

**MEDICIÓN Y CONTROL DE POSICIÓN ANGULAR EMPLEANDO  
SYNCHRO - RESOLVER**

**DANIEL FERNANDO RODRIGUEZ PALACIOS  
HECTOR ENRIQUE PEÑARETE RODRIGUEZ**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA ELECTRÓNICA  
BOGOTÁ D.C.  
MAYO DE 2011**

**MEDICIÓN Y CONTROL DE POSICIÓN ANGULAR EMPLEANDO  
SYNCHRO - RESOLVER**

**T.G. 1022**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO ELECTRÓNICO**

**DIRECTOR  
CARLOS EDUARDO COTRINO BADILLO M.Sc.  
INGENIERO ELECTRÓNICO**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA ELECTRÓNICA  
BOGOTÁ D.C.  
MAYO DE 2011**

**Pontificia Universidad Javeriana**  
**Facultad de Ingeniería**  
**Departamento de Electrónica**

<b>Rector Magnífico</b>	PADRE JOAQUÍN EMILIO SÁNCHEZ GARCÍA S.J.
<b>Decano Académico</b>	ING. FRANCISCO JAVIER REBOLLEDO MUÑOZ.
<b>Decano del Medio</b>	PADRE SERGIO BERNAL RESTREPO S.J.
<b>Director de Carrera</b>	ING. JUAN MANUEL CRUZ BOHORQUEZ M.Edu.
<b>Director de Proyecto</b>	ING. CARLOS EDUARDO COTRINO BADILLO M.Sc.
<b>Asesor de Proyecto</b>	ING. RAFAEL PUERTA RAMIREZ

## **ARTICULO 23 DE LA RESOLUCIÓN No. 13 DE JUNIO DE 1946**

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Solo velará porque no se publique nada contrario al dogma y a la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia.”

## Tabla de contenido

1.	INTRODUCCIÓN .....	12
2.	MARCO TEÓRICO .....	13
2.1	SENSORES DE POSICIÓN ANGULAR:.....	13
2.1.1	SYNCHRO <sup>[3]</sup> :.....	13
2.1.2	RESOLVER: .....	13
2.1.3	ENCODER INCREMENTAL <sup>[10]</sup> .....	14
2.2	CONVERSIÓN RESOLVER A DIGITAL .....	18
2.2.1	LAZO DE SEGUIMIENTO (SERVO LOOP) TIPO II: .....	18
2.2.2	CONVERSION POR TRACKING O SEGUIMIENTO <sup>[15]</sup> : .....	18
2.3	MICROCONTROLADOR dsPIC <sup>[12]</sup> .....	20
2.4	CONVERSION AD2S1205 <sup>[14]</sup> .....	21
2.5	MOTOR ELÉCTRICO <sup>[13]</sup> .....	23
2.6	PUENTE H <sup>[13]</sup> .....	25
3.	ESPECIFICACIONES .....	27
3.1	DESCRIPCIÓN GENERAL .....	27
3.2	DIAGRAMA EN BLOQUES .....	29
3.2.1	Etapa de Aplicación: .....	29
3.2.2	Etapa de Procesamiento Analógico:.....	29
3.2.3	Etapa de Procesamiento Digital: .....	29
3.2.4	Etapa de Control:.....	29
3.2.5	Etapa de Interfaz con el Usuario: .....	29
3.3	Motor PITTMAN 9233 24 V <sup>[16]</sup> :.....	30
3.3.1	Encoder incremental óptico E30 <sup>[16]</sup> :.....	30
3.4	CONVERSION AD2S1205 <sup>[15]</sup> : .....	31
3.5	Synchro Resolver Clifton.....	32
3.6	dsPIC30F2010 <sup>[17]</sup> : .....	32
3.6.1	MODULO QEI <sup>[18]</sup> : .....	33
4.	DESARROLLOS .....	36
4.1	TARJETAS DE DESARROLLO: .....	37
4.1.1	Tarjeta de desarrollo para AD2S1205:.....	37
4.1.2	Tarjeta de desarrollo para dsPIC30F2010:.....	39
4.2	ETAPA DE APLICACIÓN: .....	39
4.3	ETAPA DE PROCESAMIENTO ANALÓGICO: .....	40

4.3.1	Generación de la señal de referencia para synchro-resolver. ....	40
4.3.2	Emulación de Encoder de Cuadratura. ....	42
4.4	ETAPA DE PROCESAMIENTO DIGITAL: .....	43
4.4.1	CONVERSIÓN RESOLVER A DIGITAL USANDO EL MÉTODO DE TRACKING <sup>[14]</sup> : 43	
4.5	ETAPA DE CONTROL:.....	44
4.5.1	MODELAMIENTO DEL SISTEMA .....	44
4.5.2	IMPLEMENTACIÓN DEL CONTROL: .....	45
4.5.3	PWM .....	52
4.5.4	PUENTE H L6203 <sup>[21]</sup> :.....	52
4.6	COMUNICACIÓN SERIAL, MICROCONTROLADOR-HYPERTERMINAL: .....	52
4.7	INTERFAZ CON EL USUARIO: .....	53
5	ANÁLISIS DE RESULTADOS .....	54
5.1	VALIDACIÓN DEL MODELO DEL SISTEMA: .....	54
5.1.1	CONSTANTES DEL MOTOR:.....	54
5.1.2	PRUEBA EN MALLA ABIERTA DE LA PLANTA:.....	56
5.2	PRUEBA DE GANANCIA RESOLVER.....	57
5.3	PRUEBAS DE SEÑALES:.....	57
5.3.1	SEÑAL DE ALIMENTACIÓN DE ROTOR DE SYNCHRO - RESOLVER. ....	57
5.3.2	DISTORSIÓN SEÑAL SENO DE ALIMENTACIÓN AL RESOLVER (THD) .....	58
5.3.3	SEÑALES DE SALIDA DE SYNCHRO RESOLVER .....	59
5.4	PRUEBAS DEL CONTROLADOR.....	65
5.4.1	RESPUESTA DEL SISTEMA A ENTRADA RAMPA: .....	65
5.4.2	RESPUESTA DEL SISTEMA A ENTRADA PASO.....	68
5.4.2.1	USANDO ENCODER COMO SENSOR DE POSICIÓN: .....	68
5.4.2.2	USANDO SYNCHRO – RESOLVER COMO SENSOR DE POSICIÓN: .....	69
5.4.2.3	RESPUESTA A ENTRADA PASO (UNA PRUEBA MÁS):.....	69
6.	CONCLUSIONES .....	71
7.	BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN .....	722
	ANEXO A.....	744
	ANEXO B.....	822
	ANEXO C .....	876
	ANEXO D.....	887
	ANEXO E .....	89

ANEXO F.....	900
ANEXO G.....	911

## TABLA DE FIGURAS

Figura 2.1.1.1 Señales del Synchro en función del ángulo.	13
Figura 2.1.2.1 Resolver Clásico.	14
Figura 2.1.2.2 Señales del Resolver en función del ángulo.	14
Figura 2.1.3.1. Encoder Óptico.	15
Figura 2.1.3.2. Representación de señales incrementales en disco óptico.	15
Figura 2.1.3.3 Codificación Incremental.	16
Figura 2.1.3.4. . Resolución y exactitud en encoder.	17
Figura 2.1.3.6. Errores típicos en encoders.	18
Figura 2.2.2.1 Conversión Resolver a Digital.	19
Figura. 2.4.1 Distribución de terminales del AD2S1205.	22
Figura 2.4.2 Diagrama de Bloques de la Respuesta del sistema del convertidor.	22
Figura 2.4.3 Respuesta en magnitud del convertidor.	23
Figura 2.4.4 Respuesta en fase del convertidor.	23
Figura 2.5.1. Motor DC.	24
Figura 2.6.1 Configuración básica de un puente H.	26
Figura 2.6.2 Formas de Onda de entrada y salida del Puente H.	26
Figura 3.1.1 Diagrama en Bloques del sistema.	27
Figura 3.6.2 Pines del dsPIC30F2010.	33
Figura 3.6.1.1. Diagrama en bloques simplificado del módulo QEI.	34
Figura 3.6.1.2. Diagrama de tiempo de las señales del QEI.	35
Figura 4.1. Esquema de desarrollo del sistema.	36
Figura 4.1.1.1. Conexión del convertidor A2DS1200/05.	38
Figura 4.1.1.2. Circuito búffer para AD2S1200/05.	38
Figura 4.2.1. Montaje del sistema completo.	40
Figura 4.3.1.1. Diagrama de bloques funcional del AD2S1205.	41
Figura 4.3.1.2 Especificaciones del las señales de referencia EXC y (EXC)´.	41
Figura 4.3.2.1. Diagrama de tiempo de Emulación de encoder incremental AD2S1205.	42
Figura 4.5.2.1 Simulación de la respuesta a entrada paso para la planta (Amplitud vs Tiempo[s]).	45
Figura 4.5.2.1.1 Lugar de las raíces de la planta.	46
Figura 4.5.2.1.2 Respuesta a entrada paso para la malla definitiva del control Análogo.	47
Figura 4.5.2.2.1 Bode de la función de transferencia motor.	48
Figura 4.5.2.2.2 Lazo del Controlador PID Análogo.	49



Figura 4.6.2.2.3 Lazo del Controlador PID.	49
Figura 4.6.2.2.4 Comparación control discreto y control Análogo.	50
Figura 4.5.2.3.1. Diagrama de flujo del Control.	51
Figura 4.5.4.1. Puente H.L6203.	52
Figura 4.7 Orden en que aparecen las variables en Hyperterminal	53
Figura 5.1.1.1 Constante del generador del motor.	55
Figura 5.1.1.2 Fricción viscosa del motor.	55
Figura 5.2.1.1 Respuesta de la planta en malla abierta.	56
Figura 5.3.1.1. Salidas de alimentación Synchro Resolver de Tarjeta de desarrollo para AD2S1200/05 (Referidas a tierra).	58
Figura 5.3.1.2. Salida de alimentación Synchro Resolver de Tarjeta de desarrollo para AD2S1200/05 (Diferencial).	58
Figura 5.3.2. Medición de THD para la señal de alimentación del resolver.	59
Figura 5.3.3.2. Señales de salida Synchro Resolver. Máxima amplitud S2 – S4.	60
Figura 5.3.3.3. Señales de salida Synchro Resolver. Máxima amplitud S1 – S3	60
Figura 5.3.3.4. Especificaciones de los pines de error.	64
Figura 5.4.1.1.1. Respuesta a entrada rampa con encoder.	65
Figura 5.4.1.1.2 Grafica del error de la respuesta rampa con encoder.	66
Figura 5.4.1.2.1 Respuesta a entrada rampa con Synchro-Resolver.	66
Figura 5.4.1.2.2 Grafica del error a una respuesta rampa con Synchro – Resolver.	67
Figura 5.4.2.1 Respuesta del sistema a una entrada paso de 360° con sensor encoder.	68
Figura 5.4.2.2 Respuesta del sistema una entrada paso de 360° con sensor synchro - resolver.	69
Figura 5.4.2.3.1 Respuesta a entada paso simulada y real para ambos sensores (Una prueba más).	70

## TABLA DE TABLAS

Tabla 3.1.1 Características principales del proyecto.....	28
Tabla 3.3.1 Especificaciones del motor Pittman 9233 24V .....	30
Tabla 3.3.1.1. Especificaciones Encoder E30.....	31
Tabla 3.4.1. Características del conversor AD2S1205.....	31
Tabla 3.5.1. Características Synchro Resolver.....	32
Tabla 3.6.1. Características principales del dsPIC30F2010.....	32
Tabla 5.1.1.1 Prueba motor PITMAN 9233 24V .....	54
Tabla 5.2.1. Prueba de Ganancia de Resolver.....	57
Tabla 5.3.3.1 Convenciones para la medición de señales de salida Synchro Resolver .....	59
Tabla 5.4.2.3.2. Resumen pruebas controlador.....	70

## TABLA DE ANEXOS

ANEXO A.....	;	<b>Error! Marcador no definido.</b>
ANEXO B.....	;	<b>Error! Marcador no definido.</b>
ANEXO C.....	;	<b>Error! Marcador no definido.</b>
ANEXO D.....	;	<b>Error! Marcador no definido.</b>
ANEXO E.....	;	<b>Error! Marcador no definido.</b>
ANEXO F.....	;	<b>Error! Marcador no definido.</b>
ANEXO G.....	;	<b>Error! Marcador no definido.</b>

## 1. INTRODUCCIÓN

La medición y control de movimiento de rotación es una de las aplicaciones más importantes en el control moderno, instrumentación y tecnologías informáticas. También es utilizada en el control de posición, velocidad y aceleración, en una, dos o tres dimensiones, y en muchos sistemas en los que estos parámetros deben ser sensados y/o controlados. Casi todas las máquinas, procesos y sistemas de monitoreo tienen un eje de rotación en algún lugar de su mecanismo. Por esta razón, los transductores de movimiento de rotación son elementos de gran importancia en la ingeniería moderna<sup>[3]</sup>.

Entre los transductores de movimiento de rotación se encuentran los Synchros y Resolvers, que han estado disponibles por más de 70 años, formando parte de servos electromecánicos y sistemas de posición angular, principalmente utilizados en la industria aeronáutica y militar para ayudar a evaluar diferentes tipos de variables físicas como posición y movimiento<sup>[8]</sup>. Se trata de sensores analógicos con resolución teóricamente infinita que en conjunto con la apropiada interfaz electrónica, son el corazón de un sistema de posición y medición angular digital<sup>[1]</sup>. Aunque los transductores digitales y ópticos serán el futuro, los synchros continúan teniendo un amplio mercado gracias a su robustez, resolución, exactitud, además de su inmunidad a la contaminación, humedad, altas temperaturas, vibraciones y su reducido momento de inercia que impone poca carga mecánica al eje al cual se acopla<sup>[6]</sup>.

Por otra parte, se encuentran los encoders; transductores rotacionales más populares que synchros y resolvers, que transforman un movimiento angular en una serie de impulsos digitales y son usados para cuantificar y controlar dicho movimiento. Las aplicaciones principales de estos transductores están en máquinas de elaboración de materiales, robots, en los sistemas de motores, en los aparatos de medición y control.

En este proyecto se implementaron controladores de posición para un sistema electromecánico, en el cual se acoplaron a un motor, un synchro-resolver y un encoder. Se halló y validó el modelo del sistema, para mostrar el funcionamiento y características de cada sensor por medio de prácticas de laboratorio, en las cuales se comparara el rendimiento de cada uno de los esquemas de control y estas estarán enfocadas a las asignaturas de laboratorio de control e instrumentación industrial de la carrera de Ingeniería Electrónica de la Pontificia Universidad Javeriana.

### **OBJETIVO GENERAL:**

Diseñar e implementar controladores de posición utilizando Synchro-Resolver y Encoder.

### **OBJETIVOS ESPECÍFICOS:**

1. Desarrollar experiencias de laboratorio.
2. Comparar rendimiento de los dos esquemas de control.
3. Observar y validar los modelos del sistema.

## 2. MARCO TEÓRICO

### 2.1 SENSORES DE POSICIÓN ANGULAR:

#### 2.1.1 SYNCHRO<sup>[3]</sup>:

Synchro es el término usado para una familia de instrumentos transductores, los cuales pueden ser conectados de diferentes formas para obtener sistemas de medida y posicionamiento angular de un rotor. En apariencia se asemejan a pequeños motores AC. Internamente la mayoría de los synchros son similares en construcción, tienen un rotor, capaz de rotar dentro del estator fijo. El rotor es alimentado con una señal de referencia AC ( $U_0$ ), produciendo en el estator señales proporcionales, en amplitud, al ángulo de rotación del eje, desfasadas  $120^\circ$  y con la misma frecuencia que la señal de referencia, como se observa en la Figura 2.1.1.1.

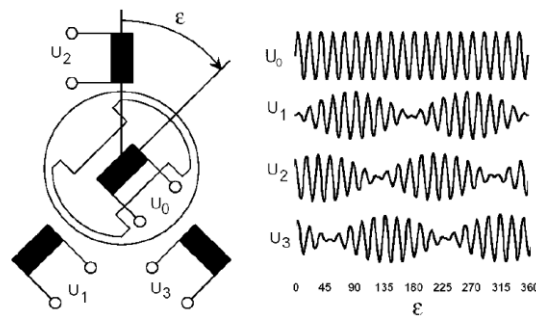
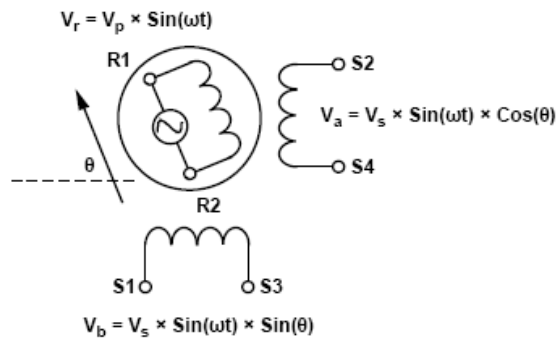


Figura 2.1.1.1 Señales del Synchro en función del ángulo<sup>[9]</sup>.

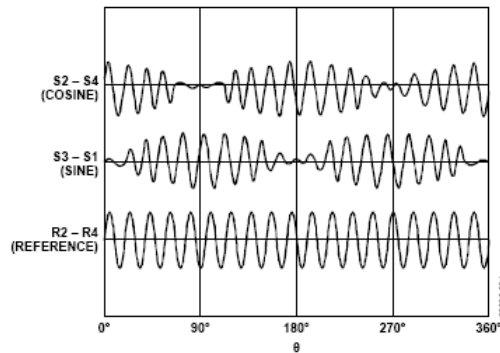
Dentro de la familia de los synchros se encuentran los “Torque Receiver” Figura 2.1.1.2 (comúnmente llamados Synchro Receiver). Son similares en construcción y apariencia a cualquier otro synchro, sin embargo, estos son empleados cuando es necesario recibir movimiento angular a través de su eje y transmitirlo en forma de señales sinusoidales en las salidas del estator. Para obtener el ángulo de movimiento es necesario utilizar una forma de conversión específica de dichas señales, la cual se explicara más adelante.

#### 2.1.2 RESOLVER:

Los resolvers son similares a los synchros en cuanto su apariencia física y funciones, sin embargo el resolver cuenta con dos devanados en el estator en lugar de tres como se observa en la Figura 2.1.2.1, estos están ubicados mecánicamente a  $90^\circ$ . De igual forma que el synchro, en el rotor del resolver se aplica una señal de referencia, generando en los devanados del estator una señal seno y otra coseno de igual frecuencia (Figura 2.1.2.2).



*Figura 2.1.2.1 Resolver Clásico<sup>[14]</sup>.*



*Figura 2.1.2.2 Señales del Resolver en función del ángulo<sup>[14]</sup>.*

Los voltajes de salida del resolver (S3 – S1, S2 – S4), son los siguientes:

$$S3 - S1 = E_o \text{Sin}(\omega t) \times \text{Sin}\theta \quad \text{Eq. 2.1.2.1}$$

$$S2 - S4 = E_o \text{Sin}(\omega t) \times \text{Cos}\theta$$

Donde:

$\theta$  es el ángulo del eje a detectar.

$\text{Sin}(\omega t)$  es la frecuencia de excitación del rotor.

$E_o$  es la amplitud de excitación del rotor.

### 2.1.3 ENCODER INCREMENTAL<sup>[10]</sup>

Los encoders son sensores que generan señales digitales en respuesta a un movimiento. Estas señales generadas pueden ser utilizadas para medir movimiento angular o lineal, y a partir de esto, calcular velocidad o posición. Los encoders pueden ser utilizados en gran cantidad de aplicaciones, ya que actúan

como transductores de realimentación en sistemas de control de velocidad y posición de motores, como sensores de medición y posicionamiento, etc.

### Encoders Ópticos:

Los encoders ópticos, utilizan un disco que contiene un patrón de líneas, se construyen de materiales tales como vidrio, metálicos o plásticos y con roturas o simplemente líneas dibujadas sobre ellos. Una luz proveniente de un LED logra pasar a través del disco, según el patrón de líneas y es detectada por un foto-detector el cual produce la salida del encoder. Un encoder incremental puede contener uno o más caminos.

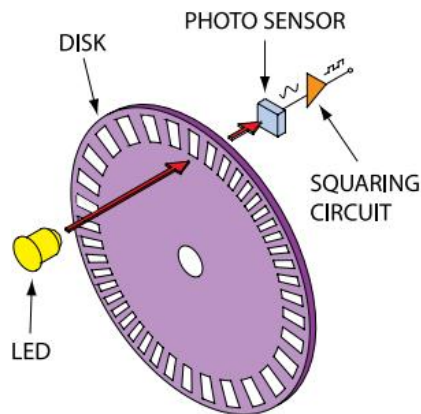
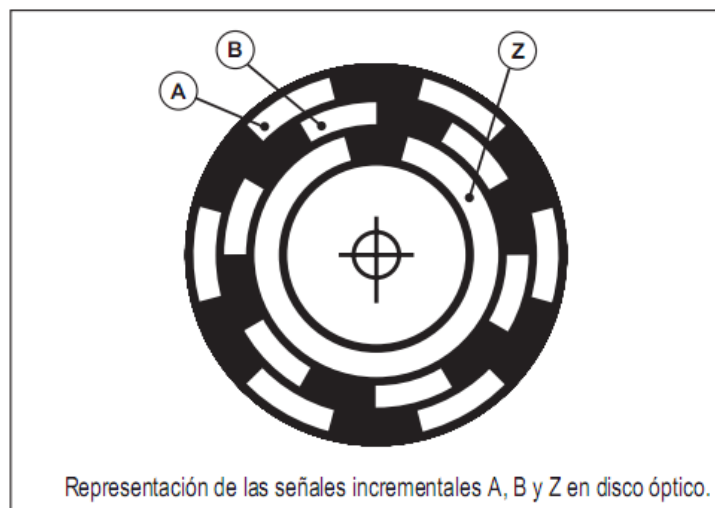


Figura 2.1.3.1. Encoder Óptico. Tomada de [www.encoder.com](http://www.encoder.com). Optical Encoders and Accesories (13/01/11).

Un encoder incremental proporciona generalmente dos formas de ondas cuadradas y desfasadas entre si 90° eléctricos, llamados comúnmente “canal A” y “canal B”. Además puede estar disponible otra señal llamada “canal Z”, “cero” o “index”; que provee la posición absoluta del cero del eje del encoder.



Representación de las señales incrementales A, B y Z en disco óptico.

Figura 2.1.3.2. Representación de señales incrementales en disco óptico. Tomada de Eltra, Encoder Incremental. Descripción General.<sup>[10]</sup>

### Codificación Incremental<sup>[11]</sup>

Los encoders incrementales proveen un número específico de pulsos equidistantes por revolución (PPR). Un solo canal de salida se utiliza para aplicaciones donde la dirección del movimiento no es importante, en caso de que esta se requiera, se utiliza la salida de cuadratura, con 2 canales desfasados 90° eléctricos y según la relación de fase entre ellos se determina la dirección del movimiento.

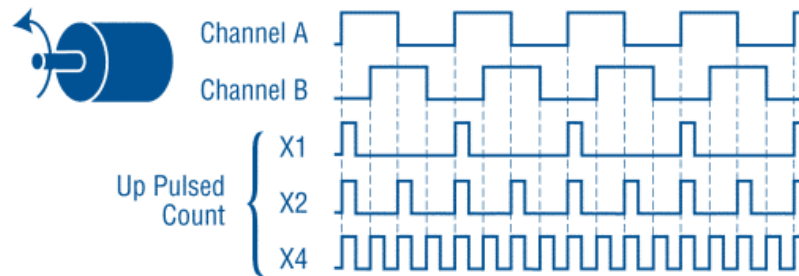


Figura 2.1.3.3. Codificación Incremental. Tomada de Encoder Application Handbook. Danaher Industrial Controls.

Cuando se requiera más resolución es posible que el contador de pulsos cuente los bordes anterior y posterior del tren de pulsos de un canal y así se duplica (x2) el número de pulsos contados por rotación. Contando ambos bordes de ambos canales, se obtiene una resolución x4.

La salida de un encoder incremental indica movimiento. Para determinar posición, sus pulsos deben ser acumulados por un contador. La cuenta esta propensa a errores durante un corte de energía o señales eléctricas transitorias, por ello al iniciar el equipo se debe ajustar una posición de referencia o cero para inicializar los contadores de posición. Algunos encoders incrementales producen una señal conocida como cero, marker o index; esta señal a menudo se utiliza para buscar una posición específica o absoluta.

### Resolución y Exactitud<sup>[11]</sup>

Resolución es el número de segmentos o unidades de medida por cada ciclo del eje del encoder. Se pueden encontrar encoders con resoluciones de hasta 10.000 pulsos por revolución (PPR), y si se usa la detección de bordes de subida y bajada de los canales A y B, pueden lograr resoluciones de hasta 40.000 PPR. El encoder seleccionado debe tener una resolución igual o mayor que la requerida por la aplicación.

Resolución y Exactitud son dos términos diferentes, y es posible tener una sin la otra. La figura 2.1.3.4 muestra una distancia X dividida en 24 incrementos o bits. X representa 360° o una revolución completa dividida en 24 partes.

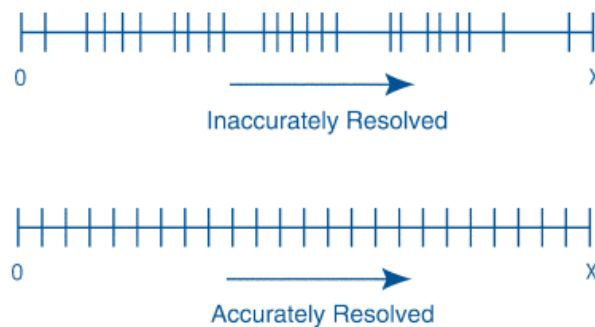


Figura 2.1.3.4. Resolución y exactitud en encoder. Tomada de Encoder Application Handbook. Danaher Industrial Controls.



En la recta de arriba de la figura 2.1.3.4, mientras hay 24 bits de resolución, las 24 partes no son uniformes. Este transductor no puede ser usado para medir posición, velocidad o aceleración con exactitud.

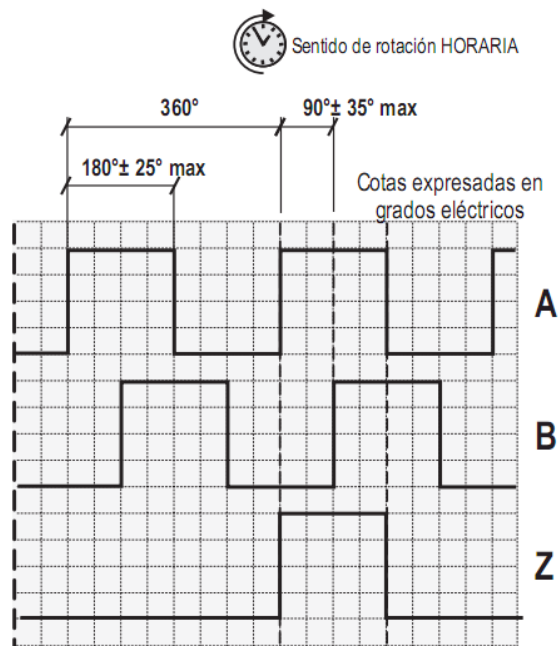
En la otra recta de la misma figura, la distancia X está dividida en 24 partes iguales. Cada incremento representa exactamente 1/24 de una revolución. Este transductor opera con exactitud y con buena resolución. La exactitud, sin embargo, puede ser independiente de la resolución; un transductor puede tener una resolución de sólo dos partes por revolución, mientras que su exactitud puede ser de 6 segundos de arco.

La precisión de un encoder incremental depende de factores mecánicos y eléctricos entre los cuales, el error de división del retículo, la excentricidad del disco, la de los rodamientos, el error introducido por la electrónica de lectura, imprecisiones de tipo óptico<sup>1</sup>.

La unidad de medida para definir la precisión de un encoder es el grado eléctrico, este determina la división de un impulso generado por el encoder: en efecto los 360° eléctricos corresponden a la rotación mecánica del eje, necesaria para hacer que se realice un ciclo o impulso completo de la señal de salida. Para saber a cuántos grados mecánicos corresponden 360 grados eléctricos es suficiente con la siguiente fórmula<sup>1</sup>:

$$360^\circ \text{ eléctricos} = \frac{360^\circ \text{ mecánicos}}{\text{No. impulsos/giro}} \quad \text{Eq. 2.1.3.5}^1$$

El error de división en un encoder, está dado por el máximo desplazamiento expresado en grados eléctricos, de dos frentes de onda consecutivos. Este error existe en cualquier encoder y se debe a los factores antes citados. A continuación se puede ver una gráfica con los errores típicos para un encoder, para este caso en particular, estos se refieren al fabricante de encoders Eltra<sup>1</sup>.



