$: 30 V
- Voltaje de aislamiento: 5 kV
- Corriente directa I_f máxima I_f : 60 mA
- Disipación de potencia PD: 150 mW
- CTR mínimo: 20 %
- Encapsulado: DIP
- Número de pines: 6
- Modelo: 4N25 +

6. Compuerta Lógica 74LS08

74LS08 Este integrado contiene cuatro compuertas independientes cada una de las cuales realiza la función lógica AND.

Imagen 6 Compuerta Lógica 74LS08



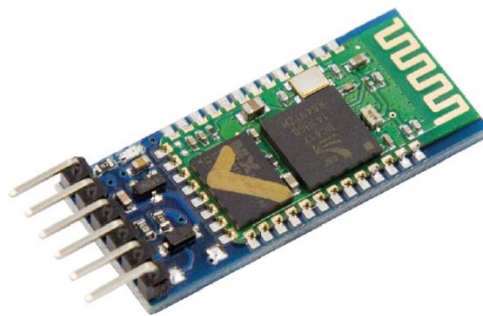
Especificaciones Técnicas.

- Tipo de familia: LS
- Tipo de la lógica: Compuerta AND
- Número de Entradas: 2
- Tensión de alimentación mínima: 4.75 V
- Tensión de alimentación máxima: 5.25 V
- Corriente de salida: 8 mA
- Temperatura de funcionamiento mínima: 0 °C
- Temperatura de funcionamiento máxima: 70° C
- Encapsulado: DIP
- 14 pines

7. Módulo Bluetooth hc-05

El módulo bluetooth HC-05 viene configurado de fábrica para trabajar como maestro o esclavo. En el modo maestro puede conectarse con otros módulos bluetooth, mientras que en el modo esclavo queda a la escucha peticiones de conexión. Agregando este módulo a tu proyecto podrás controlar a distancia desde un celular o una laptop todas las funcionalidades que desees.

Imagen 7 Modulo Bluetooth HC-05



Especificaciones Técnicas.

- Tipo de familia: LS
- Tipo de la lógica: Compuerta AND
- Voltaje de operación: 3.6V - 6V DC
- Consumo corriente: 50mA
- Bluetooth: V2.0+EDR
- Frecuencia: Banda ISM 2.4GHz
- Modulación: GFSK (Gaussian Frequency Shift Keying)
- Potencia de transmisión: 4dBm, Class 2
- Sensibilidad: -84dBm a 0.1% BER
- Alcance 10 metros
- Interface comunicación: Serial TTL
- Velocidad de transmisión: 1200bps hasta 1.3Mbps
- Baudrate por defecto: 38400,8,1,n.
- Seguridad: Autenticación y encriptación
- Temperatura de trabajo: -20C a +75C
- Compatible con Android
- Dimensiones: 37*16 mm
- Peso: 3.6 gramos

Anexo B: SOFTWARE Y PROGRAMACIÓN.

1. Bluetooth Electronics

Es una herramienta interactiva y gratuita que permite controlar un dispositivo electrónico con tu smartphone o Tablet Android. Esta aplicación se basa en una biblioteca de ejemplos que se pueden utilizar para crear tus propios proyectos. Cada ejemplo contiene una explicación, un diagrama de circuito y el código necesario para implementar el ejemplo. También puedes importar los ejemplos en tus proyectos.

Imagen 8 Bluetooth Electronics.



