



**Instituto Politécnico Nacional**

---

**CENTRO DE INVESTIGACIÓN EN CIENCIA  
APLICADA Y TECNOLOGÍA AVANZADA**

**UNIDAD QUERÉTARO**

**POSGRADO EN TECNOLOGÍA AVANZADA**

**Compensación de Zona Muerta y Variación de Carga  
en el Control de Velocidad de Motores de  
Corriente Directa**

**TESIS QUE PARA OBTENER EL GRADO DE  
MAESTRIA EN CIENCIAS EN  
TECNOLOGÍA AVANZADA**

**PRESENTA**

**ING. JESÚS PICHARDO CENTENO**



**Directores de Tesis**

**Dr. Jorge Adalberto Huerta Ruelas**

**Dr. Eduardo Castillo Castañeda**

**Querétaro, Qro. Agosto de 2008**



INSTITUTO POLITECNICO NACIONAL
SECRETARIA DE INVESTIGACION Y POSGRADO

ACTA DE REVISION DE TESIS

En la Ciudad de Santiago de Querétaro, Qro. siendo las 12:00 horas del día 13 del mes de Agosto del 2008 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CICATA- QUERETARO para examinar la tesis de grado titulada: 'COMPENSACIÓN DE ZONA MUERTA Y VARIACIÓN DE CARGA EN EL CONTROL DE VELOCIDAD DE MOTORES DE CORRIENTE DIRECTA'

Presentada por el alumno:

PICHARDO
Apellido paterno

CENTENO
materno

JESÚS
nombre(s)

Con registro: A 0 6 0 0 9 8

aspirante al grado de:

MAESTRÍA EN TECNOLOGÍA AVANZADA

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

LA COMISION REVISORA

Director de tesis

Signature of Dr. Jorge Adalberto Huerta Ruelas
Dr. Jorge Adalberto Huerta Ruelas

Director de tesis

Signature of Dr. Eduardo Castillo Castañeda
Dr. Eduardo Castillo Castañeda

Signature of M.C. Maximiano Francisco Ruiz Torres
M.C. Maximiano Francisco Ruiz Torres

Signature of Dr. Iván Domínguez López
Dr. Iván Domínguez López

EL PRESIDENTE DEL COLEGIO

Signature of Dr. Eduardo Morales Sánchez
Dr. Eduardo Morales Sánchez

Signature of Dr. Joaquín Salas
Seal of the Secretariat of Education, Public and Social Welfare
INSTITUTO POLITECNICO NACIONAL
SECRETARIA DE INVESTIGACION Y POSGRADO
CIENCIA APLICADA
Y TECNOLOGIA AVANZADA
UNIDAD QUERETARO
DIRECCION



**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

*CARTA CESION DE DERECHOS*

En la Ciudad de Santiago de Querétaro, Qro., el día 13 del mes de agosto del año 2008, el que suscribe Ing. Jesús Pichardo Centeno, alumno del Programa de Maestría en Tecnología Avanzada con número de registro A060098, adscrito a CICATA Querétaro, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Jorge Adalberto Huerta Ruelas y el Dr. Eduardo Castillo Castañeda y cede los derechos del trabajo intitulado “Compensación de zona muerta y variación de carga en el control de velocidad de motores de corriente directa”, 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 [jesus\\_pice@yahoo.com.mx](mailto:jesus_pice@yahoo.com.mx) y/o [jhuertar@ipn.mx](mailto:jhuertar@ipn.mx). Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

  
\_\_\_\_\_  
Jesús Pichardo Centeno

*A Dios,  
por haberme creado y regalarme esta vida llena de placeres.*

*A mis padres José Guadalupe y Leticia,  
por su apoyo incondicional, sus desvelos y las interminables jornadas de duro trabajo,  
sintiendo profunda admiración y respeto, con todo mi amor y gratitud...*

*A mis directores y asesores,  
por su tolerancia, enseñanzas y apoyo.*

*A mis compañeros,  
por los momentos maravillosos que compartimos, en especial a JCOR por la amistad que hemos  
construido.*

*Al CONACyT,  
por el apoyo económico recibido durante el estudio de este posgrado.*

*Al IPN,  
por la infraestructura y apoyo económico, pero sobretodo por haberme formado en la convicción  
de poner mis conocimientos al servicio de la nación.*

**COMPENSACIÓN DE ZONA MUERTA Y VARIACIÓN DE CARGA EN EL  
CONTROL DE VELOCIDAD DE MOTORES DE CORRIENTE DIRECTA**

**Jesús Pichardo Centeno**

Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada

Laboratorio de Metrología Óptica

Instituto Politécnico Nacional

Querétaro, Qro., México

## ÍNDICE

<b>CAPÍTULO 1</b>	<b>INTRODUCCIÓN</b>	<b>12</b>
1.1	Antecedentes	12
1.1.1	Tribometro de perno en disco	13
1.1.3	Flexión de eslabón en banco experimental	16
1.2	Justificación	18
1.3	Objetivo general y objetivos específicos	19
<b>CAPÍTULO 2</b>	<b>MARCO TEÓRICO</b>	<b>20</b>
2.1	Modelo Matemático	20
2.2	Identificación del Sistema	25
2.3	Diseño del Controlador	32
2.3.1	Control PI	32
2.3.2	Sintonización	35
2.4	Zona Muerta	39
2.4.1	Efectos	39
2.4.2	Compensación	40
2.5	Variación de Carga	44
2.5.1	Metodología de Compensación	44
<b>CAPÍTULO 3</b>	<b>MATERIALES Y MÉTODOS</b>	<b>48</b>
3.1	Descripción General	48
3.2	Desarrollo de software	50
<b>CAPÍTULO 4</b>	<b>RESULTADOS EXPERIMENTALES</b>	<b>54</b>
4.1	Identificación del Sistema	54
4.2	Desempeño del Controlador	58

4.3	Zona Muerta .....	69
4.4	Variación de Carga.....	71
4.5	Anti-Wind-Up .....	77
<b>CAPÍTULO 5 CONCLUSIÓN Y PERSPECTIVAS DEL TRABAJO .....</b>		<b>81</b>
5.1	Conclusión.....	81
5.2	Perspectivas de Trabajo.....	82
<b>REFERENCIAS .....</b>		<b>84</b>
<b>PUBLICACIONES.....</b>		<b>88</b>
<b>APÉNDICE 1 MANEJO DEL COMPILADOR Y ENTORNO GNU C++.....</b>		<b>89</b>
<b>APÉNDICE 2 CODIGO DE CLASES.....</b>		<b>94</b>

## Resumen

La implantación de sistemas automáticos para la manufactura es actualmente, por las ventajas que estos ofrecen en cuanto a repetibilidad, calidad y eficiencia, una de las tareas con más demanda industrial. Entre estos, los servosistemas impulsados por máquinas eléctricas brindan para un gran número de aplicaciones, el más avanzado y preciso control de movimiento disponible en el mercado. El presente trabajo forma parte del esfuerzo que se realiza en CICATA-IPN por sentar las bases para el desarrollo de tecnología propia en materia de control de movimiento basado en motores de corriente directa. El sistema experimental desarrollado en esta tesis para el control de movimiento de motores de corriente directa, fue motivado por necesidades del centro de investigación para el control de velocidad, la compensación de fenómenos de zona muerta y desviaciones por variaciones de carga en motores. El motor principal utilizado para el desarrollo del sistema, fue de imán permanente con escobillas, utilizando un encoder incremental en cuadratura como elemento de retroalimentación. El algoritmo base es un controlador PI sintonizado para un desempeño óptimo.

Experimentalmente se encontró que la zona muerta no es simétrica, depende de la carga referida al eje del motor, de su aceleración y la frecuencia del PWM utilizada para modular la señal de control. Inicialmente, mediante la función signo, se sumó a la señal de control el voltaje requerido para evitar la zona muerta y al volver a sintonizar el controlador de acuerdo a la señal de referencia deseada, se logró una compensación ante perturbaciones de par constantes. Aunque este método logró minimizar su efecto en el seguimiento de una señal de referencia sinusoidal, no de manera completa. Sintonizando el controlador PI mediante un “set-point-gain-scheduling” basado en el modelo electromecánico, se logró responder de forma más rápida a variaciones de carga aunque el sobre-impulso alcanzó en algunos experimentos valores de hasta el 50%. Para eliminar el exceso en el sobre-impulso y lograr una respuesta acorde con lo simulado, se agregó al sistema de control una estructura “anti-wind-up” para evitar los efectos causados por la saturación en el servo-amplificador, logrando sobre-impulsos cercanos al 0%.

El sistema desarrollado permite, la identificación fuera de línea de los parámetros de la función de transferencia para controlar la velocidad de motores, corregir efectos de zona muerta y, eliminar el sobre impulso que se genera cuando un sistema de control trata de compensar variaciones de carga

## Abstract

At present, motion control of devices used for the automatic product manufacturing is one of the most demanded tasks in industry, due to advantages in repeatability, efficiency and quality that are obtained after the implementation of control systems. Servo-systems driven by electrical machines offer great number of applications, and most advanced and precise motion control available industrially. The present work is part of the effort that is under progress in CICATA-IPN Querétaro research center, to expand fundamentals for the development of own technology in motion control of direct current motors. Experimental setup developed in this thesis, for motion control with direct current motors, was motivated for specific needs in research center related to speed control, dead zone compensation and load disturbances in motors. Main motor utilized in this work, was a brushed permanent magnet direct current motor with an incremental quadrature encoder as feedback element. Base algorithm is a PI controller tuned for an optimal performance.

Experimentally, it was found that dead zone is not symmetrical, depending on load referred to motor axis, its acceleration and PWM frequency used to modulate control signal. By means of sign function, required voltage to overcome static friction is added to the control signal in order to avoid dead zone and when tuning the controller according to desired reference signal, a better compensation is obtained when the system is subject to step torque disturbances. Therefore, although the used method to compensate the dead zone diminishes its effect in the follow-up of a sinusoidal reference signal, was not eliminated completely. Tuning a PI controller by means of a Set-point-gain-scheduling approach based on the electromechanical model, it was managed to respond faster to load variations although over shoot reached in some experiments went to values up to 50%. To eliminate the excess in over shoot and to obtain a behavior accorded with simulation data, an “anti-wind-up” structure was added in control system loop to avoid effects caused by servo-amplifier saturation, obtaining over shots close to 0%.

Developed system allows, off-line identification of transfer function parameters to control motor speed, correct effects of dead zone and, eliminate over shoot that is generated when a control system attempts to compensate load variations.

## Índice de figuras

Figura 1-1. Esquema del tribómetro de perno en disco.....	14
Figura 1-2. Correlación de las señales de fuerza de fricción y de control de velocidad en una prueba de desgaste.....	15
Figura 1-3.- Prototipos de manipuladores paralelos desarrolladas en CICATA.....	16
Figura 1-4. Señal RD Normalizada vs. Señal Extensométrica.....	17
Figura 1-5. Flexor donde es necesario acoplar el motor.....	18
Figura 2-1.- Modelo electromecánico de un motor de corriente directa.....	21
Figura 2-2.- Modelo en diagrama de bloques de los motores de CD.....	24
Figura 2-3.- Proceso de identificación para un sistema de primer orden.....	26
Figura 2-5.- Proceso de identificación, a) diagrama de bloques, b) respuesta del sistema desconocido.....	29
Figura 2-6.- Proceso de identificación, determinación gráfica de $T_m$ .....	31
Figura 2-7.- Diagrama de bloques del sistema de control.....	33
Figura 2-8.- Diagrama de bloques del sistema de control sintonizado.....	37
Figura 2-9.- Simulación de respuestas para señales de referencia de tipo escalón.....	38
Figura 2-14.-Compensación de variación de carga.....	47
Figura 3-1.- Elementos de los sistemas de control.....	48
Figura 3-2.- Arreglo experimental para el control de velocidad de motores de CD.....	49
Figura 4-1.- Efecto del precalentamiento.....	54
Figura 4-4.- Determinación gráfica experimental de $T_m$ .....	57
Figura 4-5.- Modelo vs. Respuesta real.....	58
Figura 4-6.- Diagrama de bloques del sistema de control.....	59
Figura 4-7.- Respuestas óptimas normalizadas, con y sin.....	61
Figura 4-8.- Polos en el plano Z de la FT de la ecuación 12 para diferentes valores de $\Delta$ . a).....	63
Figura 4-9.- Desempeño del Controlador.....	65
Figura 4-10.- Comparación entre desempeños simulado y real.....	65
Figura 4-11.- Diagrama de bloques incluyendo fenómenos no modelados.....	66
Figura 4-12.- Zona muerta experimental y función velocidad vs. voltaje.....	67
Figura 4-13.- Desempeño real y simulación considerando fenómenos no modelados.....	68
Figura 4-14.- Sistema con disco de carga.....	69
Figura 4-15.- Desempeño del sistema.....	70
Figura 4-16.- Efecto de la zona muerta.....	70
Figura 4-17.- Compensación de la zona muerta.....	71
Figura 4-18.- Sistema utilizado para inducir variaciones de carga constantes.....	72

<b>Figura 4-20.- Compensación de variación de carga. (a) y (c) Velocidad controlada para los experimentos 1 y 2, (b) y (d) Señal de control de los experimentos 1 y 2.....</b>	<b>74</b>
<b>Figura 4-21.- Compensación de variación de carga. a) y c) Velocidad controlada para los experimentos 3 y 4, b) y d) Señal de control de los experimentos 3 y 4 .....</b>	<b>76</b>
<b>Figura 4-23.- Compensación de zona muerta por set point gain scheduling .....</b>	<b>77</b>
<b>Figura 4-24.- Sistema de control PI con anti-wind-up.....</b>	<b>78</b>
<b>Figura 4-25.- Desempeño de control, (a) Set-point-gain-scheduling sin anti-wind-up, (b)Set-point-gain-scheduling con anti-wind-up .....</b>	<b>78</b>
<b>Figura 4-26.- Desempeño para una referencia de 120 rad/seg, (a) y (b) Respuesta y señal de control sin wind-up, (c) y (d) con anti-wind-up .....</b>	<b>79</b>

# Capítulo 1 Introducción

El control de movimiento ocupa un papel clave en el logro de operaciones de manufactura de bajo costo, rápida y flexible. Soluciones óptimas a estos requerimientos necesitan de una visión integrada de la mecánica, la electrónica y la programación para el control de movimiento, siendo ésta una solución de carácter mecatrónico. Actualmente, el problema de control de movimiento se puede resolver mediante la adquisición de información proveniente de un servo-actuador, la tarjeta de control, el servo-amplificador, la fuente de alimentación y el software para la programación del dispositivo, regularmente todo esto se consigue como un paquete que ofrecen distintos fabricantes.

El problema principal en los sistemas de control es que estos se encuentran sujetos a cambios súbitos que no han sido modelados. Mediante un modelo robusto de los algoritmos de control y mediante el uso de controladores adaptables, se trata de evitar que estos cambios tengan un efecto negativo en la dinámica del sistema [Chang 2003]. Particularmente, en el caso de motores de corriente directa como principales actuadores en servomecanismos, éstos se encuentran sujetos a fenómenos no modelados de variación de carga, saturación, juego mecánico, fricción, gravedad entre otros [Cheng 2007].

La primera sección de este capítulo presenta los antecedentes y la justificación que dieron origen a este trabajo. Enseguida se define el objetivo general y objetivos específicos.

## 1.1 Antecedentes

Para poder controlar la velocidad en cualquier sistema de control de movimiento, es necesario contar con un servo-actuador que permita transmitir el movimiento a los elementos mecánicos del sistema. El servo-actuador se compone de tres elementos; un servomotor, un elemento de transmisión el cuál multiplica el par del motor, y un sensor de posición. Además del servo-actuador, se necesita una computadora o bien, un microcontrolador en donde se reciben las señales y se programan algoritmos para la lectura del sensor de posición, los interruptores de límite de fin de carrera, la estrategia de control y a su vez permiten la interacción con el usuario, entregando como respuesta a estas entradas y estados predefinidos una señal eléctrica de control que es entrada a su vez de un servo-amplificador, el cual sirve como etapa de potencia y

suministra la energía controlada necesaria para manipular al servo-actuador. Al conjunto de todos los elementos mencionados se le conoce como *servomecanismo*.

El desarrollo de este trabajo fue motivado por tres diferentes razones. La motivación inicial para desarrollar sistemas de control de movimiento, surgió de la necesidad de controlar la velocidad en un tribómetro de perno en disco fabricado en colaboración con el Centro de Investigación e Innovación tecnológica del Instituto Politécnico Nacional (CIITEC-IPN) que se encuentra en el área de tribología en CICATA-IPN Unidad Querétaro. La segunda necesidad es compensar los efectos de la fricción estática en mecanismos paralelos y seriales. Finalmente, la necesidad de flexionar transversalmente eslabones en un banco experimental de esfuerzos, en la que la deformación debe ser aplicada con la menor incertidumbre en la posición y que pueda programarse diferentes rampas con diferentes perfiles de velocidad. En las siguientes subsecciones se describe a mayor detalle cada una de estas necesidades.

### **1.1.1 Tribómetro de perno en disco**

La medición de parámetros tribológicos como la razón de desgaste o la fuerza de fricción, se efectúa de manera rutinaria utilizando instrumentos conocidos como tribómetros. De estos, existe una gran variedad para simular la operación de los sistemas tribológicos en condiciones de desgaste. En particular, el tribómetro de perno en disco ha encontrado aplicaciones en estudios de sistemas tribológicos en condiciones de desgaste adhesivo. El ensayo de perno en disco consiste en hacer pasar un perno sobre la superficie de un disco girando con velocidad angular  $\omega$  constante. El perno lleva una carga definida, aplicada perpendicularmente a la superficie del disco. Como resultado del contacto, el perno deja sobre el disco una huella circular originada por el desgaste de la superficie. De acuerdo con la norma ASTM G99-05, los parámetros tribológicos que se miden en este ensayo son: la fuerza de fricción y el desgaste. Este último, se mide de varias formas, una vez que la prueba ha concluido [Pichardo 2007].

En la Figura 2-2 se presenta un esquema del aparato en el que se puede apreciar: en la parte inferior, el motor de corriente directa de 1/8 HP de potencia, conectado a un reductor con una relación de 11:1, acoplado, mediante una banda y poleas dentadas al eje de giro de la mordaza que sujeta el disco. El codificador óptico se utiliza para registrar la posición y velocidad angular del disco, la cual puede ajustarse dentro de un intervalo de 150 a 600 rpm; el par tribológico

consta del perno, situado en el extremo del brazo oscilatorio, y el disco, acoplado al motor por medio de la mordaza. El contrapeso se utiliza para balancear de forma horizontal el brazo, antes de colocar la carga sobre el perno. La fuerza de fricción se mide a partir de la flexión lateral del brazo y se registra por medio de un transformador diferencial variable lineal (LVDT).

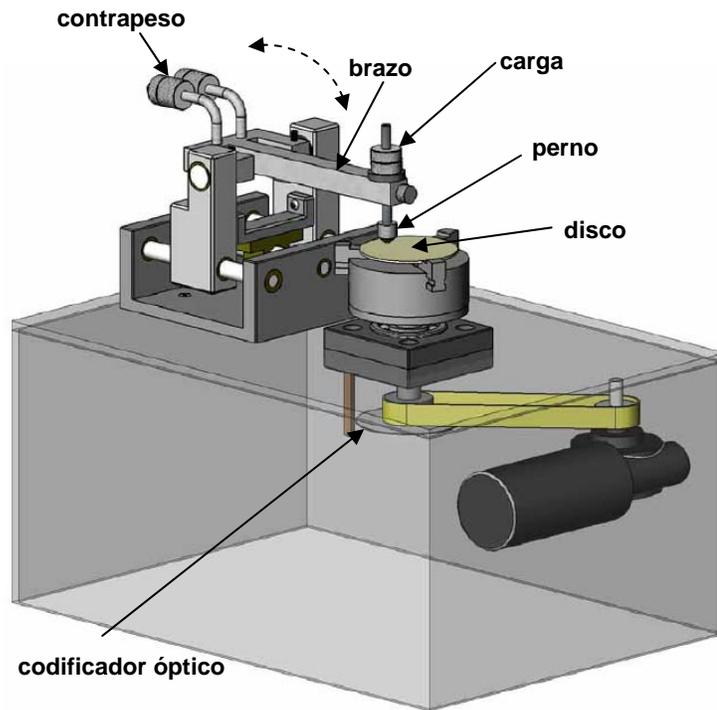


Figura 1-1. Esquema del tribómetro de perno en disco

### *Correlación de la Señal de Control y la Fuerza de Fricción*

Los sistemas de control de velocidad y de fuerza de fricción se utilizaron en pruebas de desgaste en las que inicialmente las superficies en contacto tienen un acabado superficial muy fino, por lo cual la fuerza de fricción es menor en esta etapa. Dentro de los primeros instantes del proceso, la rugosidad de las superficies cambia de manera abrupta, ocasionando un incremento en la fuerza de fricción. En la Figura 3.1 se presentan las mediciones de fuerza de fricción y las variaciones en la señal de control de velocidad durante una prueba de desgaste. La dependencia temporal en ambas gráficas es muy similar, lo que indica que los cambios en la fuerza de fricción están correlacionados con los cambios en la señal de control.

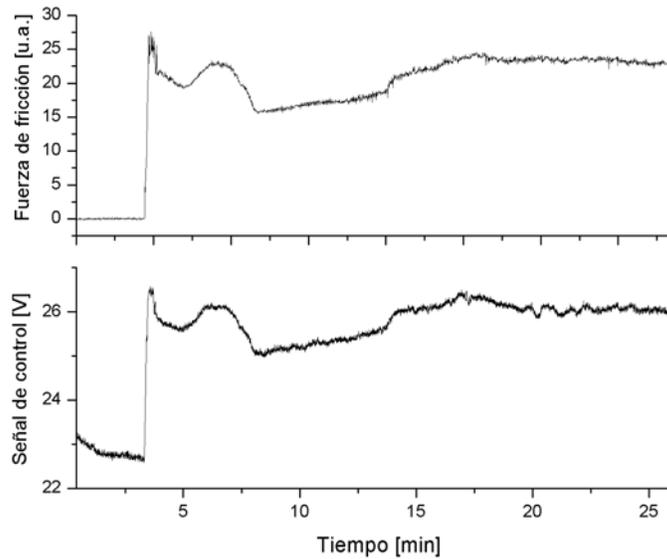


Figura 1-2. Correlación de las señales de fuerza de fricción y de control de velocidad en una prueba de desgaste.

### 1.1.2 Compensación de fricción estática en manipuladores paralelos

Los manipuladores paralelos se han convertido en un tema de investigación bastante interesante, esto debido a muchas de las ventajas que ofrecen en comparación con manipuladores seriales, tales como la facilidad de construcción, simplicidad para encontrar la cinemática inversa y su alta capacidad de aceleración. En el campo de la maquinaria para la manufactura, los robots paralelos empiezan a reemplazar a las máquinas herramientas convencionales, sin embargo algunos problemas no han sido resueltos completamente. Uno de estos problemas se debe a que los robots paralelos no son aún tan precisos como las máquinas herramientas convencionales. Esta precisión depende tanto del diseño mecánico como de los algoritmos de control de movimiento que minimicen el error en el seguimiento de trayectoria de cada junta activa [Castillo 2007].

En la sección de mecatrónica del CICATA-IPN Unidad Querétaro se han estado realizando trabajos de investigación referentes al diseño, simulación y construcción de mecanismos paralelos y los fenómenos que afectan su precisión. Tales investigaciones se presentan en [Nava 2007] y [Yañez 2007] los cuales estudian el efecto de las deformaciones en los eslabones del manipulador de Maryland y el espacio de trabajo en manipuladores planares respectivamente.

Ya que los algoritmos de control influyen en la precisión de los robots paralelos, también se han propuesto técnicas para mejorarla, como en [Castillo 2006] donde se propone la compensación de la fricción estática mediante la identificación del voltaje de rompimiento relacionado con el par necesario para empezar el movimiento en un mecanismo paralelo de seis grados de libertad. De este último, la fricción estática puede ser relacionada con la región de zona muerta en la respuesta voltaje-velocidad en motores de corriente directa, lo que se intenta compensar en este trabajo. En la figura 1-3 se muestran algunos prototipos desarrollados en estos laboratorios.



Figura 1-3.- Prototipos de manipuladores paralelos desarrolladas en CICATA

### 1.1.3 Flexión de eslabón en banco experimental

El modelado y el control de manipuladores robóticos con elementos flexibles resulta de especial interés, ya que los robots rígidos convencionales se construyen con materiales pesados y caros con objeto de que presenten una alta rigidez estructural que haga que su control sea relativamente sencillo. Sin embargo dichos manipuladores presentan varios inconvenientes, como los altos consumos de potencia, baja velocidad en el movimiento, necesidad de actuadores sobredimensionados, baja relación masa manipulable frente a la masa total del manipulador y sus elevados costos. Todos estos inconvenientes se pueden solventar con la utilización de manipuladores robóticos con elementos ligeros y flexibles, lo que conlleva el estudio de las flexibilidades estructurales del robot a la hora de diseñar su control, especialmente en aplicaciones de alta velocidad. El control de manipuladores flexibles debe resolver numerosas dificultades que no están presentes en el caso de robots rígidos, como por ejemplo el hecho de

que en un manipulador flexible el número de variables de control es menor que el número de grados de libertad.

En el laboratorio de metrología óptica donde se realizó esta tesis, está en ejecución un proyecto relativo al estudio del estado esfuerzo-deformación de estos elementos eslabón utilizando un conjunto de técnicas de diferente naturaleza. Una de estas técnicas es ya de amplio uso por su nivel de desarrollo y madurez como lo es la Extensometría. Otra de ellas está en pleno desarrollo y se están extendiendo ampliamente paralelamente al desarrollo de la computación y está basada en la solución numérica por medio del método del elemento finito MEF, mismo que permite realizar simulación de fenómenos muy complejos. Otra técnica que se aplica, de naturaleza óptica es la técnica de Reflectancia Diferencial (RD) o Reflectancia Anisotrópica (RA). Hasta la fecha se ha logrado obtener una correlación lineal entre las mediciones de extensometría y de RD. Ver la Figura 1-4.

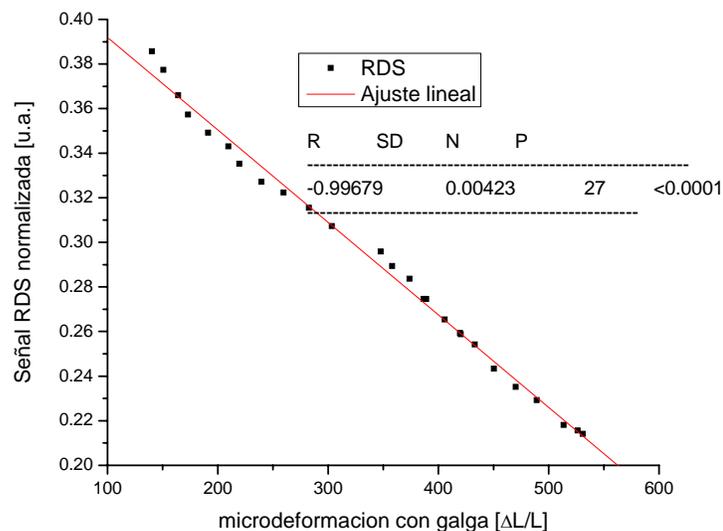


Figura 1-4. Señal RD Normalizada vs. Señal Extensométrica.

Para el desarrollo del proyecto relativo al estado de esfuerzo / deformación en eslabones, se diseñó y está en construcción un banco experimental de esfuerzos y caracterización de materiales, en el cual se requiere un mecanismo adicional que aplique flexión a eslabones, con control preciso de la posición y con la posibilidad de aplicar ciclos de flexión a diferentes velocidades. En la figura 1-5 se muestra el flexor donde se requiere acoplar el motor en el tornillo micrométrico.

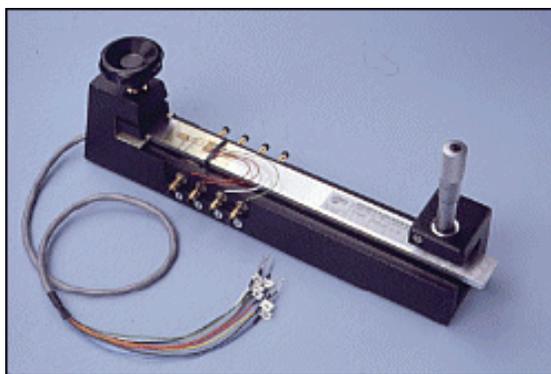


Figura 1-5. Flexor donde es necesario acoplar el motor

## 1.2 Justificación

Actualmente el problema de control de movimiento se resuelve mediante la adquisición del servo-actuador, la tarjeta de control, el servo-amplificador, la fuente de alimentación y el software para la programación del dispositivo. Regularmente todo esto se consigue como un paquete que ofrecen distintos fabricantes. Las instituciones educativas del país realizan investigación al respecto utilizando esta tecnología, el sector industrial realiza sus actividades de manufactura del mismo modo. Sin embargo, aunque en México se hace investigación relevante en control de motores, la mayor parte de la tecnología para el control de movimiento que se vende en nuestro país proviene de empresas extranjeras. Si bien, hacer uso de esta tecnología incrementa nuestra competitividad a nivel mundial, se incrementaría aún más si además de utilizarla, esta se desarrollará en México utilizando las investigaciones realizadas por investigadores nacionales en este sector. Este trabajo es parte de un intento por llenar ese espacio entre la teoría y la práctica, al estudiar investigaciones recientes tanto nacionales como internacionales, así como los productos que se encuentran en este momento en el mercado. Con los conocimientos adquiridos desarrollar tecnología propia y realizar las pruebas pertinentes para su evaluación al darle aplicación en los diferentes sistemas que requieren un control de velocidad en los laboratorios de esta institución.

Comercialmente existen básicamente dos tipos de tarjetas para el control de movimiento en motores de CD: las de arquitectura abierta y las de arquitectura cerrada. En el primer tipo se encuentran PMAC y DSPACE, las cuales permiten implementar algoritmos complejos de control diferentes al convencional PID e incluso ser descargados directamente al DSP desde el diagrama de bloques en SIMULINK. La desventaja de estas es su costo el cual oscila entre los 2,500 y

10,000 USD. Las del segundo tipo son mucho más baratas, su precio está entre los 125 y 650 USD para PICSERVO y PCI174XXX de National Instruments respectivamente. La desventaja de estas últimas es que solo permiten programar algoritmos de control basados en PID. Con el sistema que se presentará en este trabajo es posible programar algoritmos de control de cualquier tipo al introducir la ecuación en diferencias de la ley de control, lo que coloca a este desarrollo entre ambas arquitecturas.