<sup>1</sup> Tomado de Eltra. Encoder Incremental<sup>[10]</sup>.

## 2.2 CONVERSIÓN RESOLVER A DIGITAL

Con el objeto de obtener en formato digital la posición angular, es necesario convertir y almacenar las señales en formato resolver, esto se logra usando técnicas para convertir de resolver a señales digitales. En este caso se usó la conversión por *tracking* o seguimiento, empleada por el conversor AD2S1205, del cual se explicarán sus principales características más adelante.

### 2.2.1 LAZO DE SEGUIMIENTO (SERVO LOOP) TIPO II:

La tipo de conversión realizada por el conversor AD2S1205 utiliza un lazo de seguimiento tipo II.

Existen tres tipos de lazos de seguimiento, el tipo 0, tipo I y tipo II, básicamente el número se refiere al número de integradores dentro del lazo. Un tipo 0 no tiene integradores, solamente una ganancia proporcional independiente de la frecuencia, si la entrada aplicada al lazo es una rampa de posición (velocidad constante), el error se incrementaría al mismo tiempo que la velocidad aumenta, así que la rampa de salida tendría una pendiente diferente. Un lazo tipo I tiene un integrador, si la entrada es una rampa en posición, la salida será una rampa en posición del mismo valor (velocidad), pero atrasada en posición, esto debido a que un integrador generada a la salida una rampa de posición (velocidad) con un error constante (voltaje). Las ventajas del lazo tipo I respecto a el lazo tipo 0 son, error 0 en estado estacionario y una respuesta precisa a entradas con velocidad constante (rampas de posición).

Un lazo tipo II tiene 2 integradores, cuando se considera una entrada rampa en posición, se consideran dos inferencias lógicas. El segundo integrador debe tener una entrada constante para proporcionar una velocidad de salida. Esto quiere decir que el primer integrador necesita entregar una señal constante al segundo integrador, así la entrada del primer integrador debe ser 0. Si la entrada al primer integrador es 0, entonces no hay error de posición en el sistema a velocidad constante. Así, un sistema tipo II, adicionalmente a no tener error de posición en reposo, tampoco lo tiene con entrada rampa de posición en estado estacionario (después de que la aceleración termina).

### 2.2.2 CONVERSION POR TRACKING O SEGUIMIENTO<sup>[15]</sup>:

En este tipo de conversión las dos salidas del resolver son conectadas a los multiplicadores seno y coseno como se observa en la Figura 2.2.1.1. Se asume que el estado actual del contador *up/down* es un número digital que representa el ángulo de prueba,  $\phi$ . El convertidor busca ajustar el ángulo digital,  $\phi$ , siguiendo continuamente a  $\theta$ , el ángulo análogo que se está midiendo, hasta llegar a ser igual a este.

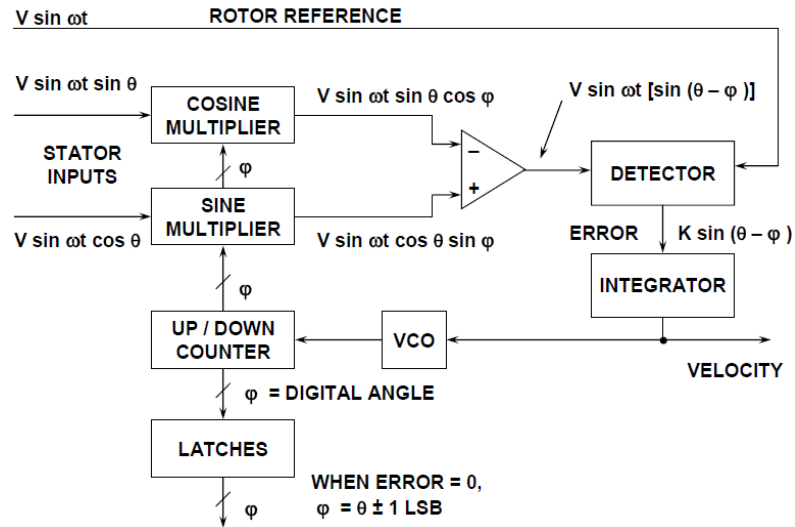


Figura 2.2.2.1 Conversión Resolver a Digital<sup>[15]</sup>.

Los voltajes de salida del estator se escriben como:

$$V_1 = V (\sin \omega t) (\sin \theta) \quad \text{Eq. 2.2.2.1}$$

$$V_2 = V (\sin \omega t) (\cos \theta) \quad \text{Eq. 2.2.2.2}$$

donde  $\theta$  es el ángulo del rotor del resolver. El ángulo digital  $\phi$  se aplica al multiplicador coseno, y este coseno es multiplicado por  $V_1$  para obtener el término:

$$V (\sin \omega t) (\sin \theta) (\cos \phi) \quad \text{Eq.2.2.2.3}$$

El ángulo digital  $\phi$  también es aplicado al multiplicador seno y multiplicado por  $V_2$  para obtener el término:

$$V (\sin \omega t) (\cos \theta) (\sin \phi) \quad \text{Eq. 2.2.2.4}$$

Estas dos señales son restadas entre sí, por el amplificador para obtener una señal ac de error de la forma:

$$V (\sin \omega t) [(\sin \theta) (\cos \phi) - (\cos \theta) (\sin \phi)] \quad \text{Eq. 2.2.2.5}$$

Usando una identidad trigonométrica, esta ecuación se reduce a:

$$V (\sin \omega t) [\sin (\theta - \phi)]. \quad \text{Eq. 2.2.2.6}$$

El detector síncrono demodula esta señal ac de error, usando como referencia el voltaje del rotor del resolver. Esto resulta en una señal dc de error proporcional a  $\sin (\theta - \phi)$ .

La señal de error alimenta el integrador, el cual a la salida genera un oscilador controlado por voltaje (VCO). El VCO, hace que el contador *up/down* cuente en la dirección apropiada hasta obtener:

$$\sin (\theta - \phi) \rightarrow 0 \quad \text{Eq. 2.2.2.7}$$

Cuando esto es logrado,

$$\theta - \phi \rightarrow 0, \quad \text{Eq. 2.2.2.8}$$

y por lo tanto,

$$\phi = \theta \quad \text{Eq. 2.2.2.9}$$

dentro de un conteo. De esta forma, la salida del contador digital,  $\phi$ , representa el ángulo  $\theta$ . A través de los *latches*<sup>2</sup> permiten que esta información sea transferida externamente sin interrumpir el lazo de seguimiento.

Este circuito es equivalente a un lazo de seguimiento tipo II, debido a sus 2 integradores. Uno es el contador, el cual acumula pulsos y el otro es el integrador a la salida del detector.

Debido a que la conversión de resolver a digital depende únicamente de la diferencia de amplitudes de las señales de entrada, las atenuaciones causadas en las conexiones al resolver no afectan su desempeño significativamente. Aun más, pueden operar hasta con una distorsión armónica del 10% en las señales de entrada.

Adicionalmente, ya que la conversión por tracking realiza una doble integración del error de la señal, ofrece un alto grado de inmunidad al ruido. Aunque el área neta bajo cualquier pico de ruido produce error, típicamente estos picos de ruido tienen iguales formas de onda positiva y negativa, por lo tanto cuando se integran, resulta en un error neto de señal igual a cero. La inmunidad al ruido, combinada con la insensibilidad a las caídas de voltaje, permiten al usuario ubicar el conversor a distancias considerables del resolver.

### 2.3 MICROCONTROLADOR dsPIC<sup>[12]</sup>

Un DSP es un procesador de señales digitales muy rápido y poderoso, capaz de procesar datos en tiempo real, con velocidades máximas de hasta 30 MIPS. Existe una gran variedad de ellos para pequeños y grandes requerimientos.

La mayoría de procesadores DSP son diseñados con las mismas operaciones básicas, por lo que comparten el mismo conjunto de características básicas. Estas son: alta velocidad de cálculos aritméticos, transferencia de datos en tiempo real y arquitecturas de memoria de múltiple acceso. Las principales funciones son: suma y multiplicación, retrasos y manejo de arreglos.

La mayoría de operaciones requieren que sumas y multiplicaciones se realicen simultáneamente, por lo que existe hardware para suma y multiplicación que se puede usar en paralelo con una sola instrucción. Los *delays* o retrasos, requieren que los valores intermedios sean mantenidos para su uso posterior y los DSPs tienen muchos registros que pueden ser utilizados para mantener valores intermedios.

---

<sup>2</sup> Un latch es un circuito electrónico usado para almacenar información en sistemas lógicos asíncronos.

El manejo de arreglos requiere que los datos sean eficientemente adquiridos desde posiciones de memoria consecutivas, esto implica la generación de la próxima dirección de memoria, otra característica de los DSPs, así como la capacidad de generar nuevas direcciones de forma eficaz.

## 2.4 CONVERTOR AD2S1205<sup>[14]</sup>

El ADS1205 es un conversor de resolver a digital que emplea el método de *tracking*, cuenta con una resolución de 12 bits y con un oscilador sinusoidal que proporciona la señal seno que alimenta el rotor del synchro/resolver. Un lazo de seguimiento tipo II, el cual se explicó anteriormente, es empleado para rastrear las señales de entrada Seno y Coseno y convertirlas en una representación digital del ángulo y velocidad de entrada. El desempeño del AD2S1205 está garantizado para un rango de frecuencias de cristal externo de  $8.192 \text{ MHz} \pm 25\%$ , permitiendo una velocidad máxima de rastreo de 1250 rps.

Las características más importantes del AD2S1205 se describen en la tabla 3.4.1. La señal sinusoidal de excitación (EXC) para el synchro - resolver y su complemento (EXC') tiene 4 frecuencias programables, en donde 10 kHz es la frecuencia por defecto. La amplitud de nesta señal está centrada 2.5 V y tiene una amplitud de 3.6 V p-p. Se puede acceder de 2 formas a los datos de la posición angular de 12 bits, ya sea, a través de un puerto paralelo o una interfaz serial de 3 terminales. Por otra parte, el AD2S1205 emula un encoder incremental de 1024 pulsos, en el formato estándar con los canales A y B en cuadratura, y el terminal que indica la dirección de movimiento, además, cuenta con un terminal que genera un pulso cuando la posición angular absoluta pasa por cero.

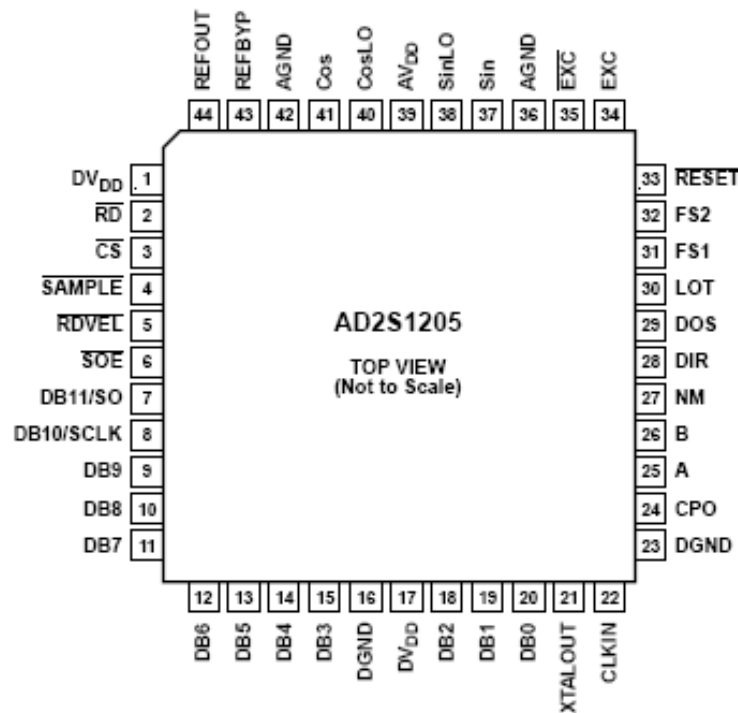


Figura. 2.4.1 Distribución de terminales del AD2S1205<sup>[14]</sup>.

La Figura 2.4.1 muestra la distribución de terminales dentro del conversor. Con los terminales No. 31 (FS1) y No. 32 (FS2) se selecciona la frecuencia de la señal de referencia del synchro - resolver. La interfaz paralela se selecciona manteniendo en alto el terminal No. 6 (SOE') y sus salidas son los terminales No. 20 (DB0) a el No. 7 (DB11). Los datos de los integradores de posición y velocidad son transferidos a los registros de posición y velocidad, respectivamente, después de una transición de alto a bajo en el terminal No. 4 (SAMPLE'). El terminal No. 5 (RDVEL') selecciona entre el registro de posición y el de velocidad para transmitir la información al registro de salida. El terminal No. 3 (CS') debe mantenerse en bajo para transferir la información del registro seleccionado al registro de salida. La interfaz serial se accede con los terminales No. 7 (SO), No. 2 (RD') y No. 8 (SCLK), la cual opera a una velocidad de reloj de 25 MHz y es compatible con el protocolo SPI. La interfaz serial se secciona manteniendo en bajo el terminal No. 6 (SOE'). Los terminales No. 25 (A) y No. 26 (B) son las salidas del emulador de encoder incremental y los terminales No. 28 (DIR) y No. 29 (NM), indican la dirección y la posición de cero absoluto, respectivamente.

### Respuesta del Modelo del Conversor:

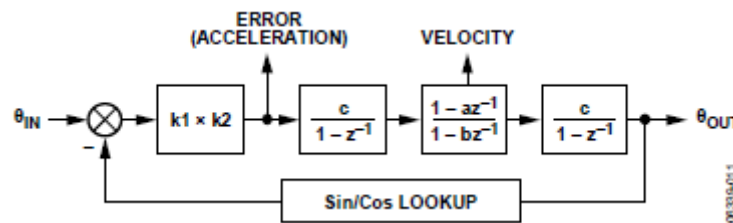


Figura 2.4.2 Diagrama de Bloques de la Respuesta del sistema del Conversor<sup>[14]</sup>.

El conversor AD2S1205 es un dispositivo que usa dos ADCs para digitalizar las señales del resolver y un lazo de seguimiento tipo II para convertir estas señales en términos de posición digital y velocidad.

La primera ganancia consiste en la ganancia del ADC en las entradas Sin/Cos y la ganancia de la señal de error hacia el primer integrador. El primer integrador genera una señal proporcional a la velocidad. El filtro de compensación contiene un polo y un cero que son usados para proporcionar margen de fase y reducir ruido de alta frecuencia. El segundo integrador es igual al primero y genera la salida de posición de la señal de velocidad. El *Sin/Cos lookup* tiene ganancia unitaria.

La función de transferencia del conversor en lazo abierto es:

$$G(z) = k_1 * k_2 * I(z)^2 * C(z)$$

La función de transferencia del conversor en lazo cerrado es:

$$H(z) = \frac{G(z)}{1 + G(z)} = \frac{\theta_{IN}}{\theta_{OUT}}$$

Las respuestas en magnitud y fase del lazo cerrado son las de un filtro pasa bajas de segundo orden y se muestran en las figuras 2.4.3 y 2.4.4.

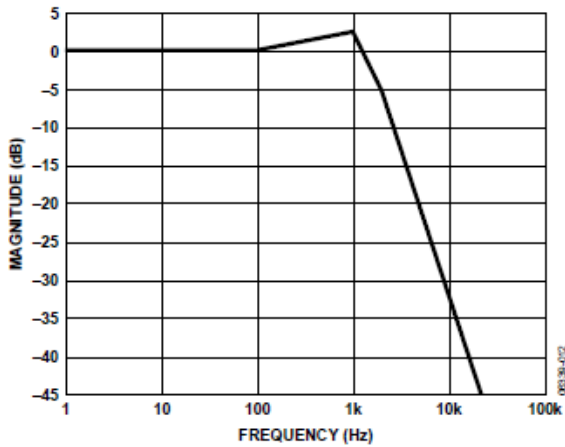


Figura 2.4.3 Respuesta en magnitud del conversor<sup>[14]</sup>.

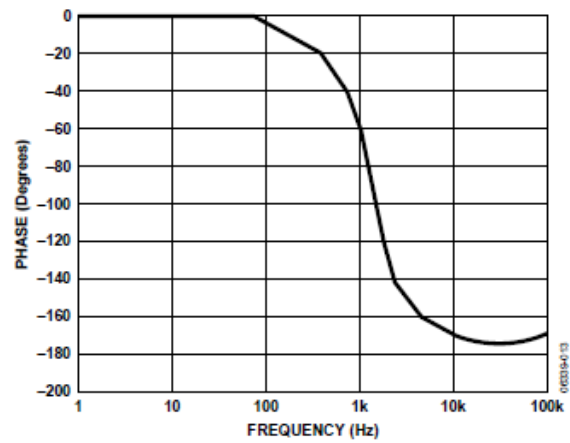


Figura 2.4.4 Respuesta en fase del conversor<sup>[14]</sup>.

## 2.5 MOTOR ELÉCTRICO<sup>[13]</sup>

Probablemente el uso más sencillo sea el de motores DC, dado que es relativamente fácil encontrar algunos de este tipo con un bajo consumo de potencia y lograr un movimiento adecuado para el proceso a controlar, gracias a que es posible acoplarlos a motorreductores, además son fáciles de controlar a partir de los módulos de PWM de microcontroladores comerciales.

Los polos magnéticos del campo de la armadura intentarán alinearse con los polos magnéticos opuestos, generados por el campo del estator. La sección del rotor donde la electricidad entra en los embobinados del rotor se llama conmutador. La electricidad es conducida entre el rotor y el estator por las escobillas de cobre o los conductos de grafito (localizados sobre el rotor), los cuales están en contacto con los anillos que se encuentran sobre el estator.

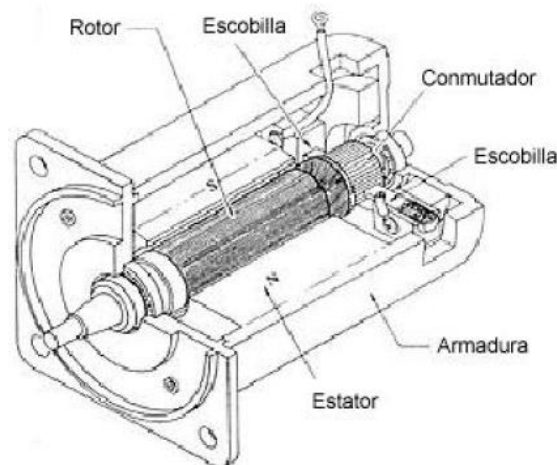
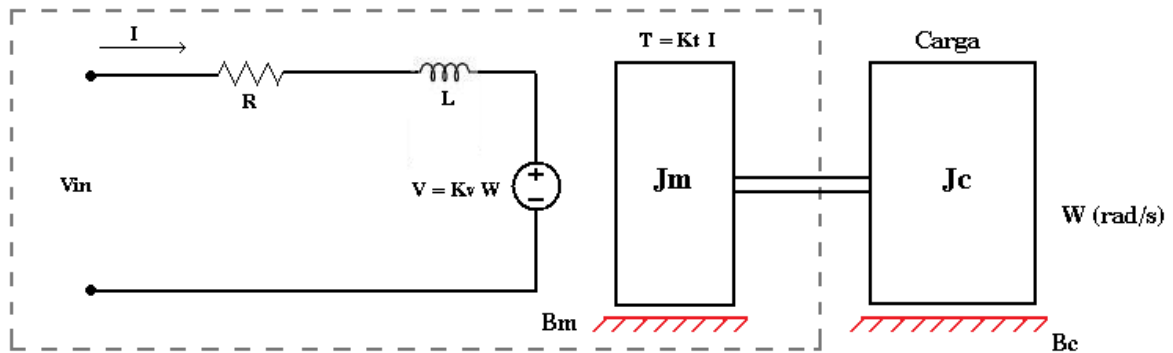


Figura 2.5.1. Motor DC.

### Modelamiento del Motor DC:

El modelo del motor se puede dividir en una sección eléctrica y una mecánica, con base en sus características, de la siguiente forma:



Donde:

- L: Inductancia de los devanados [mH]
- J: Momento de inercia del rotor [kg m<sup>2</sup>]
- Bt: Fricción Viscosa [Nm s/rad]
- Kv: Constante del generador [V/rad/s]
- Kt: Constante de torque [Nm/A]
- R: Resistencia de los devanados [Ω]
- ω: Velocidad angular [rad/s]
- θ: Posición angular [rad]

Circuito eléctrico:

$$V_{in}(t) = I_{\alpha}(t)R_{\alpha} + L \frac{dI_{\alpha}(t)}{dt} + K_v w(t) \quad \text{Eq. 2.5.1}$$

Balance de fuerzas (torques) a la salida:

$$I_{\alpha}(t)K_T = Bw(t) + J \frac{dw(t)}{dt} \quad \text{Eq. 2.5.2}$$

La posición es:

$$\theta = \int_0^t w(t)dt + \theta(0) \quad \text{Eq. 2.5.3}$$

La ecuación diferencial para w(t) de Eq. 2.5.1 y Eq. 2.5.2 queda:

$$V_{in}(t) = \frac{1}{K_T} \left( Bw(t) + J \frac{dw(t)}{dt} \right) R_{\alpha} + L \frac{d(Bw(t) + J \frac{dw(t)}{dt})}{dt} \frac{1}{K_T} + K_v w(t)$$

$$V_{in}(t) = \left( R_{\alpha} \frac{B}{K_T} + K_v \right) w(t) + \left( R_{\alpha} \frac{J}{K_T} + \frac{BL}{K_T} \right) \frac{dw(t)}{dt} + \left( \frac{JL}{K_T} \right) \frac{d^2w(t)}{dt^2} \quad \text{Eq. 2.5.4}$$

Aplicando transformada de Laplace:



$$V_{in}(S) = \left(R_a \frac{B}{K_T} + K_v\right)W(S) + S \left(R_a \frac{J}{K_T} + \frac{BL}{K_T}\right)W(S) + S^2 \left(\frac{JL}{K_T}\right)W(S)$$

$$\frac{W(S)}{V_{in}(S)} = \frac{1}{\left(R_a \frac{B}{K_T} + K_v\right) + S \left(R_a \frac{J}{K_T} + \frac{BL}{K_T}\right) + S^2 \left(\frac{JL}{K_T}\right)}$$

Simplificando:

$$\frac{W(S)}{V_{in}(S)} = \frac{K_T}{(R_a B + K_v K_T) + S(R_a J + BL) + JLS^2}$$

Eq. 2.5.5

## 2.6 PUENTE H<sup>[13]</sup>

El circuito denominado puente H recibe su nombre debido a que su esquema se asemeja a la forma de la letra “H” y su función principal es suministrar la corriente necesaria para mover el motor DC, utilizando una sola fuente de alimentación.

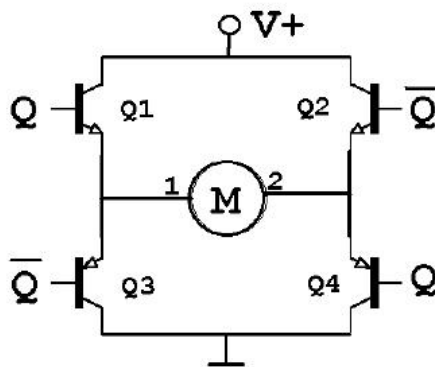


Figura 2.6.1 Configuración básica de un puente H

El puente H es alimentado por las señales del pwm, en la figura 2.6.2 en color azul y rojo. Como se observa en la Figura 2.6.1, la señal de control Q controla el paso de corriente al motor de la siguiente forma: cuando la señal Q es uno lógico, Q2 y Q3 están inactivos, mientras la corriente circula por Q1 y Q4, fluyendo de los terminales 1 al 2 del motor; cuando Q es cero lógico ocurre lo contrario, Q4 y Q1 están inactivos, mientras la corriente circula por Q2 y Q3, fluyendo de los terminales 2 al 1 del motor, haciendo que el eje del motor cambie su sentido de giro. La señal de salida típica del puente H se observa en color negro en la figura 2.6.2

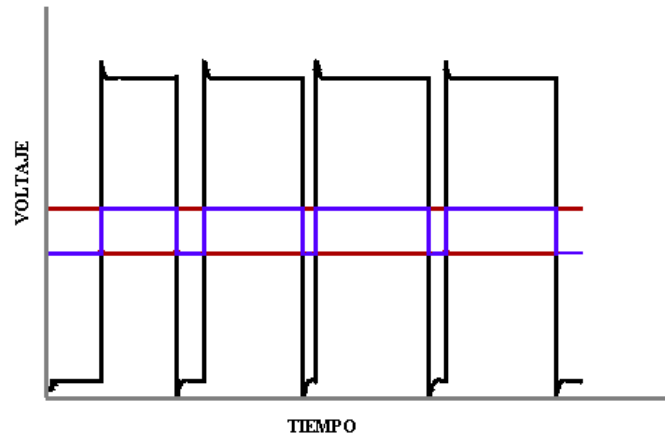


Figura 2.6.2 Formas de Onda de entrada y salida del Puente H.

### 3. ESPECIFICACIONES

#### 3.1 DESCRIPCIÓN GENERAL

El sistema de medición y control de posición angular utilizando Synchro-Resolver y encoder, recibe la referencia de posicionamiento angular para el motor. Esta referencia es comparada con la posición actual del motor, la cual puede ser sensada utilizando Synchro - Resolver o encoder. El Synchro-Resolver entrega 2 señales sinusoidales desfasadas entre sí  $90^\circ$  y cuya amplitud, en conjunto corresponde a la posición angular del rotor del Synchro - Resolver. Este conversor, mediante el método de *tracking* o seguimiento, convierte dichas señales en información digital de 12 bits correspondiente a la posición angular del rotor del Synchro - Resolver. Si se dispone de un resolver, simplemente se deben cumplir las especificaciones requeridas por el conversor para las entradas de resolver y conectar directamente las salidas del resolver al conversor. Por otra parte, se puede seleccionar el encoder del motor como sensor para el sistema, el cual entrega dos canales en cuadratura. La información proveniente del encoder, es recibida por el dsPIC, en donde se encuentra implementado un control PID y mediante una salida de PWM conectada a un puente H es posicionado el motor. El conversor también es conectado al dsPIC, entregándole información referente a la posición angular del motor medida por el resolver.

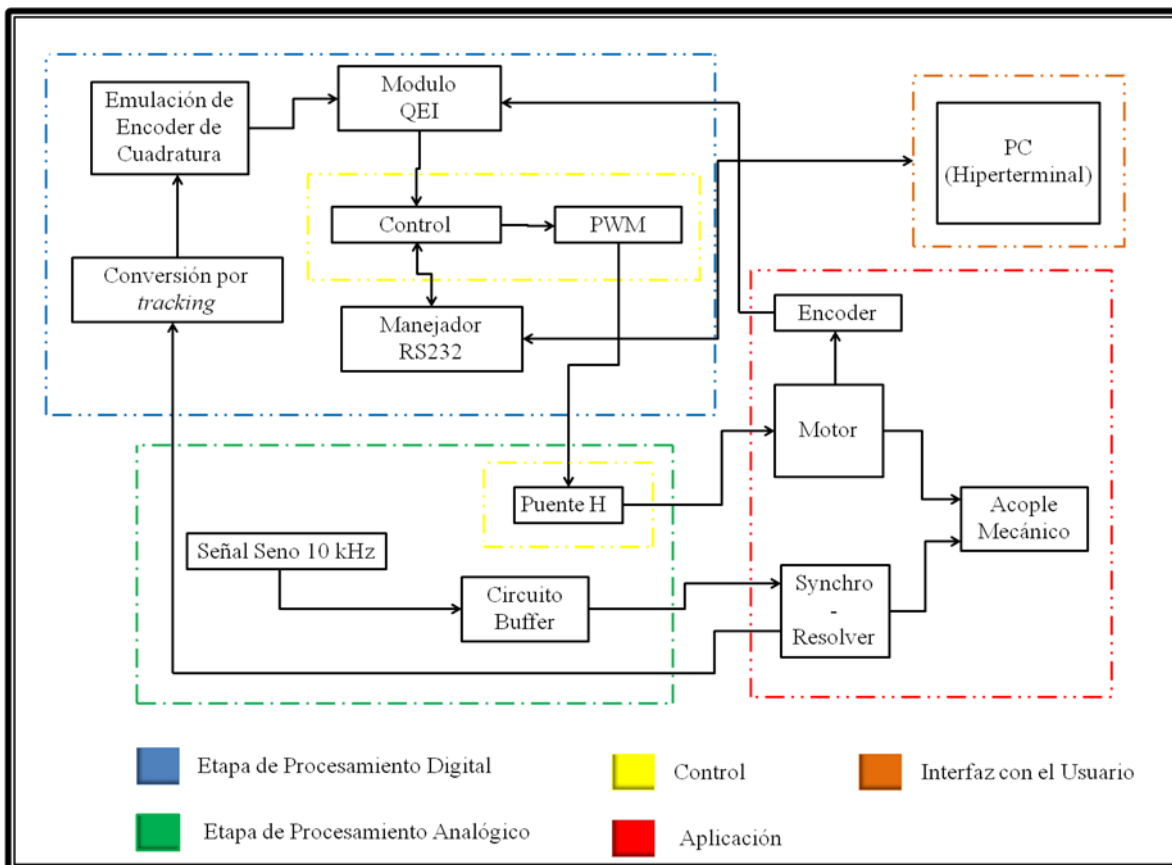


Figura 3.1.1 Diagrama en Bloques del sistema.