2. Programación del Arduino Control-SPWM.

El siguiente código es el encargado de generar las señales PWM de la etapa inversora, también es la encargada del control de sincronización y monitoreo de corrección.

```
#include <avr/io.h>
#include <avr/interrupt.h>

//-----
#include <Wire.h>

const byte vfbPin = A0,
          tfbPin = A1;
volatile double percentMod;
volatile int numXtd;
volatile int num;
int phs;
int ph;
int tstInt=0;
volatile int newICR1;
volatile int lockDet;
volatile int nivel = 0;
volatile int fase = 1;

int lookUp1[] = {
  0, 30, 60, 90, 120, 150, 179, 208, 237, 266, 294, 322, 349, 376, 402, 428, 453, 478, 501, 524,
  547, 568, 589, 609, 628, 646, 664, 680, 695, 710, 723, 735, 747, 757, 766, 774, 781, 787, 792, 796,
  798, 800, 800, 799, 797, 794, 790, 784, 778, 770, 762, 752, 741, 729, 717, 703, 688, 672, 655, 637,
  619, 599, 579, 558, 536, 513, 489, 465, 441, 415, 389, 363, 335, 308, 280, 252, 223, 194, 164, 135,
  105, 75, 45, 15
};

void setup() {
  TCCR1A = 0b10110000;
  TCCR1B = 0b00010001;
  TIMSK1 = 0b00000001;

  ICR1 = 796; //estaba en 796 1550
  sei();
  DDRB = 0b00011110;
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(4, INPUT);
  pinMode(8, INPUT); //Entrada para sincronizar
  digitalWrite(8, LOW);
  pinMode(7, OUTPUT); //Salida para activar la bocina
  digitalWrite(7, LOW); //Salida para activar la bocina
  pinMode(6, OUTPUT); //Salida para activar el Relé
  digitalWrite(6, LOW); //Salida para activar el Relé

  attachInterrupt(digitalPinToInterrupt(2), PulsoSerie, RISING);

  Wire.begin(1);
  Wire.onReceive(controlFP);
  Serial.begin(9600);

  percentMod = 0.01;
  for (int i = 0; i < 75; i++) { // Soft Start
    percentMod = percentMod + 0.01;
    delay(1);
  }
}
```

```

//función conexión serie
void PulsoSerie(){
    if(ph==fase && fase==1){
        if(digitalRead(3) == HIGH){
            if(num>nivel){
                num=num-(num-nivel);
                ICR1++;
            }else if(num<nivel){
                num=num+(nivel-num);
                ICR1--;
            }
        }else{
            num = 0;
        }
    }

    if(ph==fase && fase==0){
        if(num>nivel) num=num-(num-nivel);
        if(num<nivel) num=num+(nivel-num);
    }

    if(fase==0 && digitalRead(3) == HIGH && num<=(nivel+1) && num>=0 && num<=75){// estaba en 5 y 0
        ph=0;
        num = 0;
        if(num>nivel) num=num-(num-nivel);
        if(num<nivel) num=num+(nivel-num);
    }

    ICR1=constrain(ICR1,793,800);
}

void loop() {

ISR(TIMER1_OVF_vect) {
    static int dtA = 0, dtB = 10;
    OCR1A = int(lookUp1[num] * percentMod) + dtA; // SPWM width update
    OCR1B = int(lookUp1[num] * percentMod) + dtB; // note: 0.7 used to reduce inveter output voltage
    num++;
    if (num >= 83) {
        if (ph == 0) { // OC1A as SPWM out
            TCCR1A = 0b10110000; // clear OC1A, set OC1B on compare match
            dtA = 0; // no dead time
            dtB = 5; // adding dead time to OC1B
        } else {
            TCCR1A = 0b11100000; // OC1B as SPWM out
            dtA = 5;
            dtB = 0;
        }
        ph ^= 1;
    }

    if (num >= 83) { // toggle left bridge (60Hz)
        delayMicroseconds(60);//estaba en 50
        if (ph == 1) {
            digitalWrite(12, LOW);
            digitalWrite(11, HIGH);
            phs = 1;
        } else {
            digitalWrite(11, LOW);
            digitalWrite(12, HIGH);
            phs = 0;
        }
        num = 0;
    }
    numXtd++;

    if (num == 0 && digitalRead(8) == HIGH) digitalWrite(6, HIGH);
    if (digitalRead(8) == LOW) digitalWrite(6, LOW);
}
}

