

Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería

Centro de Investigación en Tecnologías de Información y Sistemas

*“Control de velocidad de motores de cd con propósitos de
bombeo en sistemas hidráulicos”*

T E S I S

Para obtener el título de Licenciado en Sistemas Computacionales

Presenta:

Meneses Sánchez José Guadalupe

Director de Tesis:

Dr. Omar Arturo Domínguez Ramírez

Co-director de Tesis:

**Ing. Silvionel Vite Medécigo
M. en C. Mariano Javier Pozas Cárdenas**

Pachuca de Soto a 28 de Abril de 2006

ÍNDICE	Página
Capítulo 1: Introducción.	1
1.1. Antecedentes.	2
1.2. Justificación.	3
1.3. Planteamiento del problema.	4
1.4. Solución propuesta.	4
1.5. Contribución del trabajo de investigación.	4
1.6. Descripción del contenido de la tesis.	4
 Capítulo 2. Control de velocidad de motores de cd.	 6
2.1. Antecedentes.	7
2.2. Configuraciones de controladores de velocidad.	7
2.2.1. Control por tensión variable.	8
2.2.2. Control de velocidad por par Ward Leonard.	8
2.2.3. Control por retroalimentación.	9
2.2.4. Control de velocidad por excitación Shunt.	10
2.2.5. Control de velocidad por tensión de rotor.	10
2.2.6. Control de velocidad de un motor con excitación serie.	11
2.2.7. Circuitos de contactores.	11
 Capítulo 3. El motor de cd con retroalimentación taquimétrica.	 13
3.1. Principio de funcionamiento de un motor de cd.	14
3.2. Circuito equivalente aproximado.	17
3.3. Obtención del modelo matemático de un motor de cd.	17
3.4. Principio de funcionamiento del generador taquimétrico.	18
3.5. Descripción de la caracterización experimental de un generador taquimétrico.	19
3.6. Aplicaciones.	20
 Capítulo 4. Características del motor de cd experimental.	 22
4.1. Especificaciones técnicas.	23
4.2. Circuito equivalente aproximado del motor de cd.	24
4.3. Modelo matemático del motor de cd.	25
4.4. Simulaciones digitales del motor de cd en lazo abierto.	27
4.5. Caracterización experimental del generador taquimétrico.	27
4.6. Modelo matemático del generador taquimétrico.	28
4.7. Simulaciones digitales del generador taquimétrico.	30
4.8. Descripción de la obtención del modelo matemático de un motor de cd con generador taquimétrico.	31
 Capítulo 5. Control de velocidad de motores de cd.	 33
5.1. Esquema clásico de control.	34
5.2. Estrategias clásicas de control.	34
5.2.1. Acciones clásicas de control y controles automáticos industriales.	34
5.2.2. Clasificación de los controles automáticos.	34

5.2.3. Elementos de control automáticos industriales.	35
5.2.4. Controles autoactuantes.	36
5.2.5. Acciones de control.	36
5.2.6. Acción de dos posiciones o de si-no.	37
5.2.7. Acción de control proporcional.	39
5.2.8. Acción de control integral.	39
5.2.9. Acción de control proporcional e integral.	40
5.2.10. Acción de control proporcional y derivativo.	41
5.2.11. Acción de control proporcional y derivativo e integral.	42
5.3. Control no lineal: PID NL.	44
5.4. Control PX.	46
5.5. Simulaciones digitales.	47
Capítulo 6. Integración de la plataforma experimental.	51
6.1. Diagrama de bloques del sistema.	52
6.2. Sistema de adquisición de datos.	54
6.3. Interfaz electrónica y acondicionamiento de señales.	54
6.4. Interfaz gráfica de usuario.	55
6.5. Segmentos del código fuente.	56
6.6. Control PD con aceleración estimada y regulación.	57
6.7. Control PI con regulación.	59
6.8. Control PD con aceleración estimada y regulación basada en seguimiento.	62
6.9. Control PI con regulación basada en seguimiento.	64
6.10. Control PID con aceleración estimada y regulación.	67
6.11. Control PID con regulación basada en seguimiento.	70
6.12. Control PID NO LINEAL con aceleración estimada y regulación.	73
6.13. Control PID NO LINEAL con regulación basada en seguimiento.	76
6.14. Control PX con aceleración estimada y regulación.	80
6.15. Control PX basada en seguimiento.	82
6.16. Gráficas de los controles aplicando perturbación.	85
6.17. Control PD con perturbación.	86
6.18. Control PI con perturbación.	88
6.19. Control PID con perturbación.	91
6.20. Control PID NL con perturbación.	94
6.21. Control PX con perturbación.	97
6.22. Comparación final de las gráficas de velocidad.	99
Capítulo 7. Perspectivas y Conclusiones.	103
7.1. Perspectivas.	104
7.2. Conclusiones.	104
Referencias.	106
Glosario de términos.	107
Apéndice 1.	109
Apéndice 2.	111
Apéndice 3.	153

ÍNDICE DE FIGURAS

Página

Capítulo 1

Figura 1.1: Esquema básico de un circuito convencional. 2

Capítulo 2

Figura 2.1: Arrancador de contactor. 11

Capítulo 3

Figura 3.1: Líneas de campo magnético y sentido del flujo de la corriente. 15

Figura 3.2: Torque y sentido de giro. 15

Figura 3.3: Representación del motor de cd. 15

Figura 3.4: Conductor con respecto al eje de rotación. 16

Figura 3.5: Cancelación del giro del motor. 16

Figura 3.6: a).-Diagrama esquemático del controlado por el inducida b).-Diagrama de bloques. 17

Capitulo 4

Figura 4.1: Plataforma de trabajo. 23

Figura 4.2: Motor de cd experimental. 23

Figura 4.3: Diagrama esquemático del motor de cd con retroalimentación taquimétrica. 24

Figura 4.4: Comportamiento de la velocidad e intensidad de corriente del motor. 27

Figura 4.5: Esquema de medición para la toma de muestras. 27

Figura 4.6: Gráfica de la regresión lineal aproximada. 30

Capitulo 5

Figura 5.1 Esquema de control. 34

Figura 5.2: Diagrama de bloques de un control automático. 35

Figura 5.3: Control automático. 36

Figura 5.4: Diagrama de bloques de control si-no. 38

Figura 5.5: Sistema de control de nivel de liquido. 38

Figura 5.6: Gráfica del sistema de control. 38

Figura 5.7: Diagrama de bloques de controles PD y PI. 39

Figura 5.8: Diagrama del bloques de control PI, gráficas de entrada y salida. 41

Figura 5.9: Diagrama de bloques del control PD, gráficas de entrada y salida. 41

Figura 5.10: Diagrama de bloques de controles PID, gráficas de entrada y salida. 43

Figura 5.11: Gráfica de la función signo. 45

Figura 5.12: Gráfica de la tangente hiperbólica variando a β . 46

Figura 5.13: Diagrama de bloques integral del sistema de control PX. 46

Figura 5.14: Diagrama de bloques del sistema de control PX dependiente exclusivamente del estado de velocidad. 47

Figura 5.15: Gráfica de la velocidad del motor usando control PD, ampliación de la gráfica. 48

Figura 5.16: Gráfica de la velocidad del motor usando control PI, ampliación de la gráfica. 48

Figura 5.17: Gráfica de la velocidad del motor usando control PID, ampliación de la gráfica. 49

Figura 5.18: Gráfica de la velocidad del motor usando control PX, ampliación de la gráfica. 49