### **1.3 Objetivo general y objetivos específicos**

El objetivo principal de este proyecto es el desarrollo de software-hardware para la implementación de algoritmos de control de velocidad en motores de corriente directa utilizados como generadores de movimiento en aplicaciones que se encuentran en desarrollo en los laboratorios de Metrología Óptica, Mecatrónica y Tribología.

Para cumplir con este objetivo principal se han planteado los siguientes objetivos específicos:

- Integrar los elementos del sistema de control, instrumentarlos y desarrollar el software para el manejo, envío y procesamiento de señales.
  - Identificar experimentalmente los parámetros principales en motores de corriente directa de imán permanente con escobillas y proponer la estrategia de control.
  - Implementar el algoritmo de control propuesto y evaluar su desempeño.
  - Identificar el efecto de la zona muerta en el desempeño del sistema y proponer la estrategia para compensarla.
  - Proponer una metodología para una rápida compensación de efectos debidos a variaciones en la carga referida al eje del motor.
-

## Capítulo 2 Marco Teórico

Este capítulo presenta los fundamentos teóricos utilizados para realizar el control de velocidad en motores de corriente directa. Se presenta el modelo matemático que describe el comportamiento de los motores de CD, después el proceso de identificación de los parámetros del modelo, seguido del diseño del algoritmo de control para una respuesta óptima del sistema. La parte final aborda la identificación y compensación de los fenómenos de zona muerta y variación de carga.

### 2.1 Modelo Matemático

Avances recientes en el desarrollo de materiales magnéticos y en dispositivos electrónicos de estado sólido han hecho posible la miniaturización de máquinas eléctricas y sus elementos de control, gracias a esto, los motores de corriente directa con imanes permanentes en sus polos se han vuelto muy populares en servosistemas. Los motores más comunes de este tipo, son los motores de corriente directa de imán permanente con escobillas y los motores de corriente directa de imán permanente sin escobillas, PMDCM y PMBDCM por sus siglas en inglés, este trabajo trata sólo de los primeros, en adelante referidos por simplicidad únicamente como motores de corriente directa o motores de CD.

En esta sección, se presenta como punto de partida el modelo matemático lineal para motores de corriente directa denominado modelo electromecánico. No se tratan a detalle los principios de funcionamiento y conversión de energía relacionados con las máquinas de corriente directa, estos pueden ser consultados en [Fitzgerald 1985] y [Chapman 1987]. Modelos más complejos sobre el comportamiento dinámico de los motores de CD se presentan en [Glowacz 2007] y [Lobosco 1997].

El modelo electromecánico estándar [Kuo 1978] considera el voltaje de armadura y el par debido a la carga como variables de entrada y a la velocidad como variable de salida. Los componentes principales de los motores de CD son modelados como se muestra en la figura 2-1.

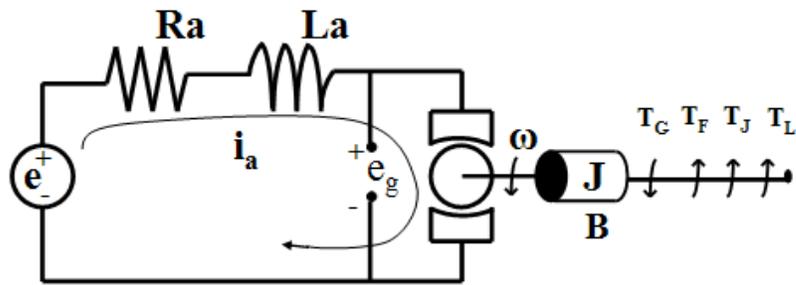


Figura 2-1.- Modelo electromecánico de un motor de corriente directa

Donde los parámetros  $i_a$ ,  $R_a$  y  $L_a$  son la corriente, la resistencia y la inductancia de armadura, respectivamente;  $e$  es la tensión de entrada al motor;  $e_g$  es un voltaje generado que se produce cuando los conductores de la armadura se mueven a través del campo magnético establecido por los imanes permanentes de los polos del motor;  $\omega$  es la velocidad angular;  $T_G$ ,  $T_F$ ,  $T_J$  y  $T_L$  son el par generado, el par debido a la fricción, el par debido a la inercia y el par de carga respectivamente;  $J$  representa el momento de inercia y  $B$  el coeficiente de fricción viscosa.

Aplicando la ley de tensiones de Kirchhoff al modelo de la figura 2.1

$$e(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_g(t) \quad (1)$$

De acuerdo a la ley de inducción de Faraday, una vuelta de bobina conductora que gira a través de un campo magnético constante, presenta en sus extremos un voltaje inducido

$$v(t) = \frac{d\lambda(t)}{dt} \quad (2)$$

donde  $\lambda(t)$  representa los enlaces del flujo magnético en la bobina. En un motor de corriente directa, cada bobina del rotor genera sobre ella un voltaje dado por la ecuación 2, donde  $d\lambda(t)/dt$  es proporcional al flujo magnético y a la velocidad angular  $\omega(t)$ , es decir

$$e_g(t) = K\phi(t)\omega(t) \quad (3)$$

Ignorando los cambios en el flujo magnético debidos a la reacción de armadura y otros efectos, y ya que el campo magnético es generado por imanes permanentes, el flujo del campo es considerado constante, y la ecuación 3 se modifica a

$$e_g(t) = K_e \omega(t) \quad (4)$$

donde  $K_e$  es la constante de velocidad del motor.

Otro efecto fundamental en la armadura es la inducción de una fuerza debida al flujo de corriente en las bobinas conductoras en presencia de un campo magnético perpendicular a este flujo. La magnitud de esta fuerza es proporcional a la magnitud de la corriente que fluye en los conductores, la intensidad del campo magnético y la longitud de los conductores. La fuerza inducida total es igual a la suma de las fuerzas individuales en cada bobina, esta fuerza opera a una distancia determinada por la construcción del rotor y de esta manera un par electromagnético es generado. Considerando un flujo constante, el par generado es proporcional a la corriente y está dado por

$$T_G(t) = K_t i_a(t) \quad (5)$$

donde  $K_t$  es la constante de par del motor.

La potencia mecánica en el motor está dada por

$$P_m(t) = T_G(t) \omega(t) \quad (6)$$

Una parte de la energía es almacenada como energía cinética en la masa del rotor, es importante notar que no toda esta potencia generada puede ser entregada a la carga conectada al eje del motor, ya que existen pérdidas debidas a la resistencia eléctrica de los devanados y a fenómenos de fricción, entre otros. Este tipo de pérdidas son consideradas en el modelo como pérdidas por fricción. El par generado debe ser en cualquier instante igual y opuesto a la suma del par necesario para romper la fricción, la inercia y el par de carga, por lo tanto

$$T_G = T_F(t) + T_J(t) + T_L(t) \quad (7)$$

El par inercial debido al momento de inercia del rotor y a la carga sobre el eje del motor, puede ser representado por el producto del momento de inercia total  $J$  y la aceleración angular  $d\omega(t)/dt$ . El par debido a la fricción tiene dos componentes, la componente del par total de fricción que es una función lineal de la velocidad angular, se denomina componente de fricción viscosa y es separada de otros términos que involucran fenómenos de fricción. Con estas consideraciones, la ecuación 7 puede reescribirse de la siguiente manera

$$T_G = T_F(t) + T_L(t) + J \frac{d\omega(t)}{dt} + B\omega(t) \quad (8)$$

donde  $B$  representa la fricción viscosa del motor y de la carga referida al eje del motor y  $T_F$  es la suma de los elementos considerados como pérdidas por fricción, excluyendo la fricción viscosa.

Las ecuaciones 1, 4, 5 y 8 forman las cuatro ecuaciones fundamentales que modelan a los motores de CD, combinaciones entre estas, llevan a funciones de transferencia (FT) para varios modos de operación del motor. Para manejar de manera mas sencilla este conjunto de ecuaciones, se obtiene la transformada de Laplace de cada una de ellas, y reacomodando para una representación en diagrama de bloques más intuitiva de estas expresiones, se tiene que

$$E(s) - E_g(s) = (R_a + sL_a)I_a(s) \quad (9)$$

$$E_g(s) = K_e\Omega(s) \quad (10)$$

$$T_G(s) = K_t I_a(s) \quad (11)$$

$$T_G(s) - T_F(s) - T_L(s) = (B + sJ)\Omega(s) \quad (12)$$

En la figura 2-2 se muestra la representación en diagrama de bloques del modelo electromecánico de los motores de corriente directa, el cual consiste de dos entradas:  $E(s)$  y  $T_L(s) + T_F(s)$  y una salida:  $\Omega(s)$ .

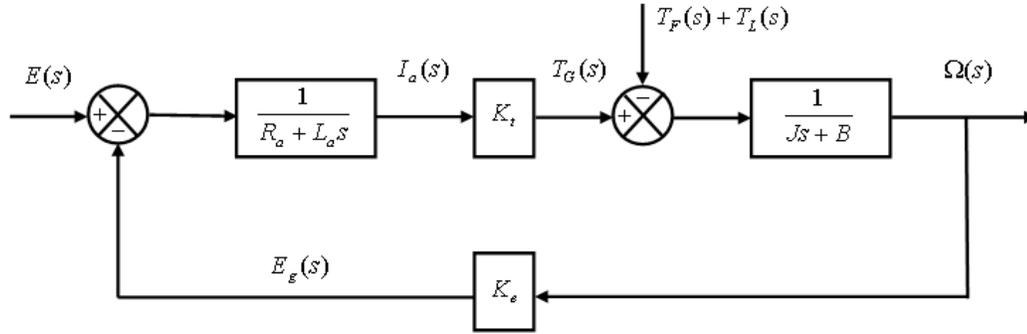


Figura 2-2.- Modelo en diagrama de bloques de los motores de CD

Utilizando el principio de superposición en el diagrama de bloques de la figura 2-2, la FT voltaje-velocidad puede representarse como

$$\frac{\Omega(s)}{E(s)} = \frac{K_t}{R_a B \left( \frac{L_a}{R_a} s + 1 \right) \left( \frac{J}{B} s + 1 \right) + K_t K_e} \quad (13)$$

donde  $L_a/R_a$  y  $J/B$  son las constantes de tiempo eléctrica y mecánica respectivamente del sistema de segundo orden descrito por esta FT.

Normalmente, el valor de  $L_a$  es de magnitud menor que  $R_a$ , de modo que, para valores pequeños de  $L_a : L_a \rightarrow 0$ , la constante de tiempo eléctrica es despreciable con respecto a la constante de tiempo mecánica, reduciendo la FT a un sistema de primer orden con la siguiente forma

$$\frac{\Omega(s)}{E(s)} = \frac{K_t}{R_a (J s + B) + K_t K_e} \quad (14)$$

Esta es una manera relativamente simple de eliminar polos insignificantes de la FT. Un enfoque más sofisticado para la simplificación de sistemas lineales mediante la reducción de orden de su FT puede consultarse en [Dorf 2005].

Utilizando el mismo análisis, la FT que considera el par de carga y la velocidad como variables de entrada y salida respectivamente es

$$\frac{\Omega(s)}{T_F(s) + T_L(s)} = \frac{-R_a \left( \frac{L_a}{R_a} s + 1 \right)}{R_a B \left( \frac{L_a}{R_a} s + 1 \right) \left( \frac{J}{B} s + 1 \right) + K_t K_e} \quad (15)$$

y para valores pequeños de  $L_a$

$$\frac{\Omega(s)}{T_F(s) + T_L(s)} = \frac{-R_a}{R_a (Js + B) + K_t K_e} \quad (16)$$

Las funciones de transferencia descritas en las ecuaciones 14 y 16, son la base para la realización de la identificación experimental de los parámetros del sistema, que condensados se presentan en las ecuaciones 17 y 18. Este método se aborda en la siguiente sección.

$$K_m = \frac{K_t}{R_a B + K_t K_e} \quad (17)$$

$$T_m = \frac{R_a J}{R_a B + K_t K_e} \quad (18)$$

## 2.2 Identificación del Sistema

La determinación de los coeficientes de la FT del modelo electromecánico del motor puede realizarse mediante varios métodos, los más comunes son a partir de los datos de placa del motor, mediante la medición directa o indirecta (estimación) de sus diferentes parámetros o utilizando alguna técnica de identificación experimental.

Este trabajo se basa en una técnica gráfica fuera de línea para la identificación experimental de sistemas, conocida como “*Identificación Gráfica de Parámetros como Respuesta a una Señal Escalón para Procesos de Primer Orden*” [Graupe 1972]. Otras técnicas utilizadas comúnmente para la identificación fuera de línea o en línea, se llevan a cabo mediante métodos para el ajuste de curvas mediante polinomios de interpolación de Lagrange [Burden 1997] o utilizando el método de los mínimos cuadrados [Markovsky 2005]. En todos los casos se

requiere aplicar una señal de excitación al sistema, para la técnica que se aplicó, se trata de una señal escalón. En la figura 2-3 se muestra el diagrama de bloques del proceso de identificación.

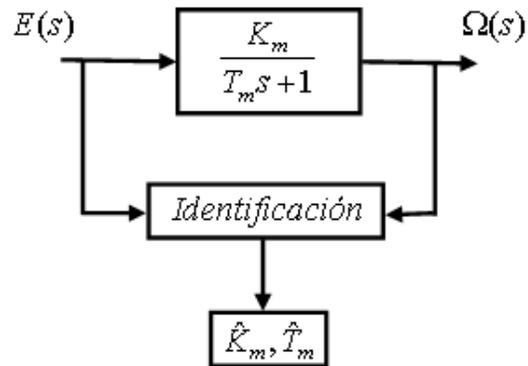


Figura 2-3.- Proceso de identificación para un sistema de primer orden

El proceso descrito en la figura 2-3 consiste en aplicar una excitación de entrada de tipo escalón y observar el comportamiento (salida) del sistema para estimar los parámetros del modelo.

Para la identificación experimental de los parámetros del sistema, se considera que este se encuentra libre de perturbaciones, de esta manera, la FT que describe el comportamiento total del sistema es la descrita en la ecuación 14, que con sus parámetros condensados puede escribirse como

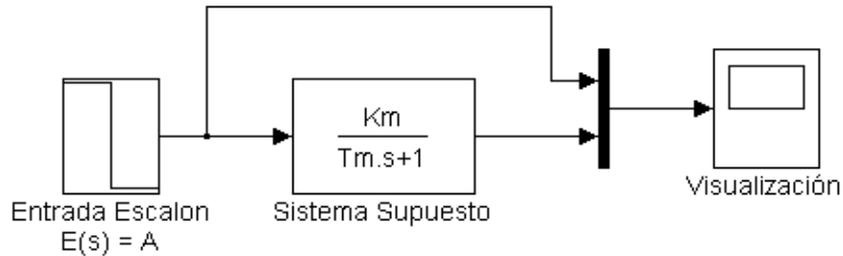
$$\frac{\Omega(s)}{E(s)} = \frac{K_m}{T_m s + 1} \quad (19)$$

Para conocer numéricamente el valor de  $K_m$  y  $T_m$ , es de utilidad encontrar la representación temporal de la FT del sistema. Resolviendo la transformada inversa de Laplace de la ecuación 19, se tiene que

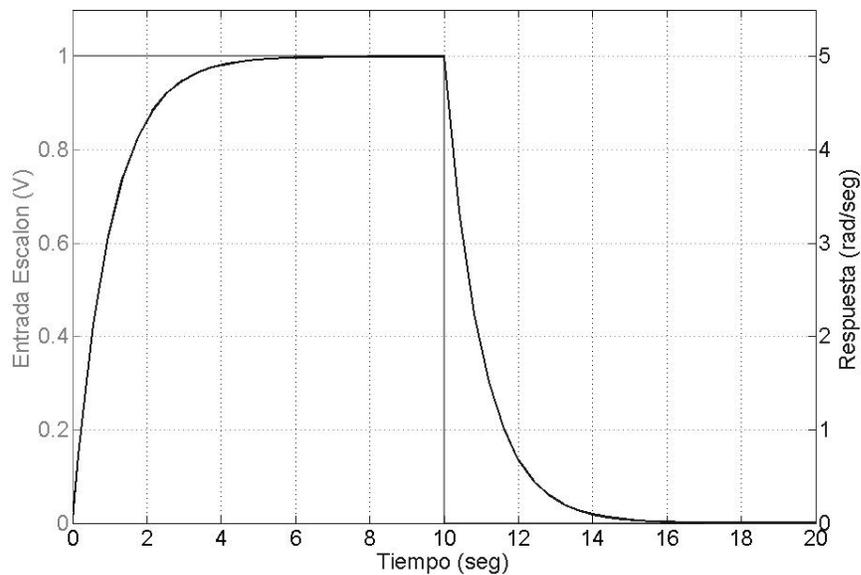
$$w(t) = AK_m \left(1 - e^{-\frac{t}{T_m}}\right) \quad (20)$$

Donde, A es la amplitud de la señal de excitación para una entrada escalón y  $K_m$  y  $T_m$  son los valores que determinan, según lo modelado, el comportamiento del sistema ante un voltaje de entrada.

Para determinar el valor de  $K_m$  y  $T_m$ , se supone un sistema con la estructura de la ecuación 19 y se utiliza SIMULINK<sup>1</sup> para simular la respuesta en velocidad angular ante una entrada escalón  $E(s) = A/s$  Volts que tendría un motor de CD con esta estructura. En la figura 2-4 se muestra el diagrama de bloques y la respuesta a una entrada escalón de este sistema supuesto.



(a)



(b)

Figura 2-4.- Sistema supuesto, a) diagrama de bloques, b) respuesta al escalón unitario

---

<sup>1</sup> Todas las simulaciones realizadas en este trabajo fueron hechas utilizando MATLAB 7 (R14) y SIMULINK 6 (R14), ambos de The MathWorks, Inc.

Puesto que se trata de un modelo lineal, se espera que la respuesta del sistema, cuando este ha alcanzado su estado estacionario, se comporte de manera lineal con respecto a la amplitud de la señal de excitación, esto quiere decir que, si para un voltaje de entrada de 1 V, la respuesta en estado estacionario es de 5 rad/seg, como se muestra en la figura 2-4-b, entonces para una entrada de 24 V la respuesta del motor de CD debe ser de 120 rad/seg.

Para encontrar de manera experimental los valores de  $K_m$  y  $T_m$  se aplica, partiendo del reposo, el voltaje nominal al motor de CD el tiempo suficiente para que este alcance su régimen permanente, una vez que esto sucede, se suprime la alimentación hasta que el motor se encuentre nuevamente en reposo para enseguida volver a aplicar el voltaje nominal pero invirtiendo su polaridad de manera que el motor gire hacia el lado contrario y se repite el procedimiento varias veces para asegurar la repetibilidad de la respuesta.

Se considera que el sistema ha alcanzado su régimen permanente una vez que la velocidad angular se mantiene estable en respuesta al escalón de excitación, esto equivale en el modelo electromecánico a evaluar la ecuación 20 cuando el tiempo tiende a infinito, es decir

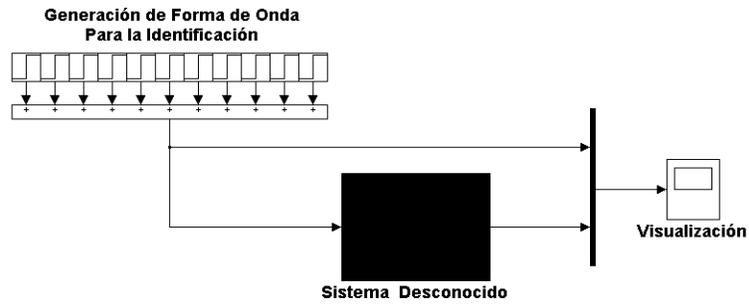
$$w(t) = AK_m (1 - e^{-\frac{t}{T_m}}) = AK_m \quad (21)$$

$$t \rightarrow \infty$$

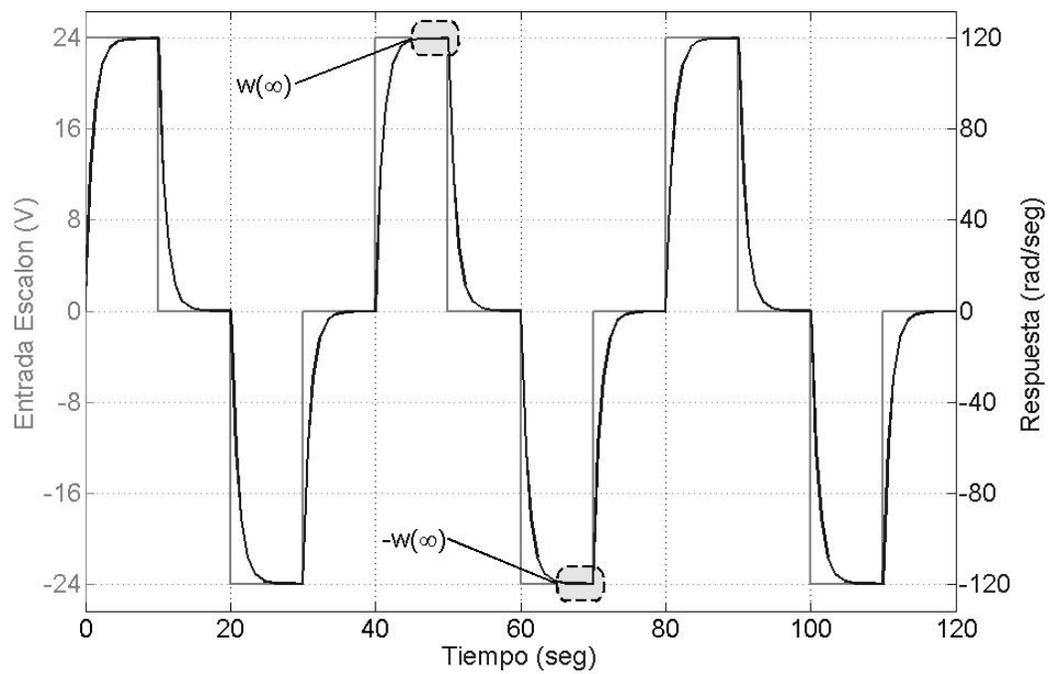
de esta manera

$$K_m = \frac{w(\infty)}{A} \quad (22)$$

donde  $w(\infty)$  se obtiene al medir la velocidad angular cuando el sistema ha alcanzado su estado estacionario y A es el máximo voltaje aplicado. Este procedimiento se describe utilizando el modelo de la figura 2-4, suponiendo que no se conocen sus parámetros. El diagrama de bloques en SIMULINK y la gráfica de la respuesta se muestran en la figura 2-5.



(a)



(b)

Figura 2-5.- Proceso de identificación, a) diagrama de bloques, b) respuesta del sistema desconocido

Con el conocimiento del voltaje aplicado y observando  $w(\infty)$  en la gráfica de la figura 2-5-b, utilizando la ecuación 22 se obtiene el valor de  $K_m$ , de modo que para el sistema supuesto se tiene que

$$K_m = \frac{w(\infty)}{A} = \frac{120 \frac{rad}{seg}}{24V} = 5 \frac{rad}{seg \cdot V} \quad (23)$$

Con esto, sólo resta encontrar el valor de  $T_m$  para completar el proceso de identificación del motor de CD. Según [Dorf 2005],  $T_m$  es la constante de tiempo, y se refiere al intervalo de tiempo necesario para que un sistema cambie de un estado a otro, un porcentaje especificado. Para un sistema de primer orden, la constante de tiempo es el tiempo que se requiere para que la salida manifieste un cambio del 63.2% debido a una entrada escalón.

De la misma forma que para la determinación de  $K_m$ , se observa en la figura 2-5-b el valor de  $w(\infty)$ , se multiplica por 0.632 y se determina el tiempo que tomó al sistema alcanzar tal valor de velocidad, este tiempo es el tiempo correspondiente a  $T_m$ . La constante de tiempo  $T_m$  puede también encontrarse al trazar una recta desde  $t=0$ , con la pendiente inicial (la tangente en  $t=0$ ) de la respuesta al escalón hasta que esta se interseca con el valor en amplitud correspondiente a la respuesta en estado estacionario. El punto de intersección entre la recta y  $w(\infty)$  se alcanza en un tiempo igual  $T_m$  después de haber sido aplicado el voltaje de alimentación. La pendiente en  $t=0$  se encuentra al derivar la ecuación 20 y evaluarla para  $t=0$ , como se muestra a continuación

$$\left. \frac{dw(t)}{dt} \right|_{t=0} = \frac{AK_m}{T_m} e^{-\frac{t}{T_m}} \Big|_{t=0} = \frac{AK_m}{T_m} \quad (24)$$

El comportamiento con respecto al tiempo para una recta cuya pendiente es la descrita por la ecuación 24 es

$$x(t) = \frac{AK_m t}{T_m} \quad (25)$$

alcanzando el valor de  $AK_m = w(\infty)$  cuando  $t=T_m$ .

Para mostrar ambos métodos y establecer el valor de  $T_m$  para el sistema de la figura 2-5, se hace un acercamiento sobre la respuesta, solo al primer escalón aplicado, como se muestra en la figura 2-6. Se observa que  $x(t)$  de la ecuación 25, alcanza el valor de  $w(\infty)$  en un tiempo

aproximado de 1seg y  $w(t)$  toma el valor de  $0.632w(\infty)$  en el mismo lapso de tiempo, demostrando así que ambos métodos son equivalentes en la determinación de  $T_m$ , por tanto

$$0.632w(\infty) = AK_m(1 - e^{-\frac{1}{T_m}}) = w(1) \quad \therefore \quad 0.632 = 1 - e^{-\frac{1}{T_m}} \quad (26)$$

$$T_m = -\frac{1}{\ln(1 - 0.632)} \approx 1\text{seg} = t$$

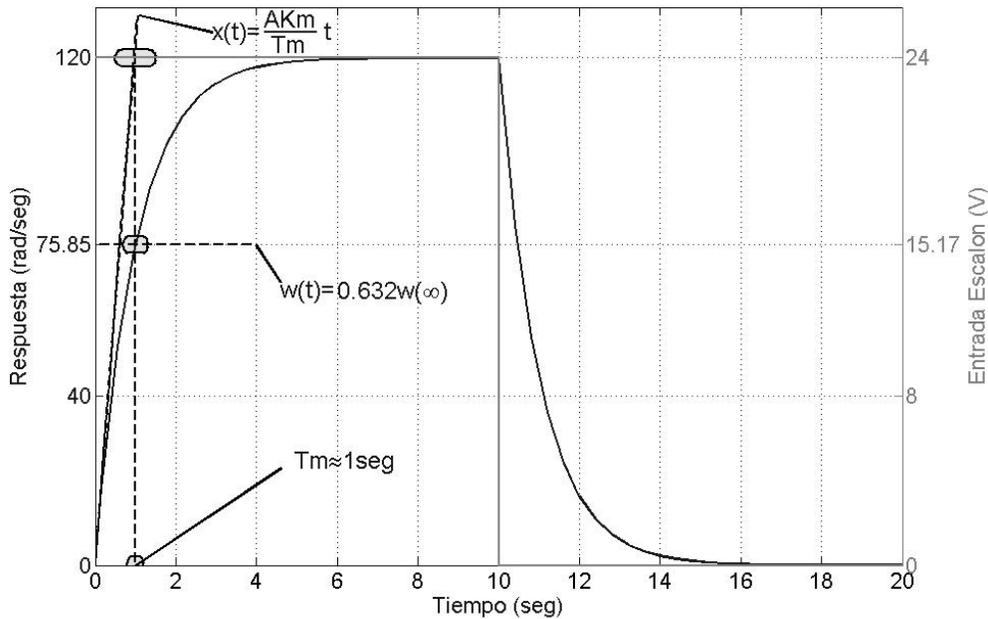


Figura 2-6.- Proceso de identificación, determinación gráfica de  $T_m$

Una vez determinados  $K_m$  y  $T_m$ , del análisis de las figuras 2-5-b y 2-6 mediante el método gráfico de identificación experimental, se puede concluir que la FT que describe el comportamiento del sistema antes desconocido es

$$\frac{\Omega(s)}{E(s)} = \frac{K_m}{T_m s + 1} = \frac{5}{s + 1} \quad (27)$$

Con el conocimiento de la FT del motor de corriente directa, se puede ahora diseñar un algoritmo de control que permita manipular el comportamiento del sistema dentro de límites especificados. El diseño del algoritmo de control se propone en la siguiente sección.

## 2.3 Diseño del Controlador

Mientras mejor conocimiento se tiene de un sistema, resulta más fácil decidir que estrategia de control ha de utilizarse, y por tanto el desempeño real de este, bajo la acción de control, será más parecido a lo esperado por los modelos y simulaciones realizadas. Fue este el objetivo de las secciones anteriores. Aunque existen numerosos modelos y teorías para el control automático de sistemas, el controlador Proporcional, Integral, Derivativo (PID), es por mucho, el más utilizado con fines académicos e industriales ya que además de su presencia histórica —que de acuerdo con [Åström 2006] data por lo menos alrededor de 250 años—, los conceptos sobre los que encuentra fundamento son en cierto modo sencillos e incluso intuitivos. Debido a estos factores y a su facilidad de implementación, este trabajo basa su estrategia para el control de velocidad de motores de corriente directa en el algoritmo de control PID.

### 2.3.1 Control PI

En general, un sistema de control requiere una buena respuesta a cambios en la señal de referencia, atenuación a perturbaciones, inmunidad al ruido e insensibilidad a variaciones en el proceso a controlar. El diseño de un sistema de control también incluye aspectos sobre la dinámica del sistema, saturación en los actuadores y el tipo de perturbaciones a las que puede estar sometido. Sorprendentemente, un controlador tan simple como el PID ha resultado a través de los años funcionar razonablemente bien para la mayoría de los procesos industriales, aunque cabe mencionar que para sistemas con un orden superior a dos, sistemas con largos tiempos de retraso y sistemas con modos oscilatorios, el control puede ser mejorado utilizando controladores más complejos que el PID [Åström 2006].

La estructura estándar descrita en los libros de texto para un controlador PID es la siguiente

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (28)$$

la cual puede expresarse como

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (29)$$

y al aplicar la transformada de Laplace se tiene

$$U(s) = K_p E(s) + \frac{K_i}{s} E(s) + K_d s E(s) \quad (30)$$

donde U es la señal de control, E el error de control y los parámetros del controlador  $K_p$ ,  $K_i$  y  $K_d$  son las ganancias proporcional, integral y derivativa respectivamente. Mediante la correcta selección de estos valores es posible ajustar la respuesta del sistema a controlar. Una explicación detallada y resumida de la contribución de cada una de las ganancias se encuentra en [Åström 1995].

