

**DISEÑO E IMPLEMENTACIÓN DE UN INVERSOR TRIFÁSICO CONTROLADO
EN TENSIÓN A PARTIR DE UN BUS DC**

POR:

JUAN FELIPE SILVA PERICO

TRABAJO DE GRADO PRESENTADO PARA OTORGAR EL TÍTULO DE
INGENIERO ELECTRÓNICO DE LA UNIVERSIDAD SANTO TOMAS

BOGOTA DC
UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA 2019
DIVISION DE INGENIERIAS

**DISEÑO E IMPLEMENTACIÓN DE UN INVERSOR TRIFÁSICO CONTROLADO
EN TENSIÓN A PARTIR DE UN BUS DC**

POR:

JUAN FELIPE SILVA PERICO

DIRIGIDO POR:

ING.CARLOS JAVIER MOJICA CASALLAS

BOGOTA DC

UNIVERSIDAD SANTO TOMÁS

FACULTAD DE INGENIERÍA ELECTRÓNICA 2019

DIVISION DE INGENIERIAS

Nota de aceptación

EDWIN FRANCISCO FORERO GARCIA
Jurado

OSCAR MAURICIO GELVEZ LIZARAZO
Jurado

Bogotá, 2 de mayo del 2019

A mis papas que siempre me han guiado por el camino correcto formándome como persona correcta y siempre pendientes en mi proceso de aprendizaje, a mi novia por ser ese apoyo incondicional y por la gran ayuda prestada por parte de ella en el desarrollo de este proyecto, al profesor e ingeniero de la universidad Santo Tomas Carlos Javier Mojica quien ha sido el acompañante y director de este proceso de aprendizaje aportando con sus conocimientos en este proceso de aprendizaje, a dios por permitirme culminar mis estudios universitarios creciendo como persona y como profesional.

Tabla de contenido

Capítulo 1 DESCRIPCION DEL PROYECTO	10
Capítulo 2 OBJETIVOS	12
OBETIVO GENERAL	12
OBJETIVOS ESPECIFICOS	12
Capítulo 3 JUSTIFICACION	13
Capítulo 4 ANTECEDENTES	14
Capítulo 5 EXPERIMETER KIT TMS320F28335	18
GPIO	18
PIE	20
PWM	21
BASE DE TIEMPOS	22
COMPARACION	23
SELECCIÓN DE POLARIDAD	23
BANDA MUERTA	24
DETECCIÓN DE FALLAS	25
ADC	26
Capítulo 6 INVERSOR IMPLEMENTADO	29
MODELAMIENTO DEL CIRCUITO INVERSOR	29
MODELO DE SEÑAL GRANDE	34
MODELO DE SEÑAL PEQUEÑA	34

CÁLCULO DE LA INDUCTANCIA	35
CÁLCULO DE LA CAPACITANCIA	37
Capítulo 7 DISEÑO COMPENSADOR INVERSOR	39
CONTROL POR REALIMENTACIÓN DE ESTADOS	39
DISEÑO POR UBICACIÓN DE POLOS A TRAVÉS DE REALIMENTACIÓN DE ESTADOS	39
IMPLEMENTACION DEL CONTROL POR REALIMENTACIÓN DE ESTADOS	40
IMPLEMENTACIÓN DEL CONTROL POR REALIMENTACIÓN DE ESTADOS EN TIEMPO DISCRETO	45
Capítulo 8 ETAPA DE MEDICIÓN	50
Capítulo 9 RESULTADOS	52
RESULTADOS EXPERIMENTALES DEL INVERSOR	52
Capitulo10 CONCLUSIONES	70
Capítulo 11 REFERENCIAS	73
Capitulo12 ANEXOS	74
PROGRAMA MATLAB	74
CALCULO PARA EL VECTOR K EN TIEMPO DISCRETO	75
PROGRAMA EN LENGUAJE C IMPLEMENTANDO EN LA DSP	76

Tabla de Figuras

Figura 5.1 Experimenter kit	18
Figura 5.2 Multiplexado de interrupciones	20
Figura 5.3 Diferentes niveles de comparación	23
Figura 5.4 Modulo banda muerta	24
Figura 5.5 formas de onda banda muerta para casos típicos	25
Figura 5.6 Configuración ADC modo cascada.	27
Figura 6.1 plano general del inversor	29
Figura 6.2 Análisis del circuito	30
Figura 7.1 Diagrama de bloques del sistema de control mediante realimentación de estados	39
Figura 7.2 Diagrama de bloques total del sistema de control mediante realimentación de estados	43
Figura 7.3 Diagrama de bloques total del sistema de control mediante realimentación de estados	44
Figura 7.4 señal de referencia (Azul) Vs señal de salida (Roja) del sistema con perturbaciones y variaciones de amplitud	45
Figura 7.5 Diagrama de bloques total del sistema en tiempo discreto de control mediante realimentación de	47
Figura 7.6 Diagrama de bloques total del sistema de control mediante realimentación de estados	48
Figura 7.7 señal de referencia (roja) Vs señal de salida (azul) del sistema con perturbaciones	49
Figura 8.1 Aplicación típica	51

Figura 9.1 salida inversor a plena carga	53
Figura 9.2 salida inversor con dos cargas activas	54
Figura 9.3 salida inversor con una carga activa	55
Figura 9.4 salida inversor con carga en vacío	56
Figura 9.5 salida inversor a plena carga - tensión de referencia mínima	57
Figura 9.6 salida inversor con una carga activa - tensión de referencia mínima	58
Figura 97 salida inversor con plena carga - tensión de referencia 70V	59
Figura 98 salida inversor con carga en vacío - tensión de referencia 70V	60
Figura 99 salida inversor diferencia de cargas en las líneas	61
Figura 910 salida inversor diferencia de cargas en cada línea	62
Figura 9.11 salida inversor con plena carga - tensión del bus DC 300V	63
Figura 9.12 Corriente Vs alta Tensión plena carga línea A	64
Figura 9.13 corriente Vs alta Tensión plena carga línea B	65
Figura 9.0.14 Corriente Vs alta Tensión plena carga línea C	66
Figura 9.15 Corriente Vs baja tensión plena cargar línea A	67
Figura 9.16 Corriente Vs baja tensión plena carga línea B	68
Figura 9.17 corriente Vs baja tensión plena carga línea C	69

INDICE DE TABLAS

Tabla 9.1 Equivalencia de colores	53
Tabla 9.2 Equivalencia de colores	54
Tabla 9.3 Equivalencia de colores	55
Tabla 9.4 Equivalencia de colores	56
Tabla 9.5 Equivalencia de colores	57
Tabla 9.6 Equivalencia de colores	58
Tabla 9.7 Equivalencia de colores	59
Tabla 9.8 Equivalencia de colores	60
Tabla 9.9 Equivalencia de colores	61
Tabla 9.10 Equivalencia de colores	62
Tabla 9.0.11 Equivalencia de colores	63
Tabla 9.0.12 Equivalencia de colores	64
Tabla 9.13 Equivalencia de colores	65
Tabla 9.14 Equivalencia de colores	66
Tabla 9.15 Equivalencia de colores	67
Tabla 9.16 Equivalencia de colores	68
Tabla 9.17 Equivalencia de colores	69

Capítulo 1 DESCRIPCION DEL PROYECTO

Actualmente en la industria nos encontramos con inversores de tensión monofásicos o trifásicos con características específicas dependiendo su uso; si la potencia a trabajar es superior a 1/2 kVA, son equipos grandes (inversor solar X3 “46.4cm*47.6cm*18cm” o inversor mastervolt mass “47.2cm*31.8cm*17.8cm”) y pesados (26.9kg – 15.3kg)[14] que en muchos casos solo se pueden trabajar a una frecuencia establecida de 60 Hz, además demandan costos significativos (\$10.697.759 Colombianos o 2967.77 euros - \$8.442.566 Colombianos), disminuyendo eficiencia y aumentando costos por la energía consumida. Por otra parte, en las empresas o industrias es muy posible que se tengan herramientas de trabajo manual que solo pueden ser conectados a los inversores instalados localmente, haciéndose necesario tener que doblar los costos en la adquisición de herramientas eléctricas para las diferentes redes de tensión. Es de gran utilidad contar con un inversor que pueda generar una o varias salidas trifásicas, que sea un equipo más pequeño, portátil, de ser posible de menor costo y con la capacidad de tener variación en un rango delimitado tensión controlada.

Este proyecto de grado busca diseñar y construir un inversor DC - AC que tenga posibilidad de entregar salidas monofásicas y generar una tensión trifásica, el sistema debe tener la capacidad de controlar la tensión trifásica, este proyecto se dividió en tres etapas principales para su desarrollo:

- Diseño e implementación de un Inversor trifásico el cual entrega cerca de 100W/fase, se analiza paso a paso la construcción de este, relacionando la matemática requerida con el correspondiente diseño.
- La etapa de censado por medio de la tarjeta sensora y un convertidor analógico digital, permite sensar la tensión y corriente de cada línea de manera diferencial y por medio del convertidor analógico digital dirigirlas al DSP y procesar los valores sensados para luego realizar la etapa de control.

- El proyecto contiene una etapa de control la cual será implementada utilizando la tarjeta DSP. esta etapa permitirá realizar una topología de control más robusta con respecto a las tradicionales sobre el sistema construido, ya que será la encargada de mantener los niveles de tensión dentro de los valores establecidos por el diseño.

Este proyecto contiene un análisis enfocado a diferentes dispositivos electrónicos como transistores, sensores y circuitos de disparo, que a ser manejados de manera correcta permitirán realizar la implementación de un convertidor DC-AC o también llamado inversor.

Este proyecto implemento un inversor de tensión trifásica que contó con una tensión de salida de 85 Vrms por fase. Se cuenta con 300W totales, esta se divide en tres para tener una potencia por fase cerca de 100W.

Finalmente se efectuará el análisis de información adquirida, para realizar las correspondientes conclusiones del proyecto.

A continuación, se presentan los objetivos generales y específicos que constituyen el núcleo principal para el desarrollo de este proyecto.

Capítulo 2 OBJETIVOS

OBETIVO GENERAL

Diseñar e implementar un inversor de voltaje trifásico controlado en tensión a partir de un bus DC

OBJETIVOS ESPECIFICOS

- Diseñar e implementar un convertidor DC-AC el cual permita obtener una salida de tensión trifásica.
- Implementar los algoritmos de control digital para la tensión del convertidor DC-AC mediante una “DSP”.
- Implementar un circuito electrónico el cual permita sensor cada una de las líneas de tensión y corriente deseados

Capítulo 3 JUSTIFICACION

Se realizo la implementación de un inversor de tensión trifásica a partir de un bus DC, el diseño pretendió estar acorde con diferentes proyectos o sectores laborales e iniciar un proceso de aprendizaje con mayor profundidad al abordado en los diferentes espacios académicos. En la academia, estos equipos son muy comunes y pueden ser de tipo monofásicos o trifásicos.

Se utilizaron los conceptos adquiridos a lo largo de la carrera para diseñar un inversor trifásico partiendo de un bus DC, que tenga la posibilidad de tener variación y control en un rango determinado para la tensión de salida; el tema de electrónica de potencia es de mi total agrado y, además, combinar los temas de control aplicados a la potencia lo serán aún más.

La implementación con semiconductores y técnicas de conmutación por PWM permiten tener la posibilidad de variación en voltaje, al aumentar la frecuencia de conmutación se garantizan una reducción en el tamaño de los embobinados y capacitancias, lo cual disminuye el tamaño final del sistema, esto permite tener portabilidad del equipo construido. Muchos de los equipos actuales funcionan con transformadores que trabajan con una frecuencia fija (60Hz) causando un aumento de tamaño en los embobinados los cuales, a su vez, afectan el factor de potencia, eficiencia, costos y tamaños.

Capítulo 4 ANTECEDENTES

Los reguladores de voltaje han sido implementados de tal manera que sean manipulados y utilizados en las áreas de trabajo para las cuales fueron requeridos y el diseño e implementación de estos depende en muchos casos del ingeniero a cargo y de las topologías que este desee manejar.

Para el año 1968 la empresa Fairchild logró lanzar al mercado un circuito integrado el uA 723 como primer regulador de voltaje lineal de la historia, configurado por un amplificador operacional, un diodo zener, unas resistencias y unos transistores. Logrando así el primer regulador sólido el cual se utiliza ampliamente. La capacidad de entrega de corriente es limitada por lo que el diseñador debe aumentarle transistores externos para aumentar dicha entrega dependiendo de los parámetros a diseñar por el ingeniero a cargo. También se adiciona una resistencia para limitar la corriente a un valor aceptable en caso de que haya un corto circuito en la salida para los primeros diseños [1].

Para el año 2007 se publicó un artículo en el cual se titula “Modulación pwm aplicada a inversores trifásicos dentro del esquema de accionamientos eléctricos” de la universidad del cauca, en él se dice que para aplicaciones en motores se utilizan inversores ya que permiten tener variación en tensión y frecuencia.

El artículo está enfocado en accionamiento eléctricos los cuales pueden estar en baja o mediana potencia, utilizan una topología de inversor VSI (inversores con fuente de voltaje) de puente completo, el cual utiliza par de diodos que pueden estar en corte o saturación haciendo que su comportamiento sea a manera de switch, donde la salida de estos va a tener un valor de $V_s/2$ con respecto a la terminal de neutro, para que se dé la conmutación de los transistores se emplean modulaciones por anchos de pulso; este artículo aborda estos temas a manera de secciones en el cual hablarán acerca de moduladores de pwm, accionamientos eléctricos, fundamentos de pwm con enfoques CB-PWM y SV-PWM, proponen una modulación por vectores espaciales para pwm híbridos.[2]

Para el año 2008 se publicó un artículo en el cual se menciona el tema de diseño y construcción de un inversor trifásico, titulado “*DISEÑO Y CONSTRUCCIÓN DE UN INVERSOR TRIFÁSICO*”, publicado por la universidad tecnológica de Pereira en el cual establecen los parámetros de diseño y construcción de este sistema, haciendo énfasis en el tipo de carga a utilizar y la tensión de prueba, además, la implementación de este sistema tiene la finalidad de poder tener una corrección en el factor de potencia y tiene en cuenta el avance que se ha venido experimentando con semiconductores, para frecuencias de conmutación, pérdidas en conducción y facilidad de manejo y diseño optimizando en gran parte este tipo de convertidores.[3]

Un año después en el 2010 se publicó un documento titulado “Diseño e implementación de un prototipo de inversor trifásico orientados a redes de distribución” donde busca implementar un inversor que pueda operar en baja tensión con la capacidad de trabajar en isla sincronizándose a la red controlando su potencia activa y reactiva, el quipo planteado en el documento busca recibir energía de pequeños generadores, para ser entregada cuando sea requerida su control será en la tensión y estabilización la red.

El documento esta seccionado por capítulos en los primero abarcan temas sobre los conceptos de generación virtual, micro-red, sistemas de almacenamiento de energía en baterías.

En los siguientes capítulos hablan de un puente inversor acoplado a red mediante un filtro LC, y para un diseño de control PI síncrono configurado como esclavo y maestro, esclavo para controlar potencia activa y reactiva, maestro controlando la tensión, este control de manera digital programado en un microcontrolador. [4]

Para el año 2012 se realizó una publicación sobre inversores de la universidad Carlos III de Madrid el cual se titula “Análisis comparativo de inversores fotovoltaicos de conexión a red con potencia igual o superior a 100kW” donde en el se pretende analizar a manera más detalla un inversor fotovoltaico de conexión a la red.

En ese proyecto en el capítulo 4 mencionan un bloque de conversión donde se refieren a el inversor como un circuito el cual transforma la corriente continua producida por el generador fotovoltaico en una corriente alterna con unas características especificadas por la red de 230V como valor eficaz para la tensión a una frecuencia de 50Hz, detallan más a fondo las características que deben cumplir los inversores conectados a la red, donde si este tiene una potencia nominal menor a los 5kW debe tener una conexión monofásica, mientras si es igual o mayor a los 5kW su conexión debe ser de manera trifásica.

El documento hace una comparativa muy detalla de esta configuración de inversores abarcando temas de pérdidas producidas por cada una de las partes que lo compone, rendimiento del inversor y la celda fotovoltaica, protecciones eléctricas, sistemas de control y compara modelos de inversores fotovoltaicos fabricados por múltiples empresas.[5]

En el año 2012 se publica un proyecto titulado “Diseño y análisis de desempeño de un inversor de voltaje utilizando controladores inteligentes” este proyecto busca analizar alternativas de controladores para la tensión con el fin de disminuir la distorsión armónica generada a la salida, modelan un inversor monofásico configurado en medio puente con modulación senoidal por ancho de pulso utilizando “MatLab”, diseñan un control clásico, neuronal y difuso para así poder comparar las ventajas de cada uno de los métodos aplicados a este dispositivo.[6]

Para el año 2014 se publicó un proyecto titulado “Análisis y diseño de un puente inversor Trifásico VSI controlado mediante modulación por vector espacial (SVM por sus siglas en inglés) utilizando microcontrolador” de la universidad tecnológica de Pereira, en el cual diseñan e implementan una estrategia de control para un inversor trifásico, como herramienta para reducción de pérdidas por conmutación y distorsión de señales de corriente y tensión con los que alimentará la carga de salida.[7]

Basados en las investigaciones anteriores, las diferentes topologías, implementaciones y diseños de inversores de tensión, se ha tenido gran impacto

frente a la elaboración de diferentes inversores de tensión, Se tiene una gran expectativa con la implementación de un inversor de tensión que permita tener diversas salidas, menores pérdidas, que será implementado con semiconductores, control digital, circuitos para la reducción del ruido tanto en las diferentes salidas como entradas de este, buscando además un menor tamaño y optimizando ambientes de trabajo.

Capítulo 5 EXPERIMETER KIT TMS320F28335

El Proyecto contará con una DSP de Texas instruments de referencia “TMS320F28335”, la cual se puede conseguir como kit académico, contiene dos tarjetas diseñadas por el mismo fabricante.

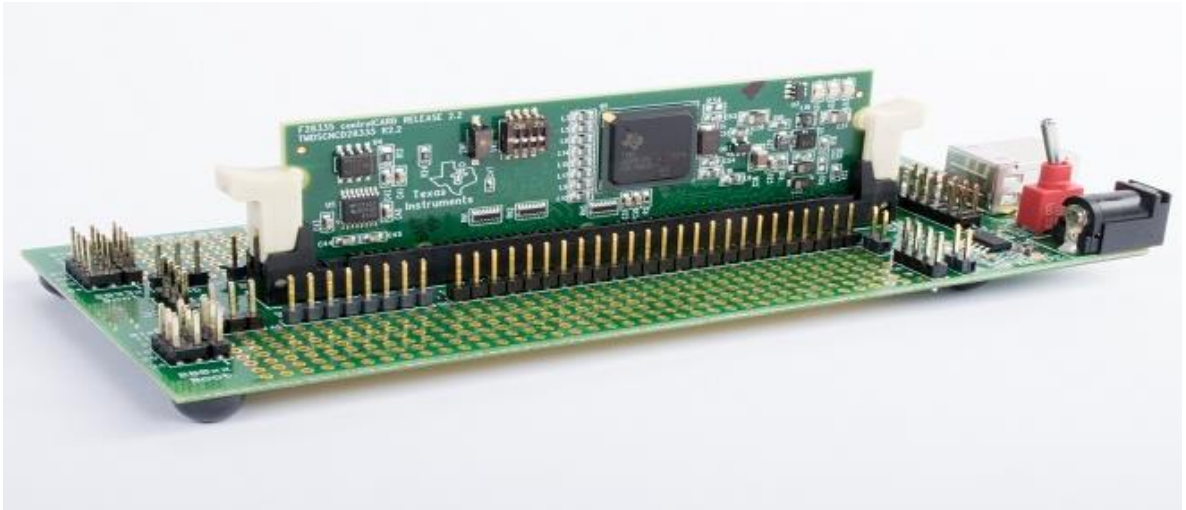


Figura 5.1 Experimenter kit

Fuente: Texas Instruments

Este procesador contiene módulos y características principales en listadas a continuación:

- ❖ Frecuencia máxima de funcionamiento 150 MHz.
- ❖ Periféricos utilizados en el proyecto:
 - GPIO, Puerto de propósito general el cual contiene entradas y salidas.
 - PWM, modulador por ancho de pulso, esta referencia cuenta con 6 canales para este propósito.
 - ADC, Conversor analógico Digital.
 - PIE, Periférico de expansión de interrupciones [8].