Capítulo 6

Figura 6.1: Diagrama de bloques del sistema.	52
Figura 6.2: Tarjeta de adquisición de datos SCB 100 de National Instruments.	54
Figura 6.3: Interfaz electrónica.	54
Figura 6.4: Interfaz de usuario creada en Delphi 5.	55
Figura 6.5: Gráfica de velocidad, ampliación de la señal, de control PD.	57
Figura 6.6: Gráfica de la señal de control, ampliación de la señal, de control PD.	57
Figura 6.7: Gráfica de error de velocidad, ampliación de la señal, de control PD.	58
Figura 6.8: Gráfica de la derivada del error, ampliación de la señal, de control PD.	58
Figura 6.9: Gráfica de los retardos del sistema.	59
Figura 6.10: Gráfica de velocidad, ampliación de la señal, de control PI.	59
Figura 6.11: Gráfica de señal de control, ampliación de la señal, de control PI.	60
Figura 6.12: Gráfica del error de velocidad, ampliación de la señal, de control PI.	60
Figura 6.13: Gráfica del índice de integración, ampliación de la señal, de control PD.	61
Figura 6.14: Gráfica de los retardos del sistema.	61
Figura 6.15: Gráfica de velocidad, ampliación de la señal de control PD con seguimiento.	62
Figura 6.16: Gráfica de la señal de control, del control PD.	62
Figura 6.17: Gráfica del error basada en seguimiento, de control PD.	63
Figura 6.18: Gráfica de la derivada del error basada en seguimiento, de control PD.	63
Figura 6.19: Gráfica de los retardos del sistema.	64
Figura 6.20: Gráfica de velocidad basada en seguimiento, de control PI.	64
Figura 6.21: Gráfica de la señal de control, del control PI con seguimiento.	65
Figura 6.22: Gráfica del error, de control PI con seguimiento.	65
Figura 6.23: Gráfica del índice de integración, de control PI con seguimiento.	66
Figura 6.24: Gráfica de los retardos del sistema.	66
Figura 6.25: Gráfica de velocidad, ampliación a 135 rpm, de control PID.	67
Figura 6.26: Gráfica de la señal de control, del control PID.	68
Figura 6.27: Gráfica del error de velocidad, del control PI.	68
Figura 6.28: Gráfica de la derivada del error, del control PI.	69
Figura 6.29: Gráfica del índice de integración, del control PI.	69
Figura 6.30: Gráfica de los retardos del sistema.	70
Figura 6.31: Gráfica de velocidad, aplicación a 135 rpm, de control PID con seguimiento.	71
Figura 6.32: Gráfica de la señal de control; control PID con seguimiento.	71
Figura 6.33: Gráfica del error de velocidad; control PID con seguimiento.	72
Figura 3.34: Gráfica del índice de integración; control PID con seguimiento.	72
Figura 6.35: Gráfica de los retardos del sistema.	73
Figura 6.36: Gráfica de velocidad, ampliación a 135 rpm, de control PID NL.	73
Figura 6.37: Gráfica de la señal de control; control PID NL.	74
Figura 6.38: Gráfica del error de velocidad; control PID NL.	74
Figura 6.39: Gráfica de la derivada del error; control PID NL.	75
Figura 6.40: Gráfica del índice de integración; control PID NL.	75
Figura 6.41: Gráfica de los retardos del sistema.	76

Figura 6.42: Gráfica de velocidad, ampliación a 135 rpm, de control PID NL con seguimiento.	77
Figura 6.43: Gráfica de la señal de control; control PID NL con seguimiento.	77
Figura 6.44: Gráfica del error de velocidad; control PID NL con seguimiento.	78
Figura 6.45: Gráfica de la derivada del error; control PID NL con seguimiento.	78
Figura 6.46: Gráfica del índice de integración; control PID NL con seguimiento.	79
Figura 6.47: Gráfica de los retardos del sistema.	79
Figura 6.48: Gráfica de velocidad, ampliación a 135 rpm, de control PX.	80
Figura 6.49: Gráfica de la señal de control del control PX, ampliación de la gráfica.	80
Figura 6.50: Gráfica del error del control PX, ampliación de la gráfica.	81
Figura 6.51: Gráfica de la derivada del error del control PX, ampliación de la gráfica.	81
Figura 6.52: Gráfica de los retardos del sistema.	82
Figura 6.53: Gráfica de velocidad, ampliación a 135 rpm, del control PX con seguimiento.	83
Figura 6.54: Gráfica de la señal de control del control PX con seguimiento.	83
Figura 6.55: Gráfica del error de velocidad del control PX con seguimiento.	84
Figura 6.56: Gráfica de la derivada del error del control PX con seguimiento.	84
Figura 6.57: Gráfica de los retardos del sistema.	85
Figura 6.58: Gráfica de la velocidad, control PD con perturbaciones.	86
Figura 6.59: Gráfica de la señal de control, control PD con perturbaciones.	86
Figura 6.60: Gráfica del error, control PD con perturbaciones.	87
Figura 6.61: Gráfica de la derivada del error, control PD con perturbaciones.	87
Figura 6.62: Gráfica de los retardos del sistema, control PD con perturbaciones	88
Figura 6.63: Gráfica de la velocidad, control PI con perturbaciones.	88
Figura 6.64: Gráfica de la señal de control, control PI con perturbaciones	89
Figura 6.65: Gráfica del error, control PI con perturbaciones.	89
Figura 6.66: Gráfica del índice de integración, control PI con perturbaciones.	90
Figura 6.67: Gráfica de los retardos del sistema, control PI con perturbaciones.	90
Figura 6.68: Gráfica de la velocidad, control PID con perturbaciones.	91
Figura 6.69: Gráfica de la señal de control, control PID con perturbaciones.	91
Figura 6.70: Gráfica del error, control PID con perturbaciones.	92
Figura 6.71: Gráfica de la derivada del error, control PID con perturbaciones.	92
Figura 6.72: Gráfica del índice de integración, control PID con perturbaciones.	93
Figura 6.73: Gráfica de los retardos del sistema, control PID con perturbaciones.	93
Figura 6.74: Gráfica de la velocidad, control PID NL con perturbaciones.	94
Figura 6.75: Gráfica de la señal de control, control PID NL con perturbaciones.	94
Figura 6.76: Gráfica del error, control PID NL con perturbaciones.	95
Figura 6.77: Gráfica de la derivada del error, control PID NL con perturbaciones.	95
Figura 6.78: Gráfica del índice de integración, control PID NL con perturbaciones.	96
Figura 6.79: Gráfica de los retardos del sistema, control PID NL con perturbaciones.	96
Figura 6.80: Gráfica de la velocidad, control PX con perturbaciones.	97
Figura 6.81: Gráfica de la señal de control, control PX con perturbaciones.	97
Figura 6.82: Gráfica del error, control PX con perturbaciones.	98
Figura 6.83: Gráfica de los retardos del sistema, control PX con perturbaciones.	98

ÍNDICE DE TABLAS.

Página

Capítulo 4.

Tabla 1: Datos obtenidos en laboratorio.	28
---	----

Apéndice 2.

Tabla de referencia de las terminales de la tarjeta de adquisición de datos SCB-100.	111
--	-----

LISTA DE UNIDADES Y SÍMBOLOS.

Variable.	Unidad.	Término.
ω	rad/seg o rpm	Velocidad angular
i_a	amperes	Intensidad de corriente de armadura
R_a	Ohms.	Resistencia del devanado del inducido
L_a	Henrios.	Inductancia del devanado del inducido en henrios.
i_f	Ampers.	Corriente de campo
e_a	Volts.	Tensión aplicada a la armadura.
e_b	Volts.	Fuerza contra- electromotriz.
θ_m	Radianes.	Desplazamiento angular del eje del motor.
T_m	libras-pie.	Par desarrollado por el motor.
J_m	slug-pie ²	Momento de inercia equivalente del motor y carga con referencia al eje del motor
B_m	libras-pie/rad/seg	Coefficiente de fricción viscosa equivalente del motor y carga referido a eje del motor

CAPÍTULO I

INTRODUCCIÓN

1.1. Antecedentes.

Los controladores se emplean en muchas industrias incluyendo la de papel, farmacéuticos, niquelado, electrónica, bebidas y generación de potencia. Dos aplicaciones muy comunes son el lavado de botellas y aplicaciones en torres de enfriamiento.

Los sistemas hidráulicos a base de aceite fueron desarrollados en la década de los años veinte y desde aquellos tiempos, han ido perfeccionándose e incorporando nuevas tecnologías de control. En la figura 1.1 se muestra el esquema básico de un control hidráulico.

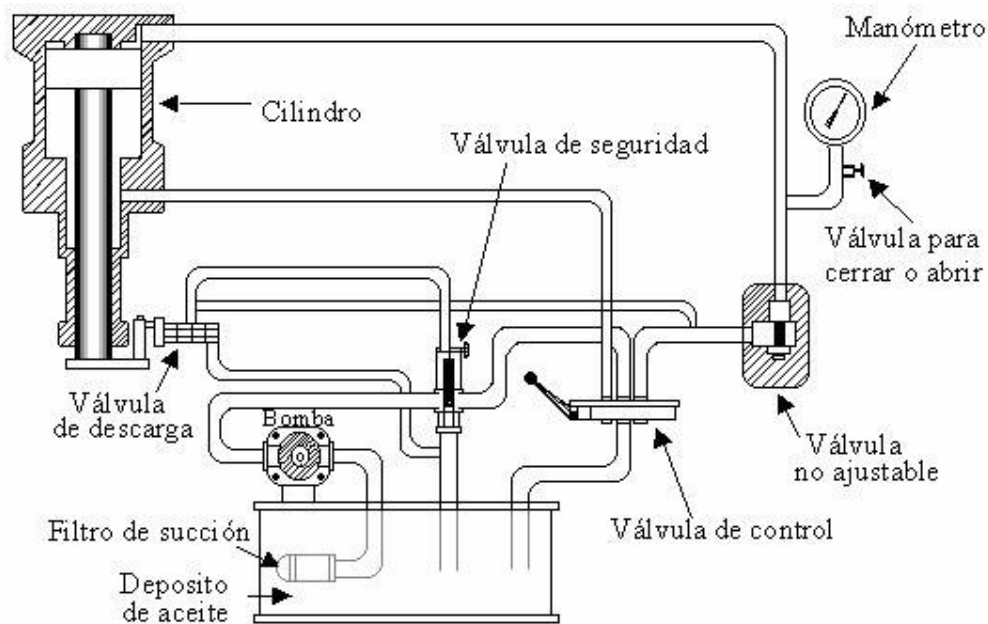


Figura 1.1: Esquema básico de un circuito convencional.