No siempre es necesaria la presencia de estas tres ganancias en el control, para sistemas estables y con requerimientos modestos de desempeño, la acción integral (I) es más que suficiente, aunque añadir la parte proporcional (P) brinda mejoras adicionales. En general, puede demostrarse que un controlador PI es adecuado para todos los procesos donde la dinámica es esencialmente de primer orden. Por tanto, sin considerar efectos de saturación, para el caso del sistema que se ha supuesto de primer orden, identificado en la sección 2.2 conforme al modelo de orden reducido de la sección 2.1, correspondiente a la respuesta al escalón de un motor de corriente directa, un esquema de control PI es suficiente. El diagrama de bloques del sistema de control de velocidad para el modelo electromecánico del motor de CD, con un controlador PI se muestra en la figura 2-7.

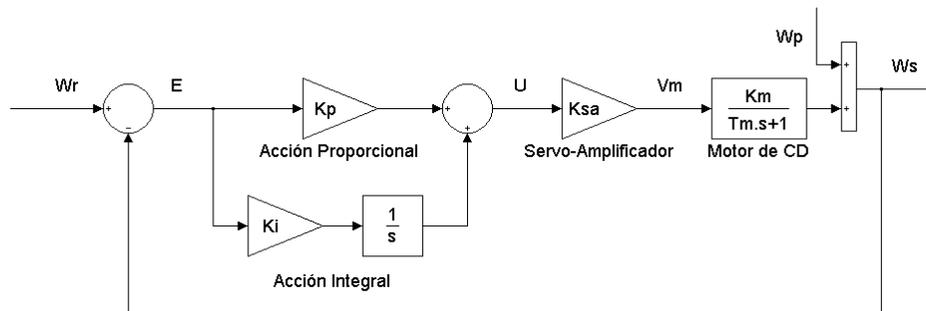


Figura 2-7.- Diagrama de bloques del sistema de control

Donde  $W_r$  es la velocidad de referencia,  $E$  es la señal de error,  $K_p$  y  $K_i$  son las ganancias del controlador PI,  $U$  la señal control,  $K_m$  y  $T_m$  los parámetros obtenidos del proceso de identificación del motor de CD y  $K_{sa}$  la constante de proporcionalidad entre la señal de control  $U$  y el voltaje de alimentación del motor  $V_m$  considerando que la respuesta en velocidad del motor es lineal con respecto al voltaje de alimentación suministrado por el servo-amplificador. Las perturbaciones debidas a cambios en el par de carga, como las descritas por la ecuación 16 de la sección 2.1, se relacionan directamente con variaciones en la velocidad de salida  $W_s$  y pueden por tanto ser simuladas como perturbaciones en velocidad como lo indica  $W_p$ .

El sistema de control que se ilustra en la figura 2-7, puede ser expresado en forma reducida por la siguiente FT

$$\frac{W_s(s)}{W_r(s)} = \frac{\frac{K_{sa}K_mK_i}{T_m} + \frac{K_{sa}K_mK_p}{T_m}s}{s^2 + \frac{K_{sa}K_mK_p + 1}{T_m}s + \frac{K_{sa}K_mK_i}{T_m}} \quad (31)$$

Puesto que se ha considerado un comportamiento lineal proporcional al voltaje de alimentación, el producto  $K_{sa}K_m$  puede considerarse unitario, transformando la FT de la ecuación 31 en

$$\frac{W_s(s)}{W_r(s)} = \frac{\frac{K_i}{T_m} + \frac{K_p}{T_m}s}{s^2 + \frac{K_p + 1}{T_m}s + \frac{K_i}{T_m}} \quad (32)$$

Si se considera despreciable la influencia del cero en la FT de la ecuación 32, esta puede reescribirse como

$$\frac{W_s(s)}{W_r(s)} = \frac{\frac{K_i}{T_m}}{s^2 + \frac{K_p + 1}{T_m}s + \frac{K_i}{T_m}} \quad (33)$$

La FT de la ecuación 33 servirá como base para la sintonización de los parámetros de controlador.

### 2.3.2 Sintonización

Se conoce como sintonización al proceso de selección de las ganancias del controlador, de acuerdo a las especificaciones de desempeño que se desea el sistema tenga. Existen diferentes técnicas para encontrar estos valores, entre ellas las que más han sido utilizadas en los últimos 50 años se basan en el trabajo de [Ziegler 1942], en el cual se propone un procedimiento para encontrar las ganancias óptimas del controlador basándose en la observación de la respuesta del sistema ante entradas de tipo escalón y su respuesta en frecuencia, esta técnica se analiza en [Åström 2006], además del planteamiento de algunas otras que ofrecen mejoras a los métodos de Ziegler-Nichols, como el método desarrollado por [Chien 1952]. La primera aproximación utilizada en este trabajo, para la sintonización de los parámetros del controlador, se hace utilizando los índices de desempeño ITAE, descrito por [Dorf 1995] basado en el trabajo de [Lathrop 1953].

De sus siglas en inglés el índice de desempeño ITAE, es la integral del tiempo multiplicado por el valor absoluto del error. Lo que este método busca, son los coeficientes de la función de transferencia en lazo cerrado generalizada, tal que el ITAE, ya sea para una entrada escalón o para una rampa, se vea minimizado. Los coeficientes óptimos para la ecuación característica de la FT de un sistema en lazo cerrado de segundo orden, como el que se muestra en ecuación 33 y cuya forma generaliza es la descrita por la ecuación 34, basados en el criterio ITAE para una entrada escalón se muestran en la ecuación 35.

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (34)$$

donde  $Y(s)$  y  $R(s)$  son las señales de salida y referencia respectivamente,  $\xi$  es la razón adimensional de amortiguamiento y  $\omega_n$  la frecuencia natural del sistema.

$$s^2 + 1.4\omega_n s + \omega_n^2 \quad (35)$$

Relacionando las ecuaciones 33, 34 y 35

$$\omega_n^2 = \frac{K_i}{T_m} \quad (36)$$

$$2\xi\omega_n = 1.4\omega_n = \frac{K_p + 1}{T_m} \quad (37)$$

Con esto, lo que resta es encontrar  $\omega_n$  para hacer uso de las relaciones anteriores, determinar los valores de  $K_p$  y  $K_i$  para los que el sistema mantendrá una respuesta óptima a entradas del tipo escalón. De acuerdo con [Kuo 1996], el tiempo de establecimiento después de aplicar una entrada escalón a un sistema de segundo orden, utilizando el criterio del 2% está dado por

$$t_{ss} = \frac{4}{\xi\omega_n} \quad (38)$$

por tanto, al proponer el tiempo de establecimiento deseado, es posible determinar el valor de  $\omega_n$  y con el conocimiento previo de  $K_m$  y  $T_m$  debido a la identificación experimental, sustituyendo en las ecuaciones 36 y 37, se determinan los valores óptimos de  $K_p$  y  $K_i$  para el sistema de control de la figura 2-7.

Para mostrar la simplicidad y utilidad del método, se utilizará el sistema identificado en la sección 2.2, se encontrarán los valores de las ganancias  $K_p$  y  $K_i$  para este y se analizará el desempeño del controlador ante entradas de tipo escalón con diferentes amplitudes, observando la salida del sistema. Se espera obtener el mismo desempeño para cada una de las entradas.

De la ecuación 37 se deduce que la razón óptima de amortiguamiento  $\xi$ , de acuerdo con el criterio ITAE tiene un valor de 0.7 puesto que

$$\begin{aligned} 2\xi\omega_n &= 1.4\omega_n \\ \therefore & \\ \xi &= \frac{1.4}{2} = 0.7 \end{aligned} \quad (39)$$

Si se requiere que el sistema tenga un tiempo de establecimiento de 2seg, entonces de la ecuación 38, el valor numérico de  $\omega_n$  es

$$\omega_n = \frac{4}{t_{ss}\xi} = \frac{4}{2 \cdot 0.7} \approx 2.86 \quad (40)$$

En la ecuación 26, para el sistema bajo análisis, se encontró que  $T_m=1\text{seg}$ , por lo tanto, sustituyendo este valor y el de la ecuación 40 en la ecuación 36, se tiene que el valor de  $K_i$  está dado por

$$K_i = T_m \omega_n^2 = 1 \cdot 2.86^2 \approx 8.16 \quad (41)$$

Finalmente, sustituyendo en la ecuación 37

$$K_p = 1.4T_m \omega_n - 1 = 1.4 \cdot 1 \cdot 2.86 - 1 \approx 3 \quad (42)$$

Una vez determinadas  $K_p$  y  $K_i$ , el proceso de sintonización ha concluido. El diagrama de bloques del sistema identificado en la sección 2.2 con un controlador PI y sus ganancias sintonizadas para un desempeño óptimo de acuerdo con los índices del criterio ITAE, se muestra en la figura 2-8.

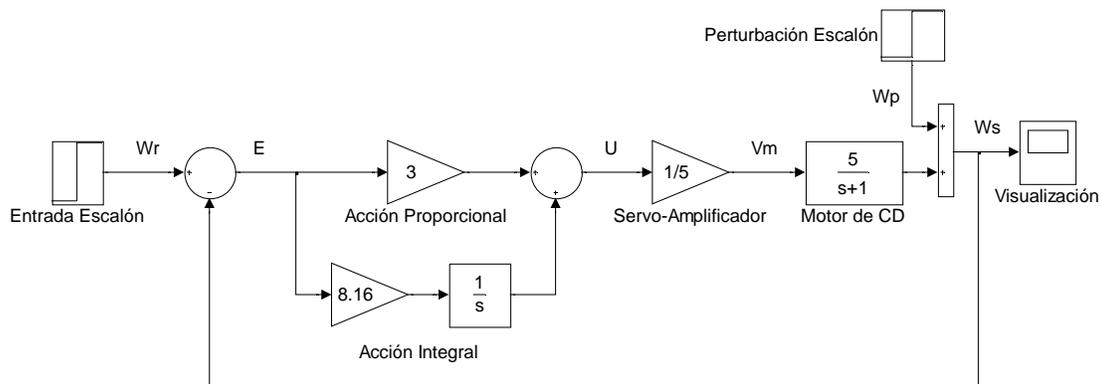
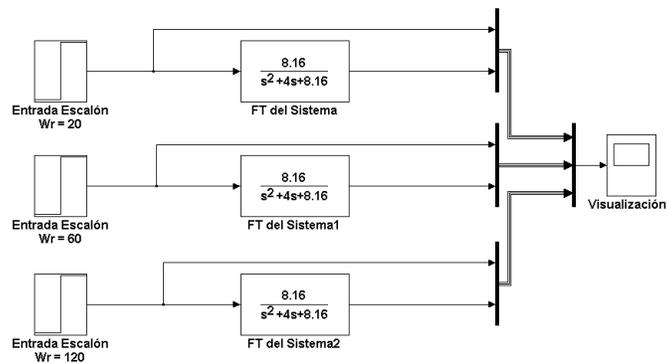


Figura 2-8.- Diagrama de bloques del sistema de control sintonizado

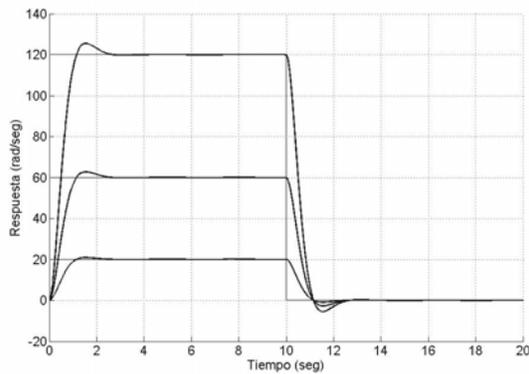
De acuerdo a la ecuación 33, sin considerar perturbaciones, la FT del sistema de la figura 2-8 es

$$\frac{W_s(s)}{W_r(s)} = \frac{8.16}{s^2 + 4s + 8.16} \quad (43)$$

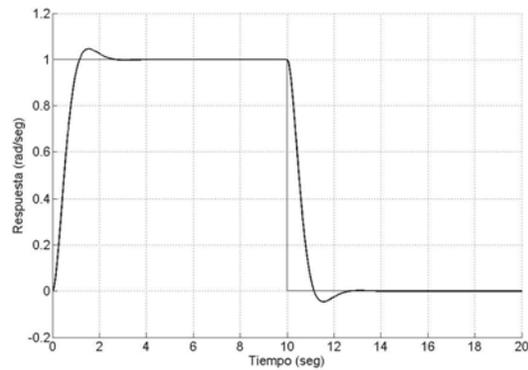
La simulación del algoritmo de control se expone en la figura 2-9 y muestra que las respuestas a diferentes señales de referencia para el sistema de la figura 2-8 tienen siempre el mismo desempeño. Conforme a los índices de desempeño utilizados, la razón de amortiguamiento es  $\xi=0.7$ , lo que corresponde a un máximo sobre-impulso de aproximadamente 5% en la respuesta a una entrada escalón y conforme al tiempo de establecimiento que se fijó, el sistema debe de entrar en la banda del 2% de su valor final en un tiempo de 2seg. En la figura 2-9-c se observa la respuesta normalizada del sistema de control de las entradas graficadas en la figura 2-9-b, en esta se puede constatar que el máximo sobre-impulso y el tiempo de establecimiento son conforme a lo esperado.



(a)



(b)



(c)

Figura 2-9.- Simulación de respuestas para señales de referencia de tipo escalón.a) Diagrama de bloques, b) Respuestas a referencias de 20, 60 y 120rad/seg, c) Respuestas normalizadas

## 2.4 Zona Muerta

El concepto de zona muerta se refiere al rango para el cual las señales de entrada no provocan efecto alguno en la salida del sistema. La zona muerta en motores de CD se relaciona de manera directa con el par necesario para iniciar el movimiento partiendo del reposo. Algunos autores se refieren a este concepto como fricción estática o sticktion [Armstrong 1991]. Al voltaje necesario para sobrepasar la fuerza debida a la fricción estática, se le conoce como voltaje de rompimiento ( $V_r$ ). A diferencia de la fricción viscosa, incluida en el modelo electromecánico descrito en la sección 2.1, cuyo par se considera proporcional a la velocidad angular, el par debido a la fricción estática es un fenómeno no lineal y por tanto el controlador PI basado en la teoría clásica de control, desarrollado en la sección anterior, es incapaz de compensarlo. Debido a esta limitación, y a los efectos negativos que la zona muerta causa en el posicionamiento y movimiento a bajas velocidades en motores de corriente directa, es necesario incluir en el lazo de control un algoritmo que se anticipe a sus efectos y ayude a corregirlos. Aunque existen modelos más completos como en [Canudas 1995], este trabajo se basa en el método para la compensación de zona muerta que se presenta en [Castillo 2006], el cuál realiza una identificación experimental de  $V_r$  y utiliza la función signo para agregarlo a la señal de control.

### 2.4.1 Efectos

Los errores en el problema de seguimiento, causados por el efecto de la zona muerta, se presentan cuando existen cambios de dirección en el movimiento, lo que corresponde, en la vecindad del cambio, a regiones en las que las velocidades tienden a ser muy bajas, alcanzan un nivel de cero y vuelven a subir de nivel, hasta abandonar la región de zona muerta. En máquinas de control numérico convencionales y robots manipuladores, el método empleado con regularidad para la identificación de fuentes de error, entre las que se encuentran los errores en el posicionamiento provocados por el fenómeno de sticktion, se realizan mediante un experimento que se conoce como prueba circular [Hang 1997], que consiste en programar las trayectorias de movimiento en los ejes x e y, de tal forma que la herramienta se mueva alrededor de un círculo. Esta prueba requiere que los ejes x e y sigan trayectorias sinusoidales en posición, también correspondientes a referencias sinusoidales en velocidad.

Para mostrar los efectos negativos en el posicionamiento, debidos a la zona muerta en motores de corriente directa, en la figura 2-10 se presentan los resultados de la simulación de los efectos de sticktion en para el problema de seguimiento de una trayectoria circular para un robot paralelo planar [Castillo 2007].

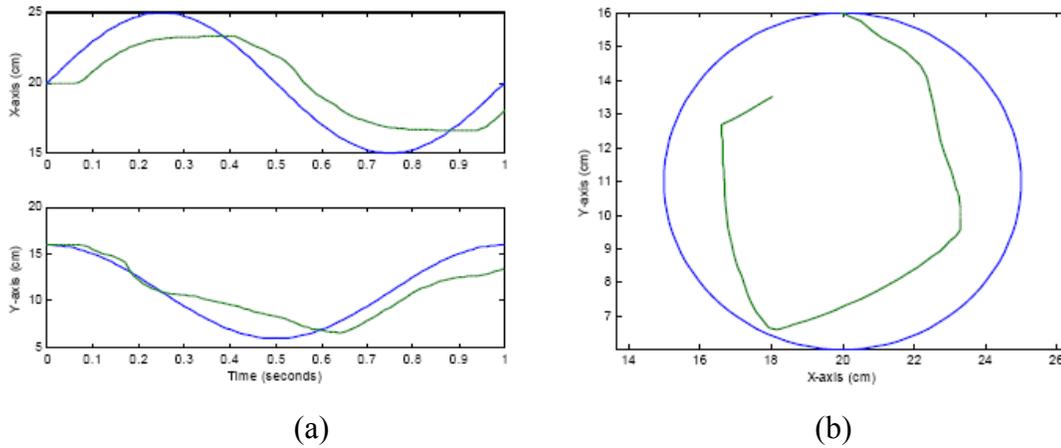


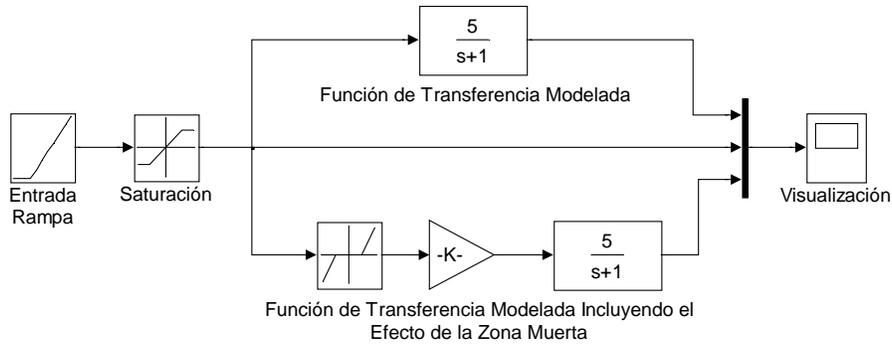
Figura 2-10.- Simulación del efecto de sticktion para una trayectoria circular.  
a) Componentes X-Y, b) Curva X vs. Y.

## 2.4.2 Compensación

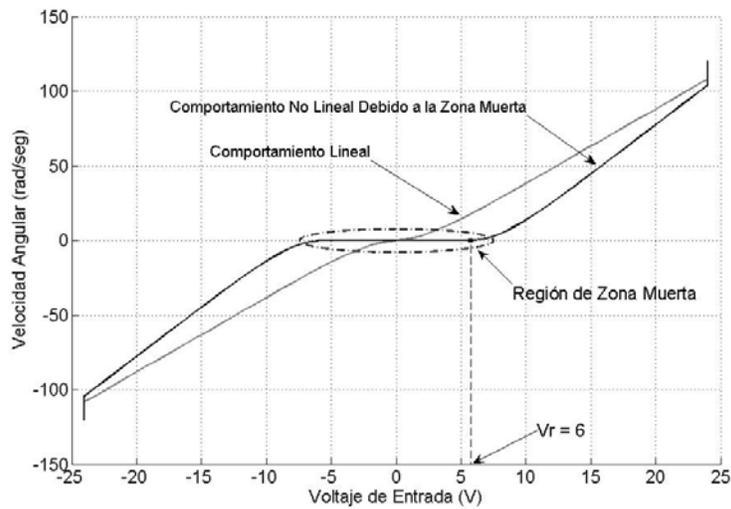
El primer paso para la compensación de la zona muerta, es realizar la identificación experimental del voltaje de rompimiento  $V_r$ . Para esto, se aplica un voltaje rampa en lazo abierto al motor de CD, registrando el voltaje de entrada al motor y la velocidad angular de salida, se grafica la velocidad con respecto al voltaje y se determina cuándo el voltaje aplicado comienza a generar movimiento, este voltaje es  $V_r$ . Para mostrar el proceso de identificación de  $V_r$ , se utilizará la FT del motor de CD que se identificó en la sección 2.2. La respuesta que se espera del modelo electromecánico y el comportamiento considerando el efecto de la fricción estática según el procedimiento del párrafo anterior, se muestran en la figura 2-11.

Aunque la sintonización del controlador realizada en la sección 2.3.2, se basó en los índices de desempeño para una entrada escalón, las ganancias seleccionadas siguen funcionando para referencias sinusoidales, siempre que, de acuerdo a la ecuación 44 [Dorf 2005], la frecuencia de dichas referencias sea menor a la frecuencia de ancho de banda  $\omega_B$ .

$$\omega_B = (-1.196\xi + 1.85)\omega_n ; \quad 0.3 \leq \xi \leq 0.8 \quad (44)$$



(a)



(b)

Figura 2-11.- Identificación del voltaje de rompimiento  $V_r$ , a) Diagrama de bloques, b) Determinación de  $V_r$

Para el sistema de control de la figura 2-8, la frecuencia de la señal de referencia debe ser menor o igual a

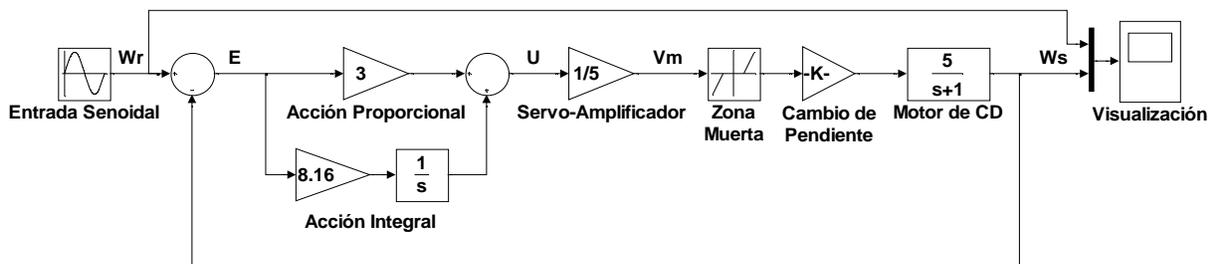
$$\omega_B = (-1.196 \cdot 0.7 + 1.85) \cdot 2.86 \approx 2.9 \frac{rad}{seg} \quad (45)$$

Puesto que el efecto de la zona muerta se presenta a bajas velocidades y las pruebas de desempeño de las máquinas de control numérico y robots manipuladores, requieren señales de referencia sinusoidales, es indispensable tomar en cuenta los límites que fijan las ecuaciones anteriores.

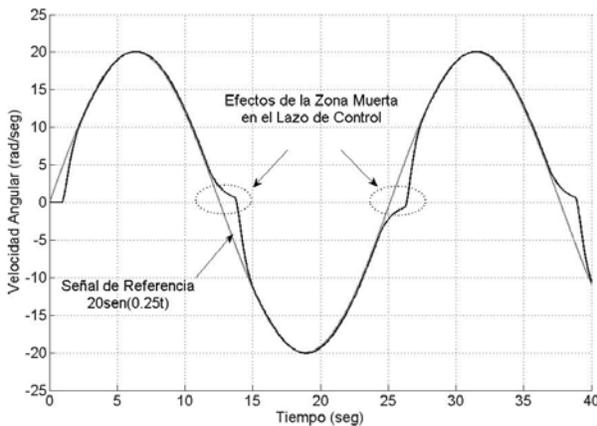
En la figura 2-12, se muestra el efecto que tiene la zona muerta para el sistema de control la figura 2-8 en el seguimiento de una señal de referencia de  $20\text{sen}(\omega t)$  con  $\omega=0.25\text{rad/seg}$ . Se escoge esta señal como ejemplo ya que la mínima velocidad de referencia para el arreglo experimental será de  $20\text{rad/seg}$  en amplitud y  $\omega=0.25\text{rad/seg}$  porque cumple con la ecuación 44 para este sistema.

Para evitar los efectos provocados por el fenómeno de sticktion, basados en el método presentado por [Castillo 2006], una vez identificado el voltaje de rompimiento  $V_r$ , el cuál se sabe por la figura 2-11 tiene un valor absoluto de  $6\text{V}$ , la señal de control con compensación de la zona muerta será

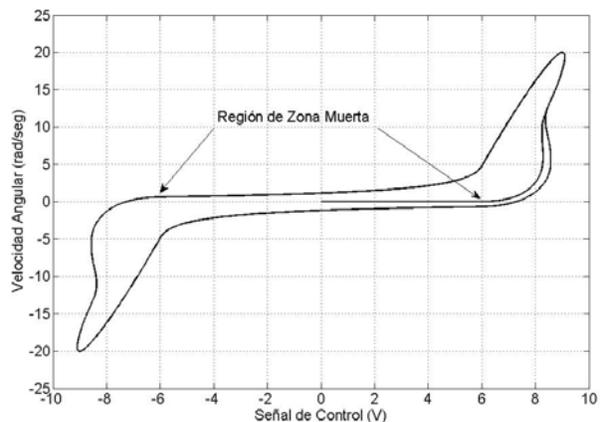
$$U = U + \text{sign}(U) \cdot V_r \quad (46)$$



(a)



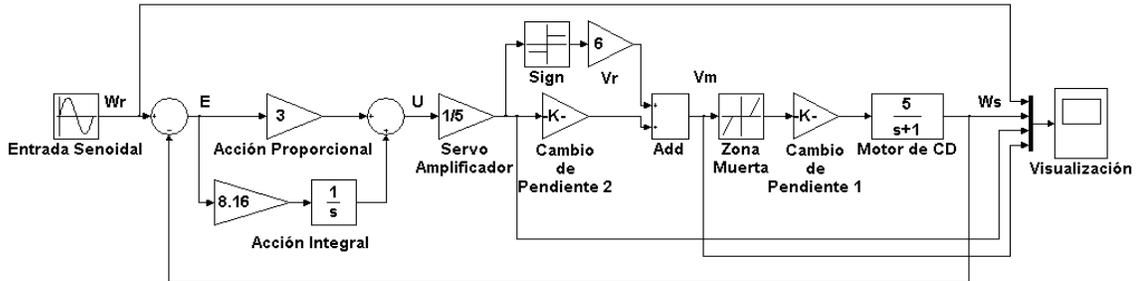
(b)



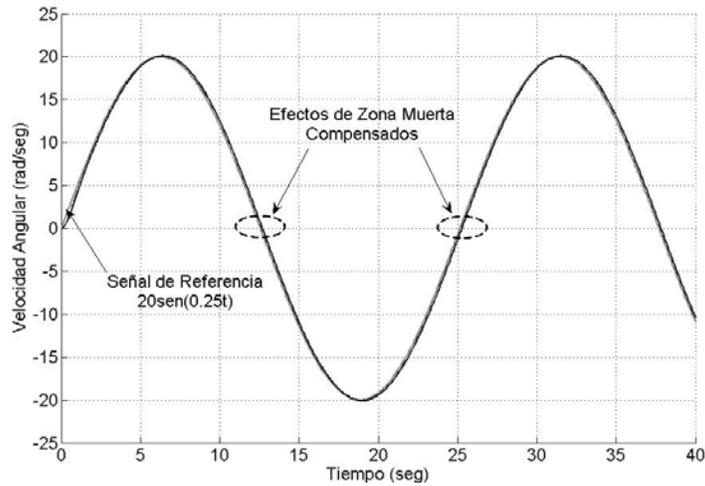
(c)

Figura 2-12.- Efecto de la zona muerta en el lazo de control. a) Diagrama de bloques, b) Efectos de la zona muerta, c) Esfuerzo de control

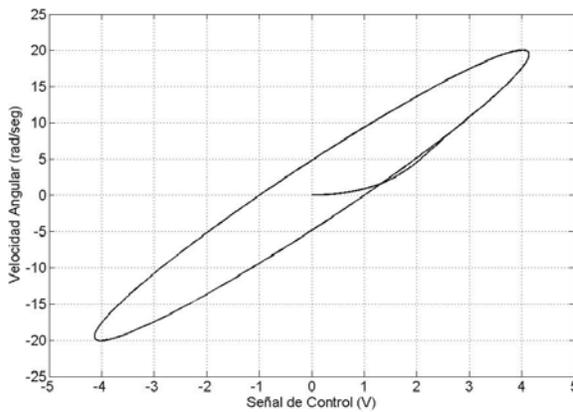
El diagrama de bloques del sistema de control que incluye la compensación de la zona muerta se muestra en la figura 2-13-a. La figura 2-13-b muestra el seguimiento de una señal de referencia de  $20\text{sen}(\omega t)$  para  $\omega=0.25\text{rad/seg}$  con compensación de zona muerta, en la figura 2-13-c y 2-13-d se observa el esfuerzo de control y el voltaje de entrada al motor respectivamente.



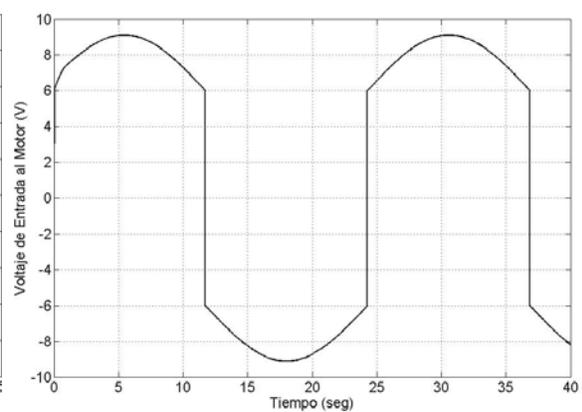
(a)



(b)



(c)



(d)

Figura 2-13.- Compensación de zona muerta mediante la función signo. a) Diagrama de bloques, b) Seguimiento con compensación, c) Esfuerzo de control y d) Voltaje de entrada al motor

## 2.5 Variación de Carga

Aunque, puede demostrarse que el control PI para el sistema de primer orden identificado en la sección 2.2 con las ganancias seleccionadas en la sección 2.3.2, conserva, ante perturbaciones de tipo escalón, un error en estado estacionario igual a cero, la velocidad de respuesta a estas variaciones puede mejorarse en algunos casos. Las perturbaciones en el control de velocidad de motores de corriente directa se refieren según el modelo electromecánico, a variaciones en el par necesario para mantener a estos girando a una velocidad deseada, y pueden verse, para el caso de escalones, cómo cambios súbitos en la carga sujeta a la flecha del motor, en adelante referidos como variaciones de carga.

