



INSTITUTO POLITÉCNICO NACIONAL

**Escuela Superior de Ingeniería Mecánica y
Eléctrica
Unidad Azcapotzalco**

Sección de Estudios de Posgrado e Investigación

**“TÉCNICAS DE CONTROL DE
POSICIONAMIENTO DE ESLABONES PARA UN
BRAZO DE 4 GRADOS DE LIBERTAD”.**

T E S I S

**QUE PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERÍA DE MANUFACTURA**

P R E S E N T A

ING. FUNES LORA MIGUEL ANGEL

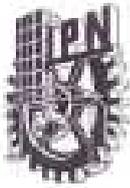
DIRECTORES:

**DR. JOSÉ DE JESÚS RUBIO AVILA
M. EN C. RAUL RIVERA BLAS**

MÉXICO, D. F. JUNIO DE 2011.



ESIME AZCAPOTZALCO INSTITUTO POLITÉCNICO NACIONAL



SEP-16-BIS

INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:00 horas del día 30 del mes de Mayo del 2011, se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de ESIME UA para examinar la tesis titulada:

"TÉCNICAS DE CONTROL DE POSICIONAMIENTO DE ESLABONES PARA UN BRAZO DE 4 GRADOS DE LIBERTAD".

Presentada por el alumno:

Funes Miguel Angel
Apellido paterno Lora Apellido materno Miguel Angel Nombre(s)
Con registro:

B	0	9	1	5	6	9
---	---	---	---	---	---	---

aspirante al grado de:

Maestro en Ingeniería de Manufactura

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de tesis


Dr. José de Jesús Rubio Avila
1er Vocal

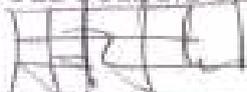

Dr. Jaime Picheco Martínez
Presidente


M. en C. Raúl Rivera Blas
2do Vocal


Dr. Salvador Antonio Rodríguez Paredes
Secretario


Dra. María Concepción Escamela García
3er Vocal

PRESIDENTE DEL COLEGIO DE PROFESORES DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN


Dr. Jaime Picheco Martínez



UNIDAD PROFESIONAL
AZCAPOTZALCO
SECCIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 25 del mes Mayo del año 2011 , el (la) que suscribe Funes Lora Miguel Angel alumno (a) del Programa de Maestría en Ingeniería de Manufactura con número de registro B091569, adscrito a SEPI ESIME UA, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. José de Jesús Rubio Ávila. y el M. en C. Raúl Rivera Blas. y cede los derechos del trabajo intitulado **“TÉCNICAS DE CONTROL DE POSICIONAMIENTO DE ESLABONES PARA UN BRAZO DE 4 GRADOS DE LIBERTAD”.** al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguientes direcciones jrubioa@ipn.mx; rrblas@hotmail.com; apertura_dr@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Funes Lora Miguel Angel

Nombre y firma



Agradecimientos

Agradezco a Dios por darme el privilegio de continuar aprendiendo a ser mas fuerte mi fe, a mis padres por proveerme de lo indispensable y poder continuar con mis estudios, al CONACYT por poderme apoyar en mi tiempo como estudiante, al IPN que ha sido como otra casa para mí y gracias a aquellos maestros que han llegado a ser parte de mi familia, que siempre me han apoyado, con los que he trabajado, me he divertido y con los que me han dado consejos y comentarios, les agradezco pues todos ustedes me han formado y seguirán formando un mundo mejor...

Resumen

Este trabajo se basa en la utilización de Técnicas de Control Difuso para el posicionamiento independiente de motores de corriente directa y la comunicación entre un Robot de 4 Grados de Libertad utilizando código Matlab y programación en paralelo por medio de la paquetería de LabView. Para la utilización de la Cinemática Inversa y Directa. En la ubicación de coordenadas y la generación de Trayectorias.

Abstract

This work is based on the use of Fuzzy Control Techniques for independent positioning of DC motors and the communication between a Robot of 4 degrees of freedom using Matlab code and parallel programming using the LabView package. To use Forward and inverse kinematics. In the location coordinates and trajectory generation.

Contenido

Antecedentes.....	8
Introducción.....	10
1 ESTADO DEL ARTE.....	14
2 <i>MARCO TEÓRICO</i>	19
2.1 Desarrollo del Proyecto	20
2.2 Diseño mecánico.....	20
2.2.1 Modelado Matemático del Brazo.....	20
2.3 Control del sistema	21
2.3.1 Simulación y Experimentación.....	21
3 DISEÑO MECÁNICO Y MODELADO MATEMATICO DEL BRAZO.....	22
3.1 Diseño mecánico del prototipo	23
3.2 Cinemática del prototipo de 4 Grados de Libertad	32
3.2.1 Ecuaciones Cinemáticas para los manipuladores	33
3.2.2 Cinemática Directa del Prototipo	34
3.2.3 Cinemática inversa del prototipo.....	38
3.3 Cinemática de un Robot Industrial de 6 grados de Libertad.....	47
3.3.1 Cinemática directa del Robot industrial	49
3.3.2 Cinemática Inversa de un robot de 6 grados de Libertad	53
4 CONTROL DEL BRAZO.....	57
4.1 Control con PID de un motor para brazo robótico industrial.....	58
4.1.1 Diseño del control.....	58
4.1.2 Resultados de Simulación.....	66
4.1.3 Estabilidad por Ruth-Hurwitz.....	74
4.1.4 El primer método de sintonización Ziegler Nichols	77
4.1.5 Sintonización por el método de Ziegler Nichols.....	84
4.1.6 Control de Posicionamiento PID en un Motor de Corriente directa por medio de LabVIEW.....	93
4.1.7 Resultados experimentales	98
4.1 Experimentación	110
4.2 Control PID del prototipo	112
4.2.1 Diseño del control.....	112
4.2.2 Resultados experimentales	114
4.2.3 ¿Qué procedimiento es necesario para poder controlar cada articulación?..	115

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

4.2.4	Relevancias que tiene el software National Instruments.....	122
5	CONCLUSIONES Y TRABAJO FUTURO.....	140
5.1	Conclusiones.....	141
5.2	Trabajo futuro.....	142
6	ANEXOS.....	144
	REFERENCIAS.....	198

Antecedentes

Hoy en día los robots han sido muy exitosos en la manipulación y simulación de ambientes controlados; tales como fábricas, ensambladoras y plantas automotrices. Pero fuera de estos lugares los robots pueden llegar a tener tareas sofisticadas cuando son solo operadas por un humano.

Muchos robots virtuales usan algoritmos para la planificación de movimiento con el fin de buscar caminos a través de un espacio de estado que modele la cinemática y dinámica de su espacio de trabajo. (LaValle, 2006). La Percepción de los Robots en la simulación y la manipulación en ambientes controlados indican que puedan identificar en qué posición y orientación se encuentran con el mundo que los rodea. Aunque los robots trabajan en los entornos humanos estos se encuentran limitados con problemas prominentes a su visión y a un mundo cambiante. Los sistemas de percepción tienen el potencial de reducir esta incertidumbre y permitir su funcionamiento autónomo. Como tal, la percepción es uno de los más importantes desafíos que enfrenta el campo de la robótica.

A través de sensores los robots pueden sencillamente percibir, medir e interactuar con su medio. Por ejemplo un robot puede seleccionar posturas obtenidas de un orden visual y utilizarlas para una tarea específica.

En el *Massachusetts Institute of Technology* (MIT), los robots visuales a menudo inducen a percibir mejor el movimiento del mundo. Por ejemplo, haciendo girar un efector final con una herramienta, como un destornillador o un bolígrafo, Un robot puede utilizar Domo visión monocular para buscar movimiento rápido convexo en regiones irregulares con el fin de detectar con firmeza la punta de una herramienta y su control. (Edsinger, et al., 2005).

De acuerdo a (Sian, et al., 2006) La visión es probablemente la modalidad mas estudiada de percepción de la máquina. Gran parte de su investigación presentada involucra de alguna forma visión. Por ejemplo, la investigación de la NASA / JSC con Robonaut (véase la fig. 1.2) y la investigación de AIST con HRP-2 utilizan la base de modelos para la percepción visual. Cada robot tiene un pequeño número de modelos 3-D para conocer los objetos que pueden ser combinados. De esta manera registra a los objetos vistos por la cámara estéreo a fin de permitir a los robots realizar tareas; como la apertura de un refrigerador o la acción de recoger una caja con las dos manos. Hasta el momento, la capacidad de estos sistemas de visión han respondido de manera fiable a un gran número de objetos manipulables en la vida cotidiana. (Platt, et al., 2006).

Tipos de Sensores

La Percepción táctil desde el robot de manipulación se basa fundamentalmente en el contacto entre el mismo y el mundo que lo rodea y un ejemplo de cómo interactuarlo sería por medio de una activación por pulsador, “*Push Botón*”. Otro ejemplo es el uso de los sensores de visión, ultrasónicos, tacto y un comportamiento basado en algoritmos; como ejemplo existen los robots humanoides. Obreros que han sido capaces de tocar y localizar la posición de su mano alrededor de los objetos de baja masa y a continuación, captar, ascender y colocarlos en lugares diferentes sin usar un modelo de objetos definido (Natale,2009). Sin embargo aun no se ha implementado de forma radical el desarrollo de control para el reconocimiento de objetos y la toma de decisiones de forma independiente; para robots con tareas industriales, lo cual puede ayudar a mejorar el desempeño de estos en el área de trabajo.

Introducción

Alrededor de la historia de la humanidad nuestra incapacidad para repetir procesos o crear objetos casi idénticos ha empujado a una gran cantidad de culturas a buscar maneras de reemplazar el trabajo humano por sistemas que puedan desempeñar estos procesos, aunque muchos de estos implementos rudimentarios eran en su mayoría mecánicos y solo podían realizar funciones básicas; conforme fue pasando el tiempo se encontraron técnicas que mejorarían el desempeño de estos mecanismos, una de ellas es el procesamiento de imágenes; el cual menciona González y Woods, las investigaciones se remontan al siglo XX, cuando se transmitían imágenes transoceánicas a través de cable submarino. Las fotografías periodísticas entre Europa y América tardaban una semana en llegar a través de los barcos. Al emplear las primeras técnicas de procesamiento de imágenes se logró reducir su tiempo de transmisión a sólo a tres horas. Las imágenes se codificaban a cinco niveles de grises y se transmitían por teléfono. No obstante, éste podría ser el principio de las técnicas de procesamiento de las imágenes, sin embargo no el de la Visión Artificial. El concepto de Visión Artificial es más amplio y recupera para sí, todos los conocimientos de análisis de las imágenes desempeñado por otras disciplinas desde los inicios de la fotografía.

Sin embargo un tema más relevante para la elaboración de un robot de 4 grados de libertad es por el momento un seguidor de trayectoria. El cual está definido como un dispositivo que se encarga de seguir trayectorias por medio de algún tipo de reconocimiento visual (censado electrónico), que permite que un robot pueda seguir una trayectoria posicionada por su efector final cuando este se ve afectado por dos definiciones que representan la columna vertebral de la programación; la inserción de la Cinemática Directa y la Cinemática Inversa.

OBJETIVO GENERAL

El objetivo general del proyecto se basa en el control de posicionamiento individual para cada una de las articulaciones de un robot de 4 grados de libertad con aplicación de control PID y PID Difuso, en la que puedan desempeñarse las funciones básicas del robot como es la integración de la Cinemática Directa y la Cinemática Inversa.

OBJETIVOS PARTICULARES

Sincronizar al robot de 4 grados de Libertad con el fin de asegurar la comunicación entre el componente maestro y el esclavo que controle el ángulo de giro de cada articulación.

La precisión de cada una de las articulaciones debe de tener un ± 1 grado de tolerancia.

La programación realizada en la PC debe de poder ser reprogramada en otros dispositivos electrónicos, sin contener bloques especializados que no puedan ser reutilizados en otros lenguajes de programación.

La programación utilizada en *LabVIEW* debe tener la flexibilidad de resolver procesos utilizados en el programa *Matlab*, y además debe de poder interactuar con diferentes programas en la adquisición de información para la aplicación de la visión artificial.

REQUISITOS OBLIGATORIOS

Los requisitos obligatorios que se buscan obtener son el Bajo Costo y la Precisión angular necesaria para el seguimiento de trayectorias. Una de las necesidades que necesita el robot es conocer la cinemática Directa, es decir que es necesario obtener la posición del efector final con respecto a su base. Una vez obtenida podemos utilizar la matriz de transformación homogénea para visualizar la posición en "x", "y" y "z".

También es útil estimar los torques máximos que puede soportar el robot. Para esto es necesario hacer un análisis estático en el cual involucre todas las cargas que interactúan con el peso del robot y de esta forma se puede obtener el torque que requieren los motores para poder soportar la estructura y el peso que va a cargar el efector final. Otra acción importante es conocer la cinemática inversa pues esta es esencial para casos de generación de trayectorias. Además, es importante porque cumple uno de los principios básicos en los movimientos de los robots, el cual se basa en moverse con respecto al marco de referencia de la base y encontrar los ángulos de cada articulación una vez que contamos con la posición deseada. Finalmente por medio de una sistema de control nosotros podremos posteriormente controlar un ángulo deseado con el ángulo real que obtenemos en la práctica, esencialmente para controlar el equilibrio y le permita llegar a sus objetivos

REQUISITOS DESEABLES

Los requisitos deseados es la adaptabilidad programacional para poder utilizar el control propuesto en diversos robots sin necesidad de realizar modificaciones que implicarán la pérdida de tiempo y el incremento en consumo de recursos programacionales.

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

JUSTIFICACIÓN: Dentro de la rama industrial los robots ocupan un papel irremplazable haciendo de forma cotidiana tareas que son difíciles de reproducir con calidad; sin embargo sigue existiendo la necesidad de un programador que a su vez pueda enseñarle diversas tareas al manipulador. Sin embargo, se debe de considerar que es necesario tener el equipo adecuado para realizar la tarea correspondiente, es decir muchas veces la realidad se apega a que se debe de trabajar con el software y con el equipo que especifica el proveedor, y en muchos casos se pierde el enfoque de saber si se pueden implementar otros tipos de instrumentos. Es por eso que se deben de buscar alternativas para comunicar un manipulador con su entorno; sin la necesidad de manejar directamente una aplicación en específico y plantear posibles soluciones



1 ESTADO DEL ARTE.



Los robots se encuentran presentes interactuando de forma cotidiana en las industrias, laboratorios y algunos de ellos se pueden encontrar desarrollando labores automotrices o alimentarias. Pero no debemos de olvidar que muchos otros tipos de robots, hacen más que un trabajo cotidiano en la tierra. Y un ejemplo es el Manipulador serial de 7 grados de libertad Canadarm2 véase fig. (1.1). Utilizado por la Estación Espacial Internacional (ISS).



Fig. 1.1 El Canadarm2 unido a la Estación Espacial Internacional (cortesía de la NASA).

Algunos otros como el "El Robonaut" (véase la fig. 1.2). Utilizan una base de modelos con enfoque en la percepción visual. Cada robot tiene un pequeño número de modelos 3-D para conocer los objetos que pueden ser combinados y almacenados con todo tipo de objetos vistos por el robot de la cámara a fin de permitir a estos realizar tareas como la apertura de un refrigerador y la sujeción de un desarmador. Hasta el momento, la capacidad de estos sistemas de visión ha demostrado de manera fiable la manipulación de un gran número de objetos. (Platt, et al., 2006).

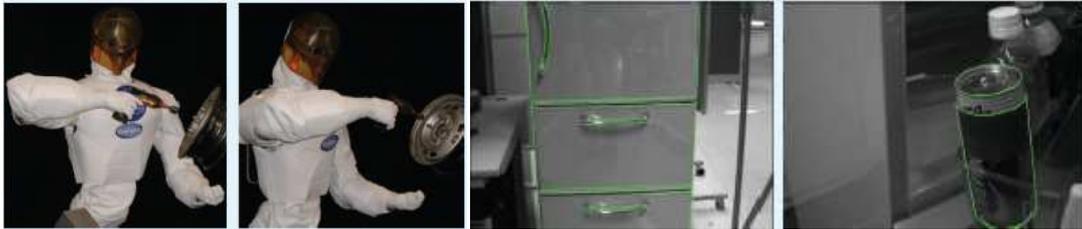


Fig. 1.2 Robot de la Nasa y sujeción de objetos.

Actualmente todos estos dispositivos y robots ejecutan diferentes funciones dependiendo del fin con que fueron diseñados y es necesario un control de trayectoria para que puedan posicionarse en la ubicación deseada.

Una de las ventajas del WAM "*whole arm manipulator*" es tener una baja fricción y la capacidad de determinar de forma rápida la localización de un objeto. Véase en la Fig. (1.3). como lo dicen sus siglas en Inglés (Whole Arm Manipulator) el WAM es un robot de 7 grados de Libertad, con la posibilidad de interactuar con un sistema de visión artificial. Cuando una bola de baseball es arrojada; el mismo sistema de visión calcula el centroide de la bola que está constantemente cambiando en un lapso de 1 ms. Mientras que el robot empieza a moverse por lo menos en 2ms. De una prueba realizada de 12 disparos el WAM no fallo ninguno.



Fig. 1.3. EL nuevo brazo WAM desarrollado con financiamiento de la NASA, El Departamento de Energía de los Estados Unidos y la Fundación Nacional de Ciencia en los Estados Unidos pero creado originalmente por Barrett Technology y MIT.

Un ejemplo con un enfoque similar al desarrollo de la tesis se puede observar en la fig. (1.4). En el control de trayectoria utilizado en este robot se controla su posición y velocidad, por lo que el objetivo de control es hacer que el error de estas variables tienda asintóticamente a cero.



Fig. 1.4. Robot Antropomórfico de 3 grados de libertad.

Otro trabajo realizado que se puede encontrar es “Seguimiento intemporal de trayectorias en la imagen basado en control visual dinámico”, véase fig. (1.5) el cual maneja un control visual dinámico que es una técnica que permite incrementar la velocidad de la tarea de control visual y a su vez la eliminación del lazo de control interno del robot permitiendo un procesamiento de la imagen a una frecuencia muy elevada. Esto con el fin de poder tomar fotos sin la necesidad de saturar la memoria del dispositivo de almacenamiento.

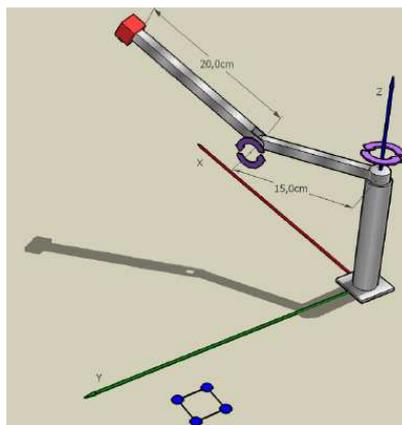


Fig. 1.5. Robot de 2 grados de libertad con pruebas de seguimiento

En algunos trabajos como el de Wilmer Sanz (2004), de la Universidad de Carbo se han dedicado a desarrollar ambientes virtuales por medio de herramientas nuevas las cuales permitan interfazar programas y pueden ser útiles para la simulación de objetos sin la necesidad de construirlos. Y para ser esto utilizaron utilidades como lo fueron *LabVIEW* y *Matlab*; para ser más preciso en este ultimo la utilidad del guía (*Gui*), para poder realizar botoneras. fig. (1.6). en la aplicación de la Cinemática Directa.

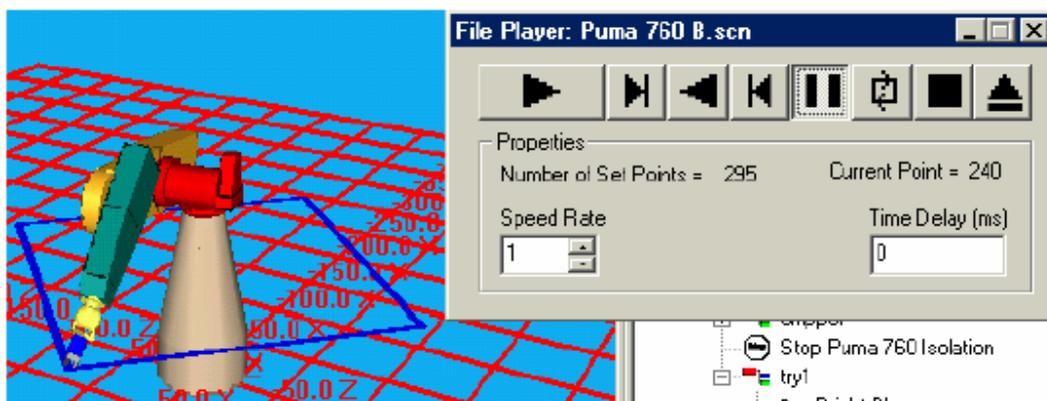


Fig. 1.6. Animación de Seguimiento de una trayectoria en Roboworks.



2 *MARCO TEÓRICO*



2.1 Desarrollo del Proyecto

El desarrollo de este proyecto implica la elaboración de una serie de trabajos los cuales se pueden seccionar en el diseño mecánico, el modelado matemático del brazo, el control y la simulación.

2.2 Diseño mecánico

En el diseño mecánico se busca generar una figura bidimensional que sirva como base para representar el modelo a utilizar e inclusive ser aprovechado para analizar de forma computacional, de esta manera se puede obtener datos más concisos de nuestro robot y un mejor resultado en la experimentación. Además se debe de considerar en el diseño mecánico el cálculo de los Torques de los motores del robot vía momentos de esta forma podemos deducir la capacidad que requerimos en cada motor para poderse mover. Lo que facilita su posicionamiento con respecto a su marco de referencia y su aérea de trabajo.

2.2.1 Modelado Matemático del Brazo.

En el modelado requerimos primero de las dimensiones del robot para poder determinar la cinemática directa, además se debe de considerar la Cinemática inversa la cual sirve a la adquisición de información en la generación de trayectorias así como el control pertinente para el posicionamiento por cada articulación.

2.3 Control del sistema

La etapa de control es fundamental en la cual cabe mencionar que es aquella que nos garantiza matemáticamente que cada una de las articulaciones va a situarse en la posición y en el tiempo deseado, Utilizando técnicas que permitan comparar cada uno de los ángulos deseados con los ángulos reales; como resultado del control el error será igual a la diferencia entre ellos y esta tiene que tender a ser cero.

2.3.1 Simulación y Experimentación

Una vez obtenido el modelo y control del robot se pueden realizar pruebas si efectivamente trabaja de forma óptima en cada una de las articulaciones. Para esto se realizó una interfaz la cual comunica al control con el robot, con el margen de error mínimo que se pueda obtener.



3 DISEÑO MECÁNICO Y MODELADO MATEMATICO DEL BRAZO



3.1 Diseño mecánico del prototipo

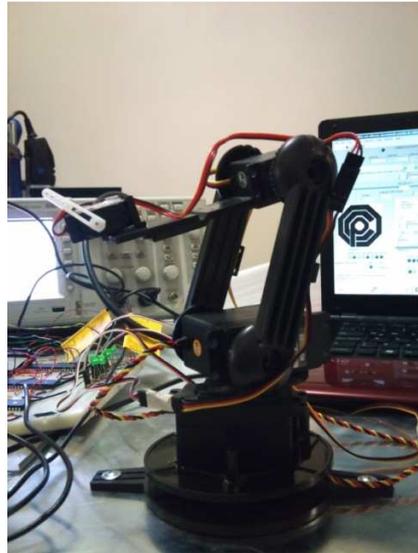


Fig. 3.1. Robot de cuatro grados de libertad.

Considerando la geometría en la que se realizó el robot de 4 grados de Libertad véase fig. (3.1), se puede mencionar que debido a su misma forma y a un peso muy ligero de parte de sus eslabones se realizó un análisis simbólico de lo que podría ser el estudio estático por medio de análisis de vigas. Por lo tanto el peso de los motores pueden ser considerados como cargas muertas. Según Cosgalla (2000) en su libro “Análisis de Vigas” una carga muerta es aquella donde las fuerzas que estas generan; siempre permanecen constantes y en la mayoría de las ocasiones son inamovibles. Dentro de este tipo de cargas se considera el peso propio de la estructura. Así que debido a esto se deben de tomar algunos conceptos mas, como el centroide el cual se define como el centro geométrico de una figura, por tanto no se tendrá mucho problema en considerar a nuestro sistema como simétrico. Y el peso en los motores a utilizar el cual se ejemplifico por una carga puntual repartida según las dimensiones de los eslabones. Debido a la practicidad que implica el proceso de manejar las ecuaciones por medio de variables se opto por hacerlo así con la oportunidad de usarlo posteriormente y poder realizar modificaciones. En el siguiente modelo. Se necesita determinar el torque máximo para poder mover el efector final. Por lo que el dimensionamiento está regido por la siguiente línea blanca véase fig. (3.2):

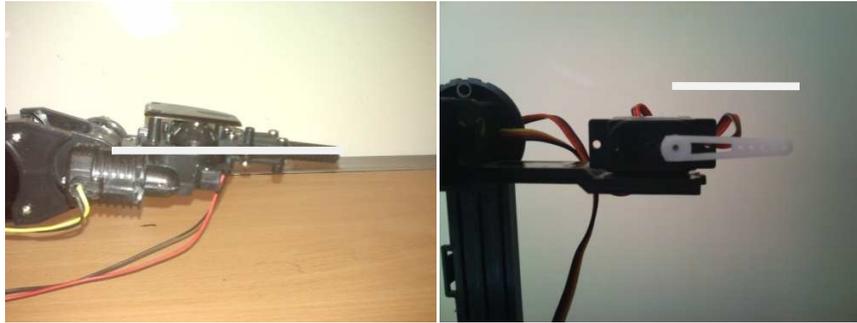


Fig. 3.2.Eslabones del Efector Final

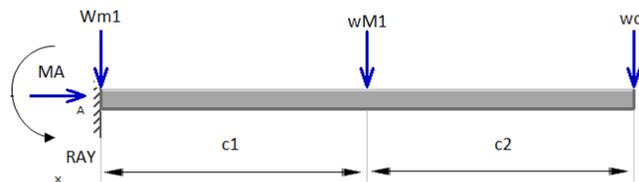


Fig. 3.3.Representación del efector final en forma de viga en voladizo.

Como se muestra en la Fig. (3.3), el peso del motor es representado por la variable "Wm", el peso del eslabón es representado por la variable "wM" y el peso de la carga es representada por "wq". Por lo el el tramo 1 es representado por:

tramo1

$$0 \leq x \leq c1$$

$$V_x = RAY - w_m1$$

$$M_x = -Ma + RAY(x) - w_m1(x)$$

Tramo2

$$c1 \leq x \leq c2$$

Donde las ecuaciones (3.1) y (3.2), representan a la cortante máxima y al Momento Máximo de la estructura...

Ecuación 3.1

$$V_x = RAY - wM1 - wm1$$

Ecuación 3.2

$$M_x = -MA + RAY(x) - wM1 (c2)$$

En el tramo de la fig. (3.4). es necesario analizar el torque que requiere el motor 2 para poder soportar el peso de la articulación rotacional pasada y el efector final.



Fig. 3.4. Motor2 y efector final.

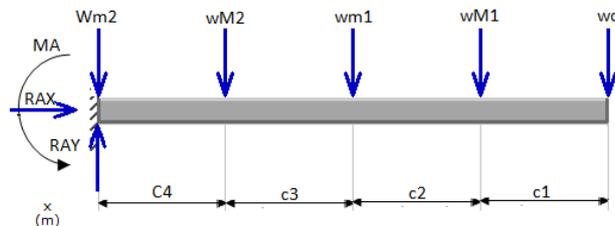


Fig. 3.5. Representación en Voladizo del Segundo motor más el efector final.

El peso del motor 2 es representado por la variable "Wm2" y el peso del eslabón 2 es representado por la variable "wM2", los valores Wm, wM y wq fueron mencionados con anterioridad véase fig. (3.5). Por lo que el tramo 2 es representado por:

$$0 \leq x \leq c4$$

$$V_x = RAY - w_m2$$

$$M_x = -MA - w_m2(x) + RAY$$

$$C4 \leq x \leq c3$$

$$V_x = RAY - w_m2 - wM2$$

$$M_x = -MA - w_m2(x) - wM2(c3) - w_m1 + RAY(x)$$

$$C3 \leq x \leq c2$$

$$V_x = RAY - w_m2 - wM2 - w_m1$$

$$M_x = -MA + RAY(x) - w_m2(x) - wM2(c3+c2) - w_m1(c2)$$

Las ecuaciones (3.3) y (3.4), representan a la cortante máxima y al Momento Máximo de la viga mostrada en la fig. (3.5).

$$C2 \leq x \leq c1$$

Ecuación 3.3

$$V_x = Ray - w_m2 - wM2 - w_m1 - wM1$$

Ecuación 3.4

$$M_x = RAY(x) - MA - w_m2(x) - wM2(c1+c2+c3) - w_m1(c1+c2) - wM1(c1)$$

En este paso véase fig. (3.6). se aplico el mismo procedimiento que en los casos anteriores. Para seguir determinando el torque necesario que necesita cada motor, y además un punto importante a mencionar es que nuestro objetivo es obtener la cortante máxima y el momento máximo.



Fig. 3.6. Dos fotografías de análisis de motores mas efector final.

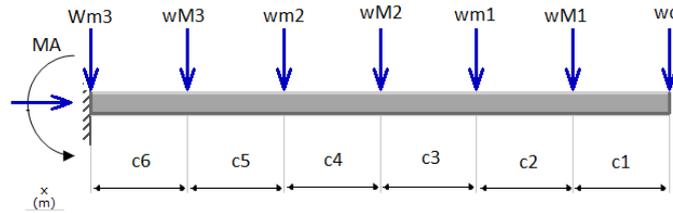


Fig. 3.7. Representación del análisis de los motores más el efector final.

El peso del motor 3 es representado por la variable "Wm3" y el peso del eslabón 3 es representado por la variable "wM3", véase fig. (3.7). El tramo 3 es representado por:

$$0 \leq x \leq c6$$

$$V_x = RAY - w_m3$$

$$M_x = -MA + RAY(x) - w_m3(x)$$

$$c6 \leq x \leq c5$$

$$V_x = RAY - w_m3 - w_M3$$

$$M_x = -MA + RAY(x) - w_m3(x) - w_M3(c5)$$

$$c5 \leq x \leq c4$$

$$V_x = RAY - w_m3 - w_M3 - w_m2$$

$$M_x = -MA + RAY(x) - w_m3(x) - w_M3(c5 + c4)$$

$$c4 \leq x \leq c3$$

$$V_x = RAY - w_m3 - w_M3 - w_m2 - w_M2$$

$$M_x = -MA + RAY(x) - w_m3(x) - w_M3(c5 + c4 + c3) - w_m2(c4 + c3)$$

$$c3 \leq x \leq c2$$

$$V_x = RAY - w_m3 - w_M3 - w_m2 - w_M2 - w_m1$$

$$M_x = -MA + RAY(x) - w_m3(x) - w_M3(c5 + c4 + c3 + c2) - w_m2(c4 + c3 + c2) - w_M2(c3 + c2) - w_m1(c2)$$

Las ecuaciones (3.5) y (3.6), representan a la cortante máxima y al Momento Máximo de la estructura mostrada en la fig. (3.7).

$$C2 \leq x \leq c1$$

Ecuación 3.5

$$V_x = RAY - w_m3 - wM3 - w_m2 - wM2 - w_m1 - wM1$$

Ecuación 3.6

$$M_x = -MA + RAY(x) - w_m3(x) - wM3(c5 + c4 + c3 + c2) - w_m2(c4 + c3 + c2 + c1) - wM2(c3 + c2 + c1) - w_m1(c2 + c1)$$



Fig. 3.8. Fotografía completa de la representación del Robot.

El modelo a realizar véase fig.(3.9), determina la distancia máxima existente en este punto como se puede observar en la ecuaciones, sin embargo de acuerdo al modelo realizado con respecto al momento de flexión, el sistema se puede reducir si consideramos un sistema en el cual las distancias c 's son iguales.

El peso del motor 4 es representado por la variable " W_m4 " y el peso del eslabón 4 es representado por la variable " $wM4$ ". El tramo 3 es representado por el eslabón que contiene al eslabón 4 y a los eslabones anteriores véase fig. (3,8):

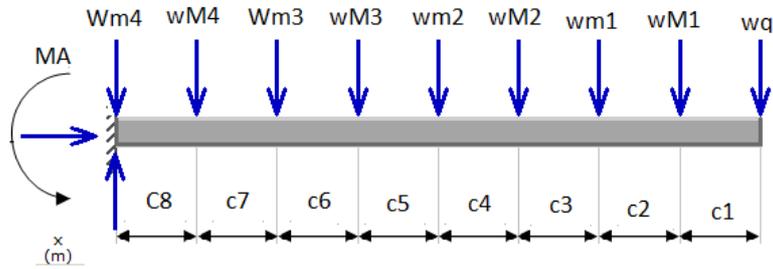


Fig. 3.9. Representación de los 4 motores en voladizo.

$$0 \leq x \leq c_8$$

$$V_x = RAY - w_m4$$

$$M_x = -MA + RAY(X) - w_m4(x)$$

$$c_8 \leq x \leq c_7$$

$$V_x = RAY - w_m4 - w_M4$$

$$M_x = -MA + RAY(X) - w_m4(x) - w_M4(c_7)$$

$$c_7 \leq x \leq c_6$$

$$V_x = RAY - w_m4 - w_M4 - w_m3$$

$$M_x = -MA + RAY(X) - w_m4(x) - w_M4(c_7 + c_6) - w_m3(c_6)$$

$$c_6 \leq x \leq c_5$$

$$V_x = RAY - w_m4 - w_M4 - w_m3 - w_M3$$

$$M_x = -MA + RAY(X) - w_m4(x) - w_M4(c_7 + c_6 + c_5) - w_m3(c_6 + c_5) - w_M3(c_5)$$

$$c_5 \leq x \leq c_4$$

$$V_x = RAY - w_m4 - w_M4 - w_m3 - w_M3 - w_m2$$

$$M_x = M_x = -MA + RAY(X) - w_m4(x) - w_M4(c_7 + c_6 + c_5 + c_4) - w_m3(c_6 + c_5 + c_4) - w_M3(c_5 + c_4) - w_m2(c_4)$$

$$c_4 \leq x \leq c_3$$

$$V_x = RAY - w_m4 - w_M4 - w_m3 - w_M3 - w_m2 - w_M2$$

$$M_x = -MA + RAY(X) - w_m4(x) - w_M4(c_7 + c_6 + c_5 + c_4 + c_3) - w_m3(c_6 + c_5 + c_4 + c_3) - w_M3(c_5 + c_4 + c_3) - w_m2(c_4 + c_3) - w_M2(c_3)$$

$$C3 \leq x \leq c2$$

$$V_x = RAY - wM4 - wM4 - wM3 - wM3 - wM2 - wM2 - wM1$$

$$M_x = -MA + RAY(X) - wM4(x) - wM4(c7 + c6 + c5 + c4 + c3 + c2) - wM3(c6 + c5 + c4 + c3 + c2) - wM3(c5 + c4 + c3 + c2) - wM2(c4 + c3 + c2) - wM2(c3 + c2) - wM1(c2)$$

$$C2 \leq x \leq c1$$

Donde las Ecuaciones (3.7) y (3.8), representan a la cortante máxima y al Momento Máximo de la estructura posicionada en forma horizontal véase fig. (3.9).

Ecuación 3.7

$$V_x = RAY - wM4 - wM4 - wM3 - wM3 - wM2 - wM2 - wM1 - wM1$$

Ecuación 3.8

$$M_x = -MA + RAY(X) - wM4(x) - wM4(c7 + c6 + c5 + c4 + c3 + c2 + c1) - wM3(c6 + c5 + c4 + c3 + c2 + c1) - wM3(c5 + c4 + c3 + c2 + c1) - wM2(c4 + c3 + c2 + c1) - wM2(c3 + c2 + c1) - wM1(c2 + c1) - wM1(c1)$$

Y en base a la gráfica se puede determinar donde existe el cortante máximo para este caso:

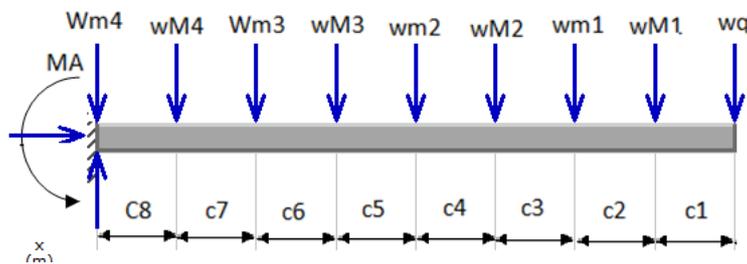


Fig. 3.10. Representación completa de la viga en voladizo.

También podemos considerar que para una viga en voladizo de la forma de la figura (3.10), el análisis de las reacciones cortantes interactúan de la forma de la figura (3.12), por esto podemos determinar:

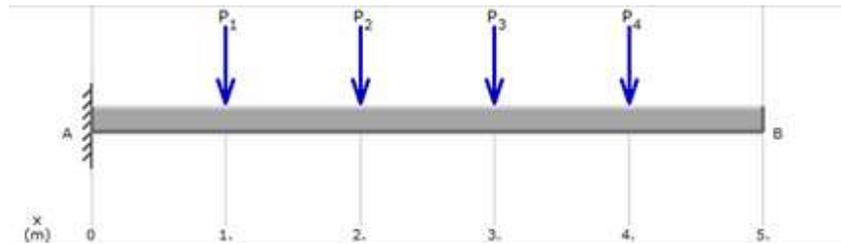


Fig. 3.11. Viga en Voladizo.

Siempre y cuando las cargas sean las mismas como lo es el caso, podemos observar que el esfuerzo cortante máximo se encuentra situado entre la base y la segunda articulación:

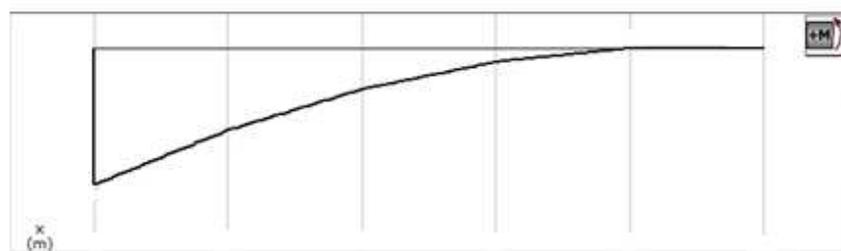


Fig. 3.12. Representación gráfica de los esfuerzos cortantes y momentos flexionantes en una viga en cantiliver cuando se le aplica una carga sobre una determinada área.

En base a esto se define:

$$\tau_{\max 1} = Tc/J$$

$$\tau_{\max 2} = Tc/J$$

$$\tau_{\max 3} = Tc/J$$

$$\tau_{\max 4, 5} = Tc/J$$

Siendo el caso:

C es el radio de la flecha o eje, J el momento polar de Inercia, y T el torque que queremos calcular.

$$T1 = (\tau_{\max 1} * J)/C$$

$$T2 = (\tau_{\max 2} * J)/C$$

$$T3 = (\tau_{\max 3} * J)/C$$

$$T45 = (\tau_{\max 4,5} * J)$$

3.2 Cinemática del prototipo de 4 Grados de Libertad

INTRODUCCIÓN

Como lo menciona el autor Craig (2006), en su Libro “Robótica” la Cinemática es la ciencia del movimiento que trata el tema sin las fuerzas que lo ocasionan. Dentro de esta ciencia se estudia la posición, la velocidad, la aceleración y todas las derivadas de alto orden de las variables de posición. (Con respecto al tiempo u otra variable). En consecuencia, el estudio de la cinemática de manipuladores se refiere a todas las propiedades geométricas basadas en el tiempo del movimiento. Las relaciones entre estos movimientos, las fuerzas y momentos de torsión que los ocasionan constituyen el problema de la Dinámica.

El objeto de los procedimientos siguientes se basan en el principio básico para poder llegar a la ecuación que controla al ángulo deseado véase fig. (3.13). Y primero necesitamos determinar las características físicas del robot, es decir para llegar a esta etapa, primero debemos de formular un sistema que nos ayude a determinar los ángulos y distancias utilizadas en el manipulador para poder conocer su posicionamiento véase fig. (3.14).

Otro concepto importante que maneja FU (1989), acerca del algebra vectorial y matricial es que se utiliza para desarrollar un método generalizado y sistemático para describir y representar la localización de los elementos de un brazo respecto a un sistema de referencia fijo” es decir que lo que trata de explicar es que es la representación de los ángulos de un robot y su posición con respecto a su base o eje de referencia, el cual es representado por la última columna de la matriz de transformación homogénea”. Además como lo marca FU (1989). Un manipulador mecánico consiste en una secuencia de cuerpos rígidos, llamados elementos, conectados mediante articulaciones prismáticas o de revolución. Y cada par articulación-elemento constituye un grado de libertad.

3.2.1 Ecuaciones Cinemáticas para los manipuladores

La matriz Homogénea 0T_1 que especifica la localización del sistema de coordenadas i -ésimo con respecto al sistema de coordenadas de la base es el producto en cadenas de matrices de transformación de coordenadas sucesivas ${}^{i-1}A_i$ y se expresa como:

Ecuación 3.9

$${}^0T_i = {}^0A_1 * {}^1A_2 \dots {}^{i-1}A_i = \prod_{j=1}^i {}^{j-1}A_j \quad \text{Para } i=1,2,\dots,n.$$

$$= \begin{bmatrix} X_i & y_i & z_i & p_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0R_i & {}^0P_i \\ 0 & 1 \end{bmatrix}$$

Ecuación 3.10

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos a_i * \text{sen} \theta_i & \text{sen} a_i * \text{sen} \theta_i & a_i * \cos \theta_i \\ \text{sen} \theta_i & \cos a_i * \cos \theta_i & -\text{sen} a_i \cos \theta_i & a_i * \text{sen} \theta_i \\ 0 & \text{sen} a_i & \cos a_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

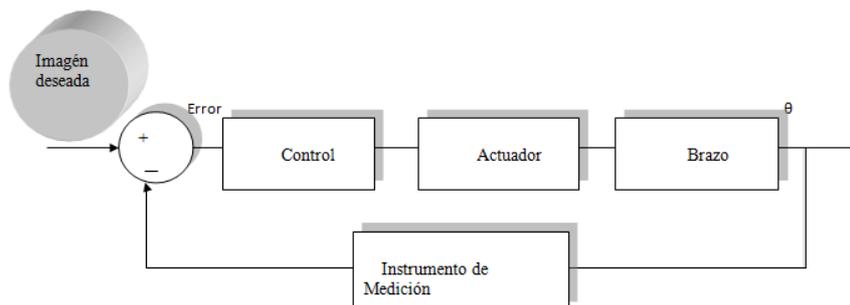


Fig. 3.13. Diagrama del Control Deseado.

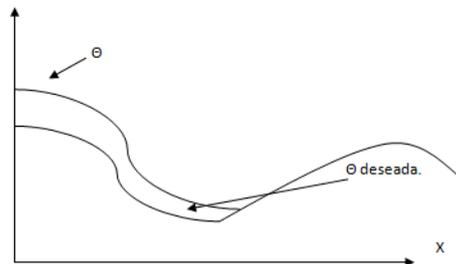


Fig. 3.14. Comparativa entre el ángulo deseado y nuestro ángulo real.

3.2.2 Cinemática Directa del Prototipo

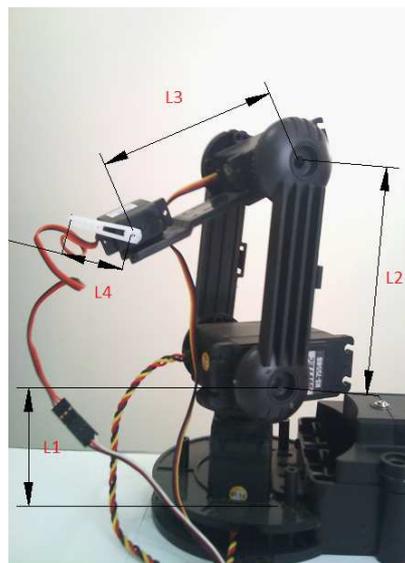


Fig. 3.15. Dimensiones reales del robot de 4 grados de Libertad utilizadas en Dh.

Lo primero que se necesita realizar para determinar las coordenadas del robot es empezar con los parámetros de Denavith Hartemberg, de esta forma se obtiene un robot con determinados ángulos y eslabones en forma de variables; así no hay que reajustar el sistema en caso de cambiar las dimensiones véase fig. (3.15).

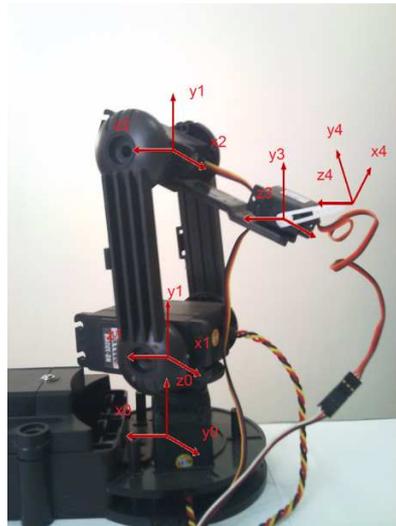


Fig. 3.16. Representación del robot por medio de marcos de referencia en el cual se maneja distancia cero entre el primer marco y el segundo.

En el caso particular de la tabla los valores fueron obtenidos en base a la colocación de los marcos de referencia en el robot véase fig. (3.16). Los valores de Θ , α , β , y Υ , son diferentes a Theta y Alpha debido a que estos últimos representan a la convención de los parámetros de Denavith Hartemberg mientras Θ , α , β , y Υ son los ángulos de cada una de las articulaciones.

i	Ai	di	Alpha i	Theta i
1	0	0	90	Θ^*
2	l2	0	0	α^*
3	l3	0	0	β^*
4	l4	0	0	Υ^*

Tabla 3.1. Representación de los parámetros de Denavith Hartemberg.

Una vez obtenida la tabla se requiere pasar a la matriz de transformación Homogénea, esta matriz nos sirve para localizar coordenadas, solo necesitaríamos modificar los ángulos de cada articulación para poder determinar su posición final.

Al tratar de resolver la matriz se procedió por cambiar variables de cada ángulo de esta forma podemos obtener una solución por algún método computacional de forma rápida:

Ecuación 3.11

$${}^0_1T = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ \sin(\theta) & 0 & -\cos(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.12

$${}^1_2T = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & x * \cos(\alpha) \\ \sin(\alpha) & \cos(\alpha) & 0 & x * \sin(\alpha) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.13

$${}^2_3T = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & y * \cos(\beta) \\ \sin(\beta) & \cos(\beta) & 0 & y * \sin(\beta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.14

$${}^3_4T = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & z * \cos(\gamma) \\ \sin(\gamma) & \cos(\gamma) & 0 & z * \sin(\gamma) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Desarrollo de los parámetros de la matriz de Transformación $T=A1A2A3A4$ por medio de la ecuación (3.15):

Ecuación 3.15

T=

$$\begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ \sin(\theta) & 0 & -\cos(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & x * \cos(\alpha) \\ \sin(\alpha) & \cos(\alpha) & 0 & x * \sin(\alpha) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & y * \cos(\beta) \\ \sin(\beta) & \cos(\beta) & 0 & y * \sin(\beta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & z * \cos(\gamma) \\ \sin(\gamma) & \cos(\gamma) & 0 & z * \sin(\gamma) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.16

$$T = \begin{bmatrix} xu & xv & xw & Px \\ yu & yv & yw & Py \\ zu & zv & zw & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde la ecuación (3.16) se puede expresar de la siguiente forma:

$$Xu = -(\cos\theta\sin\alpha) (\cos\beta\sin\gamma + \cos\gamma\sin\beta) + (\cos\theta\cos\alpha)(\cos\beta\cos\gamma - \sin\beta\sin\gamma)$$

$$Yu = -(\sin\theta\sin\alpha) (\cos\beta\sin\gamma + \cos\gamma\sin\beta) + (\cos\alpha\sin\theta)(\cos\beta\cos\gamma - \sin\beta\sin\gamma)$$

$$Zu = (\cos\alpha) (\cos\beta\sin\gamma + \cos\gamma\sin\beta) + (\sin\alpha)(\cos\beta\cos\gamma - \sin\beta\sin\gamma)$$

$$Xv = -(\cos\theta\sin\alpha)(\cos\beta\cos\gamma - \sin\beta\sin\gamma) + (\cos\theta\cos\alpha)(-\cos\beta\sin\gamma - \cos\gamma\sin\beta)$$

$$Yv = -(\sin\theta\sin\alpha)(\cos\beta\cos\gamma - \sin\beta\sin\gamma) + (\cos\alpha\sin\theta)(-\cos\beta\sin\gamma - \cos\gamma\sin\beta)$$

$$Zv = (\cos\alpha)(\cos\beta\cos\gamma - \sin\beta\sin\gamma) + (\sin\alpha)(-\cos\beta\sin\gamma - \cos\gamma\sin\beta)$$

$$Xw = \sin\theta$$

$$Y_w = -\cos\theta$$

$$Z_w = 0$$

$$P_x = x \cos\theta \cos\alpha - (\cos\theta \sin\alpha)(y \sin\beta + z \cos\beta \sin\gamma + z \cos\gamma \sin\beta) + (\cos\theta \cos\alpha)(y \cos\beta + z \cos\beta \cos\gamma - z \sin\beta \sin\gamma)$$

$$P_y = x \cos\alpha \sin\theta - (\sin\theta \sin\alpha)(y \sin\beta + z \cos\beta \sin\gamma + z \cos\gamma \sin\beta) + (\cos\alpha \sin\theta)(y \cos\beta + z \cos\beta \cos\gamma - z \sin\beta \sin\gamma)$$

$$P_z = x \sin\alpha + (\cos\alpha)(y \sin\beta + z \cos\beta \sin\gamma + z \cos\gamma \sin\beta) + (\sin\alpha)(y \cos\beta + z \cos\beta \cos\gamma - z \sin\beta \sin\gamma)$$

El resultado que nos muestra al final denota una parte rotacional que son las primeras tres columnas. Mientras que la cuarta representa la posición en P_x , P_y , P_z . De esta forma podemos obtener la representación del efector final con respecto a la base.