```

3. Programación del Arduino Control-Master

El siguiente código se encarga del control difuso, lo cual es el encargado de igualar el voltaje RMS de la etapa inversora con el voltaje de la red eléctrica, también esta el algoritmo neuronal que determina el voltaje de compensación antes de la sincronización, y por ultimo y no menos importante, el control P&O encargado de modificar el voltaje RMS de la etapa inversora para modificar el Factor de Potencia medido.

```
#include <Wire.h>
#include <Adafruit_MCP4725.h> //Libreria del DAC
Adafruit_MCP4725 dac; //variable del DAC

#include <Fuzzy.h> //Libreria control Fuzzy
Fuzzy *fuzzy = new Fuzzy();

//-----libreria PZEM-----
#include <PZEM004Tv30.h>
#include <SoftwareSerial.h>
#if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)
#define PZEM_RX_PIN 5
#define PZEM_TX_PIN 4
#endif

SoftwareSerial pzemSwSerial(PZEM_RX_PIN, PZEM_TX_PIN);
PZEM004Tv30 pzem(pzemSwSerial);
//-----fin libreria PZEM-----

volatile int contPorcen = 0; //para enviar porcentaje por medio de I2C
volatile int primera = 0; //para darle tiempo en el loop, despues se puede borrar
volatile int x = 3000; // Valor inicial del DAC para empezar con pulsos 0%
volatile int VSinc = 0; //indicador de sincronización
volatile float Vref; //guarda el voltaje medido de la red eléctrica
volatile float vinversor; //guarda el voltaje medido del inversor
volatile float volDC; //Guardar el voltaje DC medido
const byte entrada = A0; //Entrada del inversor al arduino
const byte entrada2 = A3; //Entrada de la señal AC al arduino
const byte entrada3 = A2; //Entrada voltaje DC convertidor dc/dc 820 serían 4v a 40,36v

String msincronizar = "Sincronización ";
String medicion = "Mediciones";
String controldac = "Control DAC";

//-----Inicio variables Función controlFP----
float FPcorriente = 0;
float FPwat = 0;
float FPvar = 0;
int FPcontrol = 0;
//-----Fin variables Función controlFP----
```

```

void setup() {
  pinMode(4,OUTPUT);
  pinMode(5,INPUT);
  pinMode(10,OUTPUT); // Pin que le avisa al otro Arduino
  digitalWrite(10,LOW); // Pin que le avisa al otro Arduino
  Serial.begin(9600);
  Wire.begin();
  dac.begin(0x60); //Conexión I2C del DAC
  dac.setVoltage(x, false); //Inicializar el DAC lo mas rapido posible

  Serial.print("*s"+String(msincronizar)+"*");
  Serial.print("*n"+String(medicion)+"*");
  Serial.print("*d"+String(controldac)+"*");

  for(int i=0;i<10;i++){
    // delay(200);
    Vref = MedirVoltaje();
    Serial.print("*b"+String(Vref)+"*");
  }
  codigofuzzy();
  //factorFUZZY();
}

double MedirVoltaje(){
  float lectura2 = 0, valor=0;
  for(int i=0;i<8400;i++){
    lectura2 = analogRead(entrada2);
    valor = lectura2*lectura2+valor;
  }
  valor = ((sqrt(valor/8400))/205)*(101);
  return valor;
}

//-----inicio función porcentajeduty I2C-----

```

```

void sincronizacion()
{
    Serial.print("*e"+String("sincronizacion")+"*");
    static int val = 0; // variable para enviar el valor I2C al otro arduino
    x = constrain(x,2000,3000);
    float lectura = 0, duty = 0, and3 = 0;

    //-----variables lectura PZEM-004T---
    float voltage = pzem.voltage();
    float current = pzem.current();
    float power = pzem.power();
    float energy = pzem.energy();
    float frequency = pzem.frequency();
    float pf = pzem.pf();
    float qreactiva = sin((acos(pf))*(voltage)*(current));
    //-----fin variables lectura PZEM-004T

    for(int i=0;i<8400;i++){
        lectura = analogRead(entrada);
        vinversor = lectura*lectura+vinversor;
    }
    vinversor = ((sqrt(vinversor/8400))/205)*(101);
    Serial.print("*v"+String(vinversor)+"*");

    //-----inicio código para enviar via I2C aumento del porcentaje duty del arduino SPWM-----
    while((contPorcen < 15) && (vinversor <= Vref*0.98)){
        medicion = "enviando datos I2C ";
        Serial.print("*n"+String(medicion)+"*");
        delay(500);
        val = 1;
        Wire.beginTransmission(1);
        Wire.write(val);
        Wire.endTransmission();
        for(int i=0;i<8400;i++){
            lectura = analogRead(entrada);
            vinversor = lectura*lectura+vinversor;
        }
        vinversor = ((sqrt(vinversor/8400))/205)*(101);
        contPorcen++;
        Serial.print("*v"+String(vinversor)+"*");
    }
    //-----fin código para enviar via I2C aumento del porcentaje duty del arduino SPWM-----