Esencialmente un sistema hidráulico comprende la producción, transmisión y control de energía hidráulica, utilizando aceite como fluido. Se suministra energía al aceite, por lo general en forma de presión, mediante bombas, y se conduce a través de tuberías hasta motores o cilindros hidráulicos que se encargan de transformar la energía en trabajo. Las presiones utilizadas son relativamente elevadas, desde algunas decenas a varios cientos de kg/cm^2 .

Así pues, las bombas aspiran el aceite de un depósito, previamente hay que introducir un filtro que elimina posibles impurezas. Las bombas son generalmente de tipo rotativo, a base de engranajes, pero también pueden ser de pistón o incluso superiores a los $75 \text{ kg}/\text{cm}^2$. Las bombas son accionadas directamente por motores eléctricos o bien van acopladas al motor principal. En el circuito se incorporan válvulas de seguridad, de dirección y de control de caudal, tuberías y finalmente el motor o cilindro hidráulico.

Los cilindros hidráulicos constituyen el sistema de accionamiento más sencillo, pero cualquier bomba de las anteriormente descritas puede transformarse en motor con sólo suministrarles fluido a alta presión. La diferencia entre unos y otros es que los cilindros efectúan un desplazamiento lineal y los motores o bombas convertidas generan un movimiento rotativo.

Los controles utilizados para el funcionamiento del sistema pueden ser de tipo manual (discontinuo o variable) y de tipo automático (de bucle abierto, cerrado o retroalimentación).

Todos estos tipos de control no son tan eficaces al tratar de compensar dinámicas no lineales, por ejemplo tienen pérdidas de presión por aceleración y desaceleración en la línea de dosificación.

Actualmente, los sistemas de control de procesos son una combinación de computadoras (incluyendo las personales), estaciones de trabajo, software gráfico y diversos algoritmos de control, tarjetas de adquisición de señales digitalizadas de propósito general y específico, redes de comunicación, etc. [20]

1.2. Justificación.

El control que se implementa, control propuesto PX, no requiere de la derivada e integral del estado para regular a una velocidad de referencia a la motobomba experimental, tiene muchas ventajas que pueden manifestarse en:

- El caudal se mantiene constante, independientemente del cambio de las condiciones de funcionamiento de la instalación.
- Ajusta automáticamente las pérdidas de carga necesarias para mantener el caudal constante, incluso con presiones variables.
- Se conocen de antemano los caudales reales de trabajo para así poder seleccionar las bombas en su punto óptimo de rendimiento.
- Necesitan menos mantenimiento: No necesita ninguna intervención de ajuste ni mediciones en obra.

Estas características no podrían mantenerse con un control clásico, ya que no presentan la robustez en condiciones de perturbación, tales como el golpe de ariete en los sistemas hidráulicos, entre otras.

La ventaja de este control es que puede utilizarse en infinidad de aplicaciones. Por ejemplo:

- La elección de una bomba depende de la velocidad de giro del motor que la acciona.
- Si los niveles de los depósitos fluctúan, es difícil alcanzar un rendimiento alto para todos los modos de funcionamiento. En casos extremos a veces se utiliza un motor con velocidad variable.

1.3. Planteamiento del problema.

La implementación de un control dependiente de la derivada del estado y de su integral implica un costo adicional en la implementación, requiere emplear la derivada del error para control de velocidad, esto implica estimar la derivada de la velocidad instantánea, es decir la aceleración, siendo una señal ruidosa, situación que complica la acción de control, la sintonización de las ganancias de los controladores, etc.

1.4. Solución propuesta.

Diseñar e implementar un control de regulación basada en seguimiento que permita hacer regulación global sin medición de la derivada del estado.

1.5. Contribución del trabajo de investigación.

Diseño e implementación de un control de velocidad de motores de cd y motobombas, que permita la regulación global a partir de seguimiento de trayectorias, sin la medición de la derivada del estado.

1.6. Descripción del contenido de la tesis.

En el capítulo 2 se tratan antecedentes de los distintos tipos de arrancadores y controladores de motores eléctricos, configuraciones de controladores de velocidad, control por tensión variable, control de velocidad por par Ward Leonard, control por retroalimentación, control de velocidad por excitación Shunt, control de velocidad por tensión de rotor, control de velocidad de un motor con excitación serie, circuitos de contactores.

El capítulo 3 está dedicado a la descripción del motor de cd con retroalimentación taquimétrica, al principio de funcionamiento, al circuito equivalente aproximado, la obtención del modelo matemático, al principio de funcionamiento de generador taquimétrico, a la descripción de la caracterización experimental de un motor de cd con generador taquimétrico y contiene también algunas aplicaciones.

En el capítulo 4 se presentan la características del motor, la estación de trabajo, las especificaciones técnicas del motor de cd, el circuito equivalente aproximado, el modelo matemático del motor de cd, las simulaciones digitales del motor, la caracterización experimental del generador taquimétrico, al modelo matemático del generador taquimétrico, la simulación digital del generador taquimétrico, y la descripción de la obtención del modelo matemático de un motor de cd con generador taquimétrico.

En al capítulo 5 se muestra el esquema clásico de control, las estrategias clásicas de control, las acciones clásicas de control y controles automáticos industriales, la clasificación de los controles automáticos, los elementos de control automáticos industriales, los controles autoactuantes, las acciones de control, la acción de control de dos posiciones o acción de control si-no, la acción de control proporcional, la acción de control integral, la acción de control proporcional e integral, la acción de control proporcional y derivativo, la acción de control proporcional y derivativo e integral, el control no lineal: PID NL, el nuevo control implementado, control PX, y las simulaciones digitales.

En el capítulo 6 se presenta la integración de la plataforma y las gráficas de los experimentos para cada control que demuestran el comportamiento real del motor, el diagrama de bloques del sistema, la tarjeta de adquisición de datos, la interfaz electrónica, la interfaz gráfica de usuario, segmentos de código de los programas en Delphi 5 y los experimentos correspondientes a las acciones de control PD, PI, PID, PID NO LINEAL y PX con aceleración estimada, regulación y seguimiento.

En el capítulo 7 se presentan las perspectivas y conclusiones de la investigación.

Por último se encuentran referencias, glosario de términos y apéndices, los cuales contienen lo programas realizados en Delphi 5 y Matlab 6, así como también una tabla de la tarjeta de adquisición de datos.

CAPÍTULO 2

CONTROL DE VELOCIDAD DE MOTORES DE CD

INTRODUCCIÓN

En el presente capítulo se tratan los distintos tipos de arrancadores y controladores de motores eléctricos.

- Antecedentes.
- Configuraciones de controladores de velocidad.
- Control por tensión variable.
- Control de velocidad por par Ward Leonard.
- Control por retroalimentación.
- Control de velocidad por excitación Shunt.
- Control de velocidad por tensión de rotor.
- Control de velocidad de un motor con excitación serie.
- Circuitos de contactores.

2.1. Antecedentes.

Los motores eléctricos proporcionan una de las fuentes principales de energía para impulsar las modernas máquinas herramientas u otro tipo de equipo industrial. El motor ha llegado a relacionarse tan íntimamente con la impulsión de los elementos de las máquinas que en la mayoría de los casos, se incluye como parte integral del diseño de ella. Esta relación de motor y máquina a través de la impulsión directa ha enfocado la atención en el diseño, construcción, instalación y mantenimiento del equipo para controlar el motor.

El término control del motor en la comunicación moderna, se refiere al control de la velocidad e inversión de la rotación, métodos de aceleración y desaceleración, y muchas otras funciones de los controladores del motor, que se agregan a los conceptos más antiguos del simplemente arranque y frenado de motores.