3.2.3 Cinemática inversa del prototipo

Para la obtención de la cinemática inversa se cuenta con las Transformaciones individuales de cada vínculo. De las cuales se puede obtener además su matriz inversa, Se utilizan ${}^0_1T^{-1}$ siendo la matriz de transformación individual inversa de 0 a 1, ${}^0_2T^{-1}$ la matriz de transformación inversa de cero a dos, ${}^0_3T^{-1}$ la de cero a 3 , y la multiplicación de las matrices de transformación individual para 2_4T y 3_4T .

La matriz de transformación individual inversa de 0 a 1 (**3.17**).

$${}^0_1T^{-1}$$

La matriz de transformación individual inversa de 0 a 2 (3.18)

$${}^0_2T^{-1}$$

La matriz de transformación individual inversa de 0 a 3 (3.19)

$${}^0_3T^{-1}$$

La matriz de transformación individual de 2 a 4 (3.20)

2_4T

La matriz de transformación individual de 3 a 4 (3.21)

3_4T

Se realiza la igualdad geométrica entre las transformaciones para obtener la solución correspondiente de los ángulos Θ y Y de los parámetros de Denavith Hartenberg véase la tabla (3.1). Primero se obtiene la primera ecuación por medio de la matriz (3.17) con multiplicación de la matriz de 0 a 4.

Ecuación 3.22

$${}^0_1T^{-1} * {}^0_4T = {}^1_4T$$

La segunda se obtiene multiplicando a la matriz (3.18), con multiplicación de la matriz de 0 a 4.

Ecuación 3.23

$${}^0_2T^{-1} * {}^0_4T = {}^2_4T$$

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

La tercera se obtiene multiplicando a la figura (3.19) con multiplicación de la matriz de 0 a 4, el resultado se expresa como la matriz. (3.21).

Ecuación 3.24

$${}^0_3T^{-1} * {}^0_4T = {}^3_4T$$

Estas son las ultimas 3 igualaciones finales:

Ecuación 3.25

$$\begin{bmatrix} \cos(\alpha + \beta + \gamma) & -\sin(\alpha + \beta + \gamma) & 0 & PX\cos\theta + PY\sin\theta \\ \sin(\alpha + \beta + \gamma) & \cos(\alpha + \beta + \gamma) & 0 & PZ \\ 0 & 0 & 1 & -PY\cos\theta + PX\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha + \beta + \gamma) & -\sin(\alpha + \beta + \gamma) & 0 & x\cos\alpha + y\cos(\alpha + \beta) + z\cos(\alpha + \beta + \gamma) \\ \sin(\alpha + \beta + \gamma) & \cos(\alpha + \beta + \gamma) & 0 & x\sin\alpha + y\sin(\alpha + \beta) + z\sin(\alpha + \beta + \gamma) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.26

$$\begin{bmatrix} \cos(\beta + \gamma) & -\sin(\beta + \gamma) & 0 & -x + PZ\sin\alpha + \frac{1}{2}PX\cos(\theta + \alpha) + \frac{1}{2}PY\sin(\theta + \alpha) + \frac{1}{2}PX\cos(\theta - \alpha) + \frac{1}{2}PY\sin(\theta - \alpha) \\ \sin(\beta + \gamma) & \cos(\beta + \gamma) & 0 & PZ\cos\alpha + \frac{1}{2}PY\cos(\theta + \alpha) - \frac{1}{2}PX\sin(\theta + \alpha) - \frac{1}{2}PY\cos(\theta - \alpha) + \frac{1}{2}PX\sin(\theta - \alpha) \\ 0 & 0 & 1 & -PY\cos\theta + PX\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \cos\beta\cos\gamma - \sin\beta\sin\gamma & -\cos\beta\sin\gamma - \cos\gamma\sin\beta & 0 & y\cos\beta + z\cos\beta\cos\gamma - z\sin\beta\sin\gamma \\ \cos\beta\sin\gamma + \cos\gamma\sin\beta & \cos\beta\cos\gamma - \sin\beta\sin\gamma & 0 & y\sin\beta + z\cos\beta\sin\gamma + z\cos\gamma\sin\beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.27

$$\begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & -y - x\cos\beta + PZ\sin(\alpha + \beta) + \frac{1}{2}PX\cos(\theta + \alpha + \beta) + \frac{1}{2}PY\sin(\theta + \alpha + \beta) + \frac{1}{2}PX\cos(\theta - \alpha - \beta) + \frac{1}{2}PY\sin(\theta - \alpha - \beta) \\ \sin\gamma & \cos\gamma & 0 & x\sin\beta + PZ\cos(\alpha + \beta) + \frac{1}{2}PY\cos(\theta + \alpha + \beta) - \frac{1}{2}PX\sin(\theta + \alpha + \beta) - \frac{1}{2}PY\cos(\theta - \alpha - \beta) + \frac{1}{2}PX\sin(\theta - \alpha - \beta) \\ 0 & 0 & 1 & -PY\cos\theta + PX\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & z + \cos(\gamma) \\ \sin(\gamma) & \cos(\gamma) & 0 & z + \sin(\gamma) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

De esto se puede utilizar el mismo método geométrico y algebraico para poder solucionar el robot de la siguiente forma:

Ecuación 3.28

$$-Py \cos\theta + Px\sin\theta = 0$$

Ecuación 3.29

$$Px\sin\theta - Py \cos\theta = 0$$

Ecuación 3.30

$$P \cos\theta = Px$$

Ecuación 3.31

$$P \sin\theta = Py$$

Ecuación 3.32

$$P = \sqrt{Px^2 + Py^2}$$

Ecuación 3.33

$$\theta = \text{Atan2}(Py, Px)$$

Ecuación 3.34

$$\theta = \text{Atan2}\left(\frac{Py}{Px}\right)$$

Ecuación 3.35

$$-Py * C1 + Px * S1 = 0$$

O sea: 360 grados de $\text{Atan}(Py, Px)$ esto debido a que el Atan solo va de 90 a 90.

$$-(PS1) C1 + (PC1) S1 = 0$$

Ecuación 3.36

$$P[S\phi C1 - C\phi S1]_o = 0$$

Ecuación 3.37

$$P[sen(\phi - \theta_1)] = 0$$

Ecuación 3.38

$$[sen(\phi - \theta_1)] = 0$$

Ecuación 3.39

$$[cos(\phi - \theta_1)] = \pm\sqrt{1}$$

Ecuación 3.40

$$[(\phi - \theta_1)] = \text{Atan2}(0, \pm\sqrt{1})$$

Ecuación 3.41

$$[(\theta_1)] = \phi - \text{Atan2}(0, \pm\sqrt{1})$$

De esto se concluye que:

$$[(\theta_1)] = \text{Atan2}(Py, Px) \text{ Solución del primer ángulo.}$$

De la Ecuación general se puede considerar que:

Ecuación 3.42

$$\text{Tan}\gamma = \frac{z \sin \gamma}{z \cos \gamma}$$

Ó en 360 grados:

Ecuación 3.43

$$\gamma = \arctan 2 \frac{z \sin \gamma}{z \cos \gamma}$$

Ahora como tal se pueden encontrar por medio de las igualaciones matriciales de las ecuaciones (3.25) a (3.27) la solución tanto del primer ángulo como el del cuarto.

Ecuación 3.44

$$\begin{aligned} Z * \cos \gamma = & -y - x \cos \beta - Pz * \sin(\alpha + \beta) + \frac{1}{2} Px * \cos(\theta + \alpha + \beta) \\ & + \frac{1}{2} Pysin(\theta + \alpha + \beta) + \frac{1}{2} Px \cos(\theta - \alpha - \beta) + \frac{1}{2} Py \cos(\theta - \alpha - \beta) \end{aligned}$$

Ecuación 3.45

$$\begin{aligned} Z * \sin \gamma = & x \sin \beta - Pz * \cos(\alpha + \beta) + \frac{1}{2} Py * \cos(\theta + \alpha + \beta) - \frac{1}{2} Pysin(\theta + \alpha + \beta) \\ & - \frac{1}{2} Px \cos(\theta - \alpha - \beta) + \frac{1}{2} Px \sin(\theta - \alpha - \beta) \end{aligned}$$

Por lo tanto la ecuación siguiente representa el cuarto ángulo del robot:

Ecuación 3.46

$$\gamma = \arctan 2 \frac{z \sin \gamma}{z \cos \gamma}$$

En base a los valores de la tabla se pueden obtener los ángulos beta y gama del robot:

I	Ai	Di	Alpha i	θ_i^*
1	0	0	90°	Θ^*
2	L2	0	0	α^*
3	L3	0	0	β^*
4	L4	0	0	γ^*

Tabla 3.2. Representación de los parámetros de Denavith Hartenberg para Robot articulado de 4 grados de Libertad.

Donde el asterisco significa que un ángulo puede variar de acuerdo a la circunstancia. Si L_4 es muy pequeño de forma que sea igual a cero entonces se puede evitar cualquier suceso inesperado para usar el método geométrico. Si los eslabones 2 y 3 valen lo mismo se puede interpretar codo arriba y abajo sin complicaciones debido a que su simetría no altera las distancias.

Al mencionar que el arreglo del robot no cuenta con 14 se tendría que ajustar; en caso de que se deseara implementar con alguno que si lo utilizara. En este caso permitimos que el marco se simplifique Geométricamente véase fig. (3,17).

De acuerdo a la convención de ángulos los resultados obtenidos se pueden utilizar con un método que simplifica la solución para el cuarto ángulo. Como se puede ver en el libro de Robótica por J. f. Craig (2006):

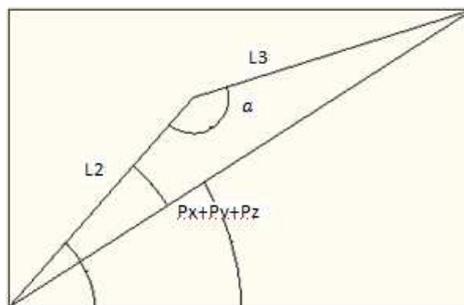


Fig. 3.17. Imagen bidimensional para la obtención del primero y segundo eslabón físico .

Ecuación 3.47

$$A^2 = B^2 + C^2 - 2BC \cos a$$

$$L2=B$$

$$L2^2=B^2$$

$$L3=c$$

$$A = P_x + P_y + P_z$$

$$L_3^2 = c^2$$

Ecuación 3.48

$$\cos a = \frac{B^2 + C^2 - A^2}{BC}$$

Ecuación 3.49

$$\cos \theta_3 = \frac{P_x^2 + P_y^2 + P_z^2 - l_2^2 - l_3^2}{2l_2 * l_3}$$

O visto desde otra forma:

Ecuación 3.50

$$\cos \theta_3 = \frac{[P_x^2 + P_y^2] + P_z^2 - [l_2^2 + l_3^2]}{2l_2 * l_3}$$

Ahora se puede observar de donde viene la solución trigonométrica ya que el agrupamiento es: $[P_x^2 + P_y^2]$, véase fig.(3,18).

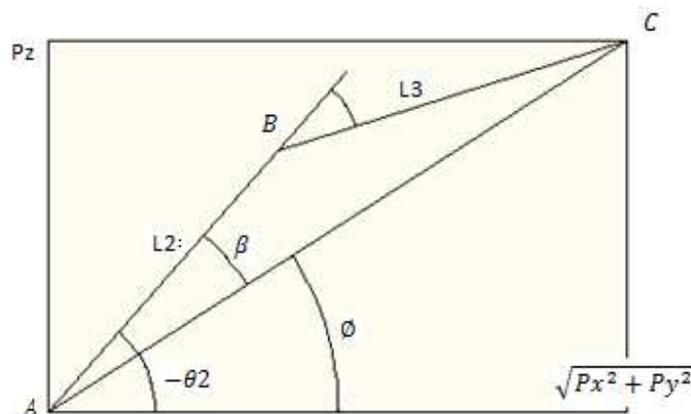


Fig. 3.18. Plano Geométrico para robot articulado de 4 grados de libertad.

Ecuación 3.51

$$B^2 = A^2 + C^2 - 2AC \cos \beta$$

Ecuación 3.52

$$\cos \beta = \frac{A^2 + C^2 - B^2}{2AC}$$

Por esto se puede concluir que:

Ecuación 3.53

$$\theta_3 = \text{Atan2} \left(\sqrt{1 - \cos^2 \theta_3}, \cos \theta_3 \right)$$

Ecuación 3.54

$$\theta_2 = -\text{Atan2} \left(P_z, \sqrt{P_x^2 + P_y^2} \right) - \text{Atan2} (l_3 \sin \theta_3, l_2 + l_3 \cos \theta_3)$$

Donde se considera que $\theta_2 = \alpha$ y $\theta_3 = \beta$, en el sistema original no el plano geométrico.

Dado que este en el plano $\theta_2 = -\phi - \beta$ por lo tanto para obtener a θ_2 se debió obtener el valor de ϕ que ya lo teníamos y el de β que se puede sacar de la forma:

Ecuación 3.55

$$\cos \beta = \frac{A^2 + C^2 - B^2}{2AC}$$

3.3 Cinemática de un Robot Industrial de 6 grados de Libertad

Introducción

Como lo comenta J. Craig (2006), en el libro de "Robótica", El problema de resolver las ecuaciones de un manipulador es no lineal. Esto se refiere a que dado una matriz de transformación Homogénea. Tratamos de encontrar los ángulos en cada uno de las articulaciones. De otra forma esto quiere decir que cuando se hizo la cinemática directa en el caso anterior se buscaba obtener la posición final del último efector (dispositivo final de sujeción o "*gripper*" etc....) con respecto a su base. La cual sirve cuando queremos mover cada articulación en un robot hasta llegar a un punto deseado; esto es comúnmente utilizado como experiencia personal, para posicionar de forma empírica a un robot cerca de su objetivo y es una de las formas actualmente utilizadas para la programación y utilización de los robots. Otro concepto muy útil es la existencia de Soluciones, pues esta depende directamente del espacio de trabajo. Es decir de las distancias en las cuales el robot puede trabajar sin problemas de singularidades (posiciones en las cuales pueda tender el robot a hacer movimientos indeseables. Estas singularidades se basan en algunos casos a propiedades del método algebraico o algún otro método en el cual puede ser que un ángulo de un eslabón dependa de otro antes o después cuando este se aproxima a cero. Así como el caso en el que puede depender de la programación de generación de trayectorias, (hablando de métodos numéricos), pues este debe de encontrar una solución para llegar a su trayectoria). Aunque en este caso se hable con algunas experiencias reflejadas en la práctica es importante que refleje total respeto hacia estos temas pues cada caso tiene algunas cuestiones las cuales se pudieron haber omitido, y sean de gran trascendencia. Sin expandirse más en el tema el autor J. Craig (2006), define al espacio de trabajo alcanzable el cual es el volumen de espacio que puede alcanzar el robot en por lo menos una orientación. Evidentemente, el espacio de trabajo diestro (codo arriba codo abajo) es un subconjunto del trabajo alcanzable. Algo muy interesante que comenta este autor es que dado θ_{11} y θ_{12} que representan a los eslabones de un manipulador común rotacional, (estos son después de la primera articulación). Si θ_{11} es diferente de θ_{12} entonces no hay un espacio de trabajo diestro, y esto es muy importante para poder trabajar con un manipulador.

Solución algebraica

Un método importante para la resolución de la cinemática Inversa ha sido el método algebraico, este se basa en el principio de utilizar las matrices de transformación homogénea y multiplicar esta por algunas inversas que forman parte del mismo. Anteriormente se buscó una solución para el caso del robot de 4 grados de libertad. Una cosa muy importante que cabe resaltar es que el método a utilizar va a variar dependiendo de las mismas necesidades que tiene el programador para que el robot pueda realizar una tarea de forma eficiente.

Considerando esto se pueden utilizar diferentes técnicas para llegar a la solución; pero no todas ellas necesariamente se van a poder utilizar, por eso es importante conocer el mayor número de técnicas en la búsqueda de las cinemáticas. Como tal entrando en tema la Cinemática Inversa es aquella en la cual nosotros tenemos la posición final a la que queremos que llegue el robot. Ahora a nosotros nos interesa llegar a esa coordenada como sea. Por medio de esto podemos pensar que el robot puede llegar a ese punto por diferentes trayectorias.

A final de cuenta no queremos que esto sea un caso tan complejo. La cinemática inversa es aquella en la cual de esa posición nosotros podemos obtener las posibles soluciones de cada ángulo de los eslabones por la cual va a llegar ese punto, por lo tanto ahí dependerá de la astucia de cada programador y el tipo de solución propuesta. Algunos expertos en el campo hablan de que la forma más útil para poder resolver la cinemática inversa aplicada a la generación de trayectorias es buscar aquellas distancias la cuales sean más cercanas a su posición pasada.

Solución Geométrica

Como lo menciona Craig (2006), en su libro de Robótica la solución Geométrica es un método para encontrar la solución de un manipulador, y para esto se trata de descomponer la geometría espacial del brazo en varios problemas de geometría plana.

3.3.1 Cinemática directa del Robot industrial

Se obtienen cada una de las transformadas individuales de cada vínculo, representadas por medio de la ecuación (3.56) a la ecuación (3.61), como en el caso del robot de cuatro grados de libertad. Ahora para un Robot de 6 grados de Libertad

Ecuación 3.56

$${}^0A_1 = \begin{bmatrix} \cos \theta_1 & 0 & \text{sen}\theta_1 & a_1 \cos \theta_1 \\ \text{sen} \theta_1 & 0 & -\cos \theta_1 & a_1 \text{sen} \theta_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.57

$${}^1A_2 = \begin{bmatrix} \cos \theta_2 & -\text{sen}\theta_2 & 0 & a_2 * \cos \theta_2 \\ \text{sen} \theta_2 & \cos \theta_2 & 0 & a_2 * \text{sen} \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.58

$${}^2A_3 = \begin{bmatrix} \cos \theta_3 & 0 & \text{sen} \theta_3 & a_3 * \cos \theta_3 \\ \text{sen} \theta_3 & 0 & -\cos \theta_3 & a_3 * \text{sen} \theta_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.59

$${}^3A_4 = \begin{bmatrix} \cos \theta_4 & 0 & -\text{sen} \theta_4 & 0 \\ \text{sen} \theta_4 & 0 & \cos \theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.60

$${}^4A_5 = \begin{bmatrix} \cos \theta_5 & 0 & \text{sen} \theta_5 & 0 \\ \text{sen} \theta_5 & 0 & -\cos \theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3.61

$${}^5A_6 = \begin{bmatrix} \cos \theta_6 & -\text{sen} \theta_6 & 0 & 0 \\ \text{sen} \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El primero se refiere al ejercicio de robótica ejemplificado por los marcos de referencia según *Tsai* (1999), de 0 a 3 y que equivale a la multiplicación de las transformadas individuales de los vínculos correspondientes véase ecuación (3.62).

Ecuación 3.62

$${}^0T_1 * {}^1T_2 * {}^2T_3 = {}^0T_3$$

Este es el resultado del robot de 6 grados de libertad 1rb 1600... De 0 a 3.

Ecuación 3.63

$$\begin{bmatrix} -\cos\theta_1 \sin\theta_2 \sin\theta_3 + \cos\theta_1 \cos\theta_2 \cos\theta_3 & \sin\theta_1 & \cos\theta_1 \sin\theta_2 \cos\theta_3 + \cos\theta_1 \cos\theta_2 \sin\theta_3 & a_1 \cos\theta_1 + a_2 \cos\theta_1 \cos\theta_2 - a_3 \cos\theta_1 \sin\theta_2 \sin\theta_3 + a_4 \cos\theta_1 \cos\theta_2 \cos\theta_3 \\ -\sin\theta_1 \sin\theta_2 \sin\theta_3 + \cos\theta_1 \sin\theta_1 \cos\theta_2 & -\cos\theta_1 & \sin\theta_1 \sin\theta_2 \cos\theta_3 + \sin\theta_1 \cos\theta_2 \sin\theta_3 & a_1 \sin\theta_1 + a_2 \cos\theta_2 \sin\theta_1 - a_3 \sin\theta_1 \sin\theta_2 \sin\theta_3 + a_4 \cos\theta_2 \sin\theta_1 \cos\theta_3 \\ \cos\theta_2 \sin\theta_3 + \cos\theta_2 \sin\theta_3 & 0 & -\cos\theta_2 \cos\theta_3 + \sin\theta_2 \sin\theta_3 & a_2 \sin\theta_2 + a_3 \cos\theta_2 \sin\theta_3 + a_4 \cos\theta_2 \sin\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Este sería ahora la solución del Irb 1600 de ABB de 4 a 6 que correspondería a la multiplicación de las transformadas individuales de los vínculos de 3 a 4, de 4 a 5 y de 5 a 6, véase ecuación (3.64).

Ecuación 3.64

$${}^3T_4 * {}^4T_5 * {}^5T_6 = {}^4T_6$$

..... Ahora la multiplicación de A= de 0 a 6.

Ecuación 3.65

$${}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6 = {}^0T_6$$

Ecuación 3.66

$${}^0_6T = \begin{bmatrix} ux & vx & wx & Px \\ uy & vy & wy & Py \\ uz & vz & wz & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El resultado se puede interpretar en una matriz de 4 por 4 en la cual se dividió por columnas para poder entenderla mejor.

(Los resultados obtenidos fueron realizados en código Matlab para ser utilizado en LabView). La matriz (3.66), tiene en su cuarta columna las variables Px,Py y Pz, las cuales pueden ser representadas también por xcdirecta, ycdirecta y zcdirecta, véase ecuaciones (3.67) al (3.69).

El resultado de la Transformada Homogénea que nos sirve para determinar su posición es:

Ecuación 3.67

$$\begin{aligned} xcdirecta = & x * \cosd(t) * \cosd(a) - (\cosd(t) * \sind(a)) * (y * \sind(b) + z * \cosd(b) * \sind(g) + \\ & z * \cosd(g) * \sind(b)) + (\cosd(t) * \cosd(a)) * (y * \cosd(b) + z * \cosd(b) * \cosd(g) - z * \sind(b) * \sind(g)) \end{aligned}$$

Ecuación 3.68

$$\begin{aligned} ycdirecta = & (x * \cosd(a) * \sind(t)) - (\sind(t) * \sind(a)) * (y * \sind(b) + z * \cosd(b) * \sind(g) + \\ & z * \cosd(g) * \sind(b)) + (\cosd(a) * \sind(t)) * (y * \cosd(b) + z * \cosd(b) * \cosd(g) - z * \sind(b) * \sind(g)) \end{aligned}$$

Ecuación 3.69

$$zcdirecta = [x * \sind(a) + (\cosd(a)) * (y * \sind(b) + z * \cosd(b) * \sind(g) + z * \cosd(g) * \sind(b)) + (\sind(a)) * (y * \cosd(b) + z * \cosd(b) * \cosd(g) - z * \sind(b) * \sind(g))]$$

3.3.2 Cinemática Inversa de un robot de 6 grados de Libertad

Ecuaciones Representativas para el robot de 6 Grados de Libertad

En base al estudio de Tsai (1999), en la representación de Cinemática Directa se pueden concluir algunos de los siguientes parámetros para la resolución del problema de cinemática inversa en el caso de un robot de 6 grados de libertad como se puede ver en la fig. (3.19).

Joint i	Alpha i	Ai	Di	Theta i
1	90	a1	0	θ_1
2	0	a2	0	θ_2
3	90	a3	0	θ_3
4	-90	0	d4	θ_4
5	90	0	0	θ_5
6	0	0	d6	θ_6

Tabla 3.3.Representación de los parámetros de Denavith Hartenberg para un robot articulado de 6 grados de libertad.

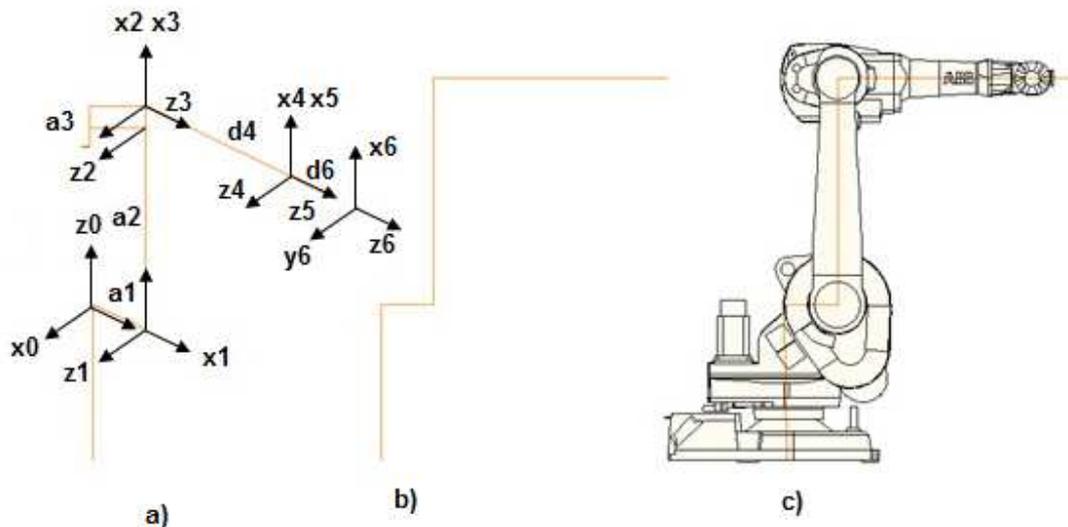


Fig. 3.19.Marcos de Referencia para robot de 6 grados de libertad.

Procedimiento para Cinemática Inversa para el robot de 6 grados de libertad

Como se observó en el ejemplo del robot de 4 grados de libertad; el primer ángulo obtenido del robot es una relación de la distancia total en la que se encuentra el robot desde su efector final con respecto a su base en los ejes "x" y "y" véase ecuación (3.70).

Ecuación 3.70

$$\theta_1 = \tan^{-1}(Py, Px)$$

$$K1 = 2a_2d_4$$

$$K2 = 2a_2a_3$$

$$K3 = Px^2 + Py^2 + Pz^2 - 2Px * a1 * \cos\theta_1 - 2Py * a1 * \sin\theta_1 + a1^2 - a2^2 - a3^2 - d4^2$$

Ecuación 3.71

$$\theta_3 = \tan^{-1} \left[\frac{k1 \pm \sqrt{k1^2 + k2^2 + k3^2}}{k3 + k2} \right]$$

$$\mu1 = a2 + a3\cos\theta3 + d4\sin\theta3$$

$$v1 = -a3\sin\theta3 + d4\cos\theta3$$

$$\gamma1 = px\cos\theta1 + py\sin\theta1 - a1$$

$$\mu2 = a3\sin\theta3 - d4\cos\theta3$$

$$v2 = a2 + a3\cos\theta3 + d4\sin\theta3$$

$$\gamma2 = pz$$

$$\cos\theta_2 = \frac{\gamma2 - \frac{v2\gamma1}{v1}}{\mu2 - \frac{v2\mu1}{v1}}$$

$$\sin\theta_2 = \frac{\gamma1 - \mu1\cos\theta_2}{v1}$$

Ecuación 3.72

$$\theta_2 = \text{Atan2}(\text{Sen}\theta_2, \text{cos}\theta_2)$$

Los valores mencionados que corresponden a las variables $u_x, u_y, u_z, v_x, v_y, v_z, y$ w_x, w_y y w_z . Son las resultantes de la matriz de transformación homogénea expuesta en la ecuación (3.65).

$$w_x = \sin\theta_1 \sin\theta_4 \sin\theta_5 + (\cos\theta_5)(\cos\theta_1 \sin\theta_2 \cos\theta_3 + \cos\theta_1 \cos\theta_2 \sin\theta_3) \\ + (\cos\theta_4 \sin\theta_5)(-\cos\theta_1 \sin\theta_2 \sin\theta_3 + \cos\theta_1 \cos\theta_3 \cos\theta_2)$$

$$w_y = -\cos\theta_1 \sin\theta_4 \sin\theta_5 + (\cos\theta_5)(\sin\theta_1 \sin\theta_2 \cos\theta_3 + \sin\theta_1 \cos\theta_2 \sin\theta_3) \\ + (\cos\theta_4 \sin\theta_5)(-\sin\theta_1 \sin\theta_2 \sin\theta_3 + \cos\theta_3 \sin\theta_1 \cos\theta_2)$$

$$w_z = (\cos\theta_4 \sin\theta_5)(\cos\theta_2 \sin\theta_3 + \cos\theta_3 \sin\theta_2) + (\cos\theta_5)(-\cos\theta_2 \cos\theta_3 \\ + \sin\theta_2 \sin\theta_3)$$

Ecuación 3.73

$$\theta_5 = \cos^{-1}(w_x * \text{Cos}\theta_1 * \text{Sen}\theta_{23} + w_y * \text{Sen}\theta_1 * \text{Sen}\theta_{23} - w_z * \text{Cos}\theta_{23})$$

$$\cos\theta_4 = \frac{w_x(\cos\theta_1 * \cos\theta_{23}) + w_y(\sin\theta_1 * \cos\theta_{23}) + w_z(\sin\theta_{23})}{\text{Sen}\theta_5}$$

$$\sin\theta_4 = -\frac{w_x(\sin\theta_1) - w_y(\cos\theta_1)}{\text{Sen}\theta_5}$$

Ecuación 3.74

$$\theta_4 = \tan^{-1}(\sin\theta_4, \cos\theta_4)$$

$$v_x = (\sin \theta_5 \sin \theta_6)(\cos \theta_1 \sin \theta_2 \cos \theta_3 + \cos \theta_1 \cos \theta_2 \sin \theta_3) + (\sin \theta_1)(\cos \theta_4 \cos \theta_6 - \cos \theta_5 \sin \theta_4 \sin \theta_6) + (-\cos \theta_1 \sin \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \cos \theta_2)(-\sin \theta_4 \cos \theta_6 - \cos \theta_4 \cos \theta_5 \sin \theta_6)$$

$$v_y = (\sin \theta_5 \sin \theta_6)(\sin \theta_1 \sin \theta_2 \cos \theta_3 + \sin \theta_1 \cos \theta_2 \sin \theta_3) - (\cos \theta_1)(\cos \theta_4 \cos \theta_6 - \cos \theta_5 \sin \theta_4 \sin \theta_6) + (-\sin \theta_1 \sin \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_1 \cos \theta_2)(-\sin \theta_4 \cos \theta_6 - \cos \theta_4 \cos \theta_5 \sin \theta_6)$$

$$v_z = (\sin \theta_5 \sin \theta_6)(-\cos \theta_2 \cos \theta_3 + \sin \theta_2 \sin \theta_3) + (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)(-\sin \theta_4 \cos \theta_6 - \cos \theta_4 \cos \theta_5 \sin \theta_6)$$

$$u_x = (\sin \theta_1)(\cos \theta_4 \sin \theta_6 + \cos \theta_6 \cos \theta_5 \sin \theta_4) - (\cos \theta_6 \sin \theta_5)(\cos \theta_1 \sin \theta_2 \cos \theta_3 + \cos \theta_1 \cos \theta_2 \sin \theta_3) + (-\sin \theta_4 \sin \theta_6 + \cos \theta_4 \cos \theta_6 \cos \theta_5)(-\cos \theta_1 \sin \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \cos \theta_2)$$

$$u_y = -(\cos \theta_1)(\cos \theta_4 \sin \theta_6 + \cos \theta_6 \cos \theta_5 \sin \theta_4) - (\cos \theta_6 \sin \theta_5)(\sin \theta_1 \sin \theta_2 \cos \theta_3 + \sin \theta_1 \cos \theta_2 \sin \theta_3) + (-\sin \theta_4 \sin \theta_6 + \cos \theta_4 \cos \theta_6 \cos \theta_5)(-\sin \theta_1 \sin \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_1 \cos \theta_2)$$

$$u_z = -(\cos \theta_6 \sin \theta_5)(-\cos \theta_2 \cos \theta_3 + \sin \theta_2 \sin \theta_3) + (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2)(-\sin \theta_4 \sin \theta_6 + \cos \theta_4 \cos \theta_6 \cos \theta_5)$$

$$\text{sen} \theta_6 = \frac{v_x(\cos \theta_1 * \text{sen} \theta_{23}) + v_y(\text{sen} \theta_1 * \text{sen} \theta_{23}) - v_z(\cos \theta_{23})}{\text{Sen} \theta_5}$$

$$\text{cos} \theta_6 = -\frac{u_x(\cos \theta_1 * \text{sen} \theta_{23}) + u_y(\text{sen} \theta_1 * \text{sen} \theta_{23}) - u_z(\cos \theta_{23})}{\text{Sen} \theta_5}$$

Ecuación 3.75

$$\theta_6 = \tan^{-1}(\text{sen} \theta_6, \text{cos} \theta_6)$$



4 CONTROL DEL BRAZO



4.1 Control con PID de un motor para brazo robótico industrial

4.1.1 *Diseño del control*

4.1.1.1 *¿Qué es el control PID?*

Como tal la entrada del controlador es usualmente una señal de error, basada en la diferencia de la señal de referencia deseada y la señal real de salida. Tal como lo menciona el libro “PID Control New Identificación and Design Methods”, el nemónico PID hace referencia a las primeras letras de los nombres individuales que en común conforman al controlador. Estas son “P” para proporcional, “I” para integral y “D” para derivativo. Y Estos tres términos son probablemente los más usados en los controladores industriales.

Como tal el control PID es una herramienta que se sustenta en base a los datos obtenidos con anterioridad lo que ofrece una amplia disponibilidad y simplicidad al usarlo. Por lo que otros tipos de controladores no son capaces de remplazar al mismo PID.

4.1.1.2 *Lógica Difusa*

Tal como lo menciona el libro “Inteligencia artificial con aplicaciones a la ingeniería”, de Ponce (2010), habla que un controlador difuso se integra por medio de cuatro partes principales: interfaz de difusificación, base de conocimientos, lógica de decisiones (inferencias) e interfaz de desdifusificación véase fig. (4.1).

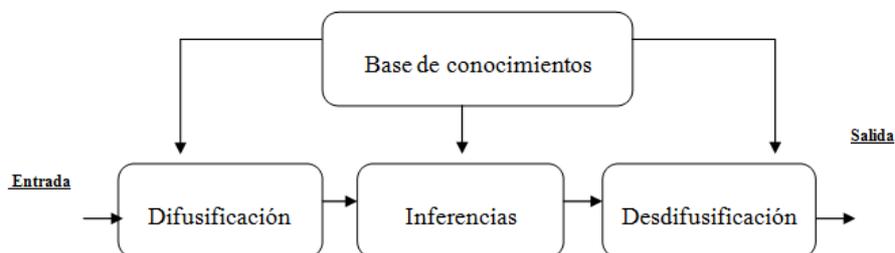


Fig. 4.1. Estructura del controlador Difuso

4.1.1.3 *Interfaz de difusificación*

La difusificación convierte los datos de entrada en valores lingüísticos que son las etiquetas de las funciones de pertenencia o conjuntos difusos. La representación de información a través de conjuntos difusos puede realizarse en forma discreta, de hecho esta es esencial cuando se están utilizando microcontroladores y se aplica control por medio de estos. Al discretizar la información, es decir tomar muestras con respecto al tiempo en un universo; y en un número definido por partes, es posible definir un conjunto difuso asignando un grado de pertenencia a cada elemento genérico del nuevo discurso discreto.

Una variable lingüística es un conjunto de términos generalmente y está definido en el mismo universo de discurso. Para encontrar cuantos términos son necesarios en un conjunto se emplean particiones difusas. De esta manera el número de conjuntos difusos determina la complejidad del controlador, lo que puede generar que consuma más energía o memoria en la ejecución del programa véase fig. (4.3). Por eso se debe de ser cauteloso en el momento de utilizarlo. Estos como tal tienen significados lingüísticos como: negativo grande, cero y positivo pequeño, aunque cabe mencionar que pueden llamarse de cualquier otra forma y podrían ser mas. Así como además comúnmente son normalizados utilizando un rango de -1 a 1, véase fig. (4.2).

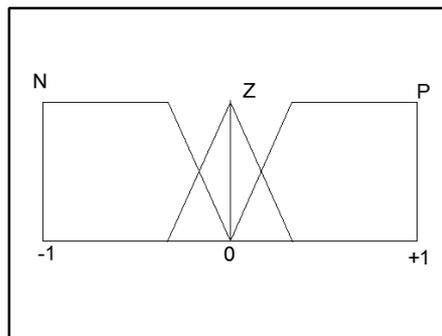


Fig. 4.2. Partición difusa con tres términos

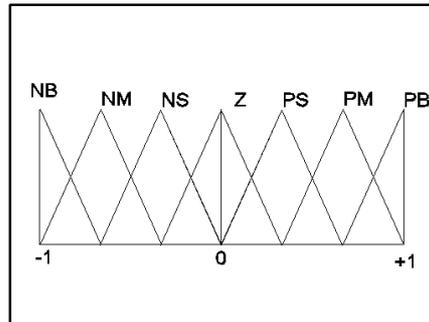


Fig. 4.3. Partición difusa con 7 términos.

4.1.1.4 Base de conocimientos

Contiene toda la información que se va a controlar, así como el resultado a donde queremos enfocar el control, es decir consta de una base de datos y una base de reglas lingüísticas para controlar la variable. La base de datos proporciona las definiciones para el establecimiento de reglas y la manipulación de datos difusos. La base de reglas caracteriza metas de control y la política que llevan los expertos para realizar el control, empleando proposiciones. Un algoritmo de control difuso debe de ser capaz de inferir en una acción de control correspondiente para cada estado del proceso que se va a controlar, propiedad que se denomina unidad, La estrategia de la base de datos comprende los soportes de la definición de los conjuntos difusos.

Existen cuatro métodos de derivación de las reglas difusas de control y las acciones de control de un operador, con la forma de proposiciones condicionales que relacionan las variables de estado en el antecedente del proceso con las variables de control del proceso en las consecuencias. El conjunto de reglas se define como:

Ecuación 4.1

$$(*_1 \text{ es } L_{x_1}^{(k)} \text{ y } (\dots)\dots(\dots)\text{y}(*_n \text{ es } L_{x_n}^{(k)})), k=1,2,\dots,m$$

Donde:

X: Dominio físico Actual donde tienen sentido los valores lingüísticos.

L_x : Conjunto de valores lingüísticos que * puede tomar, un elemento arbitrario es L_x .

*: Nombre simbólico.

U_x : Función semántica de interpretación de un valor lingüístico en términos cuantitativos.

Ecuación 4.2

$$\mu_x: L_x \rightarrow \widetilde{L}_x.$$

4.1.1.5 *Lógica de decisiones*

La lógica utilizada para tomar decisiones de un controlador difuso es el núcleo del mismo. A partir de la misma se simula la lógica que utilizan las personas para tomar decisiones, en base a conceptos difusos y a la inferencia de acciones de control, empleando implicaciones y las reglas establecidas según la base de conocimientos.

El resultado de utilizar estas reglas con valores de entrada físicos nítidos será m conjuntos difusos recortados los cuales se denotan por:

Ecuación 4.3

$$C_{\widetilde{L}_u}^{(1)}, \dots, C_{\widetilde{L}_u}^{(m)}$$

O m conjuntos escalados:

Ecuación 4.4

$$S_{\widetilde{L}_u}^{(1)}, \dots, S_{\widetilde{L}_u}^{(m)}$$

Lo que gráficamente se puede ver representado por la fig. (4.4) y la fig. (4.5).

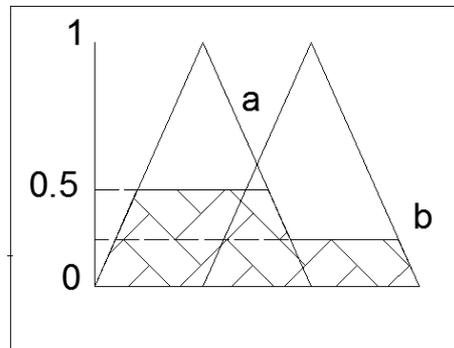


Fig. 4.4. Conjuntos recortados

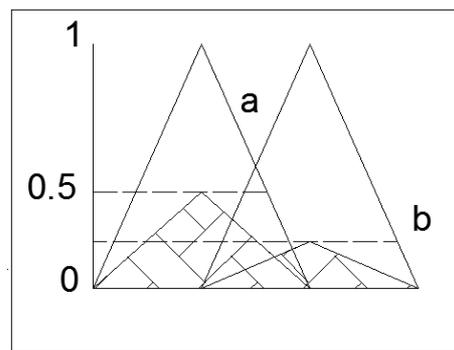


Fig. 4.5. Conjuntos escalados

4.1.1.6 *Interfaz de desdifusificación*

La interfaz de desdifusificación como tal se encarga del muestreo escalar que convierte el rango de valores de las variables de salida a universos de discurso correspondiente. Y como tal se puede definir como la herramienta para obtener la acción de control nítida a partir de una acción de control difusa.

La salida de control \tilde{u} o μ_u se obtiene como la unión de las salidas de control cortadas o escalonadas.

Ecuación 4.5

$$\tilde{u} = u_{k=1}^m c\tilde{L}u^k$$

Donde el valor nítido (que representa al conjunto difuso) se denota por u^* y el área del conjunto se define por:

Ecuación 4.6

$$\int_u \mu_u(u)du$$

4.1.1.7 *Diseño de Controlador Mamdani*

Mamdani propuso el primer controlador difuso en los años setenta y comprende el siguiente procedimiento básico:

Siendo el error la diferencia entre el valor deseado y el valor real de la variable que se va a controlar, se seleccionan las funciones de pertenencia que realizarán la difusificación.

Se establecen las reglas a partir de proposiciones condicionales, y el dispositivo de inferencia será una composición máx.-min que ha sido previamente definida como:

Ecuación 4.7

$$X_T(x, z) = \vee_{y \in Y} (X_R(x, y) \wedge X_S(y, z))$$

Se seleccionan las funciones de pertenencia para la desdifusificación y el método que se va a usar para encontrar el valor nítido de la salida, normalmente correspondiente al método del centroide donde:

Ecuación 4.8

$$\frac{\sum \mu(x) \cdot x}{\sum \mu(x)} \text{ Para el caso discreto...}$$

4.1.1.8 *Porque usar Matlab?*

Como tal *Matlab* es una poderosa Herramienta en la resolución de Matrices, Desarrollador de ambientes gráficos, y simulador de Procesos control por medio de su Herramienta (*Simulink*), Además es un programa ampliamente utilizado por investigadores en la simulación de procesos.

Además, puede realizar comunicaciones entre la *PC* y un dispositivo exterior; así como lo hace *LabView* pero a diferencia de esto lo más fácil es desarrollar una interfaz con alguna comunicación serial, o en su caso utilizando un dispositivo por los puertos COM tales como el modulo Ft232 y un complemento como antenas de comunicación *wifi, bluetooth, Lan* entre otras. Así como Matlab existe la ventaja de poder utilizar programas que tengan una interacción similar como lo es la aplicación *Octave* que permite realizar un código similar y que es de uso libre aunque para su practicidad se utilizará el programa Matlab.

Una ventaja mas es que el programa *LabView* tiene la capacidad de utilizar un código similar a Matlab por lo tanto puede haber compatibilidad para integrar los programas y poder desarrollar aplicaciones de mayor calidad.

4.1.1.9 *Que es el modelado?*

Como tal es modelado se puede interpretar de forma sencilla como un método para controlar un proceso que tiene condiciones de operación, una o más entradas, una o unas variables de estado y una o más salidas. Es decir se puede interpretar un sistema mecánico, eléctrico, neumático o hidráulico en el cual relaciona los datos conocidos como entradas, Información que influye en los procesos que tengan relación con la entrada y la misma salida que es en efecto la información que queremos monitorear.

Como lo menciona el libro de Ogata (2003) de "Ingeniería de control moderno", Si se puede obtener un modelo matemático de la planta, es posible aplicar diversas técnicas de diseño con el fin de determinar los parámetros del controlador que cumpla las especificaciones en estado transitorio y en estado estable del sistema en lazo cerrado. Sin embargo, si la planta es tan complicada que no es fácil obtener su modelo matemático, tampoco es posible un enfoque analítico para el diseño de un controlador PID. En este caso, debemos recurrir a los enfoques experimentales para la sintonización de los controladores PID.

El proceso de selección de los parámetros del controlador para que cumplan con las especificaciones de desempeño se conoce como sintonización del controlador. Ziegler y Nichols sugirieron más reglas para sintonizar los controladores PID (lo cual significa establecer valores K_p , T_i y T_d), con base en las respuestas escalón experimentales o basadas en el valor de K_p que se produce en la estabilidad marginal cuando sólo se usa la acción de control proporcional.

4.1.2 Resultados de Simulación

4.1.2.1 Control Proporcional-Derivativo Difuso para el control de posicionamiento de un motor de Cd marca Pittman. Serie 14200.

Este trabajo se basa en la posibilidad de poder utilizar una amplia gama de motores de Cd, con el fin de poderlos integrar con fines Robóticos; y para eso se ha modelado el comportamiento de un motor de acuerdo a sus características especificadas por el fabricante. De esta forma se podrá generar un análisis el cual pretenda parecerse en lo más posible al real. Por medio de técnicas de Control tales como control P, PI difuso y PID Difuso, se podrá observar su comportamiento lo cual servirá como referencia para posteriores aplicaciones.

El objeto de este Trabajo se basa en el fundamento de poder utilizar técnicas fuera de las convencionales para que nos puedan ayudar a desarrollar diferentes tipos de estrategias en el control de Motores de CD. Como es bien sabido esta parte del trabajo involucra de forma peculiar la manera de poder realizar una interacción entre la posición de un motor con respecto a una entrada en específico, trabajo conjunto que servirá para el desarrollo de un Robot. Es decir este trabajo se basa en el control de posición de motores de forma independiente.

El objeto de este desarrollo es poder definir en primera instancia una planta la cual simule el comportamiento de un motor de CD. Una vez obtenido esto se podrá interactuar con diferentes técnicas de control para poder interpretar algún sistema que pueda ser óptimo en el funcionamiento del mismo.

Una vez vista la planta en lazo abierto, aplicada ante una entrada escalón unitario, se observa que su respuesta es una función rampa, véase figura (4.6), si se considera que la

entrada es un voltaje constante entonces el ángulo de salida tendera a incrementarse de forma desmesurada como es el caso.