En la siguiente tabla se encuentran las características más relevantes del sistema de medición y control angular.

<b>Características Eléctricas</b>	
<b>PARÁMETROS</b>	<b>VALOR</b>
<b>Alimentación Externa</b>	<b>Fuente Dual Regulada DC 0 - 24 V, 0 - 3 A</b>
<b>Frecuencia de Procesamiento (AD2S1205)</b>	<b>8.192 MHz</b>
<b>Frecuencia de Procesamiento (dsPIC30F2010) cristal x8</b>	<b>80 MHz</b>
<b>Temperatura Máxima</b>	<b>40°</b>
<b>Características Mecánicas</b>	
<b>Acople entre motor y Synchro – Resolver (Discos de 64 dientes)</b>	<b>1:1</b>
<b>Acople entre motor y encoder</b>	<b>1:1</b>
<b>Resolución máx. con Synchro - Resolver (1024 pulsos x2(QEI) (Emulador Encoder de Cuadratura)</b>	<b>0.17°</b>
<b>Resolución máx. con encoder (500 pulsos x2(QEI))</b>	<b>0.36°</b>
<b>Características de Control</b>	
<b>Sensores</b>	<b>Synchro - Resolver Encoder</b>
<b>Medida del error</b>	<b>Digital</b>
<b>Variable Controlada</b>	<b>Posición</b>
<b>Actuador</b>	<b>Puente H</b>
<b>Características de la Interfaz con el Usuario</b>	
<b>Software</b>	<b>Hyperterminal</b>
<b>Variables</b>	<b>Referencia, Consts. Controlador; P,I,D, Selección control según sensor</b>
<b>Comunicación</b>	<b>Serial</b>
<b>Requisitos</b>	<b>Windows XP</b>
<b>Características de Comunicación</b>	
<b>Velocidad</b>	<b>115200 baudios/seg</b>
<b>Puerto</b>	<b>COM1</b>
<b>Paridad</b>	<b>Ninguna</b>
<b>Control de flujo</b>	<b>Ninguno</b>

Tabla 3.1.1 Características principales del proyecto.

## 3.2 DIAGRAMA EN BLOQUES

El diagrama de la Figura 3.1.1 muestra el esquema general del control de posición angular usando synchro-resolver y encoder.

### 3.2.1 *Etapa de Aplicación:*

Es la etapa donde se encuentra el montaje mecánico entre la planta, es decir, el motor, y los sensores de posición angular; el Synchro - resolver y el encoder incremental. En el caso del encoder, este se encuentra acoplado directamente al eje del rotor del motor; para el synchro - resolver se realizó un acople al eje del motor con una relación de 1:1.

### 3.2.2 *Etapa de Procesamiento Analógico:*

En esta etapa, el AD2S1205, genera la señal seno de 10 kHz con la que se alimentara la referencia del synchro - resolver a través de una etapa de salida instalada en la tarjeta de desarrollo para AD2S1205. También se encuentra el driver L6203, el cual que alimentará el motor.

### 3.2.3 *Etapa de Procesamiento Digital:*

En esta etapa el dsPIC30F2010 recibe las señales de cuadratura del encoder incremental (acoplado al motor pittman 9233 24V), a través del módulo QEI. El AD2S1205 recibe las señales de posición angular en formato resolver y las convierte a información digital por el método de *tracking* o seguimiento, luego estos datos son convertidos por el emulador de encoder incremental a señales en cuadratura A y B. De igual manera estas señales son recibidas por el módulo QEI del dsPIC30F2010. En esta etapa también se encuentra el algoritmo que transforma la información recibida por el módulo QEI a una variable que indica el valor del ángulo de posición del motor interpretado por el módulo QEI. Por último está el circuito integrado manejador del estándar RS-232, encargado de la transmisión serial entre el dsPIC30F2010 y el PC donde se encuentra la interfaz con el usuario

### 3.2.4 *Etapa de Control:*

En esta etapa esta implementado el control digital, se recibe el dato del valor del ángulo de posición del motor y se compara con el valor de la referencia, luego este dato es procesado por el control hasta disminuir la señal de error a cero, ajustando el valor de salida de la señal PWM del dsPIC30F2010, la cual controlara el movimiento del motor. Esta señal de PWM es entregada al driver L6203, para que suministre la potencia suficiente para el motor.

### 3.2.5 *Etapa de Interfaz con el Usuario:*

La interfaz con el usuario se realiza a través del software de comunicaciones, Hyperterminal. Esta tiene 3 funciones principales: el usuario puede digitar el valor de referencia del ángulo deseado de movimiento del motor, de igual forma puede escoger las constantes del control y el sensor con el cual se desea medir el movimiento angular del rotor del motor. La interfaz se explica con mayor profundidad en el numeral 4.7

### 3.3 Motor PITTMAN 9233 24 V<sup>[16]</sup>:

Para la planta del sistema se utilizó un motor PITTMAN 9233 de 24V con las siguientes características:

Motor Data	Symbol	Units	24.0 V
			Supply Voltage (Reference)
Continuous Torque	$T_c$	oz-in	4.7
		Nm	0.033
Speed @ Cont. Torque	$S_c$	rpm	5410
Current @ Cont. Torque	$I_c$	A	1.35
Continuous Output Power	$P_{o,c}$	W	19
Motor Constant	$K_M$	oz-in/ $\sqrt{W}$	2.7
		Nm/ $\sqrt{W}$	0.019
Torque Constant	$K_T$	oz-in/A	5.28
		Nm/A	0.0373
Voltage Constant	$K_E$	V/krpm	3.90
		V s/rad	0.0373
Terminal Resistance	$R_{mt}$	$\Omega$	3.94
Inductance	L	mH	3.29
No-Load Current	$I_{nl}$	A	0.15
No-Load Speed	$S_{nl}$	rpm	5880
Peak Current (Stall)	$I_{pk}$	A	6.09
Peak Torque (Stall)	$T_{pk}$	oz-in	31.4
		Nm	0.222
Coulomb Friction Torque	$T_f$	oz-in	0.60
		Nm	0.0042
Viscous Damping Factor	D	oz-in/krpm	0.034
		Nm s/rad	2.3E-06
Electrical Time Constant	$\tau_e$	ms	0.84
Mechanical Time Constant	$\tau_m$	ms	9.2
Thermal Time Constant	$\tau_{th}$	min	11
Thermal Resistance	$R_{th}$	$^{\circ}C/W$	19
Max. Winding Temperature	$\theta_{max}$	$^{\circ}C$	155
Rotor Inertia	$J_r$	oz-in-sec <sup>2</sup>	4.6E-04
		kg m <sup>2</sup>	3.2E-06
Motor Weight (Mass)	$w_M$	oz	8.90
		g	252

Tabla 3.3.1 Especificaciones del motor Pittman 9233 24V.

#### 3.3.1 Encoder incremental óptico E30<sup>[16]</sup>:

El motor pittman 9233 utilizado en el proyecto se adquirió con encoder óptico ensamblado de fábrica. Este encoder es de referencia E30 con resolución de 500 pulsos por vuelta, a continuación se muestran las especificaciones, para ver en detalle el modo de conexión, ver Anexo D.

E30A Encoder Data	Symbol	Units	Value
Available Resolutions	-	-	100, 200, 256, 360, 500, 512, 1000, 1024
Outputs	-	-	2-Channel Quadrature Output
Output Interface	-	-	TTL Compatible
Supply Voltage	$V_{CC}$	V	4.5 to 5.5
Supply Current	$I_{CC}$	mA	40 max
High Level Output Voltage	$V_{OH}$	V	2.4 min
Low Level Output Voltage	$V_{OL}$	V	0.4 max
Maximum Operating Frequency	$f_{max}$	kHz	100
Maximum Operating Temperature	$\theta_{max}$	°C	-40 to +100
Encoder Weight (Mass)	$W_E$	oz g	2.0 57

Tabla 3.3.1.1. Especificaciones Encoder E30.

### 3.4 CONVERTOR AD2S1205<sup>[15]</sup>:

El convertor AD2S1205 utilizado para realizar la conversión de formato resolver a digital, cuenta con las siguientes características más importantes:

Característica	Valor
<b>RESOLUCIÓN ANGULAR</b>	
<b>Resolución Exactitud</b>	12 bits ±11 Arc min
<b>Resolución Emulador de Encoder Incremental</b>	1024 pulsos
<b>SALIDAS EXC, EXC'</b>	
<b>Voltaje Central</b>	3.6 Vp-p 2.47 V
<b>Frecuencias Programables</b>	10 kHz, 12 kHz, 15 kHz, 20 kHz.
<b>THD</b>	-58 dB
<b>Velocidad máxima de tracking</b>	1250 rps

Tabla 3.4.1. Características del convertor AD2S1205.

### 3.5 Synchro Resolver Cliffton

Se adquirió un Synchro Resolver con las siguientes características:

Característica	Valor
Marca	Cliffton Precision Products
Referencia	HCY-8-L-70
Frecuencia Operación	400 Hz
Alimentación	15 VAC
Rotor	2 fases
Estator	2 fases
Costo	U\$ 32 (+ gastos envío)

Tabla 3.5.1. Características Synchro Resolver.

### 3.6 dsPIC30F2010<sup>[17]</sup>:

Para la realización de este proyecto se eligió el dsPIC30F2010 de Microchip, el cual cuenta con las siguientes características principales:

Característica	Valor
Frecuencia de Operación	Entrada de oscilador de 10 Mhz con PLL activo (4x, 8x, 16x)
Memoria de Programa (Bytes)	12 k
Memoria de Programa FLASH	4 k
Memoria SRAM (Bytes)	512
Memoria EEPROM (Bytes)	1 k
Fuentes de Interrupción	27 con 8 niveles de prioridad
Puertos de Entrada/Salida	6
Instrucciones	84
Temporizadores 16-bit	3
Módulos de PWM	6
Comunicaciones seriales	UART, SPI, I2C
Canales de conversión A/D	6 canales de 10 bits
<b>*Modulo de interfaz de encoder de cuadratura (QEI)</b>	si

Tabla 3.6.1. Características principales del dsPIC30F2010

A continuación se puede observar en la figura la distribución de los 28 pines del dsPIC30F2010:



## 28-Pin SDIP and SOIC

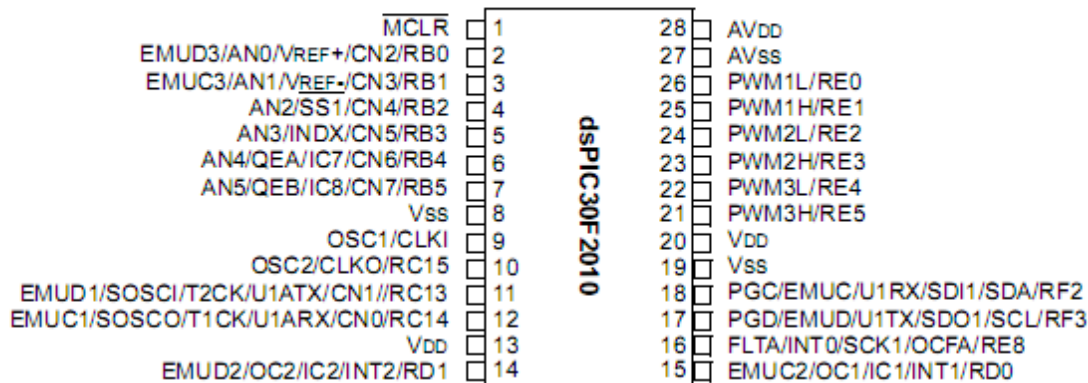


Figura 3.6.2 Pines del dsPIC30F2010

### 3.6.1 MODULO *QEI*<sup>[18]</sup>:

#### \*Interfaz de Encoder de Cuadratura:

El dsPIC30F2010 cuenta con un módulo llamado *Quadrature Encoder Interface (QEI)* ó Interfaz de Encoder de Cuadratura. Este módulo simplifica mucho la adquisición de señales en cuadratura provenientes de Encoders incrementales, ópticos, o un emulador de Cuadratura. Basta con inicializar el módulo y configurar sus registros, para obtener mediante operaciones matemáticas muy simples, la posición o velocidad angulares referentes al eje del Encoder o el transductor utilizado. Posteriormente esta posición actual se puede comparar con una posición deseada o de referencia y así realizar el esquema de control deseado para el sistema. Para la conexión del Encoder a través del módulo QEI al dsPIC, se encuentran disponibles los pines No. 5, 6 y 7.

El módulo QEI, provee una interfaz para Encoders Incrementales. Consiste en una lógica de decodificación de cuadratura para interpretar las señales fase A y fase B, y de acuerdo a esto aumentar o disminuir un contador. Filtros digitales a la entrada, acondicionan las señales de entrada. En la figura 2.4.3 se puede observar un diagrama en bloques simplificado del módulo.

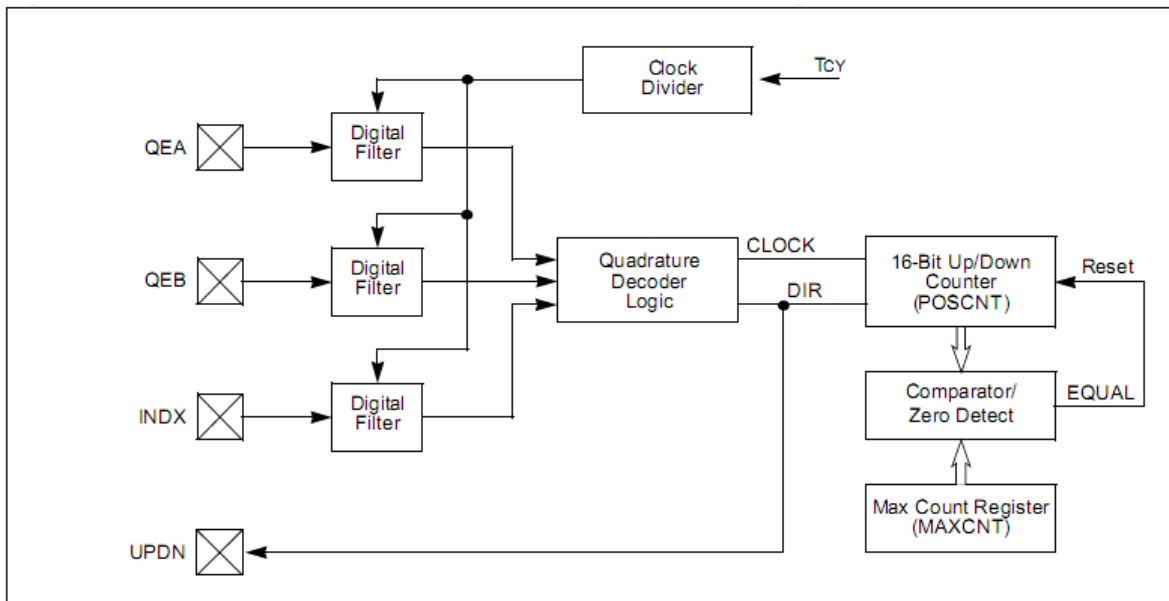


Figura 3.6.1.1. Diagrama en bloques simplificado del módulo QEI.

Las principales características del módulo QEI son:

- Tres terminales de entradas: fase A, fase B, e index.
- Filtros digitales de ruido programables en las entradas.
- Decodificador de cuadratura que proporciona pulsos al contador y dirección de conteo.
- Contador de 16 bits, de incremento o decremento.
- Estado de la dirección de conteo.
- Resolución de x2 o x4 (Para el proyecto se utilizó Resolución x2)
- 2 modos de reset para el contador.
- Interrupciones generadas por el QEI o eventos de conteo.

Los dos canales; fase A (QEA) y fase B (QEB), tienen una única relación. Si la fase A antecede a la fase B, entonces la dirección de movimiento (del motor) es positiva o hacia adelante. Si la fase B antecede a la fase A, la dirección es negativa o hacia atrás. Un tercer canal llamado pulso index, ocurre una vez por revolución y es usado para establecer una referencia absoluta. Las señales de cuadratura producidas por el encoder pueden tener cuatro estados únicos (01, 00, 10, 11), el orden de los estados se invierte cuando se cambia la dirección de rotación. El decodificador de cuadratura captura las señales de fase e index y convierte esta información en un conteo numérico de los pulsos de posición. Generalmente el conteo incrementa cuando el eje gira en una dirección y decrementa cuando el eje gira en la dirección contraria. La figura 2.4.4 muestra un diagrama de tiempo para estas tres señales.

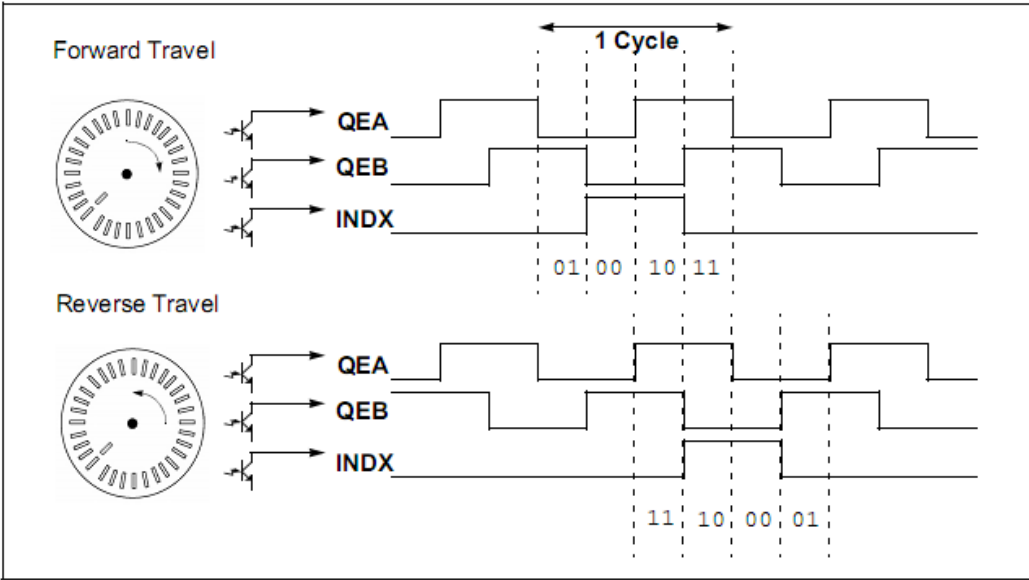


Figura 3.6.1.2. Diagrama de tiempo de las señales del QEI.

#### 4. DESARROLLOS

En la figura 4.1 se muestra el desarrollo completo del proyecto, de acuerdo con las partes mencionadas en el capítulo 3. Las etapas de funcionamiento están separadas en la tarjeta de desarrollo de dsPIC30F2010, en donde también se encuentra el puente H y el manejador de RS-232; la tarjeta del convertor AD2S1205 en la cual también esta implementado el circuito buffer; y la etapa mecánica, motor, encoder y synchro – resolver, en esta convergen las salidas de las fases anteriores.

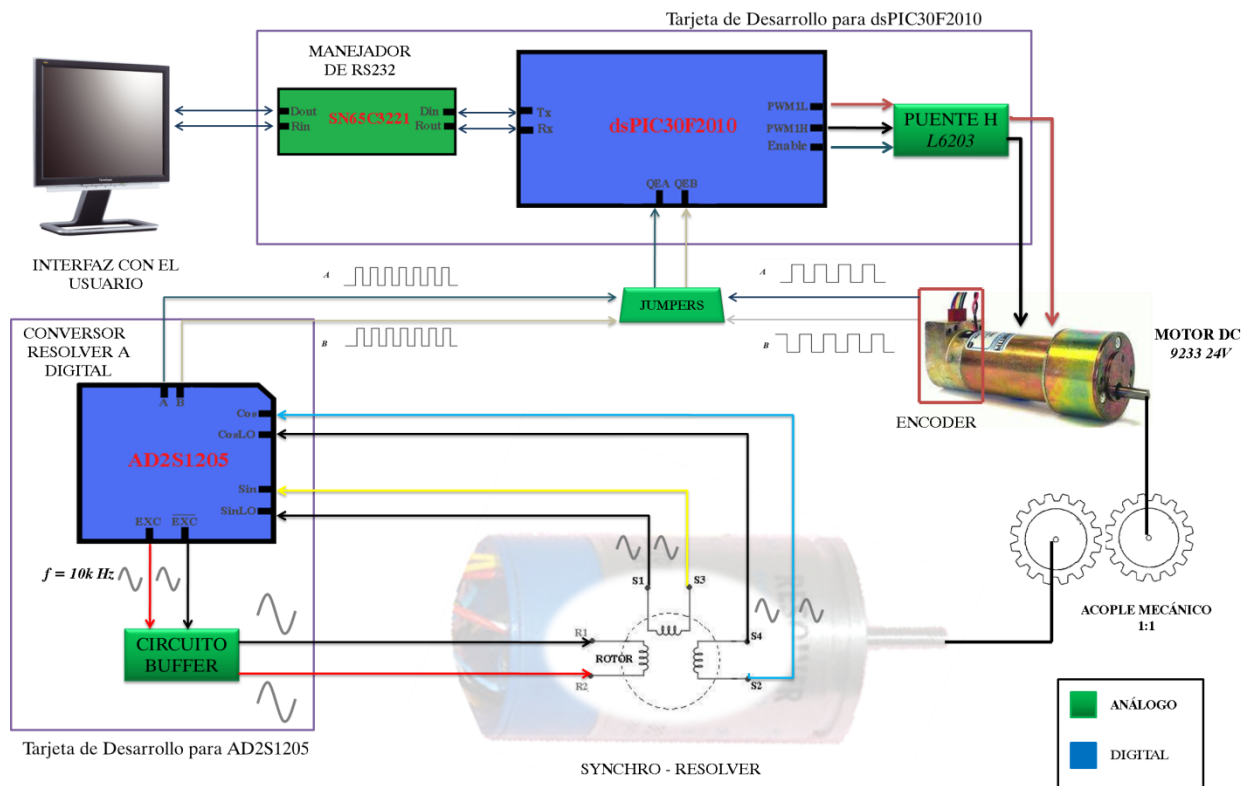


Figura 4.1. Esquema de desarrollo del sistema.

A continuación se describe los procedimientos realizados para implementar los controladores de posición para el motor DC, a partir del movimiento del rotor del synchro – resolver y encoder.

## 4.1 TARJETAS DE DESARROLLO:

Para la puesta en funcionamiento del convertor AD2S1205 y del dsPIC2010, se utilizaron 2 tarjetas de desarrollo, una para cada chip. Estas permiten que se acople todo el sistema de control el cual incluye: sensores, actuador (motor dc), microcontrolador y convertor análogo a digital.

### 4.1.1 Tarjeta de desarrollo para AD2S1205:

Esta es una tarjeta de diseño propio, basada en la hoja de especificaciones del AD2S1200 que también es funcional para el AD2S1205. Para su diseño se tuvieron en cuenta las recomendaciones de conexión del fabricante y criterios propios, apoyados por docentes del Departamento de Electrónica de la Pontificia Universidad Javeriana. En el Anexo E se puede observar el esquemático de la tarjeta.

La tarjeta consta principalmente de un regulador de voltaje a 5 voltios; para la alimentación del convertor, una etapa de salida que permite alimentar el rotor del synchro-resolver con una señal de referencia generada por el convertor (AD2S1205/00) y un transformador de Scott de estado sólido para la conversión de señales de tipo synchro a resolver ya que el convertor solo recibe las de tipo resolver. Además la tarjeta permite el acceso a todos los pines de convertor que aunque en este proyecto no son utilizados en su totalidad, en una próxima etapa del mismo podrían ser útiles.

El convertor se polariza con 5 V provenientes del regulador, también contiene condensadores de desacople para disminuir el ruido en el mismo ubicados físicamente lo más cerca posible al convertor siguiendo las recomendaciones del fabricante.

La etapa de salida consta de 4 amplificadores operacionales. Cada salida de señal de referencia proveniente del convertor pasa por 2 etapas en las cuales se realiza: amplificación, filtrado y acondicionamiento de señal, para la alimentación del rotor del synchro-resolver. Esta etapa permite adecuar el nivel DC de la señal de referencia, controlar su ganancia, filtrarla y proteger al convertor si en algún momento el synchro-resolver llegara a pedir o entregar mucha corriente según el caso. Si alguna falla se presentara, se quemarían los transistores o incluso los operacionales, pero el último dispositivo que se dañaría sería el convertor.

El transformador de Scott se basa en la nota de aplicación 252 de Analog Devices<sup>[19]</sup>. Este permite la conversión de señales de tipo synchro a resolver para que puedan ser asimiladas por los convertidores de resolver a digital como es el caso del AD2S1205.

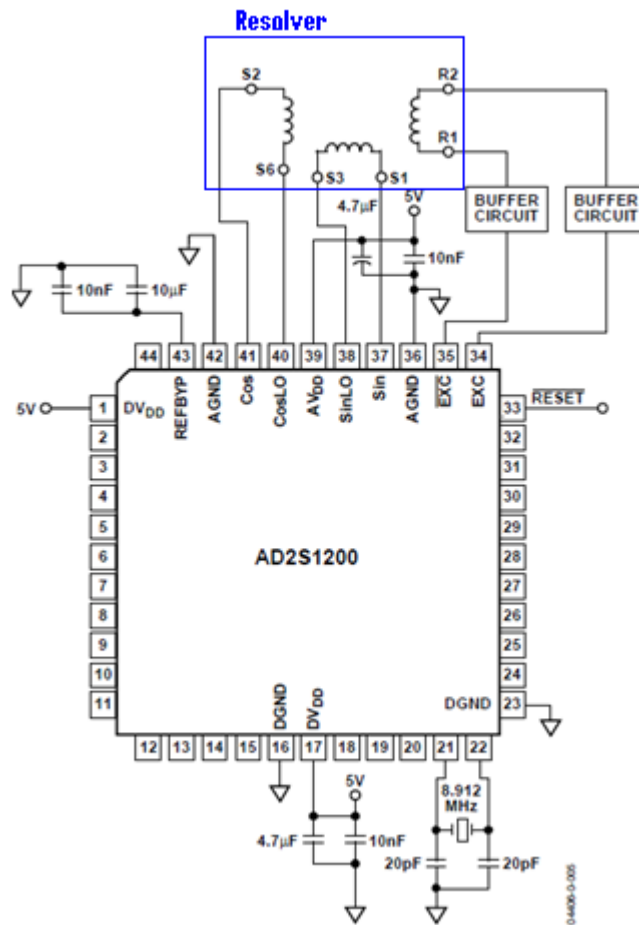


Figura 4.1.1.1. Conexión del conversor A2DS1200/05. Tomada de [http://www.analog.com/static/imported-files/data\\_sheets/AD2S1200.pdf](http://www.analog.com/static/imported-files/data_sheets/AD2S1200.pdf) (23/03/11)

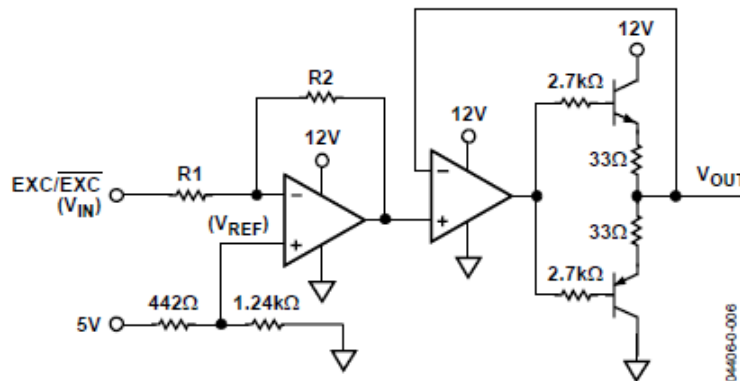


Figura 4.1.1.2. Circuito búffer para AD2S1200/05. Tomada de [http://www.analog.com/static/imported-files/data\\_sheets/AD2S1200.pdf](http://www.analog.com/static/imported-files/data_sheets/AD2S1200.pdf) (23/03/11)

$$Gain = -(R2 / R1)$$

$$\text{and } V_{OUT} = \left( V_{REF} \times \left( 1 + \frac{R2}{R1} \right) \right) - \left( \frac{R2}{R1} \times V_{IN} \right)$$

Ecuación 4.1.1.3. Cálculo de componentes para el circuito búfer. Tomada de [http://www.analog.com/static/imported-files/data\\_sheets/AD2S1200.pdf](http://www.analog.com/static/imported-files/data_sheets/AD2S1200.pdf) (23/03/11)

#### 4.1.2 Tarjeta de desarrollo para dsPIC30F2010:

En esta tarjeta se reciben las señales provenientes de los sensores, se realiza su procesamiento, se implementa el control digital y se genera la señal de alimentación del motor a través del pwm y el puente H. Esta tarjeta fue suministrada por el Ingeniero Camilo Otalora, docente de tiempo completo del departamento de Electrónica la PUJ. Fue desarrollada para la asignatura de Control Digital, y diseñada para realizar experimentos de control con motores DC. Es una tarjeta que facilitó mucho la etapa de control para el proyecto. En esta tarjeta se encuentra el chip dsPIC30F2010 de 28 pines para montaje superficial, un puente H que actúa como manejador del motor, un regulador a 5 V para la alimentación de diferentes chips presentes en la tarjeta, incluido el dsPIC, un chip RS-232, y pines de acceso al dsPIC.