GPIO

La DSP tiene con tres puertos denominados A, B y C, donde los puertos A y B tienen 32 pines y puerto C cuenta con 24 pines para un total de 88 pines.

Existen 3 registros los cuales permiten una configuración de cada uno de los pines de la DSP, el primer registro se basa en una tabla de funciones para cada pin permitiendo establecerlos así:

- ❖ 00, Pin propósito general IO.
- ❖ 01, Función periférico 1.
- ❖ 10, Función periférico 2.
- ❖ 11, Función periférico 3.

Se necesitan dos bits para la configuración desempeñada por el pin, existen multiplexores por puerto por cada dos registros, la información de estos periféricos la puede encontrar en el documento técnico SPRUFB0.pdf.

El Registro de dirección permite establecer si el pin es de salida o entrada, donde los valores de 0 o 1 indican el tipo de dirección, un cero indica que es un pin de entrada y un 1 corresponde a un pin de salida, utilizando únicamente un registro por puerto.

El tercer registro configura las resistencias internas de Pull_up, disponibles para los puertos A y B, y únicamente se necesita un registro por puerto.

Los GPIO cuentan con registros los cuales actúan como cambio de estados, asignar un dato a un pin o puerto, o un forzado a cero, estos registros son: GPxSET si el pin correspondiente se configura como salida, y al escribir un 1 en ese bit este registro establece una salida en alto y el pin correspondiente será activado, si en cambio el valor del bit es 0 los registros establecidos no tiene efecto, GPxCLEAR este registro se utiliza para mantener en bajo los pines GPIO especificados sin molestar a los otros, si el pin se configura como salida de propósito general, al escribir 1 en el bit correspondiente este registro se borrara y su pin queda anclado en bajo, al escribir un 0 en cualquier bit este no tendrá efecto, GPxTOGGLE este registro ser conoce como registro de alternancia el cual conduce los pines GPIO especificados a el nivel opuesto sin molesta los demás pines, al escribir un 1 en ese bit el pin configurado

para este caso en bajo cambia su estado quedando así en alto o viceversa, si al escribir un 0 en ese bit el pin no tienen ningún efecto alguno y GPxDAT.

PIE

El DSP tiene una sola interrupción no enmascarables y 16 interrupciones enmascarables desde la INT1 hasta INT14, RTOSINT, DLOGINT. Cuenta con un periférico expandible denominado PIE (peripheral Interrupt Expansion), el cual permite obtener una expansión de las 12 primeras líneas de interrupción, agrupando por línea 8 interrupciones, obteniendo un PIE de 96 de posibilidades de interrupciones. Estas direcciones se encuentran de manera vectorizada, donde será atendida una sola vez se haya disparado sin tener toma de decisiones.

La configuración interna de estas interrupciones se encuentra en la siguiente figura:

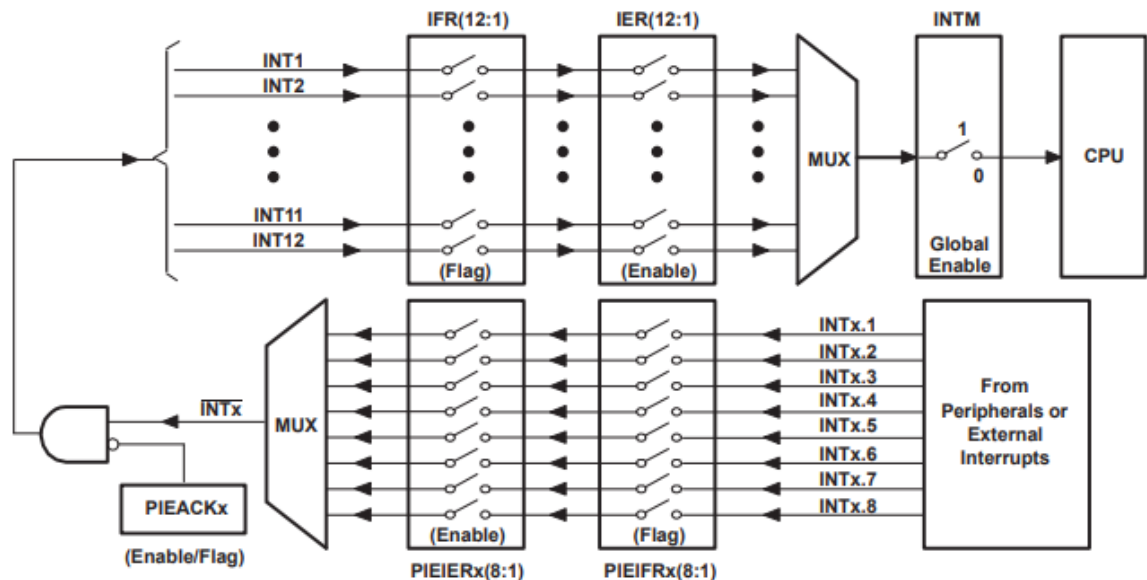


Figura 5.2 Multiplexado de interrupciones

Fuente: SPRUFB0.pdf

En la figura 5-2 se distinguen los tres registros IFR, IER e INTM, este último registro habilita globalmente todas las interrupciones que estén programadas en la DSP, el registro IER habilita por grupos de interrupciones, el registro IFR son banderas o

estados que me indican cual interrupción fue generada, una vez atendía la interrupción esta bandera o estado se borra automáticamente.

La parte inferior de la figura 6 se observan 8 líneas que agrupa una sola de las 12 líneas de interrupción del IER, estos registros de expansión que pertenecen al PIE se denominan PIEIFR, PIEIER [9].

Un ejemplo de programación realizado para este proyecto muestra la forma de utilizar las interrupciones:

```
InitPieCtrl();// Inicializa espacio de interrupciones
IER = 0x0000;// Deshabilita y borra todas las banderas de interrupción
IFR = 0x0000;
InitPieVectTable();// Inicializa la Tabla de Vectores del PIE con los punteros
para cada interrupción
EALLOW;
PieVectTable.ADCINT = &adc_isr;
PieVectTable.XINT3 = &Off_isr; // Rutina para la Interrupción de Apagado
PieVectTable.XINT1 = &WAKE_UP_isr; rutina para la Interrupción de Encendido
EDIS;
IER |= M_INT1;
IER |= M_INT3; // Habilitar interrupciones para los PWMs 1 a 3
IER |= M_INT12;
PieCtrlRegs.PIEIER1.bit.INTx4 = 1;// Habilita la interrupción externa XINT1
(Pulsador de Encendido)
PieCtrlRegs.PIEIER1.bit.INTx6 = 1; // Habilita interrupción ADC (PWM 2)
PieCtrlRegs.PIEIER3.bit.INTx3 = 1;// Habilita interrupción del PWM1
PieCtrlRegs.PIEIER12.bit.INTx1 = 1;
EINT;
```

PWM

El periférico PWM está dividido en 5 partes de la siguiente manera:

- ❖ Base de tiempos
- ❖ Comparación
- ❖ Selección de polaridad
- ❖ Banda muerta
- ❖ Detección de fallas

Cada PWM se puede configurar de manera independiente de acuerdo a las 5 etapas mencionadas anteriormente.

BASE DE TIEMPOS

La base de tiempos establece la frecuencia de funcionamiento del PWM, existen tres registros que configuran esta base, TBPRD, TBCTL, TBCTR. El registro TBPRD establece el valor máximo al que puede llegar el contador de la DSP, el registro TBCTR cuenta con 3 formas diferentes de conteo, conteo arriba, conteo abajo o conteo arriba-abajo, el tercer registro TBCTL se divide en varios campos de bits permitiendo realizar una pre - escalización de la frecuencia del sistema.

Para acceder a los registros de la maquina se manejan estructuras que el fabricante ha creado en lenguaje c, para conocer la base de tiempos se utiliza la ecuación dada por el fabricante así:

$$TBPRD = \frac{F_{DSP}}{2 * F_{PWM} * HSPCLKDIV * CLKDIV} \quad (5-1)$$

Si se desea realizar un conteo arriba-abajo se debe colocar el número 2 en el denominador, el valor del TBPRD deber ser menor o igual a 65535 ya que es un registro de 16 bits. Un ejemplo de implementación de un PWM funcionando a 20KHz es:

$$TBPRD = \frac{150000000}{2 * 20000 * 1 * 1} \quad (5-2)$$

```
EPwm3Regs.TBPRD = 3765;
EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Conteo arriba abajo
EPwm3Regs.TBCTL.bit.PRDL = TB_SHADOW; // Carga contador=0
EPwm3Regs.TBCTL.bit.CLKDIV = TB_DIV1; //Divide por 1
EPwm3Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; //Divide por 1
```

COMPARACION

El DSP cuenta con dos registros el CMPA y el CMPB, con estos se crean dos señales PWM, que pueden ser iguales o dos en complemento.

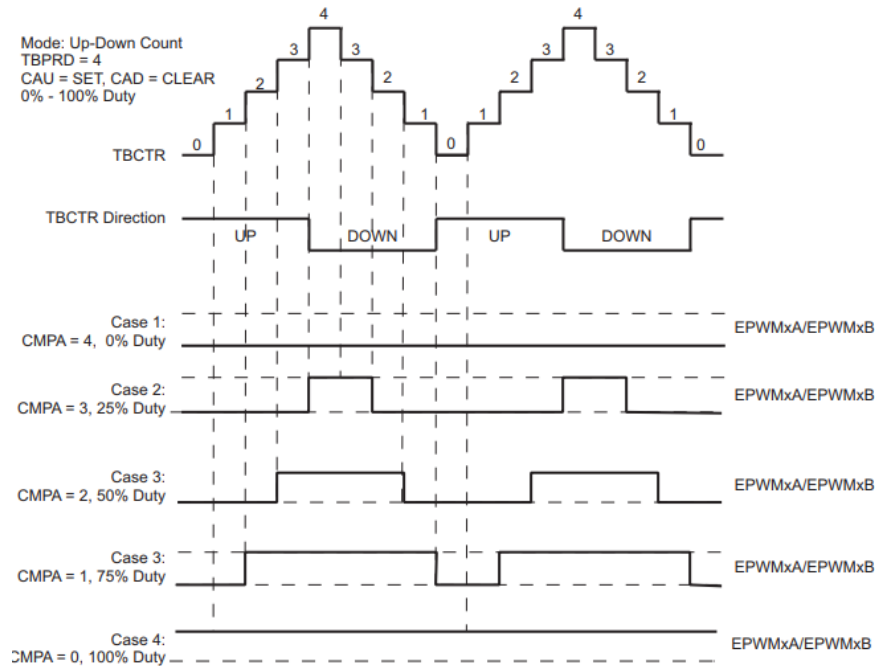


Figura 5.3 Diferentes niveles de comparación

Fuente: SPRUG04.pdf

En la Figura 5.3 se puede observar un ejemplo puesto por el fabricante con un TBPRD igual a 4 y diferentes valores que puede obtener el comparador A.

SELECCIÓN DE POLARIDAD

Cada PWM contiene dos pines, uno que contiene la señal A y otro con la señal B, cada una de ellas se puede configurar de manera independiente mediante el uso de registros de calificación AQCTLA t AQCTLB

Se puede programar eventos donde el contador llegue a cero, el periodo “PRD”, cuando el contador vaya de manera ascendente y se dé la comparación con el valor del comparador A “CAU”, si el contador cuente hacia abajo y se da una comparación con el valor del comparado A “CAD”, si el contador va de manera ascendente y se da una comparación en el comparador B “CBU”, cuando el contador va de manera

descendente y se da una comparación en el comparador B “CBD”. Siempre y cuando se tenga en cuenta las siguientes opciones:

- 00, sin cambio.
- 01, Clear, pone la señal A o B en bajo.
- 10, Set, Pone la señal A o B en alto.
- 11, Toggle, Realiza el cambio, si está en alto pasa a bajo o viceversa.

BANDA MUERTA

Si se trabaja con transistores configuración en medio puente, se necesita que las dos señales de activación provenientes de cada transistor estén en modo complementario y tenga un tiempo entre el cambio de las dos señales. La figuración mostrada a continuación presenta la distribución interna del módulo, donde se puede observar que existe una etapa de polarización en su entrada, contiene dos contadores de retardo, el cual se encargan de detectar un flanco de subida o un flanco de bajada, la polarización de la señal de salida y un interruptor el cual habilita o no el módulo.

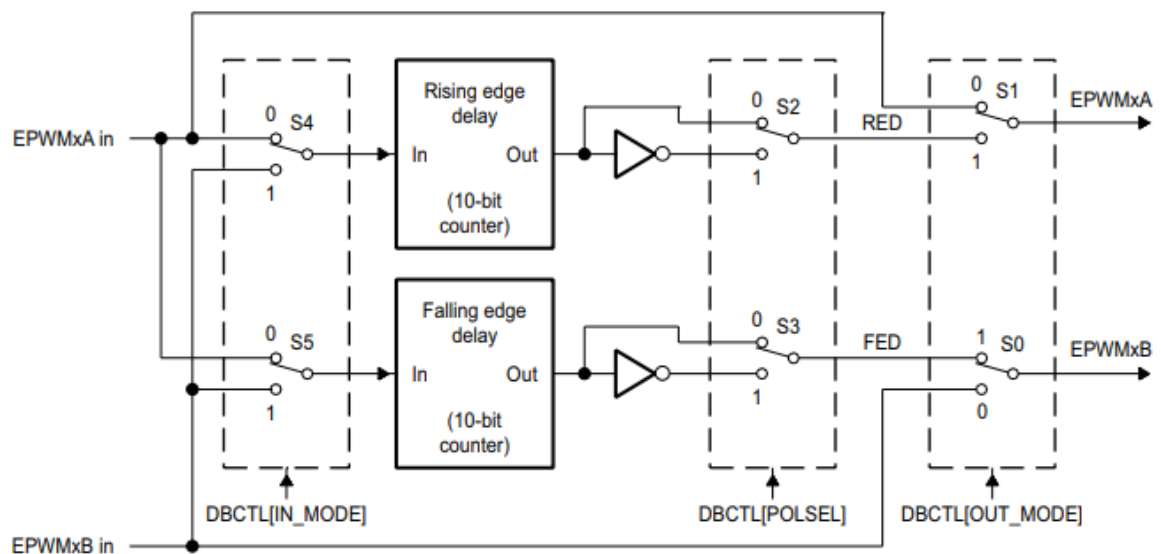


Figura 5.4 Modulo banda muerta

Fuente: SPRUG04.pdf

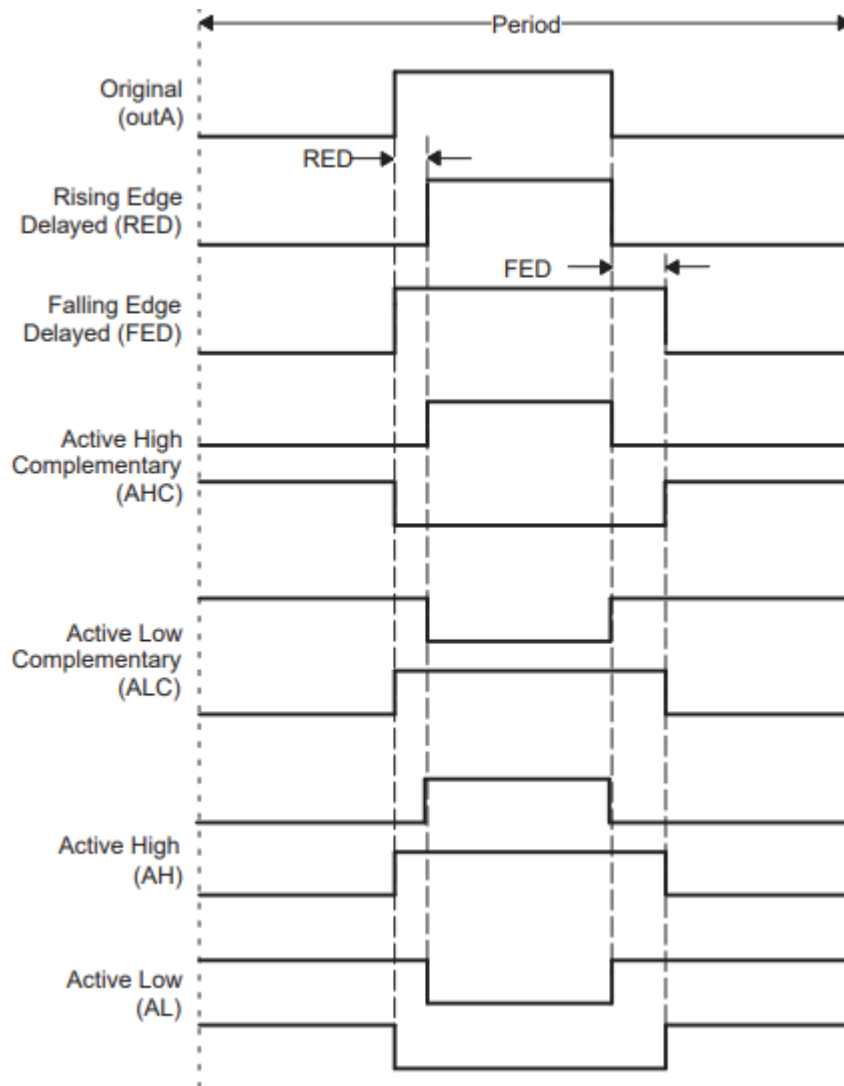


Figura 5.5 formas de onda banda muerta para casos típicos

Fuente: SPRUG04.pdf

DETECCIÓN DE FALLAS

El DSP cuenta con pines para este propósito, si se configuran y habilitan pueden ejercer una acción programada por el usuario sobre el comportamiento del PWM o tener la capacidad de generar una interrupción.

Para que una señal tenga validez, es decir, indique al procesador que debe realizar alguna acción sobre el pin, esta debe tener un flanco de bajada mediante la implementación de un hardware externo, normalmente un comparador, que tiene

como referencia un voltaje correspondiente a la variable que se desea limitar y para la cual se quiere detectar la falla.

Existen dos modos de funcionamiento para este módulo, el primero actúa en un solo ciclo o periodo de muestreo y el otro en modo lo realiza ciclo por ciclo. Cualquiera de estos dos métodos puede contener las siguientes opciones:

- 00, Forzar a alta impedancia el pin A o B.
- 01, Forzar a estado alto el pin A o B.
- 10, Forzar en bajo el pin A o B.
- 11, no realizar ninguna acción [10].

Las funciones las cuales cumple el Trip zone son:

- Los disparos de TZ1 hasta TZ6 pueden ser asignadas flexiblemente a cualquier ePWM.
- Soporte de disparo único para cortocircuitos importantes o condiciones de sobre corriente.
- Soporte de ciclo por ciclo para limitación de corriente.
- Admite disparo forzado por software.

ADC

El conversor analógico digital está conformado por 16 canales divididos en dos multiplexores de 8 canales cada uno, cada multiplexor cuenta con una unidad de "Sample and hold" y la salida de cada uno de estos es multiplexada nuevamente para tener una única señal que ingresa al ADC.

Este ADC puede ser configurado de manera secuencial o simultáneo, esto quiere decir que puede ser programado de para realizar secuencialmente hasta 16 conversiones o también puede muestrear de manera simultánea dos canales, esta opción es bastante útil si se quiere trabajar con señales diferenciales, o tomar muestras de dos variables simultáneamente; de igual manera en esta configuración, el módulo del ADC puede tener la posibilidad de realizar conversiones en parejas

de canales y además secuencialmente, solo que ahora la secuencia sería solo de ocho canales.