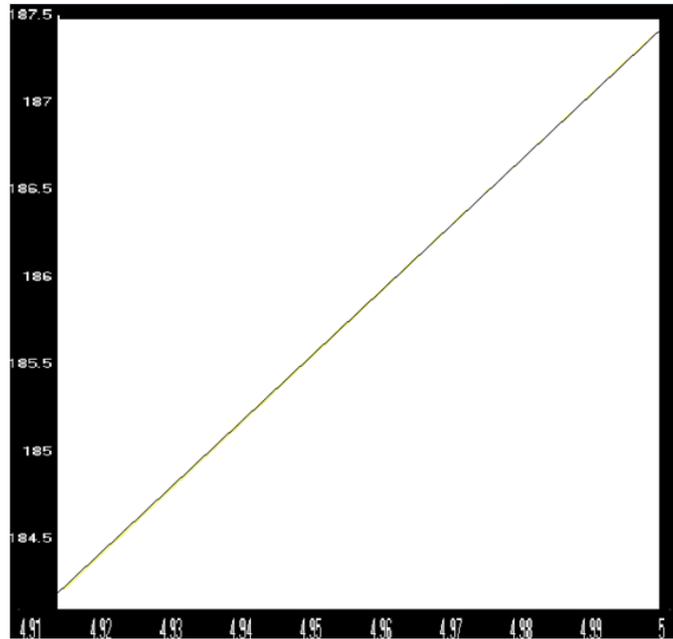


Fig. 4.6. Función de Transferencia en lazo Abierto del Motor.

La fig. (4.7).Muestra el resultado de lo que significa sintonizar mal. Dando como efecto una inestabilidad por parte de la función de Membresía de salida.

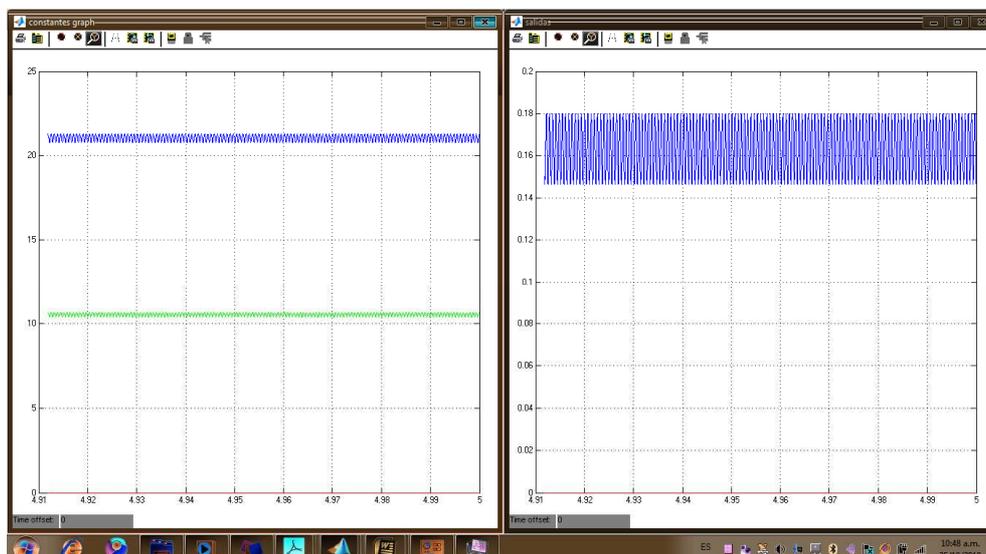


Fig. 4.7. Gráfica de una mala sintonización donde K_p es demasiado grande.

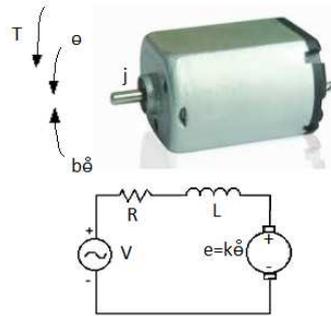


Fig. 4.8 Diagrama Básico para El modelo de un motor de CD.

El diagrama con que se puede partir en la creación de la función de transferencia se puede observar en la fig. (4.8). La cual está compuesto por la Fuente de voltaje que suministra al motor, su resistencia interna y sus bobinas.

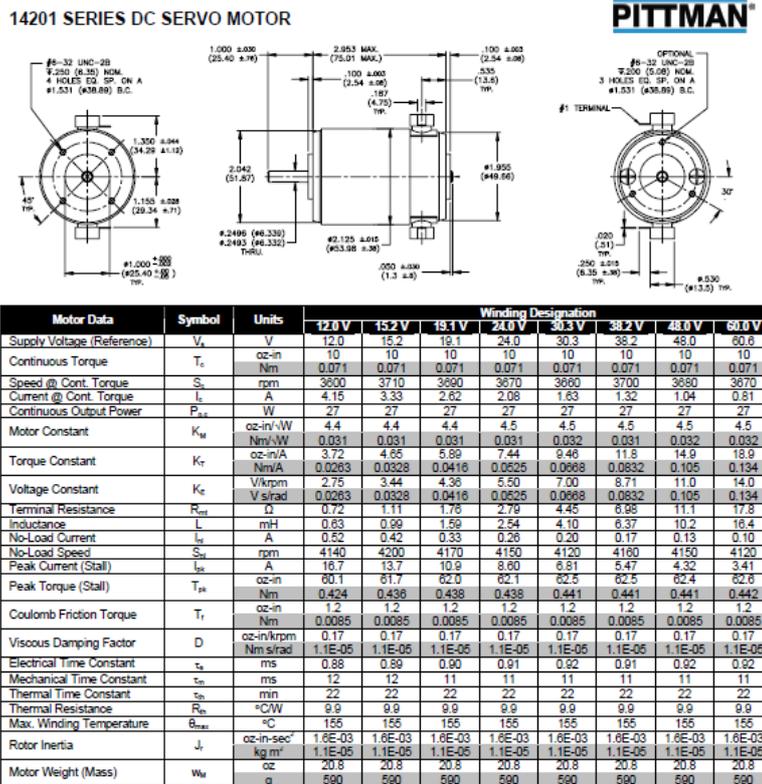


Fig. 4.9. Hojas de Especificaciones del motor Pittman 14201.



Fig. 4.10. Marca de Motores Pittman.

4.1.2.2 MOTOR 14201s001 DE PITTMAN AMETEK.

Utilizando el ejemplo mostrado con el modelo 14201s001 Pittman de motores de CD con escobillas véase la fig. (4.9). Obtendremos los siguientes datos:

Momento de inercia del rotor (J) = $1.1 \text{ E } -5 \text{ kg.m}^2/\text{s}^2$ (Para fines de motores y demás, El momento de inercia del rotor es igual a la inercia del rotor solo que uno hace referencia a fines lineales y los otros angulares).

Inercia del rotor (J_r) = $1.1 \text{ E } -5 \text{ kg}\cdot\text{m}^2$

El radio de amortiguamiento de sistema mecánico es también llamado coeficiente de fricción o como sus siglas en inglés (*Damping ratio the mechanical system*). Otra forma de llamarlo sería (*Viscous damping factor*).

En algunos casos el radian es una unidad la cual omiten por lo que en las hojas de especificaciones podrían no especificarla.

Factor de Amortiguamiento Viscoso ($B=b$) $1.1 \text{ E } -05 \text{ Nms/rad}$ ($.000011 \text{ Nm} \cdot \text{s/rad}$)

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

Constante de la fuerza electromotriz (*torque constant*) ($K=K_e=K_t$) = .0263Nm/Amp a 12 v.

Resistencia Eléctrica (R) = .72 ohm a 12 volts

Inductancia (L) = .63 E-3 Hertz a 12 v (.00063 Hertz a 12 volts).

Entrada (V): Source Voltage 12v.

Salida (θ): position of shaft *tetha*.

Con los datos mencionados se obtiene la siguiente Función de Transferencia:

Ecuación 4.9

$$\frac{\theta}{v} = \frac{k}{s^3(JL) + s^2(JR + BL) + s(K^2 + BR)}$$

Sustituyendo valores de:

El torque del motor, T es relacionado a la corriente de la armadura, i , por un factor constante K_t . La fuerza contraelectromotriz, e es relacionada por las siguientes ecuaciones:

Ecuación 4.10

$$T = K_t i$$

Ecuación 4.11

$$e = K_c \dot{\theta}$$

En las unidades del sistema internacional (la cual nosotros vamos a usar), K_t (constante de armadura) es igual a K_e (constante del motor).

Las siguientes ecuaciones están basadas en la combinación de las leyes de newton y las leyes de Kirchoff.

Ecuación 4.12

$$J\ddot{\theta} + b\dot{\theta} = K_i$$

Ecuación 4.13

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

Usando la transformada de Laplace, las siguientes ecuaciones pueden ser expresadas de la forma:

Ecuación 4.14

$$s(Js + b)\theta(s) = KI(s)$$

Ecuación 4.15

$$(Ls + R)I(s) = V - Ks\theta(s)$$

El proceso completo:

Ecuación 4.16

$$I(s) = \frac{V - K * s * \theta(s)}{(Ls + R)}$$

Ecuación 4.17

$$I(s) = \frac{s(Js + b) * \theta(s)}{(k)}$$

Ecuación 4.18

$$\frac{V - K * s * \theta(s)}{(Ls + R)} = \frac{s(Js + b) * \theta(s)}{(k)}$$

Ecuación 4.19

$$\frac{V}{(Ls + R)} - \frac{K * s * \theta(s)}{(Ls + R)} = \frac{s(Js + b) * \theta(s)}{(k)}$$

Ecuación 4.20

$$\frac{-s(Js + b) * \theta(s)}{(k)} - \frac{K * s * \theta(s)}{(Ls + R)} = -\frac{V}{(Ls + R)}$$

Ecuación 4.21

$$\frac{s(Js + b) * \theta(s)}{(k)} + \frac{K * s * \theta(s)}{(Ls + R)} = \frac{V}{(Ls + R)}$$

Ecuación 4.22

$$\theta(s) \left(\frac{s(Js + b)}{(k)} + \frac{K * s}{(Ls + R)} \right) = \frac{V(s)}{(Ls + R)}$$

Ecuación 4.23

$$\frac{\theta(s)}{V(s)} \left(\frac{s(Js + b)}{(k)} + \frac{K * s}{(Ls + R)} \right) = \frac{1}{(Ls + R)}$$

Ecuación 4.24

$$\frac{\theta(s)}{V(s)} = \frac{\frac{1}{(Ls + R)}}{\left(\frac{s(Js + b)}{(k)} + \frac{K * s}{(Ls + R)}\right)}$$

Ecuación 4.25

$$\frac{\theta(s)}{V(s)} = \frac{1}{(Ls + R) \left(\frac{s(Js + b)}{(k)} + \frac{K * s}{(Ls + R)}\right)}$$

Ecuación 4.26

$$\frac{\theta(s)}{V(s)} = \frac{1}{(Ls + R) \left(\frac{s(Js + b)}{(k)}\right) + K * s}$$

Ecuación 4.27

$$\frac{\theta(s)}{V(s)} = \frac{1}{\left(\frac{(Js + b)(Ls + R)}{(k)} + K\right) * s}$$

Ecuación 4.28

$$\frac{\theta(s)}{V(s)} = \frac{1}{\frac{1}{k} ((Js + b)(Ls + R) + K^2) * s}$$

La Función de transferencia que puede identificar la relación entre la posición y su voltaje se describe de la siguiente manera sin considerar unidades y simplificando:

Ecuación 4.29

$$\frac{\theta(s)}{V(s)} = \frac{(.0263)}{s^3(.00000000693) + s^2(.000008) + s(.0007)}$$

4.1.3 Estabilidad por Ruth-Hurwitz

La metodología que se utiliza según Bolton (2005), es determinar la estabilidad de un sistema dada su función de transferencia, lo que implica también determinar las raíces del polinomio del denominador de la función y considerar si cualesquiera son o no positivas. Sin embargo, las raíces no se pueden obtener con facilidad si el polinomio del denominador es de la forma:

Ecuación 4.30

$$a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_1 s + a_0$$

Y n es mayor que 3 o 4. El criterio de Routh Hurwitz, sin embargo, representa un método que se puede usar en tales situaciones.

La primera prueba que se aplica es revisar los coeficientes. Si estos son todos positivos y ninguno es cero, el sistema es estable. Si cualquiera de los coeficientes es negativo, entonces el sistema es inestable y si cualquiera de los coeficientes es cero, entonces en el mejor de los casos el sistema es críticamente estable.

Una vez que se puede observar si los coeficientes denotan estabilidad se lleva a cabo una segunda prueba la cual consiste en seguir el siguiente arreglo:

Ecuación 4.31

$$\begin{array}{l} s^n \\ s^{n-1} \\ s^{n-2} \end{array} \left| \begin{array}{ccc} a_n & a_{n-2} & a_{n-4} \\ a_{n-1} & a_{n-3} & a_{n-5} \\ b_1 & b_2 & b_3 \end{array} \right.$$

Donde los valores que necesitamos de b_1 y b_2 .

Ecuación 4.32

$$b_1 = a_{n-2} - \left(\frac{a_n}{a_{n-1}} \right) a_{n-3}$$

Ecuación 4.33

$$b_2 = a_{n-2} - \left(\frac{a_n}{a_{n-1}} \right) a_{n-3}$$

Por lo que utilizando el denominador de la función de transferencia se puede utilizar el método de Routh Hurwitz para su comprobación.

Ecuación 4.34

$$\frac{\theta(s)}{V(s)} = \frac{(0.0263)}{s^3(.00000000693) + s^2(.000008) + s(.0007)}$$

Utilizando las ecuaciones b1 y b2 se establece el criterio de Ruth Hurwitz:

Ecuación 4.35

$$\begin{array}{l}
 s^3 \\
 s^2 \\
 s^1
 \end{array}
 \left| \begin{array}{ccc}
 .00000000693 & .0007 & 0 \\
 .000008 & 0 & \\
 .000008 & &
 \end{array} \right.$$

Una vez que se ha completado el arreglo se verifica y si todos los elementos de la primera columna son positivos, entonces todas las raíces tienen partes reales negativas, por lo que están en el lado izquierdo del patrón de polos y ceros. Por lo que el sistema es estable. Y en su efecto de ser cero, críticamente estable.

Otra forma de determinar si el sistema es estable es buscando las raíces en el denominador de la función de transferencia, sabiendo que el denominador es igual a:

$$.00000000693s^3 + .000008s^2 + .0007s = 4.4352 \times 10^{-13}s(1.8038 \times 10^7s + 15625.0s^2 + 1.5783 \times 10^9)$$

Y sabiendo que las raíces son 0, -1059 y -95.381 el sistema es críticamente estable.

4.1.4 El primer método de sintonización Ziegler Nichols

El primer método de sintonización de Ziegler-Nichols del libro de Ogata (1998), se basa en la respuesta de la planta de una entrada escalón unitario, el cual se obtiene experimentalmente. Si la planta no contiene integradores ni polos dominantes complejos conjugados entonces la curva puede tener forma de “S”. Por lo que tales curvas de respuesta escalón se generan experimentalmente o a partir de la simulación dinámica de la planta.

La curva con forma de S se caracteriza por dos parámetros: el tiempo de retardo L y la constante de tiempo T. El tiempo de retardo y la constante de tiempo se determinan dibujando una recta tangente en el punto de inflexión de la curva con forma de S y determinando las intersecciones de esta tangente con el eje del tiempo y la línea $c(t)=K$, por lo cual para este caso, la función de Transferencia $C(s)/U(s)$ se aproxima mediante un sistema de primer orden con un retardo de transporte del modo siguiente:

Ecuación 4.36

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1}$$

Por lo que Ziegler y Nichols sugirieron establecer los valores de K_p , T_i y T_d de acuerdo con la fórmula que aparece en la tabla (4.1) (“se referencia con la clásica de Ziegler-nichols”).

Tipo de Controlador	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	2L	0.5L

Tabla 4.1. Para control PID según Ogata.

Considerando a un lazo cerrado, se utiliza la función de transferencia para evaluar el diagrama con un valor unitario para obtener su curva característica véase fig. (4.11).

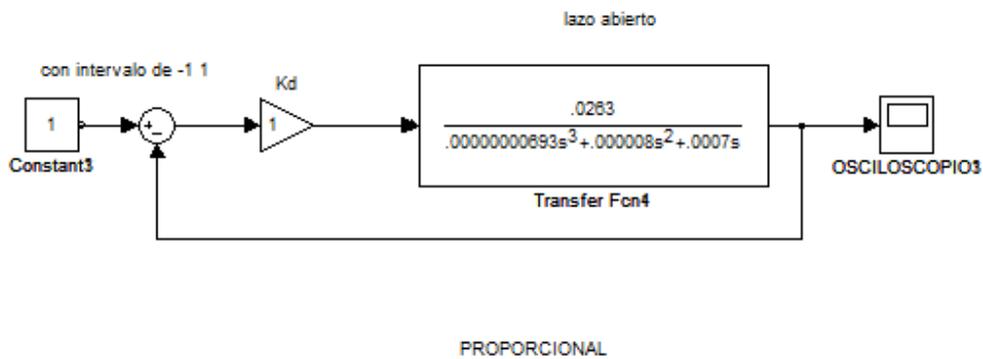


Fig. 4.11. Diagrama de control en lazo cerrado.

La función de respuesta obtenida por medio de un diagrama de control retroalimentado es el siguiente como se muestra en la fig. (4.12).

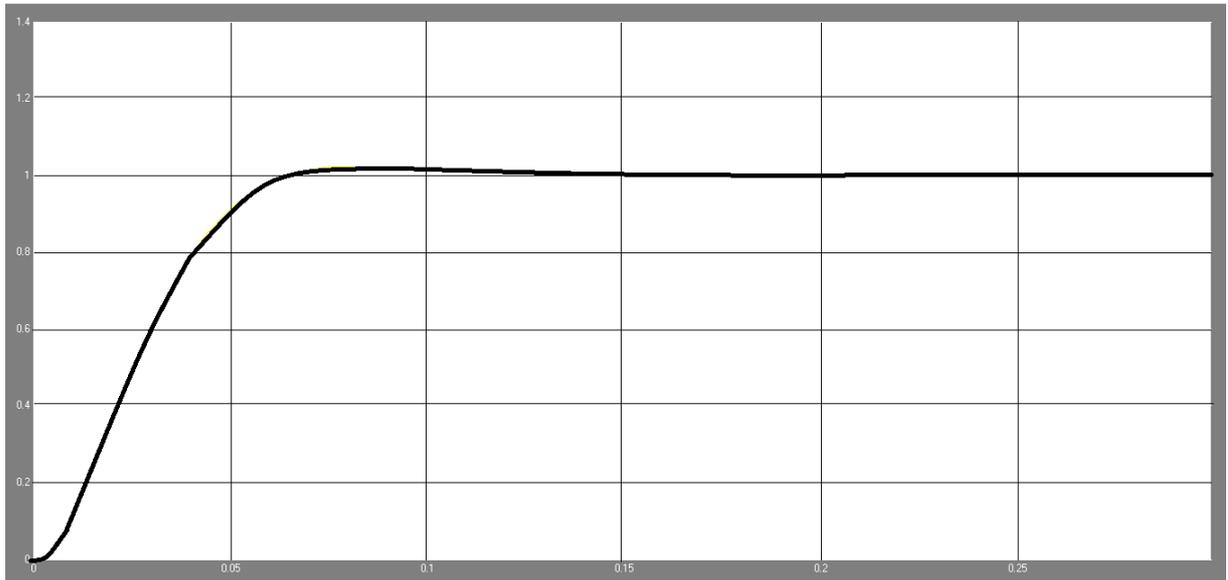


Fig. 4.12. Respuesta de Función de transferencia en lazo cerrado para control proporcional.

Tipo de Controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

Tabla 4.2. Constantes para el control PID según Ogata.

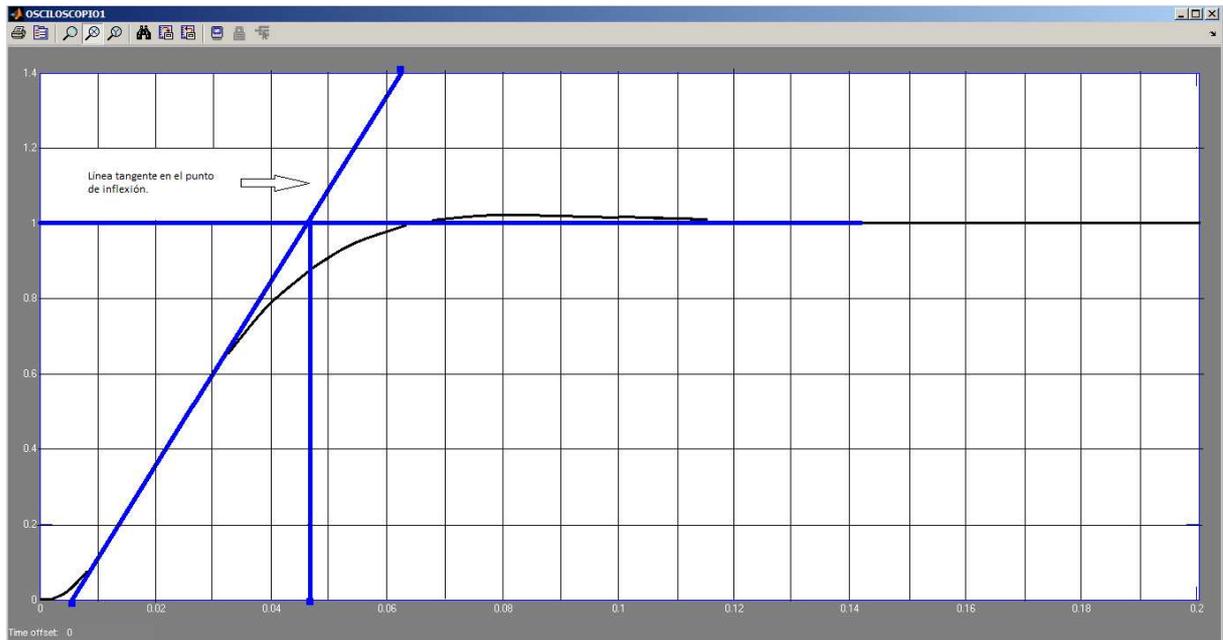


Fig. 4.13. Gráfica con la línea tangente dibujada en la función de transferencia.

Debido a que la gráfica se presenta en forma de S, se procede a utilizar el primer método de sintonización de Ziegler-Nichols véase fig. (4.13) y fig. (4.14).

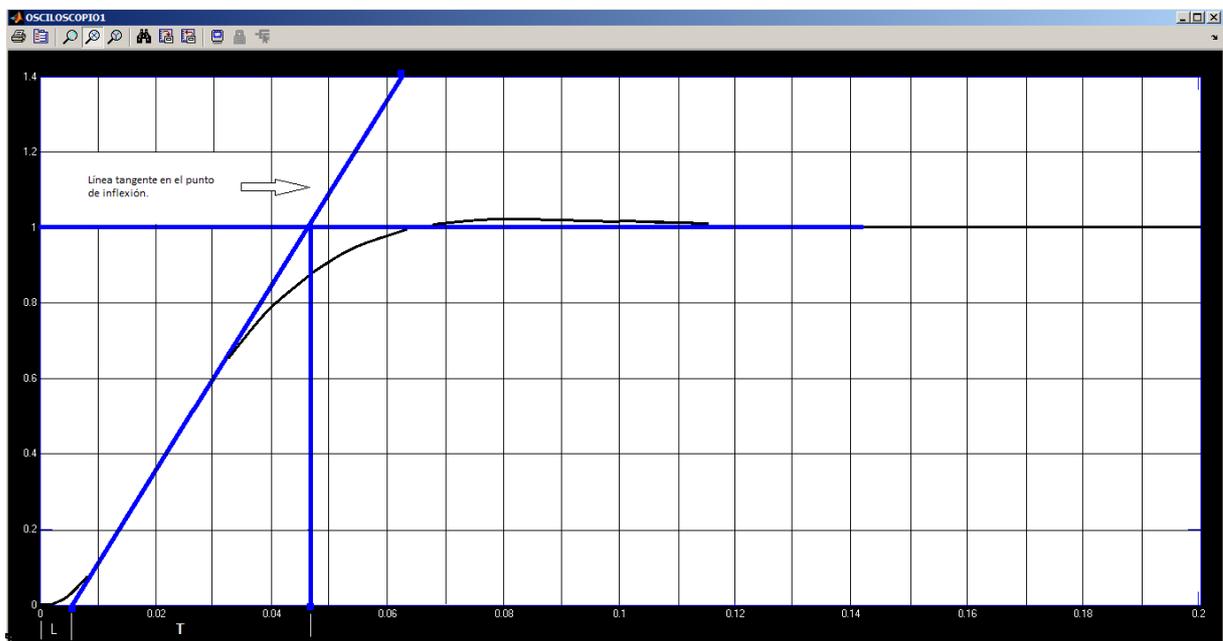


Fig. 4.14. Obtención de los valores T y L experimentales.

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

Para la gráfica anterior el valor de L es: (.006) y el valor de T es de (.0415).

Tipo de Controlador	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	2L	0.5L

Tabla 4.3. Tabla de valores para control P, PI, PID.

Los Resultados Obtenidos sustituyendo las constantes de T y L por medio de la Tabla del Ogata son:

Tipo de Controlador	Kp	Ti	Td
P	6.916666	∞	0
PI	6.2222	0.02	0
PID	8.29999	0.012	.003

Tabla 4.4. Sustitución de valores obtenidos de acuerdo a T y L.

En la fig. (4.15), se puede observar el diagrama de control proporcional sintonizado por el método Z-N, y en la fig. (4.16) su respuesta.

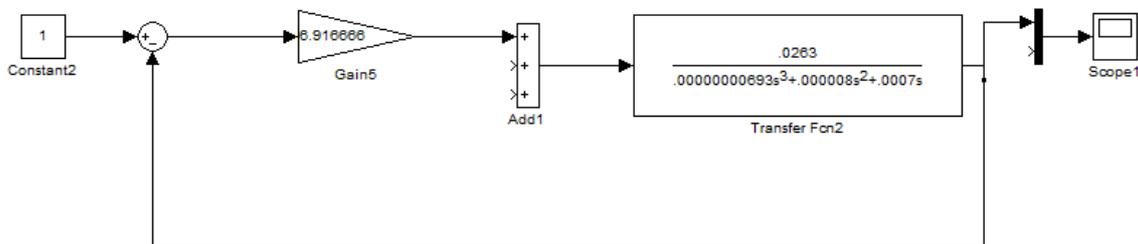


Fig. 4.15. Diagrama de Control sintonizado por el método Proporcional.

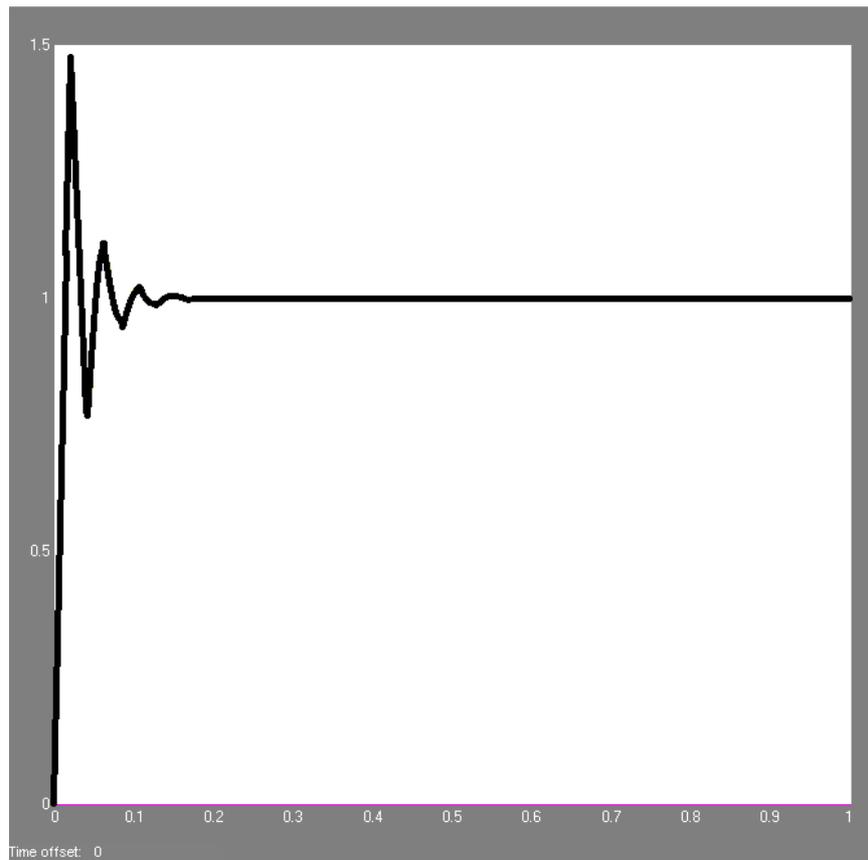


Fig. 4.16. Respuesta de Control sintonizado por el método Proporcional.

En la fig. (4.17), se muestra el diagrama de control proporcional derivativo desarrollado en Simulink, mientras que en la fig. (4.18), se muestra su correspondiente respuesta.

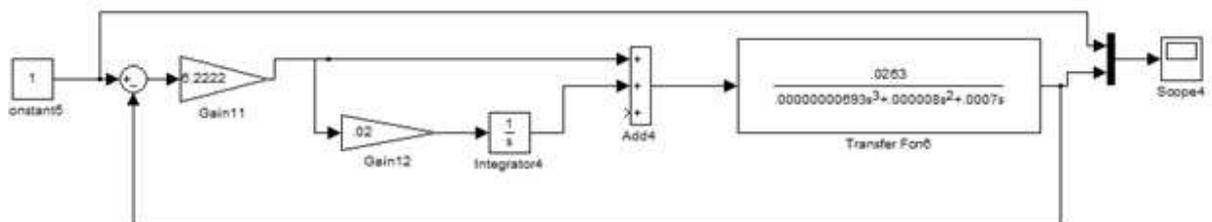


Fig. 4.17. Diagrama de Control sintonizado por el método Proporcional Integral.

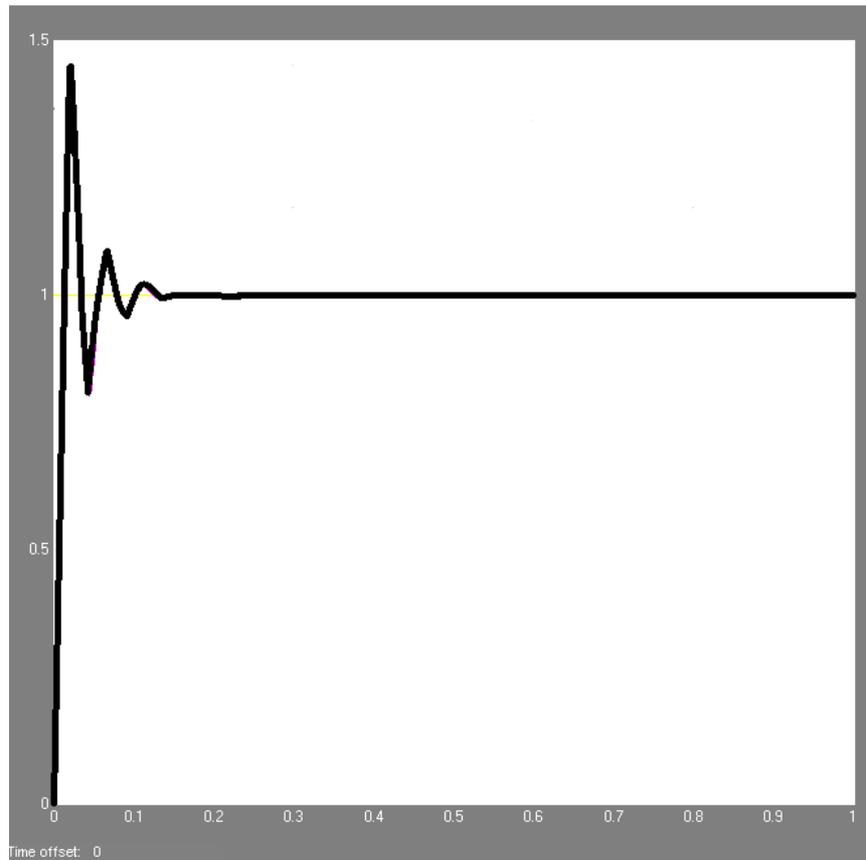


Fig. 4.18. Respuesta de Control sintonizado por el método Proporcional Integral.

Por último se observa en la fig. (4.19), el diagrama utilizado con la sintonización de método de Ziegler and Nichols, mientras que en la fig. (4.20), se muestra que la respuesta responde mejor que en los dos casos anteriores.

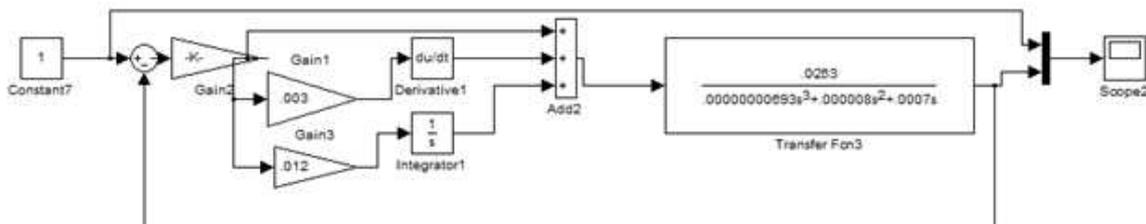


Fig. 4.19. Diagrama de Control sintonizado por el método Proporcional Integral Derivativo.

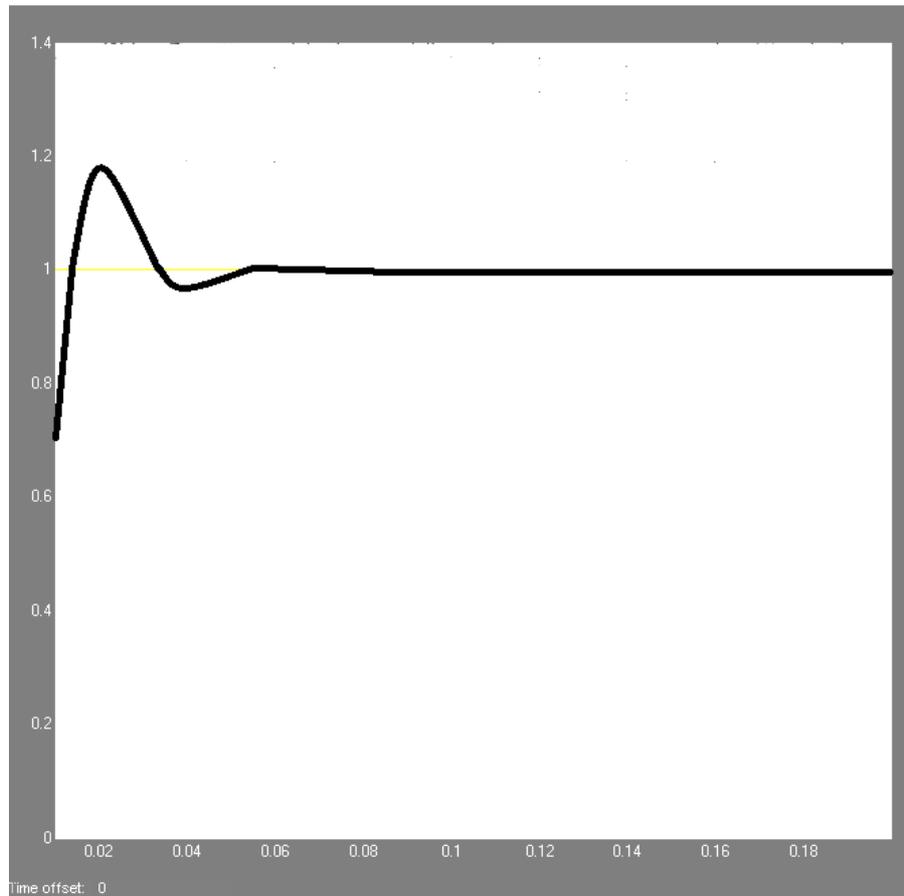


Fig. 4.20. Respuesta de Control sintonizado por el método Proporcional Integral Derivativo.

4.1.5 Sintonización por el método de Ziegler Nichols.

En lo que se basa a la experiencia, se prefiere a utilizar el método número 2 tomando en cuenta que la gráfica debe presentar un impulso mayor al 5%.

Utilizando el segundo método, primero se establece según Ogata (2001), $T_i = \infty$ y $T_d = 0$. Usando solo la acción de control proporcional, se incrementa K_p de 0 a un valor crítico K_{cr} en donde la salida trata de exhibir oscilaciones sostenidas. (Si la salida no presenta oscilaciones sostenidas para cualquier valor que pueda tomar K_p , no se debe de aplicar este

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

método). Por lo tanto la ganancia crítica K_{cr} y el periodo P_{cr} correspondiente se determinan experimentalmente. Ziegler y Nichols sugirieron que se establecieran los valores de los parámetros K_p , T_i y T_d de acuerdo a la siguiente tabla:

Tipo de Controlador	K_p	T_i	T_d
P	$.5K_{cr}$	∞	0
PI	$.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Tabla 4.5. Regla de Sintonización de Ziegler-Nichols basada en la ganancia crítica K_{cr} y en el periodo P_{cr} por el segundo método.

La simulación realizada en Simulink así como la física debe de ser regida por las siguientes ecuaciones que caracterizan al diagrama de control.

Ecuación 4.37

$$G_{pid} = E(s) [k_p + K_d s + K_i / s]$$

Ecuación 4.38

$$E(s) [K_p (1 + (1/t_i) * s + t_d * s)]$$

Y sabemos el valor de **T_i** y **T_d** :

Ecuación 4.39

$$K_d = (k_p * t_d)$$

Ecuación 4.40

$$K_i = (k_p / t_i)$$

4.1.5.1 *El control Proporcional Integrativo Difuso.*

De acuerdo al autor Ponce (2010), se realizaron algunas pruebas con algunas funciones de Membresía para determinar el comportamiento de la salida con respecto a un ángulo como referencia.

En este ejemplo se realizaron pruebas del Control Proporcional Integrativo Difuso. Para esto fue necesario realizar el siguiente diagrama de bloques por medio de Simulink véase fig. (4.21), en el cual tenemos dos graficas representadas: la primera se basa en la constante de Kp. Después de esto se utilizo el Método de Ziegler Nichols para poder sintonizar la constante necesaria para poder realizar el control.

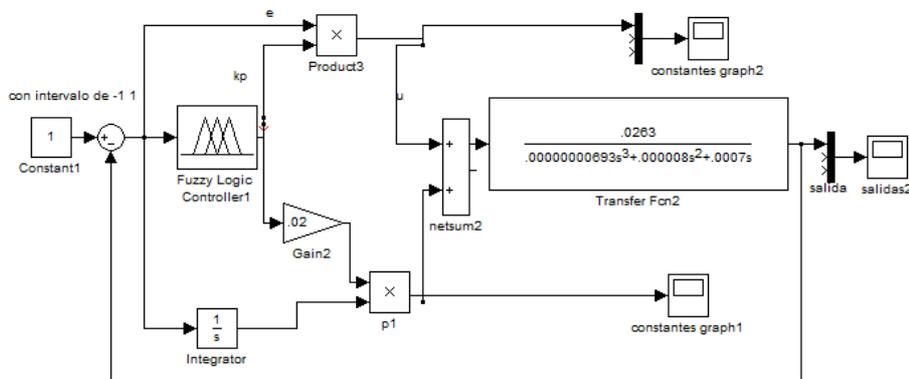


Fig. 4.21. Control Proporcional Integrativo de un control de CD.

En el próximo ejemplo véase fig. (4.22), se puede mostrar lo que equivaldría a los Sistemas cerrados considerando una constante definida y al controlador difuso.

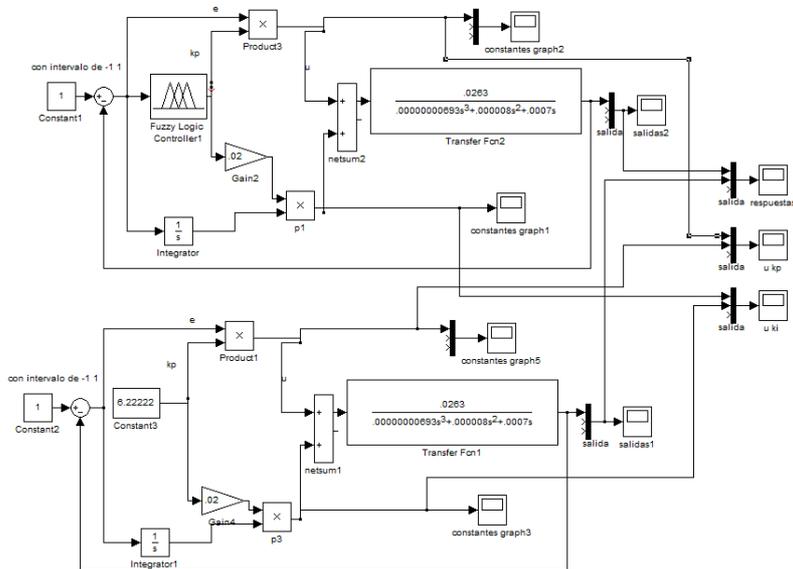


Fig. 4.22. Comparativo entre un Control Proporcional Derivativo y un Control Proporcional derivativo difuso.

El primer concepto se basa en analizar una sola posición y su estabilidad, para poder diseñar un sistema adecuado se optó en primera instancia aplicar un control proporcional a la función de transferencia antes mencionada, el diagrama consta de una entrada que simula una parte proporcional y su planta.

Ahora se puede utilizar los conceptos según Ponce (2010), para observar los resultados obtenidos en la fig. (4.23) y la fig. (4.24). Para la salida del diagrama de control por $K_p * e$, y $K_p * e$ Difuso así como la “u” del producto de $K_c r$ con T_i y T_i difuso. Así como la respuesta entre la salida del control PD Y PD Difuso, véase fig. (4.25).

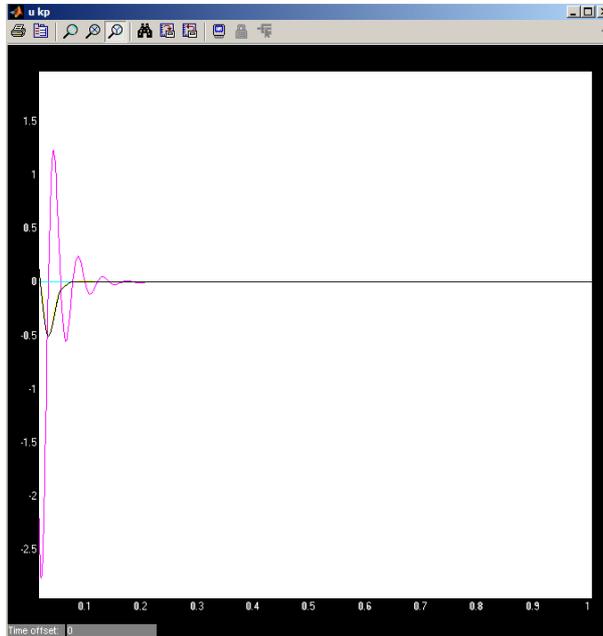


Fig. 4.23. Salida del producto “Kp por e y Kp por e Difuso”.

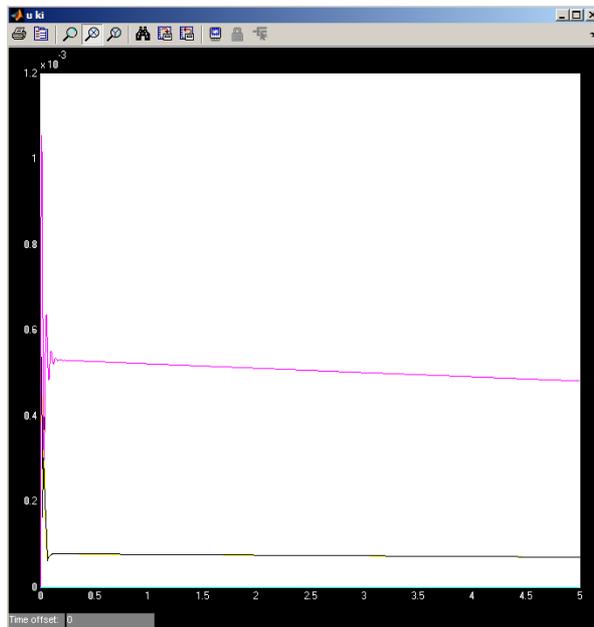


Fig. 4.24. Salida del producto “Ker con Ti y Ti Difuso”.

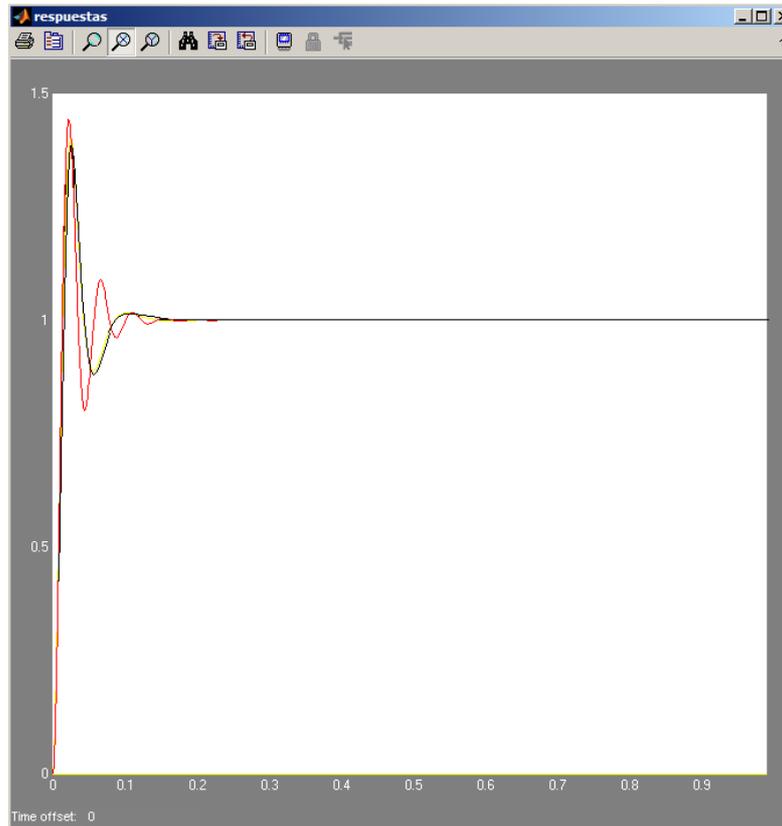


Fig. 4.25. Comparativa entre las salidas del Control PI y Control PI Difuso.

4.1.5.2 *PID Difuso*

En el último caso se desarrollo un control PID Difuso para controlar el posicionamiento del motor con respecto al voltaje de entrada. Se consideraron las siguientes funciones de membresía de entrada y salida, véase fig. (4.26) y fig. (4.27). Así como sus correspondientes reglas para determinar las salidas de **Kp** véase fig. (4.28). puesto que el sistema se sintonizara según la metodología de Ziegler-Nichols, Además cabe destacar como lo redacta el autor Ponce (2010), “se aproximan los factores del método de Ziegler y

Nichols, ya que para el método exacto no se requiere únicamente la K_p , sino también el periodo”.

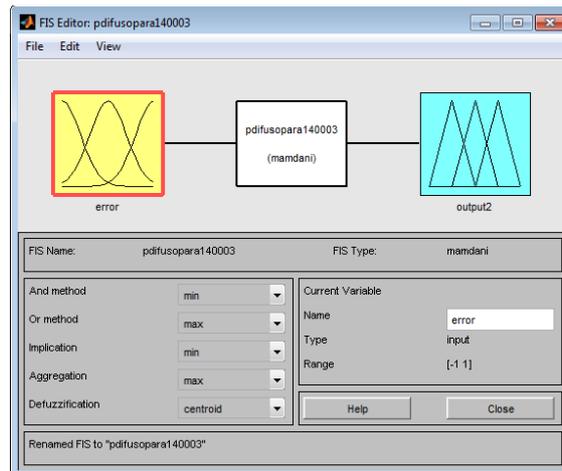


Fig. 4.26. Esquema general realizado en Simulink correspondiente a las funciones de Membresía de entrada y salida.