Esta tarjeta permite la comunicación con un PC por puerto serial, utilizando el protocolo RS-232. Así mismo permite la programación del dsPIC con un programador ICD2 o ICD3 de Microchip y específicamente para el proyecto; permite la conexión de las señales de cuadratura que llegan del encoder acoplado al motor, y de la señal de emulación de encoder que genera el conversor AD2S1205 para ser procesadas y controlar la posición del motor DC, que es el objetivo principal del presente proyecto

## 4.2 ETAPA DE APLICACIÓN:

Se realizó un acople mecánico entre el motor y el synchro-resolver, utilizando 2 discos plásticos dentados, con relación de 1:1. El motor tiene de fábrica acoplado el eje del rotor al encoder, por lo que no fue necesario realizar montajes para este. El resto de aplicación consta de la conexión entre la tarjeta de desarrollo para AD2S1205 con la tarjeta donde se encuentra el dsPIC30F2010. Para realizar esta conexión, basta con seguir ciertas instrucciones que serán dadas posteriormente. A continuación se puede observar una fotografía del montaje del sistema completo:

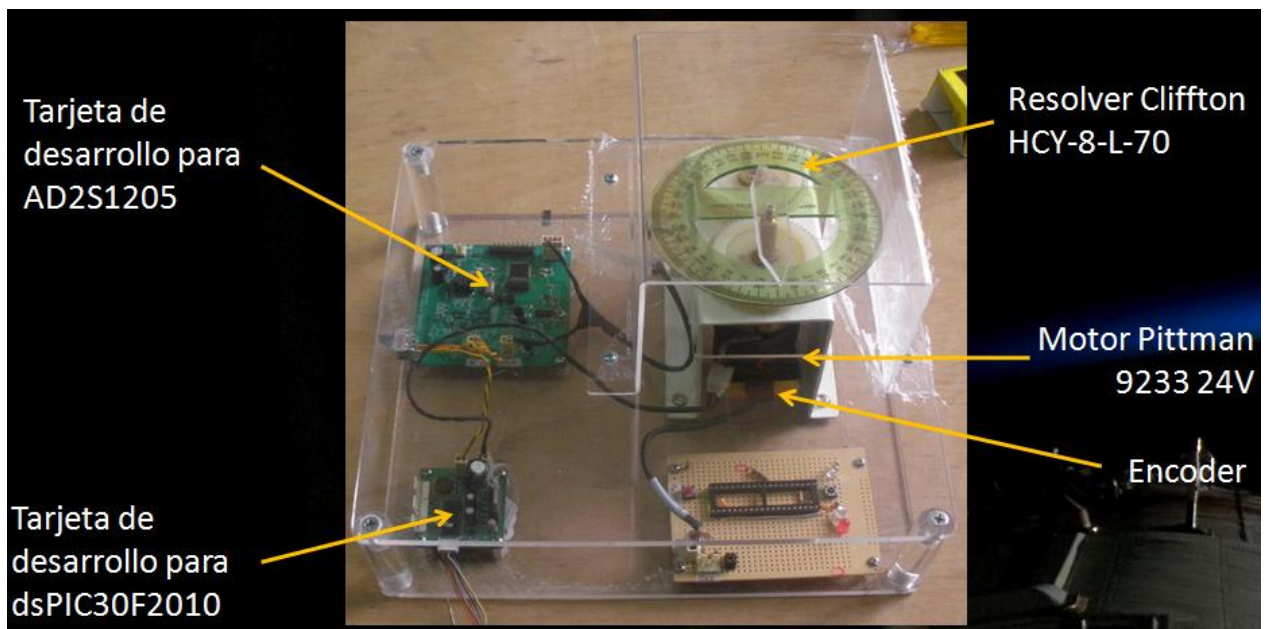


Figura 4.2.1 Montaje del sistema completo.

### 4.3 ETAPA DE PROCESAMIENTO ANALÓGICO:

En la etapa de procesamiento analógico se compone de la generación de la señal seno de referencia que es utilizada para alimentar el rotor del synchro-resolver.

#### 4.3.1 Generación de la señal de referencia para synchro-resolver.

Dentro de las características de los conversores AD2S1200/05 se encuentra la generación de la señal de referencia seno por un oscilador. En ambos conversores, la frecuencia de esta señal es programable polarizando los pines No. 31 (FS1) y No. 32 (FS2) del chip, pero por defecto es de 10 kHz (ya que estos pines son internamente *pull-ups*), que es la frecuencia utilizada para el proyecto. En la figura 4.3.1.1 se puede observar el modulo de oscilador presente en el AD2S1205.



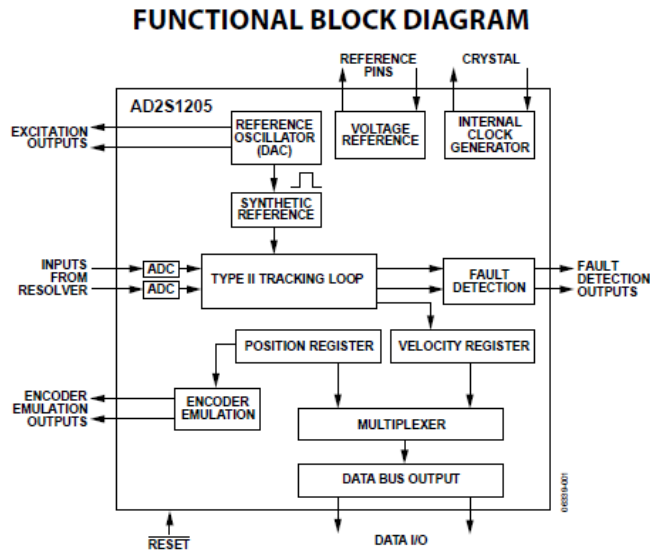


Figura 4.3.1.1. Diagrama de bloques funcional del AD2S1205. Tomada de [http://www.analog.com/static/imported-files/data\\_sheets/AD2S1200.pdf](http://www.analog.com/static/imported-files/data_sheets/AD2S1200.pdf) (23/03/11)

La salida de referencia del AD2S1205 requiere un búfer externo que provee ganancia y corriente adicional para manejar el resolver (ver figura 4.2.1.3). La señal de referencia es tomada del convertor utilizando las salidas No. 34 (EXC) y No. 35 (EXC') las cuales son señales diferenciales que posteriormente son filtradas y tratadas por la etapa de salida de la *Tarjeta de desarrollo para el AD2S1205* explicada anteriormente en el numeral 4.2.1 para. Las características eléctricas de las salidas del oscilador se pueden ver en la figura 4.3.1.2.

EXC, EXC' OUTPUTS					
Voltage	3.34	3.6	3.83	V p-p	Load $\pm 100 \mu\text{A}$ FS1 = high, FS2 = high, CLKIN = 8.192 MHz FS1 = high, FS2 = low, CLKIN = 8.192 MHz FS1 = low, FS2 = high, CLKIN = 8.192 MHz FS1 = low, FS2 = low, CLKIN = 8.192 MHz
Center Voltage	2.39	2.47	2.52	V	
Frequency		10		kHz	
		12		kHz	
		15		kHz	
		20		kHz	
EXC/EXC' DC Mismatch			35	mV	
THD	-58			dB	First five harmonics

Figura 4.3.1.2 Especificaciones de las señales de referencia EXC y (EXC)'. Tomada de [http://www.analog.com/static/imported-files/data\\_sheets/AD2S1200.pdf](http://www.analog.com/static/imported-files/data_sheets/AD2S1200.pdf) (23/03/11)

El cálculo de los componentes y sus valores para la etapa de salida se realizó siguiendo la hoja de especificaciones del convertor AD2S1200, para lo cual fue necesario calcular la ganancia del sincroresolver y así determinar la ganancia de la etapa de salida como se puede ver en la ecuación 4.1.1.3.

El AD2S1205 también contiene internamente una señal de referencia síncrona que esta enganchada en fase con las entradas *Sin* y *Cos* que son las entradas del resolver al convertor. Los errores de fase entre los

devanados primario y secundario del resolver pueden degradar la exactitud de la conversión, esto es compensado mediante esta señal de referencia síncrona. Esto además compensa los cambios de fase debidos a la temperatura y cableado y elimina la necesidad de un circuito de compensación de fase externo.

#### 4.3.2 Emulación de Encoder de Cuadratura.

El conversor tiene la opción de entregar la posición angular del resolver utilizando las salidas *A*, *B* y *NM*, que son señales en cuadratura que emulan un encoder de cuadratura incremental. Esta característica es muy importante para el proyecto, ya que aprovechando el módulo QEI presente en el dsPIC30F2010, se simplifica de forma significativa la programación del control de posición del motor y se trata al synchro-resolver como si fuera otro encoder mas.

El AD2S1205 emula un encoder con resolución de 1024 líneas que se puede aumentar a 4096 utilizando el modulo QEI del dsPIC con la configuración de 4x. Además de las salidas *A* y *B*, se encuentra disponible la señal *NM*; que es semejante a la señal index o zero de un encoder normal. En el numeral 2.1.3 del presente proyecto se explica ampliamente el funcionamiento de los encoders incrementales. En la figura 4.3.2.1 se pueden observar las señales que genera la emulación de encoder del conversor en un diagrama de tiempo, asumiendo una rotación en la dirección de las manecillas del reloj.

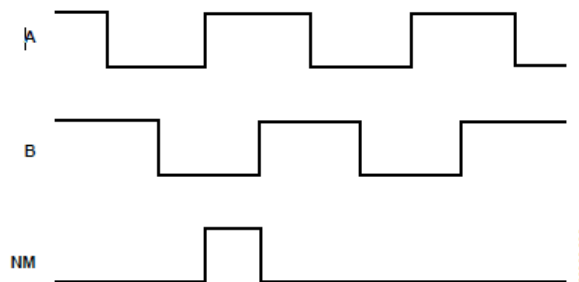


Figura 4.3.2.1. Diagrama de tiempo de Emulación de encoder incremental AD2S1205. Tomada de [http://www.analog.com/static/imported-files/data\\_sheets/AD2S1200.pdf](http://www.analog.com/static/imported-files/data_sheets/AD2S1200.pdf) (23/03/11)

Esta señal de emulación de encoder proveniente del conversor, a diferencia de los encoders reales, no esta sujeta a errores como: errores de ciclo, excentricidad, errores debidos al ancho de pulso, densidad de conteo y fase. El máximo rango de velocidad (*n*) para un encoder se calcula de su máxima frecuencia de conmutación (*f* max) y sus pulsos por revolución (PPR) según la siguiente ecuación:

$$n = \frac{60 \times f_{MAX}}{PPR}$$

Ecuación 4.3.2.2. Máximo rango de velocidad para un encoder.

Los pulsos A y B del conversor son inicializados por la frecuencia del reloj interno, la cual es justamente la mitad de la frecuencia del reloj externo. Con una frecuencia de reloj externo de 8.192 MHz, la frecuencia del reloj interno es de 4.096 MHz. La frecuencia equivalente de conmutación del encoder es:

$$\frac{1}{4} \times 4.096 \text{ MHz} = 1024$$

#### 4.4 ETAPA DE PROCESAMIENTO DIGITAL:

Se compone por las funciones del conversor AD2S1205, las cuales son, generar la señal de alimentación para el synchro – resolver, recibir sus señales de salida y convertirlas a formato digital; las funciones del microcontrolador dsPIC30F2010, generar la señal del PWM para controlar el motor y recibir y transmitir el valor de referencia y las constantes del controlador.

##### 4.4.1 CONVERSIÓN RESOLVER A DIGITAL USANDO EL MÉTODO DE TRACKING<sup>[14]</sup>:

Las señales del estator del synchro - resolver son convertidas a formato digital por el conversor AD2S1205 usando el principio del lazo cerrado de seguimiento tipo II. El lazo de seguimiento implementado digitalmente rastrea continuamente la posición del resolver sin necesidad de conversiones externas y estados de espera. A medida que el resolver se mueve a una posición equivalente al bit menos significativo, la salida del lazo de seguimiento es actualizada por 1 LSB.

El conversor rastrea el ángulo del eje ( $\theta$ ), produciendo un ángulo de salida ( $\varphi$ ) que es retroalimentado y comparado con el ángulo de entrada ( $\theta$ ); la diferencia entre los 2 ángulos es el error, el cual se reduce a cero cuando el conversor sigue correctamente al ángulo de entrada. Para medir el error, S3 - S1 es multiplicada por  $\text{Cos}(\varphi)$  y S2 - S4 es multiplicada por  $\text{Sin}(\varphi)$  para obtener:

$$E_o \text{Sin}(wt) \times \text{Sin}\theta \text{Cos } \varphi \quad \text{para S3 - S1} \quad \text{Eq. 4.4.1.1}$$

$$E_o \text{Sin}(wt) \times \text{Sin}\theta \text{Sin } \varphi \quad \text{para S2 - S4} \quad \text{Eq. 4.4.1.2}$$

Se obtiene la diferencia, dando como resultado,

$$E_o \text{Sin}(wt) \times (\text{Sin}\theta \text{Cos } \varphi - \text{Cos}\theta \text{Sin } \varphi) \quad \text{Eq. 4.4.1.3}$$

Esta señal es demodulada usando la referencia sintética, internamente generada, obteniendo,

$$E_o (\text{Sin}\theta \text{Cos } \varphi - \text{Cos}\theta \text{Sin } \varphi) \quad \text{Eq. 4.4.1.4}$$

La ecuación Eq. 4.5.1.4 es equivalente a  $E_o \text{Sin}(\theta - \varphi)$ , la cual es aproximadamente igual a  $E_o(\theta - \varphi)$  para valores pequeños de  $\theta - \varphi$ , donde  $\theta - \varphi$  es el error angular.

El valor  $E_o \text{Sin}(\theta - \varphi)$  es la diferencia entre el error angular del rotor y el ángulo digital de salida del conversor.

Un demodulador sensor de fase, integradores y un filtro de compensación conforman el sistema de lazo cerrado encargado de reducir a cero la señal de error. Cuando esto es logrado,  $\varphi$  equivale al ángulo del resolver,  $\theta$ , dentro del rango de exactitud del conversor.

#### 4.5 ETAPA DE CONTROL:

En esta sección se halló la función de transferencia del motor a partir de su hoja de especificaciones, se desarrolló el controlador análogo, se realizó su conversión a modo discreto y se explicó el desarrollo del código del control en el microcontrolador.

##### 4.5.1 MODELAMIENTO DEL SISTEMA

Desarrollando las ecuaciones del numeral 2.5, la función de transferencia que se obtiene es la siguiente:

$$\frac{W(s)}{Vin(s)} = \frac{K_T}{JLS^2 + S(RJ + B_t L) + (RB_t + K_V K_t)} \quad \text{Eq. 4.5.1.1}$$

Para el motor usado en este proyecto (Pittman 9233 24V) los valores de estas constantes, de acuerdo a su hoja de especificaciones, son los siguientes:

$J = 3.25e-6 \text{ kg m}^2$   
 $B_t = 2.28e-6 \text{ Nm s/rad}$   
 $K_V = 0.0373 \text{ V/rad/s}$   
 $K_t = 0.0373 \text{ Nm/A}$   
 $R = 3.94 \text{ } \Omega$   
 $L = 3.29 \text{ mH}$

La función de transferencia es la siguiente:

$$\frac{W(s)}{Vin(s)} = \frac{0.0373}{1,069e^{-8}s^2 + s1.281e^{-5} + 0.001397} \quad \text{Eq. 4.5.1.2}$$

Sin embargo para obtener un modelo más aproximado, se utilizaron las constantes encontradas de forma experimental, ver capítulo 5, se observó la respuesta en lazo cerrado unitaria real y se modificó la inercia con el fin de incluir los engranajes y el resolver dentro del sistema,

Así, las constantes para el motor son:

$J = 1,793e-5 \text{ kg}\cdot\text{m}^2$   
 $B_t = 2e-6 \text{ Nm s/rad}$   
 $K_v = 0.0369 \text{ V/rad/s}$   
 $K_t = 0.0369 \text{ Nm/A}$   
 $R = 3.6 \ \Omega$   
 $L = 3.61 \text{ mH}$

De esta forma se obtiene la siguiente función de transferencia:

$$\frac{W(s)}{V_{in}(s)} = \frac{0.0369}{6,474e^{-8}s^2 + s6.463e^{-5} + 0.001369} \quad \text{Eq. 4.5.1.3}$$

Finalmente se multiplica por  $\frac{1}{s}$  para obtener la salida en posición.

$$\frac{\theta(s)}{V_{in}(s)} = \frac{0.0369}{6,474e^{-8}s^3 + 6.463e^{-5}s^2 + 0.001369s} \quad \text{Eq. 4.5.1.4}$$

#### 4.5.2 IMPLEMENTACIÓN DEL CONTROL:

Para el desarrollo del control, primero se observó una simulación realizada en Matlab de la respuesta de la planta a una entrada paso, se escogió una amplitud de  $360^\circ$  ya que es el máximo valor que el sistema va a recibir, a continuación se observa la simulación:

Posición [°] vs Tiempo [s]

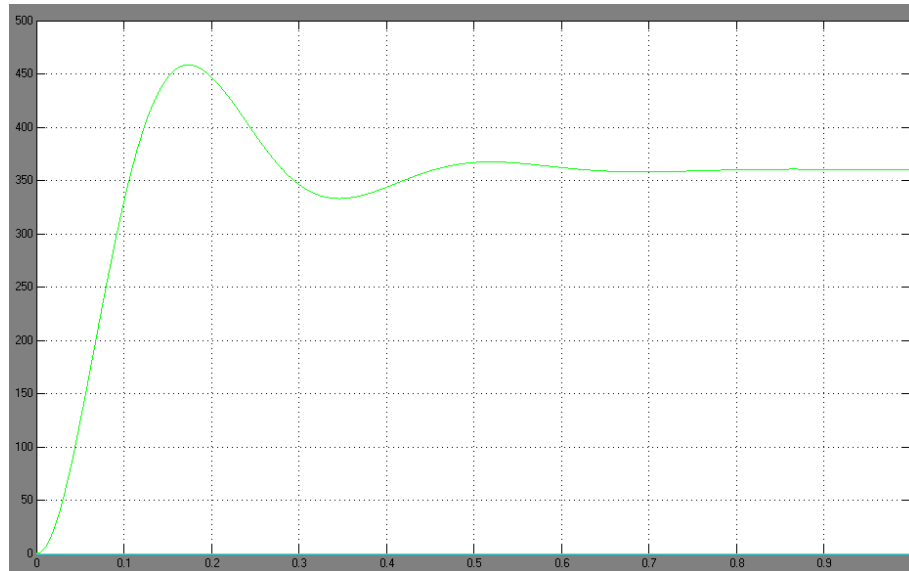


Figura 4.5.2.1 Simulación de la respuesta a entrada paso para la planta.

Se obtuvo una respuesta con sobrepico del 27%, sin error estacionario, con un tiempo de establecimiento de 0.527s y con tiempo de subida de 0.0984s.

#### 4.5.2.1 DESARROLLO DEL CONTROL ANÁLOGO:

Inicialmente el desarrollo del control para ambos sensores es el mismo, debido a que las señales que recibe el control son en formato de encoder de cuadratura a través del módulo QEI. Solamente cambia en cuanto a la resolución en el conteo del número de pulsos recibidos, lo cual se explica más adelante.

Para obtener el control análogo se utilizó la herramienta *sisotool* de *Matlab*, utilizando la gráfica del lugar de las raíces de la planta:

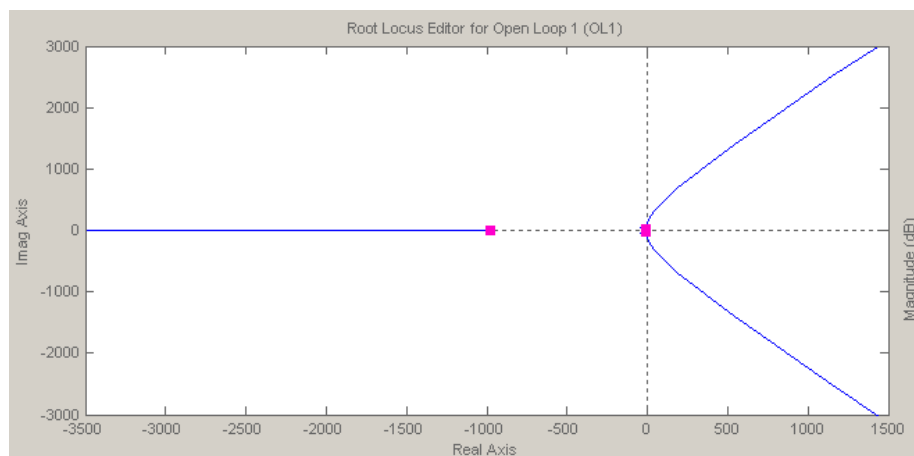
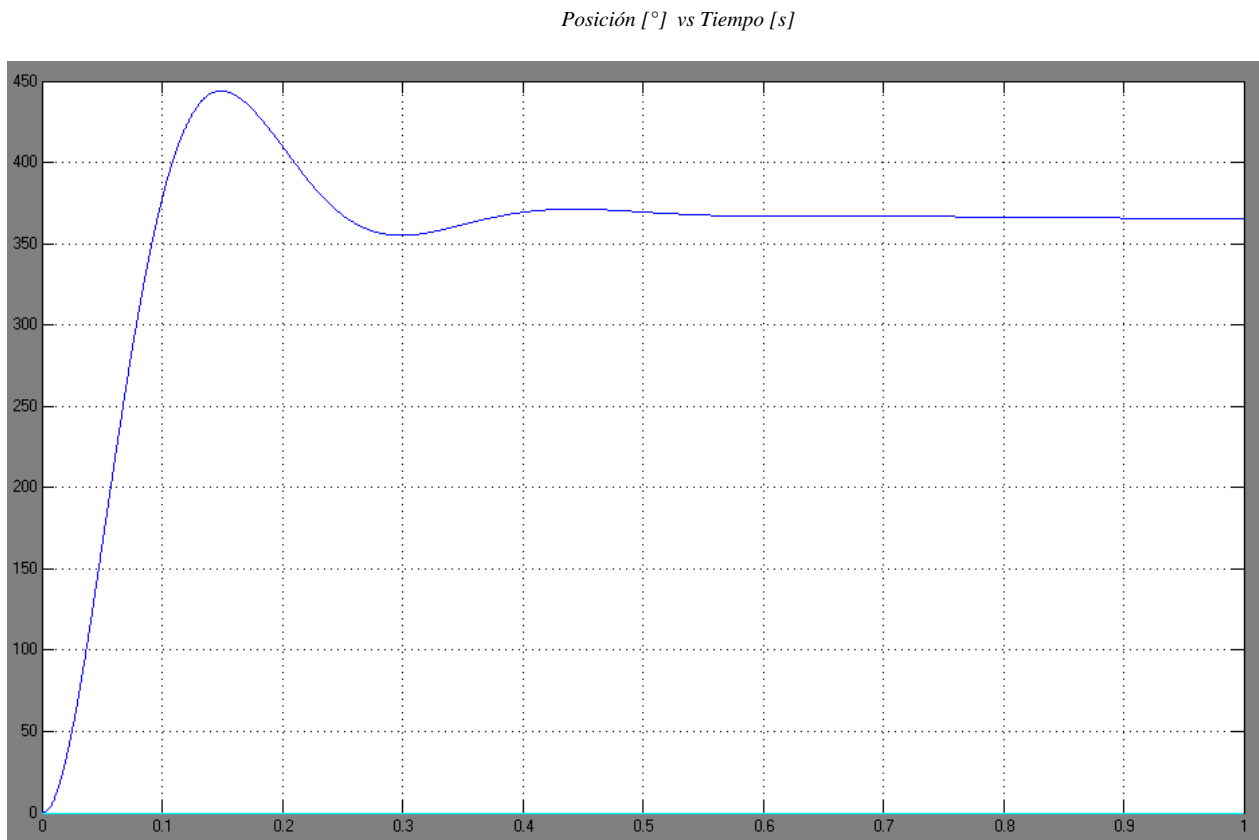


Figura 4.5.2.1.1 Lugar de las raíces de la planta

Utilizando el método de ubicación de polos y ceros, observando simultáneamente la respuesta paso del sistema, se buscó obtener la función de transferencia del control que mejorara la respuesta del sistema, con lo cual se llegó al siguiente resultado:



*Figura 4.5.2.1.2 Respuesta a entrada paso para la malla definitiva del control Análogo.*

Se obtuvo una respuesta con 25% de sobrepico, con un tiempo de subida de 0.0844s y con tiempo de establecimiento de 0.58s, como se puede observar en la figura 4.5.2.1.2

Y el controlador al que se llegó es el siguiente:

$$C = 1 + \frac{0.8}{s} + 0.001s$$

Así, los valores del PID son  $K_p = 1$ ,  $K_i = 0.8$  y  $K_d = 0.001$ .

#### **4.5.2.2 DESARROLLO DEL CONTROL DISCRETO:**

Luego de obtener el controlador análogo, se procedió a realizar su conversión a forma discreta, ya que el control se implementaría en el microcontrolador.

Para este desarrollo se utilizó la función *c2d* de MATLAB, utilizando el método de Tustin. Para saber el periodo de muestreo, primero se observó el bode de la función de transferencia del motor con el fin de obtener su ancho de banda.

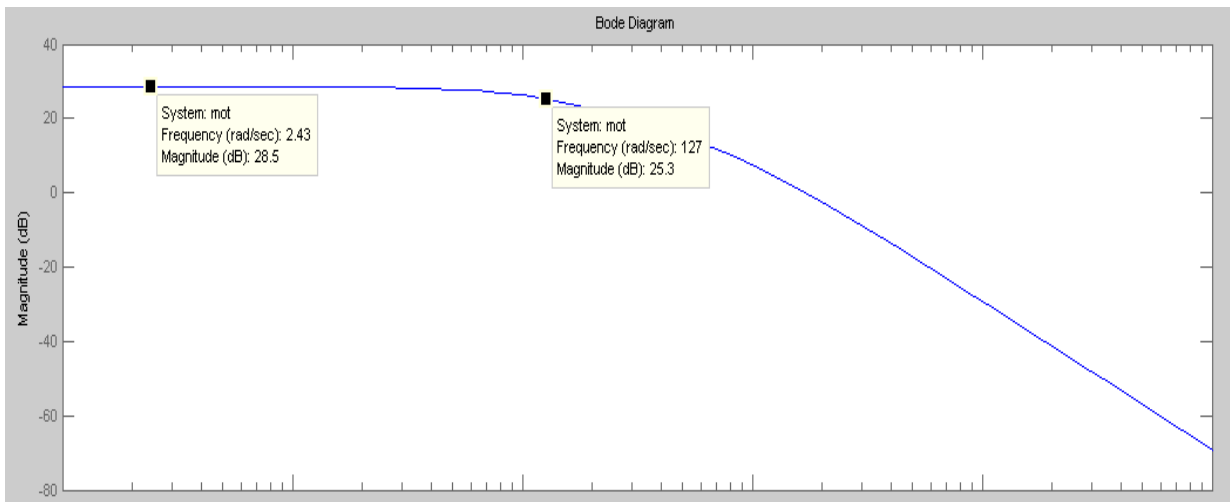


Figura 4.5.2.2.1 Bode de la función de transferencia motor

De esta forma se obtiene un ancho de banda de 20 Hz, el cual por el criterio de Nyquist y para obtener mejores resultados al muestrear la señal, se multiplica 20 veces para obtener una frecuencia de muestreo de 400Hz o un periodo de 2.5 ms.

Usando la función de c2d de MATLAB se hallan las siguientes ecuaciones para las constantes del controlador, donde  $O(z)$  es la salida y  $e(z)$  el error.