Se debe tener en cuenta el reloj para el funcionamiento del ADC trabaja linealmente a 12.5MHz como máximo, esto quiere decir un tiempo de conversión es de 80ns. A continuación, se muestra la estructura del ADC cuando está configurado en modo cascada.

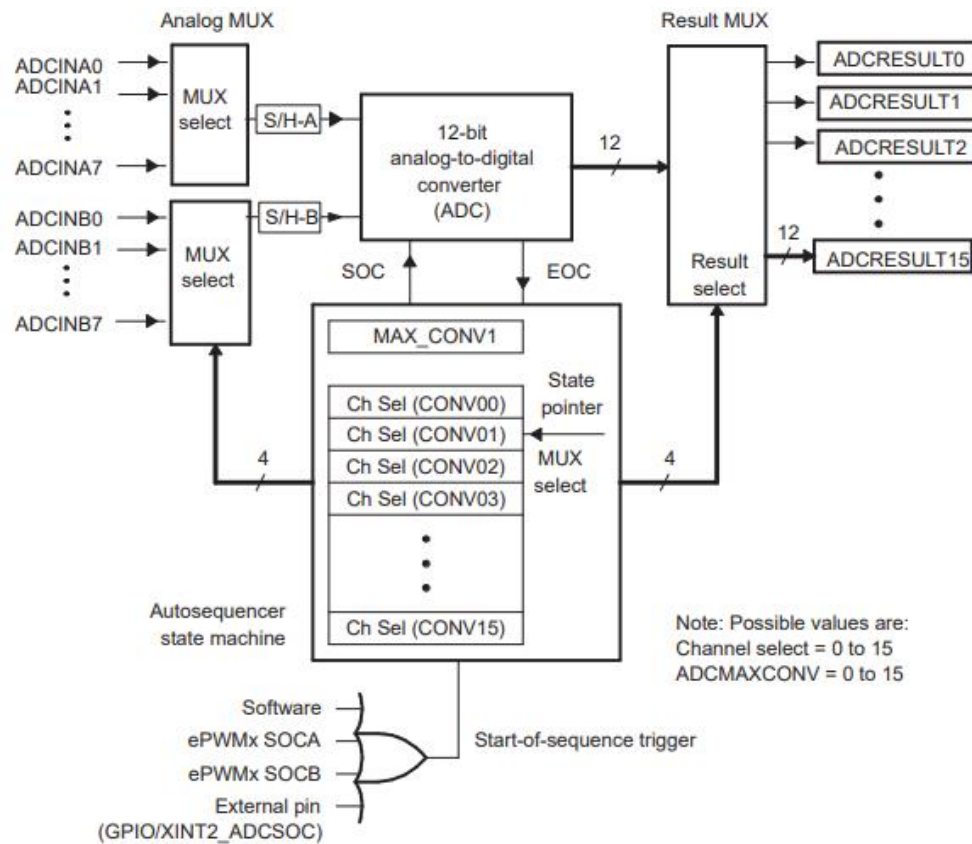


Figura 5.6 Configuración ADC modo cascada.

Fuente: SPRU812.pdf

Cabe resaltar que la frecuencia a la cual opera el ADC es diferente a la frecuencia de muestreo, que usualmente es fijada por la conmutación del PWM. En este proyecto se utilizó la configuración simultánea por que la tarjeta encargada de sensar las variables cuenta con salidas diferenciales, en el código, solo se debe

restar el resultado de cada pareja de registros y escalizarlos para obtener el valor de cada variable.[11]

Capítulo 6 INVERSOR IMPLEMENTADO

Para el diseño de esta etapa del proyecto se tiene la posibilidad de implementarlo de dos maneras, la primera es utilizar módulos inteligentes o de manera discreta, para este caso se optó realizarlo de manera discreta, donde se utilizan 6 transistores, esto implica un montaje de mayor tamaño, para cada transistor un driver, en definitiva, un poco más complejo, pero permite la reparación individual en caso de falla.

Esta configuración utiliza 3 ramas cada una con 2 transistores para obtener 3 salidas que se pueden configurar a manera de estrella o delta dependiendo el diseño.

MODELAMIENTO DEL CIRCUITO INVERSOR

Se realiza un análisis para cada tiempo correspondiente al circuito donde se analizará para el tiempo uno qué relación tiene el ciclo útil (D) con la tensión de salida y para el tiempo dos como influye $1-D$ para la tensión de salida.

Sabiendo esto se procede analizar el plano del inversor planteando las ecuaciones del sistema de esta manera:

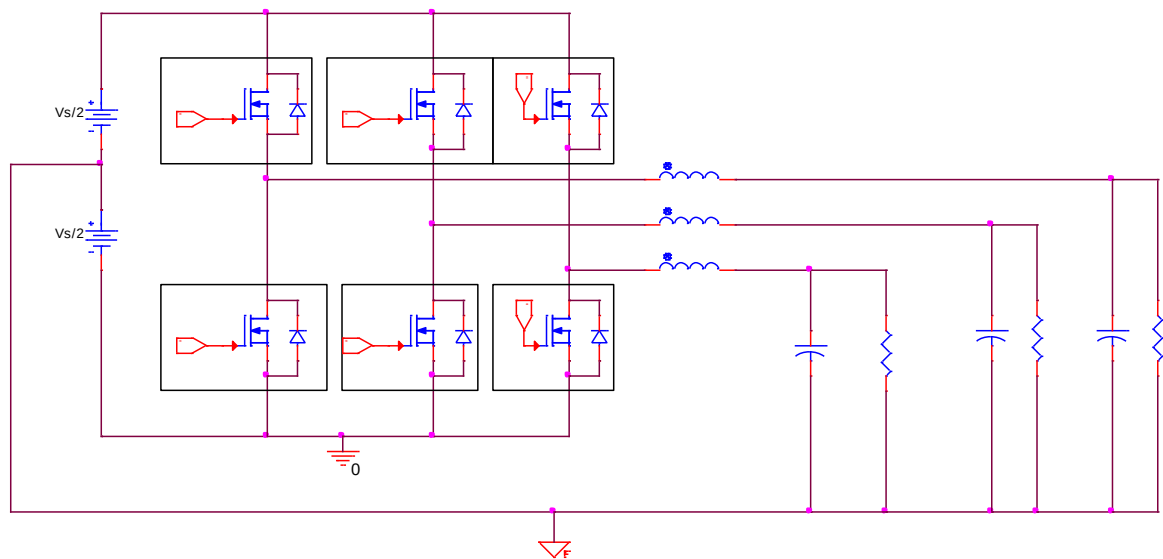


Figura 6.1 plano general del inversor

Fuente: Autor.

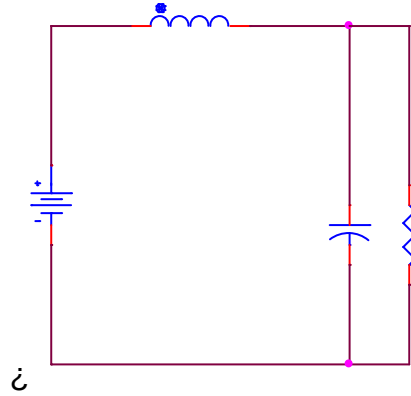


Figura 6.2 Análisis del circuito

Fuente: Autor.

Con la figura 6.2 se encuentran las ecuaciones del sistema para la tensión en un primer tiempo, se tiene en cuenta que este análisis se realiza para una sola línea y esta se duplica para las otras dos restantes, las ecuaciones están dados por:

$$i_L(t) = i_C(t) + i_R(t) \quad (6.1)$$

$$i_L = C \frac{dv_0}{dt} + \frac{v_0}{R} \quad (6.2)$$

$$C \frac{dv_0}{dt} = i_L - \frac{v_0}{R} \quad (6.3)$$

$$\frac{dv_0}{dt} = \frac{i_L}{C} - \frac{v_0}{RC} \quad (6.4)$$

Para el tiempo 1 el cual corresponde al transistor superior activo, las ecuaciones del sistema para la corriente, son dadas por:

$$\frac{V_s}{2} = V_L + V_0 \quad (6.5)$$

$$\frac{di}{dt} = \frac{V_s}{2 * L} - \frac{V_0}{L} \quad (6.6)$$

Para el tiempo 2 las ecuaciones de la tensión el cual corresponde al transistor superior activo, son dadas por:

$$I_L = I_C + I_R \quad (6.7)$$

$$I_L = C \frac{dV_0}{dt} + \frac{V_0}{R} \quad (6.8)$$

$$C \frac{dV_0}{dt} = I_L - \frac{V_0}{R} \quad (6.9)$$

$$\frac{dV_0}{dt} = \frac{I_L}{C} - \frac{V_0}{R * C} \quad (6.10)$$

Para el tiempo 2 las ecuaciones de corriente el cual corresponde al transistor superior activo, son dadas por:

$$\frac{V_s}{2} + V_L + V_0 = 0 \quad (6.11)$$

$$V_L = -\frac{V_s}{2} - V_0 \quad (6.12)$$

$$L \frac{di}{dt} = -\frac{V_s}{2} - V_0 \quad (6.13)$$

$$\frac{dI}{dt} = -\frac{V_s}{2 * L} - \frac{V_0}{L} \quad (6.14)$$

Ya obtenidas las ecuaciones para la tensión y corriente en ambos tiempos se debe realizar la multiplicación por D, ya que este es el ciclo útil con el cual se tendrá relación la tensión de salida.

$$\frac{di}{dt} = \left(\frac{V_s}{2 * L} - \frac{V_0}{L} \right) * D + \left(-\frac{V_s}{2 * L} - \frac{V_0}{L} \right) * (1 - D) \quad (6.15)$$

$$\frac{dVo}{dt} = \left(\frac{I_L}{C} - \frac{V_0}{RC} \right) * D + \left(\frac{I_L}{C} - \frac{V_0}{RC} \right) * (1 - D) \quad (6.16)$$

Separando para la ecuación (6.15):

$$\frac{di}{dt} = \frac{V_s}{2 * L} * D - \frac{V_o}{L} * D - \frac{V_s}{2 * L} - \frac{V_o}{L} + \frac{V_s}{2 * L} * D + \frac{V_o}{L} * D \quad (6.17)$$

$$\frac{di}{dt} = \frac{2V_s}{2L} * D - \frac{V_s}{2 * L} - \frac{V_o}{L} \quad (6.18)$$

$$\frac{di}{dt} = \frac{V_s}{2L} * (2D - 1) - \frac{V_o}{L} \quad (6.19)$$

Separando para la ecuación (6.16):

$$\frac{dVo}{dt} = \left(\frac{I_L}{C} - \frac{V_0}{RC} \right) * D + \left(\frac{I_L}{C} - \frac{V_0}{RC} \right) * (1 - D) \quad (6.20)$$

$$\frac{dVo}{dt} = \frac{I_L}{C} * D - \frac{V_0}{R * C} * D + \frac{I_L}{C} - \frac{V_0}{R * C} - \frac{I_L}{C} * D + \frac{V_0}{R * C} * D \quad (6.21)$$

$$\frac{dVo}{dt} = \frac{I_L}{C} - \frac{V_0}{R * C} \quad (6.22)$$

Es necesario linealizar el sistema, para ellos se usa el método de excitación de variables, donde obtendremos las siguientes ecuaciones

$$I_L = I_L + i_L \quad (6.23)$$

$$V_0 = V_0 + v_0 \quad (6.24)$$

$$V_s = V_s + v_s \quad (6.25)$$

$$D = D + d \quad (6.25)$$

Donde las letras en mayúsculas representan el punto de operación de manera constante o lineal, mientras que las letras en minúscula representan una pequeña variación o dinámica en el sistema.

Utilizando las ecuaciones (6.19), (6.22) las cuales permiten realizar el método de excitación de variables para obtener los modelos de señal grande y señal pequeña para conocer cada una de las tensiones de salida y corrientes:

Se realiza el reemplazo de las variables que deben ser excitadas en las ecuaciones encontradas para la tensión y la corriente respectivamente

$$\frac{d(I_L + i_L)}{dt} = \frac{(V_S + v_S)}{2L} * (2(D + d) - 1) - \frac{(V_O + v_O)}{L} \quad (6.26)$$

$$\frac{dI_L}{dt} + \frac{di_L}{dt} = \frac{2V_S}{2L} * d + \frac{2V_S}{2L} * D + \frac{2v_S}{2L} * D + \frac{2v_S}{2L} * d - \frac{V_S}{2L} - \frac{v_S}{2L} - \frac{V_O}{L} - \frac{v_O}{L} \quad (6.27)$$

$$\frac{dI_L}{dt} + \frac{di_L}{dt} = \frac{V_S}{L} * d + \frac{V_S}{L} * D + \frac{v_S}{L} * D - \frac{V_S}{2L} - \frac{v_S}{2L} - \frac{V_O}{L} - \frac{v_O}{L} \quad (6.28)$$

$$\frac{dI_L}{dt} + \frac{di_L}{dt} = \frac{V_S}{L} * d + \frac{V_S}{2L} * (2D - 1) + \frac{v_S}{L} * D - \frac{v_S}{2L} - \frac{V_O}{L} - \frac{v_O}{L} \quad (6.29)$$

$$\frac{d(V_O + v_O)}{dt} = \frac{I_L + i_L}{C} - \frac{(V_O + v_O)}{R * C} \quad (6.30)$$

$$\frac{dV_O}{dt} + \frac{dv_O}{dt} = \frac{I_L}{C} + \frac{i_L}{C} - \frac{V_O}{RC} - \frac{v_O}{RC} \quad (6.31)$$

Una vez se tiene las ecuaciones en términos de variables Mayúsculas y minúsculas se realiza la separación de estas dejando a un lado una ecuación en términos de variables mayúsculas y otra ecuación en término de variables minúsculas, las cuales corresponderá a los modelos de señal pequeña y señal grande.

MODELO DE SEÑAL GRANDE

El modelo de señal grande es el modelo del sistema en estado estacionario; Separando en las ecuaciones (6.27), (6.29) la variable en mayúscula se obtiene tanto para la tensión como corrientes las siguientes ecuaciones:

$$\frac{di}{dt} = \frac{Vs}{2L} * (2D - 1) - \frac{Vo}{L} \quad (6.32)$$

Se realiza el mismo procedimiento para la tensión de salida:

$$\frac{dVo}{dt} = \frac{I_L}{C} - \frac{Vo}{RC} \quad (6.33)$$

MODELO DE SEÑAL PEQUEÑA

El modelo de señal pequeña es el modelo que permite analizar la dinámica en el sistema y con este modelo se realiza el diseño de los lazos de control; La ecuación (6.29) y (6.31) realiza la separación de variables para obtener las dos ecuaciones de señal pequeña la cuales tienen como función poder obtener la función de transferencia del circuito, también son utilizadas para diseñar los lazos de control, las ecuaciones de señal pequeña que corresponden a la tensión y corriente son:

$$\frac{dI_L}{dt} = \frac{v_s}{L} * d + \frac{v_s}{2L} (2D - 1) - \frac{v_o}{L} \quad (6.34)$$

$$\frac{dvo}{dt} = \frac{i_L}{C} - \frac{v_o}{R * C} \quad (6.35)$$

Para conocer la función de transferencia de sistema se utilizan las ecuaciones (6.34), (6.35), ya que se pueden expresar de manera matricial en términos de variables de estado de la siguiente manera:

$$\vec{X} = Ax(t) + Bu(t) \quad (6.35)$$

$$y = Cx(t) + Du(t) \quad (6.36)$$

Y en forma matricial

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dv_o}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_L(t) \\ V_o(t) \end{bmatrix} + \begin{bmatrix} \frac{v_s}{L} & \frac{2D-1}{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ v_s \end{bmatrix} \quad (6.37)$$

La solución de la ecuación diferencial para las variables de estado se puede obtener utilizando el siguiente método, pero antes se debe conocer los valores de inductancia y capacitancia para después realizar los lasos de control:

$$X(s) = [sI - A]^{-1}Bu(s) \quad (6.38)$$

CÁLCULO DE LA INDUCTANCIA

$$\Delta i = \frac{\frac{V_s}{2} - 2V_s D m^2 \text{Sen}^2(wt)}{4LF_s} \quad (6.39)$$

Derivando la expresión anterior:

$$\frac{d\Delta i}{dDm} = \frac{4V_s D m \text{Sen}^2(wt)}{4LF_s} \quad (6.40)$$

Si la derivada tiene valor igual a cero entonces se dice:

$$0 = 4V_s D m \text{Sen}^2(wt); \text{ donde } Dm = 0 ; \text{Sen}^2(wt) = 0 \quad (6.41)$$

Resolviendo la ecuación anterior con los valores de $Dm=0$ y ; $\text{Sen}^2(wt)=0$, entonces:

$$\Delta i = \frac{[V_s - 2 * (0)][0.5]}{4LF_s} \quad (6.42)$$

$$\Delta i = \frac{V_s}{8LF_s} \quad (6.43)$$

La corriente se puede definir como:

$$I_s = \frac{P}{V_s * \cos(\phi)} \quad (6.44)$$

Donde la potencia para este caso son 400W, por lo tanto, por rama se tendrá aproximadamente 170W:

$$P = V_s * I_s * \cos(\phi) \quad (6.45)$$

$$170 = \frac{170}{\sqrt{2}} * I_{rms} * \cos(0) \quad (6.46)$$

Despejando I_{rms} se tiene que:

$$I_{rms} = \frac{170 * \sqrt{2}}{170} = \sqrt{2} \cong 1,4142A \quad (6.47)$$

$$I_{rms} = \frac{170}{\frac{170}{\sqrt{2}} * \cos(80)} = \frac{\sqrt{2}}{\cos(80)} = 8,14A \cong 10A \quad (6.48)$$

Una vez calculado el valor de la corriente que circulará por cada uno de los bobinados se calcula el valor de la inductancia, despejando L de la ecuación que se encuentra en la ecuación (6.43):

$$L = \frac{V_s}{8 * \Delta i * F_s} \text{ Donde } V_s = 380V \quad (6.49)$$

$F_s = 20k$, y Δi se recomienda que sea del 10%

$$L = \frac{380}{8 * 0,1 * 20k} = 23.75mH \quad (6.50)$$

CÁLCULO DE LA CAPACITANCIA

Para calcular las capacitancias del inversor se debe partir de la siguiente manera:

$$Q = CV; \text{ Donde } i = \frac{dQ}{dt} \quad (6.51)$$

Donde Q es la carga del condensador, y V es la tensión que deseamos:

$$Q = \frac{I_s}{2} * \frac{\Delta i}{2} = \frac{\Delta i}{4Fs} \quad (6.52)$$

Por otra parte, decimos que el valor de la capacitancia es la relación de la carga dividido dos veces el delta de la tensión:

$$C = \frac{Q}{2 * \Delta V} \quad (6.53)$$

Conociendo a que equivale Q se realiza el cambio de variable en la ecuación para poder conocer el valor de la capacitancia.

$$C = \frac{\frac{\Delta i}{4 * Fs}}{2 * \Delta V} = \frac{\Delta i}{8 * Fs * \Delta V}; \text{ Donde } \Delta i = 10\% Fs = 20k \text{ Hz} \quad (6.54)$$

$$C = \frac{0,1}{8 * 20k} = 625nF \quad (6.55)$$

Una vez calculados los valores de inductancia y capacitancia se procede a obtener la función de transferencia del sistema para realizar el diseño correspondiente a el lazo de control. Este procedimiento se realiza con ayuda del software Matlab obteniendo así la función de trasferencia [12].

La función de transferencia obtenida para el sistema es:

$$G_s = \frac{2.363 \times 10^{10}}{s^2 + 14.93 * s + 6.21 \times 10^7} \quad (6.56)$$

Con la ecuación 7.56 puede realizar el diseño de los lazos de control para tiempo continuo, para este proyecto se diseñó los lazos de control en tiempo discreto, ya que este se implementó de manera digital en la “DSP”.