Si bien varios libros sobre electricidad y manuales de servicio han descrito diferentes tipos de controladores, ha sido fácil obtener esta información de una sola fuente de referencia, este documento proporciona una fuente conveniente de información técnica y práctica para una comprensión total de la teoría y práctica de control de velocidad para el motor eléctrico.

2.2. Configuraciones de controladores de velocidad.

Los motores de velocidad variable son apropiados para muchas aplicaciones industriales. Cuando hemos considerado la regulación de velocidad por control de tensión, hemos encontrado muchas variaciones en los esquemas, en los que una posible ventaja para una aplicación podría representar una desventaja para otra. Es esencial para un especialista considerar las ventajas relativas de cada sistema, así como las exigencias de cada aplicación particular, antes de proceder a una elección. Estas aplicaciones se extienden desde los grandes motores para molinos de rodillos o equipos de ventilación de misma, a los motores fraccionarios. Entre estos límites están los motores para cepilladoras y otras muchas máquinas herramientas, motores para ciertas grúas y ascensores de gran velocidad, excavadoras, satinadoras y empapeladoras. Otras aplicaciones pueden exigir una coordinación y regulación fina de par y/o de la velocidad de varios motores en una misma máquina o de todos los motores de un proceso particular, como los diferentes motores de una industria de papel o los de un tren de bandas en frío o en caliente, para metales férricos o no férricos.

Cuando se consideran máquinas de esta naturaleza, la cuestión esencial es si su costo queda justificado por el aumento de producción. Aunque el costo inicial de una instalación ideal pueda parecer a primera vista alto, debe ser comprobado con el ahorro conseguido

al aumentar la producción. Cada aplicación requiere una consideración especial para determinar las características mecánicas de la máquina, antes de empezar el estudio del aparellaje eléctrico. [16]

2.2.1. Control por tensión variable.

Dado el uso tan extendido de la corriente alterna, es necesario disponer de un equipo que la convierta en corriente continua, para poder explotar las flexibles características de los motores de corriente continua. Inicialmente se empleaban conjuntos convertidores rotativos motor- generador, pero el rápido desarrollo de los rectificadores estáticos han hecho desaparecer casi completamente los primeros. Los convertidores rotativos fueron sustituidos completamente por los de mercurio, pero éstos, a su vez, están dejando paso a los de silicio y germanio.

El empleo de motores de cd. de velocidad variable, está quedando relegado a procesos donde se requiere una potencia pequeño, comparada con la potencia total de la instalación. En tales condiciones, es raramente necesario instalar una línea de corriente continua a tensión constante a través de la fabrica, pudiéndose lograr una alimentación de los motores más eficiente, mediante equipos alimentadores de continua para cada motor o grupo de motores.

El control de estos motores se puede simplificar mediante el empleo de una excitatriz para la alimentación del campo inductor del generador, mediante rectificadores controlados por rejilla, o con un amplificador magnético.[9]

2.2.2. Control de velocidad por par Ward Leonard.

El generador principal es movido por un motor de corriente continua apropiado, de un tipo que depende de la línea de potencia disponible. Cuando se emplea un motor síncrono, se obtiene una velocidad constante, con la ventaja adicional de que sirve para corregir el factor de potencia cuando funciona en sobre excitación. La excitación del generador se alimenta por medio de una excitatriz independiente que, según los casos, se puede emplear también para alimentar la excitación y se montan en la misma bancada.

El rotor del motor de cd se conecta directamente a los terminales del generador, evitando así el empleo de un contactor grande y del arrancador. El arranque se efectúa ajustando el regulador de la excitatriz para elevar la tensión aplicada al motor. Por medio de un regulador Shunt aplicado al campo de la excitatriz o mediante un regulador potenciométrico en la alimentación del campo del generador principal, a partir del campo y, por lo tanto la tensión de salida desde cero al valor máximo. [16]

2.2.3. Control por retroalimentación.

Sin este sistema de control sería difícil obtener el grado de exactitud requerido por muchos procesos industriales. Lo característico del control por retroalimentación es que, por medio de un muestreo de la salida, permite mantener la cantidad regulada entre unos límites muy estrechos.

Lo principal de un sistema con retroalimentación es que logra el control de la salida entre límites muy estrechos por medio de una señal de entrada pequeña. Esta salida es detectada por algún dispositivo que emite una señal de salida en forma eléctrica, y que se compara con la magnitud de referencia. Cualquier diferencia entre las dos, da lugar a una señal de error que inicia el funcionamiento del mecanismo de regulación del sistema. El elemento detector empleado para medir las desviaciones a partir del funcionamiento deseado, tiene invariablemente una potencia suficiente que pueda excitar el mecanismo regulador, la señal se lleva a un amplificador que proporciona la potencia de salida requerida.

El grupo Ward Leonard, es equivalente a dos etapas amplificadoras, siendo la primera etapa la excitatriz y la segunda el generador. Esto se podría ver más claramente modificando las conexiones, y alimentando la excitación del motor a partir de una fuente de tensión constante, conectando la salida de la excitatriz al devanado de campo del generador por medio de un regulador Shunt. Así controlando algunos vatios del campo excitador de la excitatriz, se regula su salida, controlando, la excitación del generador, la tensión de salida del mismo y la velocidad del motor principal.

Cuando se estudia el control por retroalimentación, se debe considerar cuidadosamente la estabilidad propia del sistema, para evitar las oscilaciones. El tiempo que necesita un sistema para responder a un cambio de carga o de velocidad, depende de la constante de tiempo de las máquinas y de la inercia de la carga. Para asegurar una respuesta rápida, la constante de tiempo se debe mantener por debajo de un mínimo, lo que aumenta el riesgo de inestabilidad, particularmente cuando la amplificación entre la señal detectada y la carga es muy grande. Generalmente es deseable una velocidad de respuesta alta, por lo que se suele disponer un circuito que realimente los transitorios en oposición a los cambios que tiene lugar en el circuito excitador, y no haga el sistema lento. Esto se puede lograr tomando una tensión de referencia en el lazo del generador y llevándola al circuito estabilizador, con la salida conectada al amplificador para reducir la velocidad de respuesta del sistema a los cambios de carga repentinos. [16]

2.2.4. Control de velocidad por excitación Shunt.

Cuando un motor Shunt trabaja con excitación débil, la corriente del inducido produce un par motor proporcional a la reducción de la excitación, pero dicha corriente no puede sobrepasar la nominal debido a la limitada capacidad térmica de la máquina. Además como la velocidad es inversamente proporcional al campo excitador para las velocidades superiores de la velocidad mínima, el producto del par por la velocidad es constante. En otras palabras, si el campo de excitación de un motor Shunt se debilita la velocidad aumenta y el par se reduce, resultado que el motor desarrolla una potencia constante. Se puede obtener para un motor Shunt y sobre la velocidad mínima, una gama de velocidad de 3:1, dependiendo de la potencia del motor. En los motores de menos de 20 HP se puede obtener una gama de 4:1 o mayor; sin embargo, para potencias de más de 1000 HP, y aun con devanados compensadores, la gama quedaría probablemente limitada 2:1. Estas cifras no se pueden aplicar a los motores especiales de corriente continua para bobinadoras en las que se requiere una característica de potencia constante entre unos márgenes de velocidad que dependen del diámetro de las bobinas que han de manejar. La velocidad queda limitada por la máxima velocidad periférica que permite el inducido y por la buena conmutación del motor. Se alcanza un punto límite cuando la distorsión debida a la reacción de inducido, con campo excitador débil, es tan grande que se produce un gran chisporroteo.

Para obtener un par de arranque igual al nominal con la mínima corriente y una buena conmutación, es importante mantener el campo excitador al máximo. Por esta razón el regulador Shunt se debe enclavar con el arrancador para evitar que se pueda arrancar con campo débil.

Como un motor con control por devanado Shunt tiene una característica de potencia constante, el tamaño de la carcasa y el precio están basados sobre la potencia a la velocidad mínima. Aunque el control por devanado Shunt es muy eficiente para una gama limitada de velocidades, existen muchas máquinas que requieren un par constante, y para tales aplicaciones las dimensiones físicas y el precio de los motores pueden no ser aceptables. [17]

2.2.5. Control de velocidad por tensión de rotor.

Cuando un motor Shunt trabaja con excitación constante, la velocidad es aproximadamente proporcional a la tensión del inducido. Una reducción en dicha tensión, manteniendo constante la corriente de excitación, causa el descenso de la velocidad por debajo de la velocidad base. En tales condiciones, y debido a la reducida tensión que coexiste de inducido constante, el motor tiene una característica de par constante. El par desarrollado a tensión reducida queda limitado por la corriente máxima de inducido y los límites de conmutación del motor.