```

```

fuzzy->setInput(1, vinversor);
fuzzy->fuzzify();
duty = fuzzy->defuzzify(1);

duty = round(duty);
x = x+(duty);
dac.setVoltage(x, false);

ard3 = analogRead(entrada3);
volDC = (ard3/205)*(111/11);

Serial.print("*t"+String(volDC)+"*");
Serial.print("*x"+String(x)+"*");
if (duty!=0){
    msincronizar = "Sincronizando";
    controldac = "Modif. DAC";
}else{
    msincronizar = "Sincronizado";
    controldac = "DAC estable";
    VSinc = 1;
}
//Serial.print("*x"+String(x)+"*");
Serial.print("*s"+String(msincronizar)+"*");
Serial.print("*d"+String(controldac)+"*");

Serial.print("*z"+String(voltage)+"*");
Serial.print("*c"+String(current)+"*");
Serial.print("*p"+String(power)+"*");
Serial.print("*a"+String(pf)+"*");
Serial.print("*f"+String(frequency)+"*");
Serial.print("*q"+String(qreactiva)+"*");
}
//-----Fin de función de sincronización-----

```

```

void codigofuzzy(){
    FuzzyInput *voltaje = new FuzzyInput(1);
    FuzzySet *voltajeB3 = new FuzzySet(Vref-6, Vref-6, Vref-5, Vref-4);
    voltaje->addFuzzySet(voltajeB3);
    FuzzySet *voltajeB2 = new FuzzySet(Vref-5, Vref-4, Vref-3, Vref-2);
    voltaje->addFuzzySet(voltajeB2);
    FuzzySet *voltajeB1 = new FuzzySet(Vref-3, Vref-2, Vref-1, Vref);
    voltaje->addFuzzySet(voltajeB1);
    FuzzySet *voltaje0 = new FuzzySet(Vref-1, Vref, Vref+2, Vref+3);
    voltaje->addFuzzySet(voltaje0);
    FuzzySet *voltajeA1 = new FuzzySet(Vref+2, Vref+3, Vref+4, Vref+5);
    voltaje->addFuzzySet(voltajeA1);
    FuzzySet *voltajeA2 = new FuzzySet(Vref+4, Vref+5, Vref+6, Vref+7);
    voltaje->addFuzzySet(voltajeA2);
    FuzzySet *voltajeA3 = new FuzzySet(Vref+6, Vref+7, Vref+8, Vref+8);
    voltaje->addFuzzySet(voltajeA3);
    fuzzy->addFuzzyInput(voltaje);

    FuzzyOutput *duty = new FuzzyOutput(1);
    FuzzySet *dutyN3 = new FuzzySet(-4, -3, -3, -2);
    duty->addFuzzySet(dutyN3);
    FuzzySet *dutyN2 = new FuzzySet(-3, -2, -2, -1);
    duty->addFuzzySet(dutyN2);
    FuzzySet *dutyN1 = new FuzzySet(-2, -1, -1, 0);
    duty->addFuzzySet(dutyN1);
    FuzzySet *duty0 = new FuzzySet(-1, 0, 0, 1);
    duty->addFuzzySet(duty0);
    FuzzySet *dutyP1 = new FuzzySet(0, 1, 1, 2);
    duty->addFuzzySet(dutyP1);
    FuzzySet *dutyP2 = new FuzzySet(1, 2, 2, 3);
    duty->addFuzzySet(dutyP2);
    FuzzySet *dutyP3 = new FuzzySet(2, 3, 3, 4);
    duty->addFuzzySet(dutyP3);
    fuzzy->addFuzzyOutput(duty);

    //Reglas de control
    FuzzyRuleAntecedent *ifvoltajeB3 = new FuzzyRuleAntecedent();
    ifvoltajeB3->joinSingle(voltajeB3);
    FuzzyRuleConsequent *tamdutyN3 = new FuzzyRuleConsequent();
    tamdutyN3->addOutput(dutyN3);
    FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifvoltajeB3, tamdutyN3);
    fuzzy->addFuzzyRule(fuzzyRule01);