**Ecuación Proporcional Discreta:**

$$\frac{O_p(z)}{e(z)} = 1$$

**Ecuación Integral Discreta:**

$$\frac{O_i(z)}{e(z)} = \frac{0.001z + 0.001}{z - 1}$$

**Ecuación Derivativa Discreta:**

$$\frac{O_d(z)}{e(z)} = \frac{0.8z - 0.8}{z + 1}$$

Ahora multiplicando por  $1/z$  ambos de lados de las ecuaciones se tiene:

**Ecuación Proporcional Discreta:**

$$O_p(z) = 1e(z)$$

**Ecuación Integral Discreta:**

$$O_i(z) - O_i(z)z^{-1} = 0.001e(z) + 0.001z^{-1}e(z)$$

**Ecuación Derivativa Discreta:**



$$O_d(z) + O_d(z)z^{-1} = 0.8e(z) - 0.8e(z)z^{-1}$$

Finalmente reemplazando  $z$  por  $k$  se obtienen las ecuaciones para las salidas del controlador discreto:

**Ecuación Proporcional Discreta:**

$$O_p[k] = 1e[k] \tag{Eq. 4.6.2.1}$$

**Ecuación Integral Discreta:**

$$O_i[k] - O_i[k - 1] = 0.001e[k] + 0.001e[k - 1]$$

$$O_i[k] = (e[k] + e[k - 1])0.001 + O_i[k - 1] \tag{Eq. 4.6.2.2}$$

**Ecuación Derivativa Discreta:**

$$O_d[k] + O_d[k - 1] = 0.8e[k] - 0.8e[k - 1]$$

$$O_d[k] = (e[k] - e[k - 1])0.8 - O_d[k - 1] \tag{Eq. 4.6.2.3}$$

Las ecuaciones 4.6.2.1, 4.6.2.2 y 4.6.2.3 son las que se implementaron en el código del control discreto, en las cuales las ganancias son:

$$K_p = 1, K_i = 0.001, K_d = 0.8$$

A continuación se presenta la simulación, realizada en *Simulink* de *Matlab*, del control análogo y del control discreto a una entrada paso:

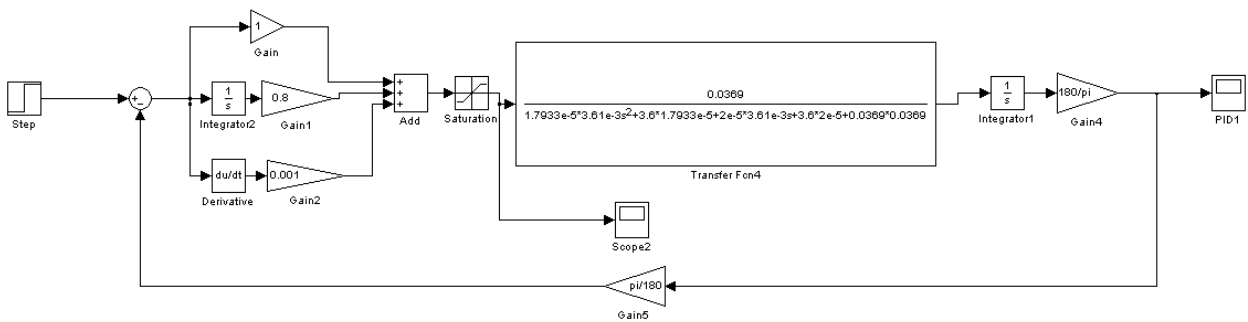


Figura 4.5.2.2.2 Lazo del Controlador PID Análogo

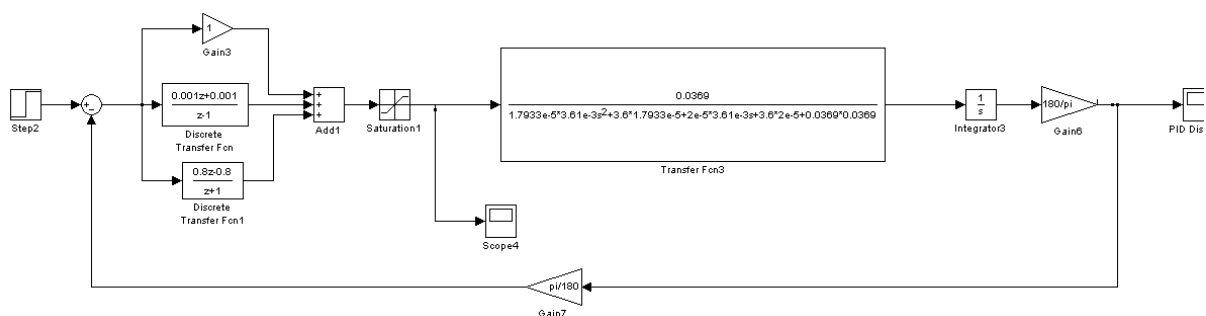


Figura 4.6.2.2.3 Lazo del Controlador PID Discreto

Posición [°] vs Tiempo [s]

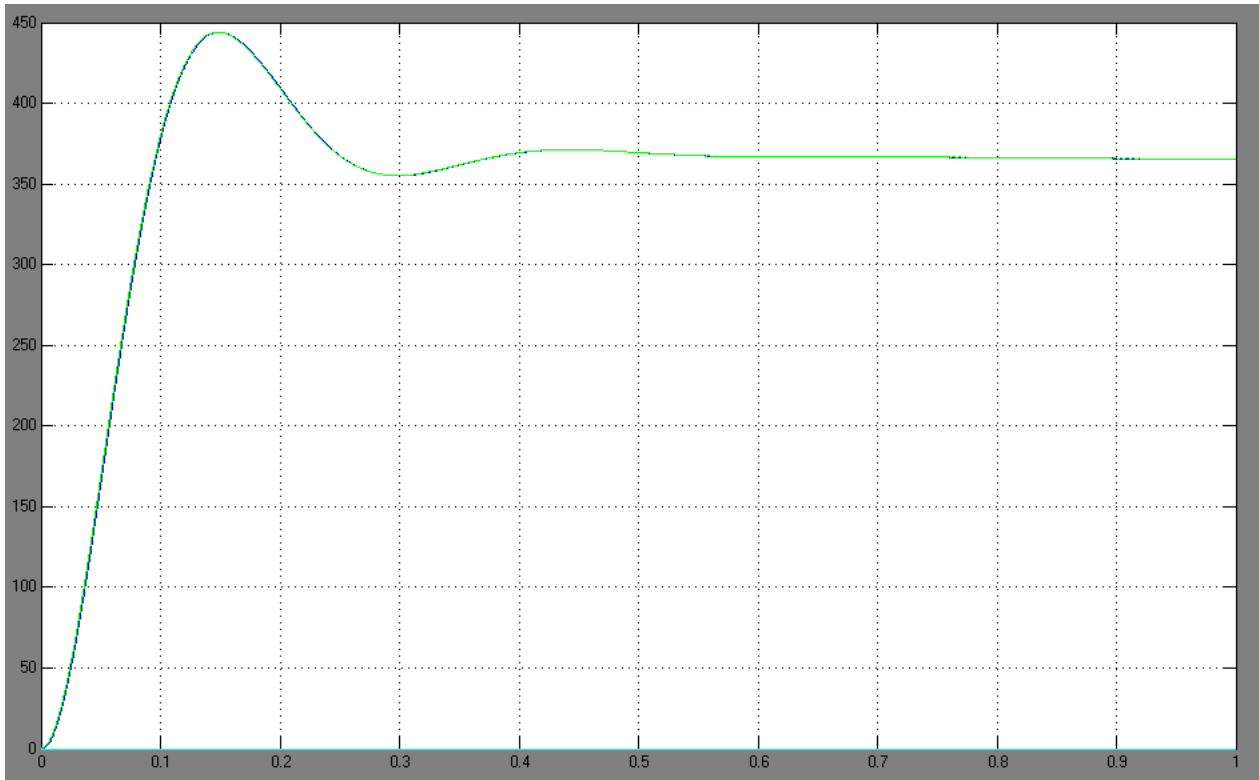


Figura 4.6.2.2.4 Comparación control discreto y control Análogo.

En la figura se observa en color azul la respuesta del control análogo y en color verde la respuesta del control discreto, con una frecuencia de muestreo de 400 Hz.

#### 4.5.2.3 IMPLEMENTACIÓN DEL CÓDIGO DEL CONTROL EN EL MICROCONTROLADOR:

El control se implementa de acuerdo a la figura 4.6.2.1. Para su desarrollo se utilizaron los conteos guardados en la variable POSCNT del módulo QEI, la cual cuenta cada vez que las señales de un encoder de cuadratura generan un pulso de subida.

Dentro del código se realizó un control para recibir y procesar los conteos de la variable POSCNT, los cuales pueden ser del encoder o del conversor AD2S1205.

En el caso del encoder acoplado al motor, se obtienen 500 pulsos multiplicados x2 por el módulo QEI, lo que nos arroja un total de 1000 pulsos, este número se utiliza para hallar la resolución con el encoder, la cual es:

$$Resolución_{Encoder} = \frac{360^\circ}{1000pulsos} = 0,36$$

Y con las señales del emulador de encoder de cuadratura, entregadas por el conversor AD2S1205, se obtienen 1024 pulsos multiplicados x2 por el módulo QEI, obteniendo 2048 pulsos y una resolución de:

$$Resolución_{Emulador\ de\ Encoder} = \frac{360^\circ}{2048\ pulsos} = 0,17578125$$

Luego se asignan los conteos de POSCNT a una variable llamada *count* para que guarde los conteos de 0 a 999 para el encoder y de 0 a 2047 para el emulador de encoder, esta variable es multiplicada por 0,36 en el control utilizando encoder y por 0,17578125 en el control utilizando synchro – resolver. De esta forma se obtiene una variable que varía de 0 a 360, la cual se guarda en otra llamada *Ang* que representa el ángulo de movimiento del rotor del motor. Esta variable se empieza a restar con la referencia ingresada por el usuario, obteniendo el error, y este error es el que se utiliza en las Eq. 4.6.2.1, 4.6.2.2 y 4.6.2.3. Luego el resultado de cada una de estas tres ecuaciones se suma en una sola variable, obteniendo un número que es la salida del controlador digital, la cual es asignada a la salida del PWM para controlar el movimiento del motor por medio del driver L603.

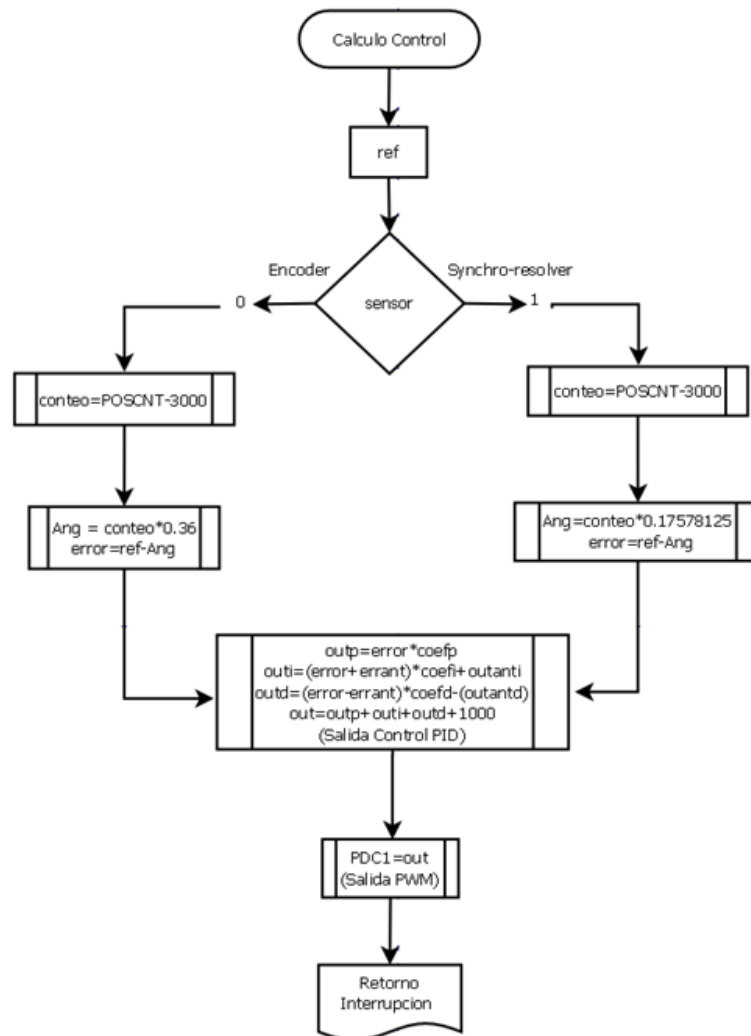


Figura 4.5.2.3.1. Diagrama de flujo del Control.

### 4.5.3 PWM

La señal del PWM es generada por el dsPIC30F2010, el valor de la frecuencia se determinó buscando mantener el valor cuadrático medio del rizado de corriente menor al 10% de la corriente nominal, es decir 0,15 A. También se necesitaba usar un valor adecuado que tolerara el dispositivo, en este caso el driver L603, por lo tanto se escogió un valor de 20 kHz. A continuación se presenta el cálculo:

$$\Delta I_{ppmax} = \frac{U_{dc} * T_s}{2L} = \frac{24 * (\frac{1}{20000})}{2 * 0.00329} = 0.1823 \text{ A}$$

Asumiendo la forma de onda del rizado triangular, se tiene

$$\Delta I_{rmsmax} = \frac{U_{dc} * T_s}{2L} * \frac{1}{\sqrt{3}} = 0.105 \text{ A}$$

### 4.5.4 PUENTE H L6203<sup>[21]</sup>:

El driver utilizado para recibir la señal de PWM del microcontrolador y alimentar el motor, L6203 (Figura 4.6.4.1), es un circuito integrado de puente completo, utilizado específicamente para aplicaciones de control de motores y realizado en la tecnología BCD-Multipower, la cual combina transistores aislados DMOS, CMOS de potencia y circuitos bipolares en el mismo chip. Mediante el uso de tecnología mixta ha sido posible optimizar la lógica del circuito y la etapa de potencia para lograr el mejor rendimiento posible. Los transistores de salida DMOS pueden operar a voltajes de alimentación de hasta 42V y ser eficientes a altas velocidades de conmutación. Todas las entradas lógicas son compatibles tecnologías TTL, CMOS y microcontroladores.

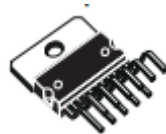


Figura 4.5.4.1. Puente H.L6203

## 4.6 COMUNICACIÓN SERIAL, MICROCONTROLADOR-HYPERTERMINAL:

Para comunicar la interfaz con el usuario realizada en el software de comunicación Hyperterminal, se utilizó la interfaz de comunicación serial RS232. Esto, debido a que la tarjeta de desarrollo del dsPIC30F2010 tenía ya implementado el dispositivo *driver/receiver* de un canal SN65C3221, este es un circuito que proporciona la interfaz eléctrica entre un controlador de comunicación asíncrona y un conector de puerto serial, además, opera a velocidades de hasta 1 Mbit/s. En el microcontrolador se implementa el código para realizar la comunicación serial, haciendo uso del módulo UART, el cual cuenta con un registro de 32 bits para datos de transmisión y un registro de 8 bits para recibir datos, este código se encuentra en el ANEXO B.

#### 4.7 INTERFAZ CON EL USUARIO:

La interfaz con el usuario se realizó con el software de comunicaciones Hyperterminal. En el microcontrolador se configuró el código para transmitir y recibir información a 115200 bits/s. A través de este se digita la referencia, el valor de las constantes Proporcional, Integral y Derivativo del controlador, adicionalmente se selecciona el controlador dependiendo del sensor que se va a conectar al dsPIC30F2010. Los valores se escriben en un registro de 6 bits, en donde el primer bit es una letra que representa la variable que se introducirá, los siguientes 4 bits son para escribir el valor de la variable y en el sexto se pulsa la tecla Enter. En el caso de la referencia, se representa con la letra 'r', en el primer bit siempre se digita el número 0 y en los siguientes 3 bits se escribe el valor, el cual está entre 0° y 360°. En el caso de las constantes del controlador, para la constante Proporcional, se digita la letra 'p', para la Integral la letra 'i' y para la Derivativa la letra 'd'. En los siguientes 4 bits se escribe el valor, el cual se puede escribir entre 0,000 y 9,999. Finalmente para seleccionar el controlador de acuerdo al sensor, se digita la letra 's', los siguientes tres bits siempre se escribe el número 0, en el quinto bit se escribe 0 para el encoder ó 1 para el synchro-resolver, por defecto se encuentra en 0.

Simultáneamente se pueden observar en pantalla las constantes digitadas, todas se encuentran en 0 por defecto. También se muestra en pantalla el valor de la variable *conteo*, el cual muestra los conteos de la variable POSCNT del modulo QEI, en caso de necesitar obtener el valor del ángulo de salida incluyendo sus decimales, para esto se multiplica este valor por 0,36 si se está usando como sensor encoder y por 0,17578125 si se está usando Synchro – Resolver.

Hay que tener en cuenta que todos los valores se muestran en formato HEXADECIMAL, a continuación se muestran el orden en aparecen estas variables en Hyperterminal.

Angulo	kp	ki	kd	conteo	sensor
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0
0000	0000	0000	0000	0000	0

Figura 4.7 Orden en que aparecen las variables en Hyperterminal

En el ANEXO C se encuentran ejemplos de cómo introducir los valores.

## 5 ANÁLISIS DE RESULTADOS

Para comprobar el correcto funcionamiento de los desarrollos teóricos implementados, a continuación se presentan las pruebas realizadas al motor y a los lazos de control con los correspondientes resultados obtenidos:

### 5.1 VALIDACIÓN DEL MODELO DEL SISTEMA:

#### 5.1.1 CONSTANTES DEL MOTOR:

La primera prueba realizada fue la validación de las constantes del motor Pittman 9233 24V, obtenidas a partir de su hoja de especificaciones. Para la realización de esta prueba, se alimentó el motor con una fuente de voltaje múltiple (marca KI) con la cual se variaba el voltaje  $V_{in}$  con incrementos de 1 V, con un DVM (Fluke 45) se medía la corriente media de entrada al motor para cada valor de voltaje, y finalmente utilizando el encoder del motor, a través del osciloscopio (Tektronix TDS-2022) se obtenía un valor de frecuencia para cada intervalo. En siguiente tabla muestran los resultados de la prueba realizada para comprobar el valor de la constante del generador  $K_v$  y la fricción viscosa  $B_t$ :

$V_{in}$ [V]	Vinducido [V]	Corriente [mA]	Frecuencia [kHz]	Revoluciones [Rpm]	Velocidad [rad/s]	Torque [Nm]
4	3,73996	66	5,5	660	69,1150383	0,0026664
5	4,7242	70	7,5	900	94,2477795	0,002828
6	5,71632	72	9,5	1140	119,3805207	0,0029088
7	6,70647	74,5	11,5	1380	144,5132619	0,0030098
8	7,69465	77,5	13,5	1620	169,6460031	0,003131
9	8,68874	79	15,7	1884	197,2920184	0,0031916
10	9,6848	80	17,8	2136	223,6813967	0,003232
11	10,68086	81	20	2400	251,327412	0,0032724
12	11,67692	82	22	2640	276,4601532	0,0033128
13	12,66116	86	24	2880	301,5928944	0,0034744
14	13,65328	88	26,1	3132	327,9822727	0,0035552
15	14,64934	89	28,2	3384	354,3716509	0,0035956
16	15,6454	90	30,2	3624	379,5043921	0,003636
17	16,64146	91	32,4	3888	407,1504074	0,0036764
18	17,63949	91,5	34,6	4152	434,7964228	0,0036966
19	18,63949	91,5	36,8	4416	462,4424381	0,0036966
20	19,64343	90,5	39,6	4752	497,6282758	0,0036562
21	20,65525	87,5	41,7	5004	524,017654	0,003535
22	21,65131	88,5	44,1	5292	554,1769435	0,0035754
23	22,64934	89	46,2	5544	580,5663217	0,0035956
24	23,64146	91	48,5	5820	609,4689741	0,0036764

*Tabla 5.1.1.1 Prueba motor PITMAN 9233 24V.*

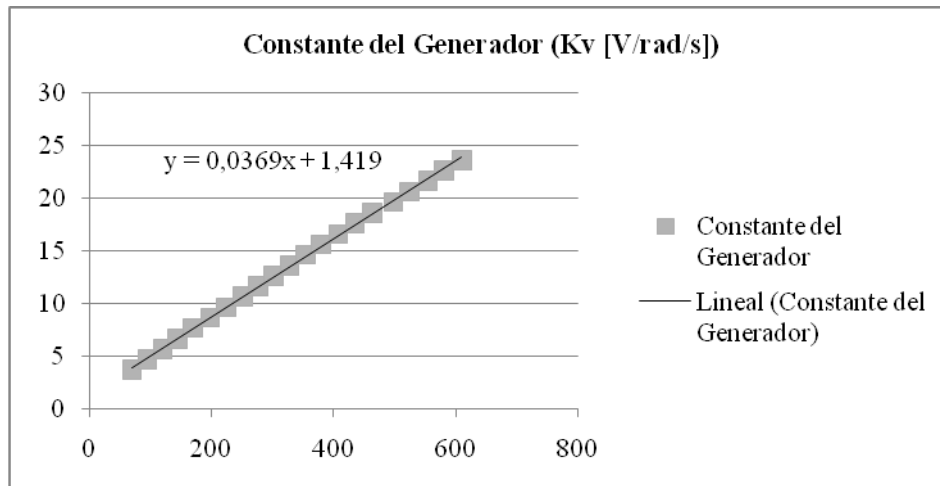


Figura 5.1.1.1 Constante del generador del motor

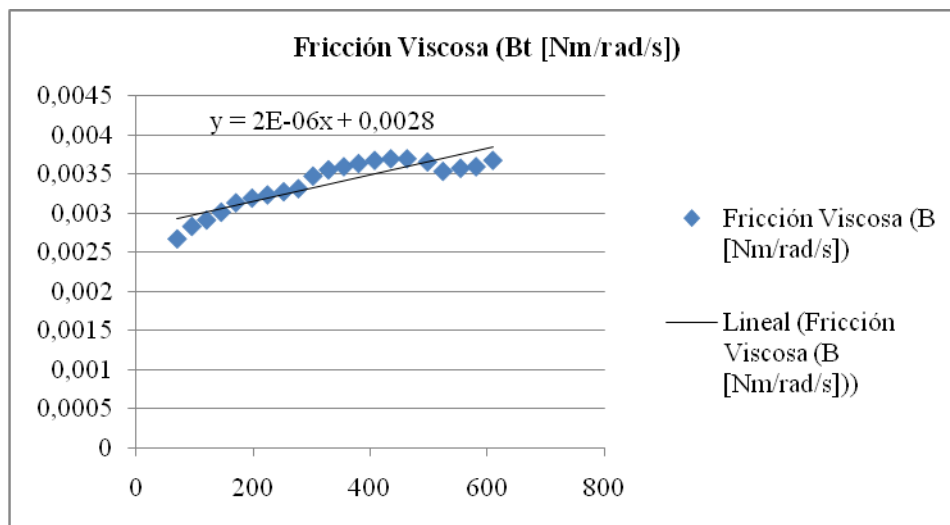


Figura 5.1.1.2 Fricción viscosa del motor

Graficando los valores obtenidos en la tabla 5.1.1, se obtiene un  $K_v = K_t = 0,0369$  [V/rad/s] y una fricción viscosa  $B_t = 2e-6$  [Nm/rad/s]).

Para medir la resistencia e inductancia del motor se utilizó el *Programmable Automatic RCL Meter PM630*, obteniendo así  $R = 3.6 \Omega$  y  $L = 3.61$  mH.

De esta forma se obtuvieron, experimentalmente con la prueba descrita al inicio del numeral 5.1.1, las siguientes constantes del motor:

- $J = 1.7933e-6$  kg m<sup>2</sup>
- $B_t = 2e-6$  Nm s/rad
- $K_v = 0.0369$  V/rad/s
- $K_t = 0.0369$  Nm/A
- $R = 3.6 \Omega$

L = 3.61 mH

### 5.1.2. PRUEBA EN MALLA ABIERTA DE LA PLANTA:

Para validar el modelo de la planta se alimentó el motor con un pulso generado manualmente de 5 V de amplitud y 1s de duración, se tomaron los datos de su posición por hyperterminal, usando como sensor el encoder. Posteriormente se simuló el mismo pulso, para comparar las respuestas de la planta en simulación y en la práctica, a continuación se muestran los resultados:

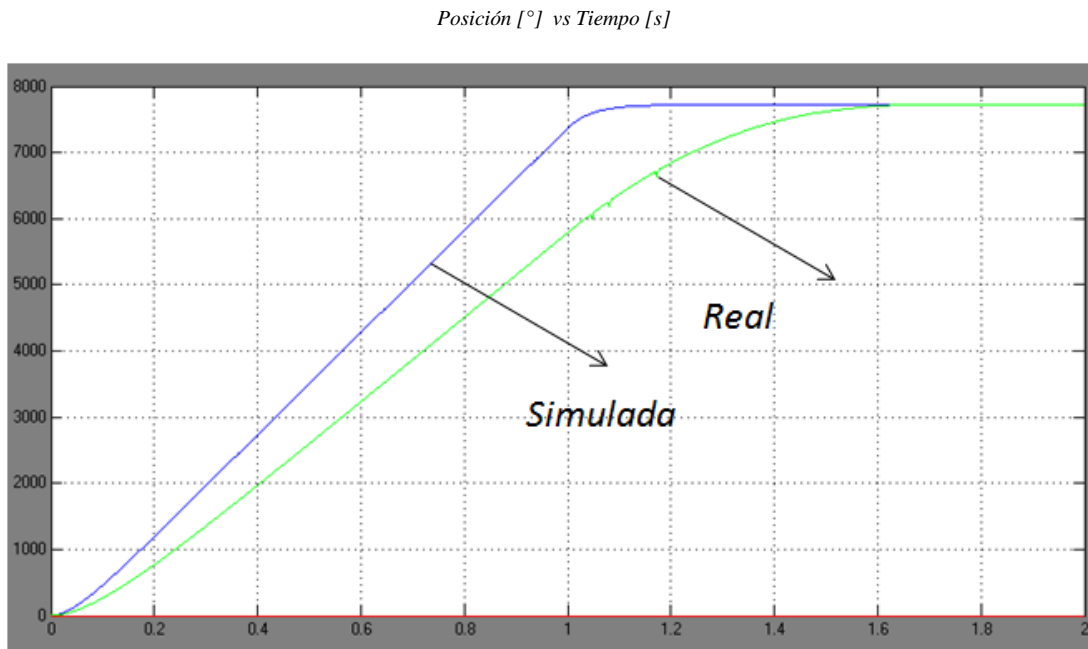


Figura 5.2.1.1 Respuesta de la planta en malla abierta.

En la gráfica se observa en color azul la respuesta simulada y en color verde la respuesta real. La respuesta simulada tiene un tiempo de subida de 0.95 s, un tiempo de establecimiento de 1.05 s y un valor final de 7722°, mientras que la respuesta real tiene un tiempo de subida de 1.23 s, un tiempo de establecimiento de 1.5 s y un valor final de 7715°. El error del estado estacionario se halla con la siguiente fórmula:

$$\text{error} = \frac{\text{Valor Final Simulado} - \text{Valor Final Real}}{\text{Valor Final Simulado}} * 100\%$$
$$\text{error} = \frac{7722^{\circ} - 7715^{\circ}}{7722^{\circ}} * 100\%$$
$$\text{error} = 0,09\%$$

Se pueden notar diferencias considerables entre la rapidez de la respuesta del sistema real y el simulado. Esto probablemente debido a que el valor tomado de la inercia del sistema fue una aproximación mas no un valor exacto, donde se desprecian engranajes, cargas y fricciones de elementos externos al motor.

En cuanto al error en estado estable es un valor pequeño que es tolerable y por esto se valida la función de transferencia hallada para el sistema.



## 5.2 PRUEBA DE GANANCIA RESOLVER

En esta prueba se halló la ganancia del synchro resolver cliffton. Es una prueba esencial, ya que con ella se ajustó la ganancia del circuito búfer de la tarjeta de desarrollo para AD2S1200 como se explica en el numeral 4.1.1. Esto con el fin de cumplir con las especificaciones de voltaje de amplitud de entrada que acepta el conversor para el resolver. Esta se pueden observar en la hoja de datos del mismo.

Vin Rotor	Vout	Ganancia (A)
3,92 Vpp	3,76 Vpp (fase S1-S3)	0,96
3,92 Vpp	3,72 Vpp (fase S2-S4)	0,94

*Tabla 5.2.1. Prueba de Ganancia de Resolver.*

A partir de esta medida se halló una ganancia del resolver promedio de 0,95. Y con este valor se ajustó la ganancia del búfer. En el siguiente numeral se pueden observar las señales de salida del búfer y del resolver con sus respectivos valores de amplitud.

### 5.3 PRUEBAS DE SEÑALES:

Para estas pruebas se utilizó el osciloscopio Tektronix ref: TDS 2022B y el generador de funciones Leader LFG-1310.