Como la potencia se reduce en proporción a la velocidad, la carcasa, el precio y el tamaño de un motor Shunt para trabajar con par constante, se basan en la potencia a velocidad máxima, lo que da lugar a un motor considerablemente más pequeño que el requerido cuando se emplea regulación de Shunt, para un mismo campo de velocidades.[16]

2.2.6. Control de velocidad de un motor con excitación serie.

La velocidad de un motor con excitación serie, para un par dado, se puede controlar con una resistencia en serie que reduzca la tensión aplicada a los devanados de inducido y de excitación y, por lo tanto, reduzca la velocidad del motor. El control de velocidad se puede efectuar también con una resistencia conectada en paralelo con la excitación o con el devanado del rotor. Cuando tal resistencia se conecta en paralelo con la excitación, la velocidad del motor aumenta, pero como el problema es normalmente reducir la velocidad del motor serie, el método no se usa ampliamente. Sin embargo, es corriente conectar una resistencia en paralelo con el inducido, además de una resistencia en serie con él. [9]

2.2.7. Circuitos de contactores.

Los equipos de control automático de motores basado en contactores emplean los métodos de arranque estudiado, pero como tiene la facilidad de la regulación mediante señales de baja energía, tienen una gran demanda, y por la sencillez de las operaciones de regulación se aplica a una gran variedad de casos. Además del simple arrancador directo accionado mediante un pulsador, existen circuitos de control asegurados mediante enclavamientos para regular los diversos motores de un proceso, pudiendo disponerse también de control automático de la velocidad y del control mediante servomecanismo para reducir el error a un mínimo. Estas características son esenciales actualmente para muchos equipos industriales. Uno de ellos son las máquinas diseñadas para trasladar piezas de un lugar a otro y ahorrar tiempo de proceso.

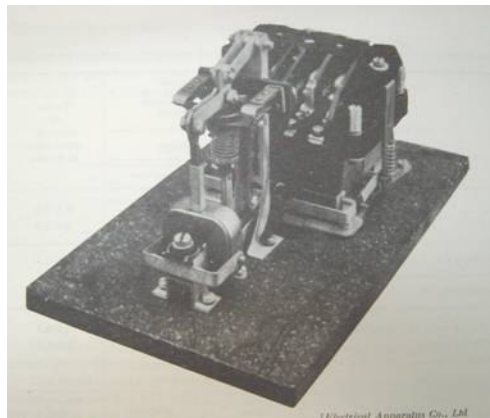


Figura 2.1: Arrancador de contactor.

La figura 2.1 muestra un arrancador de contactor. Los arrancadores de contactores consisten en un contactor que conecta y desconecta el motor de línea, y contactores de aceleración que actúan según una determinada secuencia y eliminan la resistencia de arranque que limita la corriente a un determinado valor y regula así la aceleración del motor. El funcionamiento del arrancador se puede iniciar mediante un pulsador montado en el panel o en una caja aproximada para pulsarlo con el dedo, o accionarlo mediante otros dispositivos sensibles a una corriente, tensión o tiempo. En el control automático existen casos en que es aconsejable iniciar la operación de regulación a partir de un destello luminoso, una elevación de temperatura, o la compresión de un gas o líquido.[17]

Conclusiones.

Los distintos tipos de controles que se recopilaron para este capítulo, nos demuestran las distintas aplicaciones de control, como son controles de velocidad, circuitos arrancadores, circuitos de cambio de dirección, controles de aceleración, controles de frenado, controles utilizados para convertir energía mecánica en energía hidráulica y permiten desplazar líquidos, conectar y desconectar el motor de línea, etc.

CAPÍTULO 3

EL MOTOR DE CD CON RETROALIMENTACIÓN TAQUIMÉTRICA

INTRODUCCIÓN

Este capítulo está dedicado a la descripción del motor de cd con retroalimentación taquimétrica:

- Principio de funcionamiento.
- Circuito equivalente aproximado.
- Obtención del modelo matemático.
- Principio de funcionamiento de generador taquimétrico.
- Descripción de la caracterización experimental de un motor de cd con generador taquimétrico.
- Aplicación.

3. El motor de cd con retroalimentación taquimétrica.

Los motores y generadores eléctricos, son un grupo de aparatos que se utilizan para convertir la energía mecánica en eléctrica, o a la inversa, con medios electromagnéticos. A una máquina que convierte la energía mecánica en eléctrica se le denomina generador, alternador o dínamo, y a una máquina que convierte la energía eléctrica en mecánica se le denomina motor. Dos principios físicos relacionados entre sí sirven de base al funcionamiento de los generadores y de los motores. El primero es el principio de la inducción descubierto por el científico e inventor británico Michael Faraday en 1831. Si un conductor se mueve a través de un campo magnético, o si está situado en las proximidades de un circuito de conducción fijo cuya intensidad puede variar se establece o se induce una corriente en el conductor. El principio opuesto a éste fue observado en 1820 por el físico francés André Marie Ampère. Si una corriente pasaba a través de un conductor dentro de un campo magnético, éste ejercía una fuerza mecánica sobre el conductor. [22]

La máquina dinamoeléctrica más sencilla es la dinamo de disco desarrollada por Faraday, que consiste en un disco de cobre que se monta de tal forma que la parte del disco que se encuentra entre el centro y el borde quede situada entre los polos de un imán de herradura. Cuando el disco gira, se induce una corriente entre el centro del disco y su borde debido a la acción del campo del imán. El disco puede fabricarse para funcionar como un motor mediante la aplicación de un voltaje entre el borde y el centro del disco, lo que hace que el disco gire gracias a la fuerza producida por la reacción magnética.

El campo magnético de un imán permanente es lo suficientemente fuerte como para hacer funcionar un solo dinamo pequeño o motor. Por ello, los electroimanes se emplean en máquinas grandes. Tanto los motores como los generadores tienen dos unidades básicas: el campo magnético, que es el electroimán con sus bobinas, y la armadura, que es la estructura que sostiene los conductores que cortan el campo magnético y transporta la corriente inducida en un generador, o la corriente de excitación en el caso del motor. La armadura es por lo general un núcleo de hierro dulce laminado, alrededor del cual se enrollan en bobinas los cables conductores.

3.1. Principio de funcionamiento de un motor de cd.

El funcionamiento se basa en la interacción entre el campo magnético del imán permanente y el generado por las bobinas, ya sea una atracción o una repulsión, hacen que el eje del motor comience su movimiento.

Un motor de corriente continua está compuesto de un estator y un rotor. En muchos motores cd, generalmente los más pequeños, el estator está compuesto de imanes para

crear un campo magnético. En motores cd más grandes este campo magnético se logra con devanados de excitación de campo. El rotor es el dispositivo que gira en el centro del motor y está compuesto de arrollados de cable conductores de corriente continua. Esta corriente continua es suministrada al rotor por medio de las "escobillas" generalmente fabricadas de carbón.



Figura 3.1: líneas de campo magnético y sentido del flujo de la corriente.

La figura 3.1 muestra que cuando un conductor por el que fluye una corriente continua es colocado bajo la influencia de un campo magnético, se induce sobre él (el conductor) una fuerza que es perpendicular tanto a las líneas de campo magnético como al sentido del flujo de la corriente.

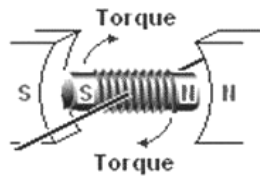


Figura 3.2: Torque y sentido de giro.

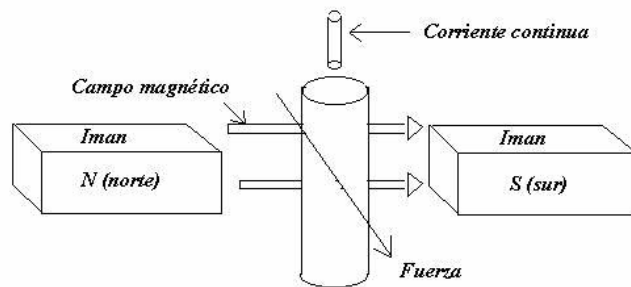


Figura 3.3: Representación del motor de cd

Para que se entienda mejor, ver como se tiene que colocar este conductor con respecto al eje de rotación del rotor para que exista movimiento. En este caso la corriente por el conductor fluye introduciéndose en el gráfico. Como se observa en las figuras 3.2 y 3.3.