```

```

FuzzyRuleAntecedent *ifvoltajeB2 = new FuzzyRuleAntecedent();
ifvoltajeB2->joinSingle(voltajeB2);
FuzzyRuleConsequent *tamdutyN2 = new FuzzyRuleConsequent();
tamdutyN2->addOutput(dutyN2);
FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifvoltajeB2, tamdutyN2);
fuzzy->addFuzzyRule(fuzzyRule02);

FuzzyRuleAntecedent *ifvoltajeB1 = new FuzzyRuleAntecedent();
ifvoltajeB1->joinSingle(voltajeB1);
FuzzyRuleConsequent *tamdutyN1 = new FuzzyRuleConsequent();
tamdutyN1->addOutput(dutyN1);
FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifvoltajeB1, tamdutyN1);
fuzzy->addFuzzyRule(fuzzyRule03);

FuzzyRuleAntecedent *ifvoltaje0 = new FuzzyRuleAntecedent();
ifvoltaje0->joinSingle(voltaje0);
FuzzyRuleConsequent *tamduty0 = new FuzzyRuleConsequent();
tamduty0->addOutput(duty0);
FuzzyRule *fuzzyRule04 = new FuzzyRule(4, ifvoltaje0, tamduty0);
fuzzy->addFuzzyRule(fuzzyRule04);

FuzzyRuleAntecedent *ifvoltajeA1 = new FuzzyRuleAntecedent();
ifvoltajeA1->joinSingle(voltajeA1);
FuzzyRuleConsequent *tamdutyP1 = new FuzzyRuleConsequent();
tamdutyP1->addOutput(dutyP1);
FuzzyRule *fuzzyRule05 = new FuzzyRule(5, ifvoltajeA1, tamdutyP1);
fuzzy->addFuzzyRule(fuzzyRule05);

FuzzyRuleAntecedent *ifvoltajeA2 = new FuzzyRuleAntecedent();
ifvoltajeA2->joinSingle(voltajeA2);
FuzzyRuleConsequent *tamdutyP2 = new FuzzyRuleConsequent();
tamdutyP2->addOutput(dutyP2);
FuzzyRule *fuzzyRule06 = new FuzzyRule(6, ifvoltajeA2, tamdutyP2);
fuzzy->addFuzzyRule(fuzzyRule06);

FuzzyRuleAntecedent *ifvoltajeA3 = new FuzzyRuleAntecedent();
ifvoltajeA3->joinSingle(voltajeA3);
FuzzyRuleConsequent *tamdutyP3 = new FuzzyRuleConsequent();
tamdutyP3->addOutput(dutyP3);
FuzzyRule *fuzzyRule07 = new FuzzyRule(7, ifvoltajeA3, tamdutyP3);
fuzzy->addFuzzyRule(fuzzyRule07);
}