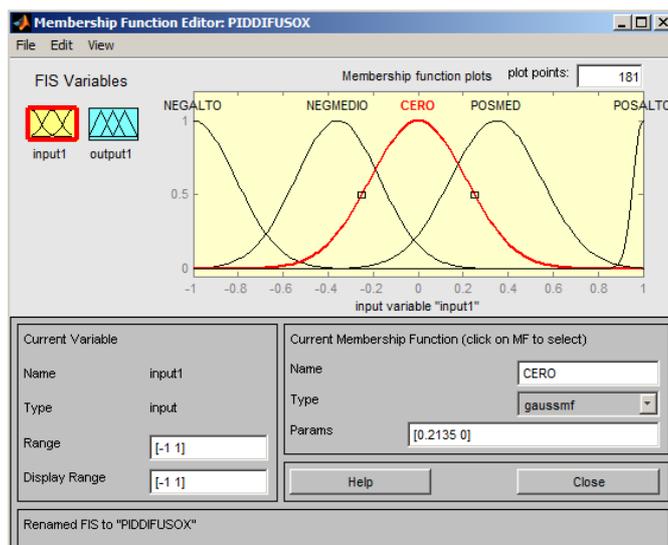


Fig. 4.27 . Funciones de Membresía de Entrada.

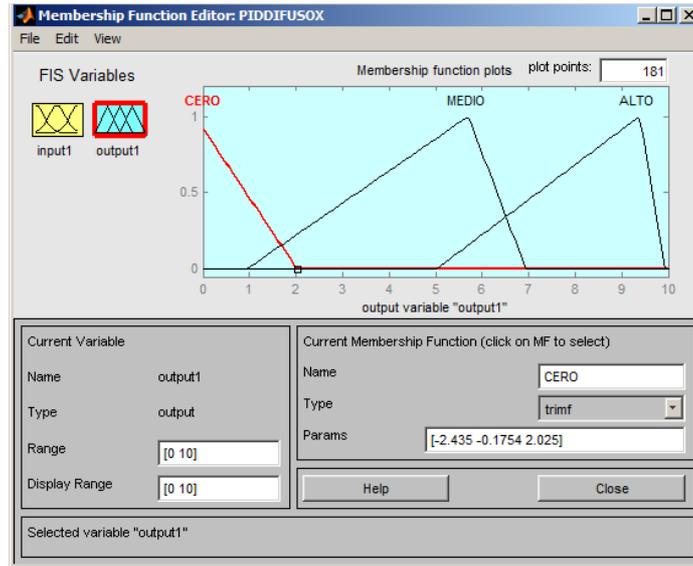


Fig. 4.28 . Funciones de Membresía de Salida.

El siguiente ejemplo, véase fig.(4.29), se basa en el concepto del PID Difuso por medio del Método Ziegler Nichols, tal como lo maneja el autor Ponce (2010), los valores de las ganancias dependerán de la experiencia que tenga el experto en área. Para poder utilizar el resultado obtenido nos muestran las gráficas tanto de la salida del producto de " $K_p * e$ " como de la Salida con respecto a la señal deseada véase fig. (4.30).

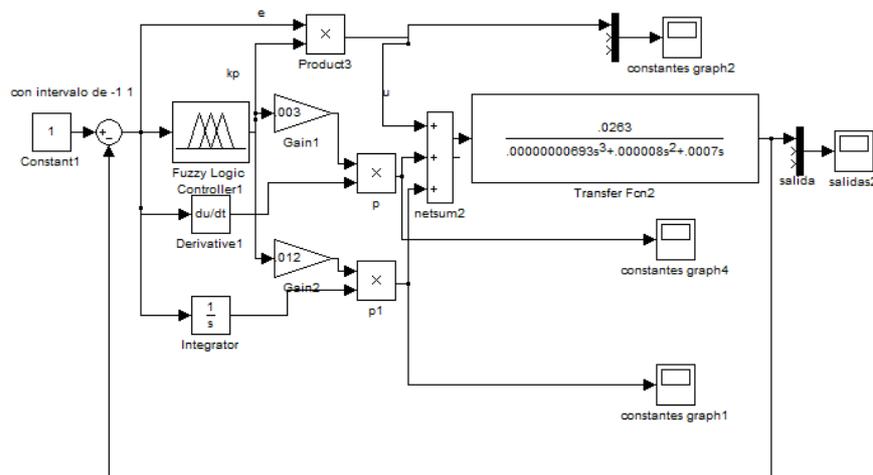


Fig. 4.29. Diagrama Esquemático del PID Difuso ya Sintonizado.

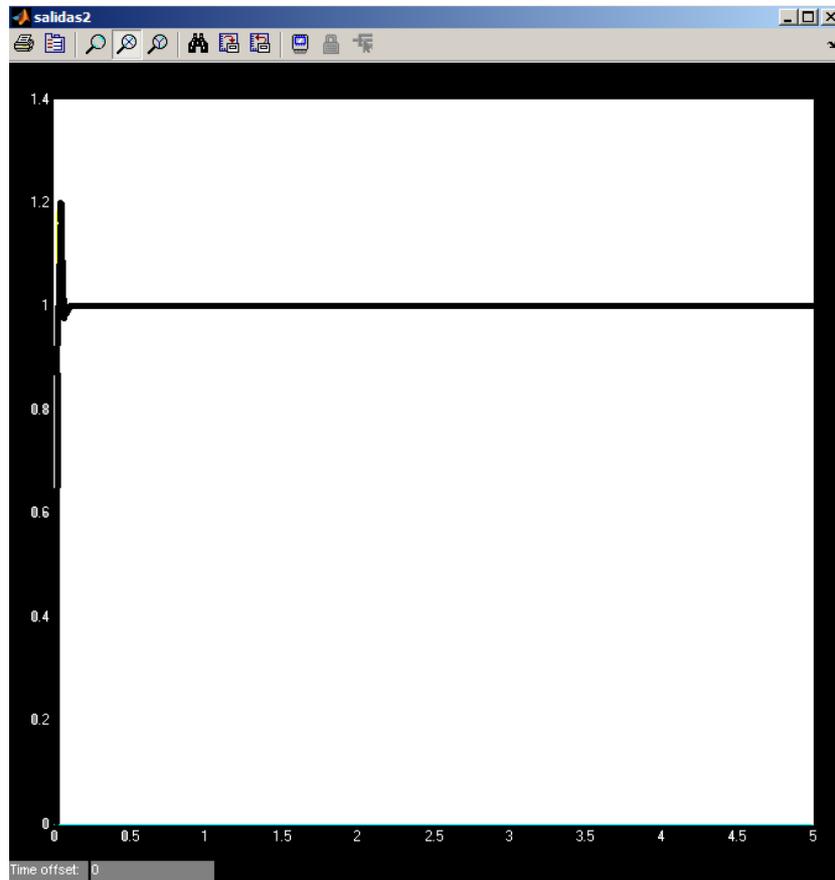


Fig. 4.30. Gráfica con el PID Difuso ya Sintonizado.

Al aumentar la salida del sistema a un rango más amplio se observa que este tarda más tiempo en resolverse lo que hace que sea ineficiente. Y no genera ningún cambio deseable. La respuesta del sistema puede depender directamente de las funciones de membresía de la entrada así como la ganancia de cada una de sus constantes (**Kd** y **Ki**). Sin embargo esto genera que el control se haga más complejo y dependa del punto de vista del experto para poder sintonizar adecuadamente su control.

4.1.6 Control de Posicionamiento PID en un Motor de Corriente directa por medio de LabVIEW.

Para la utilización del control PID Digital se Utilizaron algunos conceptos según Caruso (2010), en el cual se utiliza un sistema discreto para obtener el control PID de un Dispositivo por medio Embebido de la forma:

Ecuación 4.41

$$Spid[n] = Kpe[n] + Ki \sum_{k=0}^{n-1} e[n - k] - Kd(e[n] - e[n - 1])$$

El manual del An964 de Microchip (2004), "PID", lo expresa también de la forma:

Ecuación 4.42

$$c(t) = KpE(t) + Ki \int_0^t E(t)dt + KdE(t)/dt$$

Ecuación 4.43

$$\frac{dE(t)}{dt} \sim [E(n) - E(n - 1)]/Ts$$

Ecuación 4.44

$$dE(t)/dt \sim Ts \sum_0^N E(n)$$

Ecuación 4.45

$$c(n) = KpE(n) + KiTs \sum_0^N E(n) + Kd[E(n) - E(n - 1)]/Ts$$

$$c(n) = Kp(E(n) + \left(\frac{Ts}{Ti}\right) \sum_0^N E(n) + \left(\frac{Td}{Ts}\right) [E(n) - E(n - 1)])$$

$$Kp=K, Ki=K/Ti \text{ y } Kd=K TD$$

Un ciclo de Transcendental importancia es el que se encuentra en la figura (4.31), este está conformado por un bucle mientras (*while Loop*), así como un arreglo de tiempos los cuales hacen que pueda almacenar todos los datos de entrada en Excel, y después de almacenarlos continúe, pero sin grabar mas.

Otro punto importante de esto es que si se declaran tiempos largos; entonces consumiría demasiada memoria y lo importante es que solo arroje los datos necesarios para poder interpretar de forma real, el posicionamiento del Motor con respecto al tiempo de Ejecución.

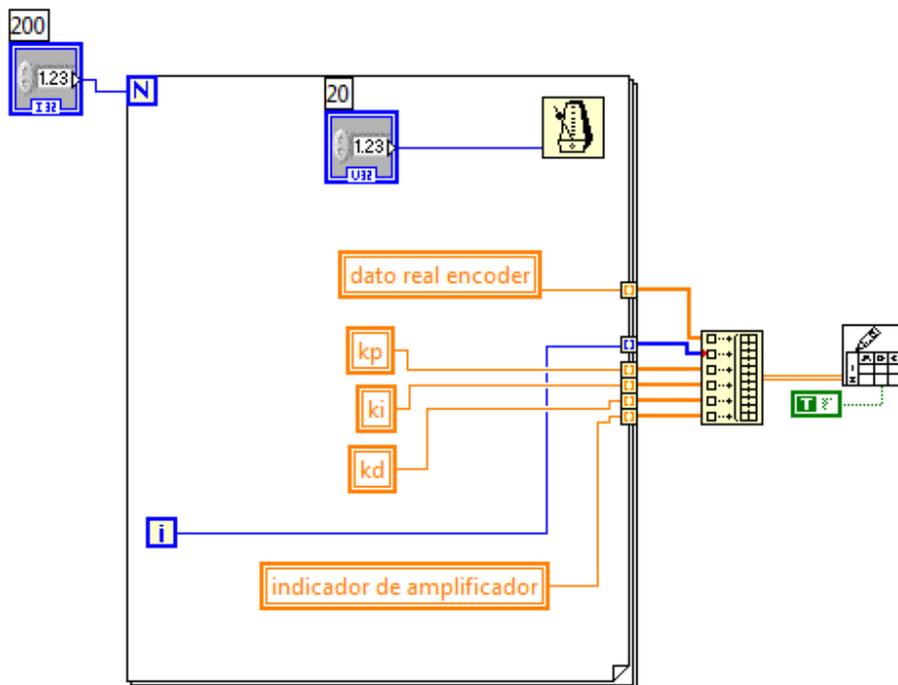


Fig. 4.31 For Loop con base de datos en Excel.

Dentro del Ciclo se encuentran las etiquetas nombradas para cada caso, es decir los datos que queremos analizar son: El dato real del *Encoder* el cual queremos monitorear para graficarlo con respecto al tiempo y saber cómo funciona su respuesta, Las constantes Kp,

Ki, y Kd, las cuales fueron presintonizadas con anterioridad, la i, la cual indica el tiempo que transcurre de 0 a infinito, y un indicador el cual determinara a la salida de la *Daq* si el signo es positivo o negativo véase fig. (4.32).

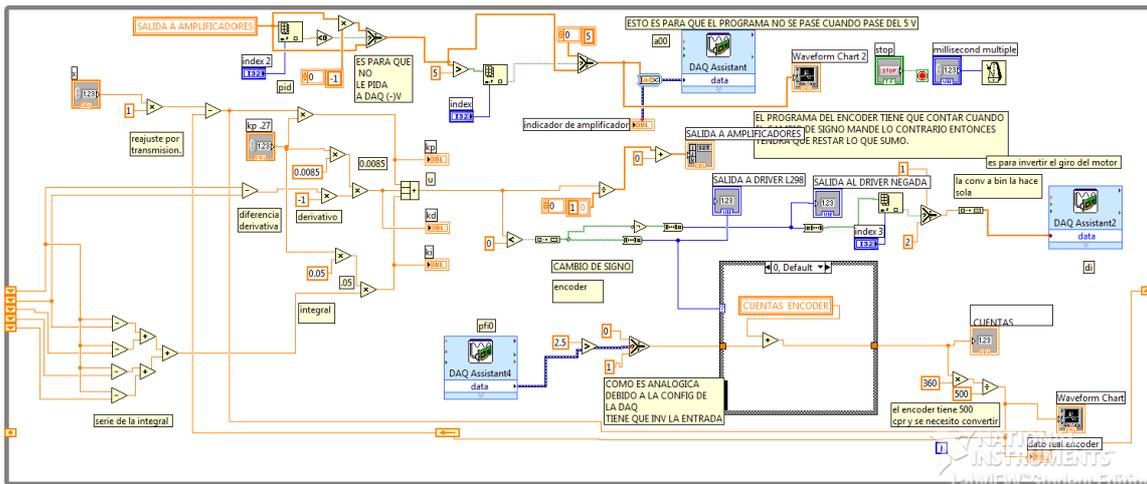


Fig. 4.32 Diagrama General de Control PID para el Posicionamiento de Motores para cada Articulación.

Una aplicación Importante del *LabVIEW* se llama *Shift Register*, y esta viene siendo un sinónimo, de lo que sería copiar un registro a otro; esto sirve cuando queremos hacer series principalmente en programas; o cuando necesitamos aplicar algún control en un microcontrolador o en algún dispositivo Embebido, en este caso fue utilizado para realizar el control PID, dentro del Software *LabVIEW*.

Para esto lo que se procede a realizar es que se obtiene una copia de registro de la retroalimentación de cada *Encoder*, de esta forma cada vez que hace una lectura la *Daq*, esos resultados se van concatenando uno a uno para poder formar sumatorias del valor que actúa en contra de los anteriores. El ejemplo de esto se puede mostrar en la figura (4.33).

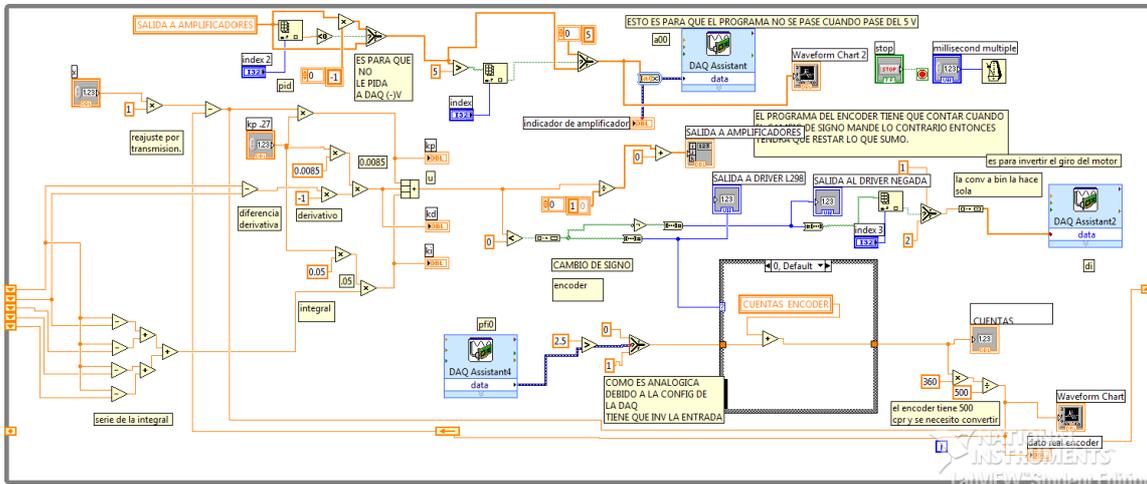


Fig. 4.33. Shift Register de While Loop

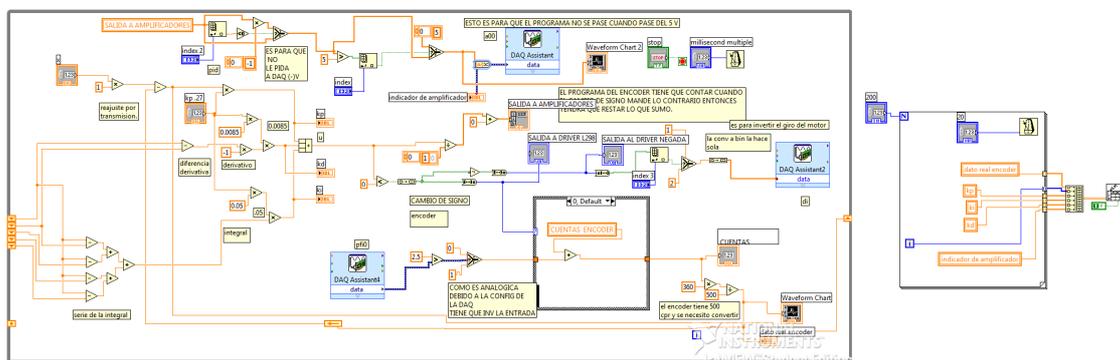


Fig. 4.34. Diagrama General de Control PID.

Cabe mencionar que del Diagrama General véase fig.(4.34) este solo puede ser simulado siempre y cuando se tenga la *Daq* conectada; de no ser así marcaría algunos errores de forma gráfica antes de ejecutarse. Por parte de las entradas y salidas de la *Daq* se representan en Bloques de color azul. La forma más sencilla para probar este tipo de controles es remplazando cada uno de esos bloques de entrada y salida por el indicador correspondiente. De esta forma se puede simular sin ningún inconveniente.

En la Figura (4.35), se puede observar parte del Panel Frontal en el cual se puede interpretar la respuesta del encoder en tiempo real, esto nos sirve para darnos una idea al momento de simulación para saber si el control está haciendo lo correcto o no.

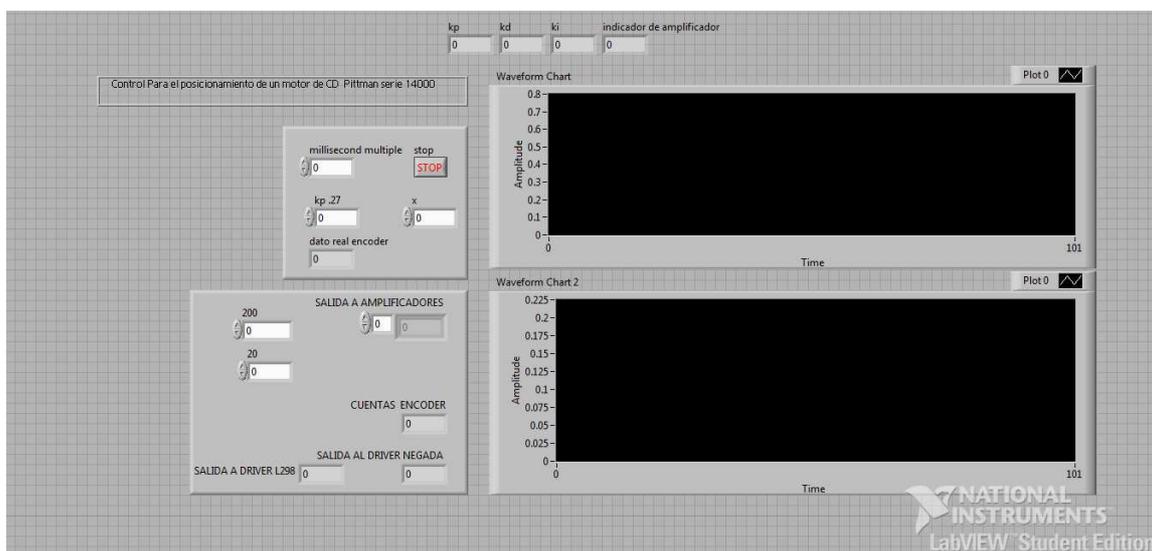


Fig. 4.35. Panel Frontal Final del diagrama de control con almacenamiento de datos en Excel.

Cuando se ejecuta el Programa por primera vez se necesita mandarle al mismo algunos ángulos de forma aleatoria o si se gusta de forma secuencial, es decir se debe de mandar datos repetidos, esto con el fin de que LabVIEW almacenen todos esos puntos y posteriormente pida la ruta donde almacenarlos, con esto podemos nosotros graficar cada uno de los resultados obtenidos y tener una interpretación mejor de los resultados véase fig. (4.36).

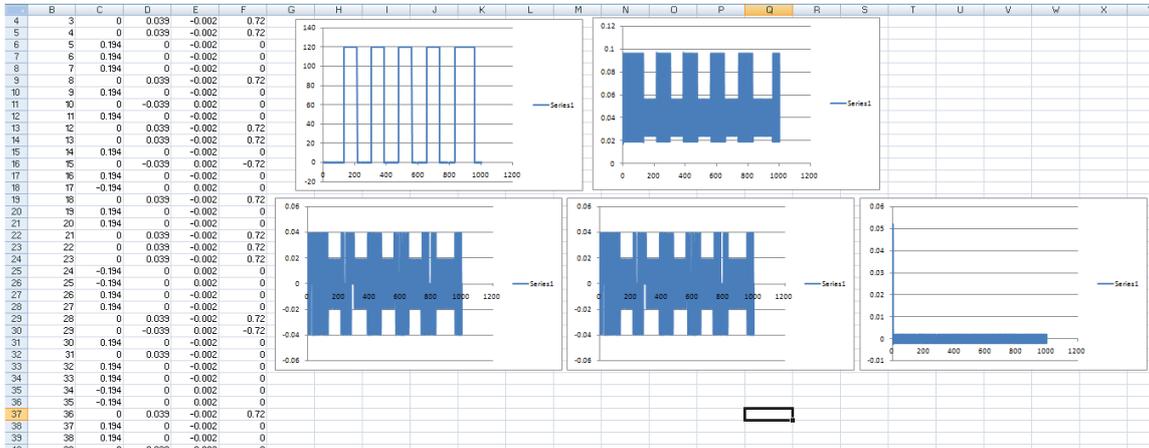


Fig. 4.36. Resultados en hojas de datos de Excel.

4.1.7 Resultados experimentales

4.1.7.1 Interfaz de Comunicación Individual entre LABVIEW y los Motores Pittman 14000.

La primera parte del prototipo para el control de posicionamiento de un motor Pittman serie 14000, se basa en la configuración para la entrada de un encoder de 512 cuentas por revolución "counter per revolution", véase fig. (4.38), que esta acoplado a la flecha del motor y va a mandar señales a la tarjeta de adquisición de datos *Daq*. El objetivo es poder diseñar la interfaz que pueda servir de comunicación entre la *Daq* y cada Motor de Dc.



Fig. 4.37. Encoder de 512 CPR mostrada un una caja diferente con el sensor óptico HEDS-9100.

El Dispositivo que utiliza el motorreductor para su funcionamiento es un HEDS-9100 dispositivo óptico para poder contar cada una de las ranuras del disco véase fig. (4.37).



Fig. 4.38. Imagen del motor Pittman serie 14000.

El disco contiene 512 ranuras por revolución, por este motivo es necesario realizar una regla de tres con la cual se pueda convertir las 512 ranuras entre los 360 grados por revolución. Véase fig. (4.39). Esto es importante pues de esto también dependerá del ángulo y precisión deseado, sin olvidar que además cuenta con una transmisión 5.9 a 1.

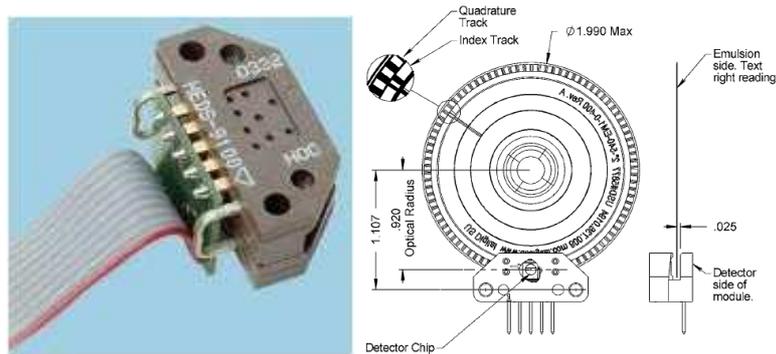


Fig. 4.39. Diagrama esquemático del posicionamiento del sensor y la adaptación con el Molex.

Los pines del dispositivo desde vista frontal para Heds 9100 son:

- 1.-Tierra
- 2.-No conectar
- 3.-Canal 1
- 4.-Vcc
- 5.-Canal B

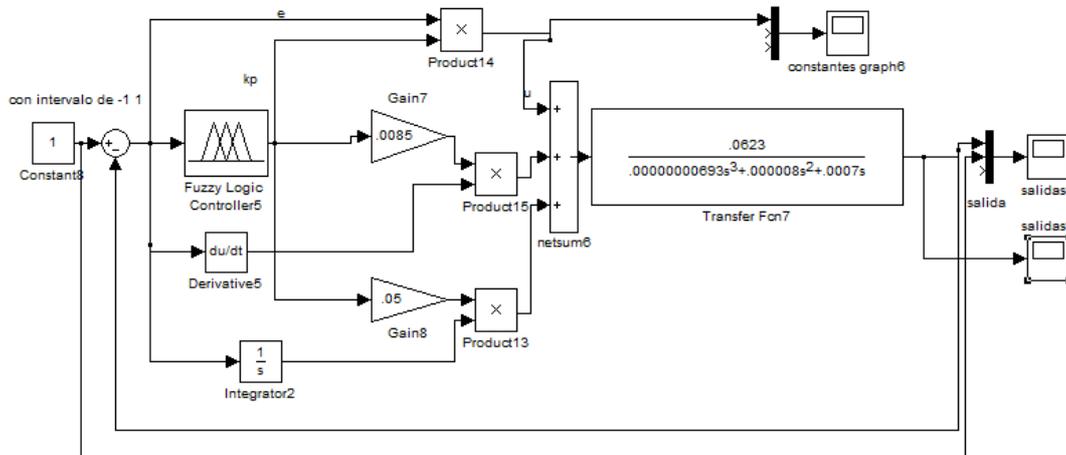


Fig. 4.40. Diagrama de control del Motor Pittman 14000 con un controlador PID difuso.

Como se puede mostrar en la figura nosotros representamos un control por medio del programa *Simulink* véase fig. (4.40), en el cual se utiliza una función de transferencia representa por nuestro motor de Cd y su respectivo *encoder*. El objetivo es poder reproducir este sistema pero ahora físico, por lo cual se sustituirá la programación realizada en Matlab y se agregara en el programa *LABVIEW* el cual ya se tenía código almacenado para la elaboración del robot de 4 G.D.L.



Fig. 4.41. Imagen de la Interfaz de comunicación usb-6009.

Para el desarrollo hay dos posibilidades de mandar el voltaje de salida de la *Daq* a el motor, Debido a que la *Daq* solo puede sacar de 0 a 5 volts en sus salidas se puede diseñar un arreglo analógico para que le suministre la entrada (u) de control al Motor o por medio de *PWM*. En este momento se optara por el arreglo de amplificadores por simplicidad aunque posteriormente se hagan otros análisis.

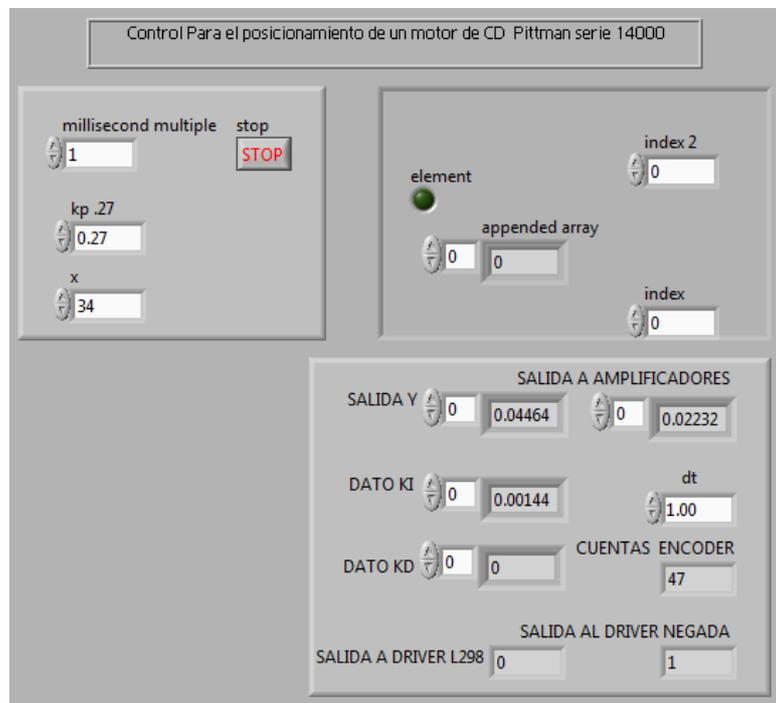


Fig. 4.42. Panel Frontal del Programa elaborado en LABVIEW.

En la siguiente figura (4.43) se puede mostrar el esquema general para la etapa de transmisión de datos por medio de LABVIEW para el motorreductor PITTMAN.

Por medio de este diagrama el cual fue encerrado por medio de un ciclo *while* este se encarga de realizar la función de seguir monitoreando el sistema hasta que se le finalice a tal proceso. De tal forma se puede ir actualizando la información de entrada y se puede ir retroalimentando al sistema.

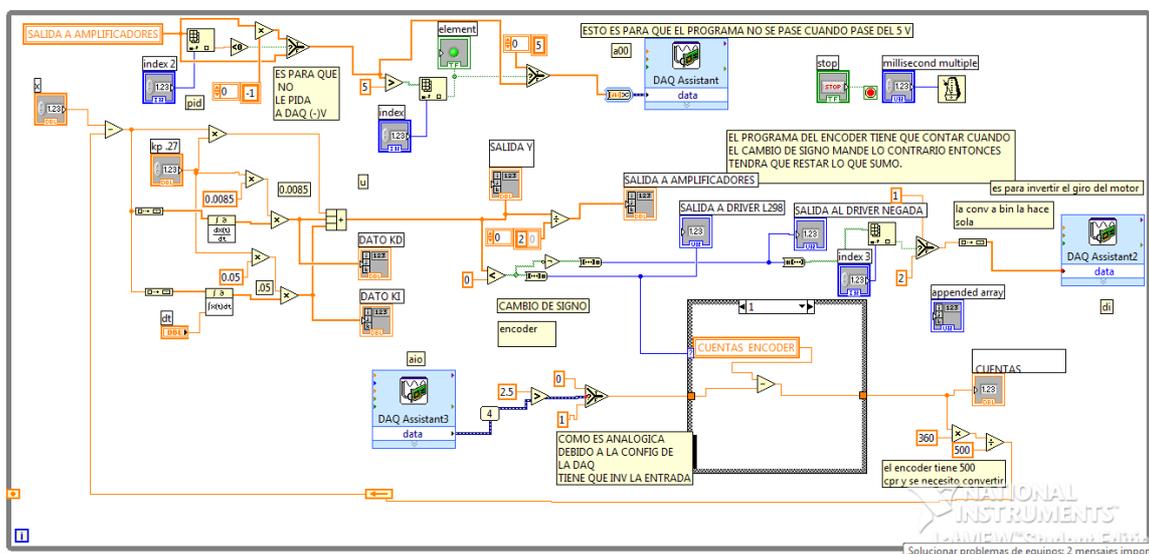


Fig. 4.43. Diagrama de Bloques de control Proporcional.

Para empezar uno de los principios a considerar es que si no se genera un proceso de sincronización adecuado el dispositivo que monitorea la señal por medio del giro del motor sería directamente proporcional al tiempo en el que pudiera leer las muestras y se podría adelantar o retrasar según el caso.



Fig. 4.44. Bloque de paro y Bloque tiempo de LABVIEW.

El primer bloque se considera por default pues este debe venir implícito con cada ciclo que se desee realizar y esto debido a que cada ciclo debe contener un botón de paro en caso de algún error véase fig. (4.44). El segundo bloque que además se debe tener en cuenta es el bloque de "WAIT UNTIL NEXT ms MULTIPLE", pues como similitud a los microcontroladores este debe de trabajar a una determinada velocidad y para esto se debe de considerar el tiempo en el que va a trabajar para evitar problemas de sincronización en el caso de *LabView* trabaja en mSeg.

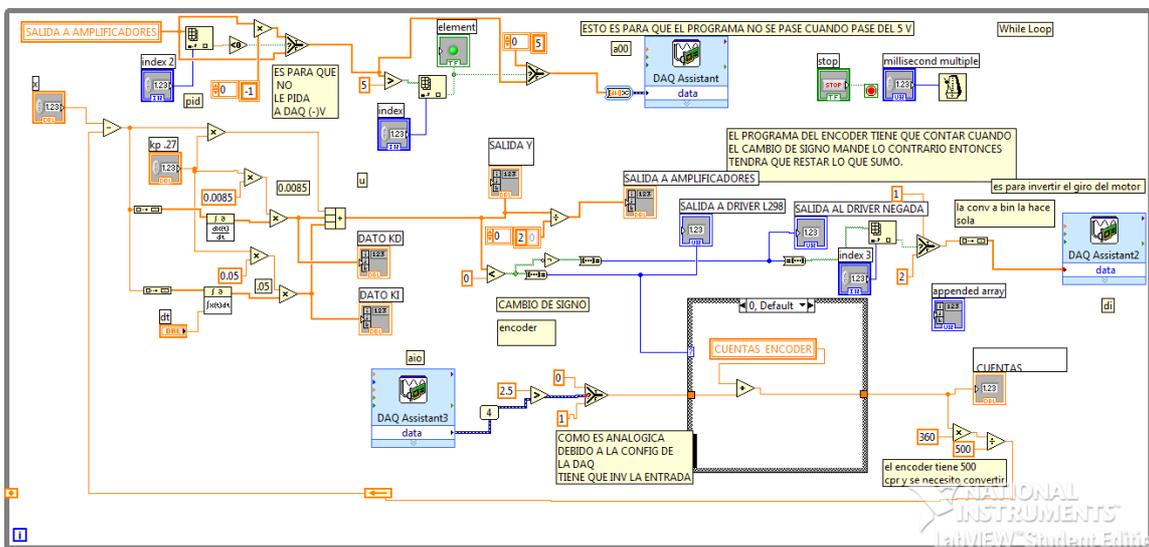


Fig. 4.45. Diagrama Demostrativo de las salidas (di) parte superior derecha y entrada (ai0) procedente del encoder.

Por otra parte la etiqueta (di) véase la fig. (4.45), indica la salida del puerto digital 1 que se encarga de hacer el cambio de giro del motor. Mientras que el ai0 es la entrada que proviene del *encoder*. Además nos podemos dar cuenta que la salida del HEDS-9100 es de 4 volts aproximadamente mientras que se compara con una constante "2.5" y de esto va a ir sumando números siempre y cuando el comparador de la salida al driver L298 no indique lo contrario; pues si es así entonces le empezaría a restar datos ya que indicaría que el motor sobrepaso el ángulo deseado.

Como similitud el diagrama realizado por medio de SIMULINK se elaboro un programa que simulara el mismo proceso por medio de LABVIEW para poder comunicar al ordenador con cada uno de los motores. El primer paso que se muestra es la entrada

llamada “x”, la cual es la variable deseada mientras que para esto se agregaron las constantes previamente sintonizadas. Del control PID realizado con anterioridad, la salida se va a un indicador llamado Salida a Amplificadores, que es otra etapa del programa, donde cabe mencionar que se realizan los ajustes necesarios pues se debe de hacer una escala que identifique el voltaje a suministrar al motor en físico ya que la salida de la *Daq* de LABVIEW es de 5 Volts.

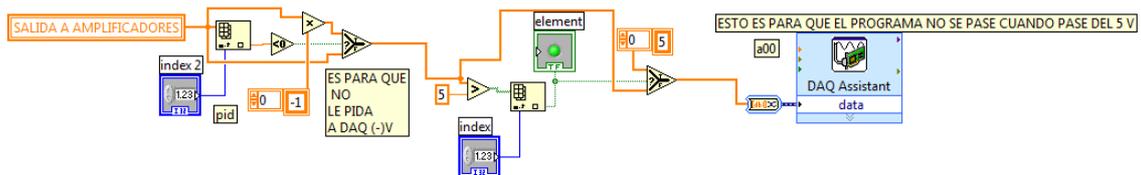


Fig. 4.46. Diagrama de salida del control PID por medio del puerto (a00) de la *Daq* usb-6009.

Esta segunda etapa, véase fig. (4.46), recibe el valor de salida de control PID y compara si es negativa o no. En el caso de ser negativa la pasara a positiva pues una de las peculiaridades de LabView es que a la salida Analógica no acepta que se le envíe voltaje negativo con esto se evita que el programa se pause y que además se pueda distinguir por otros bits cuando el motor debe cambiar de giro. Otra peculiaridad es que cuando llegue a un valor máximo; si se le agrega mal un dato que rebase 360 grados este no suministrara mas voltaje ni marcara algún error por parte de la configuración del motor véase fig. (4.46).

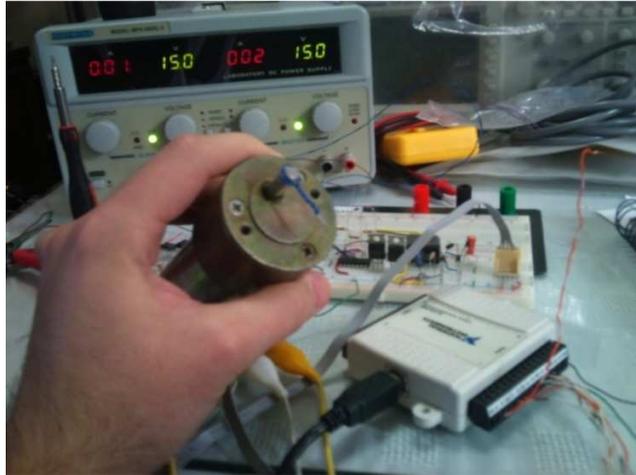


Fig. 4.47. Interfaz de Comunicación y posicionamiento del motor.

Como se puede mostrar en la figura (4.48), tenemos el diagrama general de cada esclavo realizado en el software de *National Instruments* como anterior se podrá haber notado no se utilizaron en particular "*Toolbox de National Instruments*" esto con el fin de que todo lo que se ha realizado en LabVIEW se puede reutilizar en cualquier microcontrolador o dispositivo embebido, sin embargo la capacidad que tiene el equipo de LabVIEW facilita mucho las cosas, es decir el equipo y software de LabVIEW marca la posibilidad de poder simular el circuito de forma sencilla, graficarlo, e interfazarlo con diferentes ambientes tales como los son *SIMULINK*, *C* y *MATLAB* sin contar que a la salida hay formas inimaginables de comunicarlo con dispositivos embebidos, microcontroladores e instrumentos de medición, Así como las bases de datos y archivos que se pueden almacenar en programas como *EXCEL Y WORD*.

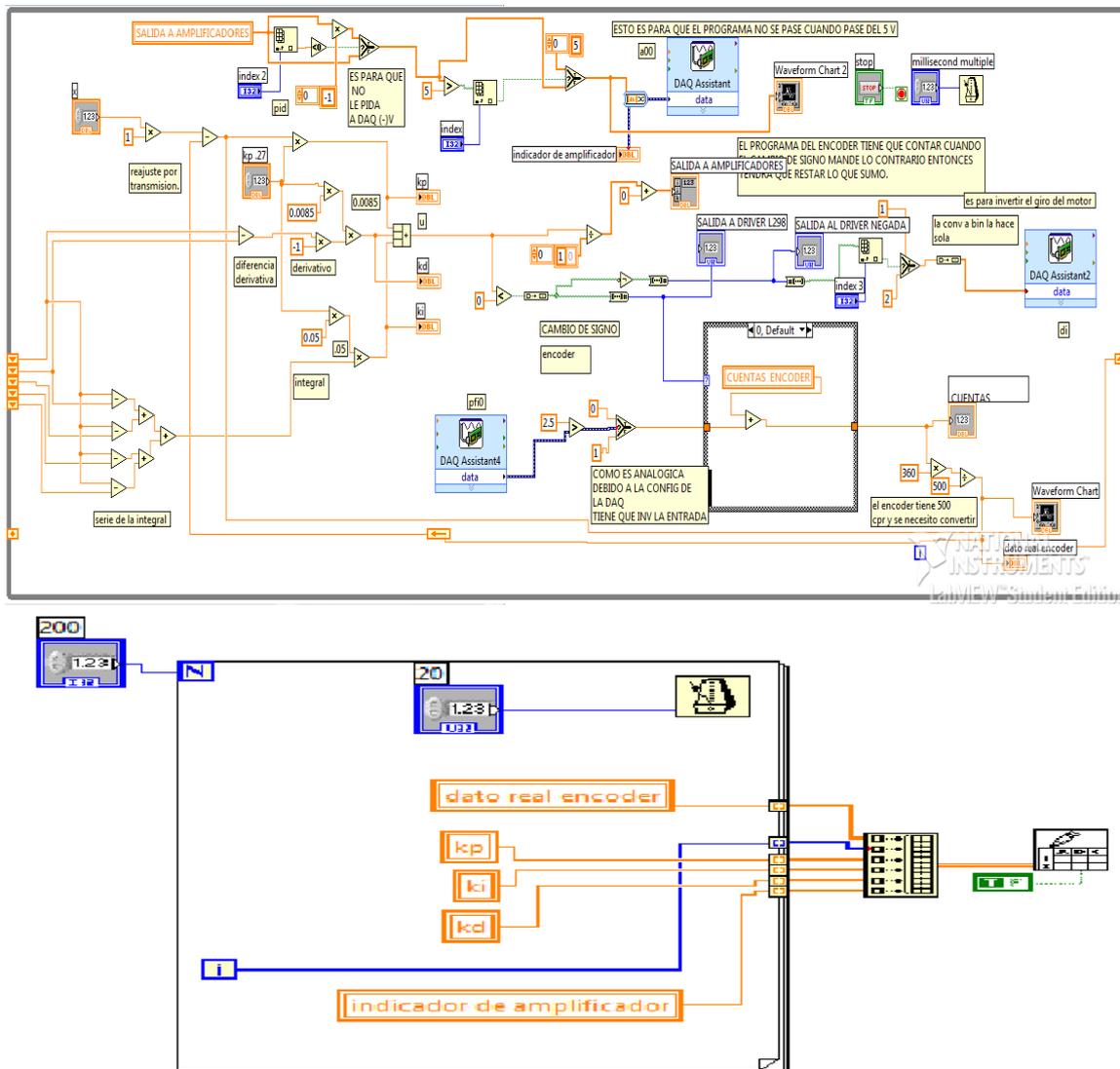


Fig. 4.48. Diagrama de control General para motor Pittman 14000.

En la figura (4.48) de la segunda imagen se realizó una rutina para poder saber si el control está haciendo su función y para esto se tomaron 1000 muestras la cual cada una tardaba 20 milisegundos, con lo que se obtuvo la siguiente gráfica en Excel, la rutina que muestra fueron cero grados, posteriormente 120 grados, se regreso a cero grados, 60 grados, 120 grados, 60 grados, 0 grados y 60 grados. Una vez que se termino esta rutina se puede observar que el control está funcionando de manera aceptable y cabe mencionar que aquí el tiempo ejecutado solo toma importancia cuando hablamos de muestreo es decir el tiempo

que tarda en dar una lectura a otra, ya cuando hablamos de interpretación de los resultados, podemos esperar de forma directa el posicionamiento como lo muestra la figura (4.49), esta denota que el tiempo en llegar al resultado es mínimo, no tiene sobreimpulso y el valor se repite de forma considerable es decir no existe demasiada variación en el momento de calcular una serie de rutinas. En otras palabras el dato no se pierde, por algún problema de sincronización con encoder o cualquier otro factor físico o Lógico.

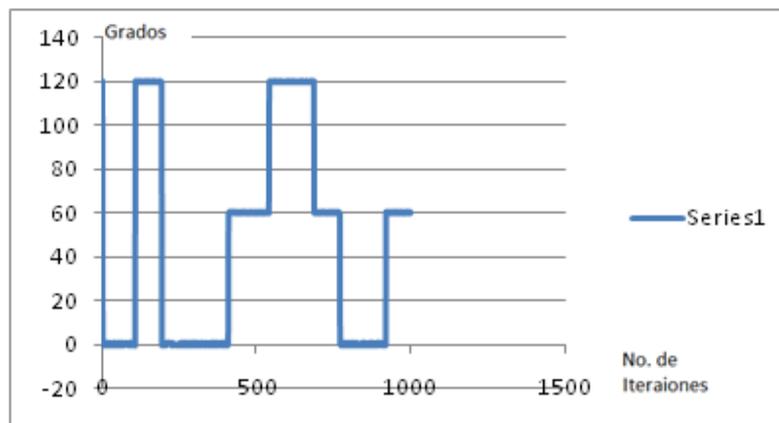


Fig. 4.49. Posicionamiento de un motor en diferentes ángulos.

Otra cosa importante que cabe mencionarse es que parte de lo que se a podido aprender del posicionamiento de Motores, se debe a que el factor K_p es de gran importancia para poder realizar el control del mismo , sin embargo así como realiza un papel para determinar que tan rápido puede llegar a estabilizar un control, otro dispositivo que se encarga de posicionar al motor es el encoder, pues este dispositivo es de gran relevancia en la precisión, y sin este ningun control podría desempeñar una función correctamente con un Motor. El Encoder utilizado para esto fue por medio de un disco de 512 *CPR*. Es decir que cada división en la cual cuenta el encoder corresponde a menos de un grado de posicionamiento.

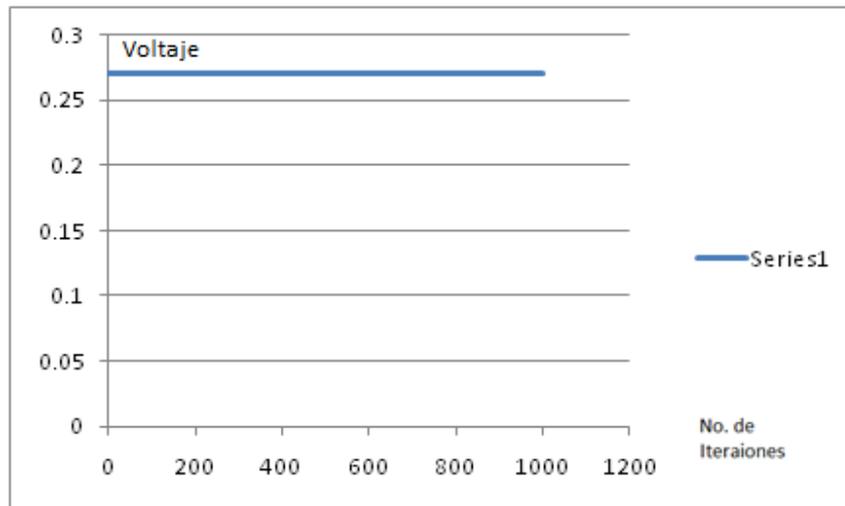


Fig. 4.50. Control Proporcional de salida en un sistema de Dc.

En las Figuras (4.50), (4.51) y (4.52) se puede observar las constantes registradas por medio del programa en *LabView* y se puede mostrar que en estas no existe ninguna variación. Por otro lado la Simulación forma parte importante en la experimentación pues de esta forma se puede conocer el valor de las constantes Proporcional, Integral y Derivativa del control Físico.

También otra forma para optimizar la implementación de un control PID Difuso, es sustituyendo el valor de las Constantes obtenidas por medio de la simulación y experimentarlo, seguramente se tendría que hacer un pequeño ajuste por algún circunstancia física pero idealmente no debería haber mayor complicación.