Capítulo 7 DISEÑO COMPENSADOR INVERSOR

El diseño del compensador utilizado para el inversor planteado se realizó por medio de una técnica conocida como realimentación de variables de estado la cual permite realizar los lazos de control de manera sencilla, este lazo de control va a depender de unos valores de ganancia encontrados los cuales permiten controlar las variables de salida y garantizar un error de estado estacionario cercano a 0 o igual a este.

CONTROL POR REALIMENTACIÓN DE ESTADOS

Una forma de implementar el control es la realimentación de estado, esta configuración se encarga de realimentar variables de estado a través de ganancias constantes en lugar de emplear controladores con trayectorias directas o realimentados. Esta configuración se puede dibujar en diagrama de bloques de la siguiente manera:

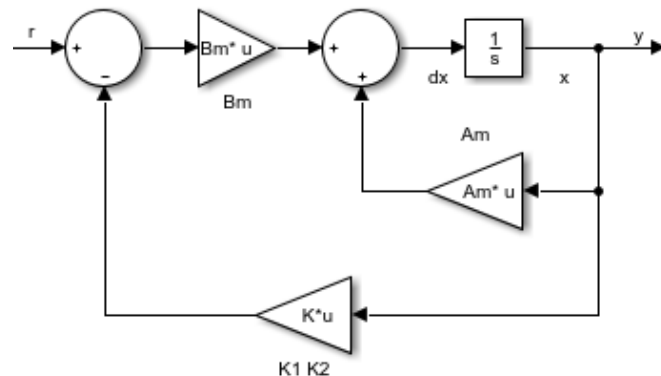


Figura 7.1 Diagrama de bloques del sistema de control mediante realimentación de estados

Fuente: Autor

DISEÑO POR UBICACIÓN DE POLOS A TRAVÉS DE REALIMENTACIÓN DE ESTADOS

Cuando se habla de diseño por ubicación de polos se tiene en cuenta que se habla del lugar geométrico de las raíces donde los polos obtenidos primeramente hacen parte de la función de transferencia en lazo cerrado donde también vienen siendo las raíces de la ecuación característica.

El control mediante la realimentación de estados es:

$$\frac{dx}{dt} = (A - BK)x(t) + Br(t) \quad (7.1)$$

Donde K es una matriz de realimentación con ganancias constantes, que permiten controlar las variables deseadas del sistema, donde la matriz k se puede expresar de la siguiente manera:

$$K = [k_1 \ k_2 \ \dots \ k_n] \quad (7.2)$$

Donde $k_1, k_2, \dots, k_n$ son los valores de las ganancias constantes.

IMPLEMENTACION DEL CONTROL POR REALIMENTACIÓN DE ESTADOS

La implementación del lazo de control se implementó como primera instancia en el dominio del tiempo donde se establecieron las matrices A, B, C, D para determinar la función de transferencia en tiempo continuo de la siguiente manera:

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dv_o}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_L(t) \\ V_o(t) \end{bmatrix} + \begin{bmatrix} \frac{v_s}{L} & \frac{2D-1}{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ v_s \end{bmatrix} \quad (7.3)$$

Donde nuestras variables de estado son $i_L(t), V_o(t)$, y se desea controlar estas variables con respecto a d, para ello la matriz C, la cual es la matriz de salida deseada corresponderá a los valores de [0 1]. [12]

Estas matrices en valores numéricos se encuentran expresada de la siguiente manera:

$$A = \begin{bmatrix} 0 & -\frac{1}{24 \times 10^{-3}} \\ \frac{1}{670 \times 10^{-9}} & -\frac{1}{67 \times 10^{-3}} \end{bmatrix}, B = \begin{bmatrix} \frac{380}{24 \times 10^{-3}} & 0 \\ 0 & 0 \end{bmatrix}, C = [0 \ 1], D = [0 \ 0] \quad (7.3)$$

Teniendo ya las matrices, se obtiene la función de transferencia en el dominio del tiempo:

$$G_s = \frac{2.363 \times 10^{10}}{s^2 + 14.93 * s + 6.21 \times 10^7} \quad (7.4)$$

Una vez obtenida la función de transferencia debemos obtener los polos de la ecuación 7.4 para determinar en qué punto se desea poner los nuevos polos para el sistema, los polos de la ecuación 8-4 son:

$$s_1 = -7,5 + 7886i, s_2 = -7,5 - 7886i \quad (7.5)$$

La ecuación 7.5 son los polos o raíces obtenidos de la función de transferencia, se observa que estas raíces tienen la parte imaginaria mucho más grande que la parte real del sistema, como consecuencia de esto el sistema tiene a tener demasiadas oscilaciones.

El sistema que se trabajó es de la forma:

$$G = \frac{W_n^2}{s^2 + 2\zeta W_n * s + W_n^2} \quad (7.6)$$

Con el denominador de la ecuación 8.4 se comienza todo el diseño del lazo de control, por ende, se estableció un valor tanto para Zeta como el tiempo de establecimiento del sistema, para conocer el ancho de banda deseado

$$W_n = \frac{5}{\zeta * T_s}; \text{ donde } \zeta = 0.7 \text{ y } T_s = 500\mu s \quad (7.7)$$

obtenido el ancho de banda deseado, se debe obtener lo que se conoce como ecuación objetivo, donde nuestro sistema debe comportarse a dicha ecuación para lograr tener control de ella.

Esta ecuación objetivo corresponde a la ecuación 7.6, y con ella debemos hallar los valores de la matriz K de la ecuación 7.1, los valores de K se obtienen a partir del polinomio generado al utilizar la siguiente ecuación:

$$\dot{X} = \det[SI - [A - BK]] \quad (7.8)$$

Obtenido el polinomio y con ayuda de Matlab, se utilizó el comando acker, el cual calcula los valores de K para dar solución a la ecuación (A-BK), las ganancias obtenidas para la corriente y la tensión o matriz K son:

$$K = [1.2622 \ 0.0060] \quad (7.9)$$

Obtenidos los valores para k1 y k2 se garantiza que el sistema se estabilice en el tiempo deseado por el diseño y siga su señal de referencia, pero aún no se garantiza un error de estado estacionario cercano a cero o igual a cero, para ello se debe agregar una ganancia k3, la cual será la encargada de controlar el error en estado estacionario de nuestra variable a controlar en este caso la tensión de salida, para ello se debe realizar una expansión matricial, para obtener este error el cual es resultado de la resta entre la referencia del sistema y la salida de este como se muestra en el diagrama de bloques:

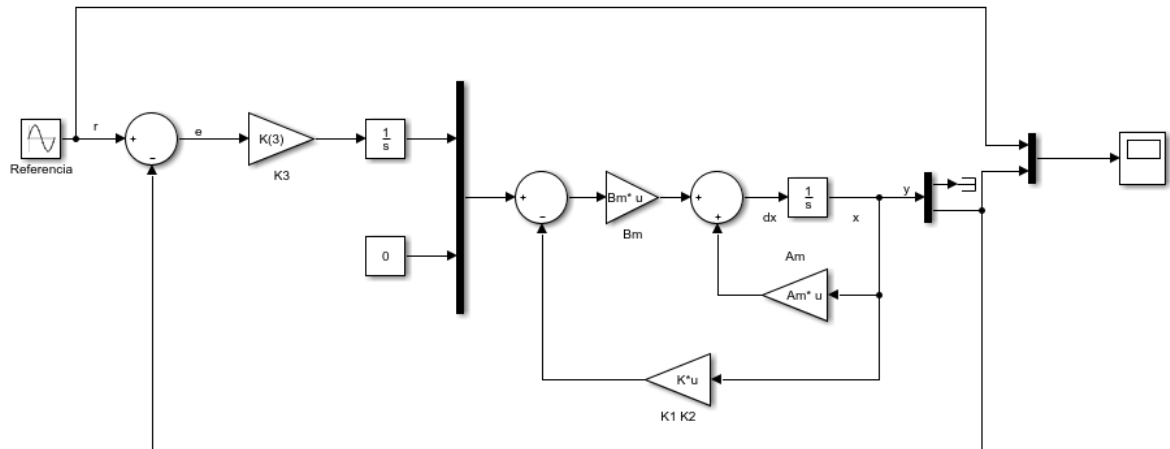


Figura 7.2 Diagrama de bloques total del sistema de control mediante realimentación de estados

Fuente: Autor

Para obtener el valor de K3 las matrices A y B quedan de la siguiente manera:

$$A = \begin{bmatrix} 0 & -\frac{1}{24 \times 10^{-3}} & 0 \\ 1 & \frac{1}{67 \times 10^{-3}} & 0 \\ 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{380}{24 \times 10^{-3}} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (7.10)$$

Utilizando nuevamente el procedimiento anterior y con ayuda del comando acker se obtiene los nuevos valores para la matriz K así:

$$K = [7.5780 \quad 0.0906 \quad 863.5875] \quad (7.11)$$

Ya con los valores de la matriz K reales se realiza la simulación del sistema con la configuración del control por realimentación de estados de esta manera:

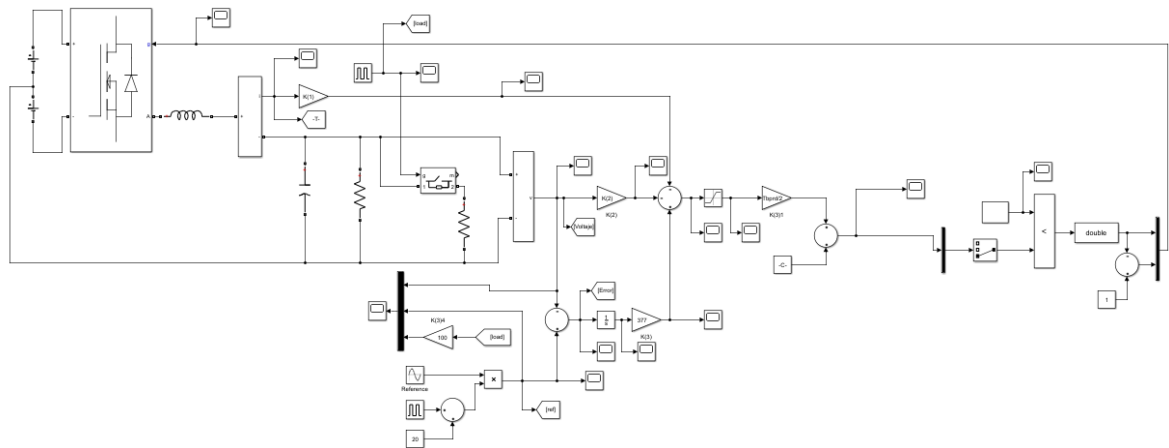


Figura 7.3 Diagrama de bloques total del sistema de control mediante realimentación de estados

Fuente: Autor

La figura 7.3 se visualiza el inversor con su respectivo lazo de control por medio de variables de estado, se tiene en cuenta que para efectos de simulación solo se realizó una rama del inversor trifásico ya que el lazo de control es el mismo para las otras dos ramas faltantes, con la diferencia que la referencia de estas están desfasadas 120 grados cada una con respecto a la rama inicial.

También se simularon posible perturbación que pueda tener el sistema en un caso real de trabajo, como resultado final de esta simulación obtenemos la siguiente gráfica:

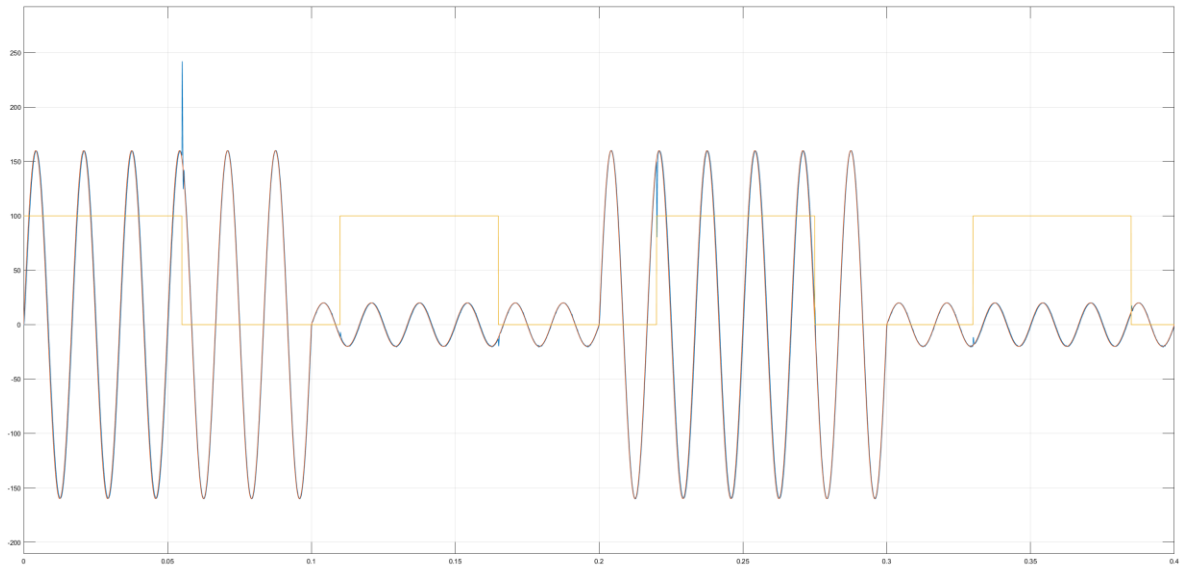


Figura 7.4 señal de referencia (Azul) Vs señal de salida (Roja) del sistema con perturbaciones y variaciones de amplitud

Fuente: Autor

IMPLEMENTACIÓN DEL CONTROL POR REALIMENTACIÓN DE ESTADOS EN TIEMPO DISCRETO

La implementación del lazo de control se debe realizar en el dominio de la frecuencia ya que este debe ser implementado de manera digital en la “DSP”. Para ello se debe transformar la matriz A y B de la ecuación 8.3 a tiempo discreto, para poder ser implementada en el “DSP”.

El método utilizado para esta transformación es por medio de integrar la multiplicación entre la inversa de Laplace obtenida de la matriz A por la matriz B original, entre 0 y el tiempo de muestreo deseado en el diseño de la siguiente manera.

$$\varphi(t) = L^{-1}[(SI - Ad)^{-1}] \quad (7.10)$$

$$\int_0^T \varphi(t - T)Bu(T)dt \quad (7.11)$$

realizado este proceso se obtiene la matriz A y B en el tiempo discreto de esta manera:

$$Ad = \begin{bmatrix} 0.9227 & -0.0020 \\ 72.9574 & 0.9219 \end{bmatrix}, Bd = \begin{bmatrix} 0.7743 & 0 \\ 29.3833 & 0 \end{bmatrix} \quad (7.12)$$

Obtenidas las Matrices A y B en tiempo discreto se procede a obtener la función de transferencia en tiempo discreto por medio de Matlab y el comando “`[numd, dend] = ss2tf(Ad, Bd, Cm, Dm, 1)`”, donde los valores para las matrices C y D siguen siendo los mismo de la ecuación 7.3.

$$Gd = \frac{29.38z + 29.38}{z^2 - 1.85z + 0.993} \quad (7.13)$$

se comienza nuevamente el diseño del lazo de control, por ende, se establece un valor tanto para Zeta, como el tiempo de establecimiento para el sistema para conocer el ancho de banda deseado

$$Wn = \frac{5}{\zeta * Ts}; \text{ donde } \zeta = 0.7 \text{ y } Ts = 500\mu s \quad (7.7)$$

Determinado el ancho de banda deseado, se debe obtener la ecuación objetivo, donde el sistema debe comportarse a dicha ecuación para lograr tener control sobre ella.

$$Eobj = \frac{0.1816z + 0.1295}{z^2 - 1.055z + 0.3664} \quad (7.14)$$

Con el denominador de la ecuación objetivo 7.14 se comienza nuevamente el diseño del lazo de control, el cual debe obtener los polos o raíces de la ecuación 8.14.

$$z_1 = 0.5275 + 0.2969i, z_2 = 0.5275 + 0.2969i \quad (7.15)$$

La ecuación 7.15 son los polos o raíces obtenidos de la función de transferencia, se debe conocer la magnitud de estas raíces para asumir un tercer polo, donde este será ubicado normalmente diez veces más grande que los polos dominantes, pero al ser discreto su valor debe ser menor a la unidad.

$$z_3 = 0.6041 \quad (7.16)$$

Ya encontrados los valores de las raíces del sistema, se hallan los valores de la nueva matriz K, los valores de K se obtienen a partir del polinomio generado al utilizar la siguiente ecuación:

$$\dot{X} = \det[SI - [Ad - Bd * K]] \quad (7.17)$$

Obtenido el polinomio y con ayuda de Matlab, se hizo el despeje matemático para dar solución a la ecuación 7.17, las ganancias obtenidas para la corriente y la tensión del vector K en tiempo discreto son:

$$K = [2.2819 \ 0.0386] \quad (7.18)$$

Con estos valores para k_1 y k_2 se garantiza que el sistema en tiempo discreto se estabilice en el tiempo deseado por el diseño y siga su señal de referencia, pero aún no se garantiza un error de estado estacionario cercano a cero o igual a cero, para ello se debe agregar una ganancia k_3 , la cual será la encargada de controlar el error en estado estacionario, para ello se realiza una expansión matricial, para obtener este error el cual es resultado de la resta entre la referencia del sistema y la salida de este como se muestra en el diagrama de bloques:

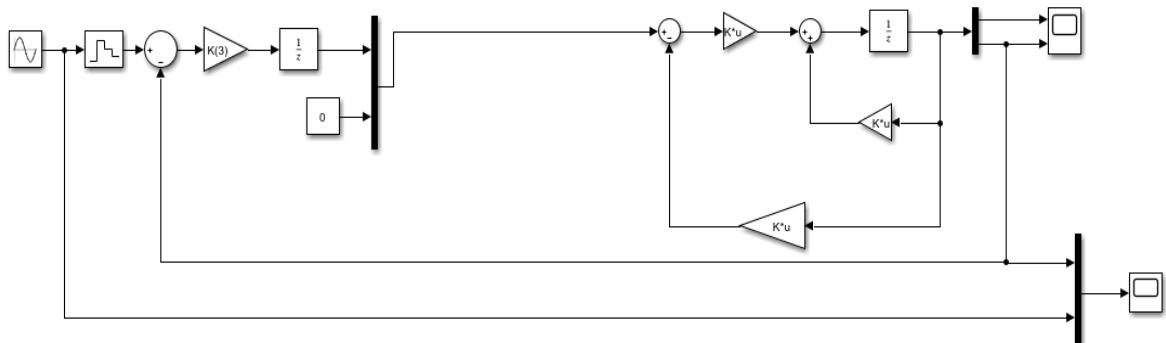


Figura 7.0.5 Diagrama de bloques total del sistema en tiempo discreto de control mediante realimentación de

Fuente: Autor

Para obtener el valor de k3 las matrices Ad y Bd quedan de la siguiente manera:

$$Ad = \begin{bmatrix} 0.9227 & -0.0020 & 0 \\ 72.9574 & 0.9219 & 0 \\ 0 & 1 & 0 \end{bmatrix}, Bd = \begin{bmatrix} 0.7743 & 0 \\ 29.3833 & 0 \\ 0 & 0 \end{bmatrix} \quad (7.19)$$

Utilizando nuevamente el procedimiento anterior, se hizo el despeje matemático para dar solución a la ecuación (8.19), para obtener la matriz K en tiempo discreto son:

$$K = [1.8635 \quad -0.0017 \quad 0.0075] \quad (7.20)$$

Ya con los valores de la matriz K discreto se realiza la simulación del sistema con la configuración del control por realimentación de estados de esta manera:

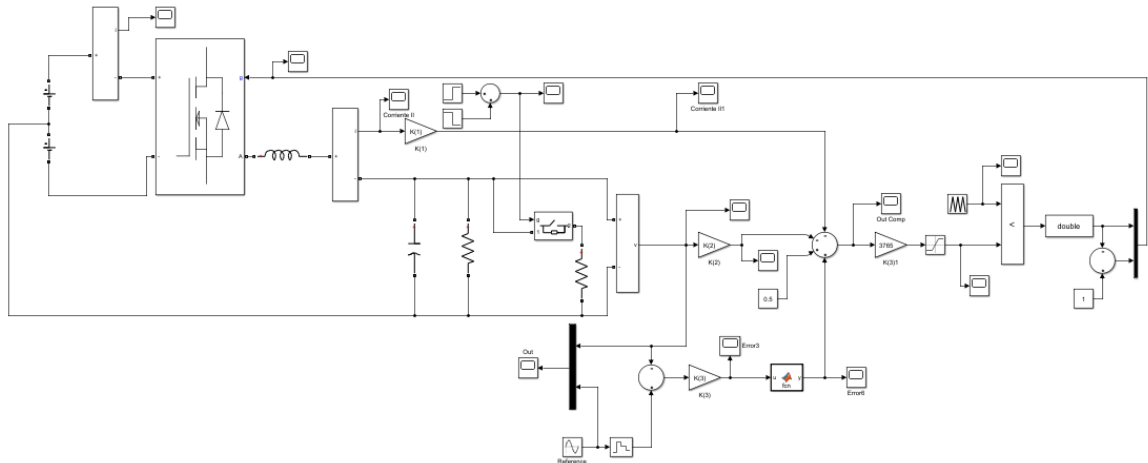


Figura 7.6 Diagrama de bloques total del sistema de control mediante realimentación de estados

Fuente: Autor

La figura 7.6 se visualiza el inversor en tiempo discreto con su respectivo lazo de control por medio de variables de estado, se debe tener en cuenta que para efectos de simulación solo se realizó una rama del inversor trifásico ya que el lazo de control es el mismo para las otras dos ramas faltantes, con la diferencia que su referencia de estas están desfasadas 120 grados cada una con respecto a la rama inicial.