```

```

void factorFUZZY(){
    //-----inicio/control del FP potencia por fuzzy-----
    FuzzyInput *factor = new FuzzyInput(2);
    FuzzySet *FPI2 = new FuzzySet(0.6, 0.6, 0.87, 0.89);
    factor->addFuzzySet(FPI2);
    FuzzySet *FPI1 = new FuzzySet(0.85, 0.88, 0.88, 0.91);
    factor->addFuzzySet(FPI1);
    FuzzySet *FPC = new FuzzySet(0.88, 0.89, 0.91, 0.92);
    factor->addFuzzySet(FPC);
    FuzzySet *FPD1 = new FuzzySet(0.89, 0.92, 0.92, 0.95);
    factor->addFuzzySet(FPD1);
    FuzzySet *FPD2 = new FuzzySet(0.91, 0.93, 1, 1);
    factor->addFuzzySet(FPD2);
    fuzzy->addFuzzyInput(factor);

    FuzzyOutput *dutyf = new FuzzyOutput(2);
    FuzzySet *DN2 = new FuzzySet(-5, -5, -4, -3);
    dutyf->addFuzzySet(DN2);
    FuzzySet *DN1 = new FuzzySet(-4, -2, -2, 0);
    dutyf->addFuzzySet(DN1);
    FuzzySet *DC = new FuzzySet(-2, 0, 0, 2);
    dutyf->addFuzzySet(DC);
    FuzzySet *DP1 = new FuzzySet(0, 2, 2, 4);
    dutyf->addFuzzySet(DP1);
    FuzzySet *DP2 = new FuzzySet(2, 4, 5, 5);
    dutyf->addFuzzySet(DP2);
    fuzzy->addFuzzyOutput(dutyf);

    //Reglas de control
    FuzzyRuleAntecedent *ifFPI2 = new FuzzyRuleAntecedent();
    ifFPI2->joinSingle(FPI2);
    FuzzyRuleConsequent *tamDN2 = new FuzzyRuleConsequent();
    tamDN2->addOutput(DN2);
    FuzzyRule *fuzzyRegla1 = new FuzzyRule(1, ifFPI2, tamDN2);
    fuzzy->addFuzzyRule(fuzzyRegla1);

    FuzzyRuleAntecedent *ifFPI1 = new FuzzyRuleAntecedent();
    ifFPI1->joinSingle(FPI1);
    FuzzyRuleConsequent *tamDN1 = new FuzzyRuleConsequent();
    tamDN1->addOutput(DN1);
    FuzzyRule *fuzzyRegla2 = new FuzzyRule(2, ifFPI1, tamDN1);
    fuzzy->addFuzzyRule(fuzzyRegla2);
}

```

```

FuzzyRuleAntecedent *ifFPC = new FuzzyRuleAntecedent();
ifFPC->joinSingle(FPC);
FuzzyRuleConsequent *tamDC = new FuzzyRuleConsequent();
tamDC->addOutput(DC);
FuzzyRule *fuzzyRegla3 = new FuzzyRule(3, ifFPC, tamDC);
fuzzy->addFuzzyRule(fuzzyRegla3);

FuzzyRuleAntecedent *ifFPD1 = new FuzzyRuleAntecedent();
ifFPD1->joinSingle(FPD1);
FuzzyRuleConsequent *tamDP1 = new FuzzyRuleConsequent();
tamDP1->addOutput(DP1);
FuzzyRule *fuzzyRegla4 = new FuzzyRule(4, ifFPD1, tamDP1);
fuzzy->addFuzzyRule(fuzzyRegla4);

FuzzyRuleAntecedent *ifFPD2 = new FuzzyRuleAntecedent();
ifFPD2->joinSingle(FPD2);
FuzzyRuleConsequent *tamDP2 = new FuzzyRuleConsequent();
tamDP2->addOutput(DP2);
FuzzyRule *fuzzyRegla5 = new FuzzyRule(5, ifFPD2, tamDP2);
fuzzy->addFuzzyRule(fuzzyRegla5);
//-----final/control del FP potencia por fuzzy-----
}

//-----Inicio de sincronización con redes neuronales-----
void ajustandoRedes(){
    Serial.print("*e"+String("ajustandoRedes")+"*");
    controldac = "Cal. DAC Nuevo";
    Serial.print("*d"+String(controldac)+"*");
    delay(5000);

    float p1, p2, p3, p4, p5, p6, n1, n2, n3, n4, n5, r1, r2, r3, r4, r5;
    float voltage = pzem.voltage();
    float current = pzem.current();
    float power = pzem.power();
    float frequency = pzem.frequency();
    float pf = pzem.pf();
    float qreactiva = sin((acos(pf)))*(voltage)*(current);