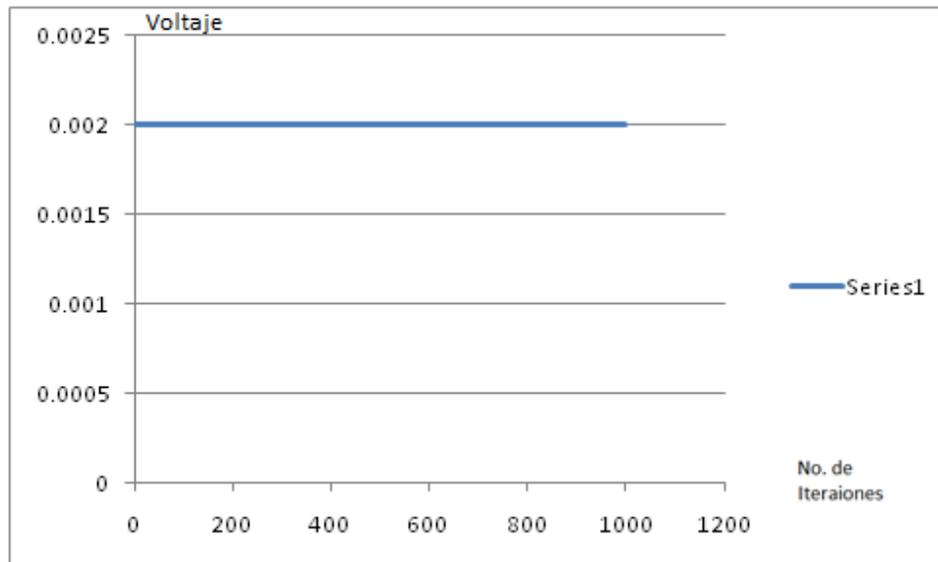


Fig. 4.51. Control Integral.

Otro comentario que cabe remarcar en todo esto es que como se había comentado al final de algunos capítulos, el objeto de la utilización de este software fue con la finalidad de poder realizar programación la cual pudiera ser fácil de comprender de manera visual y que esta pueda servir posteriormente para poder realizar un trabajo con mayor calidad, en el cual se pudiera desprenderse de un software de programación en especial como lo es *MATLAB, SIMULINK, O LABVIEW* y pasar a un dispositivo embebido el cual pudiera realizar lo mismo, a un bajo costo y sin problemas de entendimiento, pues todos los diagramas mostrados fueron pensados con el fin de poder ser programados en cualquier tipo de microcontrolador y se trataron de evitar aquellos bloques o paquetería de las mismas empresas que venden software para no depender de forma directa de todos ellos.

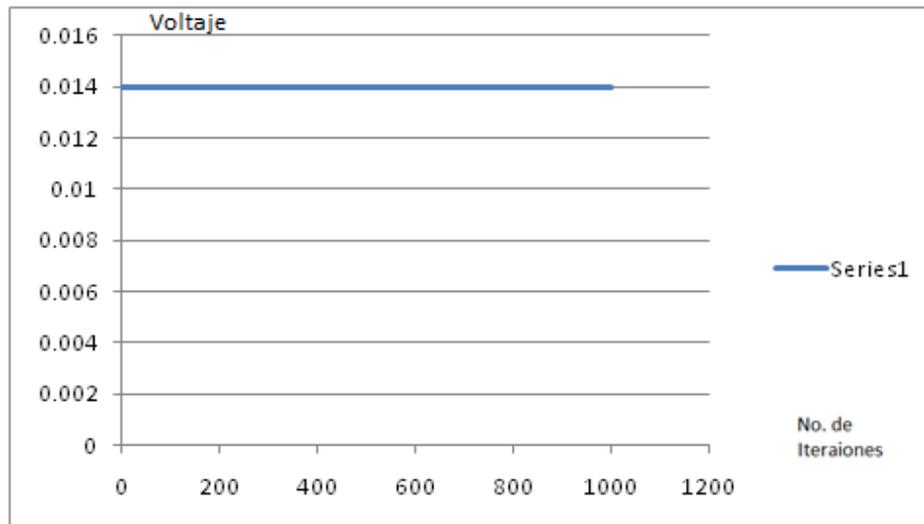


Fig. 4.52. Control Derivativo.

4.1 Experimentación

Para poder implementar la experimentación del motor se procedió a realizar una experimentación por medio del mismo programa que se había comentado con anterioridad. El primer paso a realizar fue interfazar al motor Pittman por medio de la *Daq*.

Una vez conectados se procedió a controlar el cambio de giro por medio de un driver, que funciona como puente H. los resultados obtenidos que arrojo el Encoder fueron almacenados en un hoja de Cálculo como se puede mostrar en la figura (4.53).

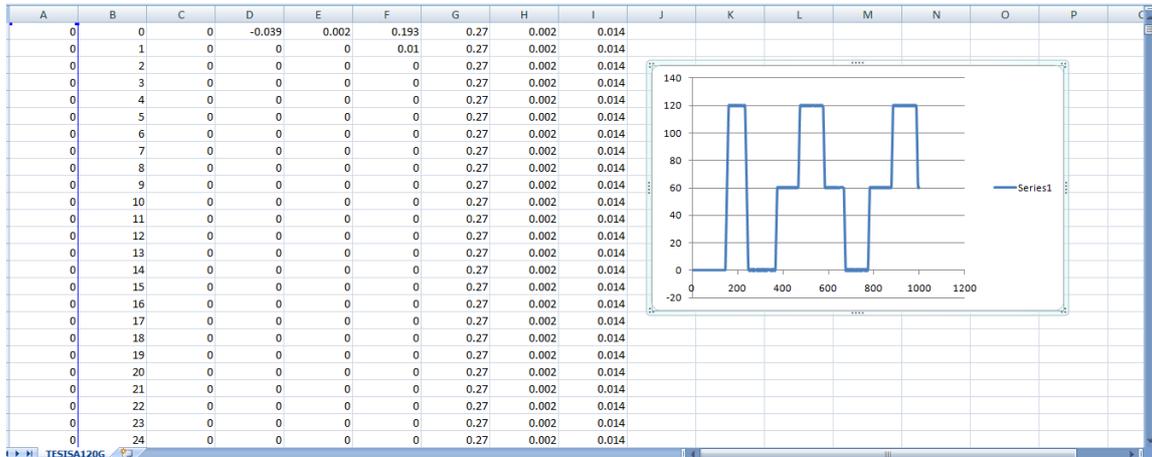


Fig. 4.53 Base de Datos obtenidas en Excel.

Al igual que el caso de la simulación se procedió a realizar la misma rutina, todo esto para poder determinar si el valor real se parece al de la simulación. La gráfica Fisca arrojó los valores que se aprecian en la figura (4.54). La cual como se puede apreciar se realizaron 1000 muestras de 20 milisegundos cada una.

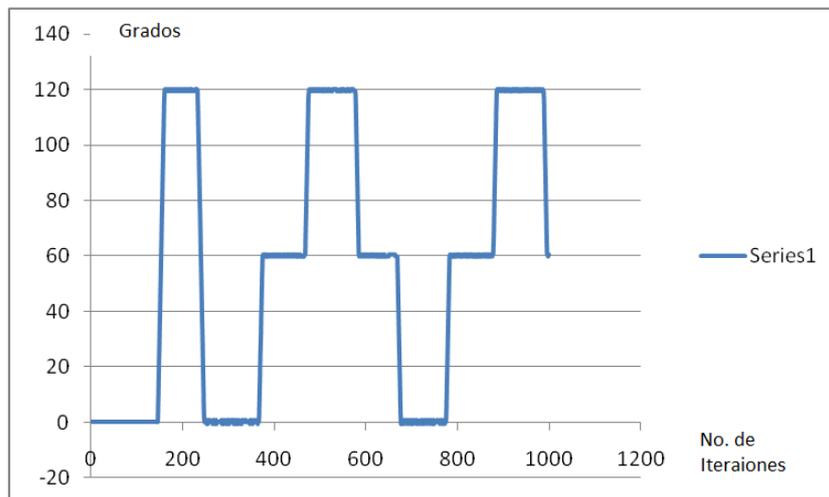


Fig. 4.54 Control de Posición en Tiempo Real.

Los datos obtenidos por la gráfica (4.54) muestran que el control no genera gran variación y mantienen su precisión. En la figura (4.55) se puede mostrar las 3 gráficas obtenidas que representan a cada una de las constantes para la utilización del PID.

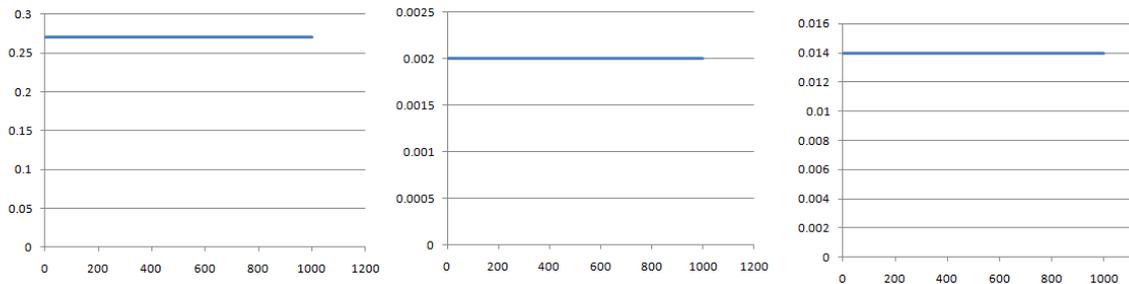


Fig. 4.55 Constante Proporcional, Integral y Derivativa.

4.2 Control PID del prototipo

4.2.1 Diseño del control

Como se puede mostrar en la figura (4.56), tenemos un diagrama general parecido al utilizado en la experimentación del motor industrial. Y como tal cada bloque singular de este va a controlar a cada esclavo que solo se encarga de recibir del valor que se les proporciona y hacer la modulación de ancho de pulso pertinente. Pero todos ellos monitoreados desde el software de *National Instruments*.

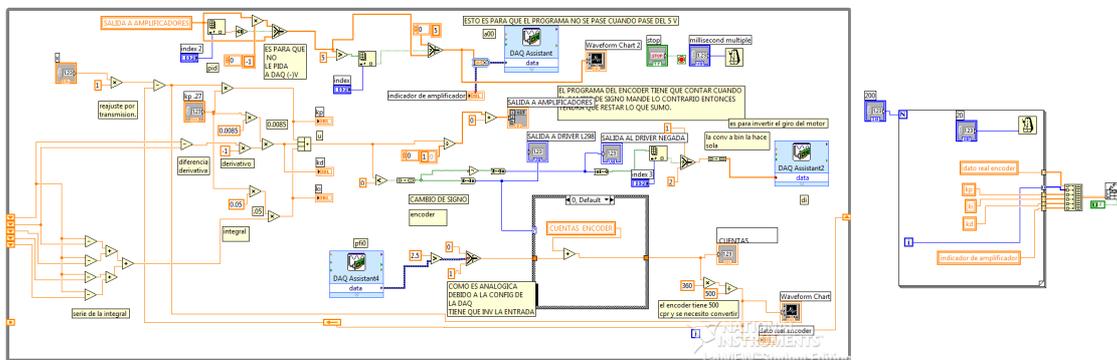


Fig. 4.56. Diagrama de control General para motor Pittman 14000.

El proceso que realiza la Interfaz va desde el momento en que el Programa en LabView resuelve el control pertinente. Posteriormente en ayuda de la *Daq 6008-6009* le envía los datos realizados junto con los algoritmos presentados por las salidas Analógico-Digitales a los Buffers los cuales son utilizados para ampliar el voltaje y proteger a la entrada, así como se cerciora de amplificar la señal con puentes H de la salida de la *Daq* porque esta saca muy poca corriente de salida, después de eso estos mandan la señal amplificada a los microcontroladores *DsPic 30f3014P*, con esto estos se disponen a realizar las operaciones pertinentes con el fin de enviarle el PWM deseado a cada articulación. A la salida de cada señal se reenvía una vez más a un sensor optoelectrónico el cual recibe la señal la vuelve a amplificar y al mismo tiempo aísla a los microcontroladores de la etapa de Potencia del Servomotor.

4.2.2 Resultados experimentales

Como se puede observar en el anexo (2) programa (3) la estructura con la que se elaboró el código para cada uno de los 4 microcontroladores tuvo el fin de poderle darle más accesibilidad a cada una de las articulaciones, tener la posibilidad de comunicarse con otros dispositivos, generar la mejor comunicación entre cada unos de los microcontroladores y además darle la posibilidad de ampliar el código anexándole un control Interno para cada una de las posiciones, es decir darle la posibilidad al microcontrolador Dspic 30f3014p la versatilidad de generar otras aplicaciones especiales con las cuales serían imposibles de realizar con un solo Dspic.

Una de las razones importantes por las cuales se opto por la implementación del Dspic a diferencia de otros tipos de microcontroladores comúnmente utilizados fue porque el Dspic cubre con amplio repertorio de instrucciones que no cualquier modelo de microcontroladores puede ofrecer, además de esto cuenta con la posibilidad de hacer operaciones aritméticas, lo cual reduce de forma considerable nuestro trabajo, puesto que necesitábamos que cada servomotor tuviera un comportamiento especial en el cual tuviera una restricción de 90 grados por cada lado pero que este se pudiera posicionar en base signos positivos y negativos.

La utilización del DsPic ha dado la pauta para configurar una plataforma la cual nos ha ayudado de forma considerable para la manipulación de datos oportuna y con posibilidades de aplicarla a otras interfaces como lo es el caso de *LabView* ver 8.2.

4.2.3 ¿Qué procedimiento es necesario para poder controlar cada articulación?

Para poder controlar cada articulación del robot fue necesario hacer cálculos entre el Programa utilizado que en este caso es el Stop Watch de Mplab y el tiempo en que tarda en llegar a 180 grados el Servomecanismo, para esto una vez que se calcula con una aproximación un valor que es referente al cero posicional del servomotor se procede a realizar una tabla que describa el comportamiento de esta, grado por grado. Es decir:

GRADOS SERVOMOTOR	DECIMAL
0	1000
45	1855
90	2711
180	4423

(Dato reajustado adimensionalmente).

GRADOS	DATO DECIMAL DS PIC
10	$1000+200=1200$
1	$1000+(20)=1020$

Tabla 4.6.Tabla PWM

El procedimiento es el siguiente: se calcula el valor aproximado hasta que llegue a cero grados este aproximadamente es igual a 1000, de aquí se propone un valor grande para que llegue a dar la vuelta completa cada servomotor una vez que llega exactamente a 180 grados se para el programa. De esta forma por medio del Stop Watch podemos determinar cuánto tiempo tardo en ciclo de instrucciones llegar hasta 180 de ahí se procede a realizar una gráfica véase fig. (4.57). y el resultado de la se puede observar de la siguiente forma: Para esto se compila de Mplab Sim véase fig. (4.58), también se abre el Stop Watch si se le asigna un marcador al inicio, posteriormente se ejecuta el programa y se vuelve a borrar el

Stop Watch, de ahí se vuelve a ejecutar el programa con las correcciones correspondientes en físico, una vez que llega a los 180 grados se para el programa y se revisa el Stop Watch el número de ciclos de trabajo que realizó seguramente van a aparecer en hexadecimales ese valor se pasa a decimal y ahora es el valor que se asigna como 180 grados. Si el valor entre cero y 180 llega a quedar desalineado por alguna razón se puede reajustar de forma fácil lo único que hay que hacer es que después de que haga el bucle referente al tiempo en que se realiza cada instrucción o el valor ya asignado, se le agrega a este un valor decimal extra para que se mueva el ángulo que le hace falta o en su defecto si el cero no lo da exactamente lo que se puede hacer es que del valor inicial calculado que era aproximadamente cero igual a 1000, le agregamos valores decimales equivalente a 1 grado, así de un grado a un grado podemos llegar al cero natural, En otros capítulos se puede observar que este mismo cero es recorrido para propósitos de diseño.

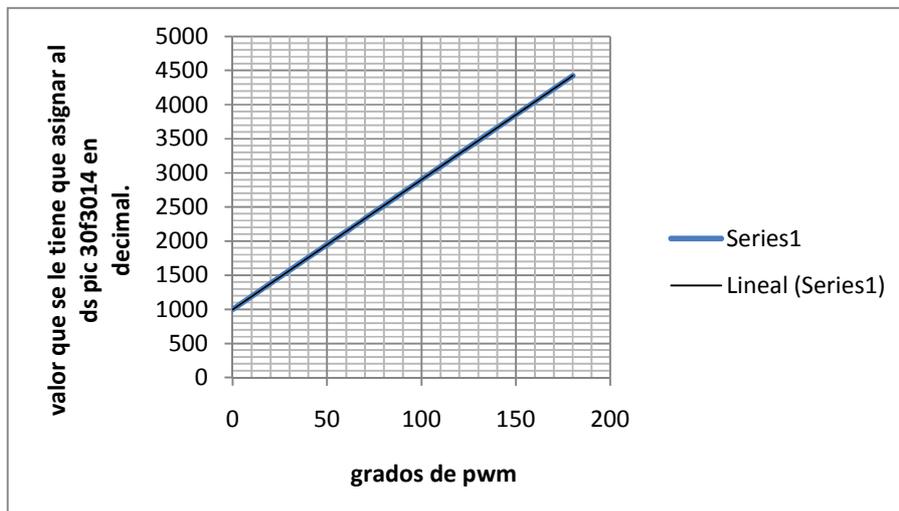


Fig. 4.57. Este procedimiento se debe de probar para cada uno del servomotor correspondiente puesto que muchas veces la modulación de ancho de pulso llega a variar, inclusive en modelos de la misma compañía.

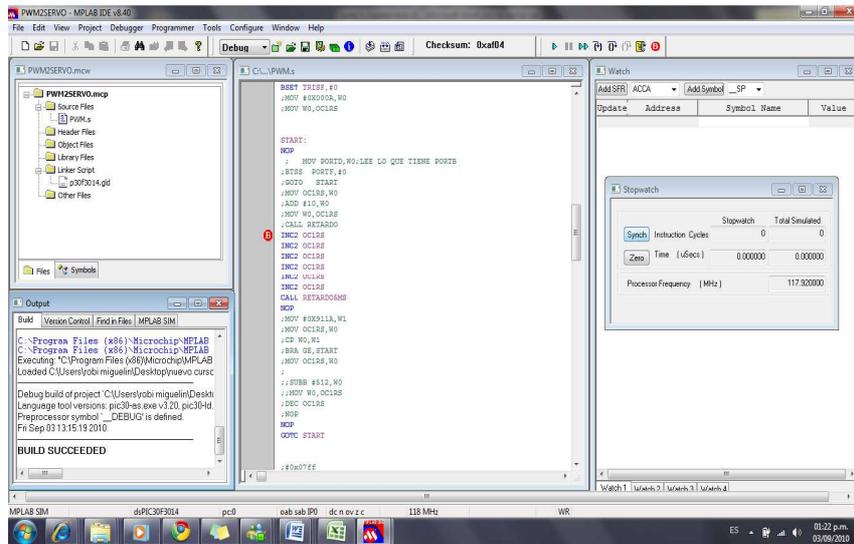


Fig. 4.58. Programa MPLAB de Microchip con Stop Watch.

En la imagen que se muestra podemos observar el programa prueba que sirve para calcular el grado en cada servomotor. A continuación se mostrara el código presentado para poder hacer el cálculo en cada uno de los servomotores véase fig. (4.59).

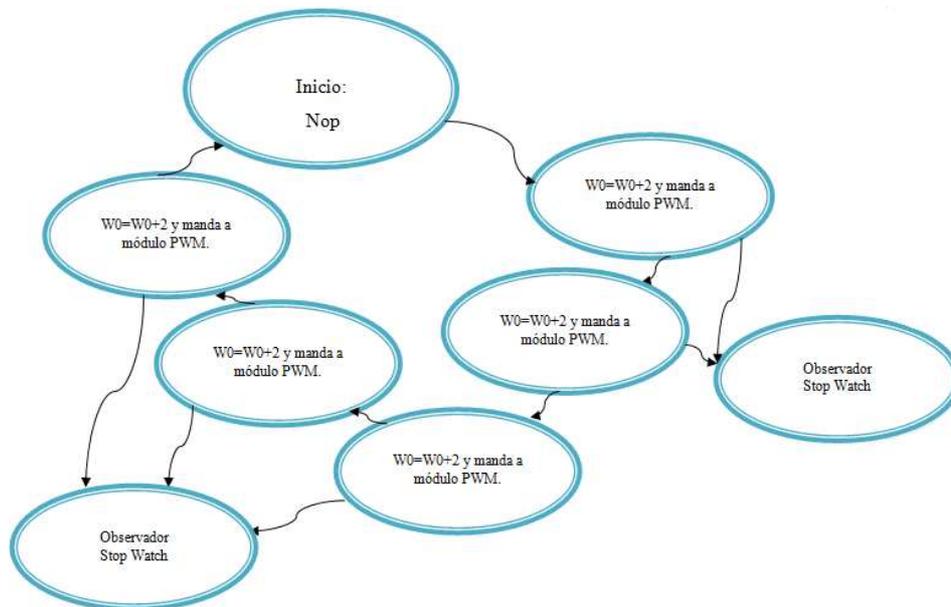


Fig. 4.59. Máquina de Estado de la forma de Calibración de un Servomotor.

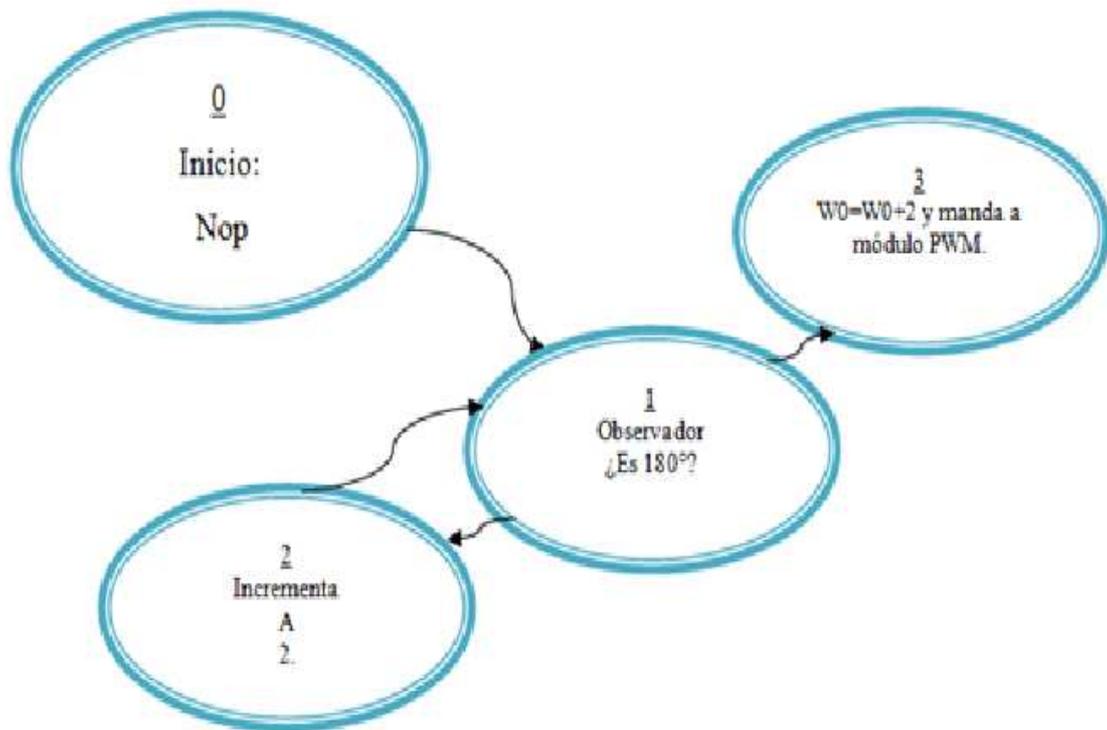


Fig. 4.60 Máquina de Estado.

En la fig. (4.60), se representa a una de las máquinas de estado, el primer paso lo representa el número 0, el cual tiene una instrucción Nop, Esta sirve para consumir un ciclo máquina, la siguiente instrucción es un observador el cual se encarga de comparar si el valor de entrada es 180, si no lo es incrementa el registro a 2, hasta que este iguala a los 180 grados lo manda a salida y le incremente un 2 como reajuste.

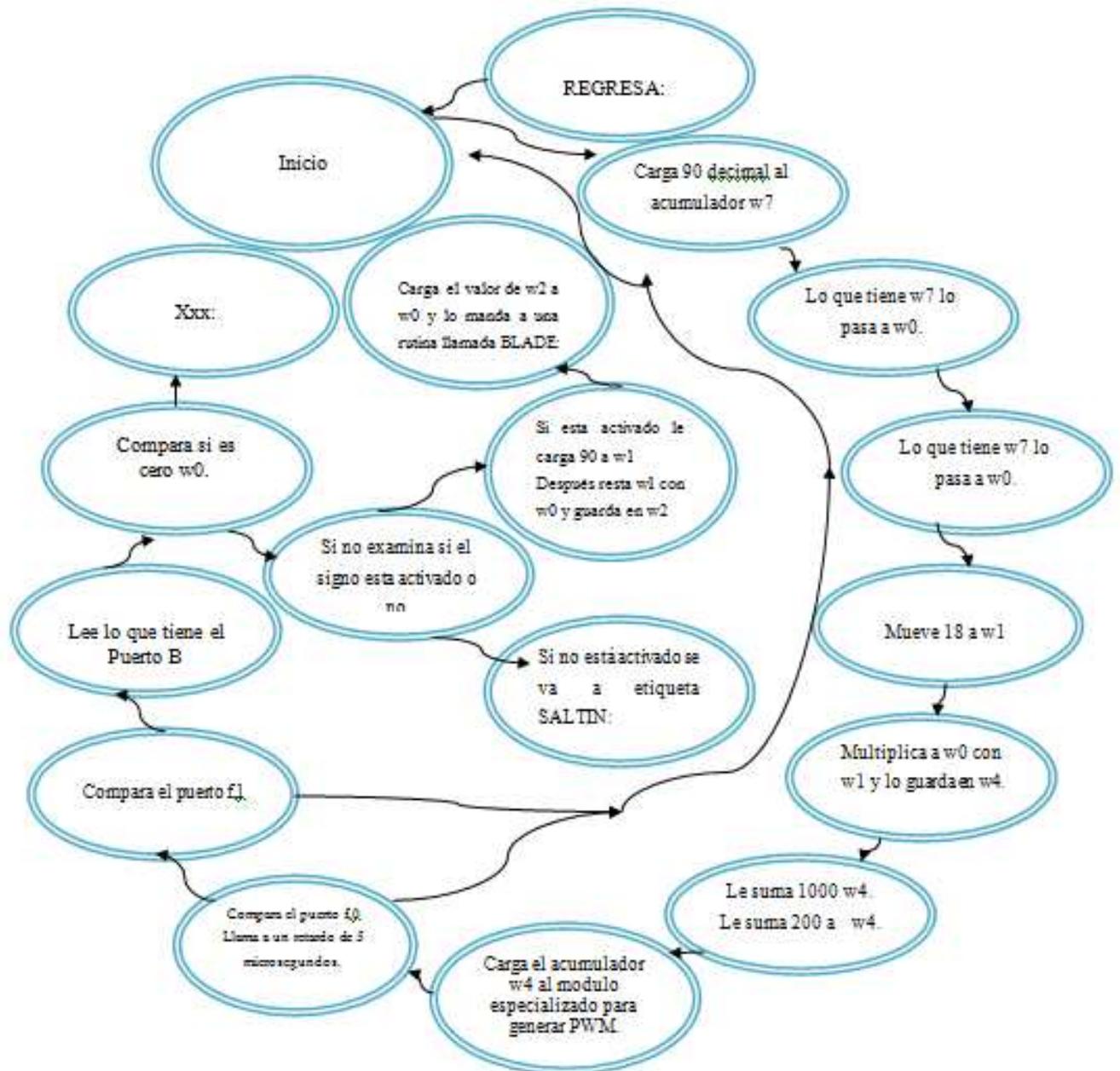


Fig. 4.61. Máquina de estado de un Esclavo DS Pic 30f3014P.

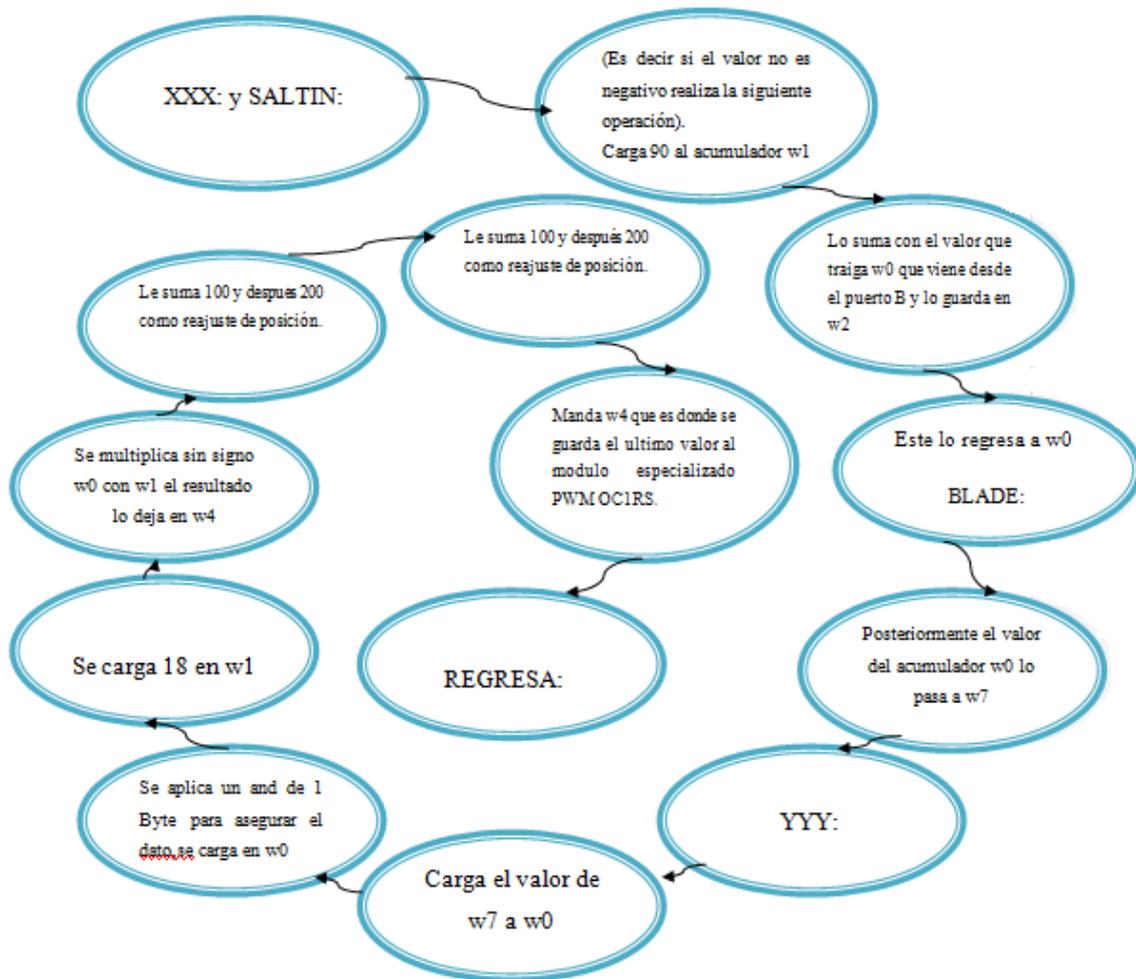


Fig. 4.62. Máquina de estado y su continuación.

Aunque el programa véase figura (4.61) y figura (4.62), fue el elaborado para implementarse en cada una de las articulaciones, este indica algunas de las instrucciones más relevantes con respecto a lo que realiza el DsPic 30f3014P, una introducción para saber lo que está realizando el programa sería la presentada en la fig. (4.63):

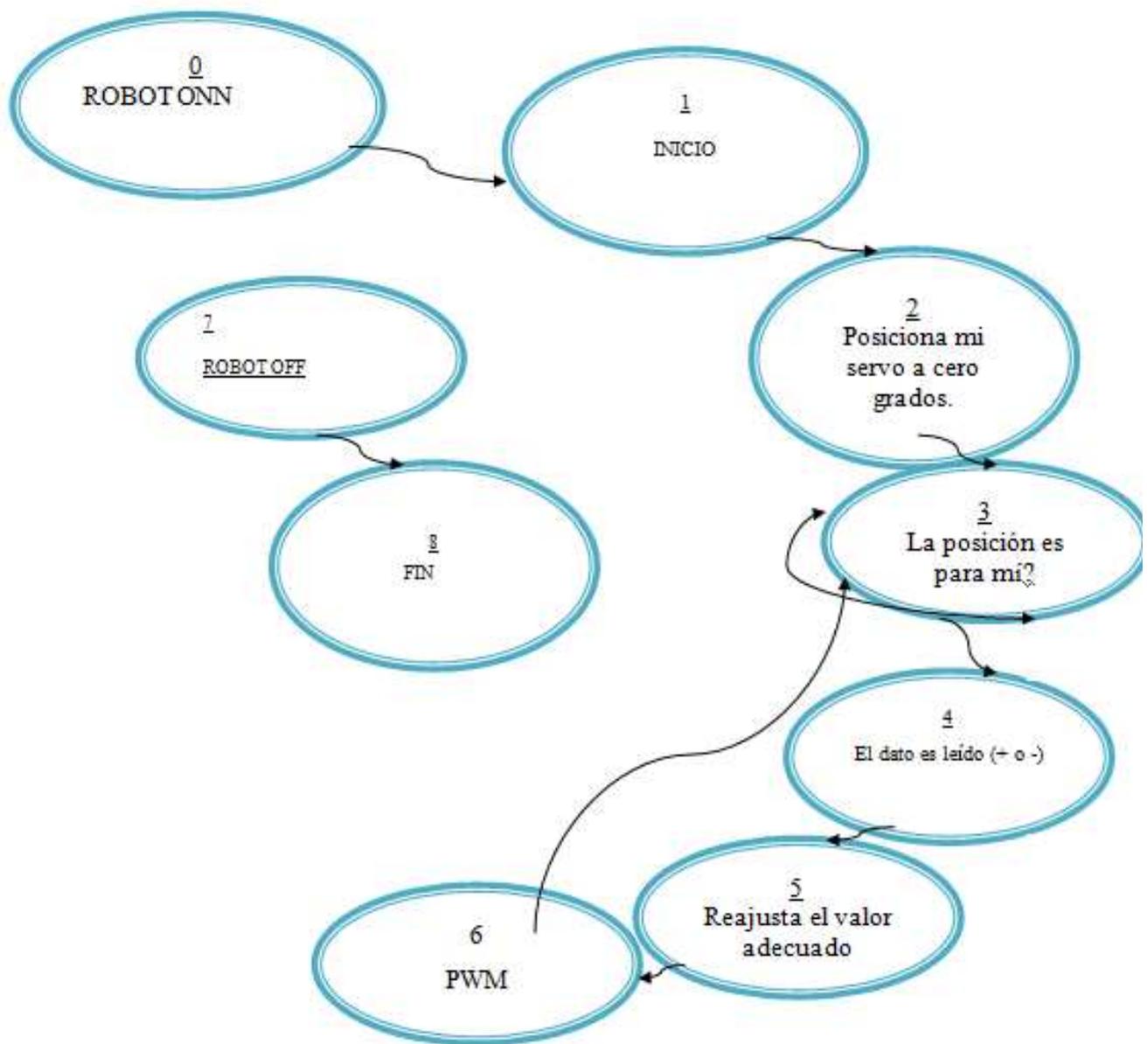


Fig. 4.63. Máquina de Estado para determinar si necesita el motor ir a la izquierda o a la derecha.

4.2.4 Relevancias que tiene el software National Instruments.

4.2.4.1 (Pruebas realizadas con Daq USB 6008-USB 6009).

Gracias a las clases tomadas por parte del M. en C. Raúl Rivera Blas pudimos utilizar la paquetería de National Instruments LabView versión 8.2. , este software aunque no es de uso libre tiene diferentes ventajas y capacidades de expandir trabajos y explotarlos al máximo de diferentes formas, entre las ventajas que maneja, es que no es necesario cambiar el código cada vez que sale una diferente versión de LabView como lo hacen algunos modelos de microcontroladores e inclusive microcontroladores que salen al mismo tiempo como ejemplo, no obedecen a las mismas instrucciones. Lo que genera que mucho de ese software se vea inservible al momento de emigrar a un dispositivo nuevo.

Otra de las características que maneja este programa es que su modo de programación es en paralelo y muchas de sus instrucciones pueden ser accionadas al mismo tiempo lo que tiene diferentes ventajas a la hora de programar, sino fuera suficiente esto también tiene la posibilidad de emigrar código de programas (con ciertas limitantes) al mismo LabView y poder trabajar en conjunto con estos, otra posibilidad es la de utilizar código que pueda ser programado en serie, tiene la versatilidad de usar cualquier módulo National Instruments, o según sea el caso se pueden acoplar diferentes maneras para comunicar nuestros dispositivos National con otros medios, un ejemplo de esto es utilizar el puerto COM de la PC como entrada y salida de datos en caso de no contar o contar con suficientes entradas y salidas del módulo que se esté utilizando, la posibilidad de utilizar antenas de comunicación las cuales puedan ser monitoreadas en cualquier parte del mundo, dispositivos *Ethernet*, *bluetooth*, *wifi*, *xbee* entre otros. Aunque se debe entender que esto que se ha comentado no es nada comparado con lo que en realidad puede ser utilizado este software para diferentes implementaciones. A un punto de vista muy particular este proyecto tiene muchas posibilidades de ser mejorado aunque el tiempo para el que se ha propuesto realizar sea muy corto esperamos de pautas para que pueda ser mejorado en todos sus aspectos.

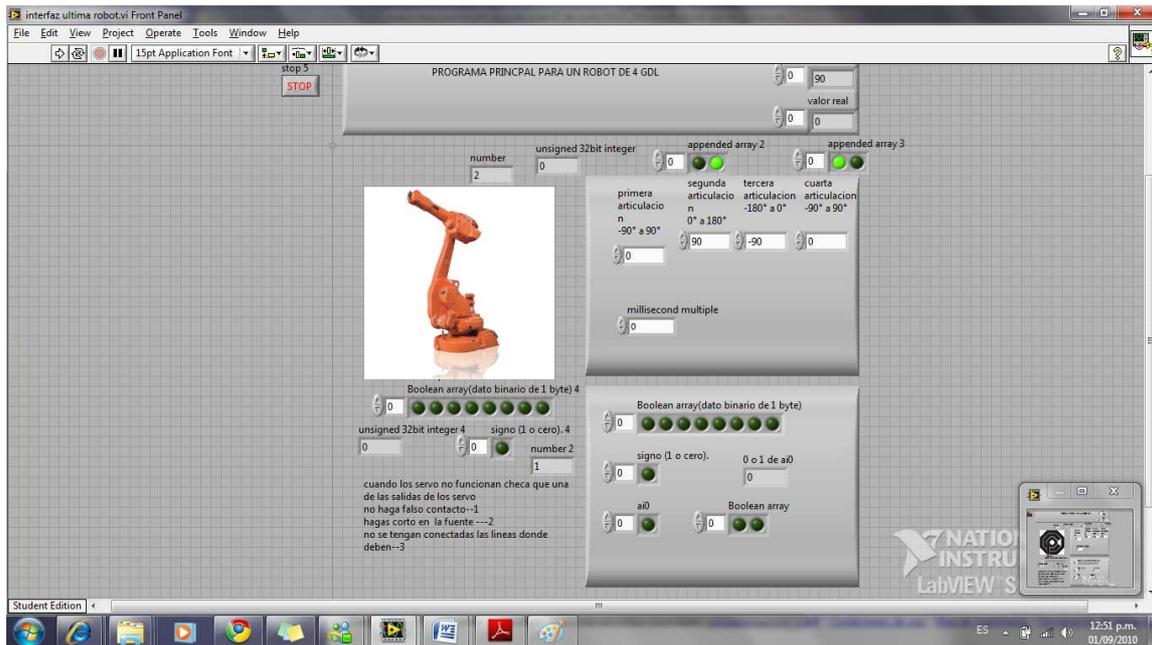


Fig. 4.64 Panel Frontal para la manipulación de Ángulos en LabVIEW.

Para la Integración completa entre el prototipo del robot de 4 grados de libertad, fue necesario usar una base en LabVIEW véase fig.(4.64), la cual sirve como plataforma para conjuntar el control realizado por los microcontroladores Dspic 30f3014p y la *Daq* assistant 6008-6009 de National Instruments, para esto fue necesario tener cuatro entradas denominadas controles, que nos sirven para iniciar nuestro proceso en *LabVIEW*, estos recibieron los nombres de primera articulación de -90 grados a 90 grados, porque es el espacio que tenemos disponible para la movilidad de la primera articulación, después tenemos el segundo control o entrada denominado de 0 a 180 grados, esto debido a que nosotros podemos darnos cuenta que al realizar la transformada Homogénea los ángulos de cada servomotor se recorren según su configuración, para esto le podemos sumar la limitante que el servomotor solo cuenta con 180 grados es decir; que se recortaría mas sus grados para cada articulación aunque posteriormente esto no tendrá importancia ya que haciendo las modificaciones pertinentes pasara a ser un motorreductor común. Esto sin considerar que la articulación numero 3 se necesita invertir el ángulo final puesto que el movimiento físico que es producido hace que el ángulo giro en sentido contrario. En resumen la articulación 1 y 4 no necesitan ser reajustadas pero la articulación número dos y tres sí, como se verá a continuación.

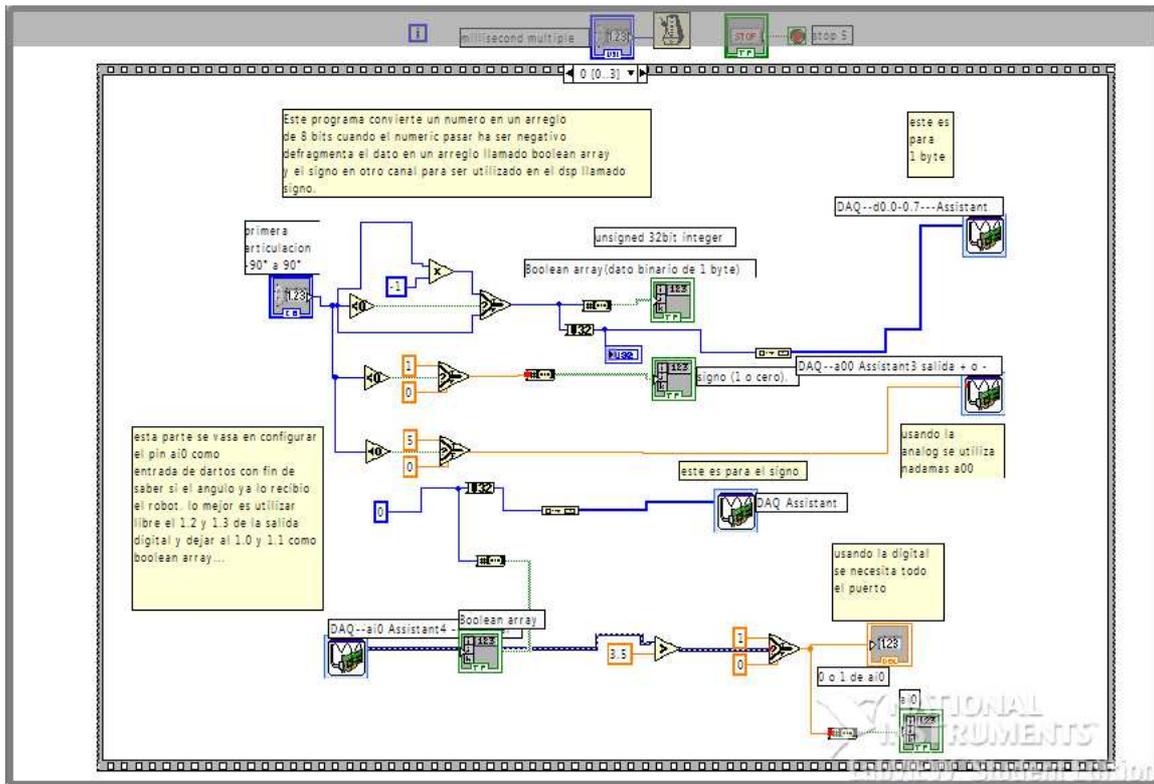


Fig. 4.65 Diagrama de Bloques con programación para la Primera Articulación.

En el primer caso podemos observar que la entrada de la primera articulación (conexión además en la cual se conectara una de las entradas de la cinemática directa) véase fig.(4.65), necesita tomar una serie de decisiones para poder ser traducida en un byte de salida configurado como la salida digital de D00 a D07, para esto necesitamos manejar números negativos, cuando usamos números negativos regularmente se desborda el contador para eso, lo que se procedió a realizar en el programa fue multiplicar por -1, todos los valores negativos de esta forma el dato que recibirán serán los adecuados, a esto agregándole que se decidió utilizar una salida analógica para determinar el signo, esta fue comparada al realizar un bloque de comparación la cual determinaba que si el dato de entrada es <0 entonces se activaba por medio de una *select* y este mandaba a la salida analógica de la *Daq* A00, pero todo esto encerrado dentro de una estructura *stacked sequence*.

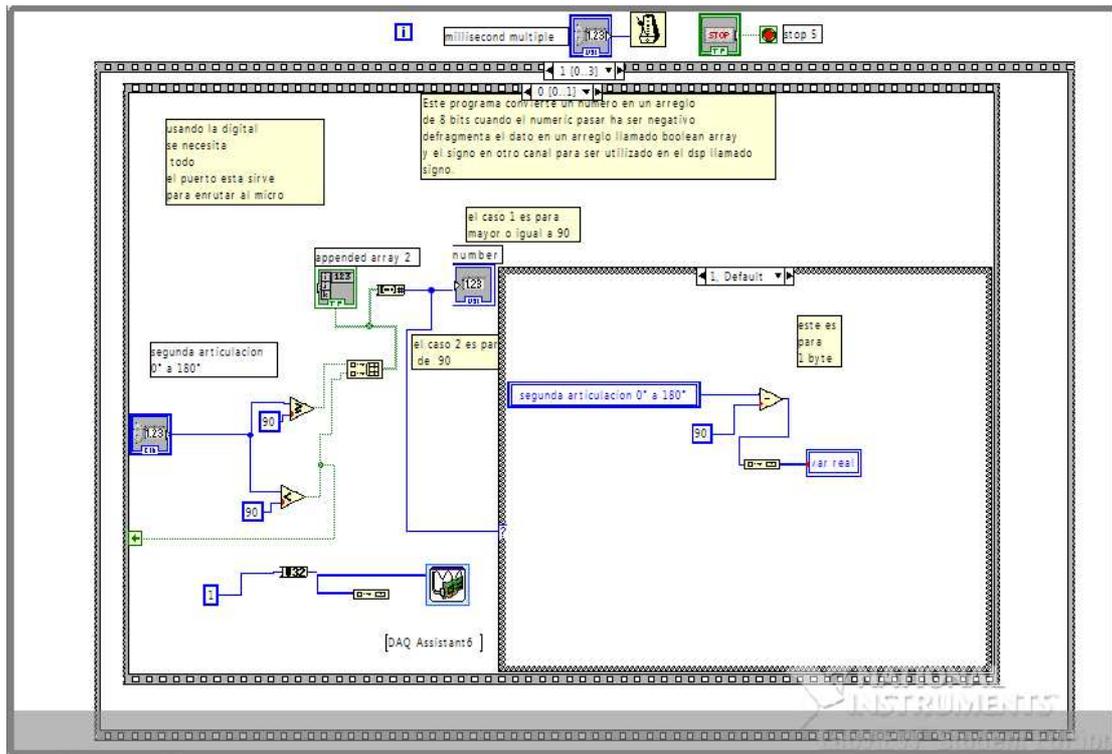


Fig. 4.66 Diagrama de Bloques para la Segunda Articulación.