También se simularon posibles perturbaciones que pueda tener el sistema en un caso real de trabajo, como resultado final de esta simulación obtenemos la siguiente gráfica:

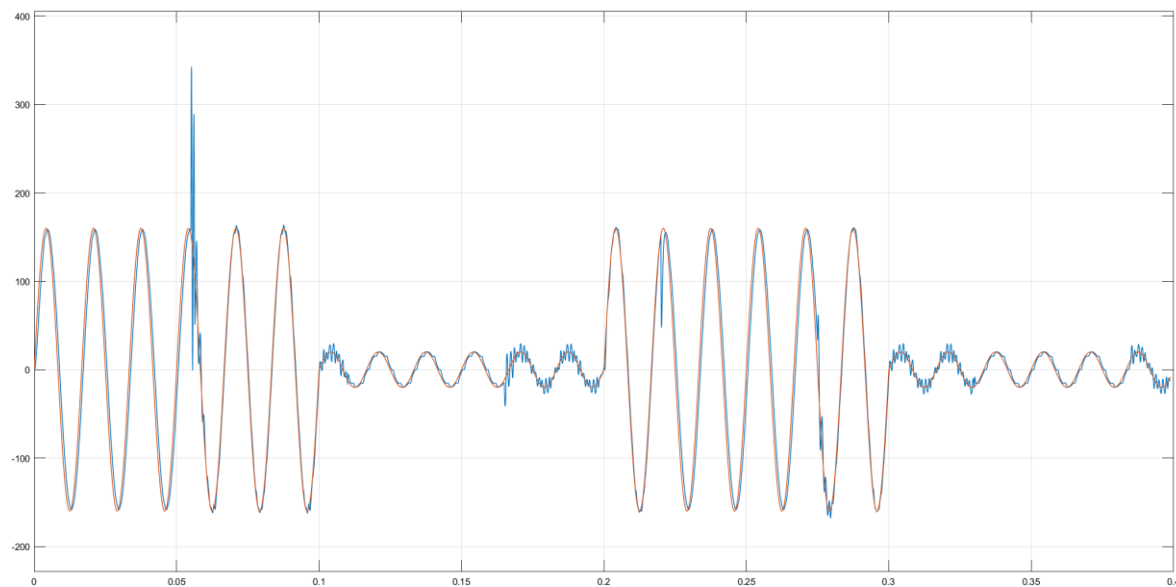


Figura 7.7 señal de referencia (roja) Vs señal de salida (azul) del sistema con perturbaciones

Fuente: Autor

Capítulo 8 ETAPA DE MEDICIÓN

Este proyecto tiene diversos puntos donde se es necesario estar monitoreando los valores medidos, el inversor tiene una etapa de medición tanto de sus tensiones de salida como corrientes de salida, llevándolas a un puerto de conversión análoga digital.

Esta etapa de medición debe tener en cuenta que las señales mencionadas anteriormente se encuentran de manera análoga, y se requiere pasarlas por una conversión, se implementó una tarjeta de medición aislada, la cual contiene tierras flotantes para cada señal y una única referencia para esta conversión.

Para utilizar los canales aislados es necesario implementar un amplificador el cual tiene doble fuente de alimentación, un aislamiento alrededor de los 4KV, con entrada y salida diferencial, la entrada puede recibir voltajes en un rango de 250mVpp y su salida es de 2Vpp sobre un nivel de offset de 1,29V, obteniendo una ganancia de 8V/V, cuenta con un ancho de banda pequeño 4 kHz, donde estará en un ambiente de trabajo con señal de 60 Hz con 50 armónicos como máximo el ancho de banda tan solo sería de 3 kHz, el cual el integrado lo soporta sin ningún esfuerzo.

En la figura 8.1 se evidencia una aplicación típica del integrado AMC1200, donde el diagrama de conexiones para la medición de corriente en una de las ramas de transistores, la salida la diseñaron para ser conectada a una DSP de la familia 2000, donde el rango de entrada de su ADC es máximo de 3V, pero es necesario amplificar por 3 la salida del integrado ya que solo entrega 1Vp, esto garantiza utilizar todo el rango del ADC. Para este proceso es necesario utilizar un THS4521, el cual es una amplificador tipo rail to rail, tanto en su entrada como salida, donde las tensiones de entrada y salida pueden llegar a estar muy cerca a los límites de polarización del integrado.

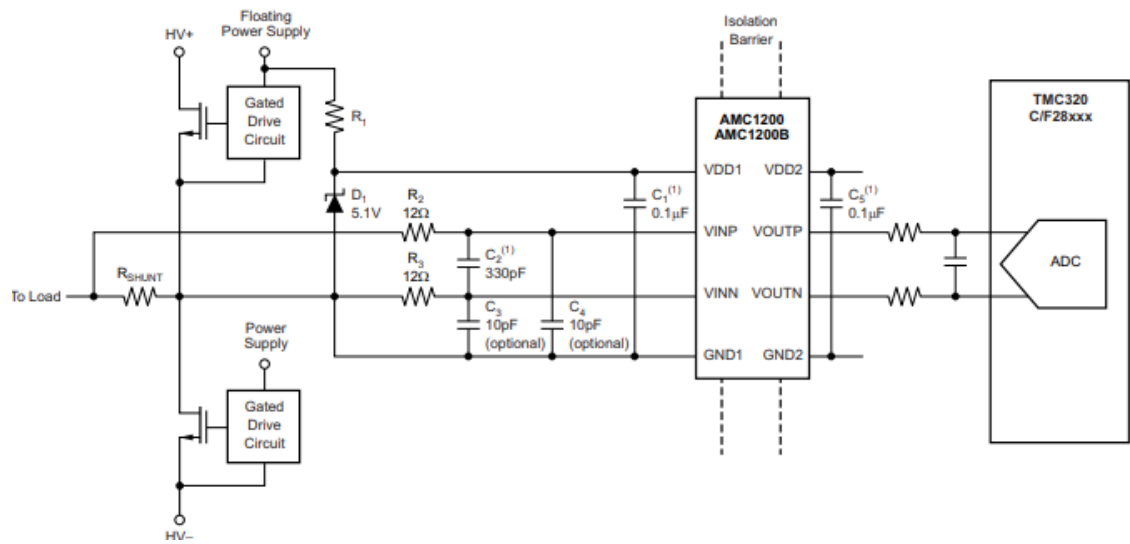


Figura 8.1 Aplicación típica

Fuente: Hoja de datos del Fabricante

Capítulo 9 RESULTADOS

El Procedimiento desarrollado para la toma de datos de este proyecto consistió en realizar diferentes pruebas al inversor, las cuales consisten en someterlo a diferentes valores de cargas resistivas por medio de un banco de trabajo el cual simula dichas resistencia, este procedimiento se realiza de manera simultánea realizando estos cambios mientras el sistema está leyendo y escribiendo a la vez, se realizaron la toma de resultados con diferentes valores de carga, con una tensión DC constante de 250V y analizando su salida.

El funcionamiento del proyecto consiste en 3 etapas las cuales son sensado de datos por medio de la tarjeta implementada para dicho fin, lectura y escritura por parte del DSP, por último, se cuenta con un potenciómetro el cual me permite realizar variación en la tensión de referencia escalada en un rango de 10V hasta 150V

RESULTADOS EXPERIMENTALES DEL INVERSOR

las figuras mostradas a continuación son los resultados experimentales del inversor sometiéndolo a diferentes variaciones en su carga, cada una de ellas le corresponde una tabla de datos las cuales expone la tensión del bus DC usado en la toma de datos, el valor de tensión AC en cada una de las salidas del inversor, la carga utilizada en ese momento, esta carga para este caso es un banco de resistencias que tiene tres salidas, cada una de estas se conectan en paralelo para dar un valor que corresponde a 171Ω por línea, este banco permite realizar variaciones en cada una de las resistencias simulando que al sistema se le inyecta más carga o se le quita dependiendo del caso, el inversor cuenta con una carga en vacío de 62Ω por línea la cual influye en la carga cuando se realiza alguna variación en ella.

En la figura 10.1 se realizó la toma de resultados con una tensión cercana los 100V pico, con carga completa (171Ω), la figura consta con las tres señales de salida del

sistema, donde se evidencia el comportamiento del control de manera correcta. La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

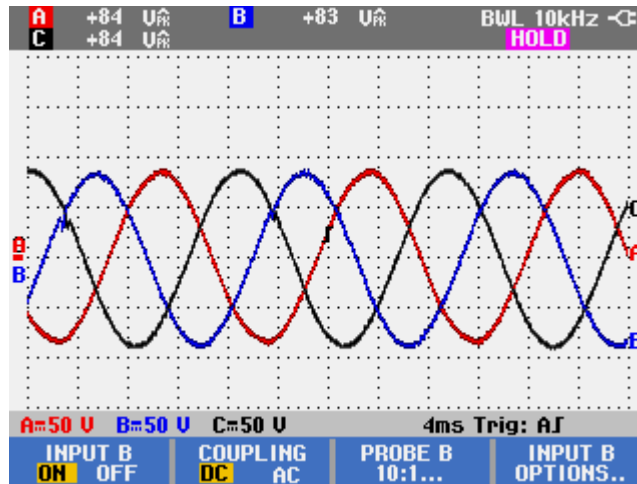


Figura 9.1 salida inversor a plena carga

Fuente: autor

1	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)
Linea A (Roja)	250	84	47
Linea B (Azul)	250	83	45
Linea C (Negra)	250	84	47

Tabla 9.1 Equivalencia de colores

Fuente: Autor

En la figura 9.2 se realizó la toma de resultados con una tensión cercana los 100V pico, con dos cargas activas (200 Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta. La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

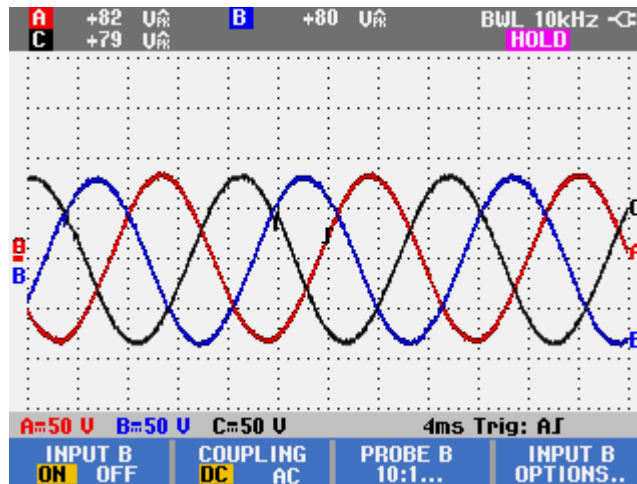


Figura 9.2 salida inversor con dos cargas activas

Fuente: autor

2	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)
Linea A (Roja)	250	82	48
Linea B (Azul)	250	80	47.8
Linea C (Negra)	250	79	47.1

Tabla 9.2 Equivalencia de colores

Fuente: Autor

En la figura 9.3 se realizó la toma de resultados con una tensión cercana los 100V pico, con una carga activa (300Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta. La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

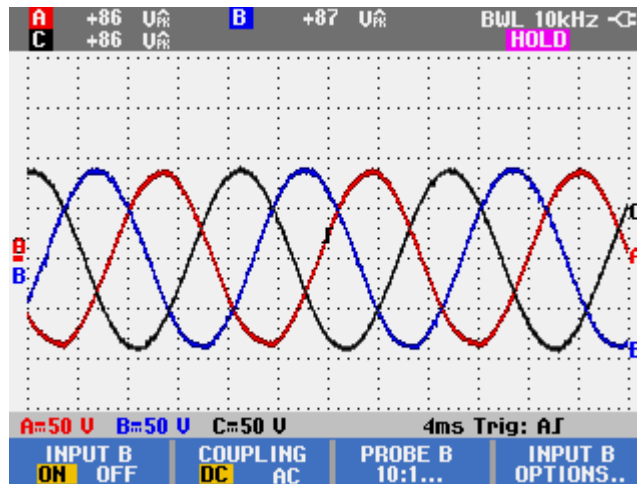


Figura 9.3 salida inversor con una carga activa

Fuente: autor

3	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)
Linea A (Roja)	250	86	52
Linea B (Azul)	250	87	52
Linea C (Negra)	250	86	52.1

Tabla 9.3 Equivalencia de colores

Fuente: Autor

En la figura 9.4 se realizó la toma de resultados con una tensión cercana los 100V pico, con carga en vacío (62Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control tiene un error en estado estacionario cuando el sistema queda completamente con carga en vacío, para este caso la disipación de potencia es mucho mayor por ende esta carga en vacío debe estar acompañada de un ventilador haciendo el sistema menos eficiente, la toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

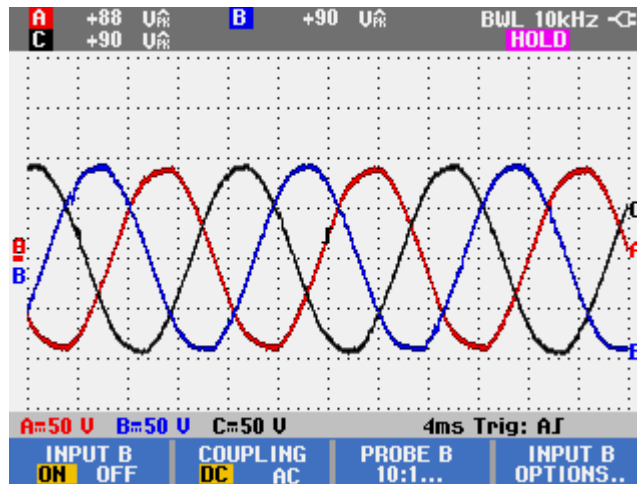


Figura 9.4 salida inversor con carga en vacío

Fuente: autor

4	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)
Linea A (Roja)	250	88	62
Linea B (Azul)	250	90	62
Linea C (Negra)	250	90	62

Tabla 9.4 Equivalencia de colores

Fuente: Autor

En la figura 9.5 se realizó la toma de resultados con una tensión cercana los 10V pico, con todas cargas activas (171Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta teniendo en cuenta que esta se realizó a muy baja tensión, también se pudo evidenciar que el sistema tiene mayor ruido debido al riple de voltaje pico generado. La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

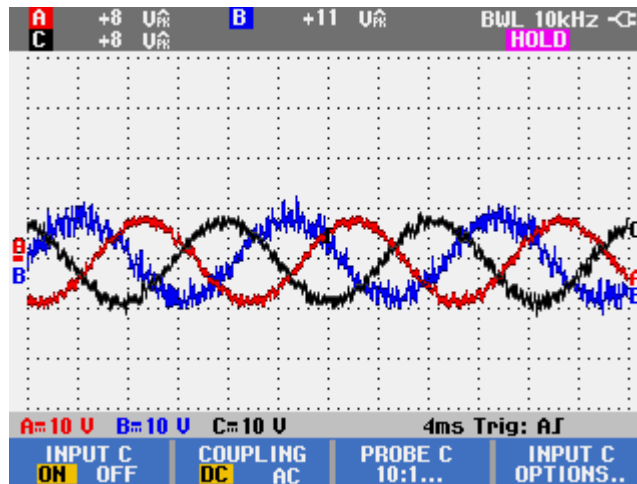


Figura 9.5 salida inversor a plena carga - tensión de referencia mínima

Fuente: autor

5	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Tension de Referencia (voltaje)
Linea A (Roja)	250	8	47	10
Linea B (Azul)	250	11	45	10
Linea C (Negra)	250	8	47	10

Tabla 9.5 Equivalencia de colores

Fuente: Autor

En la figura 9.6 se realizó la toma de resultados con una tensión cercana los 10V pico, con una carga activa (171 Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta teniendo en cuenta que esta se realizó a muy baja tensión, también se pudo evidenciar que el sistema tiene mayor ruido debido al riple de voltaje pico generado. La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

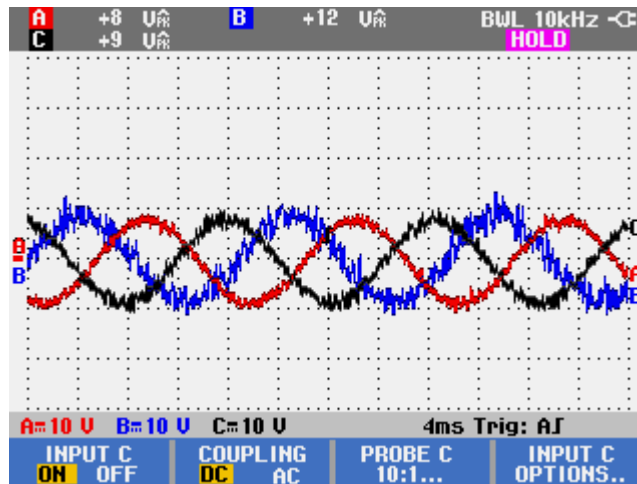


Figura 9.6 salida inversor con una carga activa - tensión de referencia mínima

Fuente: autor

6	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Tension de Referencia (voltaje)
Linea A (Roja)	250	8	48	10
Linea B (Azul)	250	12	47.8	10
Linea C (Negra)	250	9	47.1	10

Tabla 9.6 Equivalencia de colores

Fuente: Autor

En la figura 9.7 se realizó la toma de resultados con una tensión cercana los 60V pico, con toda su carga activa (171 Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta en diferentes valores de tensión de salida. La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

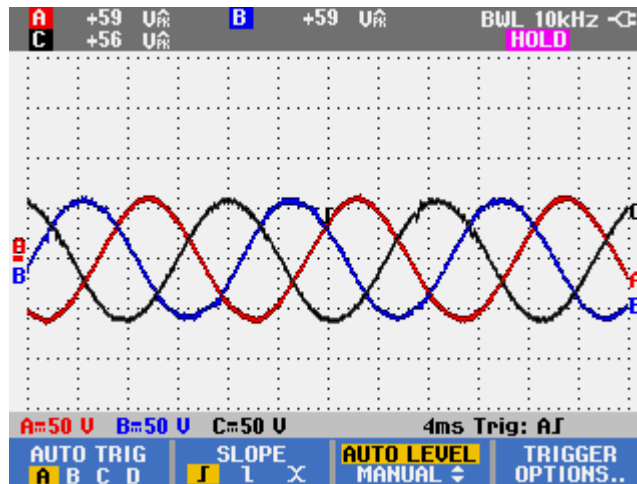


Figura 9.7 salida inversor con plena carga - tensión de referencia 70V

Fuente: autor

7	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Tension de Referencia (voltaje)
Linea A (Roja)	250	59	47	70
Linea B (Azul)	250	59	45	70
Linea C (Negra)	250	56	47	70