#### 5.3.1 SEÑAL DE ALIMENTACIÓN DE ROTOR DE SYNCHRO - RESOLVER.

Esta señal, como se explicó anteriormente, es generada por la tarjeta de desarrollo para AD2S1200. Es una señal sinusoidal diferencial que alimenta el rotor del synchro - resolver. A continuación se muestran gráficas tomadas con el osciloscopio de cada señal referida a tierra que genera la tarjeta a partir del conversor (EXC y EXC'). Esta señal se conecta de modo diferencial al rotor.

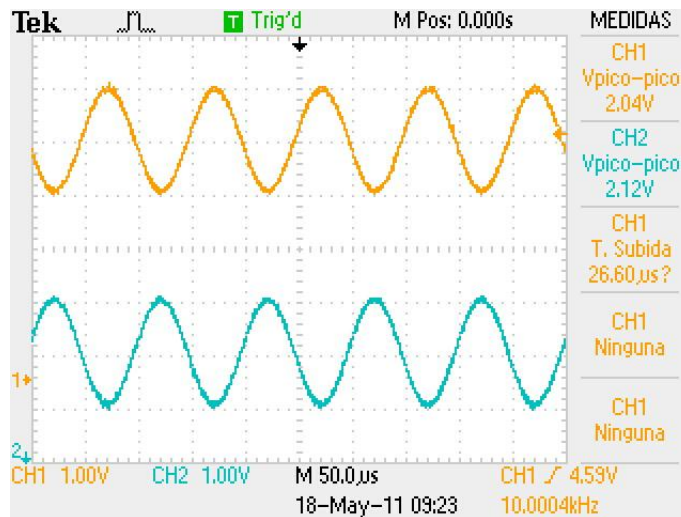


Figura 5.3.1.1. Salidas de alimentación Synchro Resolver de Tarjeta de desarrollo para AD2S1200/05 (Referidas a tierra).

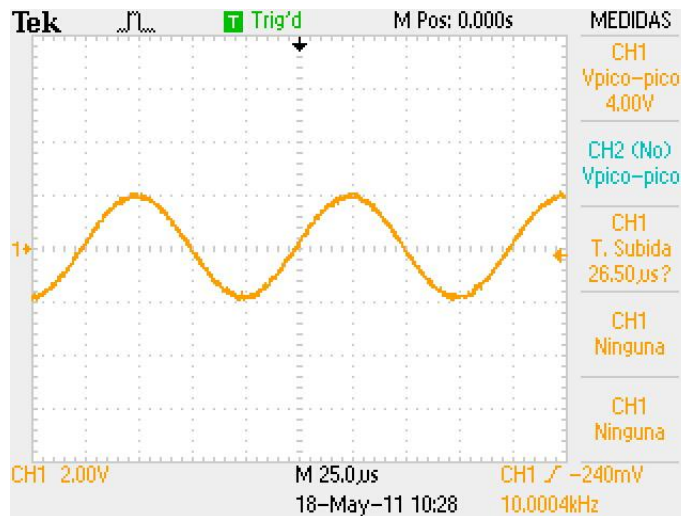


Figura 5.3.1.2. Salida de alimentación Synchro Resolver de Tarjeta de desarrollo para AD2S1200/05 (Diferencial).

### 5.3.2 DISTORSIÓN SEÑAL SENO DE ALIMENTACIÓN AL RESOLVER (THD)

Se realizó esta medición con el analizador de distorsión. Se obtuvo una distorsión baja, ya que la señal presentó una distorsión (THD) de 0,8%. Ver foto. Esta medición se realizó de modo diferencial, que es como se alimenta el synchro resolver. Es importante garantizar un valor de distorsión bajo, que cumpla los requerimientos de distorsión máxima permisible a la entrada de las señales de Resolver en el convertor AD2S1205 para que la medida digital de ángulo sea lo más exacta posible.



Figura 5.3.2. Medición de THD para la señal de alimentación del resolver.

### 5.3.3 SEÑALES DE SALIDA DE SYNCHRO RESOLVER

El Synchro Resolver es alimentado con la señal sinusoidal diferencial de 10 kHz generada por el AD2S1005. Como se explicó en la teoría, éste, genera 2 señales seno y coseno de salida, desfasadas 90° las cuales varían su amplitud según la posición del eje del Synchro Resolver. La tabla siguiente indica las convenciones para la medición de dichas señales:

COLOR	CANAL	FASE MEDIDA
Amarillo	1	S1 - S3
Azul	2	S2 - S4

Tabla 5.3.3.1 Convenciones para la medición de señales de salida Synchro Resolver.

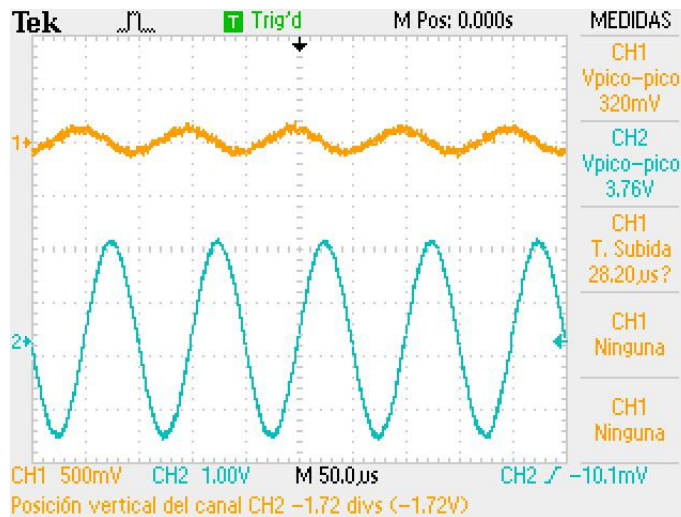


Figura 5.3.3.2. Señales de salida Synchro Resolver. Máxima amplitud S2 – S4.

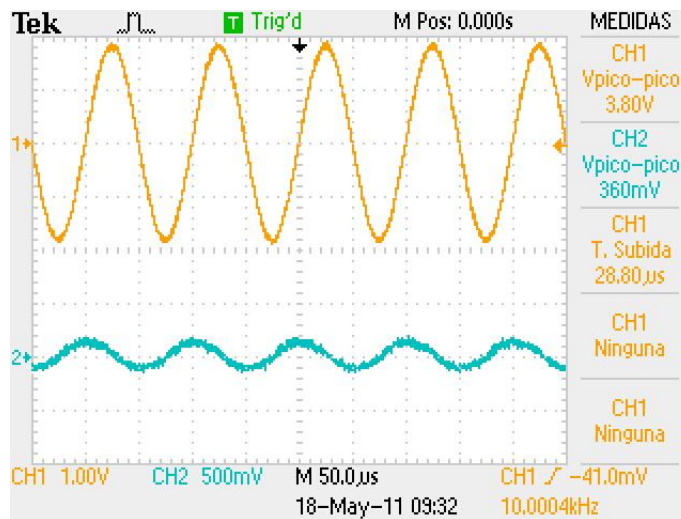
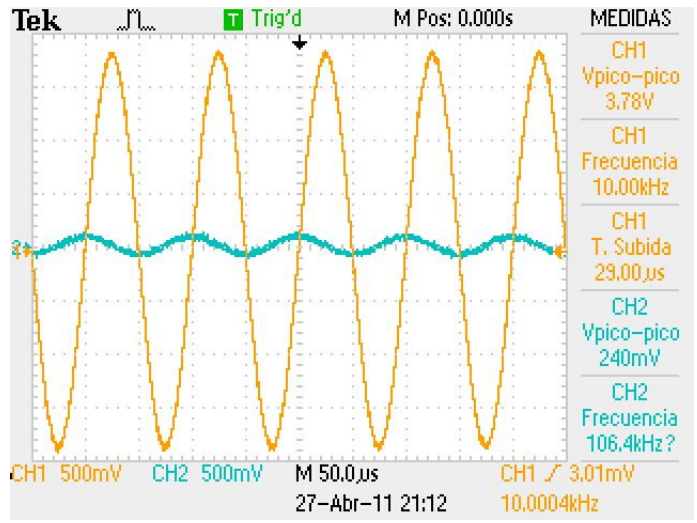
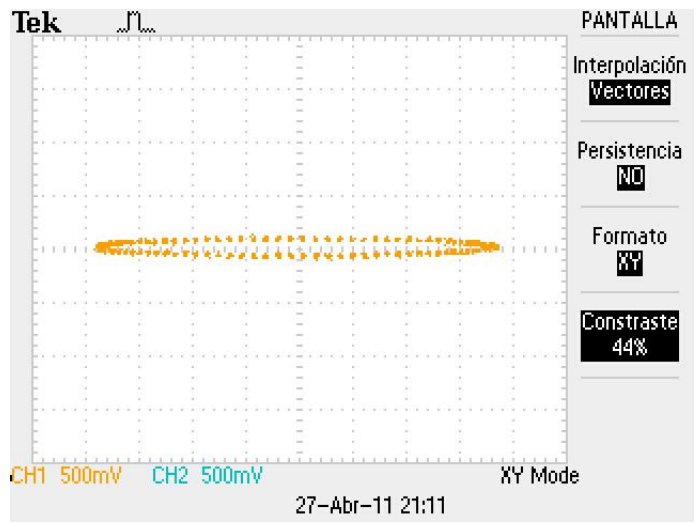


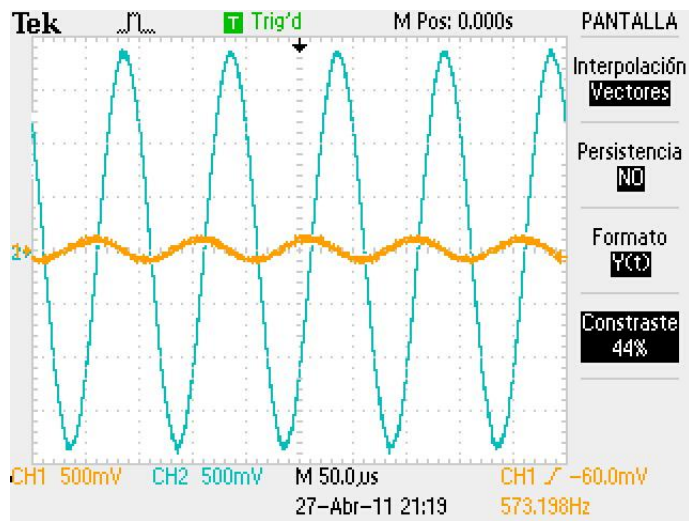
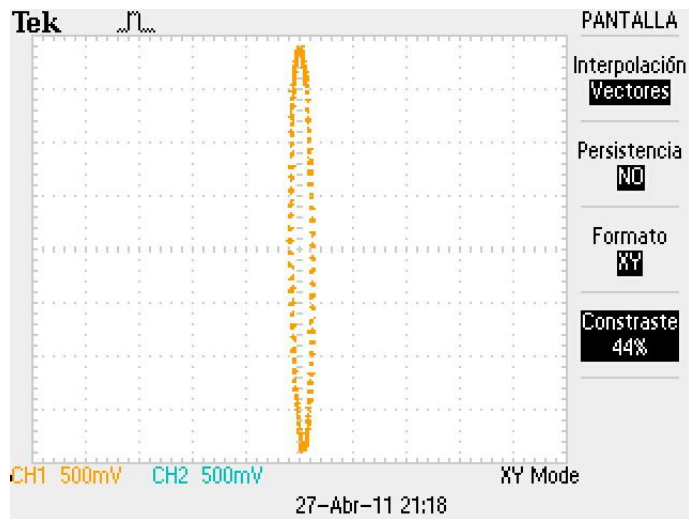
Figura 5.3.3.3. Señales de salida Synchro Resolver. Máxima amplitud S1 – S3

Como se puede observar en las figuras 5.2.2.1 y 5.2.2.2 cuando una fase tiene máxima amplitud, la otra no está en amplitud cero. Las señales son seno y coseno, cuando una tiene máxima amplitud, la otra debería de estar en cero. Esto se refleja en una distorsión de fase como se puede observar en las siguientes pruebas en donde se compara el diagrama de lissajous y la salida en tiempo para diferentes ángulos de posición del eje del synchro resolver. Esto se obtiene utilizando el osciloscopio en modo XY.

- a)  $0^\circ$

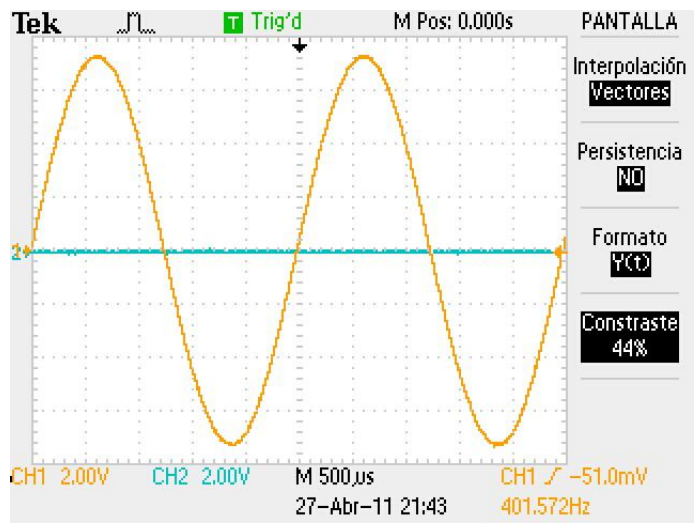
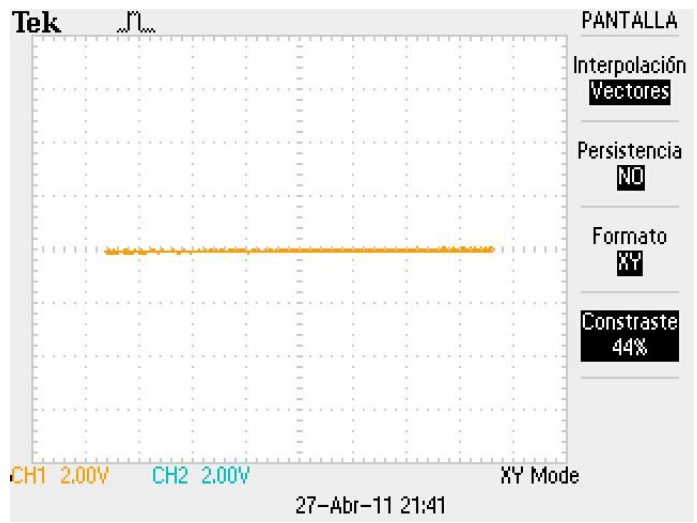


b)  $90^\circ$

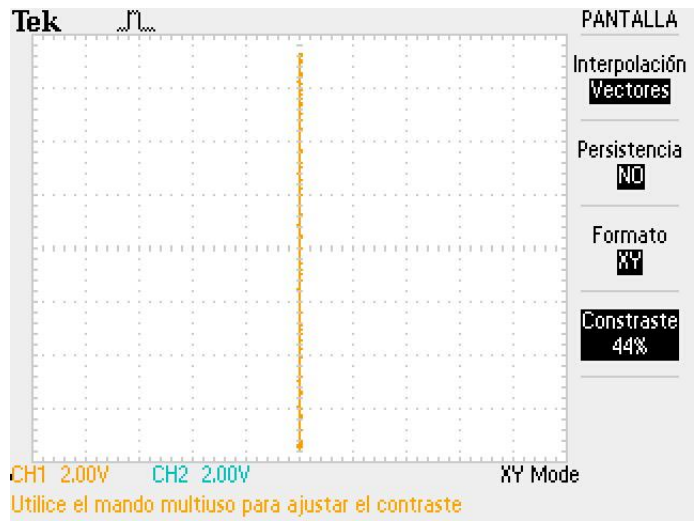


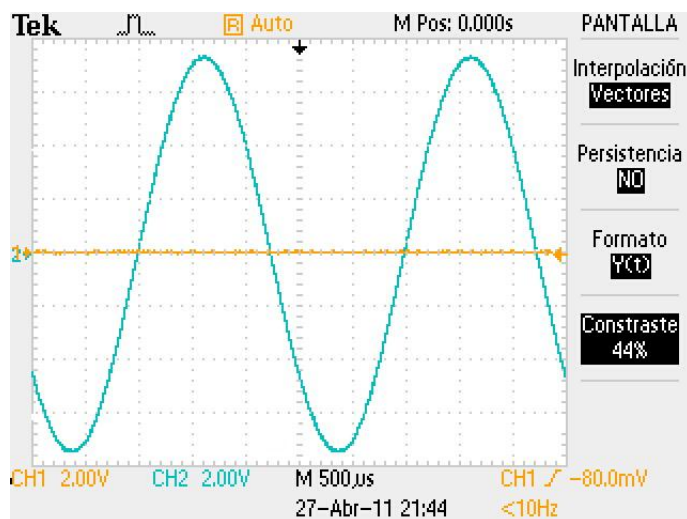
A continuación se muestran otra prueba en la cual se alimentará el Synchronizer - Resolver utilizando el generador de funciones con una señal seno pero ahora con una frecuencia de 400 Hz para los mismos ángulos ( $0^\circ$  y  $90^\circ$ ) donde una fase tiene amplitud máxima y la otra nula.

a)  $0^\circ$



b) 90°





A una frecuencia de 400 Hz de alimentación de rotor, se puede ver que cuando una fase tiene amplitud máxima, la otra es prácticamente nula como es de esperarse. El convertor no alcanza a generar una señal de referencia de tan baja frecuencia, y tal como está diseñado el proyecto, tampoco se puede alimentar el resolver con el generador de funciones, ya que el convertor para calcular la posición angular del sensor realiza internamente una realimentación de la señal de referencia con la que alimenta al sensor para corregir automáticamente errores de fase. Afortunadamente el convertor es capaz de tolerar esta nulidad de las señales, y de realizar la conversión para obtener salidas válidas, tanto en posición angular, como en emulador de encoder de cuadratura. Esto se comprueba viendo los pines de error, que se encuentran en estado ok, A continuación se puede ver en las especificaciones del convertor cuando los pines No. 29 (DOS) y No. 30 (LOT) indican error.

Parameter	Min	Typ	Max	Unit	Conditions/Comments
<b>FAULT DETECTION BLOCK</b>					
Loss of Signal (LOS)					
Sin/Cos Threshold	2.18	2.24	2.3	V p-p	DOS and LOT go low when Sin or Cos fall below threshold DOS indicated before angular output error exceeds limit (4.0 V p-p input signal and 2.18 V LOS threshold)
Angular Accuracy (Worst Case)			57	Degrees	
Angular Latency (Worst Case)			114	Degrees	
Time Latency			125	µs	Maximum electrical rotation before LOS is indicated (4.0 V p-p input signal and 2.18 V LOS threshold)
<b>Degradation of Signal (DOS)</b>					
Sin/Cos Threshold	4.0	4.09	4.2	V p-p	DOS goes low when Sin or Cos exceeds threshold DOS indicated before angular output error exceeds limit Maximum electrical rotation before DOS is indicated
Angular Accuracy (Worst Case)			33	Degrees	
Angular Latency (Worst Case)			66	Degrees	
Time Latency			125	µs	DOS latched low when Sin/Cos amplitude mismatch exceeds threshold
Sin/Cos Mismatch		385	420	mV	
<b>Loss of Tracking (LOT)</b>					
Tracking Threshold		5		Degrees	LOT goes low when internal error signal exceeds threshold; guaranteed by design
Time Latency			1.1	ms	
Hysteresis	4			Degrees	Guaranteed by design

Figura 5.3.3.4. Especificaciones de los pines de error<sup>[14]</sup>.



## 5.4 PRUEBAS DEL CONTROLADOR

Para la realización de estas pruebas se utilizaron las constantes del controlador discreto,  $k_p = 1$ ,  $k_i = 0.001$  y  $k_d = 0.8$ .

### 5.4.1 RESPUESTA DEL SISTEMA A ENTRADA RAMPA:

Se programó el microcontrolador para generar una entrada tipo rampa al motor y se probó la respuesta del sistema con ambos sensores (synchro-resolver y encoder).

#### 5.4.1.1 RESPUESTA RAMPA CON ENCODER

Con encoder como sensor, el control presentó el siguiente comportamiento para entrada rampa:

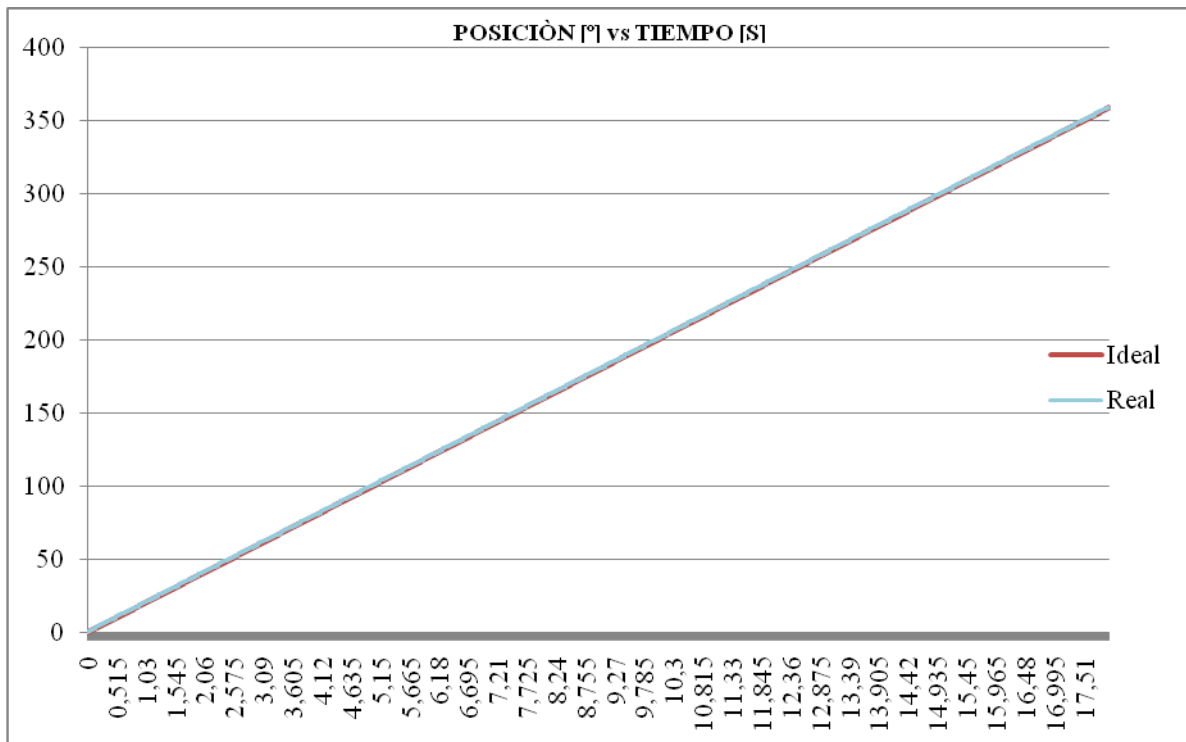


Figura 5.4.1.1.1. Respuesta a entrada rampa con Encoder.

Como se observa en la gráfica, se necesita de bastante resolución en la imagen para alcanzar a ver las diferencias que no se notan en esta vista, lo cual se observa en la gráfica del error.

**Grafica del error:**

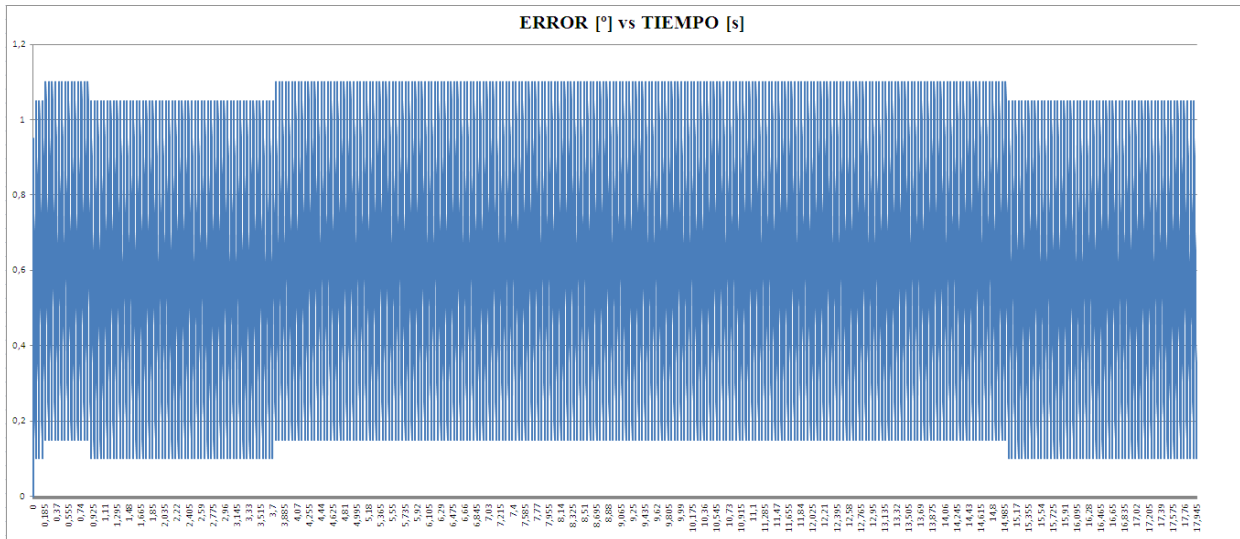


Figura 5.4.1.1.2 Grafica del error de la respuesta rampa con encoder.

En esta grafica se aprecia con claridad el error, el cual varía entre  $0.1^\circ$  y  $1.1^\circ$ , es decir un diferencial de error aproximado a  $1^\circ$ . Teniendo en cuenta que el rango máximo de medida es  $360^\circ$ , el error de  $1^\circ$ , equivale al  $0,2777\%$ . Esto comparado con la máxima resolución teórica que brinda el encoder la cual es de  $0,36^\circ$ , es decir,  $0,1\%$ , se obtiene una resolución similar en ambos casos y el error real es muy cercano al mínimo error posible con máxima resolución.

**5.4.1.2 RESPUESTA RAMPA CON SYNCHRO-RESOLVER**

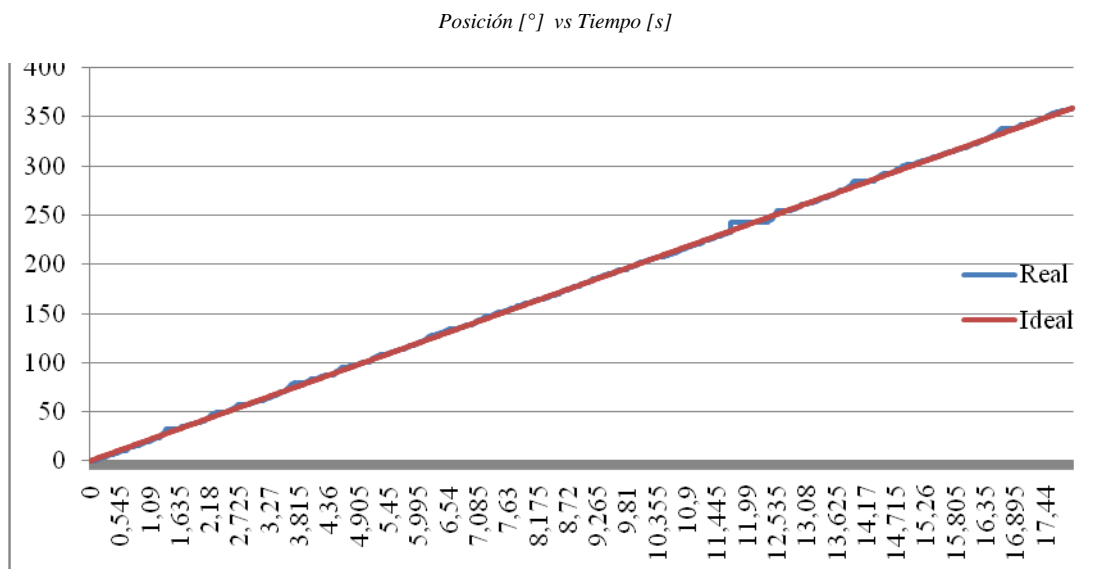


Figura 5.4.1.2.1 Respuesta a entrada rampa con Synchro-Resolver.

En la respuesta a entrada rampa usando Synchro – Resolver es más notable el error, esto debido al mecanismo por el cual esta acoplado al motor, es decir, a los discos dentados.