En el caso de la articulación número 2 véase fig. (4.66), representado como stacked número 1, podemos observar otras particularidades en el programa, tales como otros stacked dentro del principal esto porque en las pruebas realizadas con anterioridad se obtuvo problemas con el acondicionamiento de estructuras case, lo que ocasionaba que hubiera varios indicadores en la salida y se obtuvieran dos datos diferentes para una misma salida, para esto se decidió hacer dos casos de *stacked secuencia* y un *case* pero que se contara con variables locales en la salida, lo que producía tener un indicador en especial que envía los datos al robot. La estructura interna de los bucles de caso "*Case Loop*" en esta parte realizan una función importante con respecto a la metodología de la cinemática directa pues estos reacomodan los ángulos para que nuestros espacios de trabajo sigan teniendo 180° y los grados de libertad no sean una limitante. La acción que realiza el bloque véase fig. (4.67), es comparar si el dato que entra es mayor o menor que 90 de esta forma cualquiera de las dos condiciones del case pueden ser activadas, una vez que sucede todo esto realiza las siguientes operaciones: si es mayor o igual a 90, lo que hace es restarle 90 al dato de entrada. De esta forma el resultado que arroja sería equivalente a valores de grados a la

derecha, puesto que se reconfiguraron los grados, esto quiere decir que inician en cero grados y tienen 90 grados a su derecha y 90 a su izquierda.

Si dato ≥ -90

Dato = $90 + x$

$x = \text{dato} - 90$

Si dato < 90

Se Manda a 1 salidas analógica

$x = 90 - \text{dato}$

Fin.

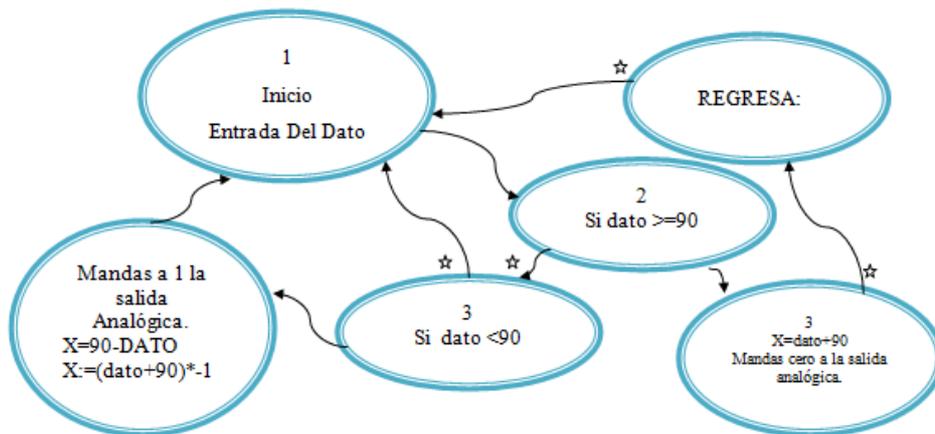


Fig. 4.67. Máquina de Estado para determinar si el Signo es Positivo o Negativo.

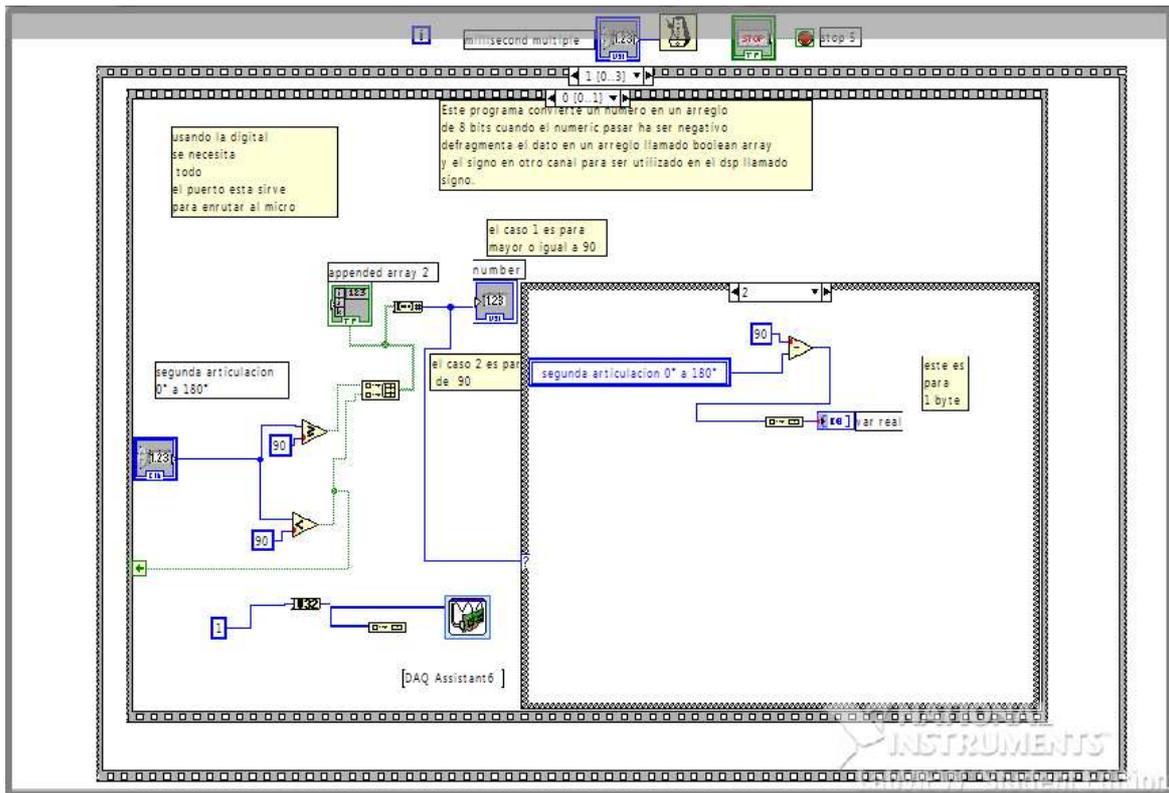


Fig. 4.68. Diagrama de Bloques de la Segunda Articulación.

En el segundo caso de las segunda articulación se puede observar que ahora lo que se procede a realizar es que se resta 90 menos el dato de entrada y de este es enviado a la salida, como lo muestra la fig. (4.68).

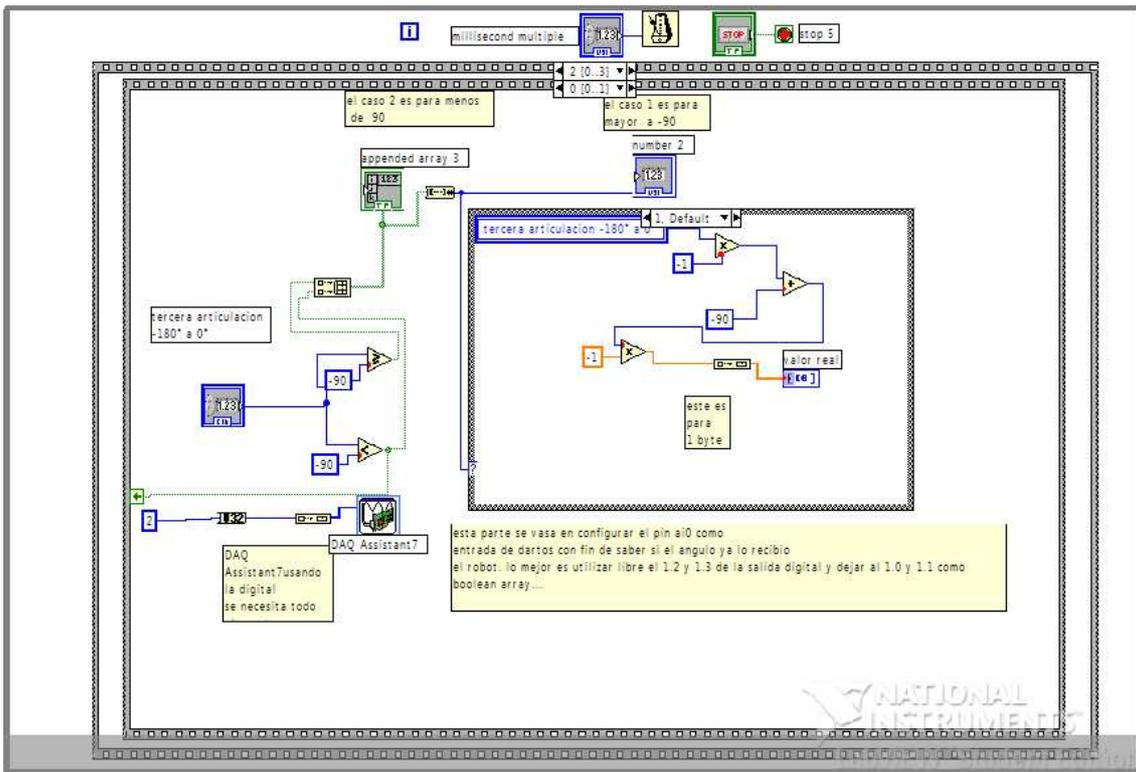


Fig. 4.70. Diagrama de Bloques de la Tercera Articulación.

Para el caso de la articulación número tres, véase fig.(4.70), aplicaría casi el mismo concepto que en el pasado a diferencia que el arreglo para el reajuste de grados con respecto a la transformada homogénea va de 0° a -180° , y para esto se hace el reajuste porque el micro fue acoplado para leer de 90 a -90 grados. Y para esto se realizó la conversión según el siguiente diagrama véase fig. (4.71).

Si el dato ≥ -90

$$\text{Dato} = x - 90$$

$$X = \text{dato} + 90$$

Si el dato < -90

Mandas a 1 salida analógica

Y el resultado es:

$$X = (\text{dato} + 90) * -1$$

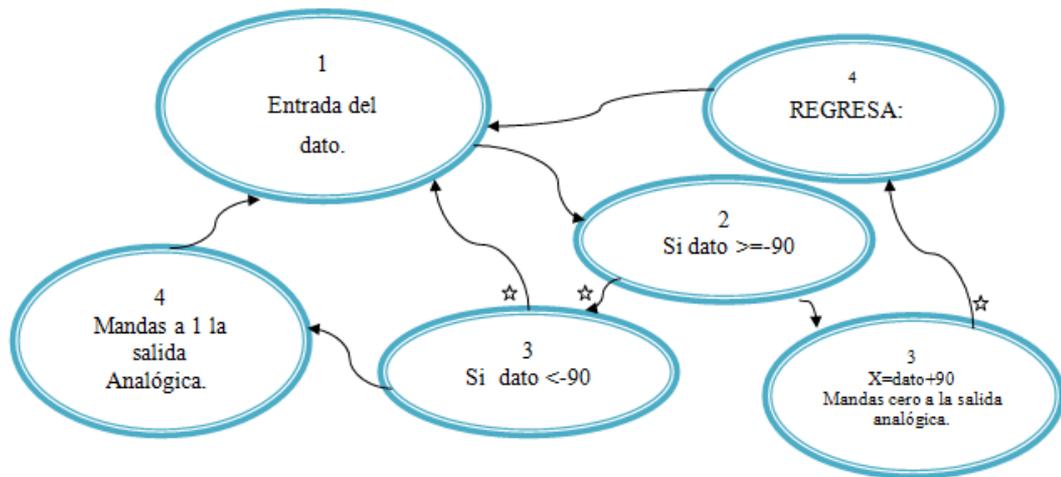


Fig. 4.71. Máquina de Estado para el reconocimiento de Datos Positivos y Negativos.

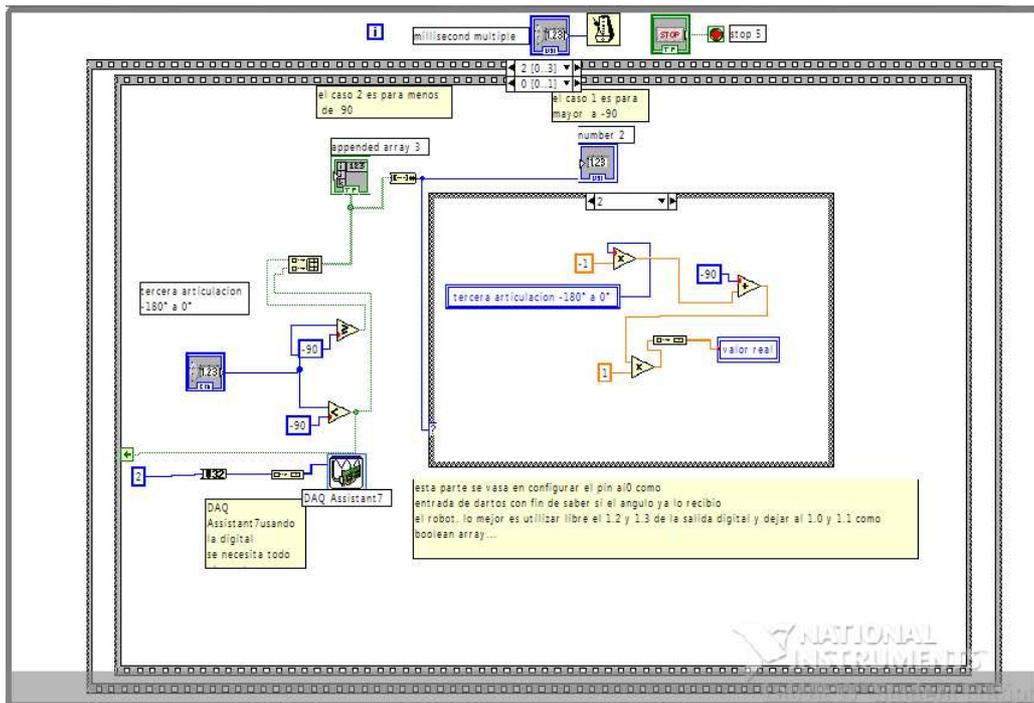


Fig. 4.72. Segundo Caso de la Tercera Articulación.

En esta imagen se puede observar el diagrama realizado para desarrollar la segunda etapa del case structure para la articulación número 3 véase fig. (4.72).

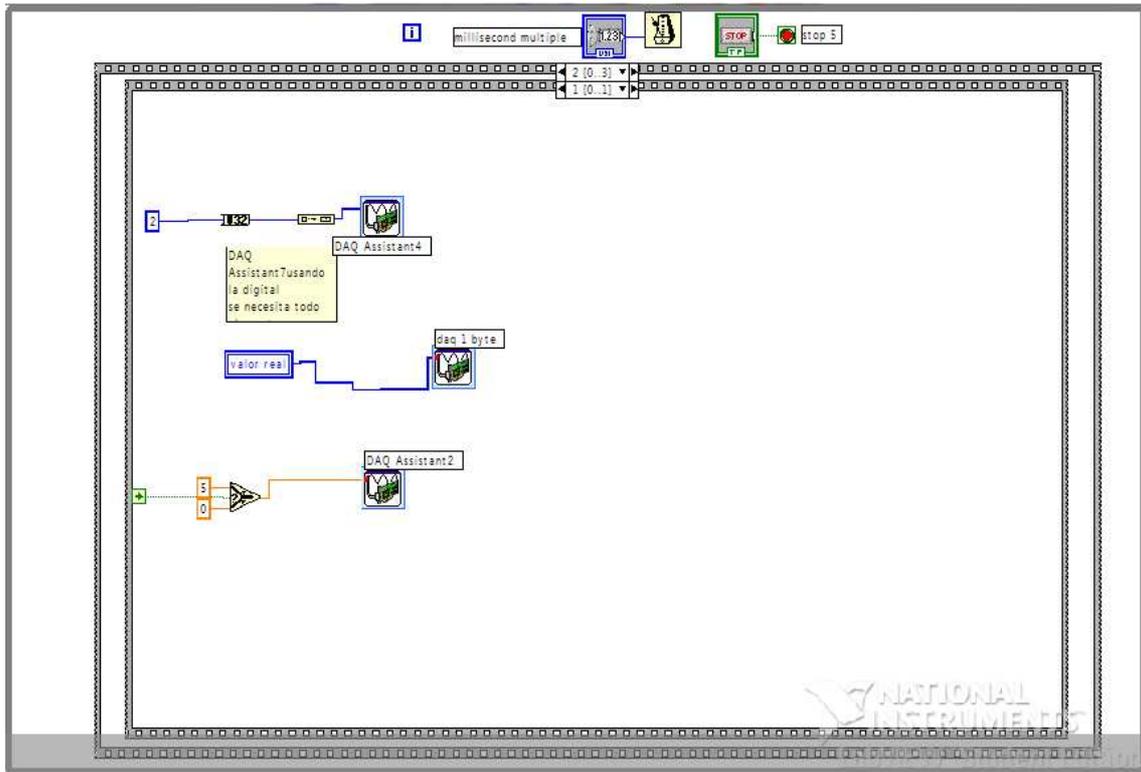


Fig. 4.73. Diagrama de Bloques de la Tercera Articulación.

Una vez que se realiza la etapa de decisión, se procede a mandar los bits necesarios para que el microcontrolador reconozca si le corresponde a él o a otro esclavo, véase fig.(4.73), además se decide si el signo debe de ser positivo o negativo.

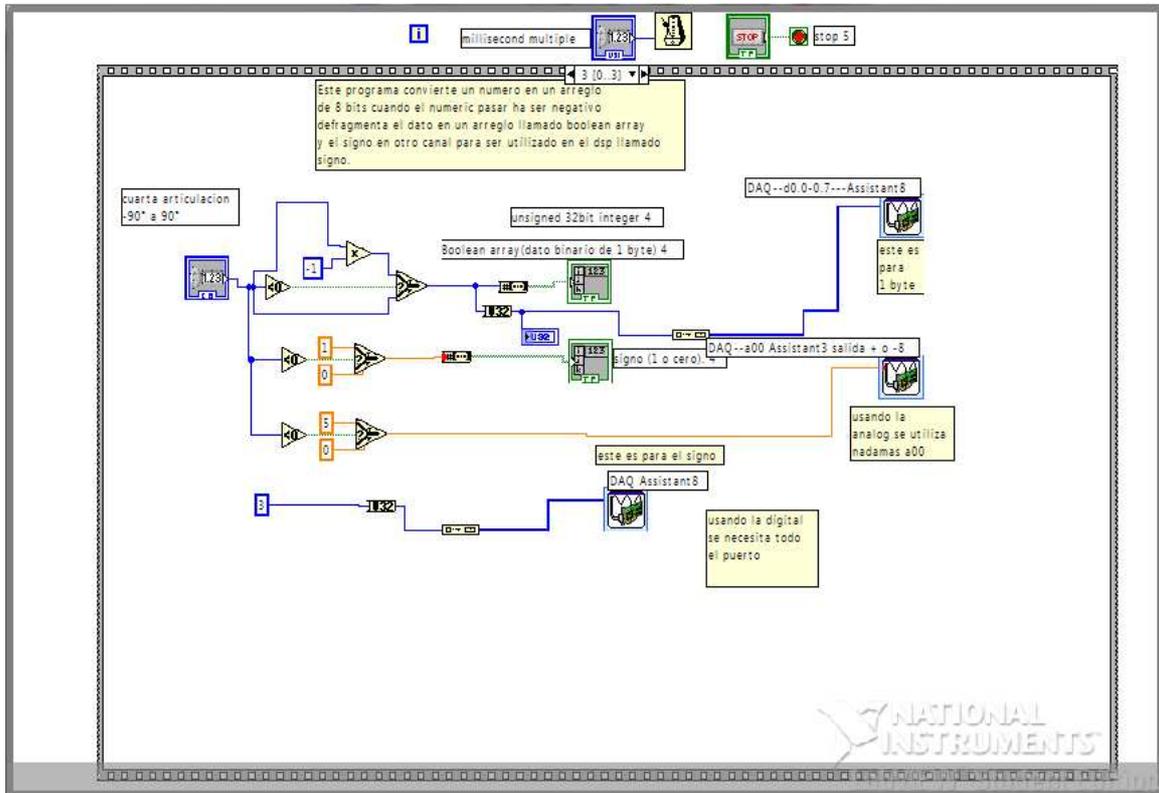


Fig. 4.74. Diagrama de Bloques de la Cuarta Articulación.

Después de haber realizado lo correspondiente a la articulación número 3, véase fig. (4.74), el programa por medio del *stacked sequence* pasa al último valor de las articulaciones de entrada que es la número 4, como esta articulación no necesita ser reajustada se hacen las comparaciones similares a la articulación número 1 y manda por la salida de la *Daq 6009-6008* el ángulo y el número de correspondencia del esclavo así como el signo para indicar si va a la izquierda o a la derecha.

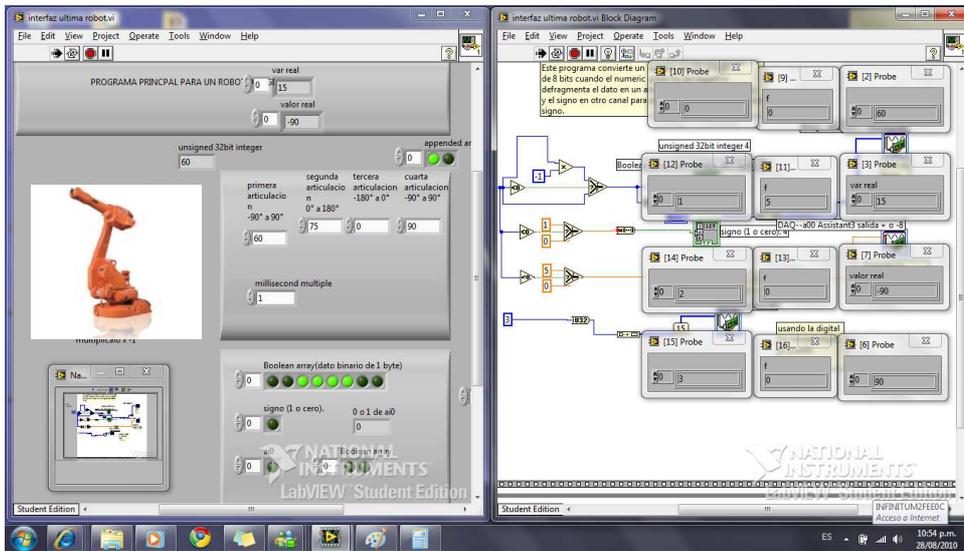


Fig. 4.75. Diagrama de Bloques y Panel Frontal Funcionando al Mismo Tiempo.

Como se puede observar en los recuadros de la derecha véase fig. (4.75), podemos darnos cuenta que el último programa realizado hasta el momento realiza los cambios pertinentes con respecto a cada articulación y además activa el ángulo cuando este es requerido como en el caso de la articulación de entrada 75° , en este proceso el robot ya está desfasado 90° , por lo tanto para llegar a 75° se procede a restar 15° a la posición del servomecanismo actual véase fig. (4.76) y fig. (4.77).

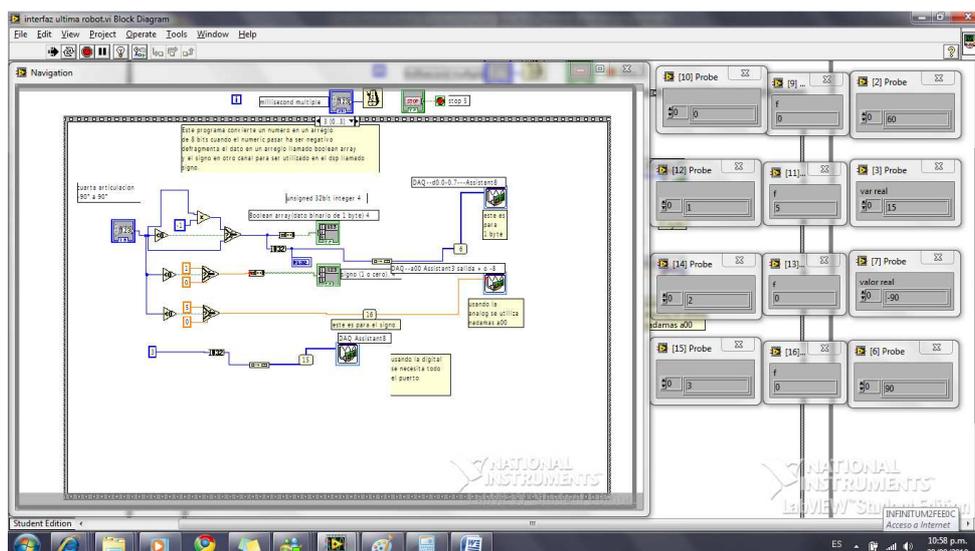


Fig. 4.76. Diagrama de Bloques examinando al Cuarto Eslabón.

Como conclusión podemos definir que una de las problemáticas que encontramos hasta el momento es la sincronización, la cual se hablara más adelante, Así como el problema de salidas múltiples para un mismo servomotor lo que ocasiona en algunos momentos la duplicidad de información, genera ruido y un indeseable control de cada eslabón.

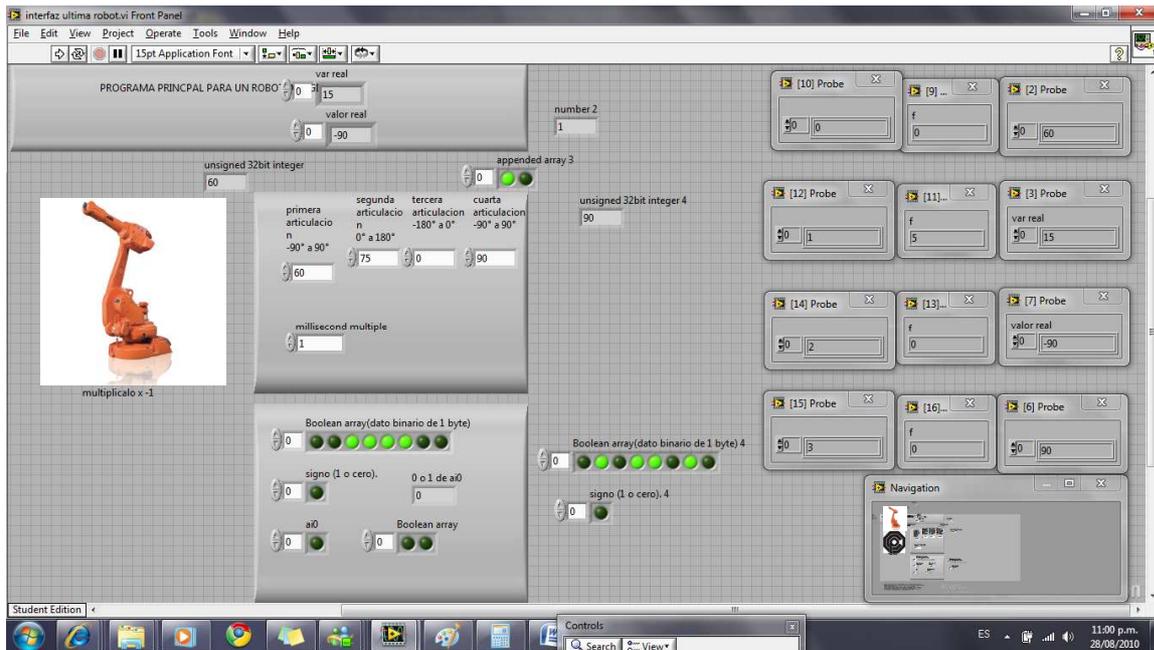


Fig. 4.77. Panel Frontal con Pruebas de Salida por medio de LabVIEW, la cual demuestran la conversión realizada para reconocer los 90 Grados.

4.2.4.2 Programa Prueba para la realización de la cinemática directa implementando resultados de operaciones en Matlab.

En este caso se realizó un programa en el cual se pudieran utilizar las ecuaciones cinemáticas del robot de 4 grados de libertad extraídas de Matlab, Utilizando la versión de LabView 8.2 véase fig. (4.78). Se introdujeron datos similares a los realizados en los ejemplos en Matlab y se declararon valores unitarios para las distancias en cada eslabón (para el efecto experimental), agregando un valor dado para determinar el comportamiento del mismo.

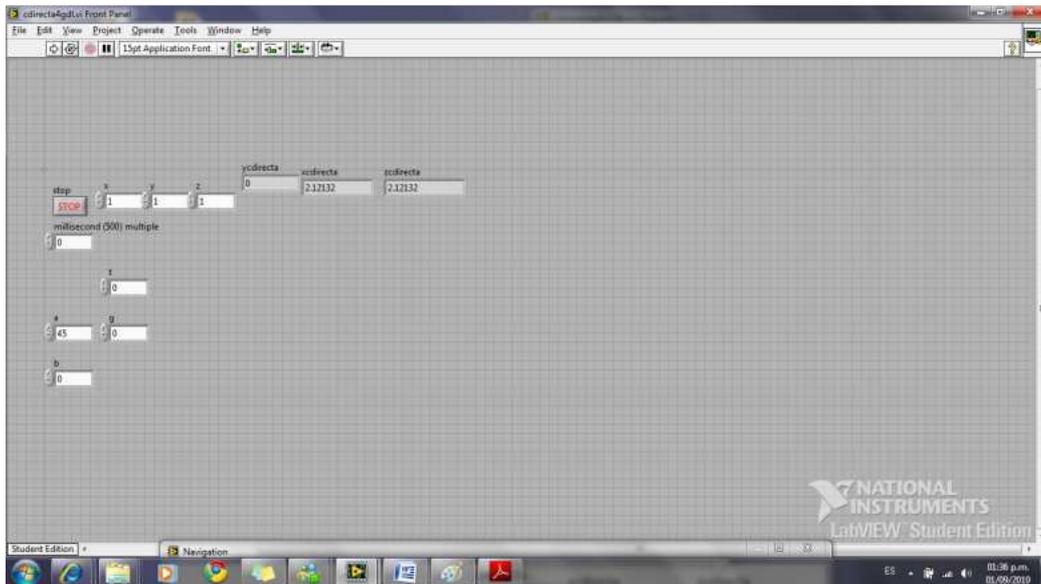


Fig. 4.78. Panel Frontal con los Resultados Obtenidos de la Cinemática Directa por medio del Math Script.

Para este ejemplo se seleccionó un alpha (a)=45 los resultados en Matlab fueron aproximados a LabView 2.12 y 2.12 para xdirecta y zdirecta.

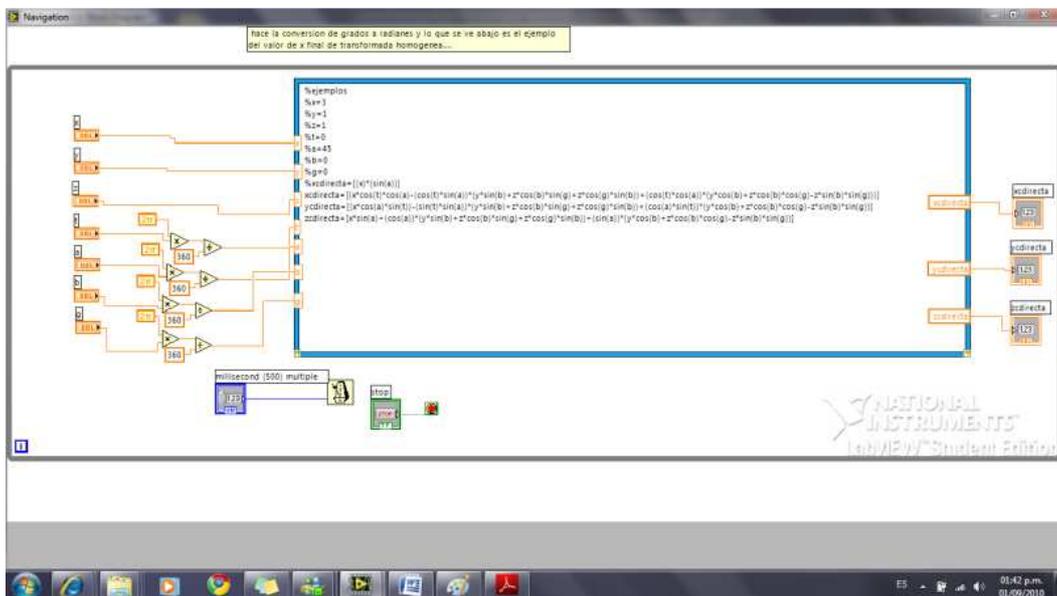


Fig. 4.79. Navigation de LabVIEW, Simulación de las Coordenadas de Posicionamiento para un robot de 4 Grados de libertad.

En el *block Diagram* se agregaron los siguientes datos de Matlab los cuales se pueden verificar en la sección de Matlab son idénticos. Además se encerró esto en la estructura de *MathScript* y en una estructura *While* que nos sirve para ciclar el programa véase fig.(4.79).

4.2.4.3 Ejemplo de la cinemática Directa Aplicada en Matlab

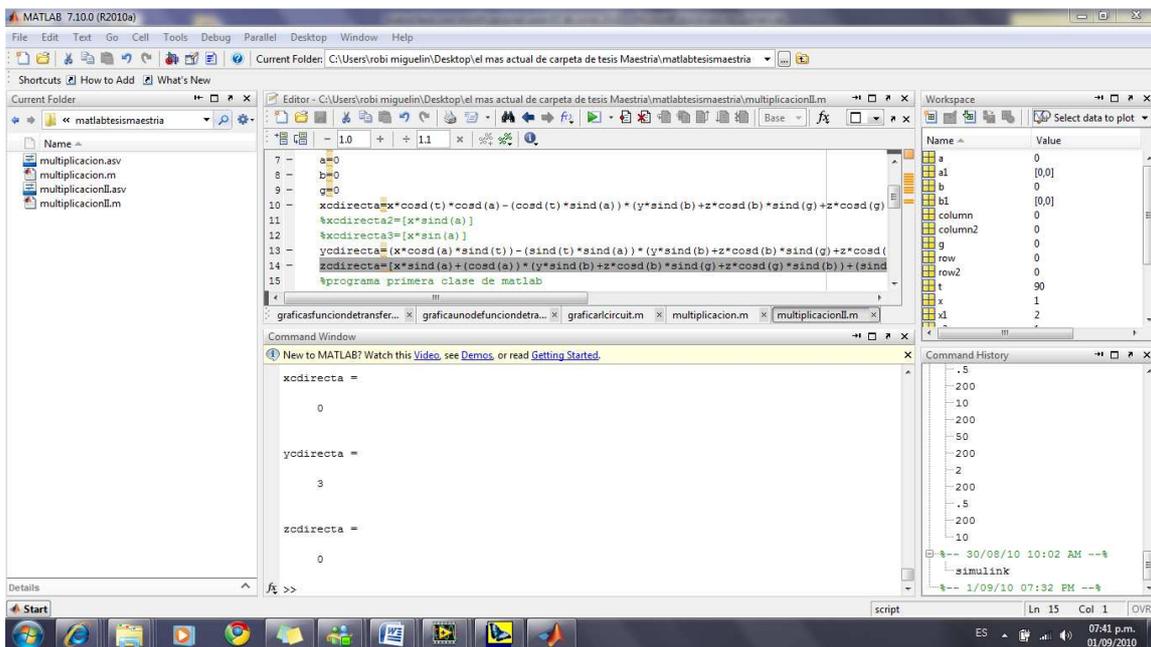


Fig. 4.80. Programa MATLAB utilizado para la Obtención de Posicionamiento de Robot de 4 G.D.L.

En este caso se utilizó la versión de Matlab 2010 para 64 bits, como se puede observar se colocaron valores como es el caso de la figura (4.80) con el fin de averiguar si en tanto Matlab como en *LabView* daban los mismos valores, el resultado fue el mismo. En este caso los ángulos declarados según su articulación fueron t, a, b, g de los cuales se declaró a t=90 el resultado nos arroja 3 en "y directa" véase fig. (4.81) y fig. (4.82), al igual que en el caso de *LabVIEW*, los valores de "l" por cada eslabón se declararon unitarios por razones únicamente de pruebas, puesto que esto se cambian dependiendo el robot a analizar.

```

1 %primera ecuacion para conocer la distancia en x de un robot de 4gdl
2 %ejemplos pero en labview se sustituyen estos valores por los externos (los reales)
3 x=1
4 y=1
5 z=1
6 t=90
7 a=0
8 b=0
9 g=0
10 xdirecta=x*cosd(t)*cosd(a)-(cosd(t)*sind(a))*(y*sind(b)+z*cosd(b)*sind(g)+z*cosd(g)*sind(b))+cosd(t)*cosd(a)*(y*cosd(b)+z*cosd(b)*cosd(g)-z*sind(b)*sind(g));
11 %xdirecta2=[x*sind(a)]
12 %xdirecta3=[x*sind(a)]
13 ydirecta=(x*cosd(a)*sind(t))-(sind(t)*sind(a))*(y*sind(b)+z*cosd(b)*sind(g)+z*cosd(g)*sind(b))+cosd(a)*sind(t)*(y*cosd(b)+z*cosd(b)*cosd(g)-z*sind(b)*sind(g));
14 %ydirecta=[x*sind(a)+cosd(a)*(y*sind(b)+z*cosd(b)*sind(g)+z*cosd(g)*sind(b))+sind(a)*(y*cosd(b)+z*cosd(b)*cosd(g)-z*sind(b)*sind(g))];
15 %programa primera clase de matlab
16 %autor
17 %entrada
18 %salida
19 %fecha y version
20 %a=[3;2;1]
21 %b=[1;2;3]
22 %c=a+b
23
24 %Esto es del dr salmeron*****
25 % a=input('da el valor de la matriz 1: ');
26 % al=size(a)
27 % row=al(1,1)
28 % column=al(1,2)
29 % x1=2
30 % x2=1
31 % b=input('da el valor de la matriz 2: ');
    
```

Fig. 4.81. Editor del Programa MATLAB.

```

MATLAB 7.10.0 (R2010a)
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

a =

     0

b =

     0

g =

     0

xdirecta =

     0

ydirecta =

     3

zdirecta =

     0

>>
    
```

Fig. 4.82. Resultados del Programa MATLAB.

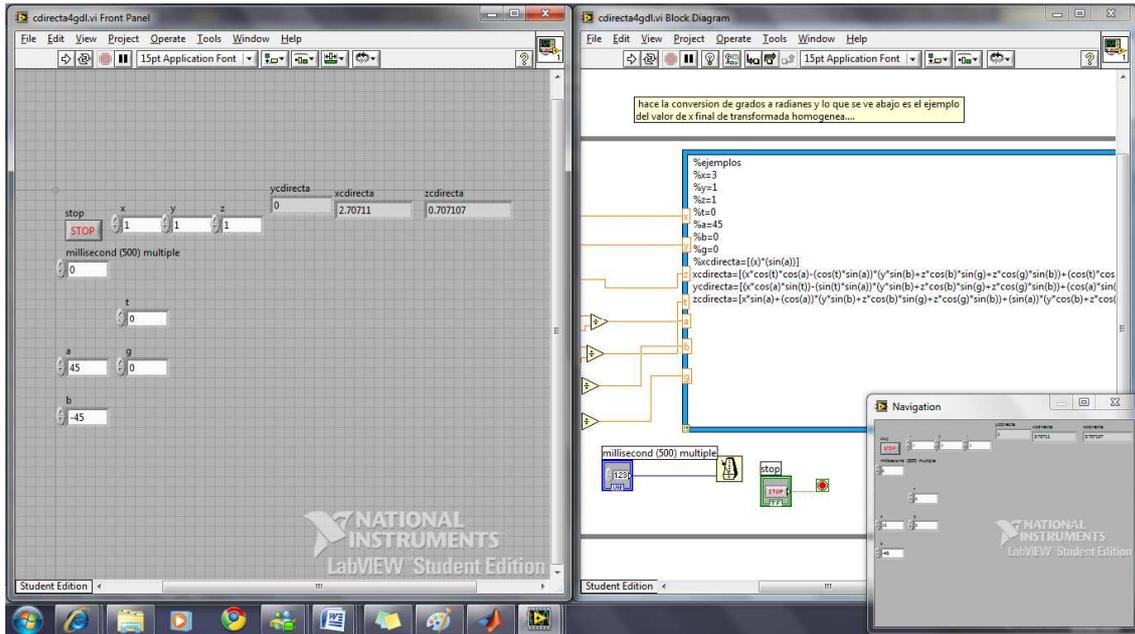


Fig. 4.84. Programa prototipo de una Transformada Homogénea de un Robot de 4 grados de Libertad.

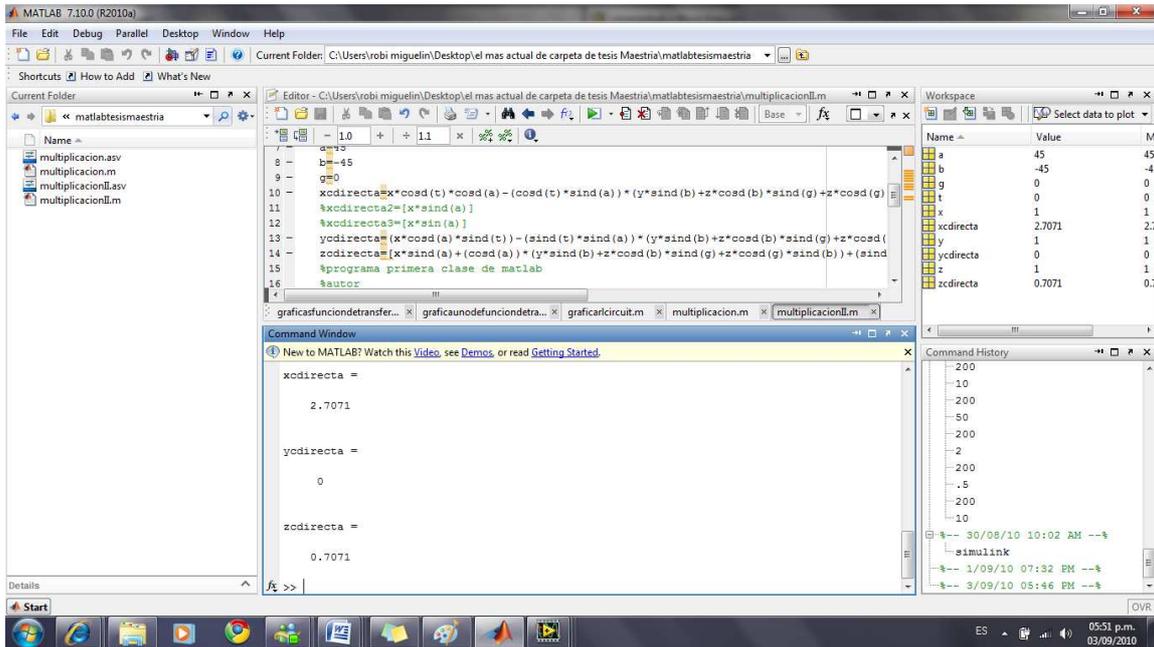


Fig. 4.85. Equivalencias con el programa MATLAB.



5

CONCLUSIONES Y TRABAJO FUTURO



5.1 Conclusiones

Gracias a las diferentes técnicas que se han podido aprender para la solución de cinemática Directa, Parámetros de Denavith Hartenberg y Cinemática Inversa, se puede decir que hay diferentes oportunidades en la utilización de estas herramientas y cabe mencionar que todos estos resultados pueden tener un enfoque diferente y pueden servir no solo en la robótica sino en otros tipos de mecanismos. Todo esto gracias a los métodos geométricos, los métodos algebraicos y numéricos, capaces de poder obtener los ángulos de cada una de las articulaciones. La ventaja en los métodos geométricos como tal se basa en la facilidad de encontrar estos ángulos por medio de planos bidimensionales, a veces los métodos algebraicos nos ayudan como soporte de los geométricos cuando es difícil trabajar con un robot, mientras que los numéricos pueden llegar a ser muy útiles cuando tenemos robots de más de 3 grados de libertad. Además es interesante saber que con estos últimos podemos manipular la orientación del efector final para acomodar al robot con la orientación deseada.

Como tal también cabe mencionar que la cinemática directa generada por las Matrices de Transformación Homogénea tienen una manera interesante de poder ponderar las especificaciones particulares en el análisis de cada configuración del robot, es decir se pueden obtener diferentes formas de representar a un mismo manipulador y de esta forma le es posible determinar la localización de la base, así como reconocer si el robot utilizado tiene al final de sus marcos de referencia un efector final o un sistema de sujeción para poder manipular otras herramientas, pues para esto entonces se necesitaría reconsiderar en donde se encuentra el marco de referencia real del manipulador industrial de la base a su extremo.

Como tal con lo que cabe mencionar en la parte de la aplicación de la Lógica Difusa al final es una herramienta útil la cual se llega a utilizar para controlar temperatura, factores ambientales, funciones trigonométricas, así como gran influencia con la toma de decisiones, que se podría aplicar en visión artificial; sin embargo puede llegar a consumir memoria y

energía una vez que este llega a ser aplicada en problemas como los mencionados por control. Considerando que ya una buena aproximación puede ser solucionada por medio de un control PID; me encontraría plenamente confiado de que este puede realizar cualquier trabajo sin complicación, ya que se debe de ponderar entre el costo, la precisión y la capacidad de sintonización. Cabe mencionar que las aplicaciones que se han podido realizar han mostrado buenas expectativas hacia mí. Sin embargo creo que la optimización de un control aplicado a un dispositivo puede ir mas allá que una simulación y la improvisación de técnicas óptimas que permita la aproximación en la caracterización de un fenómeno como lo es el control de motores. Así como la reciprocidad de la precisión con la cual nuestros instrumentos de medición pueden monitorear, generar una buena discretización obtenida de un fenómeno en particular y las habilidades para poder aplicar la electrónica en ella.

5.2 Trabajo futuro

Considero que la continuación para este proyecto es un largo camino que realizar dentro de lo que se puede mencionar, los programas realizados en LabView fueron conscientemente realizados para que posteriormente se puedan emigrar a Matlab o cualquier otro programa, por medio de operaciones básicas, con esto quiero mencionar que también hay múltiples herramientas para interfazar a Matlab con LabVIEW o con programas con comunicaciones virtuales. Con esto se puede obtener una simulación de lo que puede hacer el robot y la comunicación para que el robot físico pueda hacer lo mismo que en el modelado, además es posible la implementación de SIMILINK y el programa guía (*Gui*) de Matlab para desarrollar una ambiente gráfico tal como lo hace Matlab, con el fin de dejar de trabajar solo con la *Daq* 6008 y 6009 y empezar con una comunicación desde Matlab o mismo LabVIEW con solo un ft232 (Convertidor usb-rs232 para comunicación usart) y una comunicación como tal USART.

La implementación de diferentes técnicas de control en las cuales puedan ser introducidas a sistemas embebidos por medio de formulas discretas, obtención de modelos matemáticos que representen el comportamiento de un sistema sin la necesidad de grandes pruebas experimentales y la introducción de diferentes medios de comunicación entre sistemas móviles que ya manejen hyperterminals; como lo fue en el caso del primer proyecto realizado en la tesis, la aplicación de antenas como medios de comunicación entre el robot y el usuario, por no mencionar mas ideas que probablemente ya han sido realizadas, pero que sin duda implican una gran reto para todo estudiante.



6 ANEXOS



ANEXO 1.

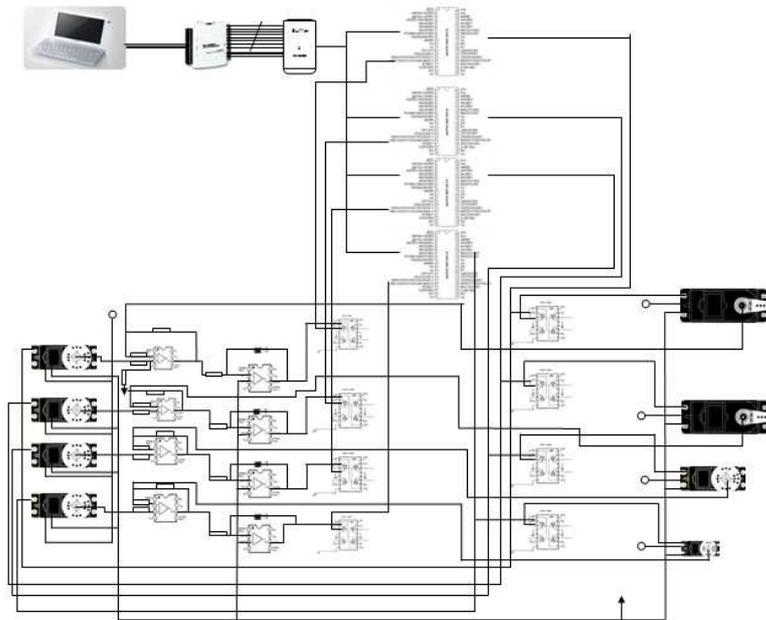


Fig. 6.1. Esquema General de control Proporcional para control de ángulo de un prototipo de 4 grados de libertad.