Tabla 9.7 Equivalencia de colores

Fuente: Autor

En la figura 9.8 se realizó la toma de resultados con una tensión cercana los 60V pico, con carga en vacío (62 Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control tiene un error en estado estacionario cuando el sistema queda completamente sin carga, para este caso la disipación de potencia es mucho mayor por ende esta carga en vacío debe estar acompañada de un ventilador haciendo el sistema menos eficiente, la toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

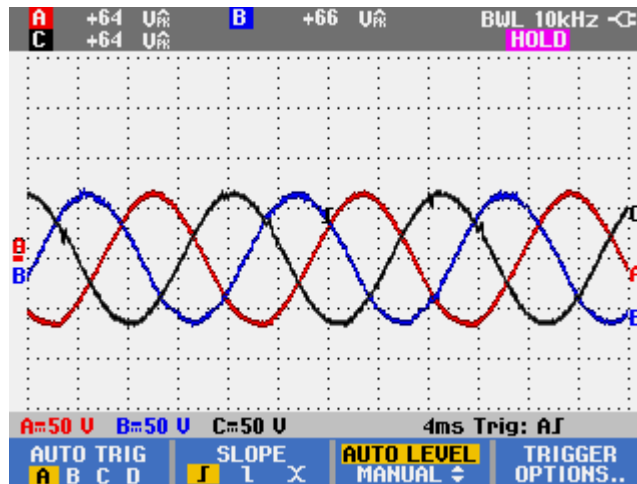


Figura 9.8 salida inversor con carga en vacío - tensión de referencia 70V

Fuente: autor

8	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Tension de Referencia (voltaje)
Linea A (Roja)	250	64	62	70
Linea B (Azul)	250	66	62	70
Linea C (Negra)	250	64	62	70

Tabla 9.8 Equivalencia de colores

Fuente: Autor

En la figura 9.9 se realizó la toma de resultados con una tensión cercana los 70V pico, la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta teniendo en cuenta que para esta toma de resultados se variaron las cargas de manera diferente donde para la línea A la carga tenía un valor de 240 Ω , para la línea B el valor de su carga era de 200 Ω y la line C tenía una carga con valor de 171 Ω . La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

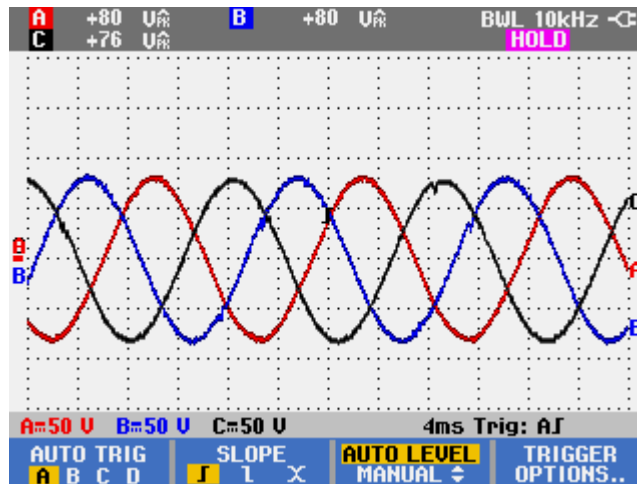


Figura 9.9 salida inversor diferencia de cargas en las líneas

Fuente: autor

9	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)
Linea A (Roja)	250	80	47.6
Linea B (Azul)	250	76	49.9
Linea C (Negra)	250	80	47.5

Tabla 9.9 Equivalencia de colores

Fuente: Autor

En la figura 9.10 se realizó la toma de resultados con una tensión cercana los 70V pico, la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta teniendo en cuenta que para esta toma de resultados se variaron las cargas de manera diferente donde para la línea A la carga tenía un valor de 171 Ω , para la línea B el valor de su carga era de 300 Ω y la line C tenía una carga con valor de 240 Ω . La toma de datos se realizó a escala de 50V con una alimentación DC de 250V.

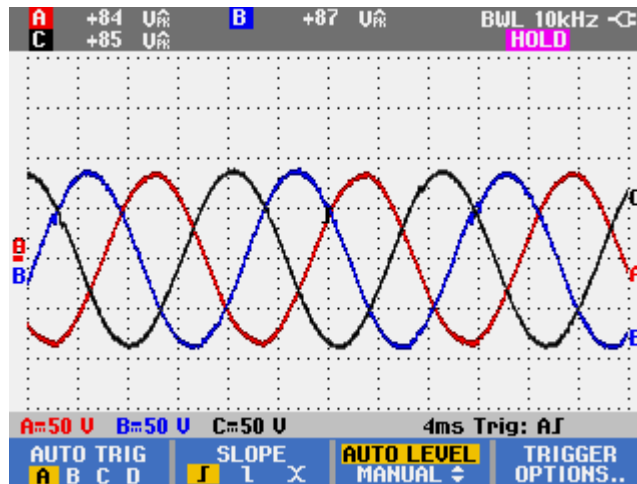


Figura 9.10 salida inversor diferencia de cargas en cada línea

Fuente: autor

10	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)
Linea A (Roja)	250	84	47.6
Linea B (Azul)	250	87	49.9
Linea C (Negra)	250	85	47.5

Tabla 9.10 Equivalencia de colores

Fuente: Autor

En la figura 9.11 se realizó la toma de resultados con una tensión cercana los 110V pico, a plena carga (171 Ω), la figura consta con las tres señales de salida del sistema, donde se evidencia el comportamiento del control de manera correcta ya que se realizó el aumento del bus DC. La toma de datos se realizó a escala de 50V con una alimentación DC de 300V.

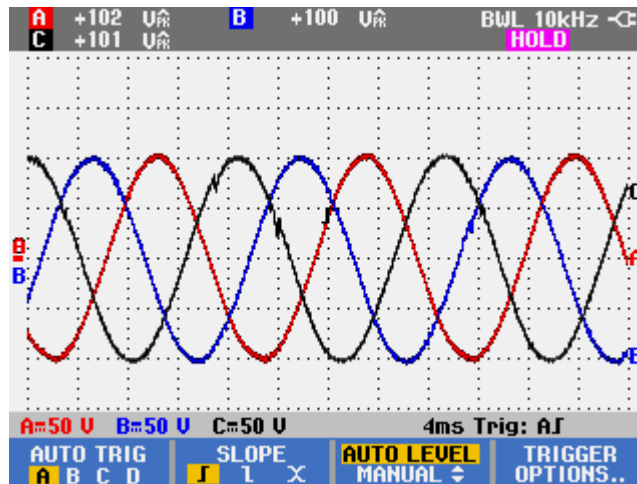


Figura 9.11 salida inversor con plena carga - tensión del bus DC 300V

Fuente: autor

11	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Tension de Referencia (voltaje)
Linea A (Roja)	300	102	47	140
Linea B (Azul)	300	100	45	140
Linea C (Negra)	300	101	47	140

Tabla 9.11 Equivalencia de colores

Fuente: Autor

Para la parte correspondiente a la medición de corriente las señales son adquiridas en por medio de una pinza amperimétrica simultáneamente la tarjeta sensora recopila los valores leídos y estos son entregados a un convertidor análogo digital donde serán tratados para poder realizar el controlador el cual necesita los valores de corriente para su desarrollo. se adquirieron los datos con plena carga, baja tensión y alta tensión.

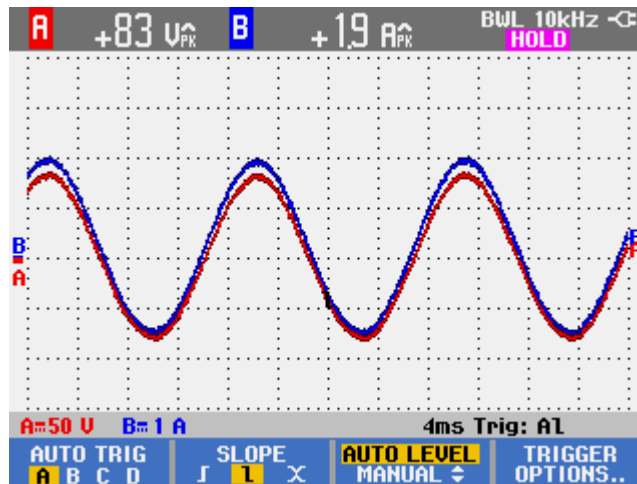


Figura 9.12 Corriente Vs alta Tensión plena carga línea A

Fuente: autor

12	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Corriente (Ampers)
Linea A (Roja)	250	83	47	1.9

Tabla 9.12 Equivalencia de colores

Fuente: autor

La figura 9.13 se puede evidenciar como la corriente se encuentra cerca a la fase de la tensión, conociendo la potencia máxima que puede entregar el equipo cuando la tensión es máxima.

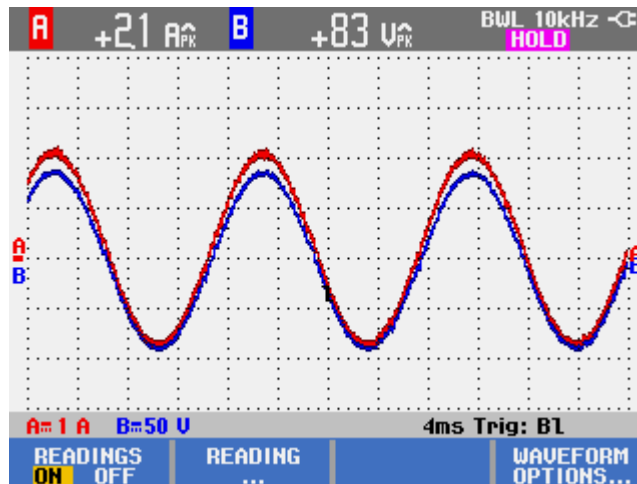


Figura 9.13 corriente Vs alta Tensión plena carga línea B

Fuente: autor

13	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Corriente (Ampers)
Linea B (Azul)	250	83	47	2.1

Tabla 9.13 Equivalencia de colores

Fuente: autor

La figura 9.14 se puede evidenciar como la corriente se encuentra cerca a la fase de la tensión, conociendo la potencia máxima que puede entregar el equipo cuando la tensión es máxima.

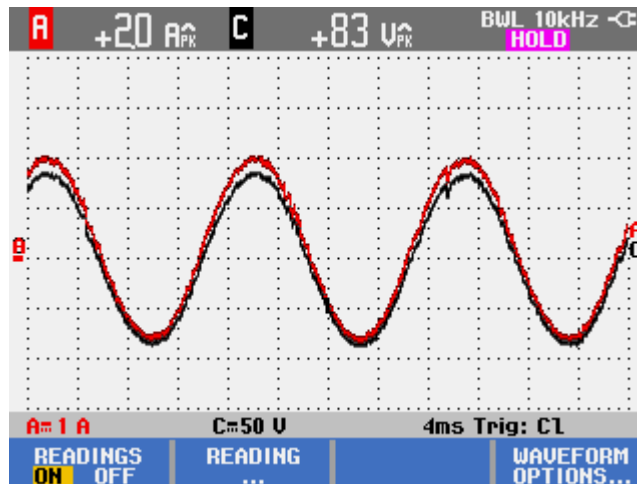


Figura 9.14 Corriente Vs alta Tensión plena carga línea C

Fuente: autor

14	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Corriente (Ampers)
Linea C (Gris)	250	83	47	2.0

Tabla 9.14 Equivalencia de colores

Fuente: autor

En la figura 9.15 se evidencia como el inversor a baja tensión se obtiene menor corriente, pero se presenta demasiado ruido en las señales, el sistema tiene un mejor comportamiento cuando trabaja en altas tensiones como se puede visualizar en las figuras 9.13 – 9.15, conociendo de esta manera la potencia activa del sistema.

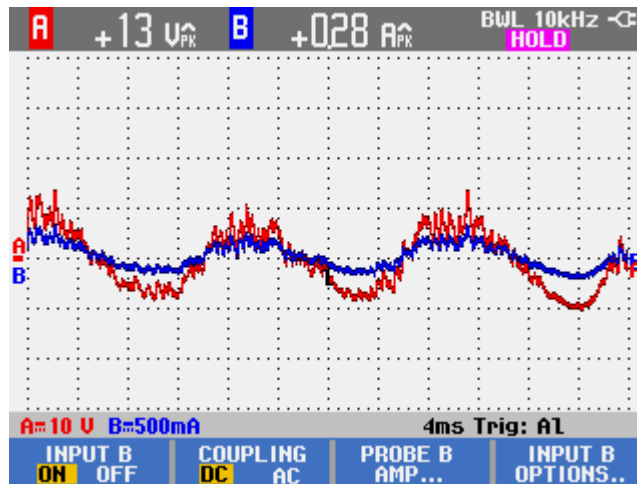


Figura 9.15 Corriente Vs baja tensión plena carga línea A

Fuente: autor

15	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Corriente (Ampers)
Linea A (Roja)	250	13	47	0.28

Tabla 9.15 Equivalencia de colores

Fuente: autor

La figura 9.16 se puede evidenciar la tensión de salida y la corriente generada por el inversor en fase con esta y en comparación a la figura 9.15 se puede ver como a medida que el sistema incrementa su tensión de salida presenta un mejor comportamiento.

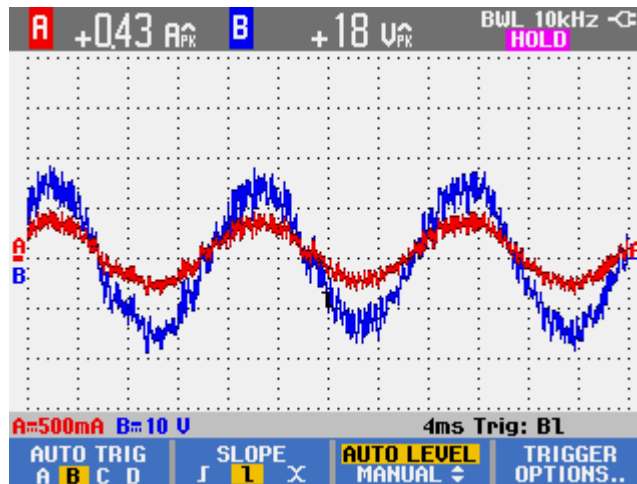


Figura 9.16 Corriente Vs baja tensión plena carga línea B

Fuente: autor

16	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Corriente (Ampers)
Linea B (azul)	250	18	47	0.18

Tabla 9.16 Equivalencia de colores

Fuente: autor

La figura 9.17 se puede evidenciar la tensión de salida y la corriente generada por el inversor en fase con esta, permitiendo conocer la potencia mínima del sistema y en comparación a la figura 9.16, nuevamente se puede evidenciar a medida que el sistema disminuye su tensión de salida va presentar más ruido en sus señales.

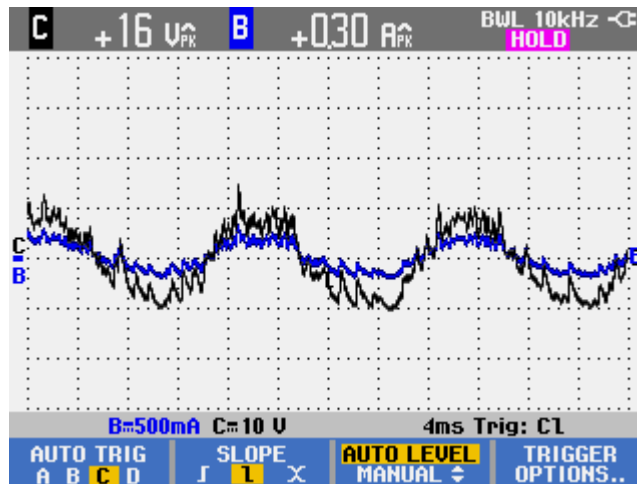


Figura 9.17 corriente Vs baja tensión plena carga línea C

Fuente: autor

17	Tension bus DC (Voltios)	Tension de Salida AC (Voltios)	carga implementada (Ω)	Corriente (Ampers)
Linea C (Gris)	250	16	47	0.30

Tabla 9.17 Equivalencia de colores

Fuente: autor

Capítulo 10 CONCLUSIONES

Garantizar una tensión AC regulada es indispensable un buen diseño de control para tener un correcto funcionamiento de todo el sistema, por ello se decide efectuar por primera vez un control por variables de estado de manera discreta, para ser implementado de forma digital en el DSP, este control permite tiene la capacidad de realimentar variables de estado a través de ganancias constantes por medio de una matriz K la cual permite controlar las variables deseadas del sistema, tomando la lectura de las variables de estado de manera simultánea.

Lograr un convertidor DC-AC funcional, requiere de un diseño el cual es primordial debido a que es el punto de partida donde cualquier error sea grande o pequeño puede causar una inestabilidad la cual conlleva a tener un equipo no funcional o nada eficiente; la implementación del convertidor es el paso a seguir después de haber corroborado que el funcionamiento del diseño por medio de una simulación, este proyecto conto con un diseño el cual lo acompaña su respectiva simulación por medio del software “Matlab”, donde permite realizar la implementación real del convertidor DC – AC, siempre en relación a la simulación realizada en el diseño.

La fuente de alimentación del bus DC posee una característica que permite subir su tensión en modo rampa hasta el valor deseado por el usuario ayudando así al que no se generen cambios abruptos en la corriente; Gracias a la implementación de un potenciómetro que permite la variación en la tensión de referencia, se garantiza que los cambios generados en la corriente no sean bruscos dado que será manipulado de manera mecánica por un ser humano reduciendo tiempos en programación de un valor a otro instantáneamente en comparación a la acción mecánica de un ser humano.

Se presentó una complicación en este proyecto ya que no se está garantizando el error en estado estacionario igual a cero por ende el equipo al quedarse en carga en vacío presenta un error cercano al 10% causando que no sea posible llegar a la

tensión pico deseada, este error se presenta debido a el desarrollo matemática en la discretización del control donde una de las ganancias no se encuentra en su valor exacto ocasionando el desborde del integrador perdiendo el control del sistema.

Principalmente el diseño del sistema se realizó con una carga en vacío alrededor de $100K\Omega$ donde esta garantiza un sistema eficiente sin demasiada disipación de potencia, al corroborar con el diagrama de bode del sistema se pudo evidenciar un pico muy grande causando que el control diseñado para dicho sistema no logre estabilizarse en ese punto, por ende se procede a utilizar una carga en vacío mucho más pequeña a la diseñada alrededor de $62\ \Omega$, para corroborar el correcto funcionamiento tanto del sistema como del control. Con este valor se logra un buen funcionamiento del inversor con la diferencia que la disipación de potencia será significativa en relación a la eficiencia del equipo.

En la etapa de sensado se tuvo en cuenta que las señales medidas se encuentran en distintos lugares y de manera análoga por ende se realizó una conversión analógica digital por medio del DSP, se utiliza una tarjeta aislada que contiene tierras flotantes para cada señal; permitiendo así tener un desacople de la parte de potencia con respecto a la de control, trabajando de manera segura con la "DSP" sin temor a un daño en esta; El ADC del DSP permite realizar la lectura de los canales diferenciales provenientes de la tarjeta sensora para cada una de las variables de tensión y corriente procesándolas y llevando estos valores a el lazo de control donde al final los valores sensados junto con la ley de control permitirán realizar la escritura hacia los canales de PWM los cuales accionaran los transistores de cada salida.