Para lograr una respuesta más rápida ante variaciones de carga, en esta sección se muestra un procedimiento de selección de ganancias para el controlador PI, conocido como “set point gain scheduling”, basado en el modelo de primer orden del sistema identificado en la sección 2.2. Procedimientos parecidos pueden consultarse en [Beaven 1994] y [Yongpeng 2005], este último considera no sólo, la compensación de variaciones de tipo escalón, sino también periódicas.

### 2.5.1 Metodología de Compensación

De acuerdo al modelo electromecánico, la velocidad angular en función del tiempo, para motores de CD está dada por

$$w(t) = AK_m \left(1 - e^{-\frac{t}{T_m}}\right) \quad (47)$$

De manera general, para sistemas de primer orden, el tiempo mínimo necesario para alcanzar una velocidad de referencia  $\omega_r$  de tipo escalón, dentro del límite posible conforme al voltaje de alimentación, despejando el tiempo de la ecuación 47 es entonces

$$t_{\min} = -T_m \cdot \ln \left(1 - \frac{\omega_r}{AK_m}\right) \quad (48)$$

El tiempo de establecimiento fue uno de los parámetros de diseño utilizado en la sección 3.2, cuyo valor se consideró constante para la selección de las ganancias óptimas del controlador.

En esta metodología, el valor del tiempo de establecimiento cambiará según la magnitud de la entrada de referencia, el cual utilizando la ecuación 48 y conforme al criterio del 2% estará determinado por

$$t_{ss} = -T_m \cdot \ln\left(1 - \frac{0.98 \cdot \omega_r}{AK_m}\right) \quad (49)$$

La razón de amortiguamiento  $\xi$  de un sistema de segundo orden como el descrito por la ecuación 33, para un máximo sobre-impulso ( $M_p$ ) específico, en respuesta a una entrada de referencia de tipo escalón, está dada por

$$\xi = \sqrt{\frac{\ln(M_p)^2}{\ln(M_p)^2 + \pi^2}} \quad (50)$$

Una vez conocidos la razón de amortiguamiento y el tiempo de establecimiento, de acuerdo a la ecuación 38, la frecuencia natural del sistema será

$$\omega_n = \frac{4}{t_{ss} \cdot \xi} \quad (51)$$

De las ecuaciones 41 y 42,  $K_p$  y  $K_i$  están determinadas por

$$K_i = \omega_n^2 T_m \quad (52)$$

$$K_p = 2\xi\omega_n T_m - 1 \quad (53)$$

Conforme a las ecuaciones 14 y 16, la constante de tiempo del sistema para una perturbación de un par de entrada , , es la misma que para la entrada de referencia. Al cambiar el valor de  $t_{ss}$  conforme la magnitud de la entrada de referencia, siempre que esta se encuentre dentro de los límites permisibles, lo que se está haciendo es reducir implícitamente el valor de  $T_m$  y por tanto aumentar la velocidad de respuesta a variaciones de carga de tipo escalón.

Por tanto, para un voltaje de alimentación  $A=24V$ , con  $K_m=5$  y  $T_m=1$ , del sistema identificado en la sección 2.2, y una velocidad de referencia de 20rad/seg, se tiene que

$$t_{ss} = -\ln\left(1 - \frac{0.98 \cdot 20}{120}\right) \approx 0.178 \text{seg} \quad (54)$$

Si se desea un  $M_p$  de 1%, entonces

$$\xi = \sqrt{\frac{\ln(0.01)^2}{\ln(0.01)^2 + 3.1416^2}} \approx 0.83 \quad (55)$$

Con los resultados de las ecuaciones 54 y 55,  $\omega_n$  es

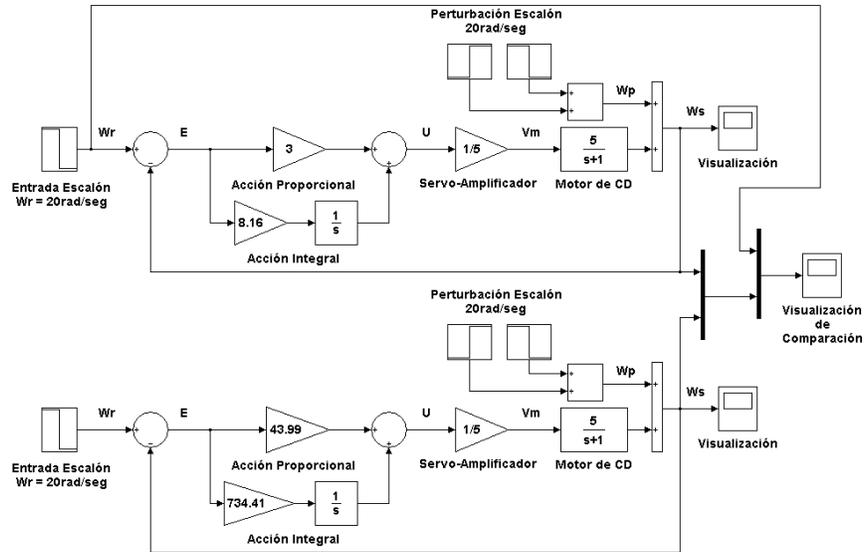
$$\omega_n = \frac{4}{0.178 \cdot 0.83} \approx 27.1 \quad (56)$$

Finalmente, utilizando las ecuaciones 52 y 53, se encuentran los valores de  $K_p$  y  $K_i$  tal que la respuesta a variaciones de carga de tipo escalón sea, como pretende el procedimiento, más rápida.

$$K_i = \omega_n^2 T_m = 27.1^2 \cdot 1 \approx 734.41 \quad (57)$$

$$K_p = 2\xi\omega_n T_m - 1 = 2 \cdot 0.83 \cdot 27.1 \cdot 1 - 1 \approx 43.99 \quad (58)$$

En la figura 2-13-a se encuentra el diagrama de bloques del sistema que ha sido objeto de simulaciones anteriores, en esta ocasión para ganancias óptimas del controlador y para las ganancias de las ecuaciones 57 y 58. En la figura 12-13-b se muestra la comparación en la respuesta y compensación de variación de carga para una entrada de referencia escalón de 20rad/seg y una perturbación también de 20rad/seg. Las gráficas en la figura 2-14, muestran la mejora en la velocidad de respuesta utilizando el procedimiento para la compensación de variaciones de carga descrito en esta sección.



(a)

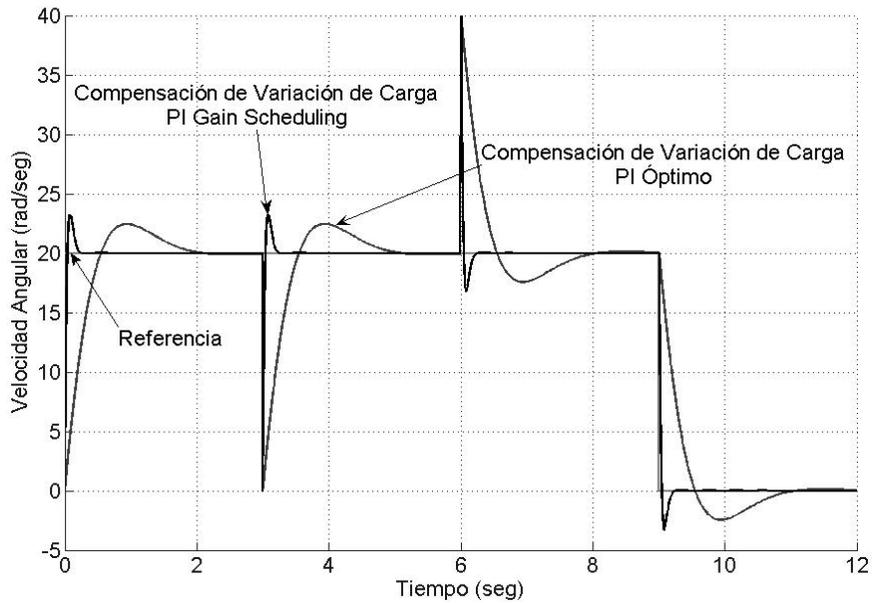


Figura 2-14.-Compensación de variación de carga. a) Diagrama de bloques, b) Comparación de compensación PI óptimo vs PI con gain scheduling

## Capítulo 3 Materiales y Métodos

Para validar experimentalmente los algoritmos planteados en el capítulo anterior, se montó un arreglo experimental. Este capítulo describe cada uno de los componentes del arreglo experimental, y está constituido principalmente por dos secciones. La primera describe los elementos considerados como hardware y la segunda el desarrollo de software para la adquisición y control.

### 3.1 Descripción General

Los sistemas de control moderno se consideran parte del campo de las señales y los sistemas, aunque también comprenden aspectos llamados los cinco elementos clave de la mecatrónica [Dorf 2005]. Tomando estos elementos como referencia, en la figura 3-1 se muestran las partes básicas de los sistemas de control.

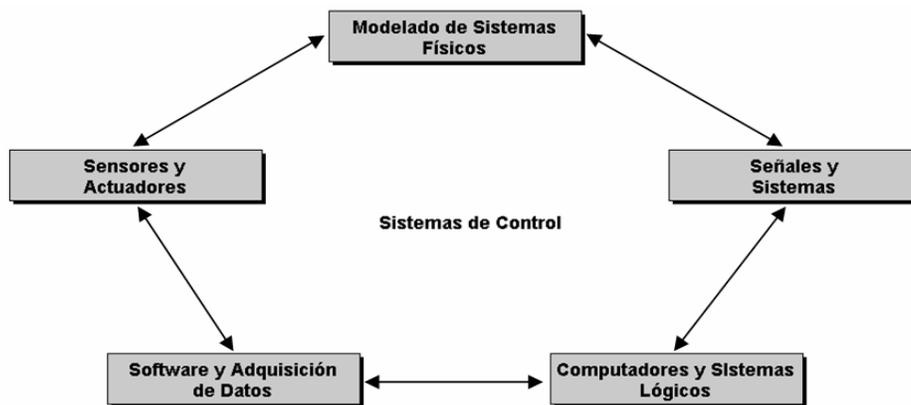


Figura 3-1.- Elementos de los sistemas de control

El sistema de control de velocidad para motores de corriente directa puede explicarse en términos del diagrama de la figura 3-1. La sección 2.1 del capítulo 2, se relaciona con el *modelado de sistemas físicos*, y las simulaciones referentes a la identificación y el control de las secciones 2.2 a 2.5, basadas en la teoría de control, están relacionadas con el bloque referente a *señales y sistemas*.

Para continuar con la explicación, en la figura 3-2 se muestra el arreglo experimental, seguido de su relación con el diagrama de la figura 3-1.

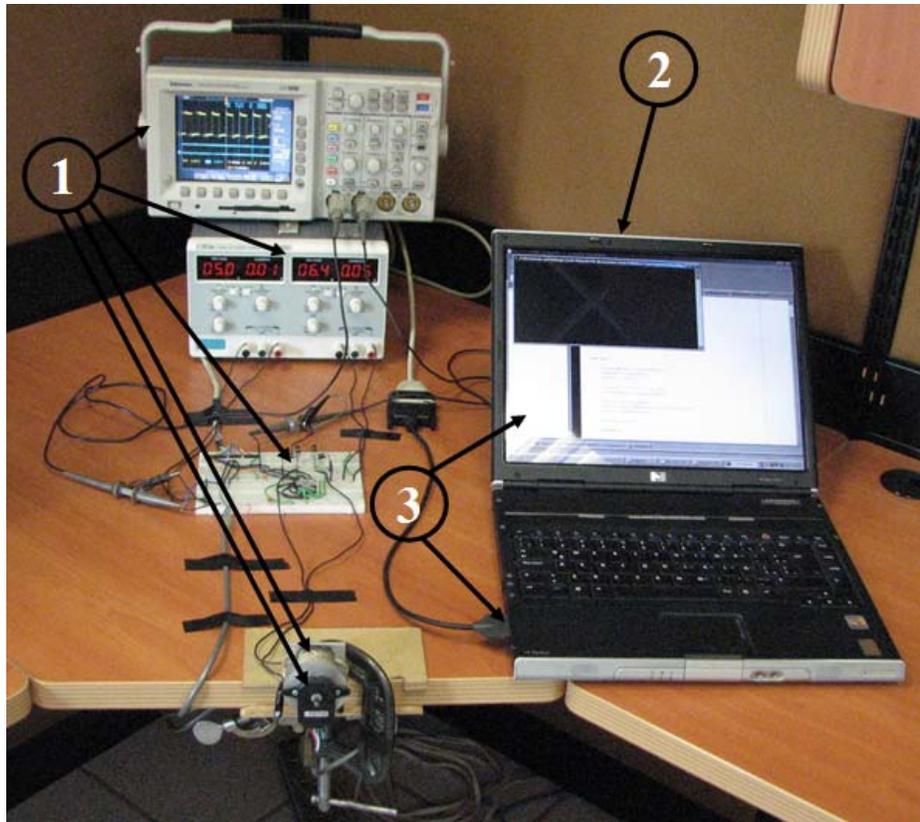


Figura 3-2.- Arreglo experimental para el control de velocidad de motores de CD

El punto 1 en la figura 3-2, indica los elementos que se consideran parte del bloque *sensores y actuadores*, el punto 2 se relaciona con el bloque *computadores y sistemas lógicos*, y el punto 3 con *software y adquisición de datos*. A continuación, una descripción breve de cada uno de ellos.

### ***Sensores y Actuadores***

#### Osciloscopio y Fuente de Alimentación

Osciloscopio digital Tektronix modelo TDS3034, con ancho de banda de 300MHz a 2.5GS/s [Tektronix 2008].

Fuente de alimentación dual de corriente directa, 0-30V (3A) en cada canal, con indicadores digitales marca UniSource modelo UNI PS-3030DU [ValueTesters 2008].

#### ***Encoder***

Encoder incremental en cuadratura, reconfigurable, marca CUI modelo AMT102-V [CUI 2008]. Información detallada acerca del funcionamiento y construcción de encoders incrementales puede encontrarse en [Electro-Craft 1980].

### ***Motor de CD***

Motor de corriente directa de imán permanente con escobillas, marca CANNON modelo FN30-U52N1E. Las principales características de los motores Canon de la serie FN30 pueden consultarse en [Canon 2008].

### ***Servo-Amplificador***

La etapa de potencia se basa en el circuito integrado L298N, el cual contiene un par de puentes “H”. El armado del circuito se realizó haciendo la conexión en paralelo de ambos puentes “H” y la incorporación de protecciones para cargas inductivas conforme a las especificaciones que brinda el fabricante en las hojas de datos del circuito [ST 2008]. Los principios de operación de un puente “H” pueden consultarse en [Mohan 2003].

### ***Computadores y Sistemas Lógicos***

Sustituyendo la labor de una tarjeta de control de movimiento, los algoritmos de control desarrollados pueden ser procesados en una PC. Para la realización de pruebas experimentales, los algoritmos se corrieron en una LAPTOP HP Pavilion modelo ze2000.

### ***Software y Adquisición de Datos***

Para la adquisición de datos y envío de señales de control se utilizó una tarjeta de puerto paralelo universal para PCMCIA de TRANS-DIGITAL CORPORATION modelo Trans PC Card. Las especificaciones de esta tarjeta pueden encontrarse en [TDC 2008]. La máxima velocidad de lectura del encoder es de 50KHz y la señal de control puede tener frecuencias moduladas en ancho de pulso desde 1Hz y hasta 2KHz. En la siguiente sección se describe el software para la adquisición y envío de señales.

## **3.2 Desarrollo de software**

Los algoritmos para la identificación y control, fueron programados en C++ siguiendo el paradigma de programación orientada a objetos, utilizando un compilador estándar GCC para Win32 de la colección de compiladores GNU. De acuerdo con [Ceballos 2004], se trata de un compilador C++ [MinGW 2008] de libre distribución que cumple con la norma ISO/IEC 14882, del cual existen versiones para prácticamente todos los sistemas operativos. Por lo tanto, el

código de los algoritmos está escrito en C++ puro, tal y como se define en el estándar C++, lo que garantizará que se ejecute en cualquier implementación que se ajuste a este estándar.

El software desarrollado está formado básicamente por ocho clases que definen ocho tipos de datos y sus métodos asociados. El apéndice 2 presenta el código completo de cada una de las clases. El entendimiento del código puede realizarse consultando [Kernighan 1998], [Stroustrup 2000] y [Petzold 1999] para conceptos relacionados con C, C++ y programación en Windows respectivamente. Referencias rápidas a conceptos y funciones se encuentran en [Schildt 2003] y [MSDN 2008].

Las clases desarrolladas, acompañadas de una breve descripción de su funcionalidad se listan a continuación.

### ***1. Clase CPuertos***

Permite la comunicación con el puerto paralelo estándar (SPP) bajo el sistema operativo Windows XP, utilizando la librería de enlace dinámico (DLL) desarrollada por Logix4u [Logix4u 2008], para lograr la adquisición y envío de señales de control mediante operaciones de lectura y escritura en los registros de estado y datos del SPP respectivamente. Información detallada acerca del SPP puede consultarse en [Axelson 2000].

### ***2. Clase CMicroTiempo***

Las funciones estándar en C++ para la medición del tiempo, permiten la medición de este, en el orden de milisegundos, pero para poder realizar mejores mediciones en la velocidad de giro del motor de CD, teniendo como elemento de retroalimentación un encoder incremental en cuadratura, es necesaria la medición del tiempo en microsegundos. Para lograr esto, se incluyó en los métodos de CMicroTiempo un par de funciones que brinda la librería Windows.h, las cuales permiten tomar un múltiplo de la frecuencia del reloj de la PC, que se encuentra en el orden de los microsegundos.

### ***3. Clase CGuardaDatos***

Mediciones realizadas experimentalmente demostraron que el tiempo aproximado para el almacenamiento en disco de un solo dato de tipo punto flotante, toma al sistema de cómputo alrededor de  $12\mu\text{seg}$  para la computadora utilizada<sup>2</sup>. Puesto que la velocidad de adquisición es un punto crítico en el sistema de control, este tiempo se torna excesivo. Para evitarlo, las funciones de CGuardaDatos almacenan dinámicamente los datos en un arreglo durante la ejecución del programa y cuando este termina de ejecutarse, los datos son recuperados y almacenados en el disco duro. Utilizando este enfoque se reduce el tiempo de almacenamiento a aproximadamente  $0.5\mu\text{seg}$  por dato, mejorando sustancialmente el control desarrollado.

#### **4. Clase CEncoder**

Las señales de los canales A y B del encoder incremental en cuadratura, se encuentran desfasadas  $90^\circ$  eléctricos una de la otra, lo que permite conocer la dirección de movimiento del eje del motor de CD e incrementa en cuatro tantos la resolución de la señal de posición que este genera. Los métodos de CEncoder realizan la inicialización de la medición e implementan el algoritmo para calcular la posición y determinar la dirección de movimiento del motor.

#### **5. Clase CPwm**

La manipulación del voltaje promedio aplicado al motor de CD para su control, se lleva a cabo mediante la Modulación del Ancho de Pulso de un voltaje fijo conmutando a una frecuencia fija, o PWM por sus siglas en inglés.

Esta técnica es ampliamente utilizada en servo-amplificadores industriales para el control de velocidad de motores de corriente directa, debido a que disipan menos energía en comparación con amplificadores lineales además de requerir solo una fuente de alimentación para lograr movimientos en ambas direcciones cuando se utiliza en conjunto con circuitos que implementan un puente “H”.

Para generar las señales de conmutación a una frecuencia fija y un ancho de pulso variable de acuerdo al voltaje promedio deseado, las funciones de CPwm utilizan los

---

<sup>2</sup> El tiempo de almacenamiento puede variar dependiendo del sistema de cómputo utilizado.

métodos de *CMicroTiempo* y *CPuertos* para la medición del tiempo y el envío de las señales de control a la entrada del servo-amplificador respectivamente, implementando así el algoritmo para la modulación del ancho de pulso.

#### **6. Clase *CCalentamiento***

La única función de esta clase, envía señales al servo-amplificador con la forma de onda de la señal de excitación de la figura 2-5, aplicando el máximo voltaje durante tiempos definidos según se desee, para provocar un precalentamiento en los motores de CD antes de comenzar con el proceso de identificación y/o control.

#### **7. Clase *CRespuesta***

Sus métodos llevan a cabo la ejecución de los procesos de identificación descritos en las secciones 2.2 y 2.4.2 del capítulo 2, haciendo uso de las funciones de todas las clases antes descritas, excepto la número 6.

#### **8. Clase *CControl***

Sus métodos realizan la ejecución de los algoritmos de control desarrollados en las secciones 2.3, 2.4 y 2.5 del capítulo 2, utilizando las funciones de las clases 1 a 5 y los resultados de la clase 7.

## Capítulo 4 Resultados Experimentales

Los resultados obtenidos al experimentar en el banco de pruebas, con los algoritmos y procedimientos vistos en secciones anteriores, se presentan en este capítulo. Se exponen también las consideraciones que deben tomarse al implementar los algoritmos de control en una computadora digital, se discuten los fenómenos no modelados encontrados y su efecto en la dinámica del sistema de control desarrollado.

### 4.1 Identificación del Sistema

Para mejorar la repetibilidad en mecanismos actuados por motores de corriente directa [Armstrong 1991], antes de realizar cualquier experimento se corre una rutina de precalentamiento. Esta rutina consiste en aplicar una señal de excitación con la forma de onda de la figura 2-5 durante un tiempo aproximado de 3 minutos, provocando así los movimientos más exhaustivos a los que estará sujeto el motor de CD durante la experimentación. Observaciones en el comportamiento con y sin calentamiento del motor utilizado indicaron que después de este tiempo, el sistema no tenía grandes variaciones en su velocidad máxima. En la gráfica de la figura 4-1 se observa la mejora en la repetibilidad de las mediciones de velocidad angular cuando estas se toman después del precalentamiento, además de su diferencia en magnitud.

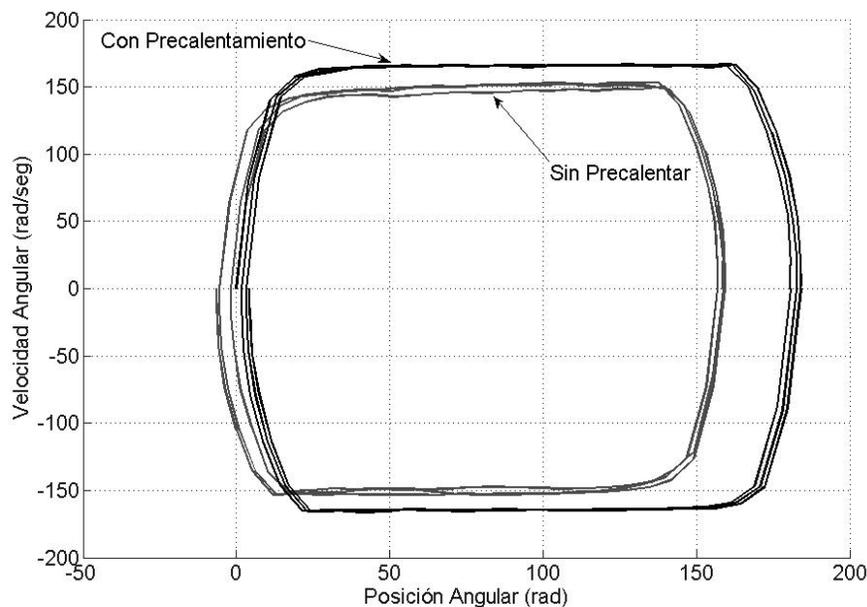


Figura 4-1.- Efecto del precalentamiento

Antes de realizar la identificación experimental, las operaciones de escaneo y multitarea, realizadas por el sistema operativo Windows XP deben ser eliminadas o al menos minimizadas. Para detectar automáticamente la conexión de dispositivos en los puertos del computador, el sistema operativo realiza constantemente ciclos de escaneo conocidos como polling. Esta tarea afecta la adquisición de datos realizada a través del SPP, por tanto debe ser desactivada. El proceso de desactivación puede encontrarse en [Axelson 2008] en la sección de preguntas frecuentes. Windows XP es un sistema operativo multitarea, lo que permite la ejecución simultánea de varios programas. La mayoría de veces esta característica resulta muy útil, no así para el sistema de control desarrollado. Por tanto, el computador debe ser reiniciado en modo básico y sus propiedades deben ser configuradas para lograr el mejor rendimiento, además de evitar la ejecución de programas al momento de correr los experimentos. Los procedimientos para realizar estas mejoras pueden encontrarse en [Startups 2008] y [SpeedUp 2008]. En la figura 4-2-a se muestran los efectos causados por el ciclo de escaneo y la multitarea, en 4-2-b se desactiva el polling y en 4-2-c se minimiza además la multitarea, demostrando la mejora que causa tomar en cuenta las consideraciones anteriores al momento de ejecutar algún experimento.

Con las consideraciones anteriores, el proceso para la identificación experimental descrito en el capítulo 2, se realizó bajo las condiciones descritas en la tabla 4-1 y la respuesta del sistema del banco experimental descrito en el capítulo 3, se muestra en la figura 4-3:

Amplitud de los escalones	$A = 5V$
Tiempo de aplicación de los escalones	$T = 1\text{seg}$
Tiempo de muestreo de la medición	$TM = 0.01\text{seg}$
Duración de la prueba	$DP = 12\text{seg}$

Tabla 4-1.- Condiciones del proceso de identificación

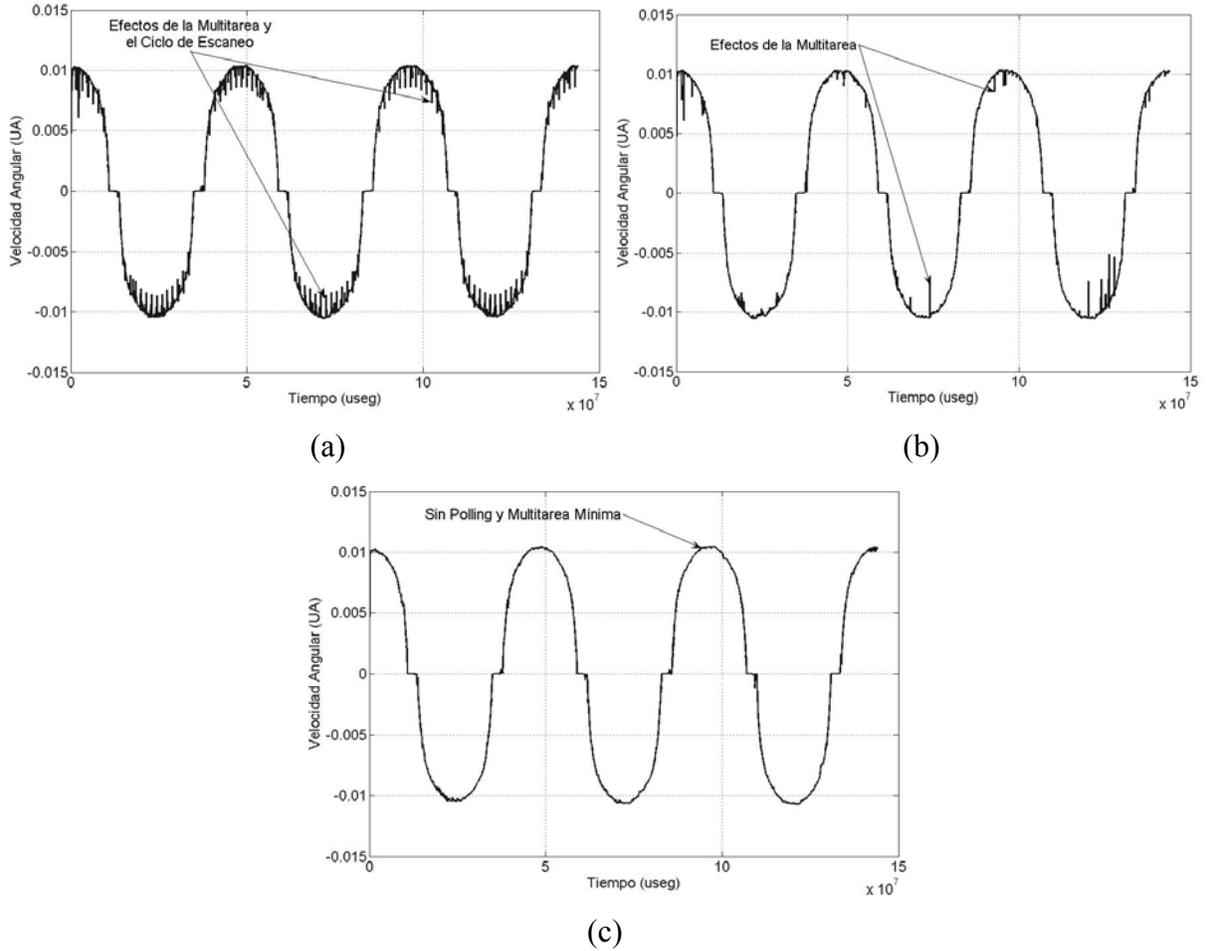


Figura 4-2.- Efectos del ciclo de escaneo y multitarea, a) Ambos efectos combinados, b) Ciclo de escaneo desactivado y c) Ambos efectos cancelados

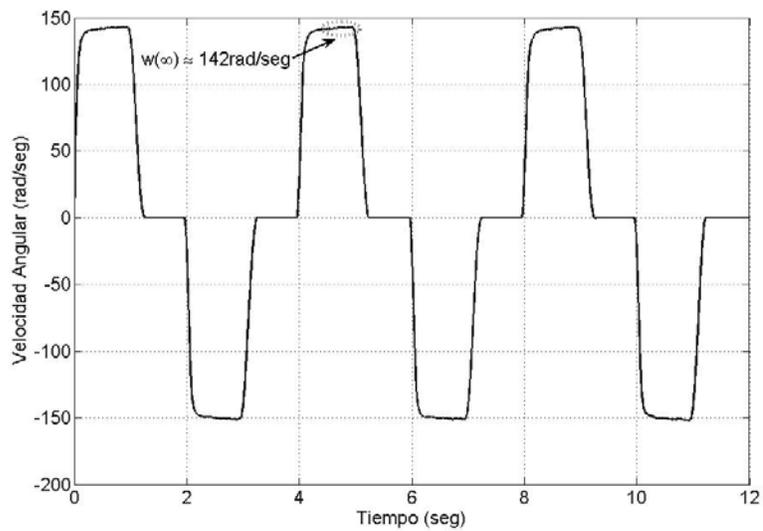


Figura 4-3.- Respuesta del sistema para el proceso de identificación

En la figura se observa el valor que toma la velocidad angular cuando la respuesta del sistema alcanza el estado estacionario. Con los datos de la tabla 4-1, el valor de  $\omega(\infty)$  de la figura 4-3 y conforme a la ecuación 23 del capítulo 2, el valor de  $K_m$  para el sistema bajo identificación es

$$K_m = \frac{\omega(\infty)}{A} = \frac{142 \frac{rad}{seg}}{5V} = 28.4 \frac{rad}{seg \cdot V} \quad (1)$$

Un acercamiento a la figura 4-3, para la determinación de la constante de tiempo se muestra en la figura 4-4, donde se observa que  $T_m \approx 0.049seg$ .