Durante el proceso se han realizado algunos casos experimentales para saber la eficacia de diferentes componentes como tal en la tesis se presentan algunos diagramas que relacionan comparaciones entre diferentes controles, y algo que es interesante comentar es que muchas veces es importante un buen control, pero cuando se utiliza un sensor como lo es un encoder es muy importante la resolución del sensor, pues si este no tiene mucha resolución por más que sea eficiente el control su resolución no podrá ofrecer más de lo que el control pueda monitorear véase fig. (6.1).



Fig. 6.2. Interfaz realizada para control de servomecanismo.

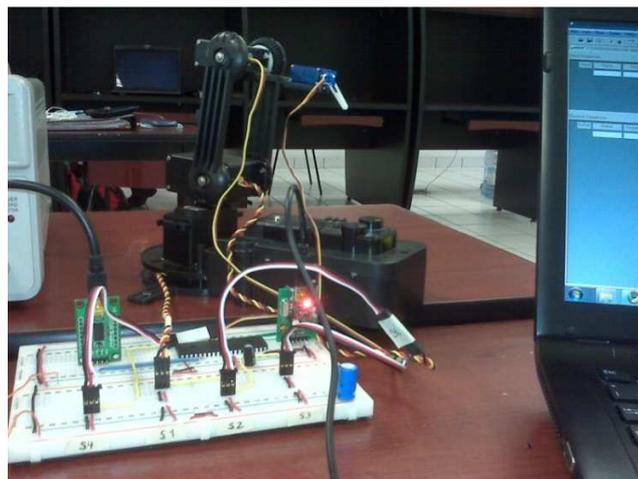


Fig. 6.3. Robot de 4 grados de libertad en lazo abierto.

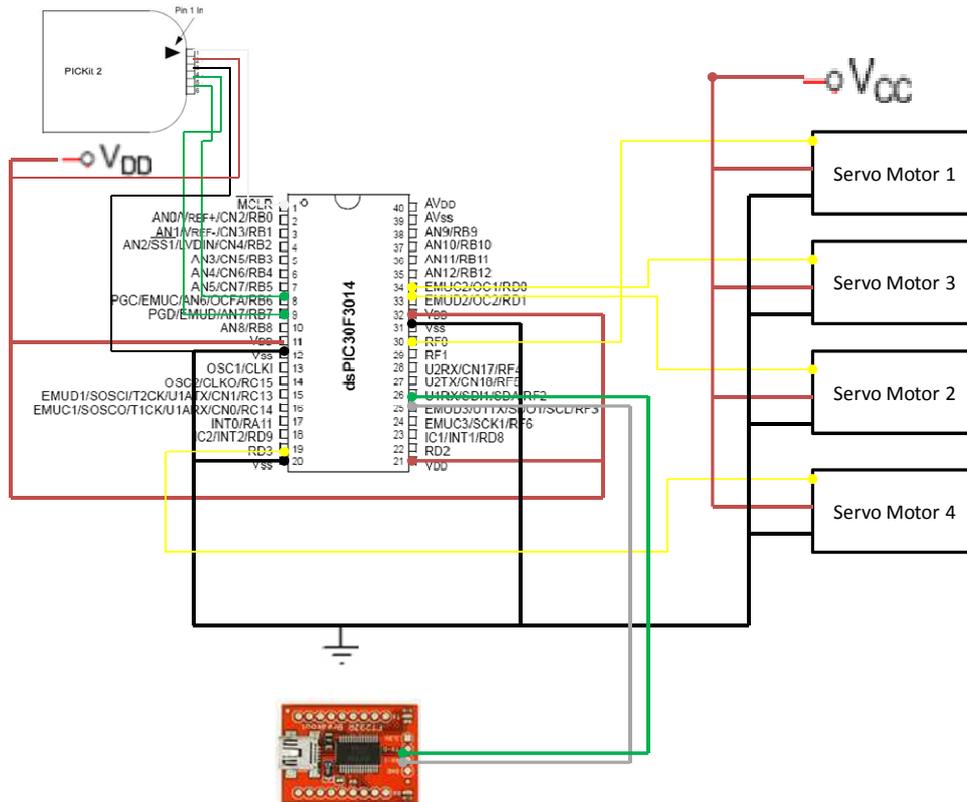


Fig. 6.4. Prototipo comparativo realizado para manipular trayectorias de un robot de 4 grados de libertad.

El objeto de este prototipo es primeramente tener un modelo el cual pudiera realizar diferentes trayectorias véase fig. (6.2) y fig. (6.3). De esta forma se podrá comparar el rango de voltajes del motor, la sensibilidad del motor sin carga (servomotor) y la linealidad debido a problemas de inercia.

Para el desarrollo del proyecto fue necesario utilizar circuitos que pueda modelar físicamente los diagramas de bloque obtenidos. De esta forma se utilizan amplificadores operacionales para poder hacer esta equivalencia véase fig. (6.5) y fig. (6.6).

Ecuación 6.1 $U=Kp e$

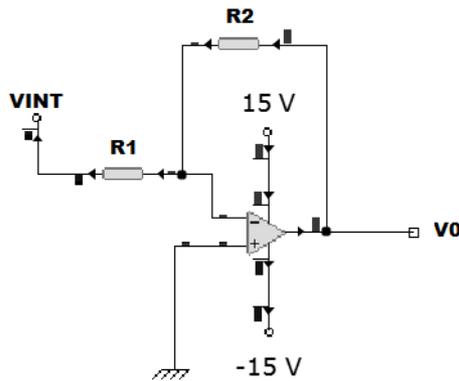


Fig. 6.5. Como se puede observar es un amplificador operacional inversor el cual se rige por la formula:

Ecuación 6.2 $v0 = -\frac{rf}{r} vi$

COMPARADOR

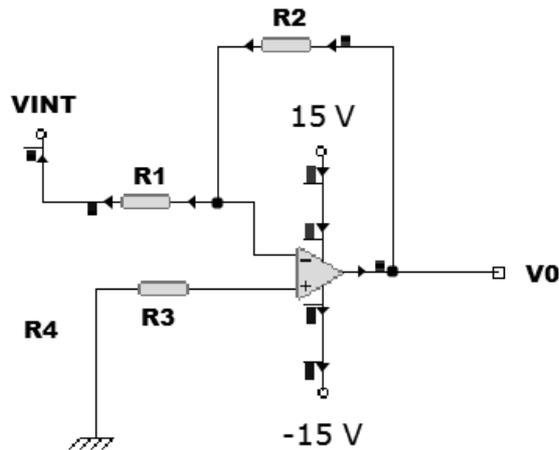


Fig. 6.6. En este esquema se representa el comparador en el diagrama de bloques es decir es aquel tomara la función tanto de restar como de retroalimentar la señal de voltaje deseado con la real.

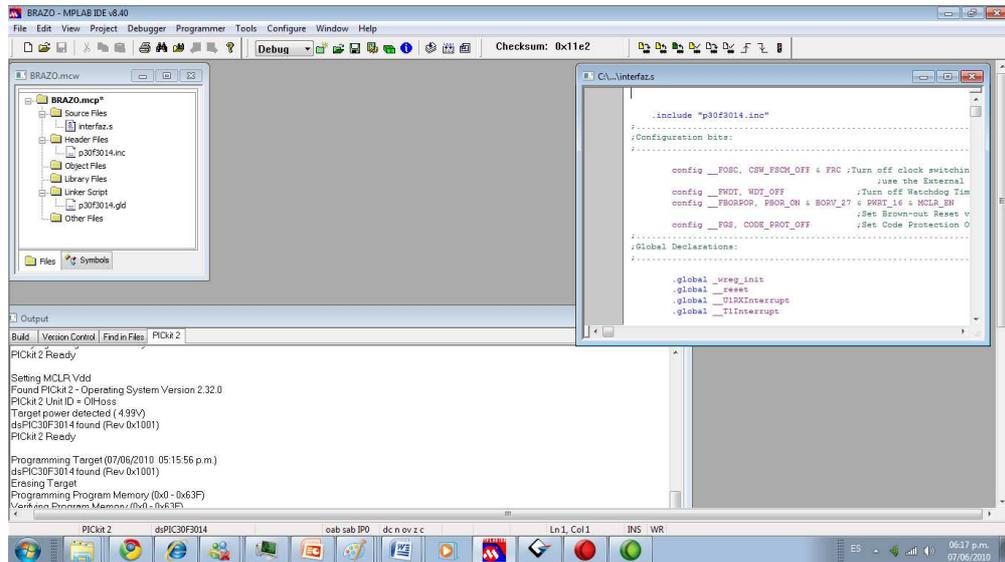


Fig. 6.5. Entorno de trabajo con Mplab.

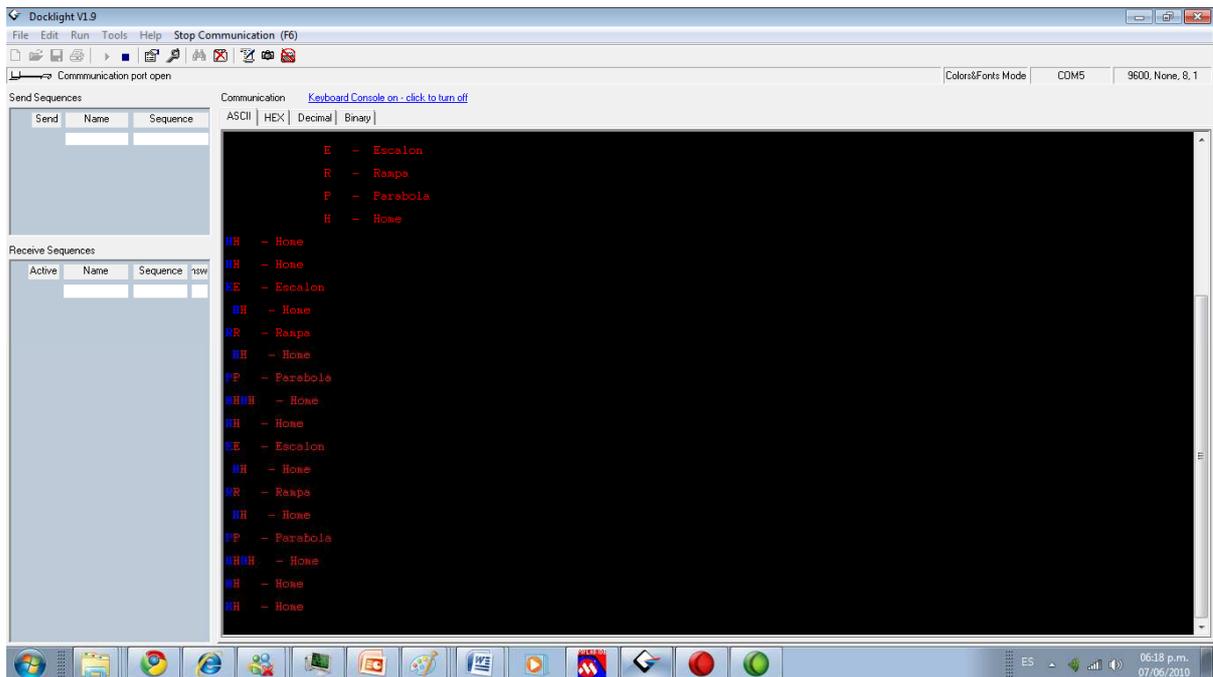


Fig. 6.6. Interfaz entre robot y ordenador con rutinas programadas de trabajo.

ANEXO 2.

Programa No 1 realizado en DsPic 30f3014 P para realización de pruebas PWM.

START:

NOP

INC2 OC1RS

INC2 OC1RS

INC2 OC1RS

INC2 OC1RS

INC2 OC1RS

INC2 OC1RS

CALL RETARDO5MS

NOP

NOP

GOTO START

PWM_CH1_INIT:

CLR OC1CON ; Turn off Output Compare 1 Module.;DESABILITA MODULO PWM

MOV #1000, w0 ;ESTE ES EL TIEMPO EN ALTO Initialize Duty Cycle to 0x0060,this

is the Uptime

MOV w0, OC1RS ;SE ESCRIBE EN EL BUFFER Write duty cycle buffer register

MOV w0, OC1R ; Write OC1R to initial duty cycle value

MOV #0x0006, w0 ;SE UTILIZA PARA COMPARARLO CON LO QUE HAY Load

the working register with the new

MOV w0, OC1CON ; compare mode and write to OC1CON

MOV #0x911a,w0 ;periodo completo 50 hz --20ms ESTE ES TIEMPO BASE...

Initialize PR2 with 0xFFFF,this is the

MOV w0, PR2

CLR TMR2 ; frequency of the pwm.

;

BSET T2CON, #TON;REGISTRO ESPECIALIZADO HABILITA MODULO DS PIC

"TON"

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

RETURN

PWM_CH2_INIT:

```
CLR    OC2CON ; Turn off Output Compare 2 Module.
MOV    #1000, w0 ; Initialize Duty Cycle to 0x0060
MOV    w0, OC2RS ; Write duty cycle buffer register
MOV    w0, OC2R ; Write OC2R to initial duty cycle value
MOV    #0x000E, w0 ; Load the working register with the new
MOV    w0, OC2CON ; compare mode and write to OC2CON
MOV    #36336, w0 ; Initialize PR2 with 0x00BF
MOV    w0, PR3;REGISTROS DEL MISMO TIMER...
CLR    TMR3;TMR DE HABILITACIÓN...LO LIMPIAMOS
MOV    #1023,W0;
MOV    w0, PR3;REGISTROS DEL MISMO TIMER...

BSET   T3CON, #TON;HABILITACION DE PWM COMO TIMER TIMER ON TON
RETURN
```

RETARDO:

```
MOV    #0X03E8,W0
MOV    W0,N2

LOOP1:
MOV    #0X1D52,W0
MOV    W0,N1

LOOP:
DEC    N1
BRA    Z,NEXT
GOTO  LOOP

NEXT:
DEC    N2
BRA    Z,NEXT1
GOTO  LOOP1

NEXT1:
RETURN
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

RETARDOMEDIOSEG:

MOV #0X01F4,W0

MOV W0,N2

OTHER1:

MOV #0X01D52,w0

MOV W0,N1

OTHER:

DEC N1

BRA Z,SIG

GOTO OTHER

SIG:

DEC N2

BRA Z,SIG1

GOTO OTHER1

SIG1:

RETURN

RETARDO5MS:

MOV #0X927C,W0

MOV W0,N2

LOOP0:

DEC N2

BRA Z,NEXT1X

GOTO LOOP0

NEXT1X:

RETURN

.end

Programa Número 2 Prototipo Robot 4 grados de Libertad.

El primer programa corresponde a la realización del primer prototipo del Robot de 4 grados de Libertad aunque este robot era constituido por un solo DSPIC 30f3014P, este carecía de diferentes posibilidades para poder controlar cada articulación por medio de control Digital y de sincronización con diferentes dispositivos por lo cual no se le pondrá tanta atención como lo es el caso del segundo prototipo.

```
.include "p30f3014.inc"
;.....
;Configuration bits:
;.....

config __FOSC, CSW_FSCM_OFF & FRC ;Turn off clock switching and fail-safe clock monitoring and
                                ;use the External Clock as the system clock
config __FWDT, WDT_OFF           ;Turn off Watchdog Timer
config __FBORPOR, PBOR_ON & BORV_27 & PWRT_16 & MCLR_EN
                                ;Set Brown-out Reset voltage and and set Power-up Timer to 16msecs
config __FGS, CODE_PROT_OFF     ;Set Code Protection Off for the General Segment
;.....
; Global Declarations:
;.....

.global _wreg_init
.global __reset
        .global __U1RXInterrupt
        .global __T1Interrupt

        .section variables bss, near
```


ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

.byte 'H','o','m','e'

Nombre_Escalon:

.byte 'E','s','c','a','l','o','n',0x0D,0X0D

Nombre_Rampa:

.byte 'R','a','m','p','a',0x0D,0X0D

Nombre_Parabola:

.byte 'P','a','r','a','b','o','l','a',0X0D,0X0D

Nombre_Home:

.byte 'H','o','m','e',0X0D,0X0D

.....

;Code Section in Program Memory

.....

.text ;Start of Code section

__reset:

```
MOV #__SP_init, W15 ;Initalize the Stack Pointer
MOV #__SPLIM_init, W0 ;Initialize the Stack Pointer Limit Register
MOV W0, SPLIM
NOP ;Add NOP to follow SPLIM initialization
```

.....

;Subroutine: Initialization of W registers to 0x0000

.....

_wreg_init: NOP

NOP

```
CLR W0
MOV W0, W14
REPEAT #12
MOV W0, [++W14]
CLR W14
```

```
MOV #0xFFFF, W0 ; SE DESABILITAN LOS COMPARADORES ANALOGICOS
MOV W0, ADPCFG ; SE DESABILITAN LOS COMPARADORES ANALOGICOS
```

INICIO:

```
MOV #0x0000, W1 ; SE INICIALIZAN LOS PUERTOS COMO SALIDA
MOV W1, TRISF ; SE INICIALIZAN LOS PUERTOS COMO SALIDA
MOV W1, TRISD ; SE INICIALIZAN LOS PUERTOS COMO SALIDA

CALL TIMER01_INIT
CALL UART1_INIT ; INICIALIZA LA UART 1 POR RECEPCION
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
CALL PWM_CH1_INIT ; INICIALIZA EL MODULO 1 DE PWM
CALL PWM_CH2_INIT ; INICIALIZA EL MODULO 2 DE PWM
```

```
BSET FLAG,#0 ;ENCIENDE EL BIT 0 DEL REGISTRO FLAG
```

```
MOV #143,W0 ; CARGA EL VALOR PARA LA FRECUENCIA DE 50 Hz
MOV W0,ANCIC ; AL REGISTRO DE ANCHO DE CICLO
MOV W0,COPANCIC ; HACE UNA COPIA DEL REGISTRO DE ANCHO DE CICLO
NOP
MOV #5,W0 ; CARGA EL VALOR DE HOME PARA EL SERVOMOTOR S1
MOV W0,CICTRA ; AL REGISTRO DEL CICLO DE TRABAJO
MOV #6,W0 ; CARGA EL VALOR DE HOME PARA EL SERVOMOTOR S4
MOV W0,CICTRA2 ; AL REGISTRO DEL CICLO DE TRABAJO 2
GOTO START
```

START:

```
CALL PRESENTACION
GOTO ESPERA_DATO
```

```
*****
;
;ETAPA DE LA PRESENTACION DE LA INTERFAZ
*****
;
```

PRESENTACION:

```
MOV #tblpage (Interfaz),W2 ;LLAMA TABLA INTERFAZ
MOV W2,TBLPAG
MOV #tbloffset (Interfaz),W2
MOV #22,W6
DO W6,MENU
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
MOV #' ',W0 ; CARACTER DE ESPACIO ENTRE CADA LETRA
MOV W0,U1TXREG
CALL RETARDO5MS
```

MENU:

```
NOP
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
CALL RETARDO5MS

MOV #0X0D,W0 ;SALTO DE LINEA
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV W0,U1TXREG
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG

MOV #tblpage (Menu),W2 ;LLAMA TABLA MENU
MOV W2,TBLPAG
MOV #tbloffset (Menu),W2
MOV #21,W6
DO W6,TECLA
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

TECLA:

```
NOP
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
CALL RETARDO5MS

MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG

MOV #tblpage (Tecla),W2 ;LLAMA TABLA MENU
MOV W2,TBLPAG
MOV #tbloffset (Tecla),W2
MOV #27,W6
DO W6,FUN_ESCALON
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

FUN_ESCALON:

```
NOP
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
CALL RETARDO5MS

MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV #tblpage (Escalon),W2 ;LLAMA TABLA Escalon
MOV W2,TBLPAG
MOV #tbloffset (Escalon),W2
MOV #27,W6
DO W6,FUN_RAMPA
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

FUN_RAMPA:

```
NOP
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
CALL RETARDO5MS
```

```
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
```

```
MOV #tblpage (Rampa),W2 ;LLAMA TABLA Rampa
MOV W2,TBLPAG
MOV #tbloffset (Rampa),W2
MOV #25,W6
DO W6,FUN_PARABOLA
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

FUN_PARABOLA:

```
NOP
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
CALL RETARDO5MS
```

```
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
```

```
MOV #tblpage (Parabola),W2 ;LLAMA TABLA Parabola
MOV W2,TBLPAG
MOV #tbloffset (Parabola),W2
MOV #28,W6
DO W6,HOME
TBLRDL.B [W2++],W4
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

HOME:

```
NOP
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
CALL RETARDO5MS
```

```
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
```

```
MOV #tblpage (Home),W2 ;LLAMA TABLA Home
MOV W2,TBLPAG
MOV #tbloffset (Home),W2
MOV #24,W6
DO W6,FIN_DO
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

FIN_DO:

```
NOP
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
MOV #0X0D,W0 ;SALTO DE LINEA
MOV W0,U1TXREG
CALL RETARDO5MS
RETURN
```

```
*****
;
;ETAPA DE ADQUISICION DE DATOS DE LA INTERRUPCION
*****
;
ESPERA_DATO:
```

```
BTSS BANDERA,#1 ; PREGUNTA POR EL BIT 1 DEL REGISTRO BANDERA
GOTO ESPERA_DATO ; SI ESTA EN BAJO REGRESA HASTA EL CAMBIO DE
ESTADO
BCLR BANDERA,#1 ; SI ESTA EN ALTO, APAGA LA BANDERA
GOTO MENU_TECLAS ; VE AL MENU DE LAS TECLAS
```

MENU_TECLAS:

```
MOV REGISTRO,W0
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV #'E',W1 ; HACE UNA COMPARACION DE LO QUE
CPSEQ W1,W0 ; LLEGO DE LA INTERRUPCION, CON LA LETRA E
GOTO COMP_R ; SI NO ES IGUAL SIGUE COMPARANDO CON
OTRAS LETRAS

GOTO TRAY_ESCALON ; SI ES IGUAL REALIZA LA TRAYECTORIA ESCALON

COMP_R:

MOV #'R',W1
CPSEQ W1,W0
GOTO COMP_P
GOTO TRAY_RAMPA

COMP_P:

MOV #'P',W1
CPSEQ W1,W0
GOTO COMP_H
GOTO TRAY_PARABOLA

COMP_H:

MOV #'H',W1
CPSEQ W1,W0
GOTO ESPERA_DATO
GOTO TRAY_HOME

TRAY_ESCALON:

NOP
MOV #tblpage (Nombre_Escalon),W2 ; LLAMA TABLA Home
MOV W2,TBLPAG
MOV #tbloffset (Nombre_Escalon),W2
MOV #14,W6
DO W6,FIN_ESCALON
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS

FIN_ESCALON:

MOV #5,W0
MOV W0,CICTRA ; SERVO 1 = 0° CONFORME AL EJE DEL MOTOR
MOV #3735,W0
MOV W0,OC2RS ; SERVO 2 = 157°
CALL RETARDO5MS
MOV #1245,W0
MOV W0,OC1RS ; SERVO 3 = 45 °
CALL RETARDO5MS
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV #10,W0
MOV W0,CICTRA2 ; SERVO 4 = 180 °
CALL RETARDO5MS
MOV #1123,W0
MOV W0, OC1RS ; SERVO 3 = 45 °
CALL RETARDO5MS
MOV #3330,W0
MOV W0, OC2RS ; SERVO 2 = 157°
CALL RETARDO5MS
MOV #1000,W0
MOV W0, OC1RS ; SERVO 3 = 45 °
CALL RETARDO5MS
MOV #2950,W0
MOV W0, OC2RS ; SERVO 2 = 157°

CALL RETARDO5MS
CLR REGISTRO
GOTO ESPERA_DATO
```

TRAY_RAMPA:

```
NOP
MOV #tblpage (Nombre_Rampa),W2 ;LLAMA TABLA Home
MOV W2,TBLPAG
MOV #tbloffset (Nombre_Rampa),W2
MOV #12,W6
DO W6,FIN_RAMPA
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

FIN_RAMPA:

```
MOV #5,W0
MOV W0,CICTRA ; SERVO 1 = 0° CONFORME AL EJE DEL MOTOR
MOV #11,W0
MOV W0,CICTRA ; SERVO 1 = 90°
CALL RETARDO5MS
MOV #4042,W0
MOV W0, OC2RS ; SERVO 2 = 157°
CALL RETARDO5MS
MOV #1245,W0
MOV W0, OC1RS ; SERVO 3 = 45 °
CALL RETARDO5MS
MOV #17,W0
MOV W0,CICTRA2 ; SERVO 4 = 180 °
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
CALL RETARDO5MS
MOV #3735,W0
MOV W0, OC2RS ; SERVO 2 = 135°
CALL RETARDO5MS
MOV #1000,W0
MOV W0, OC1RS ; SERVO 3 = 0°
```

```
CALL RETARDO5MS
CLR REGISTRO
GOTO ESPERA_DATO
```

TRAY_PARABOLA:

```
NOP
MOV #tblpage (Nombre_Parabola),W2 ;LLAMA TABLA Home
MOV W2,TBLPAG
MOV #tbloffset (Nombre_Parabola),W2
MOV #14,W6
DO W6,FIN_PARABOLA
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

FIN_PARABOLA:

```
NOP

MOV #5,W0
MOV W0,CICTRA ; SERVO 1 = 0° CONFORME AL EJE DEL MOTOR
MOV #11,W0
MOV W0,CICTRA ; SERVO 1 = 90°
CALL RETARDO5MS
MOV #3735,W0
MOV W0, OC2RS ; SERVO 2 = 135 °
CALL RETARDO5MS
MOV #1245,W0
MOV W0, OC1RS ; SERVO 3 = 45 °
CALL RETARDO5MS
MOV #3330,W0
MOV W0, OC2RS ; SERVO 2 = 112 °
CALL RETARDO5MS
MOV #1000,W0
MOV W0, OC1RS ; SERVO 3 = 0°
CALL RETARDO5MS
MOV #2950,W0
MOV W0, OC2RS ; SERVO 2 = 90 °
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
CALL RETARDO5MS
MOV #12,W0
MOV W0,CICTRA ; SERVO 1 = 105 °
CALL RETARDO5MS
MOV #3330,W0
MOV W0, OC2RS ; SERVO 2 = 112 °
CALL RETARDO5MS
MOV #1123,W0
MOV W0, OC1RS ; SERVO 3 = 23 °
CALL RETARDO5MS
MOV #14,W0
MOV W0,CICTRA ; SERVO 1 = 135 °
CALL RETARDO5MS
MOV #1370,W0
MOV W0, OC1RS ; SERVO 3 = 67 °
CALL RETARDO5MS
MOV #16,W0
MOV W0,CICTRA ; SERVO 1 = 157 °
CALL RETARDO5MS
MOV #3735,W0
MOV W0, OC2RS ; SERVO 2 = 135 °
CALL RETARDO5MS
MOV #1245,W0
MOV W0, OC1RS ; SERVO 3 = 45 °
CALL RETARDO5MS
MOV #8,W0
MOV W0,CICTRA2 ; SERVO 4 = 54 °
CALL RETARDO5MS
MOV #18,W0
MOV W0,CICTRA ; SERVO 1 = 180 °
CALL RETARDO5MS
MOV #9,W0
MOV W0,CICTRA2 ; SERVO 4 = 72 °
CALL RETARDO5MS
MOV #10,W0
MOV W0,CICTRA2 ; SERVO 4 = 90 °

CALL RETARDO5MS
CLR REGISTRO
GOTO ESPERA_DATO
```

TRAY_HOME:

```
NOP
MOV #tblpage (Nombre_Home),W2 ;LLAMA TABLA Home
MOV W2,TBLPAG
MOV #tbloffset (Nombre_Home),W2
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV #11,W6
DO W6,FIN_HOME
TBLRDL.B [W2++],W4
MOV W4,W0
MOV W0,U1TXREG
CALL RETARDO5MS
```

FIN_HOME:

```
NOP
```

```
MOV #5,W0
MOV W0,CICTRA ; SERVO 1 = 0° CONFORME AL EJE DEL MOTOR
```

```
MOV #4350,W0
MOV W0, OC2RS ; SERVO 2 = 180° CONFORME AL EJE D EL MOTOR
```

```
MOV #2490,W0
MOV W0, OC1RS ; SERVO 3 = 90° CONFORME AL EJE DE L MOTOR
```

```
MOV #6,W0
MOV W0,CICTRA2 ;SERVO 4 = 18° CONFORME AL EJE DE L MOTOR
```

```
CLR REGISTRO
GOTO ESPERA_DATO
```

```
*****
;
```

```
;INTERRUPCION POR RECEPCION EN LA UART1
```

```
*****
;
```

```
__U1RXInterrupt:
```

```
PUSH.S
MOV U1RXREG,W0
MOV W0,U1TXREG
BSET BANDERA,#1
MOV W0,REGISTRO
```

```
BCLR IFS0,#U1RXIF
POP.S
RETFIE
```

```
*****
;
```

```
;INTERRUPCION DE LA TIMER 01
```

```
*****
;
```

```
__T1Interrupt:
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
BTSS FLAG,#0
GOTO PROCESO

CARGA:

MOV ANCIC, W0
MOV W0,COPANCIC
MOV CICTRA, W0
MOV W0,COPCICTRA

MOV CICTRA2,W0
MOV W0,COPCICTRA1

BSET PORTF,#1
BSET PORTF,#0
BSET PORTD,#2
BSET PORTD,#3
BCLR FLAG,#0

PROCESO:

DEC COPANCIC
CP0 COPANCIC
BRA NZ,DECRE
BSET FLAG,#0
GOTO FIN

DECRE:

DEC COPCICTRA
CP0 COPCICTRA
BRA NZ,NEXT

;
BCLR PORTF,#0
BCLR PORTF,#1
BCLR PORTD,#2
;
BCLR PORTD,#3

NEXT:

DEC COPCICTRA1
CP0 COPCICTRA1
BRA NZ,FIN

;
BCLR PORTF,#0
BCLR PORTF,#1
;
BCLR PORTD,#2
BCLR PORTD,#3
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

FIN:

BCLR IFS0, #T1IF ; Reset Timer1 interrupt flag

RETFIE ; Return from ISR

;INICIALIZACION DE LA UART1

UART1_INIT:

```
MOV          #baudrateU1,W0 ; Set Baudrate
MOV          W0,U1BRG
BSET         IPC2,#U1TXIP2 ; Set UART TX interrupt priority
BCLR         IPC2,#U1TXIP1 ;
BCLR         IPC2,#U1TXIP0 ;
BSET         IPC2,#U1RXIP2 ; Set UART RX interrupt priority
BCLR         IPC2,#U1RXIP1 ;
BCLR         IPC2,#U1RXIP0 ;
CLR          U1STA
MOV          #0x8800,W0 ; Enable UART for 8-bit data,
; no parity, 1 STOP bit,
; no wakeup
MOV          W0,U1MODE
BSET         U1STA,#UTXEN ; Enable transmit
;BSET IEC0,#U1TXIE ; Enable transmit interrupts
BSET         IEC0,#U1RXIE ; Enable receive interrupts
RETURN
```

;INICIALIZACION DEL PWM CANAL 1

PWM_CH1_INIT:

```
CLR          OC1CON ; Turn off Output Compare 1 Module.
MOV          #2390, w0 ; Initialize Duty Cycle
MOV          w0, OC1RS ; Write duty cycle buffer register
MOV          w0, OC1R ; Write OC1R to initial duty cycle value
MOV          #0x0006, w0 ; Load the working register with the new
MOV          w0, OC1CON ; compare mode and write to OC1CON
MOV          #36336, w0 ; Initialize PR2 with 0x00BF; AQUI SE CAMBIA EL CICLO DE TRABAJO
CLR          TMR2
MOV          w0, PR2
BSET         T2CON, #TON
RETURN
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

;INICIALIZACION DEL PWM CANAL 2

PWM_CH2_INIT:

```
CLR    OC2CON ; Turn off Output Compare 1 Module.
MOV    #4350, w0 ; Initialize CICLO DE TRABAJO
MOV    w0, OC2RS ; Write duty cycle buffer register
MOV    w0, OC2R ; Write OC1R to initial duty cycle value
MOV    #0x000E, w0 ; Load the working register with the new
MOV    w0, OC2CON ; compare mode and write to OC1CON
MOV    #36336, w0 ; Initialize PR2 with 0x00BF;
MOV    w0, PR3; ANCHO DE CICLO QUE SE MANTIENE CONSTANTE 16 BITS
CLR    TMR3
```

```
BSET   T3CON, #TON
RETURN
```

;INICIALIZACION DEL TIMER 1 PARA PWM

TIMER01_INIT:

; The following code example will enable Timer1 interrupts,

; load the Timer1 Period register and start Timer1.

; When a Timer1 period match interrupt occurs, the interrupt

; service routine must clear the Timer1 interrupt status flag

; in software.

```
CLR T1CON ; Stops the Timer1 and reset control reg.
CLR TMR1 ; Clear contents of the timer register
MOV #0x00FF, w0 ; Load the Period register
MOV w0, PR1 ; with the value 0xFFFF
MOV #0B0000000000000000,W0
MOV W0,T1CON
BSET IPC0, #T1IP0 ; Setup Timer1 interrupt for
BCLR IPC0, #T1IP1 ; desired priority level
BCLR IPC0, #T1IP2 ; (this example assigns level 1 priority)
BCLR IFS0, #T1IF ; Clear the Timer1 interrupt status flag
BSET IEC0, #T1IE ; Enable Timer1 interrupts
BSET T1CON, #TON ; Start Timer1 with prescaler settings
```

; at 1:1 and clock source set to

; the internal instruction cycle

RETURN

RETARDOMEDIOSEG:

```
MOV #0x01F4 ,W0
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV W0,N2
OTHER1:
MOV #0X1D52,W0
MOV W0,N1
OTHER:
DEC N1
BRA Z,SIG
GOTO OTHER
SIG:
DEC N2
BRA Z,SIG1
GOTO OTHER1
SIG1:
RETURN
```

```
RETARDO5uS:
REPEAT #150
NOP
RETURN
```

```
RETARDO5MS:
```

```
MOV #0X927C ,W0
MOV W0,N2
```

```
LOOP:
```

```
DEC N2
BRA Z,NEXT1
GOTO LOOP
```

```
NEXT1:
```

```
RETURN
```

```
RETARDO1SEG:
```

```
MOV #0x03E8 ,W0
MOV W0,N2
```

```
OTRA1:
```

```
MOV #0X1D52,W0
MOV W0,N1
```

```
OTRA:
```

```
DEC N1
BRA Z,SIGUIENTE
GOTO OTRA
```

```
SIGUIENTE:
```

```
DEC N2
```

```
BRA Z,SIGUIENTE1
GOTO OTRA1
SIGUIENTE1:
RETURN
```

```
.END
```

Programa No 3 Modulo por cada una de las articulaciones del robot de 4 grados de libertad.

;(Lo único que cambian son los BTSS Y BTSC de f0 y f1).

;(El programa se debe de copiar tal cual para poder utilizar el programa).

```
.include "p30f3014.inc"
;PROGRAMA PWM PARA QUE FUNCIONE EL HITEC HS-311
;el puerto d2 indica a la salida que el dato fue recibido
;el puerto d3 indica si el signo es positivo o negativo
;el puerto b indica los 8 bits
;el puerto f es la configuracion para saber si el dato recibido
;de b es para el o para otro servo.
;.....
;Configuration bits:
;.....
config __FOSC, CSW_FSCM_OFF & FRC ;XT_PLL16 ;Turn off clock switching and
;fail-safe clock monitoring and
;use the External Clock as the
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```

                                ;system clock
config __FWDT, WDT_OFF          ;Turn off Watchdog Timer
config __FBORPOR, PBOR_ON & BORV_27 & PWRT_16 & MCLR_EN
                                ;Set Brown-out Reset voltage and
                                ;and set Power-up Timer to 16msecs
config __FGS, CODE_PROT_OFF     ;Set Code Protection Off for the
                                ;General Segment
;.....
;Global Declarations:
;.....
    .global _wreg_init
    .global __reset
;.....
;Uninitialized variables in Near data memory (Lower 8Kb of RAM)
;.....
                                .section HOLA, bss, near
N1:                                .SPACE 2
N2:                                .SPACE 2
VAR1: .SPACE 2
VAR2: .SPACE 2
VARBOCOP: .SPACE 2
;.....
;Code Section in Program Memory
;.....
.text

__reset:
    MOV #__SP_init, W15    ;Initalize the Stack Pointer
    MOV #__SPLIM_init, W0  ;Initialize the Stack Pointer Limit Register
    MOV W0, SPLIM
    NOP                    ;Add NOP to follow SPLIM initialization
    ;Call _wreg_init subroutine
                                ;Optionally use RCALL instead of CALL
_wreg_init:    NOP
                NOP
    CLR        W0
    MOV        W0, W14;LA DIRECCION 0X800 Y DE AHI CUENTA W14
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
REPEAT #12
MOV     W0, [++W14];ESTO SOLO ESPARA LIMPIAR LOS REGISTROS W'S
CLR     W14

        MOV     #0xffff, W0           ;CONFIGURA EL PUERTO B
        MOV     W0, ADPCFG           ;COMO PUERTO DIGITAL

; BSET TRISD,#0;DECLARA COMO SALIDA PORTD,0
```

PUERTOS:

```
        MOV     #0B01111111,W0
MOV     W0,TRISB;DECLARAS PUERTO B COMO ENTRADA MENOS EL ULTIMO...
        BCLR    TRISD,#0
        BCLR    TRISD,#1;SALIDAS PUERTO D 0 Y 1
;BSET TRISF,#0 SE DESABILITO NO PUEDE SER ENTRADA
BSET TRISF,#0;DECLARA ENTRADA F0
BSET TRISF,#1;DECLARA ENTRADA F1
BSET TRISD,#3;SERVIRA PARA COMAPARAR NEGADOS...(-)
BCLR TRISD,#2
BCLR TRISD,#8
CALL    PWM_CH1_INIT
        CALL    PWM_CH2_INIT
```

GOTO START; ASEGURA IR A START:

;BSET TRISF,#0 SE DESABILITARON PORQUE NO PUEDEN SER ENTRADAS

```
,*****
,*****
```

START;*****INICIO DE PROGRAMA*****

NOP

NOP

;;EJEMPLO CHECAR CON EL WATCH NO CREO QUE SE UTILICE

;MOV #3,W0

;MOV #4,W1

;SUBB W0,W1,W2;W2=65534,DESBORDAMIENTO....

;MOV #65535,W3;

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
;SUB W3,W2,W4;W4=1
;;FIN EJEMPLO CHECAR CON EL WATCH FIN NO CREO QUE SE UTILICE
;MOV #0,W0
MOV #90,W7;ESTE ES SOLO PARA DARLE CERO GRADOS AL INICIO
MOV #90,W0
;ADD W0,W1,W2;*****ESTE VA A SUMAR *****
;MOV W8,W0
;+SALE SOBRANDO++MOV W0,W7;{{{ESTO ES PARA ALMACENAR EL ;UTIMO VALOR YA
SEA POSITIVO O NEGATIVO...PARA OBTENER EL CERO.
;MOV W7,W0; CON ESTO VOY A ASEGURAR EL ULTIMO DATO...YA SEA POSITIVO O
NEGATIVO...
;+++++++AQUI EMPIEZA LA LECTURA DE BIT+++++++
MOV #18,W1;AQUI SE CARGA EL VALOR POR GRADO
MUL.UU W0,W1,W4
ADD #1000,W4;SEGUN YO ESTE ES EL REAJUSTE...PERO NO SUMA
ADD #200,W4;200 DECIMAL REAJUSTE PARA QUE DE BIEN Los CEROS GRADOS
CALL RETARDO5US
MOV W4,OC1RS; AQUI MANDA EL ULTIMO DATO--A LA INTERRUPCIÓN SIN EL PWM...
CALL RETARDO5US
NOP
NOP
REGRESA: ;ESTO SIRVE PARA IDENTIFICAR QUE SERVO QUEREMOS UTILIZAR

BTSC PORTF,#0;ESTO HACE EL BUS DE DATOS DEL 11-PUERTO F'S
GOTO YYY;EL BTSS BTSC ES PARA MODIFICAR EL CERO O 1.....{
BTSC PORTF,#1;
GOTO YYY;ESTO HACE EL BUS DE DATOS DEL 11-PUERTO F'S{

;DO #50,CONTEO;checar no me acuerdo si iba el do pero creo no ...
PWM1:

MOV PORTB,W0
MOV W0,VAROBOCOP;ESTE SE VE REFLEJADO EN EL PIN D0 ESPECIAL PARA PWM...
MOV VAROBOCOP,W0
CP0 W0
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

BRA Z,XXX

NOP

BTSS PORTD,#3; inidica si es positivo o negativo...

GOTO SALTIN;MOV PORTD,#3,W0;*****REPLAZO.MODIFICACION SIGNAL

RESTA::;CP0 W0

MOV #90,W1 ;BRA NZ,SALTIN

SUBB W1,W0,W2 ;*****ESTE VA A RESTAR*****

MOV W2,W0

GOTO BLADE

XXX:

SALTIN::ESTE SALTIN ES EL A DONDE TE MANDAN SI EL VALOR QUE SALE NO ES NEGATIVO.

;AQUI SE PUEDE COLOCAR UN VALOR PARA QUEDE CERO....

MOV #90,W1

ADD W0,W1,W2;*****ESTE VA A SUMAR *****

MOV W2,W0

BLADE:

MOV W0,W7;{{{ESTO ES PARA ALMACENAR EL UTIMO VALOR YA SEA POSITIVO O NEGATIVO...PARA OBTENER EL CERO.

YYY::{{{ ESTE ES EL QUE GUARDA EL VALOR A PRIMERA INSTANCIA....{{{

MOV W7,W0; CON ESTO VOY A ASEGURAR EL ULTIMO DATO...YA SEA POSITIVO O NEGATIVO...

;+++++++AQUI EMPIEZA LA LECTURA DE BIT+++++++

AND #0B1111111,W0;ESTO ES PARA QUE FILTRE DATOS QUE NO DESEAMOS DEL PUERTO B.

MOV #18,W1;AQUI SE CARGA EL VALOR POR GRADO

MUL.UU W0,W1,W4

ADD #1000,W4;SEGUN YO ESTE ES EL REAJUSTE...PERO NO SUMA

ADD #200,W4;200 DECIMAL REAJUSTE PARA QUE DE BIEN Los CEROS GRADOS

MOV W4,OC1RS

,*****

;ESTA PARTE SE DESABILITO PARA HACER EL ULTIMO PROGRAMA PWM...

;REGRESA2::;PARA ASEGURARSE

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
;BTSC PORTF,#0;ESTO HACE EL BUS DE DATOS DEL 11-PUERTO F'S
;GOTO REGRESA2;
;BTSC PORTF,#1;
;GOTO REGRESA2;ESTO HACE EL BUS DE DATOS DEL 11-PUERTO F'S
;*****
;ESTE MODULO NO SE UTILIZA PERO LO DEJAMOS AQUI,...
;*****+
;*****ESTO NO SE UTILIZA*****+
;*****+
;*****+
;PWM2:
;MOV PORTB,W0
;MOV W0,VAROBOCOP;ESTE SE VE REFLEJADO EN EL PIN D0 ESPECIAL PARA PWM...
;MOV VAROBOCOP,W0
;AND #0B11111111,W0;ESTO ES PARA QUE FILTRE DATOS QUE NO DESEAMOS DEL PUERTO B.
;MOV #18,W1;AQUI SE CARGA EL VALOR POR GRADO
;MUL.UU W0,W1,W4
;ADD #1000,W4
;ADD #200,W4;200 DECIMAL REAJUSTE PARA QUE DE BIEN Los CEROS GRADOS
;MOV W4,OC2RS
;NOP
;*****+
;*****+
;*****+
;*****+
;*****+
;*****+
GOTO REGRESA
;

;*****FIN DE PROGRAMA PRINCIPAL*****
;*****
;*****
PWM_CH1_INIT:
        CLR    OC1CON ; Turn off Output Compare 1 Module.;DESABILITA MODULO PWM
        MOV    #1000, w0 ;ESTE ES EL TIEMPO EN ALTO Initialize Duty Cycle to 0x0060,this
is the Uptime
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

```
MOV w0, OC1RS ;SE ESCRIBE EN EL BUFFER Write duty cycle buffer register
MOV w0, OC1R ; Write OC1R to initial duty cycle value
MOV #0x0006, w0 ;SE UTILIZA PARA COMPARARLO CON LO QUE HAY Load
the working register with the new
MOV w0, OC1CON ; compare mode and write to OC1CON
MOV #0x911a,w0 ;periodo completo 50 hz --20ms ESTE ES TIEMPO BASE...
Initialize PR2 with 0xFFFF,this is the
MOV w0, PR2 ;ademas sirve el 911a para servos hobbico
CLR TMR2 ; frequency of the pwm.
;
BSET T2CON, #TON;REGISTRO ESPECIALIZADO HABILITA MODULO DS PIC
"TON"
RETURN
```

PWM_CH2_INIT:

```
CLR OC2CON ; Turn off Output Compare 2 Module.
MOV #1000, w0 ; Initialize Duty Cycle to 0x0060
MOV w0, OC2RS ; Write duty cycle buffer register
MOV w0, OC2R ; Write OC2R to initial duty cycle value
MOV #0x0006, w0;#0X000E ; Load the working register with the new
MOV w0, OC2CON ; compare mode and write to OC2CON
MOV #0x911a, w0 ; Initialize PR2 with 0x00BF
MOV w0, PR3;REGISTROS DEL MISMO TIMER...
CLR TMR3;TMR DE HABILITACIÓN...LO LIMPIAMOS
;PARA LEDS ONLY; MOV #1023,W0;;este 1023 es para frecuencias altas como para leds
;PARA LEDS ONLY; MOV w0, PR3;REGISTROS DEL MISMO TIMER...