### Grafica del error:

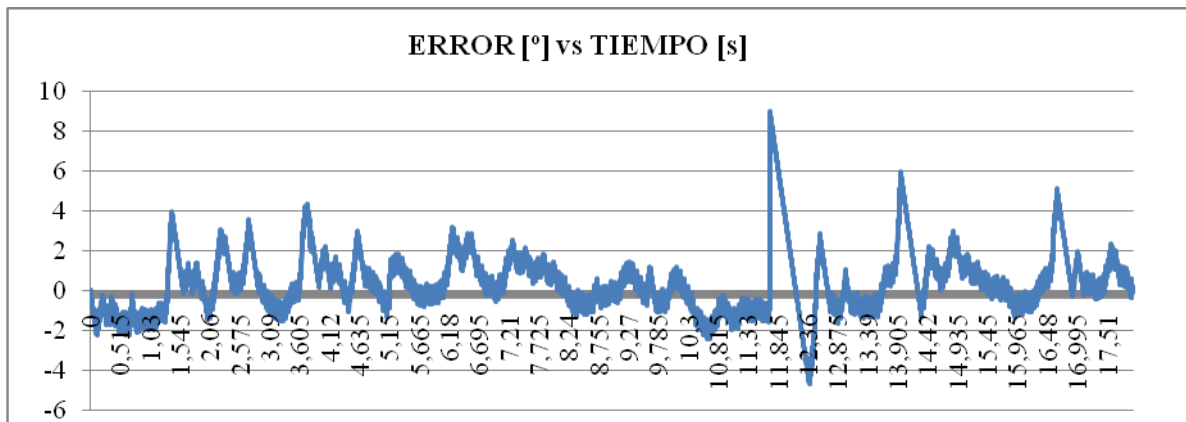


Figura 5.4.1.2.2 Grafica del error a una respuesta rampa con Synchro - Resolver

Para la respuesta a entrada rampa con Synchro - Resolver se alcanzan a notar errores considerables en ciertos tiempos. Esto probablemente debido a que el Synchro - Resolver no está acoplado directamente al eje del motor como lo está el Encoder. Este se conecta al motor por medio de piñones que aumentan el error, además es un sensor sumamente sensible y tiende a variar más su salida que el Encoder.

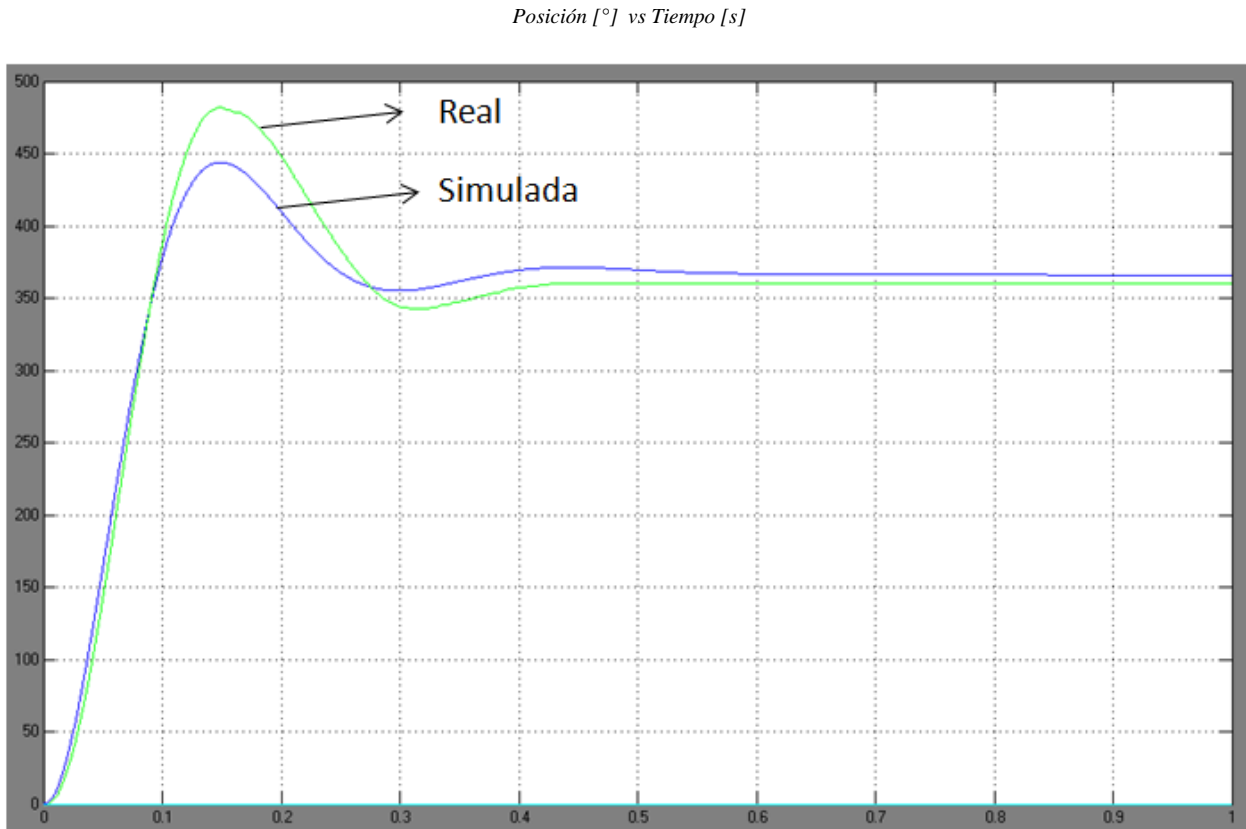
En esta gráfica el error varía entre  $9^\circ$  y  $-5^\circ$ , es decir, un diferencial de  $14^\circ$ , equivalente a 3,8%. Comparado con la resolución máxima teórica de  $0,17^\circ$ , equivalente a 0,05%, es un error importante.

Hay que notar que la respuesta del Encoder, a pesar de ser un sensor con menor resolución teórica que el Resolver, respondió mejor en esta prueba. Esto se debe probablemente al montaje mecánico del sistema; para que el motor se acople con el Resolver hay piñones en juego que proporcionan carga extra, fricción y aumentan el error debido a la dentadura de los mismos, contrario al Encoder que se encuentra directamente acoplado al eje del motor.

## 5.4.2 RESPUESTA DEL SISTEMA A ENTRADA PASO

Para esta prueba se utilizó una entrada paso de  $360^\circ$ .

### 5.4.2.1 USANDO ENCODER COMO SENSOR DE POSICIÓN:



*Figura 5.4.2.1 Respuesta del sistema a una entrada paso de  $360^\circ$  con sensor encoder.*

En la gráfica se observa en color azul la simulación del control y en color verde la respuesta real.

La respuesta real se obtuvo con un sobrepico del 34%, tiempo de subida 0.085s, tiempo de establecimiento de 0.376 s y sin error en estado estacionario.

En comparación con la simulación en donde la respuesta tiene 25% de sobrepico, con un tiempo de subida de 0.0844s y con tiempo de establecimiento de 0.58s, las dos respuestas son de características similares.

### 5.4.2.2 USANDO SYNCHRO – RESOLVER COMO SENSOR DE POSICIÓN:

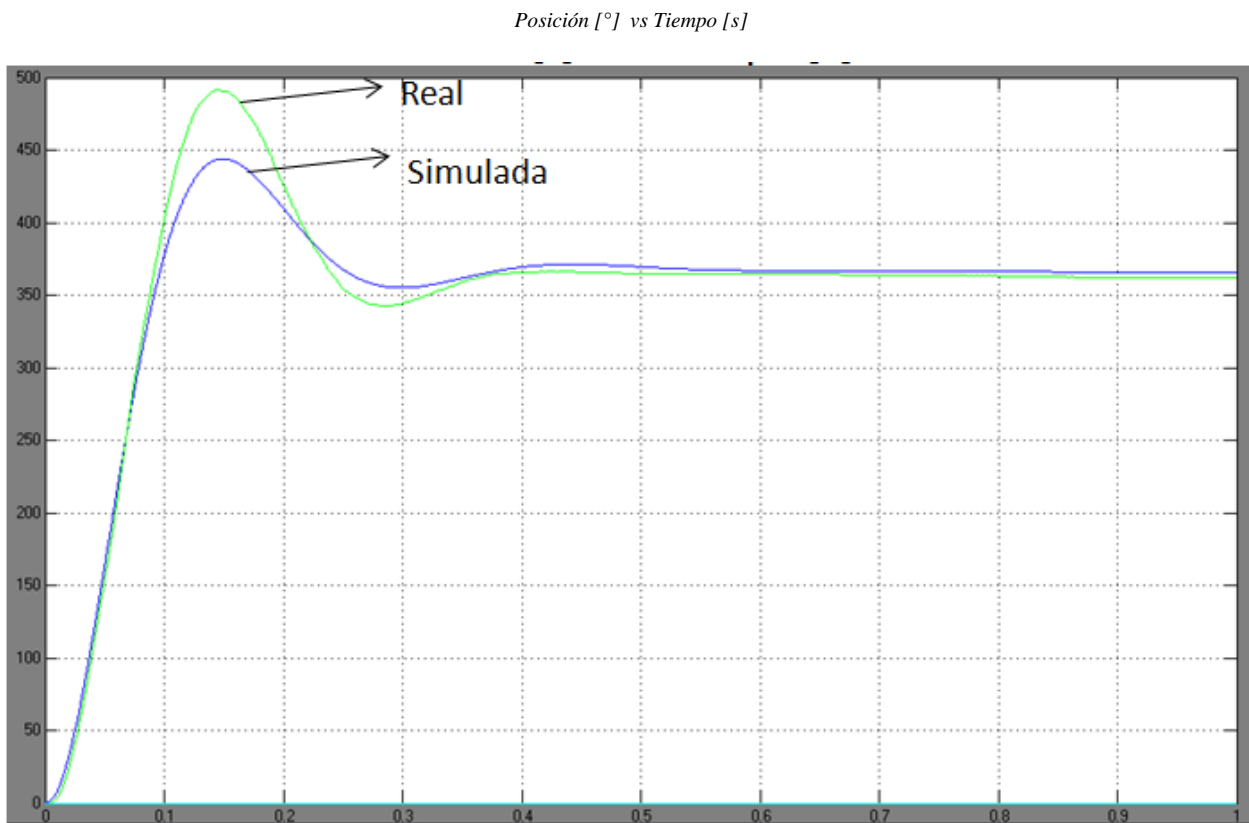


Figura 5.4.2.2 Respuesta del sistema una entrada paso de 360° con sensor synchro - resolver.

En la gráfica se observa en color azul la respuesta del control simulado y en color verde la respuesta del control real.

Se obtuvo en la respuesta real un tiempo de subida de 0.081s, sobrepico de 36.5% y un tiempo de establecimiento de 0.33s.

En comparación con la simulación en donde la respuesta tiene 25% de sobrepico, con un tiempo de subida de 0.0844s y con tiempo de establecimiento de 0.58s, también se puede decir que las dos respuestas son de características similares.

### 5.4.2.3 RESPUESTA A ENTRADA PASO (UNA PRUEBA MÁS):

Se realizaron diferentes pruebas en las cuales se buscó mejorar los parámetros de desempeño del sistema, sin embargo, siempre se obtuvieron respuestas con mayores sobrepicos, más lentas, el sistema se inestabilizaba o presentaba grandes oscilaciones, a continuación se presenta una de tantas pruebas realizadas, en la que se obtuvo una respuesta similar con ambos sensores.

Para esta prueba se utilizaron las siguientes constantes para el controlador discreto:  $k_p = 0.9$ ,  $k_i = 0.001$  y  $k_d = 0.6$ :

Posición [°] vs Tiempo [s]

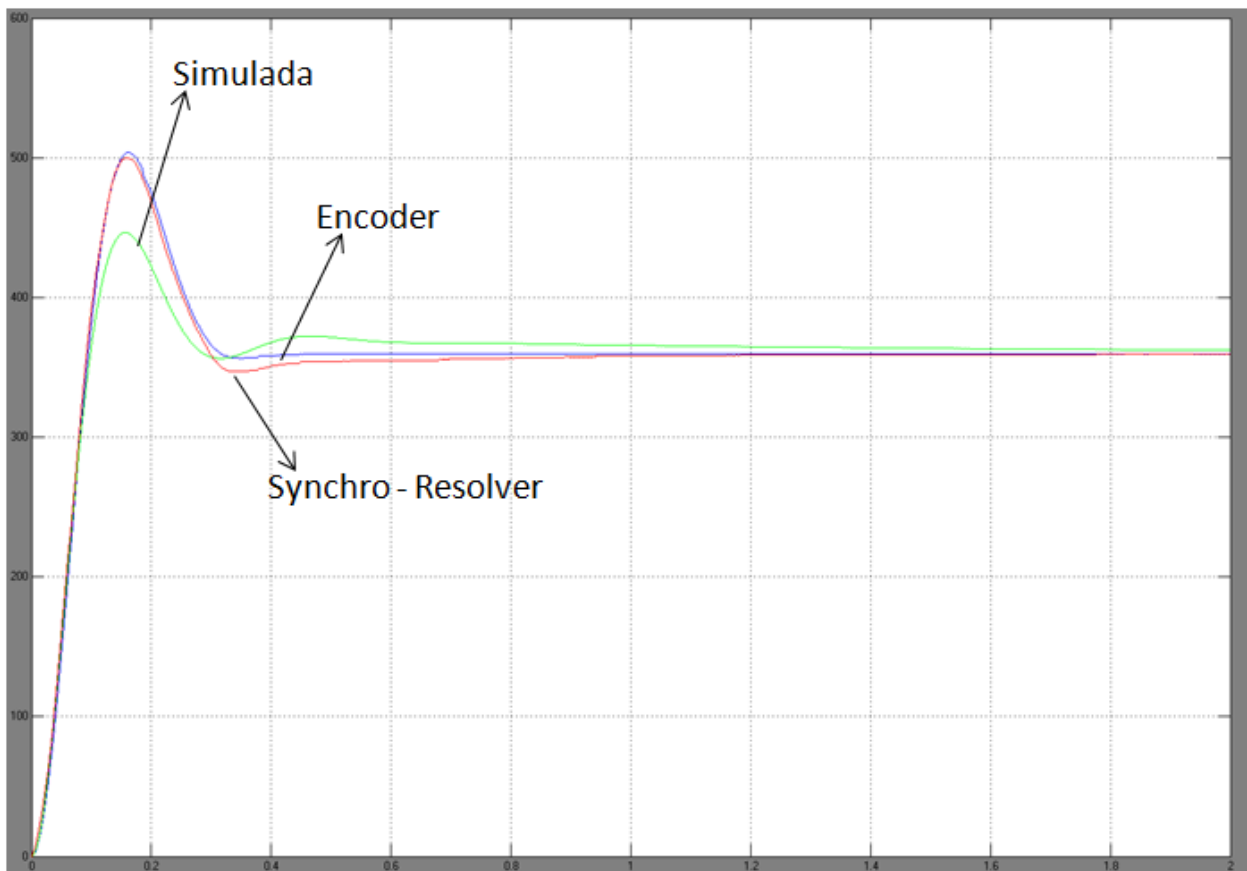


Figura 5.4.2.3.1 Respuesta a entrada paso simulada y real para ambos sensores (Una prueba más).

En color verde se observa la respuesta simulada, en color azul la respuesta usando encoder y en color rojo la respuesta usando Synchro – Resolver.

Para la respuesta usando encoder se obtuvo sobrepico del 40%, tiempo de subida de 0,086 s y un tiempo de establecimiento de 0,297 s. Usando Synchro – Resolver se obtuvo sobrepico de 38.8%, tiempo de subida de 0.084 s y tiempo de establecimiento 0,427s.

**Resumen de Pruebas Controlador:**

SENSOR	PRUEBA	$k_p$	$k_i$	$k_d$	$M_p$	$t_r$	$t_s$
ENCODER	prueba1	1,0	0,001	0,8	34,0%	0,085	0,376
	prueba2	0,9	0,001	0,6	40,0%	0,086	0,297
RESOLVER	prueba1	1,0	0,001	0,8	36,5%	0,081	0,33
	prueba2	0,9	0,001	0,6	38,8%	0,084	0,427

Tabla 5.4.2.3.2. Resumen pruebas controlador.

## 6. CONCLUSIONES

Como se observo en la prueba de respuesta del sistema a entrada rampa, el Synchro - Resolver presenta una mayor sensibilidad para menores variaciones en la referencia, en comparación con el encoder, esto debido a su mayor resolución. Por lo tanto la respuesta a entrada rampa fue más lineal utilizando como sensor encoder que utilizando Synchro -Resolver, ya que este último al tener más resolución hace que la salida del controlador sea más rápida y tienda a variar con mayor facilidad. Se esperaba que en cualquier prueba siempre el Resolver presentara mejor desempeño en comparación al Encoder pero en esta prueba se pudo observar como la respuesta del sistema fue mejor y más lineal para el Encoder.

En la práctica la resolución del Synchro - Resolver se ve limitada por diferentes factores como el conversor que se usa para obtener su medida de ángulo y la electrónica asociada a esta conversión. Por otro lado el encoder es un sensor con menor sensibilidad, pero responde adecuadamente para la aplicación en la cual se utilizó en el proyecto. Esto debido, en este proyecto, a que el encoder está acoplado directamente al eje del motor, mientras que el synchro se acopla al motor a través de discos o piñones, que además de agregar más carga y fricción, aumentan la posibilidad de error debido a su dentadura, quizá si se hubiese montado el Synchro Resolver conectado directamente al eje del motor, se esperaba un desempeño mayor al del encoder.

El Synchro - Resolver es un sensor capaz de entregar alta resolución, lo que lo hace muy útil en aplicaciones de precisión, donde los movimientos pueden ser muy pequeños y el con seguridad responderá de forma adecuada. El encoder también es un buen sensor, más común, sencillo, accesible y económico, se pueden encontrar con resoluciones de hasta 10000 pulsos por vuelta (y aumentar por software hasta 40000 utilizando encoders con 2 canales de salida), que en teoría supera la de un Synchro Resolver con un conversor análogo a digital de 12 bits por ejemplo. Sin embargo los Synchro Resolvers presentan otras ventajas sobre los Encoders como lo son su resistencia a ambientes extremos, errores de ciclo, excentricidad, errores debidos al ancho de pulso, densidad de conteo y fase, entre muchos otros. Robustez, resolución, exactitud, además de su inmunidad a la contaminación, humedad, altas temperaturas, vibraciones y su reducido momento de inercia que impone poca carga mecánica al eje al cual se acopla<sup>[6]</sup>.

No se logró mejorar en la práctica significativamente la respuesta del controlador en lazo cerrado a una entrada paso, ya que para mejorar los parámetros de desempeño, era necesario aumentar el valor de las constantes proporcional y derivativa, a tal punto que en la práctica se saturaba el controlador. Se logró hacer el sistema más rápido pero no se logro disminuir el sobrepico, después de numerosas pruebas realizadas.

## 7. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN

- [1] Basher, H.A.; Isa, S.A, *LabVIEW-Based Position Control System with Synchro*,\_SoutheastCon, 2005. IEEE, Page(s): 23 – 28.
- [2] Ayala,L;Cordoba,R;Martinez,A,“*Sistema de Medición Angular Micro controlado, de Alta Resolución, utilizando un Synchro*”. Bogota, Colombia, 2004, 135 p. Trabajo de Grado (Ingeniería Electronica). Pontificia Universidad Javeriana. Facultad de Ingeniería.
- [3] Boyes, Geoffrey S. (1980),“*Synchro and Resolver Conversion*”, Memory Devices.
- [4] J. Kessier, *Synchro/Resolver Conversion Handbook*, October 1994, < <http://www.ddc-web.com/Documents/synhdbk.pdf>>, 10 de Mayo de 2010.
- [5] Fiengo, R.; Caesar, G., *Virtual instrument for synchro-to-digital conversion*, Autotestcon, 2005. IEEE, Page(s): 413 – 415.
- [6] RVTD.  
[http://www.infopl.net/Documentacion/Docu\\_Instrumentacion/infoPLC\\_net\\_Medida\\_Desplazamiento.pdf](http://www.infopl.net/Documentacion/Docu_Instrumentacion/infoPLC_net_Medida_Desplazamiento.pdf). [Consultado el 11 de mayo de 2010].
- [7] Synchro.  
[http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=1423210&queryText%3Dsynchro+for+synchro%26openedRefinements%3D\\*%26searchField%3DSearch+All](http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=1423210&queryText%3Dsynchro+for+synchro%26openedRefinements%3D*%26searchField%3DSearch+All) [Consultado 12 de mayo de 2010].
- [8] synchro acquisition labview.  
<http://digital.ni.com/worldwide/arabia.nsf/web/all/A1FBAC9D090EF91D8625746E001E4548>. [Consultado 12 de mayo de 2010].
- [9] Trujillo,L;Prieto,A;Tinjaca,M, “*Control de Posición y Seguimiento para Telescopio con Montura Azimutal*”. Bogotá, Colombia, 2002, 105 p. Trabajo de Grado (Ingeniería Electronica). Pontificia Universidad Javeriana. Facultad de Ingeniería.
- [10] Eltra. Encoder Incremental. <http://bc.inter.edu/facultad/arincon/encoderIncrementales.pdf>. Consultado el 10 de Enero de 2011.
- [11] Danaher Industrial Controls. Encoder Application Handbook. [http://www.dynapar.com/uploadedFiles/Products/Danaher\\_Encoder\\_Handbook.pdf](http://www.dynapar.com/uploadedFiles/Products/Danaher_Encoder_Handbook.pdf). Consultado el 12 de Enero de 2011.
- [12] <http://ayudaelectronica.com/que-es-un-dsp/>. Consultado el 17 de Enero de 2011.
- [13] MUÑOZ, Andrea., PRIETO, Alberto., TINJACA, Ivan. Control de Posición y Seguimiento para Telescopio con Montura Azimutal. Pontificia Universidad Javeriana. 2003.
- [14] Analog Devices. AD2S1205 Datasheet (Rev. A). 2007-2010.



- [15] **Walt Kester**. Resolver-to-Digital Converters. MT-030. Analog Devices. Rev.A, 10/08.
- [16] AMETEK®. Motors Catalog. [www.ametektip.com](http://www.ametektip.com). 2007.
- [17] Microchip. dsPIC30F2010 DATASHEET. DS70118E. 2004.
- [18] Microchip. dsPIC30F Family Reference Manual. DS70046D. 2005.
- [19] Mark Schirmer. Using the 2S80 Series Resolver-to-digital Converters with Synchros: Solid-State Scott-T Circuit. AN-252 APPLICATION NOTE. Analog Devices.
- [20] Innovative Automation Control. <http://www.ctc-control.com/customer/elearning/servotut/svotyp.asp>. Consultado el 24 de mayo de 2011.
- [21] SGS – THOMSON Microelectronics. L62603 Datasheet. Julio 2007.

**ANEXO A**  
**CÓDIGO DEL PROGRAMA IMPLEMENTADO EN EL MICROCONTROLADOR**  
**dsPIC30F2010 (BASADO EN CÓDIGO SUMINISTRADO POR ING. CAMILO OTALORA).**

```

/*****
/*****
// CONTROL DE POSICION Utilizando SYNCHRO - RESOLVER Y ENCODER
/* El periodo de muestreo se fija con el timer 3. *
/* Reloj principal de 20 Mhz para usar con PLLx8 *
/* Se utiliza timer 3
/* Transmisión de datos por USART
//
/*****

/*****
#include "p30F2010.h"

// Constantes

#define PWM_PER 999 //La frecuencia del PWM es 20kHz = 20MHz / (999+1)
#define MAXPWM 1000 //El máximo ciclo util se obtiene cuando el conteo es de
2*(PWM_PER+1)
#define TIMER_PER 1562.5 //Frecuencia de Muestreo de 400Hz

#define enable PORTDbits.RD0 // enable para puente H.

// Rutinas de interrupción

void U1TXInterrupt(void);
void U1RXInterrupt(void);
void TxDat(int);
extern char BufRx[32];
extern char BufTx[32];

// Variables

unsigned int count1,count2,count3,k;

int XPOSCNT = 0;
int sensor;
int conteo=0;
float ref;
float Ang = 0;
float error = 0;
float aux = 0;
float refaux = 0;
float Angaux = 0;
float AngS=0;
float coefpS=0;
float coefiS=0;
float coefdS=0;
float sensorS=0;

```

```

int Dirstat;

int delay_mp = 0; //contador multiproposito

// Funciones de inicialización

void InitTIMER3(void);
void InitADC10(void);
void InitPWM(void);
void ConfigINTERRUPS(void);
void config_USART(void);
void configQEI(void);

int I;
char dat_ok;
int ch[5];
float coefp,coefi,coefd;
float outanti,outantd;
float outp,outd,outi;
signed int errant,out,coefpwm;

// Programa
void main(void)
{

//Configuración del procesador

    ConfigINTERRUPS();
    InitPWM();
    InitADC10();
    InitTIMER3();
    config_USART();
    configQEI();

    TRISD=0xFFF0;           //D0..D1 salida

    TRISBbits.TRISB1 = 0; //salida para realizar pruebas

    dat_ok=0;

//-----Inicializacion-----

ref=0;
coefp=0;
coefi=0;
coefd=0;
error = 0;
errant=0;
outanti=0;
outantd=0;
outi=0;

```

```

outp=0;
outd=0;
out=0;
sensor=0;

//-----

enable=1;          //enciende pwm

while(1){

if(dat_ok==1)      // entra cada 100 mseg.
{

    //LATBbits.LATB1 = 0;

    LATDbits.LATD1=1;

    BufTx[0]=ascii((char)(ch[0]>>12)); //Posicion Angular
    BufTx[1]=ascii((char)(ch[0]>>8));
    BufTx[2]=ascii((char)(ch[0]>>4));
    BufTx[3]=ascii((char)(ch[0]));
    BufTx[4]=0x20;

    BufTx[5]=ascii((char)(ch[1]>>12)); //Coeficiente P
    BufTx[6]=ascii((char)(ch[1]>>8));
    BufTx[7]=ascii((char)(ch[1]>>4));
    BufTx[8]=ascii((char)(ch[1]));
    BufTx[9]=0x20;

//

    BufTx[10]=ascii((char)(ch[2]>>12)); //Coeficiente I
    BufTx[11]=ascii((char)(ch[2]>>8));
    BufTx[12]=ascii((char)(ch[2]>>4));
    BufTx[13]=ascii((char)(ch[2]));
    BufTx[14]=0x20;

//

    BufTx[15]=ascii((char)(ch[3]>>12)); //Coeficiente D
    BufTx[16]=ascii((char)(ch[3]>>8));
    BufTx[17]=ascii((char)(ch[3]>>4));
    BufTx[18]=ascii((char)(ch[3]));
    BufTx[19]=0x20;                                     // toma 5 uS @ 20 Mips

    BufTx[20]=ascii((char)(conteo>>12));
    BufTx[21]=ascii((char)(conteo>>8));
    BufTx[22]=ascii((char)(conteo>>4));
    BufTx[23]=ascii((char)(conteo));
    BufTx[24]=0x20;
}
}

```

```

//          BufTx[25]=ascii((char)(ch[4]>>12));
//          BufTx[26]=ascii((char)(ch[4]>>8));
//          BufTx[27]=ascii((char)(ch[4]>>4));
//          BufTx[25]=ascii((char)(ch[4]));          //Sensor
//          BufTx[26]=0x20;

//          BufTx[27]=0x0d;

LATDbits.LATD1=0;

TxDat(28);          //strlen(BufTx); // envio toma 440uS @ 115200

dat_ok=0;
    }
    }
}

// Interrupciones

void __attribute__((__interrupt__)) _ADCInterrupt(void)
{
//if(ref>360)ref=0;          //Para generar rampa
//ref=ref+0.05;
//if (0 <= ref && ref <= 360)
{
Dirstat=QEICONbits.UPDN;

//-----
//-----Control Encoder-----
//-----

if(sensor==0){

AngS=Ang;
coefpS=coefp*1000;
coefiS=coefi*1000;
coefdS=coefd*1000;
sensorS=sensor*1000;

conteo=POSCNT-3000;
Ang = conteo*0.36; //Se asigna el valor del angulo de posicion de acuerdo a la resolucio del encoder
error = ref - Ang;

// voltaje de entrada para control PWM en AN1

LATDbits.LATD1=~PORTDbits.RD1;          // bit to test

if(count1>0) count1--;

```

```

ch[0]=AngS; //Posición Angular
ch[1]=coefpS; //Coeficiente P
ch[2]=coefiS; //Coeficiente I
ch[3]=coefdS; //Coeficiente D
ch[4]=sensorS; //Sensor

//-----PID-----

outp=(error)*coefp; //Salida Proporcional

outi=(error+errant)*coefi+outanti; //Salida Integral

outd=(error-errant)*coefd-(outantd); //Salida Derivativa

outantd=outd;
outanti=outi;
errant=error;

out=(int)(outp+outi+outd)+1000;

//-----

PDC1=out; // SALIDA A PWM
dat_ok=1;
IFS0bits.ADIF = 0; //limpiar interrupción

}
else

if ( ref > 360 || ref < 0)
{
out = 1000 ;
PDC1 = out;
IFS0bits.ADIF = 0;
}

//-----
//-----Control Synchro-----
//-----

else if(sensor==1){

AngS=Ang;
coefpS=coefp*1000;
coefiS=coefi*1000;
coefdS=coefd*1000;
sensorS=sensor*1000;

conteo=POSCNT-3000;

```

```

Ang = conteo*0.17578125; //Se asigna el valor del angulo de posicion de acuerdo a la resolucion del
synchro - resolver
error = ref - Ang;

```

```

// voltaje de entrada para control PWM en AN1

```

```

LATDbits.LATD1=~PORTDbits.RD1; // bit to test

```

```

    if(count1>0) count1--;

```

```

ch[0]=AngS; //Posición Angular
ch[1]=coefpS; //Coeficiente P
ch[2]=coefiS; //Coeficiente I
ch[3]=coefdS; //Coeficiente D
ch[4]=sensorS; //Sensor

```

```

//-----PID-----

```

```

outp=(error)*coefp; //Salida Proporcional

```

```

outi=(error+errant)*coefi+outanti; //Salida Integral

```

```

outd=(error-errant)*coefd-(outantd); //Salida Derivativa

```

```

outantd=outd;

```

```

outanti=outi;

```

```

errant=error;

```

```

out=(int)(outp+outi+outd)+1000;

```

```

//-----

```

```

    PDC1=out; // SALIDA A PWM
    dat_ok=1;
    IFS0bits.ADIF = 0; //limpiar interrupción

```

```

    }

```

```

else

```

```

    if ( ref > 360 || ref < 0)

```

```

    {

```

```

        out = 1000 ;

```

```

        PDC1 = out;

```

```

        IFS0bits.ADIF = 0;

```

```

    }

```

```

}

```

```

}

```

```

void InitADC10(void) // OK- 18-agosto-2005.