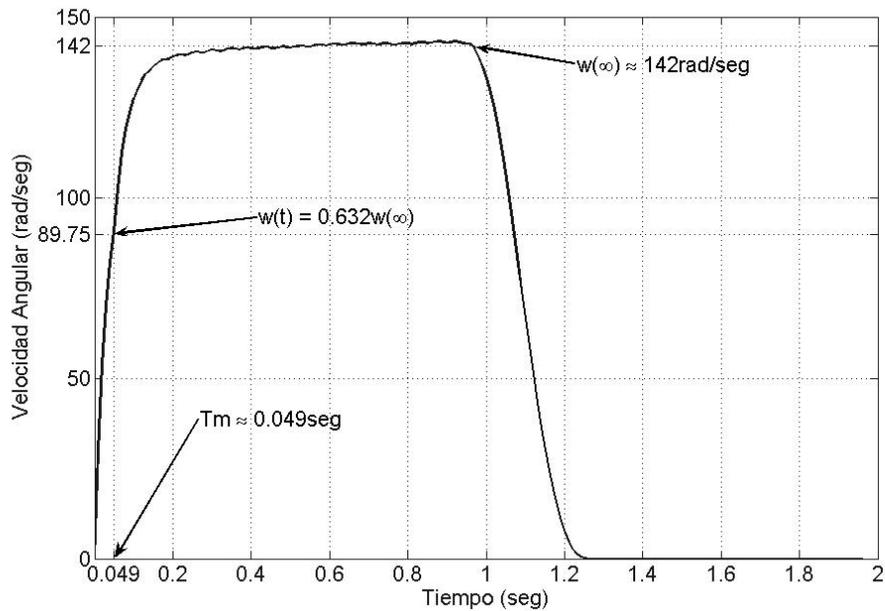


Figura 4-4.- Determinación gráfica experimental de  $T_m$

Conocidos  $K_m$  y  $T_m$ , la FT que describe el comportamiento del sistema es

$$\frac{\Omega(s)}{E(s)} = \frac{K_m}{T_m s + 1} = \frac{28.4}{0.049s + 1} \quad (2)$$

y de acuerdo al modelo, la respuesta temporal para una entrada escalón de +5V, esta dada por

$$w(t) = 142(1 - e^{-\frac{t}{0.049}}) \quad (3)$$

La comparación entre el modelo propuesto (claro) y la respuesta real del sistema (oscuro), se muestra en la figura 4-5.

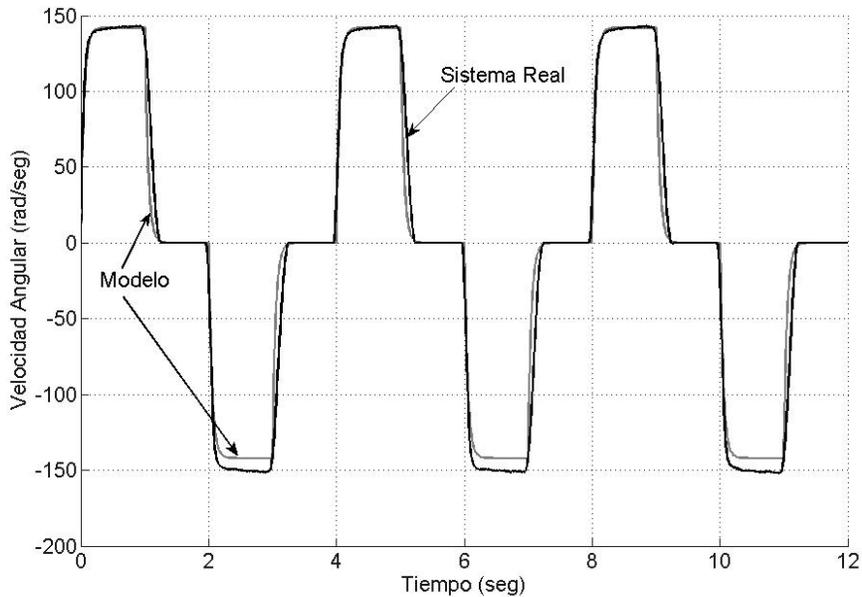


Figura 4-5.- Modelo vs. Respuesta real

Aunque para el cálculo de los parámetros del controlador, se utilizará la función de transferencia descrita en la ecuación 2, la gráfica de la figura 4-5 muestra que existe una ligera desviación en el modelo con respecto a la respuesta real del sistema, además de presentar un comportamiento asimétrico a entradas de excitación de la misma magnitud pero de signo contrario, lo que sugeriría el uso de constantes  $K_m$  y  $T_m$  diferentes dependiendo de la dirección de movimiento.

## 4.2 Desempeño del Controlador

Siguiendo la metodología descrita en la sección 2.3.2 del capítulo 2, para la sintonización del controlador con ganancias óptimas, se propone para el sistema identificado en la sección anterior, un tiempo de establecimiento de 0.098seg. Por tanto, de la ecuación 40 del capítulo 2, se tiene

$$\omega_n = \frac{4}{t_{ss}\xi} = \frac{4}{0.098 \cdot 0.7} \approx 58.31 \quad (4)$$

De las ecuaciones 41 y 42 del capítulo 2, las ganancias del controlador son

$$K_i = T_m \omega_n^2 = 0.049 \cdot 58.31^2 \approx 166.6 \quad (5)$$

$$K_p = 1.4 T_m \omega_n - 1 = 1.4 \cdot 0.049 \cdot 58.31 - 1 \approx 3 \quad (6)$$

El diagrama de bloques del sistema identificado y su controlador PI se muestra en la figura 4-6.

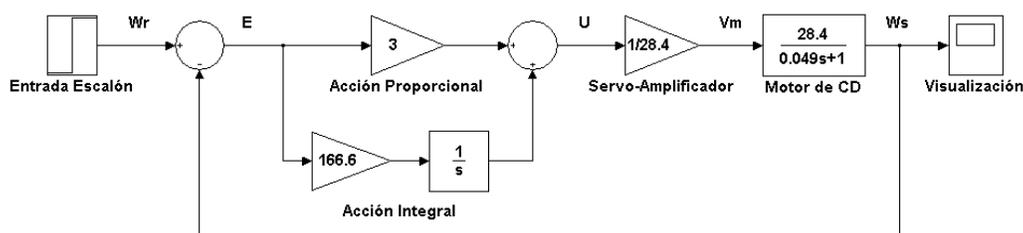


Figura 4-6.- Diagrama de bloques del sistema de control

De acuerdo a la ecuación 33 del capítulo 2, la FT del diagrama de bloques de la figura 4-6, es

$$\frac{W_s(s)}{W_r(s)} = \frac{3400}{s^2 + 81.63s + 3400} \quad (7)$$

La FT de la ecuación 7 no toma en cuenta los coeficientes del numerador que se encuentren multiplicando a  $s$ . Esto se realizó para eliminar la presencia de ceros en la FT y facilitar el cálculo de las ganancias del controlador. Sin embargo, la respuesta transitoria del sistema puede verse afectada dependiendo de la posición en el plano  $s$  del cero no considerado. La FT de la ecuación 32 del capítulo 2, la cual sí considera el cero y es la representación completa de la FT del diagrama de bloques de la figura 4-6, cuando no existe perturbación alguna en el sistema, puede reescribirse de la siguiente forma

$$\frac{W_s(s)}{W_r(s)} = \frac{\frac{\omega_n^2}{\alpha}(s + \alpha)}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (8)$$

donde

$$2\xi\omega_n = \frac{K_p + 1}{T_m} \quad (9)$$

$$\omega_n^2 = \frac{K_i}{T_m} \quad (10)$$

$$\alpha = \frac{K_i}{K_p} \quad (11)$$

Para el sistema identificado y ganancias óptimas para su controlador PI conforme a las ecuaciones 5 y 6, la FT completa, de acuerdo a la ecuación 8 es entonces

$$\frac{W_s(s)}{W_r(s)} = \frac{61.23(s + 55.53)}{s^2 + 81.63s + 3400} \quad (12)$$

El porcentaje de sobre-impulso como función de  $\alpha/\xi\omega_n$ , cuando  $\xi \leq 1$ , para una FT con la estructura de la ecuación 8, puede encontrarse en [Dorf 2005]. Para el sistema representado por la ecuación 12, la relación  $\alpha/\xi\omega_n$  tiene un valor aproximado de 1.36, lo que implica que el sistema tendrá un sobre-impulso aproximado de 12%, debido al efecto del cero en la FT, y no de 5% como se esperaba según la FT de la ecuación 7. En la figura 4-7 se muestra la simulación de la comparación de las respuestas normalizadas para el sistema identificado con y sin la consideración del cero en su FT, ante entradas de referencia de tipo escalón con amplitudes de 20, 60 y 120 rad/seg.

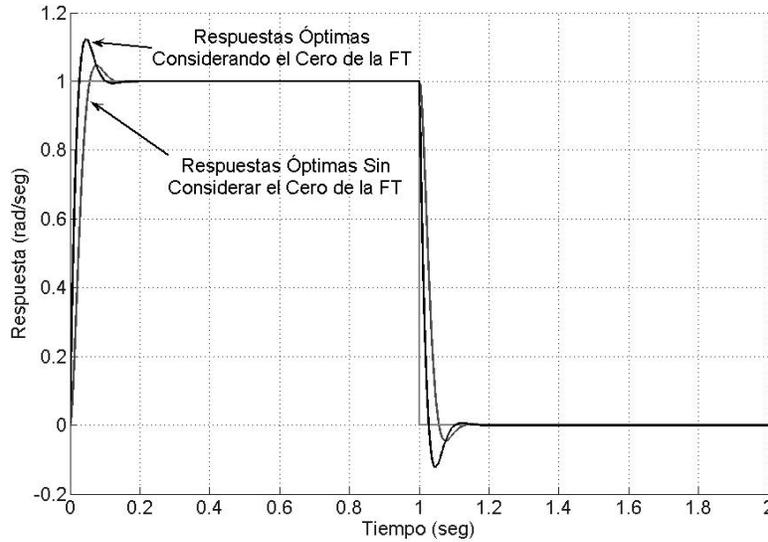


Figura 4-7.- Respuestas óptimas normalizadas, con y sin considerar el cero de la FT

De acuerdo con la figura 4-7, se espera que los resultados experimentales del desempeño del controlador concuerden con la respuesta que considera el cero en la FT del sistema, por tanto el sobre-impulso deberá ser aproximadamente del 12%.

Puesto que la implementación del controlador se realiza en una computadora digital, el algoritmo de control debe ser digitalizado. Utilizando la regla rectangular hacia atrás del método de Euler [Franklin 1998], la ecuación en diferencias del controlador PI es

$$u(k) = u(k-1) + (K_p + \Delta K_i) \cdot e(k) - K_p \cdot e(k-1) \quad (13)$$

donde  $u(k)$  y  $e(k)$  son las señales actuales de control y de error respectivamente,  $u(k-1)$  y  $e(k-1)$  las señales de control y de error un tiempo de muestreo atrás, y  $\Delta$  el tiempo de muestreo.

El impacto más importante en la implementación digital del controlador se debe al retraso asociado con el retenedor, en este caso de orden cero. El retraso en cualquier sistema retroalimentado, degrada la estabilidad y el amortiguamiento del mismo. Para garantizar la estabilidad del sistema de control, de acuerdo al teorema de muestreo, la frecuencia de este debe ser al menos dos veces la frecuencia correspondiente al ancho de banda del sistema en lazo cerrado. Sin embargo, para una respuesta suave, el rango típico de esta frecuencia se escoge entre 20 y 40 veces la frecuencia del ancho de banda del sistema.

La selección del tiempo de muestreo  $\Delta$ , determina la frecuencia de la señal de control, en este caso realizada mediante la modulación de ancho de pulso (PWM). Para selección apropiada de la frecuencia del PWM, se realiza un análisis de estabilidad de la función de transferencia en el dominio  $Z$ . El método más conocido para la determinación de la estabilidad para sistemas en tiempo discreto, y el utilizado en este trabajo, se basa en asegurar que los polos de la FT del sistema se mantengan dentro del círculo unitario en el plano  $Z$  [Franklin 1998].

Realizando la transformada  $Z$  de la FT de la trayectoria directa del diagrama de bloques de la figura 4-6, considerando en esta la dinámica de un retenedor de orden cero, la FT en lazo abierto en el dominio  $Z$  puede escribirse como

$$G(z) = \frac{m \cdot (z - n)}{(z - 1)(z - b)} \quad (14)$$

donde el polinomio característico de la FT en lazo cerrado es

$$P(z) = 1 + G(z) \quad (15)$$

e igualando a cero

$$P(z) = 1 + G(z) = z^2 + (m - b - 1) \cdot z + (b - m \cdot n) = 0 \quad (16)$$

las raíces del polinomio son

$$z_{1,2} = \frac{(b - m + 1) \pm \sqrt{(m - b - 1)^2 - 4 \cdot (b - m \cdot n)}}{2} \quad (17)$$

garantizando la estabilidad para

$$-1 \leq z_{1,2} \leq 1 \quad (18)$$

donde

$$m = a + c \quad (19)$$

$$n = \frac{a \cdot b + c}{a + c} \quad (20)$$

$$a = \Delta \cdot K_i \quad (21)$$

$$b = e^{-\frac{\Delta}{T_m}} \quad (22)$$

$$c = \left( K_p - K_i \cdot T_m \right) \left( 1 - e^{-\frac{\Delta}{T_m}} \right) \quad (23)$$

El único parámetro desconocido en las ecuaciones anteriores es el tiempo de muestreo  $\Delta$ . Por tanto, evaluando numéricamente la ecuación 17 es posible determinar para qué valor de  $\Delta$  el sistema de control será inestable. En la figura 4-8 se muestran las gráficas de los polos de la FT del sistema identificado, para valores de  $\Delta$  desde 0 hasta 0.2 segundos.

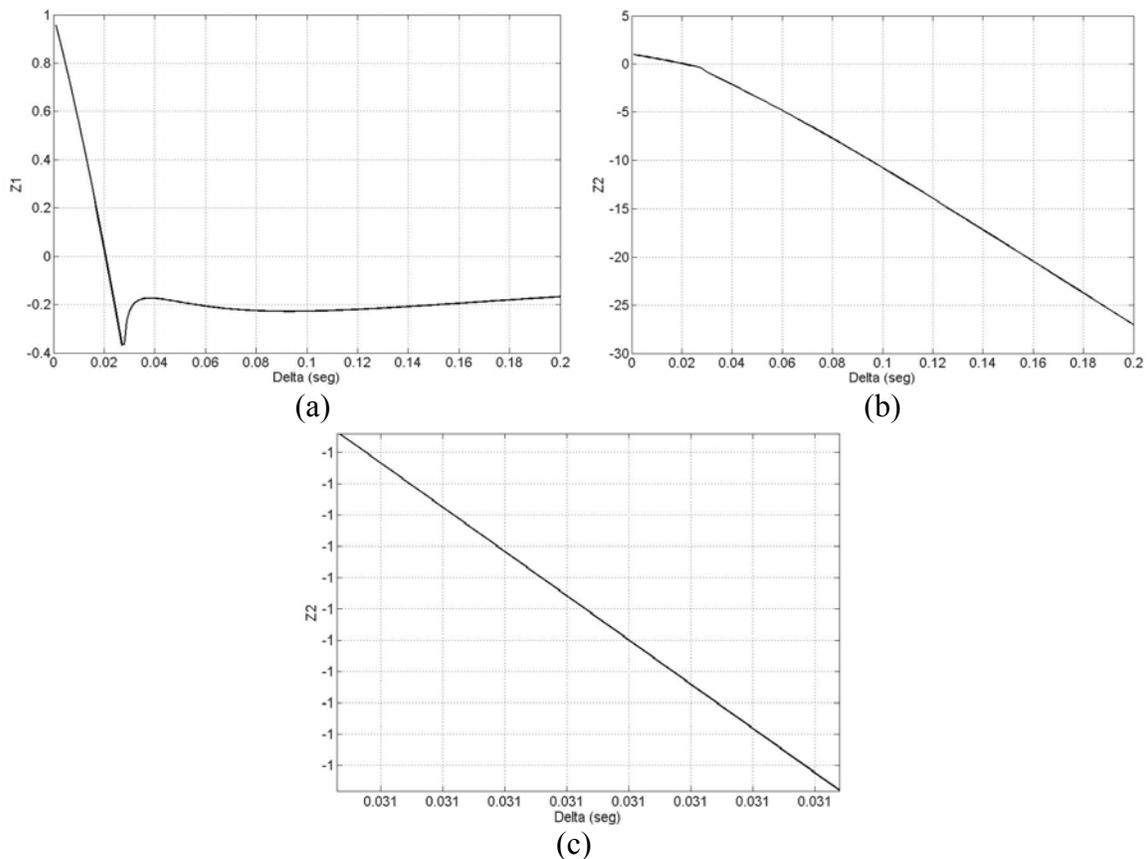


Figura 4-8.- Polos en el plano Z de la FT de la ecuación 12 para diferentes valores de  $\Delta$ . a) Polo  $z_1$ , b) Polo  $z_2$ , c) Límite de  $\Delta$  para la estabilidad

En la figura 4-8-a se observa que para cualquier valor que tome el tiempo de muestreo en el rango antes especificado, el polo  $z_1$  se mantiene dentro del círculo unitario. La gráfica 4-8-b muestra que el polo  $z_2$  sale de la región de estabilidad para la mayoría de los valores que toma  $\Delta$  y en la figura 4-b-c se hace un acercamiento a la gráfica 4-8-b, determinando así el límite de estabilidad en función del tiempo de muestreo  $\Delta$ . Por tanto, para el sistema de la FT de la ecuación 12 la frecuencia del PWM para garantizar la estabilidad deberá ser

$$f_{PWM} \geq \frac{1}{0.031seg} \approx 32.26Hz \quad (24)$$

Tomando en cuenta las consideraciones anteriores, el experimento para la evaluación del desempeño del controlador se llevó a cabo bajo las siguientes condiciones.

Señal escalón de referencia 1	$\Omega_r = 120 \text{ rad/seg}$
Señal escalón de referencia 2	$\Omega_r = 0 \text{ rad/seg}$
Señal escalón de referencia 3	$\Omega_r = -120 \text{ rad/seg}$
Señal escalón de referencia 4	$\Omega_r = 0 \text{ rad/seg}$
Tiempo de aplicación de los escalones de referencia	$T = 2\text{seg}$
Frecuencia de la señal de control (PWM)	$f_{PWM} = 100 \text{ Hz}$
Duración de la prueba	$DP = 8 \text{ seg}$

Tabla 4-2.- Condiciones para evaluar el desempeño del controlador

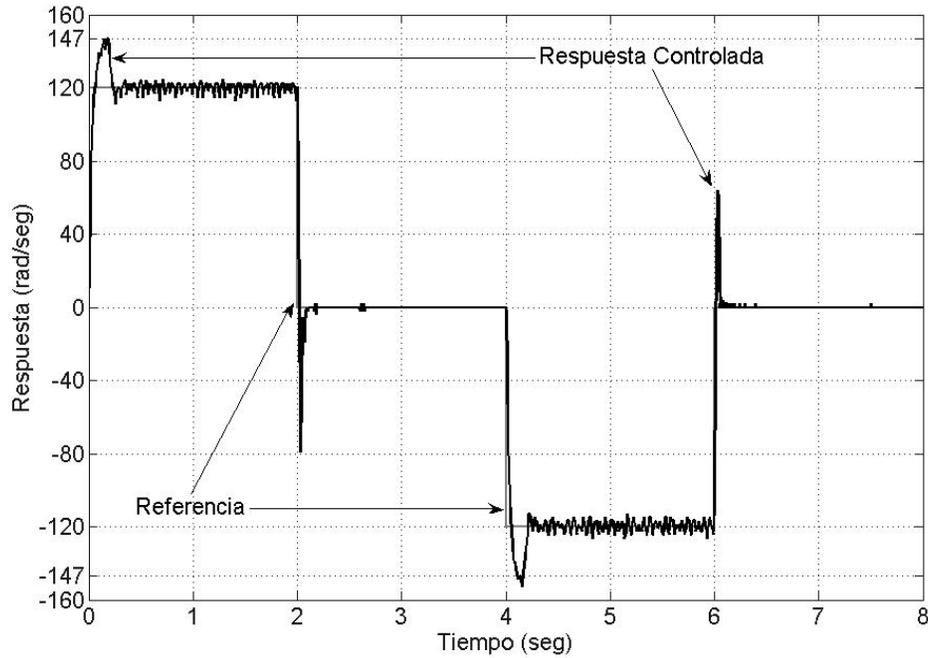


Figura 4-9.- Desempeño del Controlador

La figura 4-9 muestra el resultado experimental de la implementación del controlador. A simple vista, comparando las figuras 4-7 y 4-9, se observan desviaciones con respecto a lo esperado según el modelo y simulación del controlador. En la figura 4-10 se muestra esta comparación.

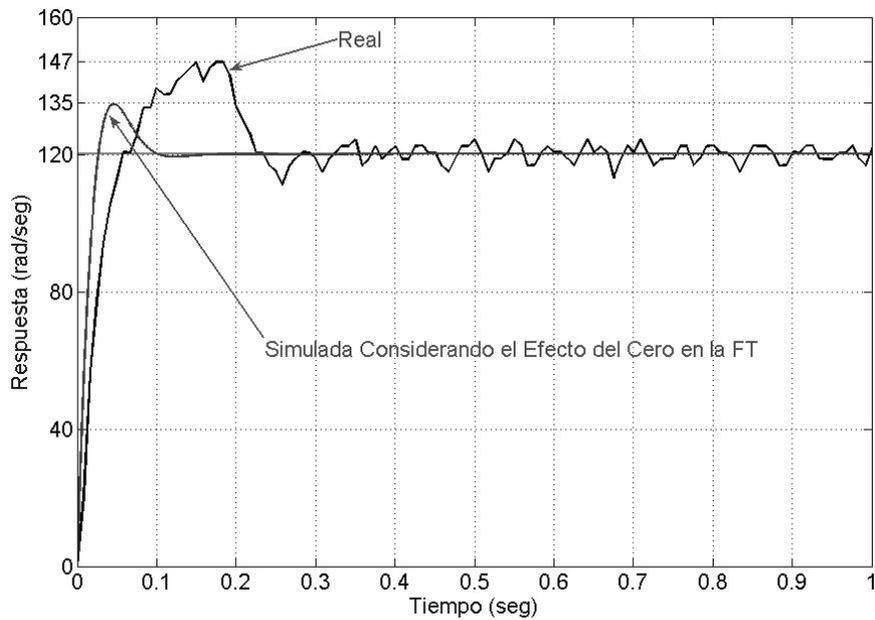


Figura 4-10.- Comparación entre desempeños simulado y real

En la figura 4-10 se observan tres desviaciones importantes:

1. Mayor sobre-impulso
2. Mayor tiempo de establecimiento
3. Fluctuaciones periódicas en estado estacionario

El mayor sobre-impulso y tiempo de establecimiento se deben principalmente a la no consideración de la saturación del servo-amplificador en el modelo del diagrama de bloques de la figura 4-6. También tienen influencia, aunque con una contribución menor, la discretización del controlador, la zona muerta y la relación no lineal de la velocidad angular con respecto al voltaje aplicado. El diagrama de bloques del sistema, incluyendo saturación, discretización, zona muerta y la relación no lineal de la velocidad angular y el voltaje aplicado se muestra en la figura 4-11.

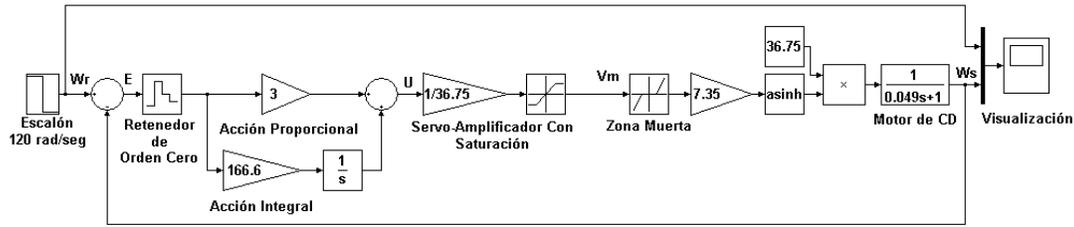


Figura 4-11.- Diagrama de bloques incluyendo fenómenos no modelados

Para poder comparar los resultados de la simulación del diagrama de bloques de la figura 4-11 con los resultados obtenidos experimentalmente presentados en la figura 4-10, y justificar así las desviaciones obtenidas, las condiciones para la simulación son las siguientes:

Tiempo de muestreo del retenedor de orden cero	$TM = \frac{1}{f_{pwm} = 120Hz} = 0.008333seg$
Voltaje de saturación del servo-amplificador	$VS^+ = +5V$ $VS^- = -5V$
Voltaje de zona muerta	$ZM^+ = +1V$
Función velocidad angular vs. voltaje aplicado	$\omega(v) = 36.75 \cdot \sinh^{-1}(7.35v)$ $ZM^+ \leq v \leq VS^+$

Tabla 4-3.- Condiciones para la simulación de fenómenos no modelados

Los datos del voltaje correspondiente a la zona muerta positiva y la función velocidad angular vs. voltaje aplicado fueron obtenidos experimentalmente al aplicar en lazo abierto un voltaje rampa al motor de corriente directa. Los resultados de este experimento se muestran en la figura 4-12.

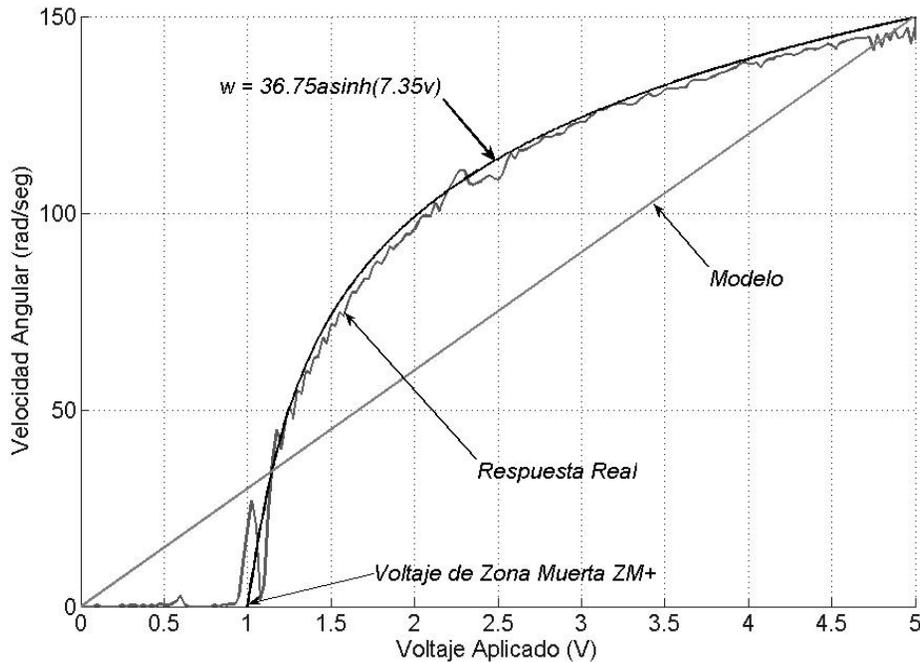


Figura 4-12.- Zona muerta experimental y función velocidad vs. voltaje

El resultado de la simulación del diagrama de bloques de la figura 4-11 y la comparación con la simulación del sistema de la figura 4-6 y los resultados obtenidos experimentalmente se muestra en la figura 4-13. Puede observarse que la respuesta real del sistema es muy parecida a lo esperado en la simulación cuando se consideran los aspectos mostrados en la figura 4-11, justificando las desviaciones en sobre-impulso y tiempo de establecimiento.

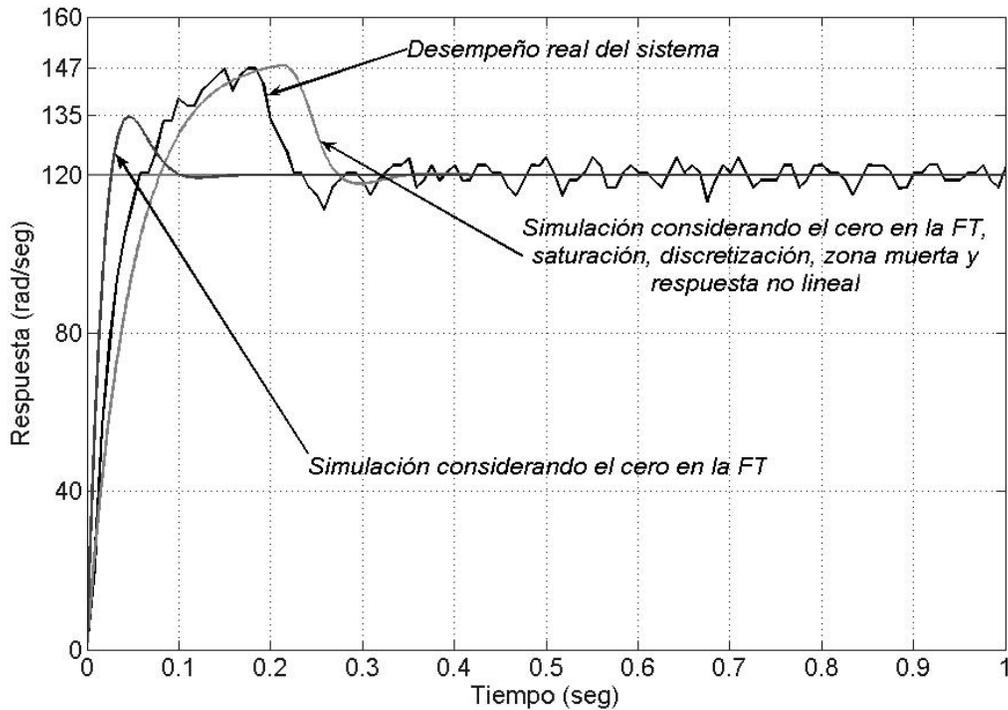


Figura 4-13.- Desempeño real y simulación considerando fenómenos no modelados

Las fluctuaciones en estado estacionario que se observan en el desempeño real, tanto en la figura 4-11 como en la 4-13, se deben a un fenómeno conocido como “torque ripple” [Electro-Craft 1980]. Estas son causadas principalmente por la conmutación en el conmutador del motor.