Se hizo un primer acercamiento a este tipo de inversores, donde el proyecto queda abierto a mejorar por otro estudiante que desee tocar temas en electrónica de potencia, control digital, también queda libre para ser implementado cualquier tipo de control diseñado por el usuario permitiendo ser programado en el DSP, por otra parte para trabajos a futuro se puede modificar la matemática del inversor agregando así las pérdidas presentadas por los transistores, inductancias y demás

componentes que presenten un cambio significativo al sistema, logrando así un equipo más robusto eficiente y posiblemente ser comercializado.

Capítulo 11 REFERENCIAS

- [1] Unicrom Electronica. (s.f). *Historia de los reguladores monolíticos de voltaje*. disponible: <https://unicrom.com/historia-reguladores-monoliticos-voltaje/>.
- [2] D. López, G. Camacho, J. Diaz, C. Gaviria. Modulación PWM aplicada a inversores trifásicos dentro del esquema de accionamiento eléctrico AC. Universidad del Cauca. 2007.
- [3] C. A. Torres P, C. Restrepo Patiño and D. Murillo Yarce. Diseño y construcción de un inversor trifásico. *Scientia Et Technica* (40), pp. 37-42. 2008. Disponible: <http://dialnet.unirioja.es/servlet/oaiart?codigo=4733118>.
- [4] A. Vargas Serrano. Diseño e implementación de un prototipo de inversor trifásico orientado a redes de distribución. Universidad de Chile. Abril del 2010
- [5] C. Ormaechea. *Análisis comparativo de inversores fotovoltaicos de conexión a la red con potencia igual o superior a 100kW*. Febrero del 2012.
- [6] J. Salazar. Diseño y análisis de desempeño de un inversor de voltaje utilizando controladores inteligentes. Escuela politécnica del ejército. Ecuador 2012
- [7] J. Aguirre Díaz, C. Chiquito Manso. Análisis y diseño de un puente inversor trifásico VSI controlado mediante SVM utilizando un microcontrolador. Universidad tecnología de Pereira. 2014
- [8] Texas Instruments. (2007). *TMS320F2833x, TMS320F2823x Digital Signal Controllers (DSCs)*. Technical Documents
- [9] Texas Instruments. (2007). *TMS320x2833x, 2823x System Control and Interrupts*. Reference Guide.
- [10] Texas Instruments. (2007). *TMS320x2833x, 2823x Enhanced Pulse Width Modulator (ePWM) Module*. Reference Guide
- [11] Texas Instruments. (2007). *TMS320x2833x Analog-to-Digital Converter (ADC) Module*. Reference Guide
- [12] R. Erickson, D. Maksimovic. (2004). *Fundamentals of Power Electronics*.
- [13] B. Kuo. (s.f). *Sistemas de control automático*.
- [14] Mastervolt, Solax power. Tienda inversores autoSolar y waveinn

Capitulo12 ANEXOS

PROGRAMA MATLAB

```
clc
close all
clear all
format short
syms s k1 k2 z k3

%Parametros iniciales

L = 24e-3;
C = 670e-9;
Vs = 380;
R = 100e3;
D = 0.5;
Fs = 19920;
Ts = 1/Fs;
t=Ts;
Vref = 170;

%Matrices de estados

Am = [0 -(1/L); 1/C -(1/(R*C))];
Bm = [Vs/L ((2*D)-1)/(2*L); 0 0];
Cm = [0 1];
Dm = [0 0];
Km=[k1 k2; 0 0];

[num,den] = ss2tf(Am,Bm,Cm,Dm,1);
disp('Función de transferencia del sistema:')
Gs = tf(num,den)

G=( (s*eye(2))-Am);
G1=inv(G);
G1=collect(G1);

Ad=ilaplace(G1);
G3=Ad*Bm;
Bd=int(G3,0,Ts);
disp('Matriz A discretizada')
Ad=eval(Ad)
disp('Matriz B discretizada')
Bd=eval(Bd)
[numd,dend] = ss2tf(Ad,Bd,Cm,Dm,1);
disp('Función de transferencia del sistema en tiempo discreto:')
Gd = tf(numd,dend,Ts)

Zeta=0.7;
```

```

ts=500e-6;

disp('Ancho de banda deseado para Ts = 500uS:')
Wn=5/(Zeta*ts);
disp('Ecuación objetivo:')
p=tf([Wn^2],[1 2*Zeta*Wn Wn^2]);
Eobj=c2d(p,Ts)
polos=roots([1 -1.055 0.3664])
Mag=abs(polos)
polo3=(abs(roots([1 -1.055 0.3664])))/1.0020

Kold=((z*eye(2))-(Ad-(Bd*Km)));
Kold1=det(Kold);

r=collect(Kold1);

k=solve(((6973824826194329*k1)/9007199254740992 +
(516917300695031*k2)/17592186044416 -
16614870997969199/9007199254740992)==1.055,k2);
k1=solve((-
(62766958555769020921698143215691*k1)/81129638414606681695789005144064 +
(9309616617099695435002367292233*k)/316912650057057350374175801344 +
81068873525066634389165852222539/81129638414606681695789005144064)==0.366
4,k1);
k1=eval(k1);
k2=solve(((6973824826194329*k1)/9007199254740992 +
(516917300695031*k2)/17592186044416 -
16614870997969199/9007199254740992)==1.055,k2);
k2=eval(k2);
disp('Ganancias para corriente y tensión')
K=[k1 k2]
Km=[k1 k2;0 0]
disp('comprobación de polos')
Det=((z*eye(2))-(Ad-(Bd*Km)));
Det1=det(Det)

disp('Ganancias para corriente, tensión y Error = 0')
K =
[433503141262483929236189679040218635532421131971484627599/23263465809814
7022786036671153426325116131890861612990464 -
27660559602763835798442147493944939356321021/1652968663976729308629648124
5755268582522880000 4984184147926754375/661555004228073684992]

```

CALCULO PARA EL VECTOR K EN TIEMPO DISCRETO

```

clc
close all
clear all
format short
syms Vs L C R D s z k1 k2 k3

disp('EXPANSION MATRICIAL')
Ad2=[0.9227 -0.0020 0;72.9574 0.9219 0;0 1 0]

```

```

Bd2=[0.7743 0;29.3833 0;0 0]
Km2=[k1 k2 k3;0 0 0]

% disp('EXPANSION MATRICIAL')
% Ad2=[0.9789 -0.0018 0;22.6550 0.9786 0;0 1 0]
% Bd2=[0.4615 0;5.2842 0;0 0]
% Km2=[k1 k2 k3;0 0 0]

Z2=-0.4509
Z=-0.2709255
ind=0.22134224

Knew=( (z*eye(3)) - (Ad2 - (Bd2*Km2)) );
Kdet=det(Knew);
Kdet=collect(Kdet)

K3=solve((41347187764254604201*k3)/1407374883553280000==ind,k3)
K2=solve(((41347187764254604201*k2)/1407374883553280000 -
(169895166732459310409*k1)/219902325555200000000 +
(4135331841591109*K3)/140737488355328 +
6848252716985090143/68719476736000000000)==Z,k2)
K1=solve(((7743*k1)/10000 + (4135331841591109*K2)/140737488355328 -
9223/5000)==Z2,k1)
K2=solve(((41347187764254604201*k2)/1407374883553280000 -
(169895166732459310409*K1)/219902325555200000000 +
(4135331841591109*K3)/140737488355328 +
6848252716985090143/68719476736000000000)==Z,k2)

% K3=solve((66032239*k3)/12500000 == ind,k3)
% K2=solve(((66032239*k2)/12500000 - (23056773*k1)/50000000 +
(26421*K3)/5000 + 49936527/50000000) == Z,k2)
% K1=solve(((923*k1)/2000 + (26421*K2)/5000 - 783/400) == Z2,k1)
% K2=solve(((66032239*k2)/12500000 - (23056773*K1)/50000000 +
(26421*K3)/5000 + 49936527/50000000)==Z,k2)
disp('valores K1,K2,K3')
K=[K1 K2 K3;0 0 0]

disp('comprobación del sistema')
Knew=( (z*eye(3)) - (Ad2 - (Bd2*K)) );
Kdet=det(Knew);
Kdet=collect(Kdet)

```

PROGRAMA EN LENGUAJE C IMPLEMENTANDO EN LA DSP

```

#include "DSP28x_Project.h"
#define GLOBAL_Q 23
#include "Tesis_JF.h"
#include <stdio.h>
#include "IQmathLib.h"

#define Lazo_Abierto 0

```

```

#define MinMax 300
#define Arg (0.01892525695)
#define N 332
#define n 2
#define KTs 0.08514056225 //0.024240556225-B2--0.024230556225-B1---0.03514056225
#define KTs_B 0.08514056225
#define KTs_C 0.085140556225
enum state ESTADO = POWEROFF;

volatile Uint32 Xint1Count = 0;
volatile Uint32 Xint2Count = 0;
Uint32 LoopCount;

_iq Vin = _IQ(0.0);

_iq Error = _IQ(0.0);
_iq Error_B = _IQ(0.0);
_iq Error_C = _IQ(0.0);

_iq U_p = _IQ(0.0);
_iq U_p_B= _IQ(0.0);
_iq U_p_C = _IQ(0.0);

_iq U_v= _IQ(0.0);
_iq U_v_B= _IQ(0.0);
_iq U_v_C= _IQ(0.0);

_iq U_i= _IQ(0.0);
_iq U_i_B= _IQ(0.0);
_iq U_i_C= _IQ(0.0);

_iq U_e = _IQ(0.0);
_iq U_e_B = _IQ(0.0);
_iq U_e_C = _IQ(0.0);

_iq presat = _IQ(0.0);
_iq presat_B = _IQ(0.0);
_iq presat_C = _IQ(0.0);

_iq Vref = _IQ(0.0);
_iq K_Vo = _IQ(0.0);

_iq U6 = _IQ(1.0);
_iq U7 = _IQ(1.0);
_iq U8 = _IQ(1.0);

_iq Theta = _IQ(0.0);

_iq Temp = _IQ(0.0);
_iq Temp_B = _IQ(0.0);

```

```

_iq Temp_C = _IQ(0.0);

_iq X[n] = {_IQ(0.0)};
_iq disp_Q[N] = {_IQ(0.0)};
_iq disp2_Q[N] = {_IQ(0.0)};
_iq disp3_Q[N] = {_IQ(0.0)};
_iq disp4_Q[N] = {_IQ(0.0)};
_iq disp5_Q[N] = {_IQ(0.0)};
_iq K = _IQ(0.0);

int disp_I[N] = {0};
//int disp_N[N] = {0};
Uint16 out;

ABC_struct line_a,line_b,line_c;
VARIABLES_CONTROL_REGS RealEst_Var;

//int16 LazoAbierto = 0;

void main(void)
{
    InitSysCtrl();
    disp_I[0] = K_TBPRD_2;
    DINT;
    InitPieCtrl();
    Inicializa espacio de interrupciones //
    IER = 0x0000; //
    Deshabilita y borra todas las banderas de interrupción
    IFR = 0x0000;
    InitPieVectTable(); //
    Inicializa la Tabla de Vectores del PIE con los punteros para cada interrupción
    EALLOW;
    PieVectTable.ADCINT = &adc_isr;
    PieVectTable.XINT3 = &Off_isr; // Rutina
    para la Interrupción de Apagado
    PieVectTable.XINT1 = &WAKE_UP_isr; // Rutina
    para la Interrupción de Encendido
    EDIS;
    IER |= M_INT1;
    IER |= M_INT3; // Habilitar
    interrupciones para los PWMs 1 a 3
    IER |= M_INT12;
    PieCtrlRegs.PIEIER1.bit.INTx4 = 1; // Habilita
    la interrupción externa XINT1 (Pulsador de Encendido)
    PieCtrlRegs.PIEIER1.bit.INTx6 = 1; // Habilita
    interrupción ADC (PWM 2)
    PieCtrlRegs.PIEIER3.bit.INTx3 = 1; // Habilita
    interrupción del PWM1
    PieCtrlRegs.PIEIER12.bit.INTx1 = 1;

```

```

line_a.v_fd=_IQ(0.0);
line_b.v_fd=_IQ(0.0);
line_c.v_fd=_IQ(0.0);
line_a.i_fd=_IQ(0.0);
line_b.i_fd=_IQ(0.0);
line_c.i_fd=_IQ(0.0);

line_a.Theta= _IQ(0.0);
line_b.Theta= _IQ(1.93732); //3.87463 //1.93732
line_c.Theta= _IQ(3.87463);

/*RealEst_Var.Error = _IQ(0.0);
RealEst_Var.K_Vo = _IQ(30.0);
RealEst_Var.U6 = _IQ(1.0);
RealEst_Var.U_ea = _IQ(0.0);
RealEst_Var.U_eb = _IQ(0.0);
RealEst_Var.U_ec = _IQ(0.0);
RealEst_Var.U_ia = _IQ(0.0);
RealEst_Var.U_ib = _IQ(0.0);
RealEst_Var.U_ic = _IQ(0.0);
RealEst_Var.U_pa = _IQ(0.0);
RealEst_Var.U_pb = _IQ(0.0);
RealEst_Var.U_pc = _IQ(0.0);
RealEst_Var.U_va = _IQ(0.0);
RealEst_Var.U_vb = _IQ(0.0);
RealEst_Var.U_vc = _IQ(0.0);
RealEst_Var.Vref = _IQ(0.0);
RealEst_Var.presat = _IQ(0.0);*/

InitAdc();
ADC_Config();
InitEPwm3();
InitEPwm4();
InitEPwm5();
Init_Sw_On_port();
Init_Sw_Off_port();
K_Vo = _IQ(30.0);
//U_p = _IQ(0.0);
//U_e = _IQ(0.0);

ESTADO = POWEROFF;
EINT;
ERTM;

while(1){

    switch(ESTADO)
    {
        case(POWEROFF):
            system_off();
            break;

```

```

        case(PWEROFF):
            system_off();
        break;
        case(EMPTY):

        break;
    }

};

}

/*****
*****/
/*****
*****/

interrupt void adc_isr(void)
{
    //_iq tmp_iq;
    static Uint16 index = 0;
    //LECTURA ADC
    //Vin = _IQmpy(K_Vin, _IQ12toIQ((_iq)AdcMirror.ADCRESULT0-
    (_iq)AdcMirror.ADCRESULT1));
    K_Vo = _IQmpy(K_Vo, _IQ12toIQ((_iq)AdcMirror.ADCRESULT0));
    line_a.v_fd = _IQmpy(K_Vout, _IQ12toIQ((_iq)AdcMirror.ADCRESULT3-
    (_iq)AdcMirror.ADCRESULT2));
    line_b.v_fd = _IQmpy(K_Vout, _IQ12toIQ((_iq)AdcMirror.ADCRESULT7-
    (_iq)AdcMirror.ADCRESULT6));
    line_c.v_fd = _IQmpy(K_Vout, _IQ12toIQ((_iq)AdcMirror.ADCRESULT9-
    (_iq)AdcMirror.ADCRESULT8));

    line_a.i_fd = _IQmpy(K_I1, _IQ12toIQ((_iq)AdcMirror.ADCRESULT5-
    (_iq)AdcMirror.ADCRESULT4));
    line_b.i_fd = _IQmpy(K_I1, _IQ12toIQ((_iq)AdcMirror.ADCRESULT11-
    (_iq)AdcMirror.ADCRESULT10));
    line_c.i_fd = _IQmpy(K_I1, _IQ12toIQ((_iq)AdcMirror.ADCRESULT13-
    (_iq)AdcMirror.ADCRESULT12));

    if(K_Vo > _IQ(160.0))
        K_Vo = _IQ(150.0);

    if(K_Vo < _IQ(10.0))
        K_Vo = _IQ(10.0);

    // Señal de referencia
    K = _IQ((Arg*index));

    line_a.Vref = _IQmpy(K_Vo, _IQsin(K+line_a.Theta));
    line_b.Vref = _IQmpy(K_Vo, _IQsin(K+line_b.Theta));
    line_c.Vref = _IQmpy(K_Vo, _IQsin(K+line_c.Theta));

    // Error estado estacionario

```

```

Error = (line_a.Vref - line_a.v_fd);
Error_B=(line_b.Vref - line_b.v_fd);
Error_C=(line_c.Vref - line_c.v_fd);
//COMPENSADOR
// compensacion Corriente
U_i = _IQmpy(line_a.i_fd,_IQ(1.4635));
U_i_B = _IQmpy(line_b.i_fd,_IQ(1.4635));
U_i_C = _IQmpy(line_c.i_fd,_IQ(1.4635));
// compensacion voltaje
U_v = _IQmpy(line_a.v_fd,_IQ(-0.0017));
U_v_B = _IQmpy(line_b.v_fd,_IQ(-0.0017));
U_v_C = _IQmpy(line_c.v_fd,_IQ(-0.0017));

#if(Lazo_Abierto == 1) //LAZO ABIERTO
Temp = _IQmpy(line_a.Vref, _IQ(0.002764705));
U_p = _IQmpy(_IQ(0.05),_IQcos(K));

Temp_B = _IQmpy(line_b.Vref, _IQ(0.002764705));
U_p_B = _IQmpy(_IQ(0.05),_IQcos(K));

Temp_C = _IQmpy(line_c.Vref, _IQ(0.002764705));
U_p_C = _IQmpy(_IQ(0.05),_IQcos(K));
#else // LAZO CERRADO
U_p = _IQmpy(Error, _IQ(0.0095));
U_p_B = _IQmpy(Error_B, _IQ(0.0095));
U_p_C = _IQmpy(Error_C, _IQ(0.0095));

#endif

U_e = U_e + _IQmpy(_IQmpy(_IQ(KTs),U_p),U6);
U_e_B = U_e_B + _IQmpy(_IQmpy(_IQ(KTs_B),U_p_B),U7);
U_e_C = U_e_C + _IQmpy(_IQmpy(_IQ(KTs_C),U_p_C),U8);

#if(Lazo_Abierto == 1) //LAZO ABIERTO
presat = Temp + _IQ(0.5) + U_e;
presat_B = Temp_B + _IQ(0.5) + U_e_B;
presat_C = Temp_C + _IQ(0.5) + U_e_C;

#else
presat = _IQ(0.5) + U_e - U_v - U_i;
presat_B = _IQ(0.5) + U_e_B - U_v_B - U_i_B;
presat_C = _IQ(0.5) + U_e_C - U_v_C - U_i_C;
#endif
if ((presat > _IQ(1.0)) || (presat < _IQ(0.0)))
U6 = _IQ(0.0);
else
U6 = _IQ(1.0);

if ((presat_B > _IQ(1.0)) || (presat_B < _IQ(0.0)))
U7 = _IQ(0.0);
else
U7 = _IQ(1.0);

if ((presat_C > _IQ(1.0)) || (presat_C < _IQ(0.0)))

```

```

        U8 = _IQ(0.0);
    else
        U8 = _IQ(1.0);

    //presat=_IQ(3755.0);
    //presat=_IQ(30.0);

    EPwm3Regs.CMPA.half.CMPA = _IQsat(_IQmpy(presat,3765),(EPWM2_TBPRD -
MinMax),MinMax);
    EPwm4Regs.CMPA.half.CMPA = _IQsat(_IQmpy(presat_B,3765),(EPWM2_TBPRD -
MinMax),MinMax);
    EPwm5Regs.CMPA.half.CMPA = _IQsat(_IQmpy(presat_C,3765),(EPWM2_TBPRD -
MinMax),MinMax);

    //U_e[1] = U_e[0];
    //U_p[1] = U_p[0];

    //BUFFER DE VISUALIZACION
    /*disp_Q[index]=U_i;
    disp2_Q[index]=U_e;
    disp3_Q[index]=presat;
    disp4_Q[index]=U_v;
    disp5_Q[index]=Error;
    disp_I[index] = EPwm3Regs.CMPA.half.CMPA;*/

    /*disp_Q[index]=U_i_B;
    disp2_Q[index]=line_b.Vref;
    disp3_Q[index]=line_c.Vref;
    disp4_Q[index]=U_v_B;
    disp5_Q[index]=Error_B;
    disp_I[index] = EPwm4Regs.CMPA.half.CMPA;*/

    disp_Q[index]=U_i_C;
    disp2_Q[index]=U_e;
    disp3_Q[index]=U_e_B;
    disp4_Q[index]=U_e_C;
    disp5_Q[index]=Error_C;
    disp_I[index] = EPwm5Regs.CMPA.half.CMPA;
    //disp_N[index] = EPwm3Regs.CMPA.half.CMPA;
    index++;
    if(index >= N)
    {
        index = 0;
    }

    AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;
    AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;
    PieCtrlRegs.PIEACK.all =PIEACK_GROUP1;
}