```

```

{

```

```

//      ADCBUF0= AN3
//      ADCBUF1= AN0
//      ADCBUF2= AN1
//      ADCBUF3= AN2

ADPCFG = 0xFF00; // Config. entrada analoga pines AN0,AN1, AN2, AN3
ADCHS = 0x0004; // CH0-->AN4 MUXB no importa, CH123neg=Vref-, CH1pos=AN0, CH0neg=Vref-,
Ch0pos=AN1
// 0 0 0 0

ADCSSL = 0x0000; // No importa, el canal0 no realiza secuencias.
ADCON3 = 0x0007; // sample time fijado por TMR3, internal clock, Tad=4Tcy

//
ADCON2 = 0x020C; // Vref=AVdd, No scan, convierte CH0, CH1,CH2,CH3
Int. cada 4 samples
ADCON1 = 0x804C; // ADC on, halt en idle, integer., trig timer3, muestreo
simultaneo, // automuestreo,

/*
1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 */

}

void InitTIMER3(void) // Para interrupcion a 400Hz.
{
TMR3 = 0; // Reinico del contador
PR3 = TIMER_PER; // Periodo de conteo
T3CON = 0xA010; // Configuración timer activo, halt en idle,
prescaler=8, gated desactivado
}

void InitPWM(void)
{
PTPER = PWM_PER; // Periodo de PWM = 50useg.
SEVTCMP = 0x0000; // Comparación de eventos especiales, deshabilitada
PWMCON1 = 0x0077; // Pines habilitados como PWM
(PWM1,2,3),complementarios
PWMCON2 = 0x0000; // Postscale para eventos especiales, override sync y
update sync.
DTCON1 = 0x0000; // configuración de Dead time, prescaler y tiempo
FLTACON = 0x0000; // Config. Fault A, deshabilitado para todos los pines
OVDCON = 0xFF00; // Control de override, sobreescritura deshabilitada
PDC1 = 1000; // Ciclo util inicial = 50%
PDC2 = 1000;
PDC3 = 1000;
PTCON = 0xA000; // PWM control, activo, halt en idle, pre y post
scale =0, conteo free running
}

```



```

void ConfigINTERRUPS(void)
{
INTCON1 = 0x0000;           // Int. anidadas = deshabilitadas, traps deshabilitados
INTCON2 = 0x0000;           // vector estandar, int. externas= flanco positivo.
IFS0  = 0x0000;           // Limpia interrupciones pendientes
IFS1  = 0x0000;           // Limpia interrupciones pendientes
IFS2  = 0x0000;           // Limpia interrupciones pendientes
IEC0  = 0x0A48;           // Int. desable, ADC int enable,RX1INT enable, TMR2 int
enable.
IEC1  = 0x0000;           // Int. desable
IEC2  = 0x0000;           // Int. desable
IPC0  = 0x4000;           // Prioridades : Timer1 =4;
IPC1  = 0x0500;           // Prioridades = 0 Timer2 =5
IPC2  = 0x6130;           // Prioridades = 0, adc =6, TX1=1; RX1=1
IPC3  = 0x0000;           // Prioridades = 0
IPC4  = 0x0000;           // Prioridades = 0
IPC5  = 0x0000;           // Prioridades = 0
//IPC6 = 0x0000;           // Prioridades = 0
//IPC7 = 0x0000;           // Prioridades = 0
//IPC8 = 0x0000;           // Prioridades = 0
//IPC9 = 0x0000;           // Prioridades = 0
//IPC10 = 0x0000;          // Prioridades = 0
//IPC11 = 0x0000;         // Prioridades = 0
}

```

```

void configQEI(void)
{
ADPCFG = 0x003F; // Configure QEI pins as digital inputs
//QEICONbits.QEIM = 0; // Disable QEI Module
QEICONbits.CNTERR = 0; // Clear any count errors
QEICONbits.QEISIDL = 0; // Continue operation during sleep
QEICONbits.SWPAB = 0; // QEA and QEB not swapped
QEICONbits.PCDOUT = 0; // Normal I/O pin operation
//QEICONbits.POSRES = 1; // Index pulse resets position counter
DFLTCONbits.CEID = 0; // Count error interrupts disabled
DFLTCONbits.QEOUT = 1; // Digital filters output enabled for QEn pins
DFLTCONbits.QECK = 5; // 1:64 clock divide for digital filter for QEn
//DFLTCONbits.INDOUT = 1; // Digital filter output enabled for Index pin
//DFLTCONbits.INDCK = 6; // 1:128 clock divide for digital filter for Index
POSCNT = 3000; // Reset position counter
QEICONbits.QEIM = 5; // 2x mode with position counter reset by match maxcnt
MAXCNT = 60000; //
}

```

**ANEXO B**  
**CÓDIGO DE LA INTERFAZ AL USUARIO (BASADO EN CÓDIGO SUMINISTRADO POR**  
**ING. CAMILO OTALORA).**

```
#include "p30F2010.h"

char const MaxBufRx = 8 ; //maximo de bytes en buffer RX.
char const MaxBufTx = 32 ; //maximo de bytes en buffer TX.
char  ap_bufRx; // apunadores a buffer de TX, Rx
char  ap_bufTx;
char  ENTER, char_in; // indica que llego ENTER
char  NB_tx; // bytes a transmitir.
char  BufRx[32];
char  BufTx[32];
int    in_serial;

float  temp=0;
float  tempP=0;
float  tempI=0;
float  tempD=0;
int    tempS=0;
int    tempQ=0;
extern float ref;
extern float coefp;
extern float coefi;
extern float coefd;
extern int sensor;

/* This is UART1 receive ISR */

// entrada de datos de la forma en ascii PdXXXX 0x0d
// d es el setpoint a cambiar.
// XXXX con ceros en la izquierda. valor decimal entre 0000 y 1024

void __attribute__((__interrupt__)) _U1RXInterrupt(void)
{

BufRx[ap_bufRx]= U1RXREG;      // almacena dato recibido.
char_in=BufRx[ap_bufRx];

if(ap_bufRx>=MaxBufRx)

    {
        ap_bufRx=0 ; // si pasa maximo de datos, pasa a cero.
    }

else ap_bufRx++;

if(char_in==0x0D)    // ENTER    y longitud correcta
    {
        if(ap_bufRx==6)
            {
```

```

if(BufRx[0]=='r')    // cambio de tiempo de muestreo. 5000=1000HZ
{

    temp=(BufRx[1]&0x0f)*1000;
    temp=temp+(BufRx[2]&0x0f)*100;
    temp=temp+(BufRx[3]&0x0f)*10;
    temp=temp+(BufRx[4]&0x0f);

    if(temp>=0 && temp<=360)    //
    {
        ref=temp;
    }

    ap_bufRx=0; // si detecta inicio reinicializa apuntador.

    ENTER=1;
}

if(BufRx[0]=='p')    // cambio de tiempo de muestreo. 5000=1000HZ
{

    tempP=(BufRx[1]&0x0f);
    tempP=tempP+(BufRx[2]&0x0f)*0.1;
    tempP=tempP+(BufRx[3]&0x0f)*0.01;
    tempP=tempP+(BufRx[4]&0x0f)*0.001;

    if(tempP>=0 && tempP<=20)    //
    {
        coefp=tempP;
    }

    ap_bufRx=0; // si detecta inicio reinicializa apuntador.

    ENTER=1;
}

if(BufRx[0]=='P')    // cambio de tiempo de muestreo. 5000=1000HZ
{

    tempP=(BufRx[1]&0x0f)*1000;
    tempP=tempP+(BufRx[2]&0x0f)*100;
    tempP=tempP+(BufRx[3]&0x0f)*10;
    tempP=tempP+(BufRx[4]&0x0f);

    if(tempP>=0 && tempP<=20)    //
    {
        coefp=tempP;
    }

    ap_bufRx=0; // si detecta inicio reinicializa apuntador.
}

```

```

        ENTER=1;
    }
    if(BufRx[0]=='i')    // cambio de tiempo de muestreo. 5000=1000HZ
    {

        tempI=(BufRx[1]&0x0f);
        tempI=tempI+(BufRx[2]&0x0f)*0.1;
        tempI=tempI+(BufRx[3]&0x0f)*0.01;
        tempI=tempI+(BufRx[4]&0x0f)*0.001;

        if(tempI>=0 && tempI<=20)    //
        {
            coefi=tempI;
        }

        ap_bufRx=0; // si detecta inicio reinicializa apuntador.

        ENTER=1;
    }

    if(BufRx[0]=='d')    // cambio de tiempo de muestreo. 5000=1000HZ
    {

        tempD=(BufRx[1]&0x0f);
        tempD=tempD+(BufRx[2]&0x0f)*0.1;
        tempD=tempD+(BufRx[3]&0x0f)*0.01;
        tempD=tempD+(BufRx[4]&0x0f)*0.001;

        if(tempD>=0 && tempD<=20)    //
        {
            coefd=tempD;
        }

        ap_bufRx=0; // si detecta inicio reinicializa apuntador.

        ENTER=1;
    }

    if(BufRx[0]=='s')    // cambio de tiempo de muestreo. 5000=1000HZ
    {

        tempS=(BufRx[1]&0x0f)*1000;
        tempS=tempS+(BufRx[2]&0x0f)*100;
        tempS=tempS+(BufRx[3]&0x0f)*10;
        tempS=tempS+(BufRx[4]&0x0f);

        if(tempS>=0 && tempS<=1)    //
        {
            sensor=tempS;
        }
    }

```

```

        ap_bufRx=0; // si detecta inicio reinicializa apuntador.

        ENTER=1;
    }

} // end if(ap_bufRx=6) //

    else // if(ap_bufRx=6) //
    {
        ap_bufRx=0; // si hay error reinicializa apuntador
        ENTER=0;
    }
}

else // char_in==0x0D
{
    ENTER=0;
}

IFS0bits.U1RXIF = 0; // borra INT.

}

void __attribute__((__interrupt__)) _U1TXInterrupt(void)
    // Rutina para Tx emty buffer. debe habilitarse INT en rutina de TX.
    // transmite NB_tx caracteres del buffer de transmicion. (asci_buf )
    // rutina de Tx debe colocar en cero ap_bufTx
{

if(NB_tx == 0) // ha transmitido todos los datos.
    {
        IEC0bits.U1TXIE=0; // deshabilita INT
        //U1STAbits.UTXEN=0; // deshabilita TX.
    }

else
    {
        while(NB_tx>0 && U1STAbits.UTXBF==0) // coloca datos en buffer de tx hasta que
                                                // no hay mas
datos o se llena el buffer
        {
            U1TXREG=BufTx[ap_bufTx++]; // Coloca dato a tx en TxReg. e incrementa apuntador.
            NB_tx--;
        }

    }

IFS0bits.U1TXIF = 0; // borra interrupción
IEC0bits.U1TXIE=1; // habilita INT
}

```

```

void TxDat(char N) // coloca numero de bytes a Tx y habilita int para Tx.
{
    U1STAbits.UTXEN=0;      // deshabilita TX.
    ap_bufTx=0;
    NB_tx=N;
    IEC0bits.U1TXIE=0;      // borra bandera de INT
    IEC0bits.U1TXIE=1;      // habilita INT
    U1STAbits.UTXEN=1;      // habilita TX.
}

void config_USART(void) // 115200,n,8,1
{
    U1MODE=0;                // en ceros
    U1MODEbits.UARTEN=1;     // habilita UART // b10000000000000; 8 bits no paridad, no loopback,
    ON.
    U1MODEbits.ALTIO=1;     // usa bits alternos para comunicaciones.

    U1STA=0x8000;           //b 1000 0000 0000 0000 Tx on , INT cada caracter.

    U1BRG=10;              // para velocidad 57600x2. (Fcicle/(16BaudRate) - 1 ;

    IFS0bits.U1RXIF = 0;
    IEC0bits.U1TXIE=0; // deshabilita INT Tx UART1.
    IEC0bits.U1RXIE=1; // habilita INT RX 1. transmite por UART1

}

char ascii ( char hex)
{
    hex=hex&0x0f ;
    if (hex >= 0x0a) hex=hex+0x37 ;
    else hex=hex+0x30 ;
    return (hex) ;
}

```

**ANEXO C**  
**EJEMPLOS PARA UTILIZAR INTERFAZ AL USUARIO CON HYPERTERMINAL**

En la ventana de Hyperterminal se digitan seis caracteres.

**Para la referencia:**

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>letra</b>	<b>0</b>	<b>1digito</b>	<b>2digito</b>	<b>3digito</b>	<b>enter</b>

Ejemplo 1: 8°, r 0 0 0 8 enter; Ejemplo 2: 45°, r 0 0 4 5 enter, Ejemplo 3: 270°, r 0 2 7 0 enter.

Nota:  $0^\circ \leq r \leq 360^\circ$

**Para las constantes Proporcional, Integral y Derivativa:**

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>letra</b>	<b>1digito</b>	<b>2digito</b>	<b>3digito</b>	<b>4digito</b>	<b>enter</b>

Es igual para las tres solo cambia la letra, se escribe p, i o d, hay que llenar los 4 valores, para los valores enteros se escribe en el primer digito y los demás se llenan con ceros; para los decimales se omite la coma, se puede escribir hasta tres decimales:

Ejemplo 1: constante p = 3, p 3 0 0 0 enter.

Ejemplo 2: constante i = 4,52, i 4 5 2 0 enter.

Ejemplo 3: constante d = 0,032 d 0 0 3 2 enter.

Ejemplo 4: constante p = 0,9 p 0 9 0 0 enter.

Ejemplo 5: constante d = 0,001 d 0 0 0 1 enter.

Nota:  $0,000 \leq P, I, D \leq 9,999$

**Para escoger el control de acuerdo al sensor:**

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>letra</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>digito</b>	<b>enter</b>

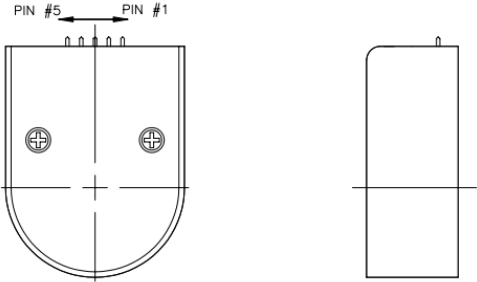
El valor del quinto digito es cero por defecto.

Ejemplo 1: sensor encoder, s 0 0 0 0 enter

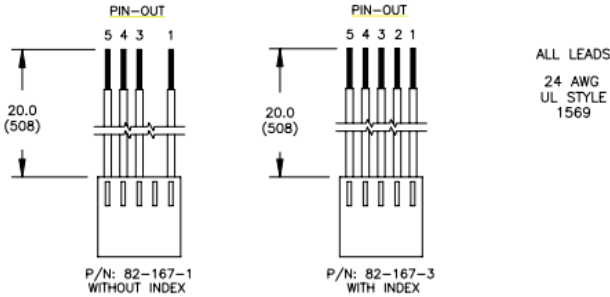
Ejemplo 2: sensor synchro – resolver, s 0 0 0 1 enter.

**ANEXO D**  
**DIAGRAMAS DE CONEXIÓN ENCODER E30**

**E30 INCREMENTAL OPTICAL ENCODER**



**Optional Leadwire Assemblies**

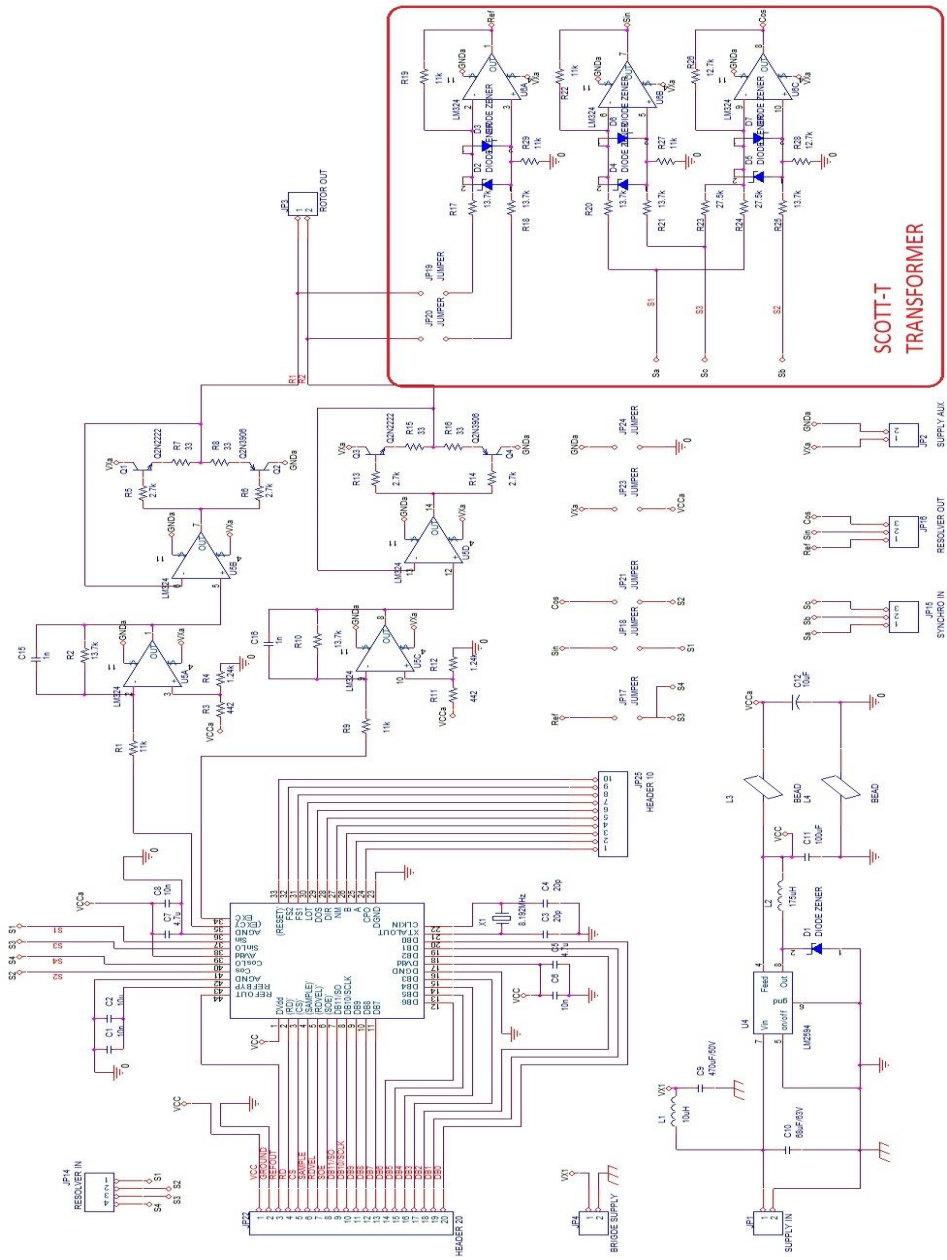


<b>E30A Connection Chart</b>		
Pin	Color	Function
1	Black	Ground
2	-	No Connection
3	Yellow	Channel A
4	Red	V <sub>cc</sub>
5	Blue	Channel B



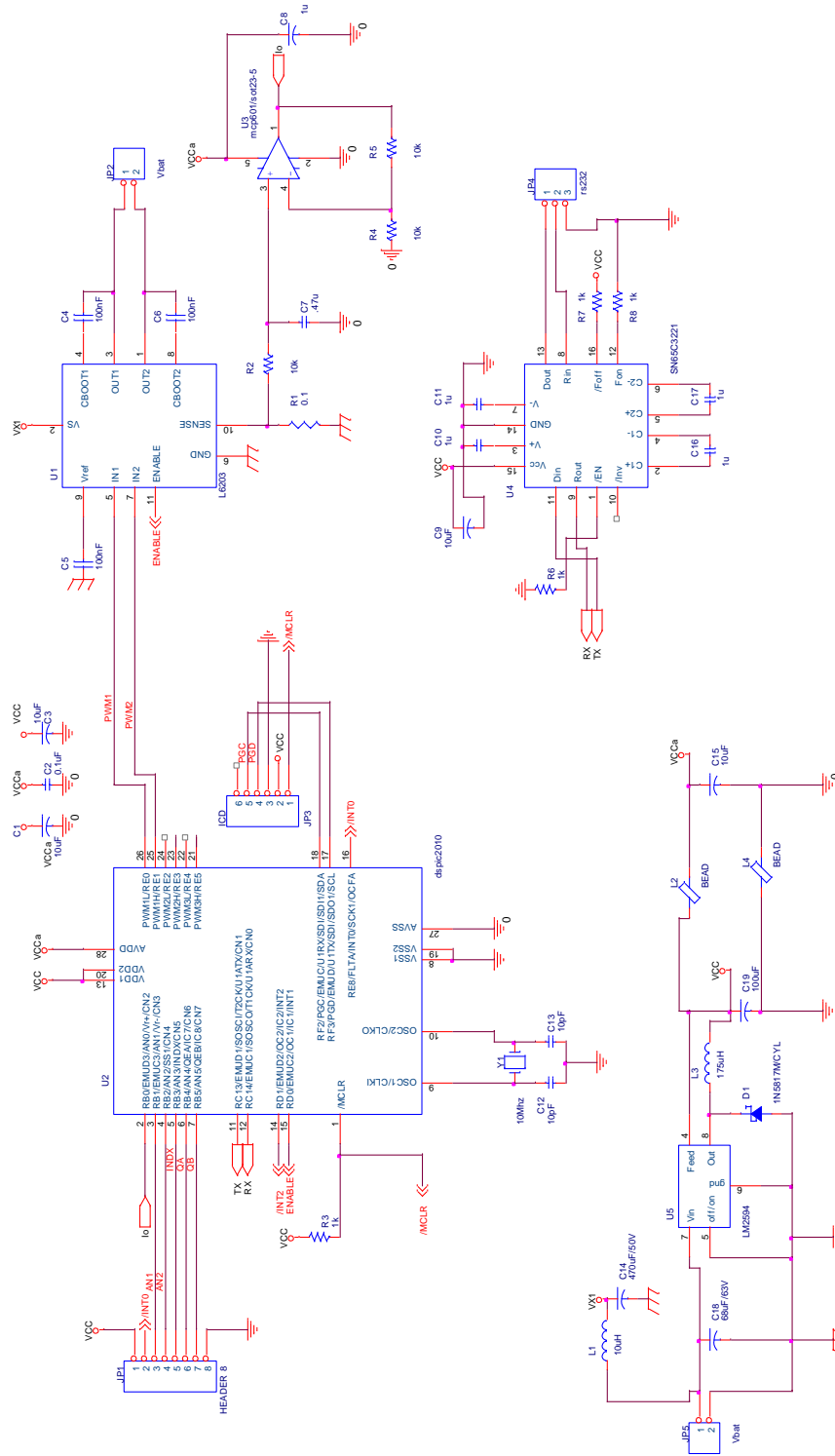
# ANEXO E

## ESQUEMATICO TARJETA DE DESARROLLO PARA AD2S1205



# ANEXO F

## ESQUEMÁTICO TARJETA DE DESARROLLO PARA dsPIC30F2010 (ING. CAMILO OTALORA).



**ANEXO G**  
**PROPUESTA DE EXPERIENCIA DE LABORATORIO**

TITULO:

Medición y control de posición angular empleando synchro resolver y encoder.

OBJETIVOS:

- Observar y comparar el funcionamiento y características del synchro resolver y encoder como sensores de posición.
- Desarrollar y sintonizar un control PID para la posición angular del motor pittman 9233 24V.
- Explorar las propiedades de las acciones proporcional, derivativa e integral.

EQUIPO NECESARIO:

- Fuente de Voltaje Sencilla (24 V, 3 Amp.).
- Computador con Windows XP , software hyperterminal y puerto serial.
- Sistema de Medición y control de posición angular empleando synchro resolver (T.G 1022).

TRABAJO PREVIO:

Con base en la función de transferencia del motor pittman 9233 24 V. Simule su respuesta en malla abierta, realimentación unitaria y desarrolle diferentes controladores PID para la posición angular del motor por el método que desee, que permitan mejorar los parámetros de desempeño del sistema. Simule con entradas de tipo paso y triangulo. Tenga en cuenta que la función de transferencia tiene como salida velocidad.

$$\frac{W(s)}{Vin(s)} = \frac{K_T}{JLS^2 + S(RJ + B_t L) + (RB_t + K_v K_t)}$$

$$J = 1,7933e-5 \text{ kg}\cdot\text{m}^2$$

$$B_t = 2e-6 \text{ Nm s/rad}$$

$$K_v = 0.0369 \text{ V/rad/s}$$

$$K_t = 0.0369 \text{ Nm/A}$$

$$R = 3.6 \Omega$$

$$L = 3.61 \text{ mH}$$

Con base en las constantes obtenidas para cada controlador, realice la conversión de continuo a discreto para cada constante (esto debido a que las constantes que recibe el sistema son las discretas). Para esto utilice la función tustin de Matlab con un tiempo de muestreo de 400 Hz.

Estudie brevemente como configurar hyperterminal para enviar y recibir datos a través de puerto serial y guardarlos para luego exportarlos a una hoja de cálculo como Excel. No olvide llevar sus resultados al laboratorio.

## PROCEDIMIENTO:

1. Revise las conexiones internas del sistema, no deben haber cables sin conexión.
2. Conecte la salida serial del sistema al PC.
3. Lea el Anexo C de la tesis “Medición y Control de posición angular empleando synchro resolver” para configurar el sistema de control mediante hyperterminal. (T.G. 1022)
4. Alimente el sistema con la fuente de voltaje a 24 V. Tenga mucho cuidado en conectar correctamente los cables, ya que podría dañar el convertidor.
5. Verifique que los leds rojos (que indican error en la medición con resolver) estén apagados, si por alguna razón hay alguno encendido, presione reset y se deberán apagar. Es posible que cuando el sistema se vuelva inestable, estos leds se enciendan. Uno indica pérdida de tracking por parte del convertidor (LOT) y el otro degradación de señal (DOS). Estos leds sólo indican errores para el resolver.
6. Es necesario manualmente cambiar de posición los jumpers para seleccionar sensor encoder o resolver.
7. Inicie la comunicación a través de hyperterminal.
8. Pruebe sus controladores hallados previamente, enviando datos desde el PC al dspic, por hyperterminal (En pantalla no se observan los datos que está escribiendo para enviar así que debe ser cuidadoso en la digitación de datos siguiendo el Anexo C. Solo verá los datos que se están recibiendo). Guarde los datos obtenidos e impórtelos a Excel. Estos datos pertenecen a la posición del motor y son muestreados cada 2,5 ms. Están en formato Hexadecimal. Una vez en Excel, haga la conversión a decimal. Grafique sus respuestas con el tiempo de muestreo mencionado.
9. Si el sistema se volvió inestable, modifique sus constantes, puede empezar probando una realimentación unitaria e ir modificando moderadamente las constantes derivativa e integral, hasta ver que el sistema aparentemente controle y luego si corroborar con los datos medidos y ajustan el control.
10. Realice de 3 a 5 pruebas con diferentes constantes.

## ANALISIS DE RESULTADOS

Realice la comparación teórico-práctica de los controles usados. Responda las siguientes preguntas:

¿Son estos semejantes?

¿Concuerda lo simulado con lo real?

¿Qué posibles causas justifican estas diferencias?

¿Para qué aplicaciones sería válido un sistema rápido sacrificando sobrepico, y viceversa; un sistema más lento pero sin sobrepico? (Manteniendo un error en estado estable nulo o lo mínimo que permita la resolución según la conversión a digital)