La misma estructura hardware-software desarrollada para el banco experimental de pruebas se utilizó con éxito en el control de velocidad del tribómetro de perno en disco. Aún cuando los algoritmos de control utilizados para las pruebas realizadas en el tribómetro no contaban con la compensación de zona muerta y variación de carga, el desempeño del controlador mantuvo la velocidad angular del disco dentro de  $\pm 1\%$  de la señal de referencia como marca la norma ASTM G99-05, incluso con perturbaciones debidas a fenómenos de desgaste. Además, las gráficas de voltaje que reflejan el esfuerzo del controlador por mantener la velocidad de referencia muestran una correlación al menos cualitativa con la fuerza de fricción debida al desgaste entre el disco y el perno (ver Figura 1-2). Para mayores detalles consultar, [J. Pichardo 2007].

### 4.3 Zona Muerta

Para simular el desplazamiento de una carga, se acopló un disco de hierro al eje del motor. Este se muestra en la figura 4-14.



Figura 4-14.- Sistema con disco de carga

Al agregar la carga al motor su dinámica se afecta y es necesario identificar nuevamente sus parámetros y volver a sintonizar el controlador. Los resultados obtenidos mediante la identificación experimental y las ganancias óptimas del controlador para este nuevo sistema se presentan en la tabla 4-4.

$K_m = 28.4$	$T_m = 0.63$	$K_p = 3$	$K_i = 13$
--------------	--------------	-----------	------------

Tabla 4-4.- Parámetros para el sistema con disco de carga

La sintonización del controlador se realizó considerando un tiempo de establecimiento de 1.26 seg. El desempeño de este sistema para una referencia de 120 rad/seg se muestra en la figura 4-15. En esta se observa además, la atenuación en las fluctuaciones en estado estacionario, esto debido al incremento en la inercia al acoplar el disco al eje del motor.

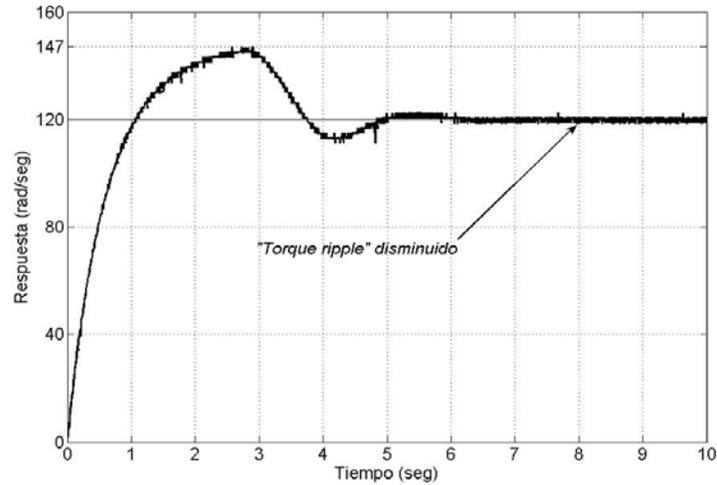


Figura 4-15.- Desempeño del sistema

Para mostrar el efecto que tiene la zona muerta en el desempeño del controlador, siguiendo con el esquema de simulación mostrado en la sección 2.4 del capítulo 2, se hace seguir al sistema una señal de referencia de  $40\sin(0.25t)\text{rad/seg}$ . En la figura 4-16 se presenta el resultado de tal experimento.

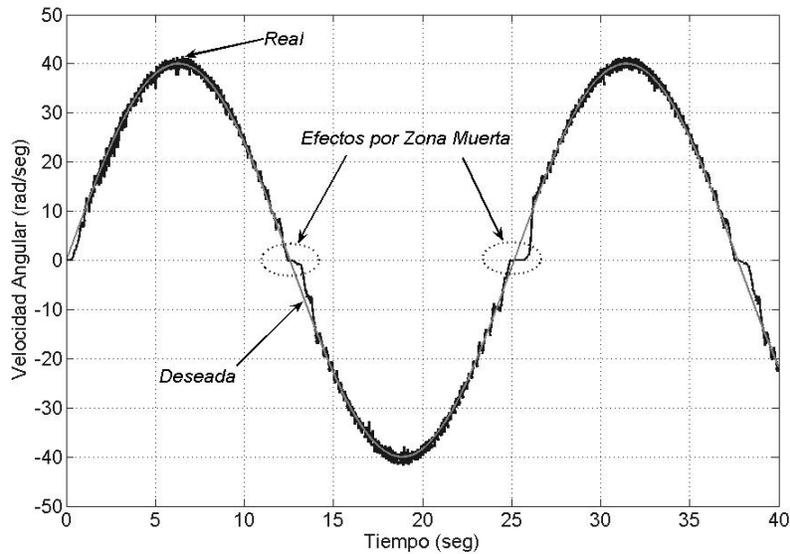


Figura 4-16.- Efecto de la zona muerta

Además de los efectos debidos a la zona muerta, en la figura 4-16 se notan oscilaciones alrededor de la señal de referencia, estas se deben a fenómenos de “torque ripple” y al desbalance del disco. En la figura 4-12 se encontró experimentalmente el valor del voltaje de

zona muerta, utilizando este valor y siguiendo el procedimiento para su compensación descrito en la sección 2.4 del capítulo 2, el resultado de la compensación se muestra en la figura 4-17.

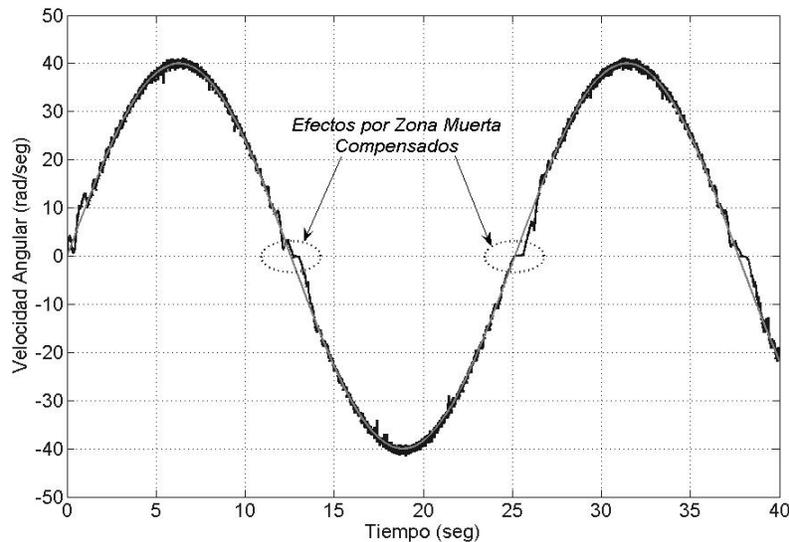


Figura 4-17.- Compensación de la zona muerta

Comparando los resultados que presentan las figuras 4-16 y 4-17, se observa que el algoritmo de control propuesto logra compensar los efectos que produce la zona muerta en el motor de corriente directa, aunque no puede eliminarlos completamente. Esto se debe a diversos factores. El algoritmo para la compensación consideró simétrico el voltaje de zona muerta, mientras que los resultados experimentales que aún no se han comprendido a satisfacción, parecen indicar que el valor de este no es constante y depende entre otras cosas de la frecuencia del PWM, de la aceleración y de la carga referida al eje del motor.

#### 4.4 Variación de Carga

Ya que para la validación del algoritmo de control descrito en la sección 2.5 del capítulo 2, no es necesario conocer la magnitud en par de la perturbación sino sus efectos en la variable a controlar y que esta tenga aproximadamente la misma amplitud en todos los experimentos para poder comparar el desempeño de los controladores con y sin compensación, se realizó la modificación que se muestra en la figura 4-18, agregando una base magnética al sistema con el que se trabajó en la sección anterior. Al activar la base magnética, ésta ejerce una fuerza de atracción que tiende a detener el movimiento en el disco del eje del motor, provocando de esta manera una variación de carga constante.



Figura 4-18.- Sistema utilizado para inducir variaciones de carga constantes

La velocidad angular en lazo abierto, al activar la base magnética, desciende de un valor aproximado de 150 rad/seg a 80 rad/seg, como se muestra en la figura 4.19.

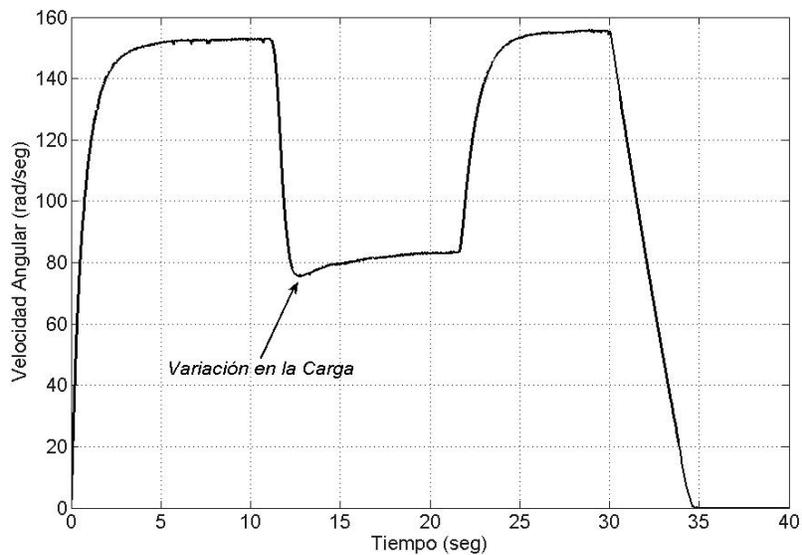


Figura 4-19.- Perturbación provocada por la base magnética

Se observa que la variación de carga provocada por la base magnética tiene un efecto negativo aproximado de 70 rad/seg en la velocidad angular, correspondiente aproximadamente al 45% de la máxima respuesta en lazo abierto del sistema.

Para la validación mediante comparación del mejor desempeño del controlador PI con “set point gain scheduling” ante variaciones de carga, utilizando las metodologías de control descritas en el capítulo 2, se realizaron los siguientes experimentos.

Experimento	Señal de Referencia (rad/seg)	Ganancias del Controlador		Frecuencia para Garantizar Estabilidad (Hz)	Tipo de Control
		$K_p$	$K_i$		
1	20	3	13	2.51	Óptimo
2	20	52.62	1656.03	41.84	Set Point Gain Scheduling
3	40	3	13	2.51	Óptimo
4	40	23.7	351.48	18.9	Set Point Gain Scheduling
5	60	3	13	2.51	Óptimo
6	60	13.95	128.83	11.16	Set Point Gain Scheduling

Tabla 4-5.- Especificaciones para los experimentos con variación de carga

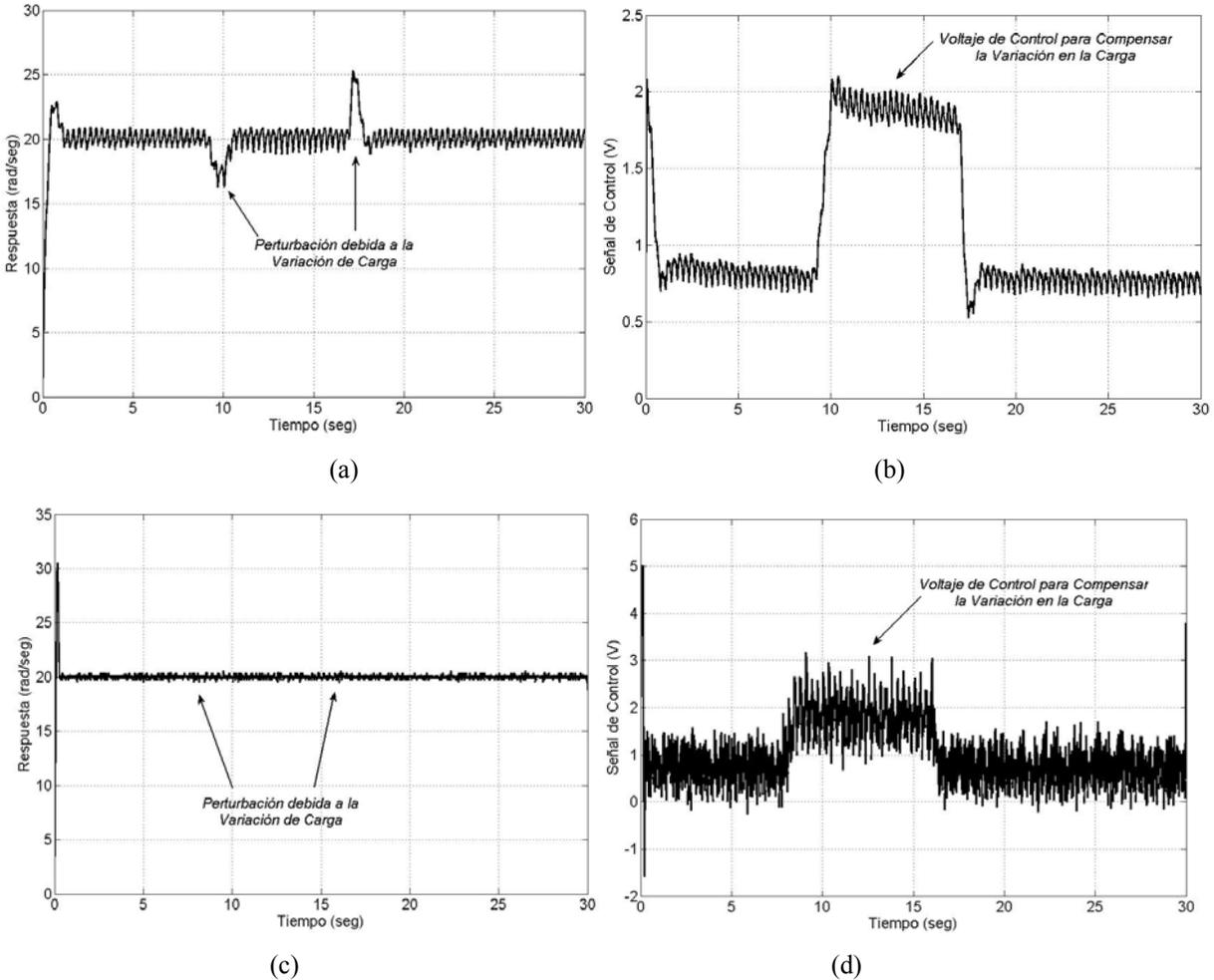
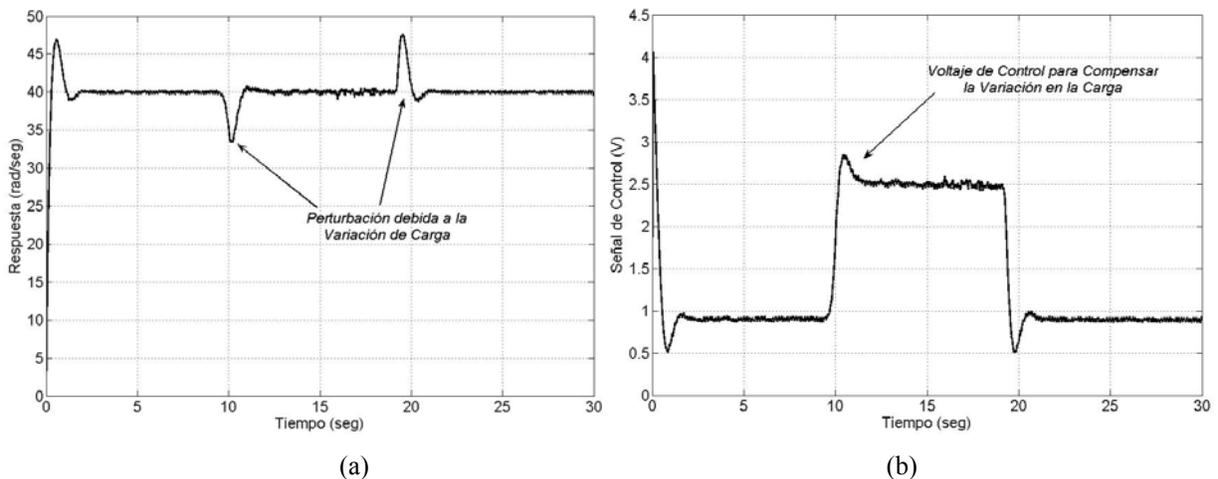


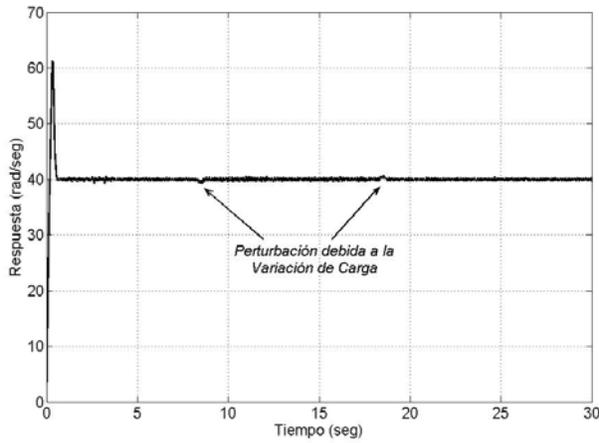
Figura 4-20.- Compensación de variación de carga. (a) y (c) Velocidad controlada para los experimentos 1 y 2, (b) y (d) Señal de control de los experimentos 1 y 2

La figura 4-20 muestra en (a) que la compensación de la perturbación debida a la variación de carga inducida por la base magnética toma más tiempo al control óptimo que al controlador con “set point gain scheduling” presentado en (c), lo que provoca fluctuaciones no deseadas en la velocidad angular del motor. Debido a la rápida respuesta del controlador la perturbación en (c) no tiene efecto alguno en la dinámica del sistema, aunque al observar la gráfica en (d) se aprecia que el voltaje de control incrementó su nivel para compensarla. La señal de control en (b) nunca alcanzó los 5V, correspondientes al máximo voltaje disponible, evitando la saturación en el servo-amplificador y disminuyendo por tanto el sobre-impulso en (a) comparado con el que presenta la respuesta en (c) cuya señal de control en (d) se satura al inicio del experimento. Puede demostrarse analíticamente que las fluctuaciones debidas a los fenómenos de “torque ripple” y desbalance en el disco se minimizan en (c) ya que si bien el

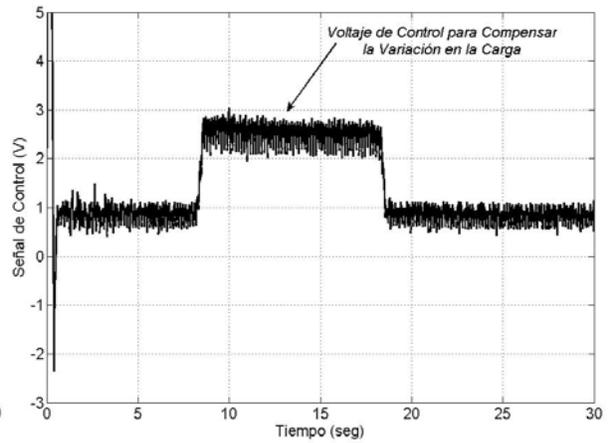
controlador no fue diseñado para compensar perturbaciones armónicas como es para estas el caso, la función de error es inversamente proporcional a  $K_i$  para cualquier tipo de perturbación. El esfuerzo de control por mantener la velocidad constante incluso con este tipo de fenómenos no modelados puede verse en (d) el cual comparado con (b) contiene armónicas de mayor amplitud en voltaje lo que explica la mejor compensación cuando el controlador se sintoniza utilizando el método descrito en el capítulo 2 para tener una mejor compensación cuando existen variaciones en la carga.

En la figura 4-21 se muestran los resultados para los experimentos 3 y 4, los cuales son análogos a 1 y 2 pero con una señal de referencia de 40 rad/seg. Una vez más, puesto que no existe saturación en la señal de control en (b) la velocidad en (a) presenta un sobre-impulso menor que en (c) y también la respuesta en (b) ante la variación de carga constante es más lenta que en (c). Esta ocasión en la gráfica (c) aunque es muy poco el efecto, se logra apreciar la perturbación en la dinámica del sistema. El efecto del desbalance y “torque ripples” es menor en (c) pero esta ocasión también en (a), esto se debe al aumento en la velocidad angular. De esta manera el esfuerzo de control tanto en (b) como en (d) contiene armónicas de menor amplitud comparadas con los experimentos 1 y 2. La misma explicación es válida para los experimentos de la figura 4-22.



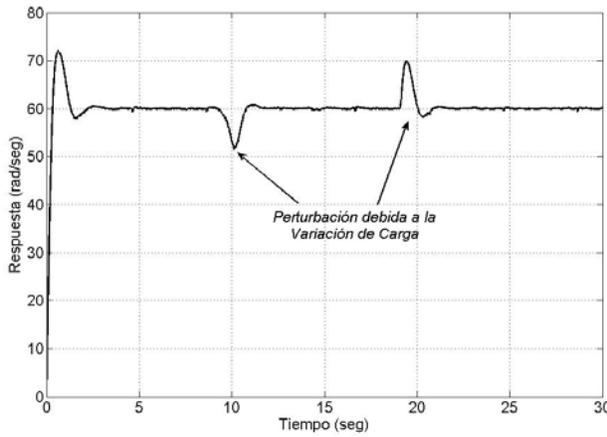


(c)

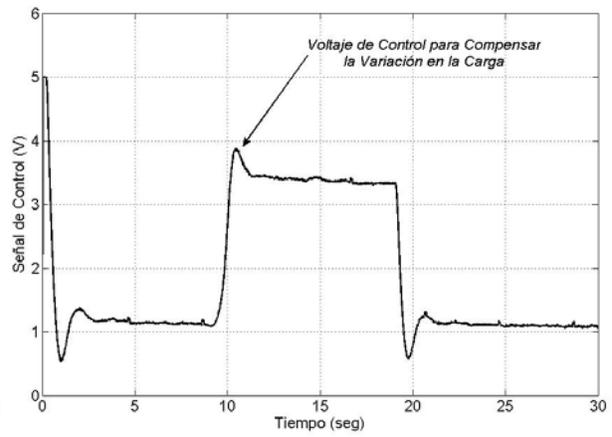


(d)

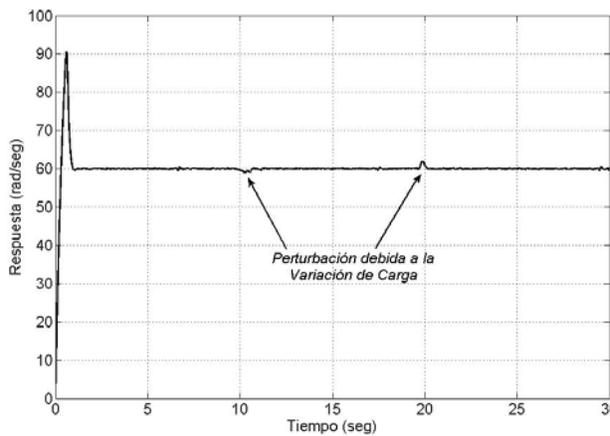
Figura 4-21.- Compensación de variación de carga. a) y c) Velocidad controlada para los experimentos 3 y 4, b) y d) Señal de control de los experimentos 3 y 4



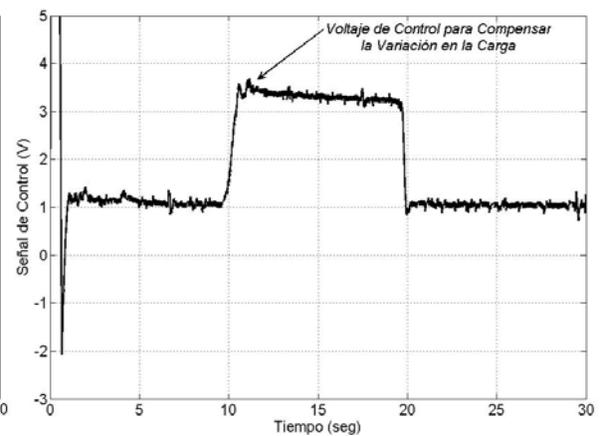
(a)



(b)



(c)



(d)

Figura 4-22.- Compensación de variación de carga. a) y c) Velocidad controlada para los experimentos 5 y 6, b) y d) Señal de control de los experimentos 5 y 6

La sintonización del controlador utilizando “set point gain scheduling” demostró además ser inmune al efecto que provoca la zona muerta, cuando menos para las condiciones del experimento que se muestra en la figura 4-16. Esto se debe a que la sintonización del controlador para una amplitud de referencia de 40 rad/seg requiere menos iteraciones para salir de la zona muerta que las requeridas por el control con ganancias óptimas. En la figura 4-23 se muestra el desempeño del controlador ante una entrada de referencia de  $40\sin(0.25t)\text{rad/seg}$ , sintonizado con las ganancias que se muestran en la tabla 4-5 correspondientes al experimento 4 de la misma. Se observa además de un mejor desempeño en general, la inmunidad a la zona muerta utilizando esta metodología de control.

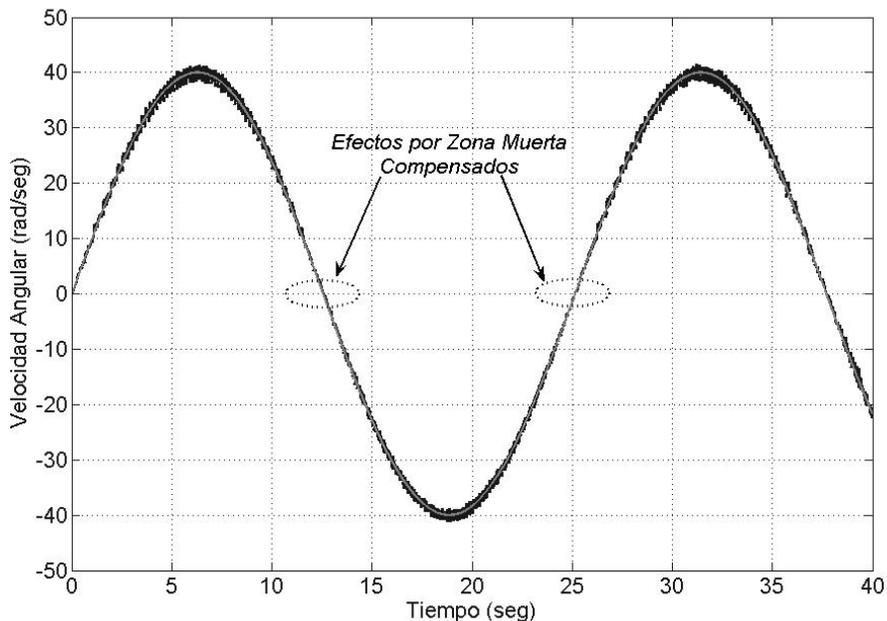


Figura 4-23.- Compensación de zona muerta por set point gain scheduling

## 4.5 Anti-Wind-Up

La saturación en el servo-amplificador ha provocado un mayor sobre-impulso y por tanto un mayor tiempo de establecimiento en la mayoría de los experimentos realizados hasta el momento. Esto es provocado por el enrollamiento o “wind-up” que se genera en la acción integral del control PI debido al error acumulado durante el periodo de saturación. Aunque el “wind up” no fue previsto en el diseño, su efecto en el desempeño del controlador es indeseable. La forma de eliminar el enrollamiento consiste en desconectar la parte integral del controlador

cuando el sistema presente saturación. Para evitar el enrollamiento, de acuerdo con [Glattfelder 2003] se puede escribir la ecuación en diferencias del algoritmo de control con “anti-wind-up” como se muestra a continuación.

$$u(k) = u(k-1) + \left( \frac{\Delta K_A K_i - K_p}{K_p} \right) e_a(k-1) + (\Delta K_i - K_p) e(k-1) + K_p e(k) + e_a(k) \quad (25)$$

donde  $u(k)$  y  $e(k)$  son las señales actuales de control y de error respectivamente,  $u(k-1)$  y  $e(k-1)$  las señales de control y de error un tiempo de muestreo atrás,  $\Delta$  es el tiempo de muestreo,  $K_p$  y  $K_i$  son las ganancias del controlador,  $e_a(k)$  y  $e_a(k-1)$  es la diferencia entre la señal de control con y sin saturación actual y un tiempo de muestreo actual respectivamente, y  $K_A$  es un parámetro de diseño. El diagrama de bloques para un controlador en tiempo continuo con la estructura descrita en la ecuación 25 se muestra en la figura 4-24.

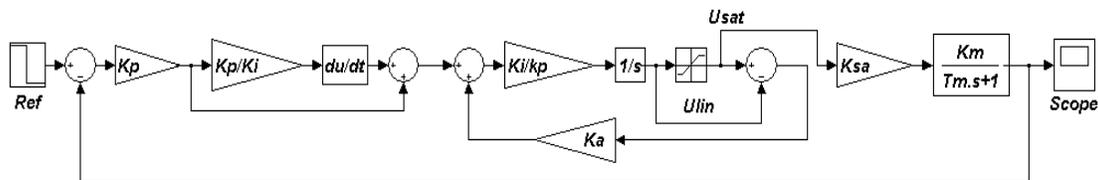


Figura 4-24.- Sistema de control PI con anti-wind-up

Utilizando las ganancias de la tabla 4-5 para el experimento 2, con  $K_a = 2$ , los resultados experimentales se muestran en la figura 4-25.