```

```

/*****
*****/
/*****
*****/
interrupt void WAKE_UP_isr(void)
{
    Xint1Count++;
    ESTADO = POWERON;
    XIntruptRegs.XINT1CR.bit.ENABLE = 0; // deshabilita interrupci n XINT1
    //XIntruptRegs.XINT3CR.bit.ENABLE = 1; // Habilita interrupci n XINT2 -
    Apagado
    // Acknowledge this interrupt to get more from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}
interrupt void Off_isr(void)
{
    Xint2Count++;
    ESTADO = POWEROFF;
    XIntruptRegs.XINT1CR.bit.ENABLE = 1; // deshabilita interrupci n XINT1
    // Acknowledge this interrupt to get more from group 1
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;
}

/*****
*****/
void system_off(void)
{
    EPwm3Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;
    EPwm4Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;
    EPwm5Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;
    PieCtrlRegs.PIEIER1.bit.INTx6 = 0; //
    Deshabilita interrupcion ADC (PWM 2)
    ESTADO=EMPY;
}

/*****
*****/
void system_on(void)
{
    Error = _IQ(0.0);
    Error_B = _IQ(0.0);
    Error_C = _IQ(0.0);

    U_e = _IQ(0.0);
    U_e_B = _IQ(0.0);
    U_e_C = _IQ(0.0);

    U6 = _IQ(1.0);
    U7 = _IQ(1.0);
    U8 = _IQ(1.0);

```

```

EPwm3Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;
DELAY_US(20);
EPwm3Regs.DBCTL.bit.POLSEL = DB_ACTV_LOC; // PWMs complementarios
DELAY_US(20);
EPwm3Regs.AQCSFRC.bit.CSFA = FORCE_DISABLE; // PWMs libres

EPwm4Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;
DELAY_US(20);
EPwm4Regs.DBCTL.bit.POLSEL = DB_ACTV_LOC;
DELAY_US(20);
EPwm4Regs.AQCSFRC.bit.CSFA = FORCE_DISABLE;

EPwm5Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;
DELAY_US(20);
EPwm5Regs.DBCTL.bit.POLSEL = DB_ACTV_LOC;
DELAY_US(20);
EPwm5Regs.AQCSFRC.bit.CSFA = FORCE_DISABLE;

PieCtrlRegs.PIEIER1.bit.INTx6 = 1; // Habilita
interrupcion ADC (PWM 2)

ESTADO=EMPY;

}

/*****
*****/
void InitEPwm3(void)
{
    InitEPwm3Gpio();
    /*
    EALLOW;
    GpioCtrlRegs.GPAPUD.bit.GPIO13 = 0; // Enable
    pull-up on GPIO13 (TZ2)
    GpioCtrlRegs.GPAQSEL1.bit.GPIO13 = 3; // Asynch
    input GPIO13 (TZ2)
    GpioCtrlRegs.GPAMUX1.bit.GPIO13 = 1; // Configure
    GPIO13 as TZ2
    GpioCtrlRegs.GPAPUD.bit.GPIO14 = 0; // Enable
    pull-up on GPIO13 (TZ2)
    GpioCtrlRegs.GPAQSEL1.bit.GPIO14 = 3; // Asynch
    input GPIO13 (TZ2)
    GpioCtrlRegs.GPAMUX1.bit.GPIO14 = 1; // Configure
    GPIO13 as TZ2
    SysCtrlRegs.PCLKCR1.bit.EPWM3ENCLK = 1; // Habilitar
    CLK EPWM2 EDIS;
    EDIS;
    */
    // Setup TBCLK
    EPwm3Regs.TBPRD = EPWM2_TBPRD; // Set timer
    period 3765 TBCLKs
    EPwm3Regs.TBPHS.half.TBPHS = 0; // Phase is
    0

```

```

    EPwm3Regs.TBCTR = 0; // Clear
counter
    EPwm3Regs.TBCTL.bit.FREE_SOFT = 3;
    // Set Compare values
    EPwm3Regs.CMPA.half.CMPA = EPWM2_TBPRD>>1; // Set
compare A value
    // Setup counter mode
    EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Count up
    EPwm3Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Disable
phase loading
    EPwm3Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // Clock
ratio to SYSCLKOUT
    EPwm3Regs.TBCTL.bit.CLKDIV = TB_DIV1;
    //Modificacion 1
    EPwm3Regs.TBCTL.bit.PRDL = TB_SHADOW;
    EPwm3Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE; // Setup
shadowing
    EPwm3Regs.TBCTL.bit.FREE_SOFT = 3;
    EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
    EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
    EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_PRD; // Load on
PRD

    EPwm3Regs.ETSEL.bit.SOCEN = 1; // Enable SOC on A group
    EPwm3Regs.ETSEL.bit.SOCSEL = ET_CTR_PRD; // Select SOC on count = PRD
    EPwm3Regs.ETPS.bit.SOCAPRD = ET_1ST; // Generate pulse on 1st event

    // Set actions
    EPwm3Regs.AQCTLA.bit.CAD = AQ_SET; // Set
PWM3A on event B, down count
    EPwm3Regs.AQCTLA.bit.CAU = AQ_CLEAR; // Clear
PWM3A on event A, up count //
    EPwm3Regs.ETSEL.bit.INTEN = 0; // Disable
inter periférico

    //Tiempo Muerto
    EPwm3Regs.DBCTL.bit.IN_MODE = DBA_ALL;
    EPwm3Regs.DBCTL.bit.POLSEL = DB_ACTV_LO; // DB_ACTV_HIC;
// Deja los dos PWMs complementarios
    EPwm3Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
    EPwm3Regs.DBRED = 400;
    EPwm3Regs.DBFED = 400;
/*
    EALLOW;
    EPwm3Regs.TZSEL.bit.CBC2 = TZ_ENABLE; //
    EPwm3Regs.TZSEL.bit.CBC3 = TZ_ENABLE; // QUEDO EN
BAJO Y NO SE PUEDE HABILITAR TZ DISPARA CON BAJO
    EPwm3Regs.TZCTL.bit.TZA = TZ_FORCE_LO; //
    EPwm3Regs.TZCTL.bit.TZB = TZ_NO_CHANGE; //
    EPwm3Regs.TZEINT.bit.CBC = TZ_ENABLE; // Habilita
interrupción
    EDIS;

```

```

*/
    //EPwm3Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;           // Deja
    los PWMs en bajo

}
/*****
*****/
/*****
*****/
void InitEPwm4(void)
{
    InitEPwm4Gpio();
    /*
        EALLOW;
        GpioCtrlRegs.GPAPUD.bit.GPIO13 = 0;                 // Enable
pull-up on GPIO13 (TZ2)
        GpioCtrlRegs.GPAQSEL1.bit.GPIO13 = 3;               // Asynch
input GPIO13 (TZ2)
        GpioCtrlRegs.GPAMUX1.bit.GPIO13 = 1;               // Configure
GPIO13 as TZ2
        GpioCtrlRegs.GPAPUD.bit.GPIO14 = 0;                 // Enable
pull-up on GPIO13 (TZ2)
        GpioCtrlRegs.GPAQSEL1.bit.GPIO14 = 3;               // Asynch
input GPIO13 (TZ2)
        GpioCtrlRegs.GPAMUX1.bit.GPIO14 = 1;               // Configure
GPIO13 as TZ2
        SysCtrlRegs.PCLKCR1.bit.EPWM3ENCLK = 1;            // Habilitar
CLK EPWM2 EDIS;
        EDIS;
    */
    // Setup TBCLK
    EPwm4Regs.TBPRD = EPWM2_TBPRD;                          // Set timer
period 3765 TBCLKs
    EPwm4Regs.TBPHS.half.TBPHS = 0;                         // Phase is
0
    EPwm4Regs.TBCTR = 0;                                     // Clear
counter
    EPwm4Regs.TBCTL.bit.FREE_SOFT = 3;

    // Set Compare values
    EPwm4Regs.CMPA.half.CMPA = EPWM2_TBPRD>>1;            // Set
compare A value
    // Setup counter mode
    EPwm4Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;         // Count up
    EPwm4Regs.TBCTL.bit.PHSEN = TB_DISABLE;                 // Disable
phase loading
    EPwm4Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;                // Clock
ratio to SYSCLKOUT
    EPwm4Regs.TBCTL.bit.CLKDIV = TB_DIV1;
    //Modificacion 1
    EPwm4Regs.TBCTL.bit.PRDLT = TB_SHADOW;
    EPwm4Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;         // Setup
shadowing

```

```

EPwm4Regs.TBCTL.bit.FREE_SOFT=3;
EPwm4Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm4Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm4Regs.CMPCTL.bit.LOADAMODE = CC_CTR_PRD;          // Load on
PRD

EPwm4Regs.ETSEL.bit.SOCAEN = 1;          // Enable SOC on A group
EPwm4Regs.ETSEL.bit.SOCASEL = ET_CTR_PRD;    // Select SOC on count = PRD
EPwm4Regs.ETPS.bit.SOCAPRD = ET_1ST;        // Generate pulse on 1st event

// Set actions
EPwm4Regs.AQCTLA.bit.CAD = AQ_SET;          // Set
PWM3A on event B, down count
EPwm4Regs.AQCTLA.bit.CAU = AQ_CLEAR;        // Clear
PWM3A on event A, up count //
EPwm4Regs.ETSEL.bit.INTEN = 0;              // Disable
inter periférico

//Tiempo Muerto
EPwm4Regs.DBCTL.bit.IN_MODE = DBA_ALL;
EPwm4Regs.DBCTL.bit.POLSEL = DB_ACTV_LO;    // Deja los
dos PWMs complementarios
EPwm4Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
EPwm4Regs.DBRED = 400;
EPwm4Regs.DBFED = 400;

/*
EALLOW;
EPwm3Regs.TZSEL.bit.CBC2 = TZ_ENABLE;      //
EPwm3Regs.TZSEL.bit.CBC3 = TZ_ENABLE;      // QUEDO EN
BAJO Y NO SE PUEDE HABILITAR TZ DISPARA CON BAJO
EPwm3Regs.TZCTL.bit.TZA = TZ_FORCE_LO;     //
EPwm3Regs.TZCTL.bit.TZB = TZ_NO_CHANGE;    //
EPwm3Regs.TZEINT.bit.CBC = TZ_ENABLE;      // Habilita
interrupción
EDIS;
*/
//EPwm3Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW;    // Deja
los PWMs en bajo

}
/*****
*****/
/*****
*****/
void InitEPwm5(void)
{
    InitEPwm5Gpio();
/*
EALLOW;
GpioCtrlRegs.GPAPUD.bit.GPIO13 = 0;          // Enable
pull-up on GPIO13 (TZ2)

```

```

        GpioCtrlRegs.GPAQSEL1.bit.GPIO13 = 3;           // Asynch
input GPIO13 (TZ2)
        GpioCtrlRegs.GPAMUX1.bit.GPIO13 = 1;           // Configure
GPIO13 as TZ2
        GpioCtrlRegs.GPAPUD.bit.GPIO14 = 0;           // Enable
pull-up on GPIO13 (TZ2)
        GpioCtrlRegs.GPAQSEL1.bit.GPIO14 = 3;           // Asynch
input GPIO13 (TZ2)
        GpioCtrlRegs.GPAMUX1.bit.GPIO14 = 1;           // Configure
GPIO13 as TZ2
        SysCtrlRegs.PCLKCR1.bit.EPWM3ENCLK = 1;        // Habilitar
CLK EPWM2 EDIS;
        EDIS;
*/
        // Setup TBCLK
        EPwm5Regs.TBPRD = EPWM2_TBPRD;                 // Set timer
period 3765 TBCLKs
        EPwm5Regs.TBPHS.half.TBPHS = 0;                // Phase is
0
        EPwm5Regs.TBCTR = 0;                            // Clear
counter
        EPwm5Regs.TBCTL.bit.FREE_SOFT = 3;
        // Set Compare values
        EPwm5Regs.CMPA.half.CMPA = EPWM2_TBPRD>>1;    // Set
compare A value
        // Setup counter mode
        EPwm5Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Count up
        EPwm5Regs.TBCTL.bit.PHSEN = TB_DISABLE;        // Disable
phase loading
        EPwm5Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;       // Clock
ratio to SYSCLKOUT
        EPwm5Regs.TBCTL.bit.CLKDIV = TB_DIV1;
        //Modificacion 1
        EPwm5Regs.TBCTL.bit.PRDL = TB_SHADOW;
        EPwm5Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE; // Setup
shadowing
        EPwm5Regs.TBCTL.bit.FREE_SOFT=3;
        EPwm5Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
        EPwm5Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
        EPwm5Regs.CMPCTL.bit.LOADAMODE = CC_CTR_PRD;   // Load on
PRD

        EPwm5Regs.ETSEL.bit.SOCAEN = 1;               // Enable SOC on A group
        EPwm5Regs.ETSEL.bit.SOCASEL = ET_CTR_PRD;      // Select SOC on count = PRD
        EPwm5Regs.ETPS.bit.SOCAPRD = ET_1ST;          // Generate pulse on 1st event

        // Set actions
        EPwm5Regs.AQCTLA.bit.CAD = AQ_SET;             // Set
PWM3A on event B, down count
        EPwm5Regs.AQCTLA.bit.CAU = AQ_CLEAR;          // Clear
PWM3A on event A, up count //

```

```

    EPwm5Regs.ETSEL.bit.INTEN = 0; // Disable
inter periferico

    //Tiempo Muerto
    EPwm5Regs.DBCTL.bit.IN_MODE = DBA_ALL;
    EPwm5Regs.DBCTL.bit.POLSEL = DB_ACTV_LO; // Deja los
dos PWMs complementarios
    EPwm5Regs.DBCTL.bit.OUT_MODE = DB_FULL_ENABLE;
    EPwm5Regs.DBRED = 400;
    EPwm5Regs.DBFED = 400;
/*
    EALLOW;
    EPwm3Regs.TZSEL.bit.CBC2 = TZ_ENABLE; //
    EPwm3Regs.TZSEL.bit.CBC3 = TZ_ENABLE; // QUEDO EN
BAJO Y NO SE PUEDE HABILITAR TZ DISPARA CON BAJO
    EPwm3Regs.TZCTL.bit.TZA = TZ_FORCE_LO; //
    EPwm3Regs.TZCTL.bit.TZB = TZ_NO_CHANGE; //
    EPwm3Regs.TZEINT.bit.CBC = TZ_ENABLE; // Habilita
interrupción
    EDIS;
*/
    //EPwm3Regs.AQCSFRC.bit.CSFA = FORCE_CONT_LOW; // Deja
los PWMs en bajo

}
/*****
*****/
/*****
*****/
void ADC_Config(void)
{
    // Fck1_ADC = 150MHz/(2*ADC_MODCLK*(ADCTRL1[11:8] + 1)times) =
    150MHz/(2*3*2) = 12.5MHz
    AdcRegs.ADCTRL1.bit.CPS = 1; // ADCCLK =
Fclk/(CPS + 1)
    AdcRegs.ADCTRL1.bit.ACQ_PS = 15; // The width
of SOC pulse is ADCTRL1[11:8] + 1 times the ADCLK period.
    AdcRegs.ADCTRL1.bit.SEQ_CASC = 1; // Setup
cascaded sequencer mode
    AdcRegs.ADCTRL2.bit.EPWM_SOCA_SEQ1 = 1; // Enable
SOCA from ePWM to start SEQ1
    AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 1; // Enable
SEQ1 interrupt (every EOS)
    AdcRegs.ADCTRL3.bit.SMODE_SEL = 0x1; // Setup
simultaneous sampling mode
    AdcRegs.ADCTRL3.bit.ADCCLKPS = 1;
    AdcRegs.ADCMAXCONV.all = 0x0007; // 8 double
conv's each sequencer (16 total)
    AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0; // Setup
conv from ADCINA0 & ADCINB0
    AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x1; // Setup
conv from ADCINA1 & ADCINB1

```

```

        AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x2;           // Setup
    conv from ADCINA2 & ADCINB2
        AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x3;           // Setup
    conv from ADCINA3 & ADCINB3
        AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x4;           // Setup
    conv from ADCINA4 & ADCINB4
        AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 0x5;           // Setup
    conv from ADCINA5 & ADCINB5
        AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 0x6;           // Setup
    conv from ADCINA6 & ADCINB6
        AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 0x7;           // Setup
    conv from ADCINA7 & ADCINB7
        // ADCINA0 -> ADCRESULT0
        // ADCINB0 -> ADCRESULT1
        // ADCINA1 -> ADCRESULT2
        // ADCINB1 -> ADCRESULT3
        // ADCINA2 -> ADCRESULT4
        // ADCINB2 -> ADCRESULT5
        // ADCINA3 -> ADCRESULT6
        // ADCINB3 -> ADCRESULT7
        // ADCINA4 -> ADCRESULT8
        // ADCINB4 -> ADCRESULT9
        // ADCINA5 -> ADCRESULT10
        // ADCINB5 -> ADCRESULT11
        // ADCINA6 -> ADCRESULT12
        // ADCINB6 -> ADCRESULT13
    }
    /*****
    *****/
void Init_Sw_On_port(void)
{
    EALLOW;
    GpioCtrlRegs.GPAPUD.bit.GPIO29 = 0; // Habilita Pull Ups
    GpioCtrlRegs.GPACTRL.bit.QUALPRD3 = 0xFF; // Especifica el periodo de
    muestreo para los pines GPIO24 al GPIO31
    GpioCtrlRegs.GPAQSEL2.bit.GPIO29 = 2; // 6 Muestras
    GpioCtrlRegs.GPAMUX2.bit.GPIO29 = 0; // Configure GPIO29 as I/O
    GpioCtrlRegs.GPADIR.bit.GPIO29 = 0; // Configure GPIO29 as Input
    //GpioIntRegs.GPIOLPMSEL.bit.GPIO29 = 1; // Configura que pin sacara al DSP
    de HALT o STANDBY
    GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 29; // Configura disparo de
    interrupción XINT1
    EDIS;
    XIntruptRegs.XINT1CR.bit.ENABLE = 1; // Habilita interrupción XINT1
    XIntruptRegs.XINT1CR.bit.POLARITY = 0; // XINT1 Flanco de Bajada
}
/*****
*****/
/*****
*****/

void Init_Sw_Off_port(void)

```

```

{
    EALLOW;
    GpioCtrlRegs.GPBPUD.bit.GPIO33 = 0; // Enable Pull Up
    GpioCtrlRegs.GPBCTRL.bit.QUALPRD0 = 0xFF; // Especifica el periodo de
muestreo para los pines GPIO32 al GPIO39
    GpioCtrlRegs.GPBQSEL1.bit.GPIO33 = 2; // 6 Samples
    GpioCtrlRegs.GPBMUX1.bit.GPIO33 = 0; // Configure GPIO33 as I/O
    GpioCtrlRegs.GPBDIR.bit.GPIO33 = 0; // Configure GPIO33 as Input
    GpioIntRegs.GPIOXINT3SEL.bit.GPIOSEL = 33; // Configura disparo de
interrupción XINT3
    EDIS;
    XIntruptRegs.XINT3CR.bit.ENABLE = 1; // Habilita interrupción XINT3
    XIntruptRegs.XINT3CR.bit.POLARITY = 0; // XINT3 Flanco de Bajada
}

```