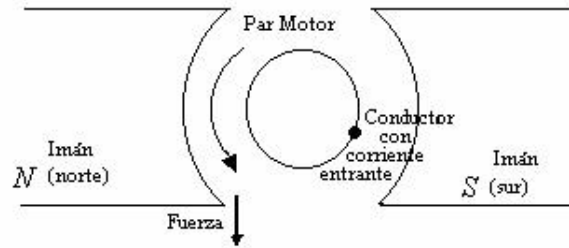


Figura 3.4: Conductor con respecto al eje de rotación.

Pero en el rotor de un motor cd no hay solamente un conductor sino muchos. Si se incluye otro conductor exactamente al otro lado del rotor y con la corriente fluyendo en el mismo sentido, el motor no girará pues las dos fuerzas ejercidas para el giro del motor se cancelan.

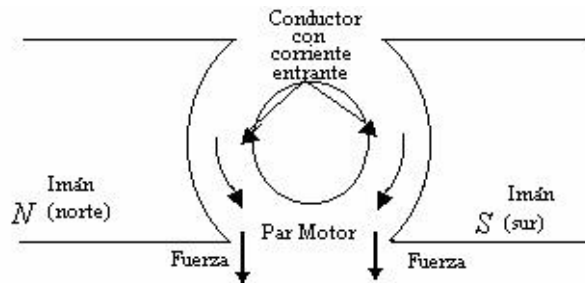


Figura 3.5: Cancelación del giro del motor.

Es por esta razón que las corrientes que circulan por conductores opuestos deben tener sentidos de circulación opuestos. Si se hace lo anterior el motor girará por la suma de la fuerza ejercida en los dos conductores.

Para controlar el sentido del flujo de la corriente en los conductores se usa un conmutador que realiza la inversión del sentido de la corriente cuando el conductor pasa por la línea muerta del campo magnético.

La fuerza con la que el motor gira (el par motor) es proporcional a la corriente que hay por los conductores. A mayor tensión, mayor corriente y mayor par motor. [22]

3.2. Circuito equivalente aproximado.

Con referencia al diagrama del circuito de la Figura 3.6, el control del motor de CD se aplica en las terminales de la armadura en la forma del voltaje e_a .

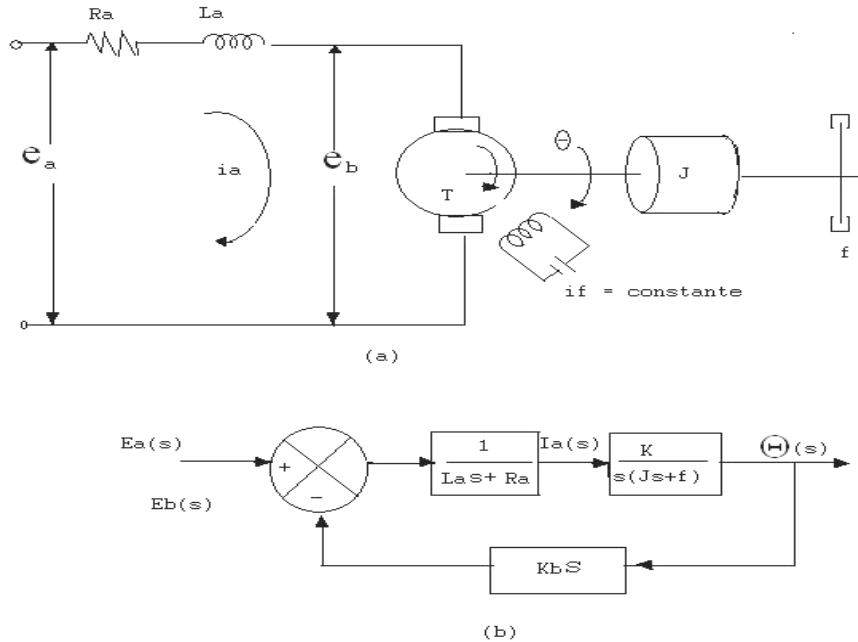


Figura 3.6: a).-Diagrama esquemático de controlado por el inducido
 b).-Diagrama de Bloques.

3.3. Obtención del modelo matemático de un motor de cd.

Para obtener un buen modelo matemático empleando técnicas de identificación, se debe alimentar el sistema con una señal de entrada de frecuencia variable que lo excite en todo su ancho de banda y, posteriormente, con la ayuda de herramientas electrónicas y computacionales, se procesan las señales de entrada y salida hasta obtener el modelo que represente en mejor forma la dinámica del sistema.

La función de transferencia de un sistema se define como la relación entre la salida y la entrada del sistema en el dominio de Laplace asumiendo condiciones iniciales nulas. Basándonos en la definición de la función de transferencia, aplicaremos una señal escalón al sistema, graficaremos la salida, hallaremos las ecuaciones de cada variable en el dominio del tiempo, las llevamos al dominio de Laplace, y la relación salida-entrada será el modelo matemático del mismo.[2]

La armadura está modelada como un circuito con resistencia R_a conectada en serie a una inductancia L_a y a una fuente de voltaje e_b que representa la fuerza contraelectromotriz en

la armadura cuando el rotor gira. Las variables y parámetros del motor se definen como sigue:

- R_a = Resistencia del devanado del inducido en ohmios.
- L_a = Inductancia del devanado del inducido en henrios.
- i_a = Corriente del devanado del inducido en amperios.
- i_f = Corriente de campo en amperios.
- e_a = Tensión aplicada a la armadura en voltios.
- e_b = Fuerza contra- electromotriz en voltios.
- θ_m = Desplazamiento angular del eje del motor en radianes.
- T_m = Par desarrollado por el motor en libras-pie.
- J_m = Momento de inercia equivalente del motor y carga con referencia al eje del motor en slug-pie²
- B_m = Coeficiente de fricción viscosa equivalente del motor y carga referido a eje del motor en libras-pie/rad/seg.[13]

3.4. Principio de funcionamiento del generador taquimétrico.

Los generadores de corriente continua son máquinas que producen tensión su funcionamiento se reduce siempre al principio de la bobina giratorio dentro de un campo magnético. Si una armadura gira entre dos polos magnéticos fijos, la corriente en la armadura circula en un sentido durante la mitad de cada revolución, y en el otro sentido durante la otra mitad. Para producir un flujo constante de corriente en un sentido, o corriente continua, en un aparato determinado, es necesario disponer de un medio para invertir el flujo de corriente fuera del generador una vez durante cada revolución. En las máquinas antiguas esta inversión se llevaba a cabo mediante un conmutador, un anillo de metal partido montado sobre el eje de una armadura. Las dos mitades del anillo se aislaban entre sí y servían como bornes de la bobina. Las escobillas fijas de metal o de carbón se mantenían en contacto con el conmutador, que al girar conectaba eléctricamente la bobina a los cables externos. Cuando la armadura giraba, cada escobilla estaba en contacto de forma alternativa con las mitades del conmutador, cambiando la posición en el momento en el que la corriente invertía su sentido dentro de la bobina de la armadura. Así se producía un flujo de corriente de un sentido en el circuito exterior al que el generador estaba conectado. Los generadores de corriente continua funcionan normalmente a voltajes bastante bajos para evitar las chispas que se producen entre las escobillas y el conmutador a voltajes altos. El potencial más alto desarrollado para este tipo de generadores suele ser de 1.500 voltios. En algunas máquinas más modernas esta inversión se realiza usando aparatos de potencia electrónica, como por ejemplo rectificadores de diodo. [29]

Los generadores modernos de corriente continua utilizan armaduras de tambor, que suelen estar formadas por un gran número de bobinas agrupadas en hendiduras longitudinales dentro del núcleo de la armadura y conectadas a los segmentos adecuados de un conmutador múltiple. Si una armadura tiene un solo circuito de cable, la corriente que se produce aumentará y disminuirá dependiendo de la parte del campo magnético a través del cual se esté moviendo el circuito. Un conmutador de varios segmentos usado con una armadura de tambor conecta siempre el circuito externo a uno de cable que se mueve a través de un área de alta intensidad del campo, y como resultado la corriente que suministran las bobinas de la armadura es prácticamente constante. Los campos de los generadores modernos se equipan con cuatro o más polos electromagnéticos que aumentan el tamaño y la resistencia del campo magnético. En algunos casos, se añaden interpolos más pequeños para compensar las distorsiones que causan el efecto magnético de la armadura en el flujo eléctrico del campo.

El campo inductor de un generador se puede obtener mediante un imán permanente (magneto) o por medio de un electroimán (dinamo). En este último caso, el electroimán se excita por una corriente independiente o por auto excitación, es decir, la propia corriente producida en la dinamo sirve para crear el campo magnético en las bobinas del inductor. Existen tres tipos de dinamo según sea la forma en que estén acoplados el inductor y el inducido: en serie, en derivación y en combinación.