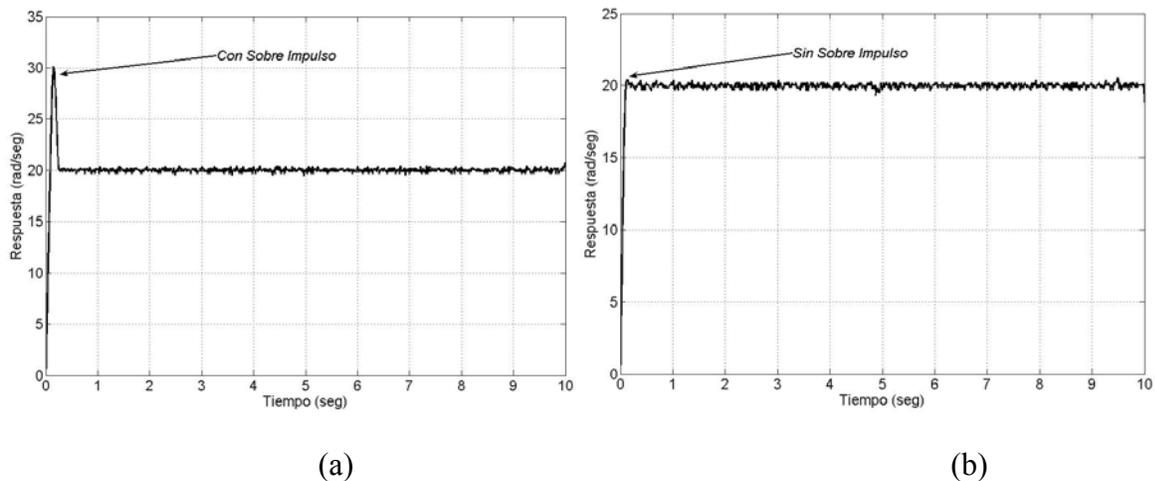


Figura 4-25.- Desempeño de control, (a) Set-point-gain-scheduling sin anti-wind-up, (b) Set-point-gain-scheduling con anti-wind-up

En la figura 4-25-b se observa un sobre-impulso cercano al 0% al utilizar el algoritmo de control con anti-wind-up, mejorando así la respuesta transitoria del sistema comparado con el 50% de sobre-impulso que se presenta en (a) cuyo algoritmo de control no incluye la eliminación del efecto de enrollamiento.

La comparación del experimento de la figura 4-15 correspondiente a una señal de referencia de 120 rad/seg cuya señal de control permanece en saturación por más tiempo que la del experimento de la figura 4-25 y por tanto el enrollamiento en la acción integral del controlador provoca efectos más visibles en la respuesta transitoria del sistema, y el desempeño para la misma señal de referencia utilizando la estrategia de control del diagrama de bloques de la figura 4-24 que implementa el anti-enrollamiento en el PI se muestra en la figura 4-26.

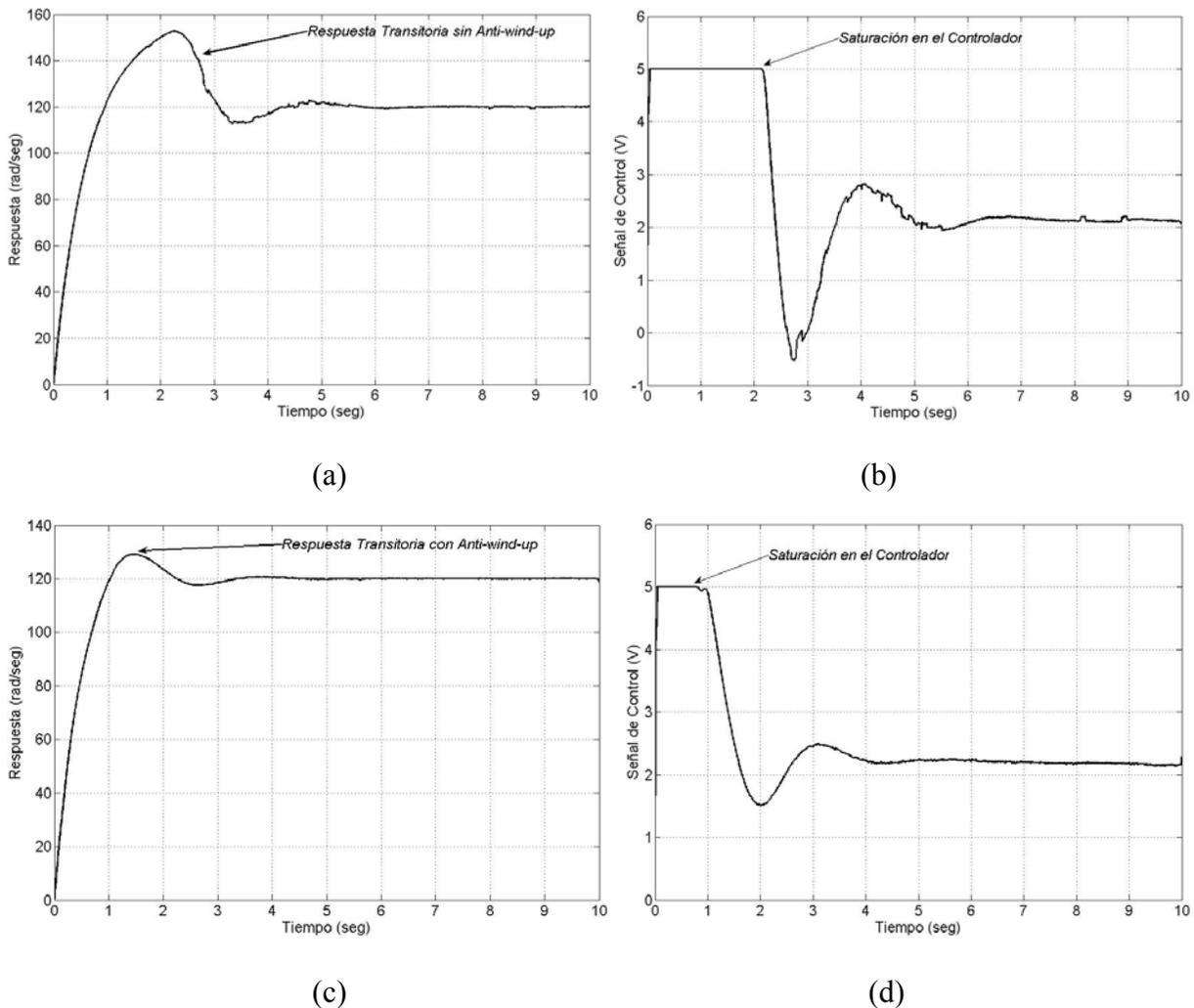


Figura 4-26.- Desempeño para una referencia de 120 rad/seg, (a) y (b) Respuesta y señal de control sin wind-up, (c) y (d) con anti-wind-up

En la figura 4-26-a y 4-26-b se observa la respuesta transitoria para el mismo sistema sin anti-wind-up en (a) y con anti-wind-up en (b). El efecto negativo en la dinámica del sistema causado por la saturación del controlador en (a) es causado por el tiempo que toma realizar el desenrollamiento de la acción integral, este tiempo en el que el controlador continua enviando su máximo voltaje puede observarse al comparar las señales de control en (b) y (d).

# Capítulo 5 Conclusión y perspectivas del trabajo

## 5.1 Conclusión

En cumplimiento de los objetivos propuestos al inicio de este trabajo, se logró la integración e instrumentación del arreglo experimental para el sistema de control de velocidad de motores de corriente directa de imán permanente con escobillas. Con el software desarrollado fue posible el manejo de datos para la visualización e identificación experimental de los parámetros necesarios para obtener el modelo del sistema e implementar los algoritmos de control de acuerdo a las estrategias propuestas.

La compensación de la zona muerta mediante el uso de la función signo demostró reducir el efecto que esta causa en el seguimiento de trayectorias sinusoidales más no eliminarlo del todo. Sin embargo, utilizando el método de sintonización por “set-point-gain-scheduling” logró compensar la zona muerta completamente y responder rápidamente ante perturbaciones debidas a variaciones súbitas en la carga del motor. El sobre-impulso excesivo que presentaba el sistema debido a la saturación en el servo-amplificador logró reducirse a valores cercanos a su eliminación utilizando la estructura del controlador con “anti-wind-up”.

Los detalles de programación y utilización del código aún no están documentados, sin embargo, con el software utilizado para el arreglo experimental es posible probar cualquier algoritmo de control en motores de corriente directa introduciendo su ecuación en diferencias. Existen inconvenientes en el sistema de control desarrollado debidos a los límites de velocidad de adquisición que determina el puerto paralelo estándar y la tarjeta de puerto paralelo universal que se utilizó, además de las inestabilidades incontrolables que presenta el sistema operativo Windows XP. Se esta trabajando en migrar el código a microcontroladores para poder mantener las tareas críticas del lazo de control separadas de Windows. Se considera que es viable la comercialización del software y se esta trabajando en colaboración con investigadores y alumnos en CICATA-IPN Unidad Querétaro para en un futuro introducirlo al mercado.

Aunque en este trabajo no se mencionó el uso de LABVIEW y MATLAB para el control de velocidad en motores de corriente directa estas se utilizaron para realizar pruebas de visualización del desempeño en tiempo real de los algoritmos de control. Ya que el desarrollo de tarjetas de control basadas en microcontroladores puede representar una solución económica para

la enseñanza y aplicación del control de movimiento, se está trabajando en la construcción de bloques funcionales para estas plataformas con interfaz USB 2.0.

## **5.2 Perspectivas de Trabajo**

La tercera necesidad que motivó este trabajo en el banco experimental de esfuerzo-deformación, puede ya ser cubierta por este desarrollo de acuerdo a las características logradas del sistema. Es posible conocer la posición de acuerdo a la necesidad inicial y aplicar trayectorias sinusoidales para las pruebas de fatiga por ciclado.

Los algoritmos de control propuestos en este trabajo se basaron en métodos analíticos y gráficos fuera de línea. Se ha pensado en utilizar métodos lineales y no lineales de identificación y control de sistemas dinámicos en línea para recalcular la planta y el controlador en tiempo real y sintonizar de manera automática al controlador.

Investigadores del departamento de Micro-Mecánica y Mecatrónica del CCADET-UNAM, están interesados en el trabajo desarrollado. Estos investigadores trabajan en alternativas de bajo costo para la manufactura de micro-componentes mediante el desarrollo de micro-máquinas herramientas (MMH's) y micro-manipuladores (MM's) y tienen la necesidad de controlar motores de corriente directa, ya que esto les permitirá hacer manufactura de micro-dispositivos de una manera más rápida debido a las características de velocidad de estos motores con respecto a los motores de pasos que es lo que ellos están utilizando como servo-actuadores en este momento.

El hecho de desarrollar MMH's de bajo costo, trae como consecuencia no tener buenas características de precisión. Por esta razón, es importante determinar las fuentes de error de las MMHs y desarrollar sistemas de control que permitan mejorar las características de operación de las MMHs [Caballero 2007]. Como trabajo futuro en colaboración con CCADET-UNAM se han planteado los siguientes objetivos:

- 1.- Investigar los fundamentos teóricos para el análisis y control de sistemas no lineales, para realizar las simulaciones de identificación y control de una MMH basada en mecanismos paralelos. En particular, simulaciones orientadas al control de la MMH sujeta a fenómenos debidos a fricción [Armstrong 1991].

2.- Evaluar, seleccionar y proponer el algoritmo de control más adecuado de acuerdo a los parámetros de desempeño requeridos para el micro-maquinado. Implementar el algoritmo en uno de los actuadores que han de utilizarse en la MMH.

3.- Integrar los actuadores e implantar el sistema de control propuesto en la MMH. Hacer el diseño de experimentos para la validación y determinación de las capacidades de posicionamiento y seguimiento de trayectorias del sistema de control desarrollado.

## Referencias

Armstrong H elouvy Brian, *Control of Machines with Friction*. Massachussets, MA. : Kluwer Academic Publishers, 1991.

 str m Karl J., and H agglund Tore, *PID Controllers: theory, design, and tuning*. Research Triangle Park, NC. : ISA – Instrument Society of America, 1995.

 str m Karl J., and H agglund Tore, *Advanced PID Control*. Research Triangle Park, NC. : ISA – Instrumentation, Systems, and Automation Society, 2006.

Axelson Jan, *Parallel Port Complete*. Madison WI. : Lakeview Research, 2000.

Axelson Jan, 2008. <http://www.lvr.com/jansfaq.htm>

Bartos Frank J., “Servo system application tips”, *Control Engineering*, December 2006.

Beaven R. W., Wilkes L. D., Garvey S. D., and Friswell M., “The application of setpoint gain scheduling to improve the performance of high speed independent drives”, IEE 1994.

Burden Richard L., and Faires Douglas, *Numerical Analysis*. Pacific Grove, CA. : Brooks-Cole, 1997.

Caballero Ruiz Alberto, Ruiz Huerta Leopoldo, Tatiana Baidyk and Ernst Kussul, “Geometrical error analysis of a CNC micro-machine tool”, *Mechatronics*, Volume 17, Issues 4-5, Elsevier 2007.

Canon, 2008. <http://www.canon-prec.co.jp/english/products/dcmotor/pdf/FN30.pdf>

Canudas de Wit C., Olsson H.,  str m K. J., and Lischinsky P., “A new model for control of systems with friction”, *IEEE Transactions on Automatic Control*, Volume 40, Issue 3, March 1995.

Castillo Casta neda E., Takeda Y., “Improving path accuracy of a crank-type 6-dof parallel mechanism by stiction compensation”, *Mechanism and Machine Theory*, Elsevier 2006.

Castillo Casta neda E., Fonseca Reyes M.J., L pez Caj n C.S., “Non-linear control to compensate low velocity friction of a planar parallel robot”, *12th IFToMM World Congress*, Besan on France, June 2007.

Ceballos Fco. Javier, *Enciclopedia del Lenguaje C++*. Madrid, Espa a. : Alfaomega Ra-Ma, 2004.

Chapman Stephen J., *M quinas El ctricas*. Santaf  de Bogot , Colombia. : McGraw-Hill Interamericana, 1987.

Chang J-L., "Design of a robust controller using only output feedback to solve the servomechanism problem", *Control Theory and Applications*, IEE Proceedings, Volume 150, Issue 1, January 2003.

Cheng Guoyang and Peng Kemao, "Robust composite nonlinear feedback control with application to a servo positioning system", *IEEE Transactions on Industrial Electronics*, Volume 54, Issue 2, April 2007.

Chien K. L., Hrones J. A., and Reswick J. B., "On the automatic control of generalized passive systems", *Transactions ASME*, 74, 1952.

CUI, 2008. <http://www.cui.com/adtemplate.asp?invky=490429>

Dorf Richard C., and Bishop Robert H., "Design using performance indices", *The Control Handbook*, Boca Raton, Florida. : CRC Press LLC, 1995.

Dorf Richard C., and Bishop Robert H., *Modern Control Systems*. Upper Saddle River, NJ. : Pearson Prentice Hall, 2005.

Electro-Craft Corporation, *DC Motors Speed Controls Servo Systems*. Hopkins, Minn. : Electro-Craft Corporation, 1980.

Fitzgerald A. E., Kingsley Charles Jr., and Umans Stephen D., *Electric Machinery*. McGraw Hill, 1985.

Franklin Gene F., Powell J. David, and Workman Michael, *Digital Control of Dynamic Systems*. Menlo Park, CA. : Addison Wesley Longman, Inc., 1998.

Glattfelder A. H., and Schaufelberger W., *Control Systems with Input and Output Constraints*. Advanced textbooks in control and signal processing, Springer-Verlag London Limited, 2003.

Glowacz Zygfryd, and Glowacz Witold, "Mathematical model of dc motor for analysis of commutation processes", *IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics, and Drives*, 2007.

Graupe Daniel, *Identification of Systems*. New York, N.Y. : Van Nostrand Reinhold Company, 1972.

Hong S.W., Shun Y.J. and Lee H.S., "An efficient method for identification of motion error sources from circular test results in NC machines", *Int. J. Mach. Tools Manufact.*, Vol. 37, No. 3, pp. 327-340, 1997.

Kernighan Brian W., and Ritchie Dennis M., *C Programming Language*. Upper Saddle Rive, NJ. : Prentice Hall PTR, 1998.

Kuo Benjamin C., and Tal Jacob., *DC Motors and Control Systems*. Champaign, Illinois. : SRL Publishing Company, 1978.

Kuo Benjamín C., *Sistemas de Control Automático*. Prentice Hall, 1996.

Lathrop R. C., and Graham. D., “The synthesis of optimum response: criteria and standard forms. II”, *Transactions AIEE* 72, November 1953.

Lobosco Orlando Silvio, “Modeling and simulation of DC motors in dynamic conditions”, IEEE 1997.

Logix4u, 2008. [http://logix4u.net/Legacy\\_Ports/Parallel\\_Port/Inpout32.dll\\_for\\_Windows\\_98/2000/NT/XP.html](http://logix4u.net/Legacy_Ports/Parallel_Port/Inpout32.dll_for_Windows_98/2000/NT/XP.html)

Markovsky Ivan, Willems Jan C., Huffel Van Sabine, De Moor Bart, and Pintelon Rik, “Application of structured total least squares for system identification and model reduction”, *IEEE Transactions on Automatic Control*, Volume 50, NO. 10, October 2005.

MinGW, 2008. <http://www.mingw.org/>

Mohan Ned, Undeland Tore M., and Robbins William P., *Power Electronics: Converters, Applications, and Design*. Danvers, MA. : John Wiley and Sons, Inc., 2003.

MSDN, 2008. <http://msdn.microsoft.com/en-us/default.aspx>

Nava Palomares Hugo, “Análisis de las deformaciones en un mecanismo paralelo utilizado para tareas de fresado”, *Tesis de Grado de Maestría*, CICATA-IPN Unidad Querétaro, Diciembre de 2007.

Petzold Charles, *Programming Windows*. Redmond, Washington. : Microsoft Press, 1999.

Pichardo Centeno Jesús, Juan de Dios Ortiz, Ricardo Durán, Iván Domínguez, Jorge Huerta, Adrián García, “Desarrollo de sistemas de medición para un tribómetro de perno en disco”, *5º Congreso Internacional sobre Innovación y Desarrollo Tecnológico*, IEEE Sección Morelos, Octubre de 2007.

Schildt Hebert, *C++ The Complete Reference*. Berkeley California. : McGraw Hill Osborne, 2003.

SpeedUp, 2008. <http://www.youtube.com/watch?v=WUyD5b-wmDU&feature=related>

ST, 2008. <http://www.st.com/stonline/products/literature/ds/1773/l298.pdf>

Startups, 2008. [http://www.pacs-portal.co.uk/startup\\_index.htm](http://www.pacs-portal.co.uk/startup_index.htm)

Stroustrup Bjarne, *The C++ Programming Language*. Indianapolis, IN. : Pearson Education, 2000.

TDC, 2008. <http://www.transdigital.net/>

Tektronix 2008. <http://www2.tek.com/cmswpt/psdetails.lotr?ct=PS&cs=psu&ci=13833&lc=EN>

ValueTesters, 2008. <http://www.valuetesters.com/UniSource-UNI-PS-3030DU-Dual-Output-DC-Power-Supply.php>

Yañez Valdez Ricardo, “Resolución de mecanismo paralelo planar 3RRR impulsado por actuadores eléctricos”, *Tesis de Grado de Maestría*, CICATA-IPN Unidad Querétaro, Diciembre de 2007.

Yongpeng Zhang, Cajetan Akujuobi, Leang-Sang Shie, and Xiaohui Gu, “Observer-based load disturbance compensation for motor drive with DSP implementation”, IEEE 2005.

Ziegler J. G., and Nichols N. B., “Optimum settings for automatic controllers”, *Transactions of the ASME*, November 1942.

## Publicaciones

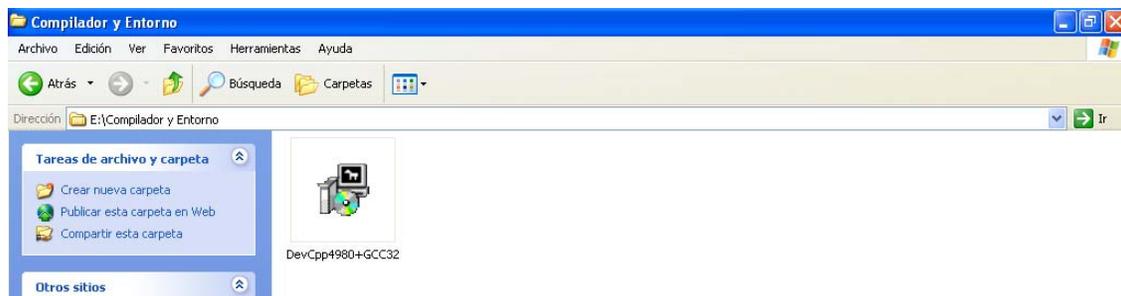
Internacionales y nacionales con refereo, memorias de congreso nacionales con refereo

1. Diagonal Recurrent Neural Networks for Speed Control of a DC Motor, Alfonso Noriega, Carlos Silva, **Jesús Pichardo**, First International Conference on Industrial Informatics, CICINDIN – Ciudad de México, México, Noviembre de 2007.
2. Desarrollo de Sistemas de Medición y Control para un Tribómetro de Perno en Disco, **Jesús Pichardo**, Juan de Dios Ortiz, Ricardo Durán, Iván Domínguez, Jorge Huerta, Adrián García, 5° Congreso Internacional sobre Innovación y Desarrollo Tecnológico, IEEE Sección Morelos, CIINDET – Cuernavaca-Morelos, México, Octubre de 2007.
3. Real-time Observations of Wear in a Pin-on-Disk Tribometer, Iván Domínguez, Jorge Huerta, Adrián García, Juan de Dios Ortiz, **Jesús Pichardo**, 16th International Conference on Wear of Materials, WOM - Montreal, Canada, Abril de 2007.
4. In Situ Measurement of Laser-Light Scattering to Monitor Wear Surface Studies in a Tribometer, Juan de Dios Ortiz, **Jesús Pichardo**, Iván Domínguez, Adrián García, Jorge Huerta, 6° Simposio La Óptica en la Industria, Monterrey-Nuevo León, México, Marzo de 2007.
5. Mediciones de Cambio de Intensidad en Luz Láser Esparcida, Aplicada al Monitoreo de Desgaste, Iván Domínguez, Jorge Huerta, Ricardo Montes, Juan de Dios Ortiz, **Jesús Pichardo**, Adrián García, Miguel Aguilar, David Jaramillo, Simposio de Metrología 2006, Querétaro-Querétaro, México, Octubre de 2006.

## Apéndice 1 Manejo del Compilador y Entorno GNU C++

El software fue escrito utilizando un compilador GCC para Win32 (un compilador C++ de la colección de compiladores GNU) y el entorno de desarrollo Dev-Cpp, los cuales se adjuntan al CD donde se encuentra la tesis, aunque puede conseguirse en Internet ya que son de libre distribución.

Para instalar el compilador y el entorno solo se debe dar doble click en el ejecutable que se encuentra en la carpeta “Compilador y Entorno” que se encuentra en el CD y seguir los pasos que vayan apareciendo.

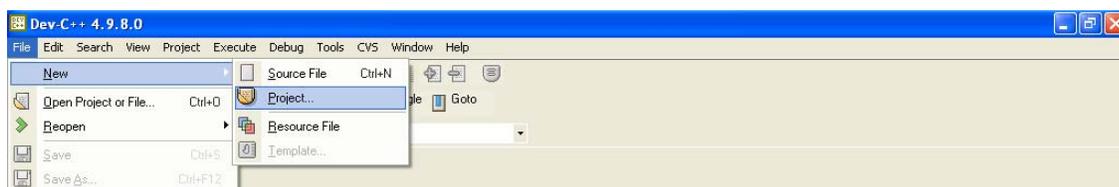


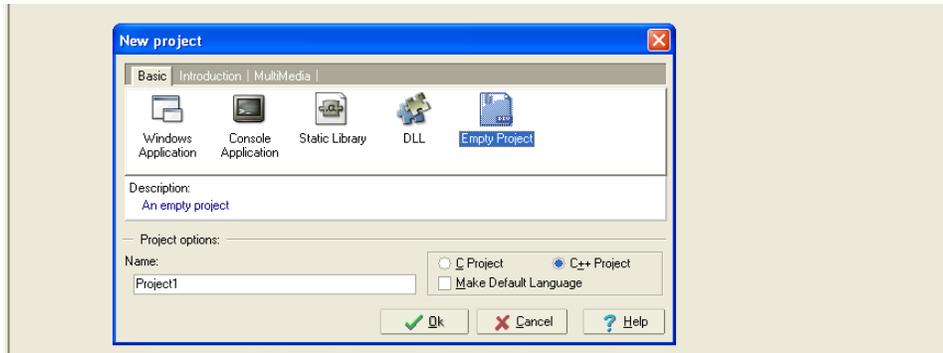
Para hacer la identificación o cualquier aplicación que se desee realizar utilizando las clases que he desarrollado es necesario crear una carpeta nueva e incluir en esta carpeta los archivos que contienen las clases así como la DLL inpout.dll, estos archivos se encuentran en el CD en la carpeta Clases.

Una vez hecho lo anterior, el procedimiento para crear un proyecto nuevo en Dev-C++ es el siguiente:

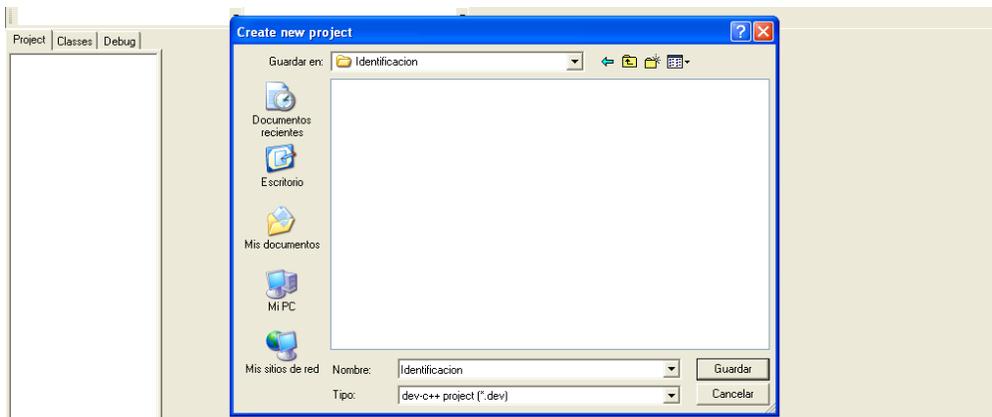
Abrir Dev-C++

Crear un proyecto nuevo

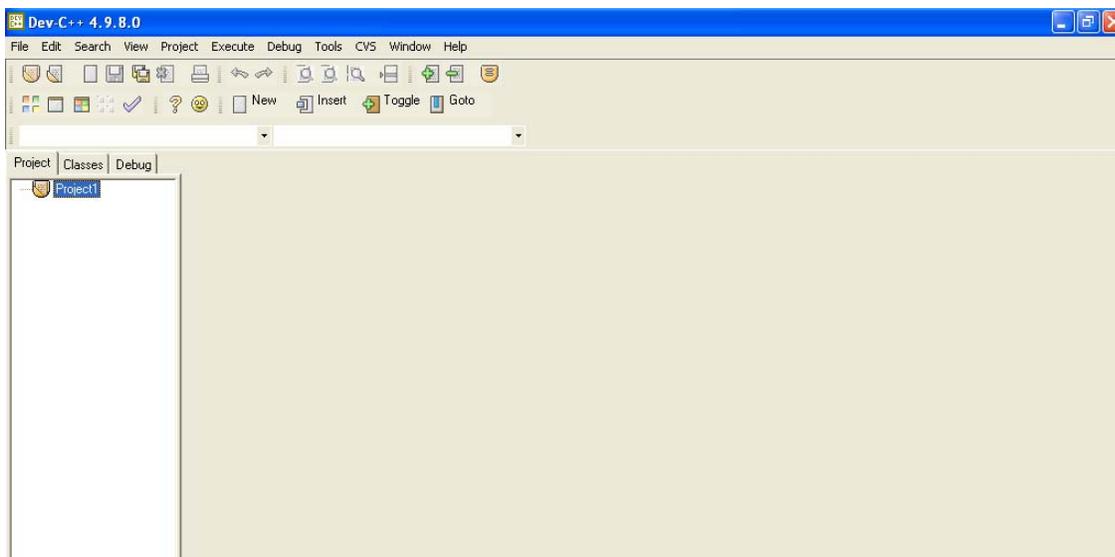




Escoger la opción crear un proyecto vacío, darle nombre al proyecto y lo guardarlo en nuestra carpeta.

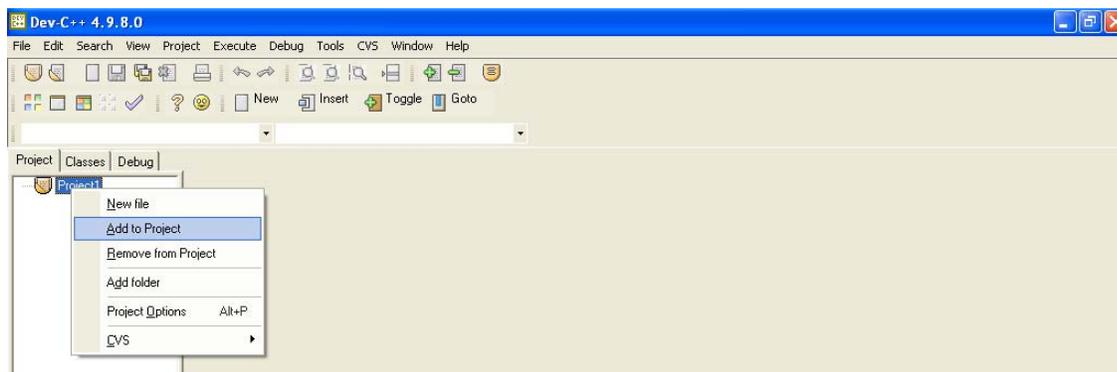


Una vez hecho esto el entorno de desarrollo muestra lo siguiente

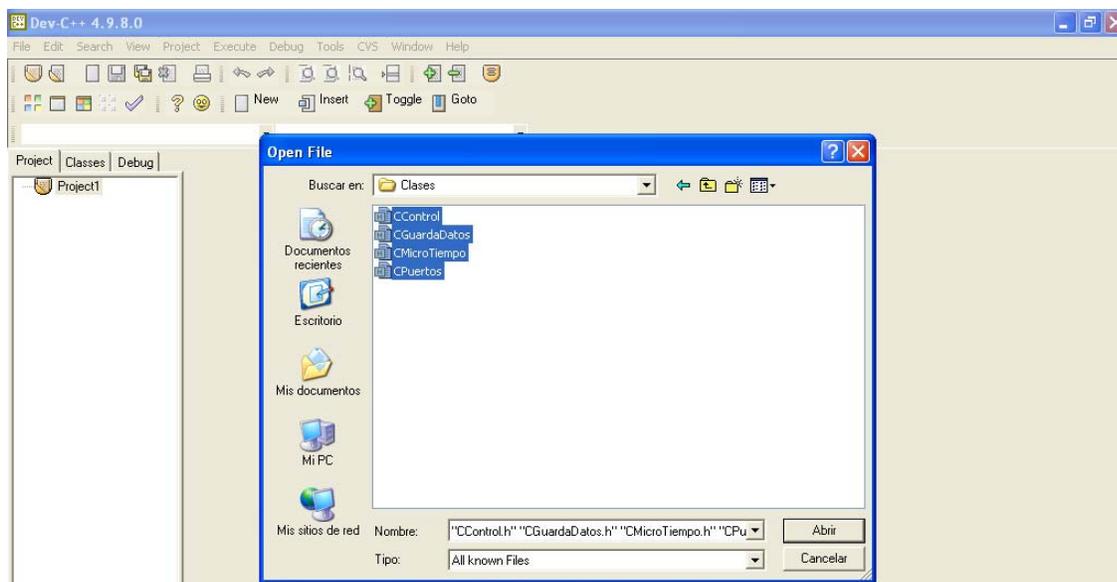


Ahora hay que agregar a nuestro proyecto las clases CPuertos, CMicroTiempo, CControl y CGuardaDatos las cuales van a ser utilizadas en la aplicación y que ahora se encuentran en

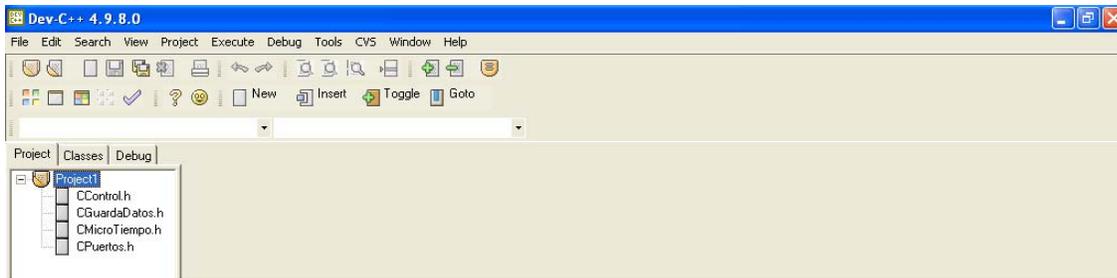
nuestra carpeta. Para agregar las clases al proyecto hacer click derecho sobre el proyecto y aparecerá una ventana como se muestra en la siguiente figura.