BSET T3CON, #TON;HABILITACION DE PWM COMO TIMER TIMER ON TON
RETURN
```

RETARDO:

```
MOV #0X03E8,W0
MOV W0,N2

LOOP1:
MOV #0X1D52,W0
MOV W0,N1
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

LOOP:

```
DEC      N1
BRA      Z,NEXT
GOTO    LOOP
```

NEXT:

```
DEC      N2
BRA      Z,NEXT1
GOTO    LOOP1
```

NEXT1:

```
RETURN
```

RETARDOMEDIOSEG:

```
MOV #0X01F4,W0
```

```
MOV W0,N2
```

OTHER1:

```
MOV #0X01D52,w0
```

```
MOV W0,N1
```

OTHER:

```
DEC N1
```

```
BRA Z,SIG
```

```
GOTO OTHER
```

SIG:

```
DEC N2
```

```
BRA Z,SIG1
```

```
GOTO OTHER1
```

SIG1:

```
RETURN
```

RETARDOS:

```
MOV #0X012C,W0
```

```
MOV W0,N2
```

OTHER1X:

```
MOV #0X01D52,w0
```

```
MOV W0,N1
```

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

OTHERX:

DEC N1

BRA Z,SIG

GOTO OTHERX

SIGX:

DEC N2

BRA Z,SIG1X

GOTO OTHER1X

SIG1X:

RETURN

RETARDO5MS:

MOV #0X927C,W0

MOV W0,N2

LOOP0:

DEC N2

BRA Z,NEXT1X

GOTO LOOP0

NEXT1X:

RETURN

RETARDO5US:

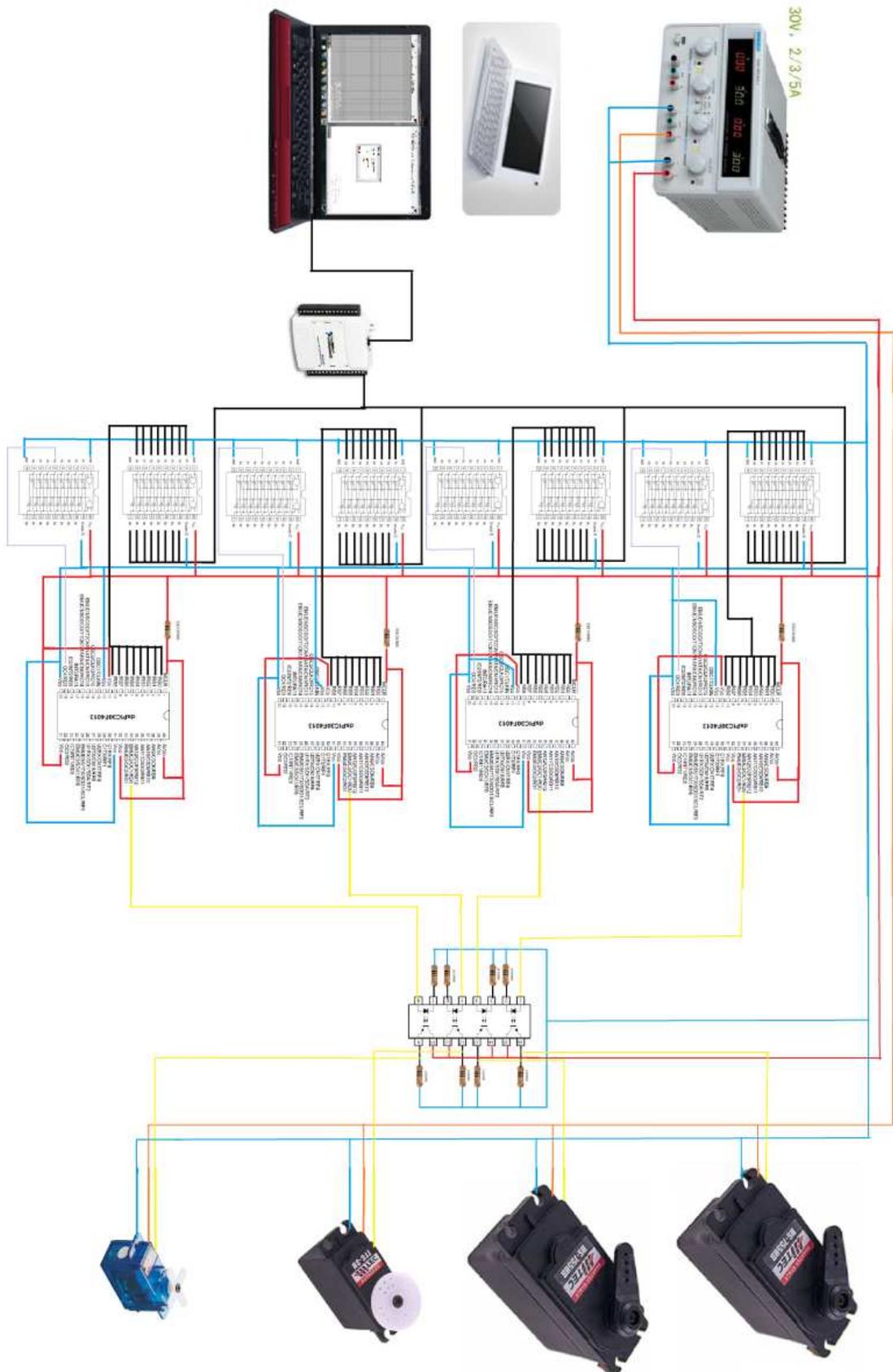
REPEAT #150

NOP

RETURN

.end

ANEXO 3.



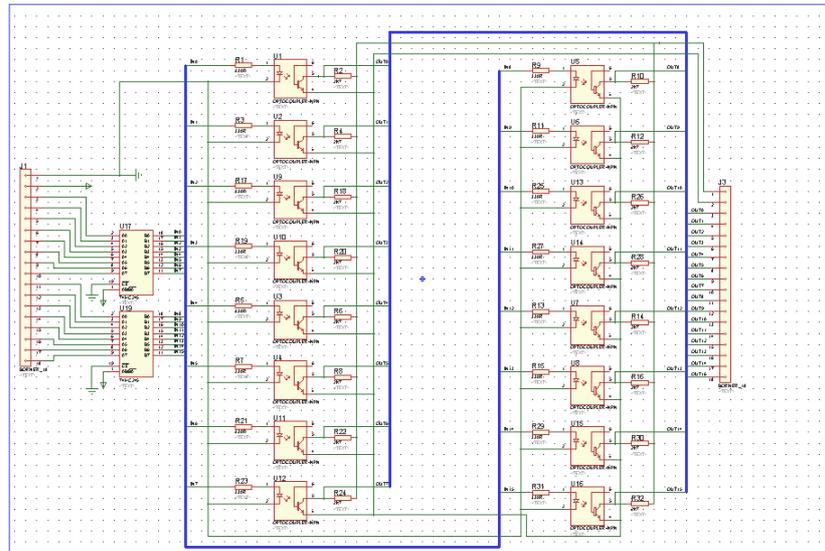


Diagrama general de Tarjeta de comunicación *Daq 6009-Irb 1600*.

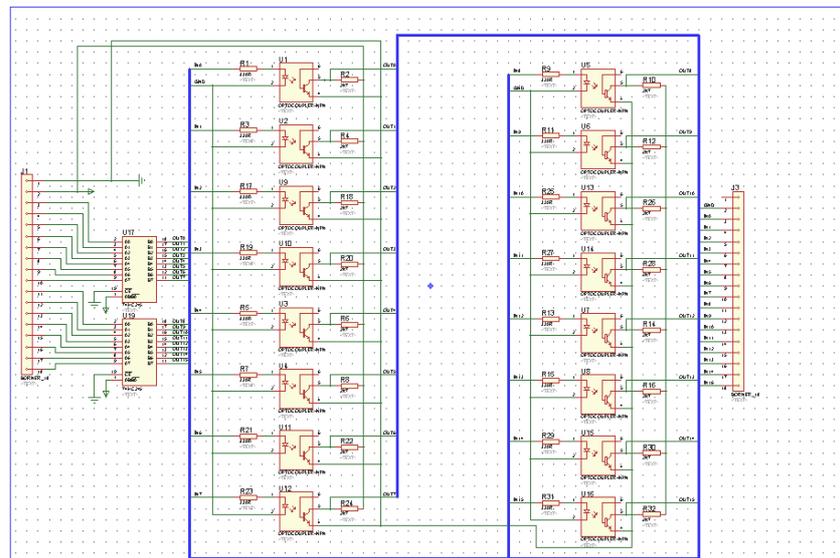
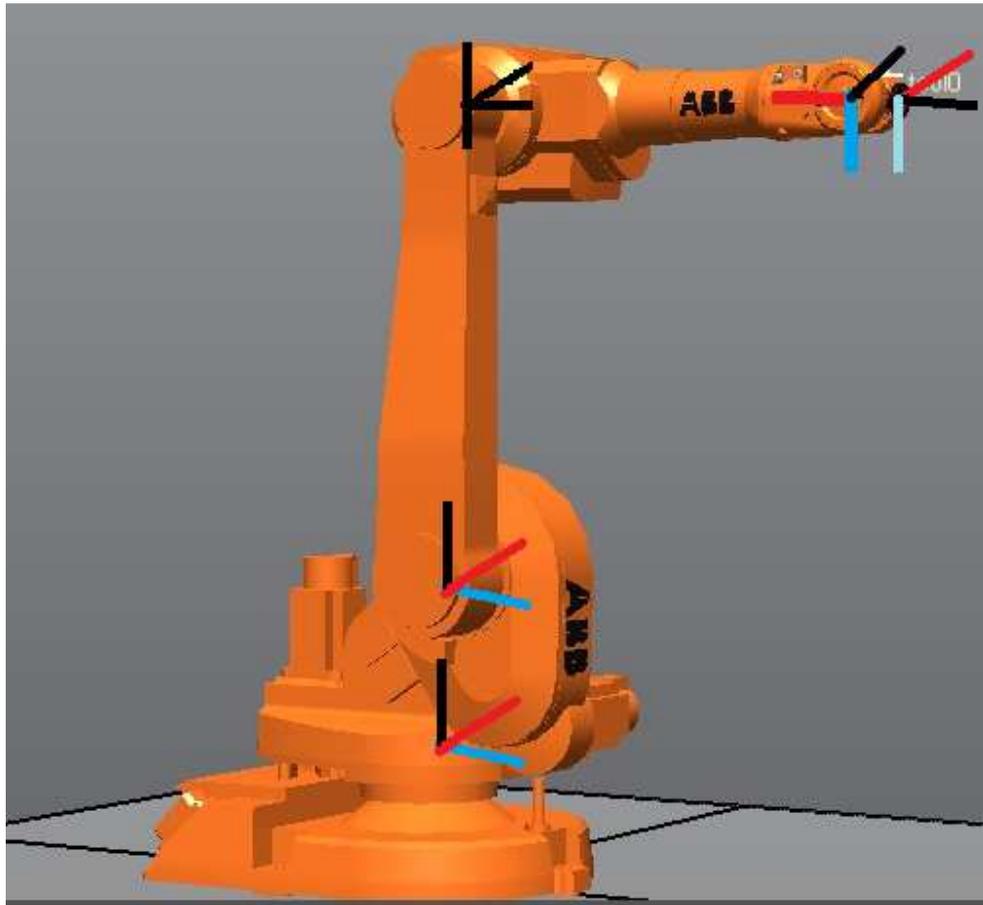


Diagrama general de Tarjeta de comunicación *Irb 1600-Daq 6009*

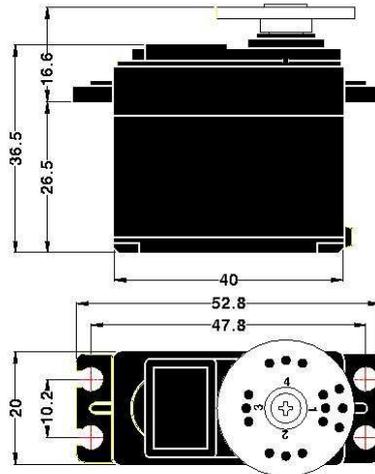


i	A_i	d_i	Alpha i	Theta i
1	.150	.4865	-90	.4865
2	.700	0	0	0
3	0	0	90	0
4	0	.600	-90	.600
5	0	0	90	0
6	0	.065	0	.065

PREPARED BY JUN HEE, LEE
UPDATE: APR 01, 2002

ANNOUNCED SPECIFICATION OF HS-311 STANDARD SERVO

1. TECHNICAL VALUE		
CONTROL SYSTEM	: +PULSE WIDTH CONTROL 1500μsec NEUTRAL	
OPERATING VOLTAGE RANGE	: 4.8V TO 6.0V	
TEST VOLTAGE	: AT 4.8V	AT 6.0V
OPERATING SPEED	: 0.19sec/60° AT NO LOAD	0.15sec/60° AT NO LOAD
STALL TORQUE	: 3.0kg.cm (42oz.in)	3.5kg.cm (48.6oz.in)
IDLE CURRENT	: 7.4mA AT STOPPED	7.7mA AT STOPPED
RUNNING CURRENT	: 160mA/60° AT NO LOAD	180mA/60° AT NO LOAD
STALL CURRENT	: 700mA	800mA
DEAD BAND WIDTH	: 5μsec	5μsec
OPERATING TRAVEL	: 40°/ONE SIDE PULSE TRAVELING 400μsec	
DIRECTION	: CLOCK WISE/PULSE TRAVELING 1500 TO 1900μsec	
MOTOR TYPE	: CORED METAL BRUSH	
POTENTIOMETER TYPE	: 4 SLIDER/DIRECT DRIVE	
AMPLIFIER TYPE	: ANALOG CONTROLLER & TRANSISTOR DRIVER	
DIMENSIONS	: 40x20x36.5mm (1.57x0.78x1.43in)	
WEIGHT	: 43g (1.51oz)	
BALL BEARING	: TOP/RESIN BUSHING	
GEAR MATERIAL	: RESIN	
HORN GEAR SPLINE	: 24 SEGMENTS/□5.76	
SPLINED HORNS	: SUPER/R-XA	
CONNECTOR WIRE LENGTH	: 300mm (11.81in)	
CONNECTOR WIRE STRAND COUNTER	: 40EA	
CONNECTOR WIRE GAUGE		



2. FEATURES

LONG LIFE POTENTIOMETER, TOP RESIN BUSHING

3. APPLICATIONS

AIRCRAFT 20-40 SIZE, STEERING AND THROTTLE SERVO FOR CARS, TRUCK AND BOATS

4. ACCESSORY & OPTION

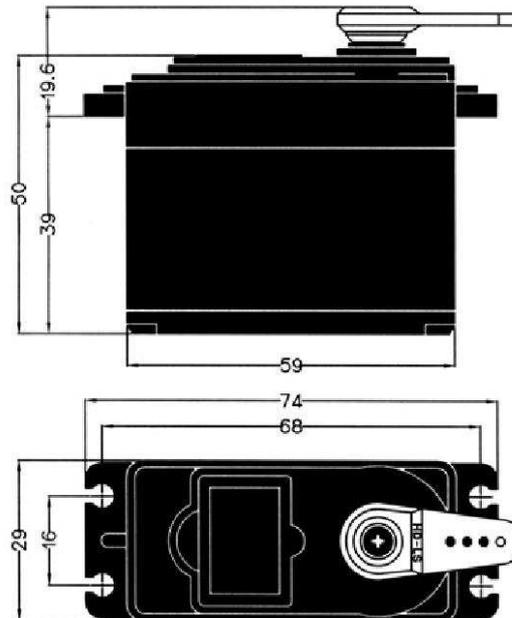
CASE SET/	GEAR SET/	HORN SET/
HS322T:1EA	HS322G1:1EA	R-XA:1EA
HS322M:1EA	HS322G2:1EA	
HS322L:1EA	HS322G3:1EA	
PH/T-2 2x30 NI:4EA	HS322G4:1EA	
	HS300RB:1EA	

HITEC RCD KOREA INC.

PREPARED BY JUN HEE, LEE
UPDATE: FEB 20, 2003

**GENERAL SPECIFICATION OF
HS-755HB QUARTER SCALE SERVO**

1. TECHNICAL VALUE	
CONTROL SYSTEM	: +PULSE WIDTH CONTROL 1500usec NEUTRAL
OPERATING VOLTAGE RANGE	: 4.8V TO 6.0V
OPERATING TEMPERATURE RANGE	: -20°C TO +60°C (-46°F TO +86°F)
TEST VOLTAGE	: AT 4.8V
OPERATING SPEED	: 0.28sec/60° AT NO LOAD
STALL TORQUE	: 11kg.cm(152.75oz.in)
STANDING TORQUE	: 8.8kg.cm(122.20oz.in)/5° HOLD OUT
IDLE CURRENT	: 8mA AT STOPPED
RUNNING CURRENT	: 230mA/60° AT NO LOAD RUNNING
STALL CURRENT	: 1500mA
DEAD BAND WIDTH	: 5usec
OPERATING TRAVEL	: 40°/ONE SIDE PULSE TRAVELING 400usec
DIRECTION	: CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec
MOTOR TYPE	: CORED/METAL BRUSH
POTENTIOMETER TYPE	: 6 SLIDER/INDIRECT DRIVE
AMPLIFIER TYPE	: ANALOG AMPLIFIER
DIMENSIONS	: 59x29x50mm(2.32x1.14x1.96in)
WEIGHT	: 110g(3.88oz)
BALL BEARING	: DUAL/MR106
GEAR MATERIAL	: HEAVY DUTY RESIN
HORN GEAR SPLINE	: 24 SEGMENTS/ø5.76
SPLINED HORNS	: HEAVY DUTY/HD-IS, HD-O, HD-LS, HD-IL, HD-LL
CONNECTOR WIRE LENGTH	: 300mm(11.81in)
CONNECTOR WIRE STRAND COUNTER	: 60EA
CONNECTOR WIRE GAUGE	: 22AWG



2. FEATURES
DURABLE HEAVY DUTY RESIN GEARS WITH BALL BEARINGS
WATER & DUST TIGHT

3. APPLICATIONS
FOR LARGE MODELS

HITEC RCD KOREA INC.



www.fairchildsemi.com

LM741

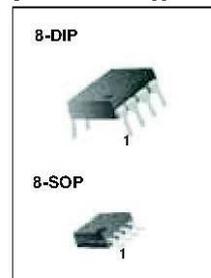
Single Operational Amplifier

Features

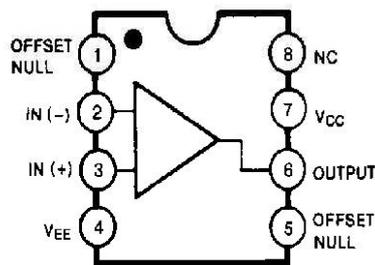
- Short circuit protection
- Excellent temperature stability
- Internal frequency compensation
- High Input voltage range
- Null of offset

Description

The LM741 series are general purpose operational amplifiers. It is intended for a wide range of analog applications. The high gain and wide range of operating voltage provide superior performance in integrator, summing amplifier, and general feedback applications.



Internal Block Diagram

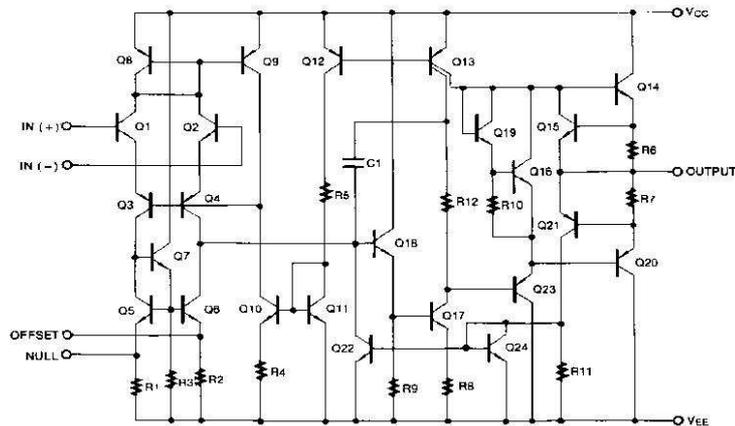


Rev. 1.0.1

©2001 Fairchild Semiconductor Corporation

LM741

Schematic Diagram



Absolute Maximum Ratings (TA = 25°C)

Parameter	Symbol	Value	Unit
Supply Voltage	V _{CC}	±18	V
Differential Input Voltage	V _{I(DIFF)}	30	V
Input Voltage	V _I	±15	V
Output Short Circuit Duration	-	Indefinite	-
Power Dissipation	P _D	500	mW
Operating Temperature Range LM741C LM741I	T _{OPR}	0 ~ +70 -40 ~ +85	°C
Storage Temperature Range	T _{STG}	-65 ~ +150	°C

Electrical Characteristics

($V_{CC} = 15V$, $V_{EE} = -15V$, $T_A = 25^\circ C$, unless otherwise specified)

Parameter	Symbol	Conditions	LM741C/LM741I			Unit	
			Min.	Typ.	Max.		
Input Offset Voltage	V_{IO}	$R_S \leq 10K\Omega$	-	2.0	6.0	mV	
		$R_S \leq 50\Omega$	-	-	-		
Input Offset Voltage Adjustment Range	$V_{IO(R)}$	$V_{CC} = \pm 20V$	-	± 15	-	mV	
Input Offset Current	I_{IO}	-	-	20	200	nA	
Input Bias Current	I_{BIAS}	-	-	80	500	nA	
Input Resistance (Note1)	R_I	$V_{CC} = \pm 20V$	0.3	2.0	-	M Ω	
Input Voltage Range	$V_{I(R)}$	-	± 12	± 13	-	V	
Large Signal Voltage Gain	G_V	$R_L \geq 2K\Omega$	$V_{CC} = \pm 20V$, $V_{O(P-P)} = \pm 15V$	-	-	-	V/mV
			$V_{CC} = \pm 15V$, $V_{O(P-P)} = \pm 10V$	20	200	-	
Output Short Circuit Current	I_{SC}	-	-	25	-	mA	
Output Voltage Swing	$V_{O(P-P)}$	$V_{CC} = \pm 20V$	$R_L \geq 10K\Omega$	-	-	-	V
			$R_L \geq 2K\Omega$	-	-	-	
		$V_{CC} = \pm 15V$	$R_L \geq 10K\Omega$	± 12	± 14	-	
			$R_L \geq 2K\Omega$	± 10	± 13	-	
Common Mode Rejection Ratio	CMRR	$R_S \leq 10K\Omega$, $V_{CM} = \pm 12V$	70	90	-	dB	
		$R_S \leq 50\Omega$, $V_{CM} = \pm 12V$	-	-	-		
Power Supply Rejection Ratio	PSRR	$V_{CC} = \pm 15V$ to $V_{CC} = \pm 15V$ $R_S \leq 50\Omega$	-	-	-	dB	
		$V_{CC} = \pm 15V$ to $V_{CC} = \pm 15V$ $R_S \leq 10K\Omega$	77	96	-		
Transient Response	Rise Time	T_R	Unity Gain	-	0.3	-	μs
	Overshoot	OS		-	10	-	%
Bandwidth		BW	-	-	-	MHz	
Slew Rate		SR	Unity Gain	-	0.5	-	V/ μs
Supply Current		I_{CC}	$R_L = \infty\Omega$	-	1.5	2.8	mA
Power Consumption		P_C	$V_{CC} = \pm 20V$	-	-	-	mW
			$V_{CC} = \pm 15V$	-	50	85	

Note:

1. Guaranteed by design.

dsPIC30F3014/4013

Special Microcontroller Features (Cont.):

- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
 - Sleep, Idle and Alternate Clock modes

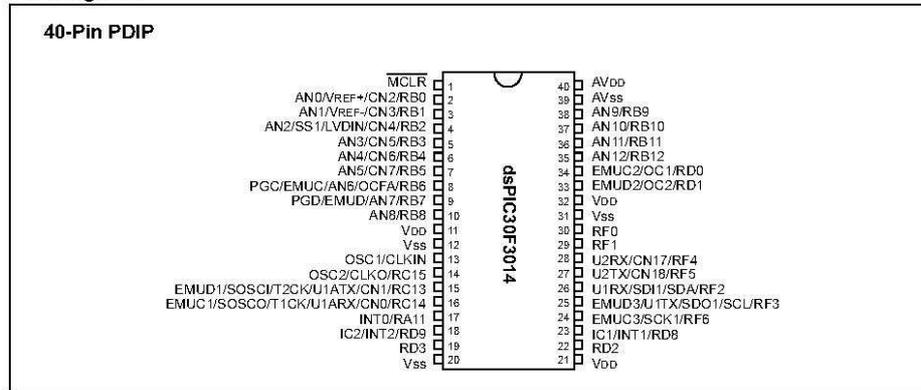
CMOS Technology:

- Low power, high speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption

dsPIC30F3014/4013 Controller Family

Device	Pins	Program Memory		SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Codec Interface	A/D 12-bit 100 Ksps	UART	SPI™	I ² C™	CAN
		Bytes	Instructions											
dsPIC30F3014	40/44	24K	8K	2048	1024	3	2	2	-	13 ch	2	1	1	0
dsPIC30F4013	40/44	48K	16K	2048	1024	5	4	4	AC'97, I ² S	13 ch	2	1	1	1

Pin Diagrams





Mejore su experiencia en ni.com. [Iniciar sesión](#) o [Crear un perfil de usuario](#).

NI USB-6009

DAQ Multifunción de Bajo Costo de 14 Bits, 48 kS/s

[Enviar esta Página](#) Configure esta Página para: [Imprimir](#) [PDF](#) [Texto](#)



[Hacer Zoom/Vistas Alternas](#)

8 entradas analógicas (14 bits, 48 kS/s)

2 salidas analógicas (12 bits a 150 S/s), 12 E/S digitales; contador de 32 bits

Energizado por bus para una mayor movilidad, conectividad de señal integrada

La versión OEM está disponible

Compatible con LabVIEW, LabWindows/CVI y Measurement Studio para Visual Studio .NET

El software de NI-DAQmx y software interactivo NI LabVIEW SignalExpress LE para registro de datos

[Comparar con productos similares](#)

[Ver Hoja de Especificaciones \(inglés\)](#)

Documentos con Especificaciones

[Especificaciones Detalladas \(2\)](#)

[Hoja de Datos \(inglés\)](#)

Resumen de Especificaciones

General

Producto	USB-6009
Familia de Productos	DAQ Multifunción
Formato Físico	USB
Sistema Operativo/Objetivo	Windows , Linux , Mac OS , Pocket PC
Familia de Productos DAQ	Serie B
Tipos de Medida	Voltaje
Compatibilidad con RoHS	Sí

Entrada Analógica

Canales	8 , 4
Canales de una sola terminal	8
Canales Diferenciales	4
Resolución	14 bits
Velocidad de Muestreo	48 kS/s

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

NI USB-6009 - DAQ Multifunción de Bajo Costo de 14 Bits, 48 kS/s - National Ins... Página 2 de 3

Rendimiento	48 kS/s
Máx. Voltaje de Entrada Analógica	10 V
Rango de Voltaje Máximo	-10 V , 10 V
Precisión Máxima del Rango de Voltaje	138 mV
Rango de Voltaje Mínimo	-1 V , 1 V
Mínima Precisión del Rango de Voltaje	37.5 mV
Número de Rangos	8
Muestreo Simultáneo	No
Memoria Interna	512 B
Salida Analógica	
Canales	2
Resolución	12 bits
Máx. Voltaje de Salida Analógica	5 V
Rango de Voltaje Máximo	0 V , 5 V
Precisión Máxima del Rango de Voltaje	7 mV
Rango de Voltaje Mínimo	0 V , 5 V
Mínima Precisión del Rango de Voltaje	7 mV
Razón de Actualización	150 S/s
Capacidad de Corriente Simple	5 mA
Capacidad de Corriente Total	10 mA
E/S Digital	
Canales Bidireccionales	12
Canales de Entrada Únicamente	0
Canales de Salida Únicamente	0
Número de Canales	12 , 0 , 0
Temporización	Software
Niveles Lógicos	TTL
Entrada de Flujo de Corriente	Sinking , Sourcing
Salida de Flujo de Corriente	Sinking , Sourcing
Filtros de Entrada Programables	No
¿Soporta Estados de Encendido Programables?	No
Capacidad de Corriente Simple	8.5 mA
Capacidad de Corriente Total	102 mA
Temporizador Watchdog	No
¿Soporta Protocolo de Sincronización para E/S?	No
¿Soporta E/S de Patrones?	No

Máximo Rango de Entrada	0 V , 5 V
Máximo Rango de Salida	0 V , 5 V
Contadores/Temporizadores	
Número de Contadores/Temporizadores	1
Operaciones a Búfer	No
Eliminación de Rebotes	No
Sincronización GPS	No
Rango Máximo	0 V , 5 V
Frecuencia Máxima de la Fuente	5 MHz
Entrada Mínima de Ancho de Pulso	100 ns
Generación de Pulso	No
Resolución	32 bits
Estabilidad de Tiempo	50 ppm
Niveles Lógicos	TTL
Especificaciones Físicas	
Longitud	8.51 cm
Ancho	8.18 cm
Altura	2.31 cm
Conector de E/S	Terminales de tornillo

[Volver arriba »](#)

[Mi Perfil](#) | [RSS](#) | [Privacidad](#) | [Aviso Legal](#) | [Comuníquese con NI](#)
© 2010 National Instruments Corporation. All rights reserved.
| [✉ Enviar esta página](#)



Mejore su experiencia en ni.com. [Iniciar sesión](#) o [Crear un perfil de usuario](#).

NI USB-6008

DAQ Multifunción de Bajo Costo de 12 Bits, 10 kS/s

[Enviar esta Página](#) Configure esta Página para: [Imprimir](#) [PDF](#) [Texto](#)



[Hacer Zoom/Vistas Alternas](#)

8 entradas analógicas (12 bits, 10 kS/s)

2 salidas analógicas (12 bits a 150 S/s), 12 E/S digitales; contador de 32 bits

Energizado por bus para una mayor movilidad, conectividad de señal integrada

La versión OEM está disponible

Compatible con LabVIEW, LabWindows/CVI y Measurement Studio para Visual Studio .NET

El software de NI-DAQmx y software interactivo NI LabVIEW SignalExpress LE para registro de datos

[Comparar con productos similares](#)

[Ver Hoja de Especificaciones \(inglés\)](#)

Documentos con Especificaciones

[Especificaciones Detalladas \(2\)](#)

[Hoja de Datos \(inglés\)](#)

Resumen de Especificaciones

General

Producto	USB-6008
Familia de Productos	DAQ Multifunción
Formato Físico	USB
Sistema Operativo/Objetivo	Linux , Mac OS , Pocket PC , Windows
Familia de Productos DAQ	Serie B
Tipos de Medida	Voltaje
Compatibilidad con RoHS	Sí

Entrada Analógica

Canales	8 , 4
Canales de una sola terminal	8
Canales Diferenciales	4
Resolución	12 bits
Velocidad de Muestreo	10 kS/s



L293B
L293E

PUSH-PULL FOUR CHANNEL DRIVERS

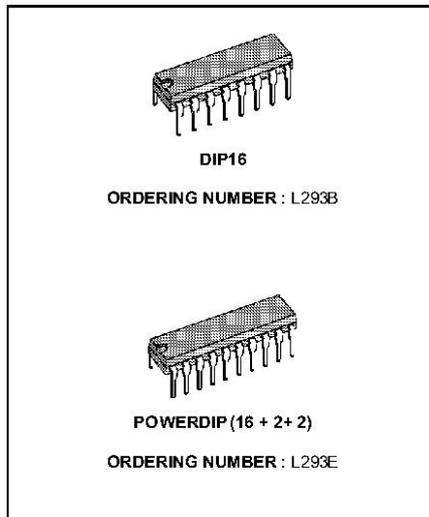
- OUTPUT CURRENT 1A PER CHANNEL
- PEAK OUTPUT CURRENT 2A PER CHANNEL (non repetitive)
- INHIBIT FACILITY
- HIGH NOISE IMMUNITY
- SEPARATE LOGIC SUPPLY
- OVERTEMPERATURE PROTECTION

DESCRIPTION

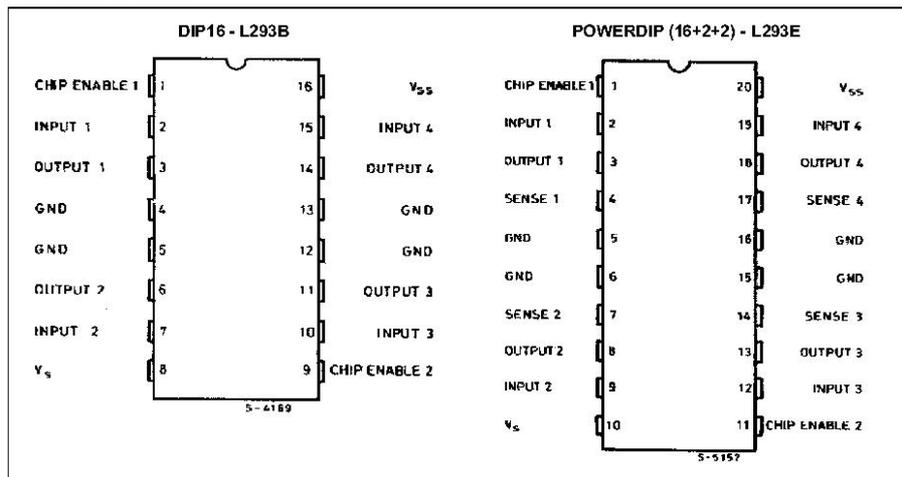
The L293B and L293E are quad push-pull drivers capable of delivering output currents to 1A per channel. Each channel is controlled by a TTL-compatible logic input and each pair of drivers (a full bridge) is equipped with an inhibit input which turns off all four transistors. A separate supply input is provided for the logic so that it may be run off a lower voltage to reduce dissipation.

Additionally, the L293E has external connection of sensing resistors, for switchmode control.

The L293B and L293E are package in 16 and 20-pin plastic DIPs respectively ; both use the four center pins to conduct heat to the printed circuit board.

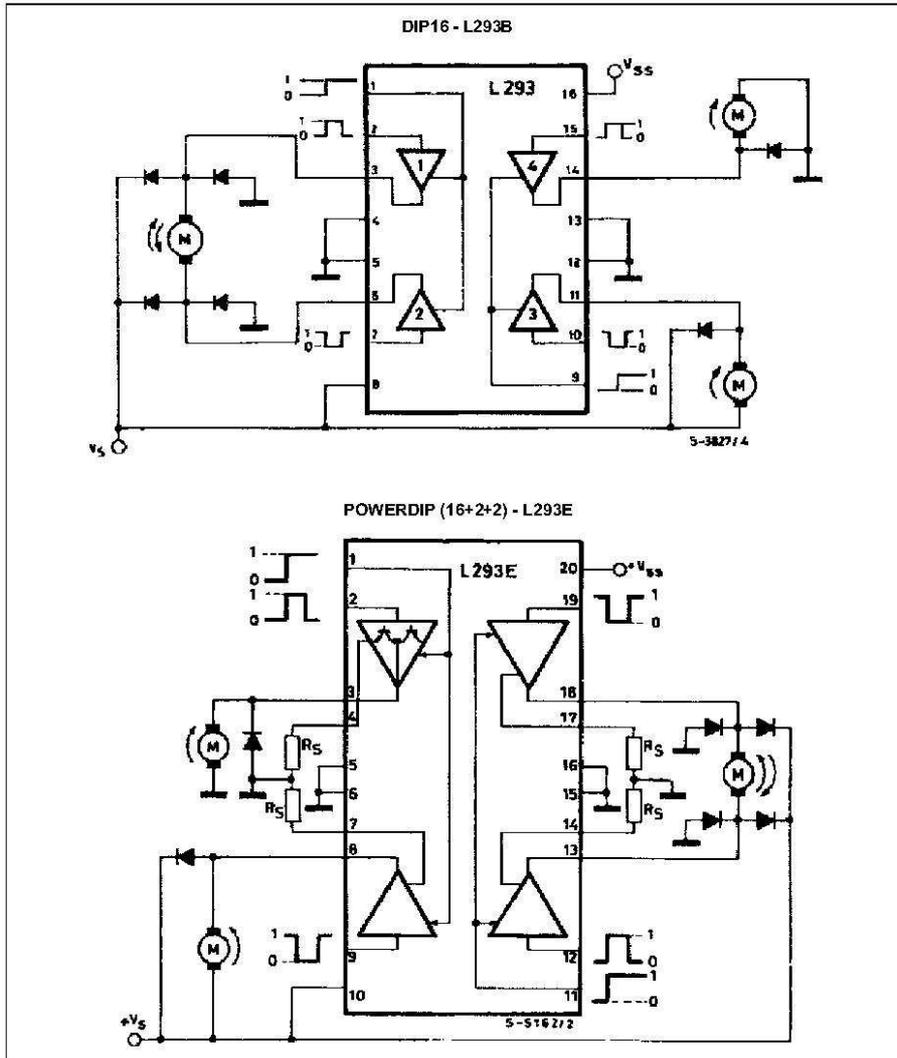


PIN CONNECTIONS



L293B - L293E

BLOCK DIAGRAMS



**SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN54S244
SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241, SN74S244
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS**

SDLS144 - APRIL 1986 - REVISED MARCH 1988

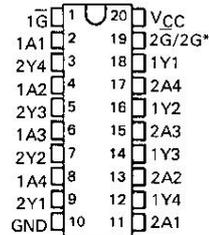
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- PNP Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

description

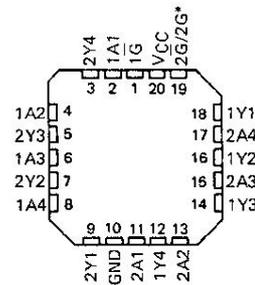
These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical \bar{G} (active-low output control) inputs, and complementary \bar{G} and G inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

The SN54' family is characterized for operation over the full military temperature range of -55°C to 125°C . The SN74' family is characterized for operation from 0°C to 70°C .

SN54LS', SN54S' . . . J OR W PACKAGE
SN74LS', SN74S' . . . DW OR N PACKAGE
(TOP VIEW)

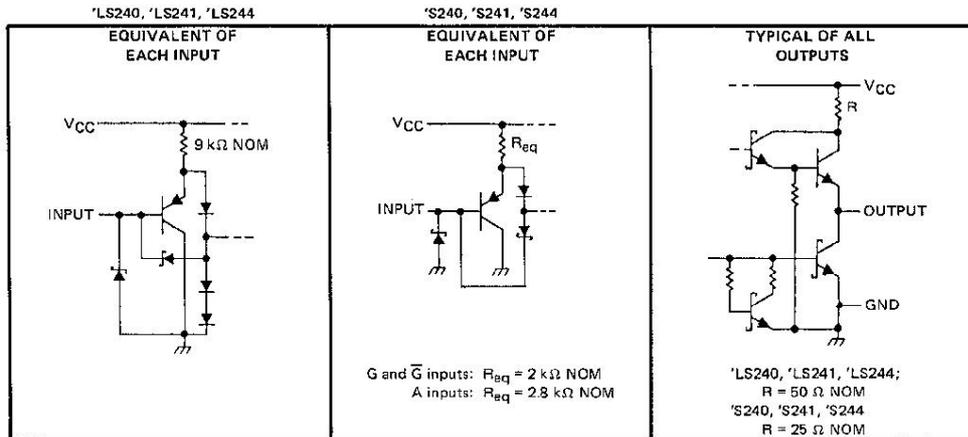


SN54LS', SN54S' . . . FK PACKAGE
(TOP VIEW)



*2G for 'LS241 and 'S241 or 2G for all other drivers.

schematics of inputs and outputs



PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1988, Texas Instruments Incorporated

Programadores



**Programador Pic
Debugger Usb**



**Convertidor de USB a
TTL (USB a serial para
Microcontroladores)***

ANEXO 4 Elaboración de interfaz de comunicación.

Esta tarjeta fue diseñada para poder interfazar la computadora con el robot ya que se necesita un voltaje de 24v para alimentar el controlador del robot y únicamente se obtienen 5V por lo tanto se utilizaron optoacopladores para que realizara esa función (Zacarías).

Los optoacopladores brindan protección contra altos voltajes mediante el aislamiento eléctrico que presentan, protegiendo a los dispositivos utilizados en el sistema de cualquier corriente de retorno. De esta manera se pueden acoplar los circuitos que se encuentran en la entrada y salida del optoacoplador, los cuales presentan diferentes voltajes de referencia.

El dispositivo que se va a encargar de enviar la información que proviene de la computadora a la tarjeta es una *Daq*, la cual proporcionara las señales necesarias para poder manipular el ROBOT de acuerdo a la aplicación.

Cuando en los puertos configurados previamente como salidas digitales de la *Daq* no se tenga ningún voltaje (0V, 0 lógico), la salida de la tarjeta tendrá 24V (1 lógico) véase fig. (6.9).

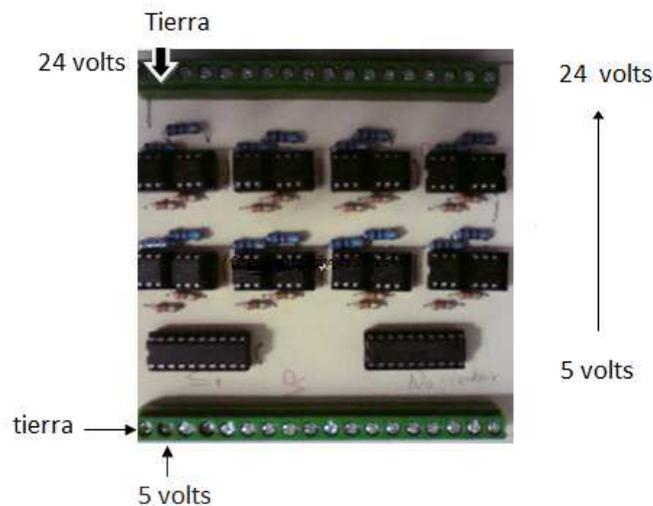


Fig.6.7. Interfaz de 5 a 24 Volts.

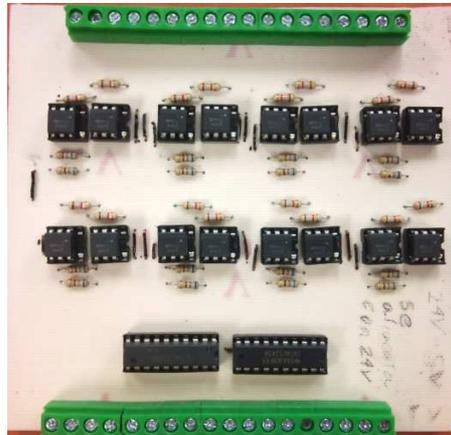


Fig. 6.8. Interfaz de 24 a 5 Volts.

La siguiente tarjeta tiene como función llevar la información que envía la *Daq* de la IRB 1600

A cualquier otro dispositivo por lo cual el robot le envía un voltaje de 24v a cada entrada (las que están más cerca de los optos que buffers) y esta la convierte a 5v para poder ser utilizada como señal digital. Véase fig. (6.10)

Las salidas se encuentran del lado de los buffers los buffers son 74ls245 que sirven de protección madamas para que no se queme la *Daq* de *Lab VIEW*, además de esto

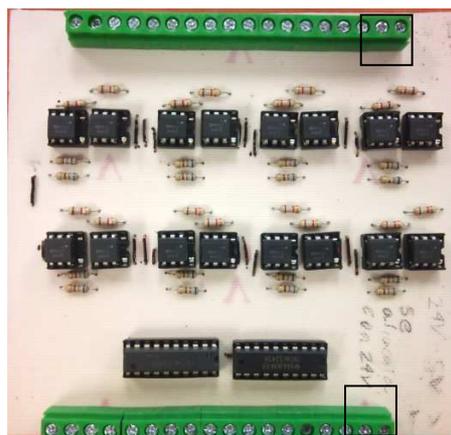


Fig. 6.9. Pin de Voltaje y Tierra.

El primer pin es tierra el segundo pin es voltaje a 5 volts las demás líneas se utilizan para sacar datos a 5v digitales.... Véase fig. (6.11).

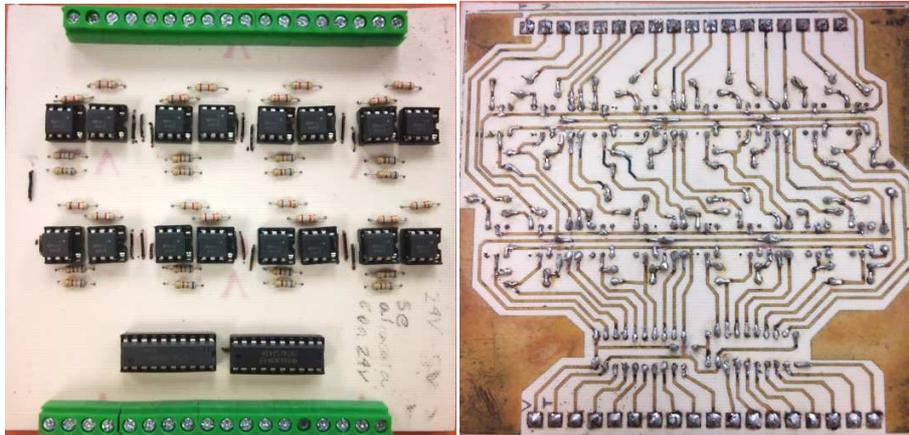


Fig. 6.10 Línea de voltaje a 24 Volts y Tierra.

El primer pin del cuadro es voltaje a 24v; el segundo es tierra además las tierras de los dos extremos se referencian. E los demás pines que tenemos alrededor del recuadro se alimentan con 24v ya cuando chequemos la salida es inversa y además está desfasado un pin de entrada con un pin de salida véase fig. (6.12).

ANEXO 4.

Miguel Angel Funes, José de Jesús Rubio, Raúl Rivera, Jaime Pacheco, Capítulo de libro en Trámite, titulado: Capitulo 2: "Fuzzy control with Labview for an articulated robot arm".

REFERENCIAS

- Fu, et al., (1989), "Robótica, Detección, Visión e Inteligencia", Editorial Mc Graw Hill, 1990, Pp. (306-462).
- Spong, et al., (1989), "Robot Modeling and Control", Editorial John Wiley & Sons Inc, 1989, Pp. (356-358).
- L. Natale and E. Torres-Jara,(2006), "A Sensitive approach to grasping", (2006, Septiembre)." *Proc. 6th Int. Conf. Epigenetic Robotics*, (2006).
- R. Platt, R. Burridge, M. Diftler, J. Graf, M. Goza, and E. Huber, (2006, Agosto). "Humanoid mobile manipulation using controller refinement," *Proc. RSS Workshop: Manipulation for Human Environments*, (2006).
- Kelly, et al.,(2003),"Control de movimiento de robots manipuladores", Editorial Pearson education,2003, Pp. (121-279).
- N. Sian, T. Sakaguchi, K. Yokoi, Y. Kawai, and K. Maruyama, (2006, Agosto). "Operating humanoid robots in human environments," *Proc. RSS Workshop: Manipulation for Human Environments*. (2006).
- S.M. LaValle, (2006). *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- A. Edsinger and C. Kemp, (2006). "Manipulation in human environments", *Proc. IEEE/RSJ Int. Conf. Humanoid, Robotics*, (2006).
- Ponce. Cruz, (2010) ,"*Inteligencia Artificial con aplicaciones en la Ingeniería*", Alfaomega grupo Editor, S.A. de C.V., México, Pp. (40-122.2010).
- K. Ogata, (2003), "Ingeniería de control Moderno", Pearson Education, 2003, Madrid, J. Ávila y G. Ochoa Análisis y Modelado de un Prototipo de un Aerogenerador. Pp.(2003).
- W. Spong, (2007), "Robot Dynamics and Control", University of Illinois of Urbana-Champaign, EUA, Pp. 169-175. PID Control Toolkit User Manual, National Instruments, USA (2007).
- M. Caruso. (2010), "Controlador tipo PID, sobre microcontrolador embebido en FPGA", Congreso de Microelectrónica Aplicada 2010.Pp (66).

ESIME AZCAPOTZALCO INSTITUTO POLITECNICO NACIONAL

A.Barrientos, (2007), "Fundamentos de Robótica", Mc Graw Hill, Interamericana de España, S.A. Pp. (640).

J. Craig, (2006), "Robótica" Editorial PEARSON Pp. (400).

A. Baturone (2001), "Robótica Manipuladores y Robots Móviles",2001, Pp. (447).

F. Torres, (2002), "Robots y Sistemas Sensoriales" Editorial Prentice Hill, (2002), Pp. (444).

J. Travis, (2009), "LabVIEW for every one Graphical Programming made easy and fun", Editorial PEARSON, Pp. (981).

G. Zill, (1997), "Ecuaciones Diferenciales con aplicaciones de Modelado", Editorial Thompson, Pp. (520).

W. Bolton, (2005), "Ingeniería de Control", Editorial Alfaomega Pp. (397).

P. Ponce, et al, (2010), "Intelligent Control Systems with Lab VIEW", Editorial Springer, Pp. (104).

W. Swokowski (1988),"Calculo con Geometría Analítica" Editorial Iberoamérica, Pp. (1095).

B.Buendía (1998), "La llave del Éxito en calculo Integral", Editorial Libudi, Pp. (238).

J. Forcada, (1996),"El amplificador Operacional", Editorial Alfaomega Pp. (273).

M. Johnson, (2005), "PID Control New Identification and Design Methods", Editorial Springer. Pp. (543).

Lung Tsai (1999), "Robot Analysis The Mechanics of serial and parallel Manipulators", Editorial Wiley-Interscience Publication (1999), Pp. (503).

Cosgalla (2000), "Análisis de Vigas", Editorial Spanta, Pp. (232).

Manuales Industriales

Microchip (2004) AN964 “Software of an Inverted Pendulum Using PIC16f684”
Pp. (14).

Microchip (2004). “Implementing a PID Controller using a pic18 MCU”, Microchip
Technologies Pp.16.

P. Corke (2008), “Robotics Toolbox for Matlab”, Publicación, MATLAB, Pp. 76.

LabVIEW (2007), “PID Control Toolkit User Manual”, Editorial National Instruments.

Víctor M. Santos (2003), “Robótica Industrial”, Universidad de Aveiro, Pp. (166).

W. Sanz (2004), “Diseño de software para la modelación de Manipuladores con Matlab y
simulación 3d con roboworks”, Ingeniería UC, Carabobo, Pp. (13).

Rick Bitter et al, (2001),”Lab VIEW Advanced Programming Techniques”, Editorial
CRC.

Tesis Referenciadas.

M. Zacarías (2009), “Implementación de un Algoritmo genético para la evasión de
obstáculos en manipuladores robóticos articulados”, Pp. (68).