Los generadores de corriente continua se clasifican según el método que usan para proporcionar corriente de campo que excite los imanes del mismo. Un generador de excitado en serie tiene su campo en serie respecto a la armadura. Un generador de excitado en derivación, tiene su campo conectado en paralelo a la armadura. Un generador de excitado combinado tiene parte de sus campos conectados en serie y parte en paralelo. Los dos últimos tipos de generadores tienen la ventaja de suministrar un voltaje relativamente constante, bajo cargas eléctricas variables. El de excitado en serie se usa sobre todo para suministrar una corriente constante a voltaje variable. Un magneto es un generador pequeño de corriente continua con un campo magnético permanente. [8]

3.5. Descripción de la caracterización experimental de un motor de cd con generador taquimétrico.

En este trabajo se propone un modelo que caracteriza el comportamiento de un tacómetro que se encuentra acoplado a un motor de cd, primeramente se obtiene resultados experimentales mismos que serán la base para el análisis numérico mediante una regresión polinomial tomando como parámetro el criterio del mínimo error cuadrático. Se utiliza una técnica para definir el modelo matemático de un motor por medio del método de ajuste de curvas de pares de datos observados (Regresión lineal o polinomial)

Dentro del control en lazo cerrado es necesario establecer una comparación entre el valor de referencia y el valor real.

El método de mínimos cuadrados es el procedimiento más adecuado para determinar las aproximaciones teóricas que lo favorecen, existen tres formas de realizarlo, por mínimas, desviación absoluta y mínimos cuadrados.

La técnica utilizada para definir el modelo matemático es el método de mínimos cuadrados donde es necesario hallar los valores $a_0, a_1, a_2, \dots, a_n$ que son los coeficientes del polinomio que caracteriza al modelo del sensor, este método concede mayor valor relativo al punto que está alejado del resto de los datos pero no permitirá que este punto domine enteramente la aproximación.[15]

El procedimiento seguido para modelar es el siguiente.

- Tomar distintas mediciones sobre la planta, con un voltímetro se toma la medida del tacómetro a una velocidad conocida de rotación del motor.
- Dichos resultados son ordenados en una tabla donde se determina cual es la variable dependiente y la variable independiente.
- Se realizan los cálculos de acuerdo al grado de polinomio que caracteriza el motor.
- Se verifica que el modelo obtenido se ajuste de mejor manera a las muestras
- En caso de que el modelo obtenido discrepe mucho de los datos muestreados, se deberá proponer un polinomio distinto al propuesto actualmente, de lo contrario puede considerarse que el modelo es válido para el sensor.

3.6. Aplicaciones.

Las bombas se utilizan para convertir energía mecánica en energía hidráulica y permiten desplazar líquidos de un lugar a otro. Se debe tener una comprensión total de los tipos de bomba disponibles y sus diferentes características. Las mediciones de carga, flujo, velocidad y par permiten determinar y comparar el rendimiento de cada bomba. [20]

> Medir la carga manométrica de una bomba usando indicadores conectados a la entrada y a la descarga

- > Medir el caudal a través de una bomba usando un tanque volumétrico o una represa de cresta de filo
- > Determinar la potencia absorbida por una bomba a partir de mediciones de par y velocidad del motor del dinamómetro
- > Determinar el rendimiento de una bomba de velocidad constante produciendo un conjunto de curvas características, a saber, carga manométrica, potencia absorbida por la bomba y eficiencia de la bomba, todos calculados respecto al flujo
- > Determinar el efecto de la velocidad sobre el rendimiento de una bomba produciendo un conjunto de curvas características
- > Comprender la diferencia entre bombas de tipo presión dinámica y bombas de desplazamiento positivo
- > Investigar el efecto de las pérdidas por aspiración en una bomba centrífuga.

Conclusiones.

El principio de funcionamiento de un motor de cd se basa en la interacción entre el campo magnético del imán permanente y el generado por las bobinas, ya sea una atracción o una repulsión, hacen que el eje del motor comience su movimiento.

La obtención del modelo matemático se realiza con la alimentación del sistema con una señal de entrada de frecuencia variable que lo excite en todo su ancho de banda y, posteriormente, con la ayuda de herramientas electrónicas y computacionales, se procesan las señales entrada y salida hasta obtener el modelo que represente en mejor forma la dinámica del sistema

En términos generales los generadores son máquinas eléctricas, son un grupo de aparatos que se utilizan para convertir la energía mecánica en eléctrica, o a la inversa, con medios electromagnéticos. A una máquina que convierte la energía mecánica en eléctrica se le denomina generador, alternador o dínamo, y a una máquina que convierte la energía eléctrica en mecánica se le denomina motor.

CAPÍTULO 4

CARACTERÍSTICAS DEL MOTOR DE CD EXPERIMENTAL

INTRODUCCIÓN

En este capítulo se presentan la características del motor, así como los componentes necesarios para la experimentación.

- La estación de trabajo.
- Especificaciones técnicas del motor.
- Circuito equivalente aproximado.
- Modelo matemático del motor de cd.
- Simulaciones digitales del motor.
- Caracterización experimental del generador taquimétrico.
- Modelo matemático del generador taquimétrico.
- Simulación digital del generador taquimétrico.
- Descripción de la obtención del modelo matemático de un motor de cd con generador taquimétrico.

La figura 4.1 muestra la estación de trabajo.

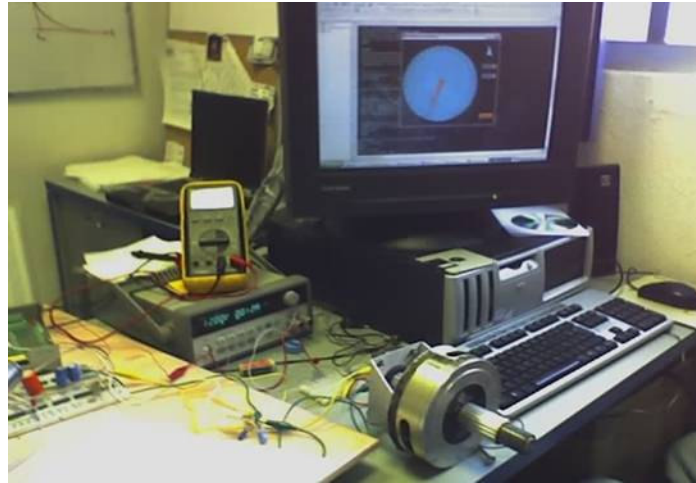


Figura 4.1: Plataforma de trabajo.

Esta estación de trabajo está constituida por: EL motor, la computadora con la tarjeta de adquisición de datos y control, la interfaz electrónica y el equipo de medición.

4.1. Especificaciones técnicas.



Figura 4.2: Motor de cd experimental.

Motor Kollmorgen /PMI U12M4/CDA-110, DC, Modelo: 00-01202-090, con generador taquimétrico Beckman modelo 1150-14-1. [21]

Características:

- Tipo: U12M4/CDA-110.
- Numero de Serie: SK36985.
- Buena velocidad de control con leva cero y bajo RFI.
- Larga vida.
- Motor robusto es ideal para muchas aplicaciones.
- Ahorra espacio y peso en aplicaciones que requirieren bajo perfil del motor.
- Gran torsión para precisión en control de velocidad y aceleración.

Especificaciones:

- Torque máximo: 1316
- Porcentaje de velocidad: 165 rpm.
- Porcentaje en voltaje terminal: 12.4 volts.
- Corriente máxima: 85 ma.
- Peso: 8.0 lbs.
- Diámetro: 5.50" in.
- Porcentaje de poder sin interrupción: 6.1

4.2. Circuito equivalente aproximado del motor cd con retroalimentación taquimétrica.

El sistema de control esta constituido por un motor que se encuentra acoplado a los mecanismos que forman la carga mecánica y la función del sistema de control es la regulación de velocidad en función del par requerido en cada etapa del funcionamiento. Un generador taquimétrico acoplado al motor produce una tensión proporcional a la velocidad y, con el fin de obtener una regulación automática de ésta, la tensión del tacómetro es realimentada a la entrada del sistema amplificador.[6]

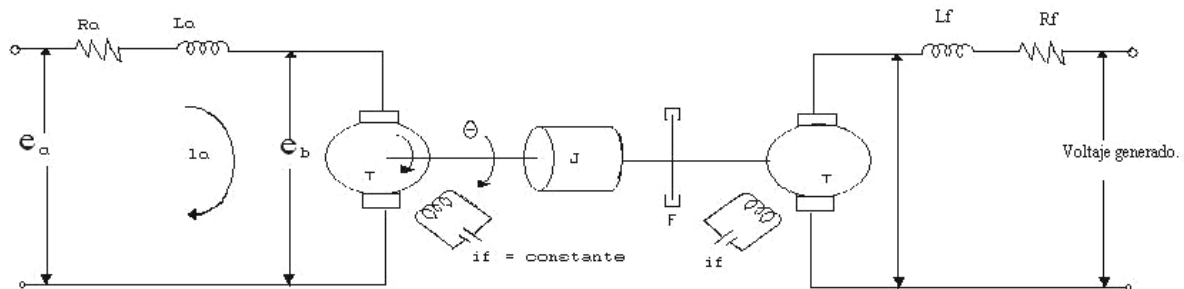


Figura 4.3: Diagrama esquemático del motor de cd con retroalimentación taquimétrica.