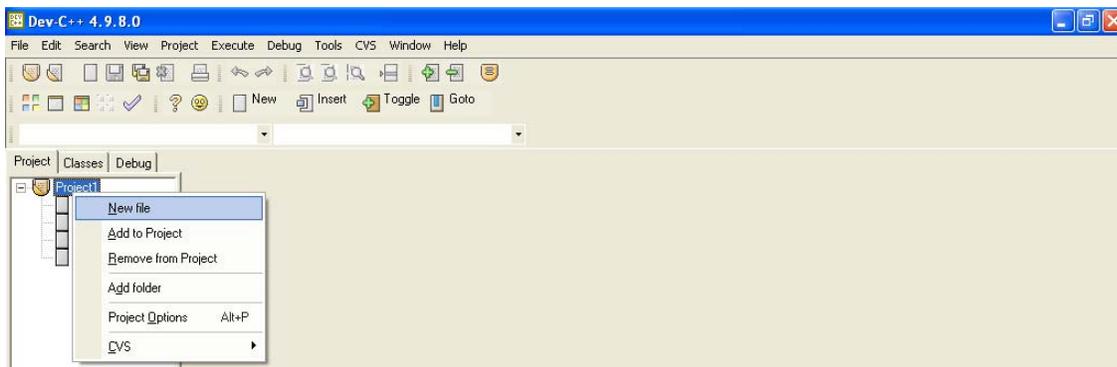
Al dar clic en Add to Project aparece otra ventana, donde podemos escoger los archivos que se van a agregar al proyecto. Desde ahí se puede acceder a nuestra carpeta, seleccionar las 4 clases y dar click en abrir.



De esta manera las clases, que son las “librerías” que va a utilizar la aplicación, se agregan al proyecto, que ahora luce así:

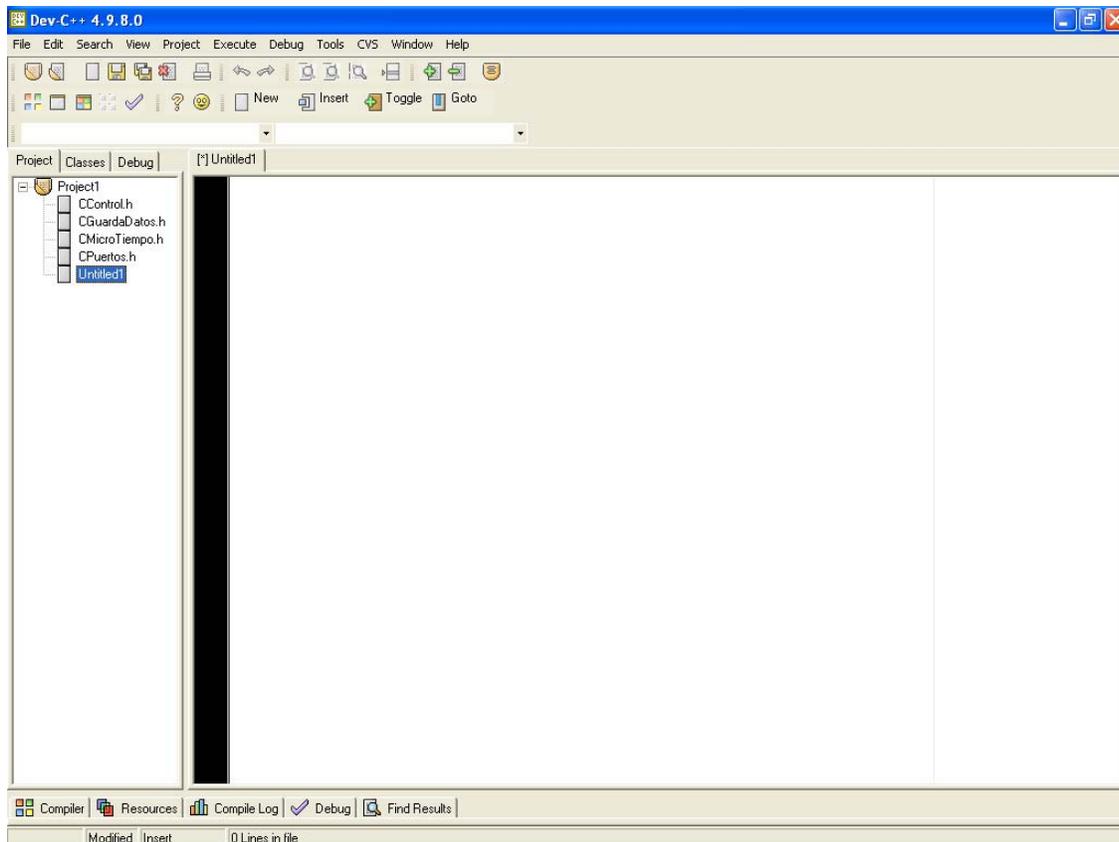


Ahora se debe agregar un archivo nuevo en cual se van a hacer las llamadas a los métodos de las clases y que es donde va a codificarse la aplicación. El nombre por omisión para un archivo nuevo es "Untitled1" esto es:



Y se agrega automáticamente al proyecto

Podemos dar el nombre que deseemos al archivo que hemos incluido, siempre debemos decir su tipo, en este caso Untitled1.cpp. Nuestro proyecto tiene ahora la siguiente forma:



Con esto podemos empezar a trabajar con los métodos de los objetos de cada una de las clases.

## Apéndice 2 Código de clases

### 1. Clase CPuertos

```
using namespace std;
```

```
class CPuertos
```

```
{
```

```
private:
```

```
    typedef short _stdcall (*infuncPtr)(short portaddr);
```

```
    typedef void _stdcall (*oufuncPtr)(short portaddr, short datum);
```

```
    HINSTANCE hLib;
```

```
    infuncPtr inp32;
```

```
    oufuncPtr oup32;
```

```
    int Out_port;
```

```
    int In_port;
```

```
public:
```

```
    inline CPuertos()
```

```
    {
```

```
        hLib = LoadLibrary("inpout32.dll");
```

```
        if (hLib == NULL)
```

```
            cout << "No pudo cargarse la libreria" << endl;
```

```
        inp32 = (infuncPtr) GetProcAddress(hLib, "Inp32");
```

```
        if (inp32 == NULL)
```

```
            cout << "No pudo obtenerse la direccion para Inp32" << endl;
```

```
        oup32 = (oufuncPtr) GetProcAddress(hLib, "Out32");
```

```
        if (oup32 == NULL)
```

```
            cout << "No pudo obtenerse la direccion para Out32" << endl;
```

```
        Out_port = 0x378;
```

```
        In_port = 0x379;
```

```
        oup32(Out_port,0);
```

```
    }
```

```
    inline ~CPuertos()
```

```

    {
        oup32(Out_port,0);
        FreeLibrary(hLib);
    }

    inline void EscribePuerto(int valor)
    {
        oup32(Out_port,valor);
    }

    inline int LeePuerto()
    {
        inp32(In_port);
    }
};

```

## 2. Clase CMicroTiempo

```
using namespace std;
```

```

class CMicroTiempo
{
protected:
    LARGE_INTEGER m_liStart;
    LARGE_INTEGER m_liStop;

    LONGLONG m_llFrequency;
    LONGLONG m_llCorrection;

double absolute_time;
public:
    inline CMicroTiempo()
    {
        LARGE_INTEGER liFrequency;

        QueryPerformanceFrequency(&liFrequency);
        m_llFrequency = liFrequency.QuadPart;

        // Calibration
        Inicia();
        Termina();
    }
};

```

```

m_lliCorrection = m_liStop.QuadPart-m_liStart.QuadPart;

absolute_time = 0;
Inicia();
}

inline ~CMicroTiempo()
{

}

inline void Inicia()
{
// Ensure we will not be interrupted by any other thread for a while
Sleep(0);
QueryPerformanceCounter(&m_liStart);
}

inline void Termina()
{
QueryPerformanceCounter(&m_liStop);
}

inline double TiempoTranscurrido() const
{
return (double)(m_liStop.QuadPart-m_liStart.QuadPart-m_lliCorrection)*1000000.0 / ((double)m_lliFrequency);
}

inline double TiempoAbsoluto()
{
Termina();

absolute_time = TiempoTranscurrido();

return absolute_time;
}

inline void Reset_TiempoAbsoluto()
{
Inicia();
}
};

```

### 3. Clase CGuardaDatos

```
using namespace std;
```

```
class CGuardaDatos
```

```
{
```

```
private:
```

```
    vector <long int> DatoEntero;
```

```
    vector <double> DatoDoble;
```

```
    int EsEntero;
```

```
public:
```

```
    inline CGuardaDatos()
```

```
    {
```

```
    }
```

```
    inline ~CGuardaDatos()
```

```
    {
```

```
    }
```

```
    inline void InsertaDato(long int d)
```

```
    {
```

```
        EsEntero = 1;
```

```
        DatoEntero.push_back(d);
```

```
    }
```

```
    inline void InsertaDato(double d)
```

```
    {
```

```
        EsEntero = 0;
```

```
        DatoDoble.push_back(d);
```

```
    }
```

```
    inline void VaciaDato()
```

```
    {
```

```
        if(EsEntero == 1)
```

```
        {
```

```
            DatoEntero.clear();
```

```
        }
```

```
        else
```

```
        {
```

```
            DatoDoble.clear();
```

```

    }
}

inline void GuardaDato(char nombre1[])
{
    ofstream out(nombre1, ios::out | ios::binary);

    out.precision(8);

    if(EsEntero == 1)
    {
        for(long int i = 0; i < DatoEntero.size(); i++)
            out << DatoEntero[i] << " ";
    }
    else
    {
        for(long int i = 0; i < DatoDoble.size(); i++)
            out << DatoDoble[i] << " ";
    }

    out.close();
}
};

```

#### 4. Clase CEncoder

```
using namespace std;
```

```
class CEncoder
```

```

{
    friend class CPuertos;

private:
    CPuertos puertos;
    int vpa, vpp, vpa2, vpp2;
    int diferencia, signo, igualdad, direccion, puerto;
    long int cuentas;
public:
    inline CEncoder()
    {
        cuentas = 0;
    }
}

```

```
vpa = puertos.LeePuerto();
vpa = vpa & 24;
switch (vpa)
{
    case 0: vpa = 1; break;
    case 8: vpa = 2; break;
    case 24: vpa = 3; break;
    case 16: vpa = 4; break;
}
}
```

```
inline ~CEncoder()
{
}
}
```

```
inline long int InicializaCuentas()
{
    cuentas = 0;
}
```

```
vpa = puertos.LeePuerto();
vpa = vpa & 24;
switch (vpa)
{
    case 0: vpa = 1; break;
    case 8: vpa = 2; break;
    case 24: vpa = 3; break;
    case 16: vpa = 4; break;
}
return cuentas;
}
```

```
inline long int CalculaCuentas()
{
vpp = vpa;
vpa = puertos.LeePuerto();
vpa = vpa & 24;

switch(vpa)
{
case 0: vpa = 1; break;
case 8: vpa = 2; break;
case 24: vpa = 3; break;
}
```

```

case 16: vpa = 4; break;
}

if(vpa != vpp)
{
diferencia = vpa - vpp;

if(diferencia < 0)
signo = (-1);
else
signo = (1);

diferencia = abs(diferencia);

if(diferencia != 3)
igualdad = (-1);
else
igualdad = (1);

direccion = signo * igualdad;

if(direccion > 0)
{
cuentas ++;
}
else
{
cuentas --;
}
}

return cuentas;
}
};

```

## 5. Clase CPwm

```
using namespace std;
```

```
class CPwm
{
friend class CPuertos;
```

```

friend class CMicroTiempo;
private:
    CPuertos puertos;
    CMicroTiempo uTiempo;
    double Tiempo, TiempoPWM, auxTiempo;
    double periodo, ApInterno, ajuste_uSeg;
    int direccion, clockWise, counterClockWise;
public:
    inline CPwm(void)
    {
        clockWise = 6;
        counterClockWise = 5;
        Tiempo = TiempoPWM = auxTiempo = 0;
        ajuste_uSeg = 1000000;
        puertos.EscribePuerto(0);
    }

    inline ~CPwm(void)
    {
        puertos.EscribePuerto(0);
    }

    inline void PulseWM(double Fpwm, double Ap)
    {
        Tiempo = uTiempo.TiempoAbsoluto();
        TiempoPWM = Tiempo - auxTiempo;

        if(Ap >= 0)
        {
            direccion = clockWise;
        }
        else if(Ap < 0)
        {
            direccion = counterClockWise;
        }
        else;

        periodo = ajuste_uSeg/Fpwm;
        ApInterno = abs(Ap*periodo);

        if(TiempoPWM <= ApInterno)
        {
            puertos.EscribePuerto(direccion);
        }
    }

```

```

    }
    else if(TiempoPWM > ApInterno && TiempoPWM <= periodo)
    {
        puertos.EscribePuerto(0);
    }
    else
    {
        auxTiempo = auxTiempo + TiempoPWM;
        TiempoPWM = 0;
    }
}
};

```

## 6. Clase CCalentamiento

```

using namespace std;
class CCalentamiento
{
    friend class CMicroTiempo;
    friend class CPwm;
private:
    CMicroTiempo uTiempo;
    CPwm pwm;
    double Tiempo, reset_Tiempo, periodo, auxTiempo, Tiempo_Muestreo, ajuste_uSeg;
public:
    inline CCalentamiento()
    {
        ajuste_uSeg = 1000000;
        Tiempo = 0;
        Tiempo = reset_Tiempo = Tiempo_Muestreo = auxTiempo = 0;
    }

    inline ~CCalentamiento()
    {
    }

    inline void Calentando(double t_escalon)
    {
        periodo = ajuste_uSeg*t_escalon;
        reset_Tiempo = uTiempo.TiempoAbsoluto();
        while(Tiempo < (periodo*4) + reset_Tiempo)

```

```

{
    Tiempo = uTiempo.TiempoAbsoluto();
    Tiempo_Muestreo = Tiempo - auxTiempo;
    if(Tiempo < (periodo*1) + reset_Tiempo)
        pwm.PulseWM(100,1);
    else if(Tiempo > (periodo*1) + reset_Tiempo && Tiempo < (periodo*2) + reset_Tiempo)
        pwm.PulseWM(100,0);
    else if(Tiempo > (periodo*2) + reset_Tiempo && Tiempo < (periodo*3) + reset_Tiempo)
        pwm.PulseWM(100,-1);
    else
        pwm.PulseWM(100,0);
    if(Tiempo_Muestreo > 10000)
    {
        auxTiempo = auxTiempo + Tiempo_Muestreo;
        Tiempo_Muestreo = 0;
    }
    else;
}
};

```

## 7. Clase CRespuesta

```

using namespace std;
class CRespuesta
{
    friend class CMicroTiempo;
    friend class CPwm;
    friend class CEncoder;
    friend class CGuardaDatos;
private:
    CMicroTiempo uTiempo;
    CPwm pwm;
    CEncoder encoder;
    CGuardaDatos gDato, gDato2, gDato3, gDato4;
    double Tiempo, reset_Tiempo, periodo, auxTiempo, Tiempo_Muestreo, ajuste_uSeg;
    long int posicion;
    double velocidad, posicion_p, ajuste_cuenta_rad, amp_rampa;
    double frecuencia_pwm;
    int indicador;
public:
    inline CRespuesta()

```

```

{
ajuste_uSeg = 1000000;
ajuste_cuenta_rad = 0.015707963;
Tiempo = reset_Tiempo = Tiempo_Muestreo = auxTiempo = 0;
velocidad = 0;
posicion = encoder.InicializaCuentas();
posicion_p = encoder.InicializaCuentas();
indicador = 0;
}

inline ~CRespuesta()
{

}

inline void IniciaRespuesta()
{
Tiempo = reset_Tiempo = Tiempo_Muestreo = auxTiempo = 0;
velocidad = 0;
gDato.VaciaDato();
gDato2.VaciaDato();
gDato3.VaciaDato();
gDato4.VaciaDato();
uTiempo.Reset_TiempoAbsoluto();
posicion = encoder.InicializaCuentas();
posicion_p = encoder.InicializaCuentas();
indicador = 0;
}

inline void Escalon(double frec_pwm,double amp_step,double dura_step, double sample_time)
{
gDato.InsertaDato(posicion*ajuste_cuenta_rad);
gDato2.InsertaDato(Tiempo/ajuste_uSeg);
gDato3.InsertaDato(velocidad*ajuste_uSeg*ajuste_cuenta_rad);
periodo = ajuste_uSeg*dura_step;
sample_time = ajuste_uSeg*sample_time;
reset_Tiempo = uTiempo.TiempoAbsoluto();
while(Tiempo < periodo*2 + reset_Tiempo)
{
posicion = encoder.CalculaCuentas();
Tiempo = uTiempo.TiempoAbsoluto();
Tiempo_Muestreo = Tiempo - auxTiempo;
}
}

```

```

if(Tiempo < periodo + reset_Tiempo)
{
    pwm.PulseWM(frec_pwm,amp_step);
}
else
{
    pwm.PulseWM(frec_pwm,0);
}

if(Tiempo_Muestreo > sample_time)
{
    velocidad = (posicion - posicion_p)/Tiempo_Muestreo;
    posicion_p = posicion;
    gDato.InsertaDato(posicion*ajuste_cuenta_rad);
    gDato2.InsertaDato(Tiempo/ajuste_uSeg);
    gDato3.InsertaDato(velocidad*ajuste_uSeg*ajuste_cuenta_rad);
    auxTiempo = auxTiempo + Tiempo_Muestreo;
    Tiempo_Muestreo = 0;
}
else;
}
}

inline void GuardaEscalon()
{
    gDato.GuardaDato("Posicion_Escalon.txt");
    gDato2.GuardaDato("Tiempo_Escalon.txt");
    gDato3.GuardaDato("Velocidad_Escalon.txt");
}

inline void Rampa(double frec_pwm,double sample_time,int dir_mvto)
{
    frecuencia_pwm = frec_pwm;
    periodo = 2*ajuste_uSeg*sample_time*200;
    amp_rampa = 1*dir_mvto;
    gDato.InsertaDato(amp_rampa);
    gDato2.InsertaDato(velocidad);
    gDato3.InsertaDato(posicion);
    gDato4.InsertaDato(Tiempo);
    sample_time = ajuste_uSeg*sample_time;
    reset_Tiempo = uTiempo.TiempoAbsoluto();
    while(Tiempo < (periodo + 4*ajuste_uSeg + reset_Tiempo))
    {

```

```

posicion = encoder.CalculaCuentas();
Tiempo = uTiempo.TiempoAbsoluto();
Tiempo_Muestreo = Tiempo - auxTiempo;

if(Tiempo < (2*ajuste_uSeg + reset_Tiempo))
{
    indicador = 0;
    pwm.PulseWM(frec_pwm,dir_mvto);
}
else if(Tiempo >= (2*ajuste_uSeg + reset_Tiempo) && Tiempo < (periodo + 2*ajuste_uSeg + reset_Tiempo))
{
    indicador = 1;
    pwm.PulseWM(frec_pwm,amp_rampa);
}
else
{
    indicador = 0;
    pwm.PulseWM(frec_pwm,-dir_mvto);
}

if(Tiempo_Muestreo > sample_time)
{
    if((indicador == 1) && (dir_mvto == -1))
    {
        amp_rampa = amp_rampa + 0.005;
    }
    else if((indicador == 1) && (dir_mvto == 1))
    {
        amp_rampa = amp_rampa - 0.005;
    }
    else;

    velocidad = (posicion - posicion_p)/Tiempo_Muestreo;
    posicion_p = posicion;
    gDato.InsertaDato(amp_rampa);
    gDato2.InsertaDato(velocidad);
    gDato3.InsertaDato(posicion);
    gDato4.InsertaDato(Tiempo);
    auxTiempo = auxTiempo + Tiempo_Muestreo;
    Tiempo_Muestreo = 0;
}
else;

```

```

    }
}

inline void GuardaRampa()
{
    gDato.GuardaDato("AnchoDePulso_Rampa.txt");
    gDato2.GuardaDato("Velocidad_Rampa.txt");
    gDato3.GuardaDato("Posicion_Rampa.txt");
    gDato4.GuardaDato("Tiempo_Rampa.txt");

}

inline void TerminaRespuesta()
{
    pwm.PulseWM(frecuencia_pwm,0);
}
};

```

#### 8. Clase CControl

```

using namespace std;
class CControl
{
    friend class CMicroTiempo;
    friend class CPwm;
    friend class CEncoder;
    friend class CGuardaDatos;
private:
    CMicroTiempo uTiempo;
    CPwm pwm;
    CEncoder encoder;
    CGuardaDatos gDato, gDato2, gDato3, gDato4;
    double ajuste_uSeg, ajuste_cuenta_rad;
    double ek0, ek1, W_salida, F_pwm, eak0, eak1, Ka;
    double Kp, Ki, uk1, uk0, Tiempo, Ksa;
    double posicion0, posicion1, Usat, Ulin;
    double Tiempo_Muestreo, auxTiempo, reset_Tiempo, auxDelta, ApVm;
public:
    inline CControl()
    {
        ajuste_uSeg = 1000000;
        ajuste_cuenta_rad = 0.015707963;
    }
};

```

```

// ajuste_cuenta_rad = 0.00076699;
ek0 = 0;
ek1 = 0;
eak0 = 0;
eak1 = 0;
Ka = 2;
Usat = 0;
Ulin = 0;
W_salida = 0;
Kp = 3;
Ki = 166.6;
Ksa = 0.035211267;
uk0 = 0;
uk1 = 0;
Tiempo = Tiempo_Muestreo = auxTiempo = auxDelta = 0;
uTiempo.Reset_TiempoAbsoluto();
posicion1 = encoder.InicializaCuentas()*ajuste_cuenta_rad;
posicion0 = encoder.InicializaCuentas()*ajuste_cuenta_rad;
ApVm = 0;
}

inline ~CControl()
{

}

inline void IniciaControl()
{
gDato.VaciaDato();
gDato2.VaciaDato();
gDato3.VaciaDato();
gDato4.VaciaDato();
ek0 = 0;
ek1 = 0;
eak0 = 0;
eak1 = 0;
Ka = 2;
Usat = 0;
Ulin = 0;
W_salida = 0;
Kp = 3;
Ki = 366.6;
Ksa = 0.035211267;

```

```

uk0 = 0;
uk1 = 0;
Tiempo = Tiempo_Muestreo = auxTiempo = auxDelta = 0;
uTiempo.Reset_TiempoAbsoluto();
posicion1 = encoder.InicializaCuentas()*ajuste_cuenta_rad;
posicion0 = encoder.InicializaCuentas()*ajuste_cuenta_rad;
ApVm = 0;
}

inline void ControlEscalon(double W_referencia, double delta, double t_prueba)
{
gDato.InsertaDato(W_salida);
gDato2.InsertaDato(Tiempo/ajuste_uSeg);
gDato3.InsertaDato(posicion1);
gDato4.InsertaDato(ApVm);
F_pwm = 1/delta;
auxDelta = ajuste_uSeg*delta;
t_prueba = ajuste_uSeg*t_prueba;
ek1 = W_referencia-W_salida;
uk1 = uk0 + (Kp + delta*Ki)*ek1 - Kp*ek0;
ApVm = uk1*Ksa/5.0;
if(ApVm >= 0 && ApVm <= 1)
    ApVm = ApVm;
else if(ApVm > 1)
    ApVm = 1;
else if(ApVm < 0 && ApVm >= -1)
    ApVm = ApVm;
else
    ApVm = -1;
ek0 = ek1;
uk0 = uk1;
reset_Tiempo = uTiempo.TiempoAbsoluto();
while(Tiempo < t_prueba + reset_Tiempo)
{
pwm.PulseWM(F_pwm,ApVm);
posicion1 = encoder.CalculaCuentas()*ajuste_cuenta_rad;
Tiempo = uTiempo.TiempoAbsoluto();
Tiempo_Muestreo = Tiempo - auxTiempo;
if(Tiempo_Muestreo > auxDelta)
{
W_salida = ajuste_uSeg*(posicion1 - posicion0)/Tiempo_Muestreo;
posicion0 = posicion1;
ek1 = W_referencia-W_salida;
}
}
}

```

```

//      uk1 = Kp*ek1+(delta*Ki-Kp)*ek0+uk0;
      uk1 = uk0 + (Kp + delta*Ki)*ek1 - Kp*ek0;
      ApVm = uk1*Ksa/5.0;
      if(ApVm >= 0 && ApVm <= 1)
          ApVm = ApVm;
      else if(ApVm > 1)
          ApVm = 1;
      else if(ApVm < 0 && ApVm >= -1)
          ApVm = ApVm;
      else
          ApVm = -1;
      ek0 = ek1;
      uk0 = uk1;
      gDato.InsertaDato(W_salida);
      gDato2.InsertaDato(Tiempo/ajuste_uSeg);
      gDato3.InsertaDato(posicion1);
      gDato4.InsertaDato(ApVm);
      auxTiempo = auxTiempo + Tiempo_Muestreo;
      Tiempo_Muestreo = 0;
    }
    else;
  }
}

```

```

inline void ControlEscalon_AWF(double W_referencia, double delta, double t_prueba)

```

```

{
  gDato.InsertaDato(W_salida);
  gDato2.InsertaDato(Tiempo/ajuste_uSeg);
  gDato3.InsertaDato(posicion1);
  gDato4.InsertaDato(ApVm);
  F_pwm = 1/delta;
  auxDelta = ajuste_uSeg*delta;
  t_prueba = ajuste_uSeg*t_prueba;
  ek1 = W_referencia-W_salida;
  eak1 = Usat - Ulin;
//      uk1 = uk0 + (Kp + delta*Ki)*ek1 - Kp*ek0;
  uk1 = uk0 + eak0*((Ka*Ki*delta - Kp)/Kp) + ek0*(Ki*delta - Kp) + Kp*ek1 + eak1;
  Ulin = uk1;
  ApVm = uk1*Ksa/5.0;
  if(ApVm >= 0 && ApVm <= 1)
  {
    ApVm = ApVm;
    Usat = Ulin;
  }
}

```

```

    }
else if(ApVm > 1)
{
    ApVm = 1;
    Usat = ApVm*5.0/Ksa;
}
else if(ApVm < 0 && ApVm >= -1)
{
    ApVm = ApVm;
    Usat = Ulin;
}
else
{
    ApVm = -1;
    Usat = ApVm*5.0/Ksa;
}
ek0 = ek1;
eak0 = eak1;
uk0 = uk1;
reset_Tiempo = uTiempo.TiempoAbsoluto();
while(Tiempo < t_prueba + reset_Tiempo)
{
    pwm.PulseWM(F_pwm,ApVm);
    posicion1 = encoder.CalculaCuentas()*ajuste_cuenta_rad;
    Tiempo = uTiempo.TiempoAbsoluto();
    Tiempo_Muestreo = Tiempo - auxTiempo;
    if(Tiempo_Muestreo > auxDelta)
    {
        W_salida = ajuste_uSeg*(posicion1 - posicion0)/Tiempo_Muestreo;
        posicion0 = posicion1;
        ek1 = W_referencia-W_salida;
        eak1 = Usat - Ulin;
//        uk1 = uk0 + (Kp + delta*Ki)*ek1 - Kp*ek0;
        uk1 = uk0 + eak0*((Ka*Ki*delta - Kp)/Kp) + ek0*(Ki*delta - Kp) + Kp*ek1 + eak1;
        Ulin = uk1;
        ApVm = uk1*Ksa/5.0;
        if(ApVm >= 0 && ApVm <= 1)
        {
            ApVm = ApVm;
            Usat = Ulin;
        }
        else if(ApVm > 1)
        {

```

```

        ApVm = 1;
        Usat = ApVm*5.0/Ksa;
    }
    else if(ApVm < 0 && ApVm >= -1)
    {
        ApVm = ApVm;
        Usat = Ulin;
    }
    else
    {
        ApVm = -1;
        Usat = ApVm*5.0/Ksa;
    }
    ek0 = ek1;
    uk0 = uk1;
    eak0 = eak1;
    gDato.InsertaDato(W_salida);
    gDato2.InsertaDato(Tiempo/ajuste_uSeg);
    gDato3.InsertaDato(posicion1);
    gDato4.InsertaDato(ApVm);
    auxTiempo = auxTiempo + Tiempo_Muestreo;
    Tiempo_Muestreo = 0;
    }
    else;
    }
}

inline void GuardaControl()
{
    pwm.PulseWM(F_pwm,0);
    gDato.GuardaDato("Velocidad_Controlada.txt");
    gDato2.GuardaDato("Tiempo.txt");
    gDato3.GuardaDato("Posicion.txt");
    gDato4.GuardaDato("Control.txt");
}
};

```