```

```

p1 = Vref;
p2 = voltage;
p3 = current;
p4 = power;
p5 = qreactiva;
p6 = pf;
n1 = p1*(1.9409899)+p2*(1.439289)+p3*(0.19086878)+p4*(-1.3682039)+p5*(1.712795)+p6*(0.52335304)+(1.1498305);
n2 = p1*(0.5543062)+p2*(-0.53611374)+p3*(-1.0040519)+p4*(-0.9338232)+p5*(-0.15242693)+p6*(-0.61381304)+(-0.2663546);
if (n1>=0) r1 = n1; else r1 = 0;
if (n2>=0) r2 = n2; else r2 = 0;
n3 = r1*(1.4657729)+r2*(0.9387726)+(0.9860326);
n4 = r1*(2.1178136)+r2*(-0.9397003)+(0.9193358);
if (n3>=0) r3 = n3; else r3 = 0;
if (n4>=0) r4 = n4; else r4 = 0;
n5 = r3*(1.8075663)+r4*(2.1481438)+(0.74887025);
if (n5>=0) r5 = n5; else r5 = 2700;

controldac = "DAC. Calculado";
Serial.print("**d"+String(controldac)+"*");
delay(5000);
x = r5;
controldac = "DAC. Registrado";
dac.setVoltage(x, false);

//Volver a medir el voltaje del inversor despues de aplicar el DAC
float medirR = 0, VoltajeR=0;
for(int i=0;i<8400;i++){
    medirR = analogRead(entrada);
    VoltajeR = medirR*medirR+VoltajeR;
}
VoltajeR = ((sqrt(VoltajeR/8400))/205)*(101);
Serial.print("**v"+String(VoltajeR)+"*");

//Volver a medir el voltaje del inversor despues de aplicar el DAC

```

```

Serial.print("*d"+String(controldac)+"");

Serial.print("*z"+String(voltage)+"");
Serial.print("*c"+String(current)+"");
Serial.print("*p"+String(power)+"");
Serial.print("*a"+String(pf)+"");
Serial.print("*f"+String(frequency)+"");
Serial.print("*q"+String(qreactiva)+"");
Serial.print("*x"+String(x)+"");
delay(5000);
VSinc = 2;
}
//-----Fin de sincronización con redes neuronales-----

//-----Control Factor de Potencia-----
void controlFP(){
  Serial.print("*e"+String("controlFP")+"");
  float dutyF;
  float voltage = pzem.voltage();
  float current = pzem.current();
  float power = pzem.power();
  float frequency = pzem.frequency();
  float pf = pzem.pf();
  float qreactiva = sin((acos(pf)))*(voltage)*(current);

  Serial.print("*z"+String(voltage)+"");
  Serial.print("*c"+String(current)+"");
  Serial.print("*p"+String(power)+"");
  Serial.print("*a"+String(pf)+"");
  Serial.print("*f"+String(frequency)+"");
  Serial.print("*q"+String(qreactiva)+"");

  if((pf<0.9) && (current <= 0.4)){
    x = x-1;
    dac.setVoltage(x, false);
    Serial.print("*x"+String(x)+"");
  }else if((pf>0.9) && (current <= 0.4)){
    x = x+1;
    dac.setVoltage(x, false);
    Serial.print("*x"+String(x)+"");
  }
}

```

```
void loop() {  
  
    if(primeras == 0){  
        Serial.print("*e"+String("Iniciando")+"*");  
        delay(8000);  
    }  
  
    if(VSinc == 0){  
        sincronizacion();  
    }else if(VSinc == 1){  
        ajustandoRedes();  
    }else if(VSinc == 2){  
        controlFP();  
    }  
  
    primeras = 1;  
}
```