4.3. Modelo matemático del motor de cd.

Para un análisis lineal, se supone que el par desarrollado por el motor es proporcional al flujo en el entre-hierro y a la corriente de la armadura.

Por tanto:

$$T_m = K_m \phi_a \quad (4.3.1)$$

Ya que ϕ (flujo de campo) es constante, la ecuación se escribe como:

$$T_m = K_i i_a \quad (4.3.2)$$

en donde K_i es la *constante de par* en N-m/A (lb-pie/A, u oz-plg/A). Al comenzar con el voltaje de entrada de control $e_a(t)$, las ecuaciones de causa y efecto para el circuito del motor son:

$$T_m = K_i i_a \quad (4.3.3)$$

$$e_a = R_a i_a + L_a \frac{di_a}{dt} + e_b \quad (4.3.4)$$

$$\frac{d^2\theta_m}{dt^2} = \frac{1}{J_m} T_m - \frac{1}{J_m} T_L - \frac{B_m}{J_m} \frac{d\theta_m}{dt} \quad (4.3.5)$$

$$e_b = K_b \frac{d\theta_m}{dt} = K_b \omega_m \quad (4.3.6)$$

Considerando al motor en vacío (es decir sin carga $T_L=0$), de la ecuación (4.3.5) tenemos:

$$T_m = J_m \frac{d^2\theta_m}{dt^2} + B_m \frac{d\theta_m}{dt} = J_m \frac{d\omega_m}{dt} + B_m \omega_m = k_i i_a \quad (4.3.7)$$

El par T_m produce un par velocidad angular ω_m y un desplazamiento θ_m . Así tenemos las siguientes relaciones:

$$\omega_m = \frac{d\theta_m}{dt} \quad \therefore \quad \frac{d^2\theta_m}{dt^2} = \frac{d\omega_m}{dt} \quad (4.3.8)$$

De la ecuación (4.3.5) resulta:

$$\frac{d^2\theta_m}{dt^2} = \frac{K_i}{J_m} i_a - \frac{B_m}{J_m} \frac{d\theta_m}{dt} \quad (4.3.9)$$

De (4.3.4) obtenemos:

$$\frac{di_a}{dt} = \frac{e_a}{L_a} - \frac{R_a}{L_a}i_a - \frac{e_b}{L_a} \quad (4.3.10)$$

Integrando:

$$i_a = \int \left(\frac{e_a}{L_a} - \frac{R_a}{L_a}i_a - \frac{e_b}{L_a} \right)$$

Suponiendo que todas las condiciones iniciales son cero, y tomando las transformadas de Laplace de las ecuaciones 4.3.3, 4.3.6 y 4.3.7, se obtienen las siguientes ecuaciones:

$$K_b s \Theta(s) = E_b(s) \quad (4.3.11)$$

$$(L_a s + R_a) I_a(s) + E_b(s) = E_a(s) \quad (4.3.12)$$

$$(J_m s^2 + B_m s) \Theta(s) = T_m(s) = K I_a(s) \quad (4.3.13)$$

Se ve el efecto de la fuerza contra electromotriz en la señal de realimentación proporcional a la velocidad del motor. Esta fuerza contraelectromotriz, transferencia de este sistema como

$$\frac{\Theta(s)}{E_a(s)} = \frac{K}{s[L_a J s^2 + (L_a B_m + R_a J)s + R_a B_m + K K_b]} \quad (4.3.14)$$

la inductancia L_a en el circuito de inducido generalmente es pequeña y se la puede despreciar. Si se desprecia L_a , la función de transferencia dada por la Ec. 4.3.14 se reduce a

$$\frac{\Theta(s)}{E_a(s)} = \frac{K}{s(Ts + 1)} \quad (4.3.15)$$

donde

$K_m = K/(R_a J + K K_b)$ = constante de ganancia del motor.

$T = R_a J / (R_a J + K K_b)$ = constante de tiempo del motor. [2]

4.4. Simulaciones digitales del motor de cd en lazo abierto.

Los parámetros del motor de cd se proporcionan a continuación: [14]

$$\omega = 250rpm$$

$$i_a = 2.0amps$$

$$e_a = 12.0volts$$

$$R_a = 3.5\Omega$$

$$L_a = 0.041H$$

$$J = 0.04Kg - m^2$$

$$f = 0.17N - m / rad / seg$$

$$K = 1.2N - m / amp$$

$$Kb = 0.1815volt - seg / rad$$

El software utilizado para la realización de las simulaciones es Matlab versión 6 para Windows. Las simulaciones de velocidad e intensidad de corriente se dan en la figura 4.4.

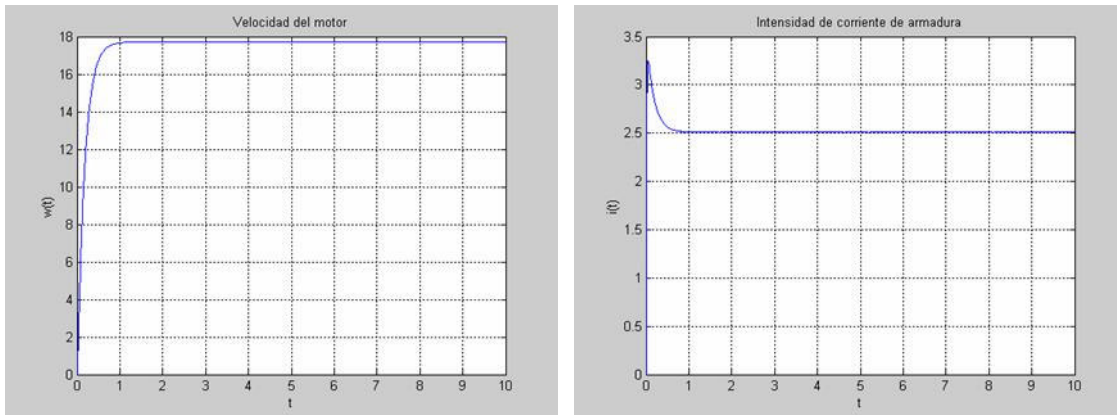


Figura 4.4: Comportamiento de la velocidad e intensidad de corriente del motor.

4.5. Caracterización experimental del generador taquimétrico.

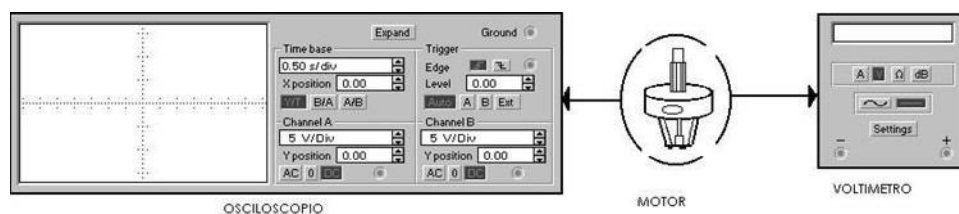


Figura 4.5: Esquema de medición para la toma de muestras.

El siguiente paso es registrar las distintas muestras, en la Tabla 1 están los resultados obtenidos.

Muestra	Voltaje(V)	Frecuencia(hz)
1	0	0
2	1	2.45
3	2.3	5.4
4	3.9	9.1
5	5.9	14
6	7	16.5
7	8	19
8	9.1	21.3
9	10.1	23.8
10	11	26.5
11	12	28.2
12	13	30.3
13	13.5	32.3
14	14	33.3
15	14.4	33.9

Tabla 4.1: Datos obtenidos en laboratorio.

Con los datos obtenidos se realiza el cálculo de la regresión lineal de orden uno en este caso se realiza mediante un programa en MATLAB.

4.6. Modelo matemático del generador taquimétrico.

Se desea modelar matemáticamente a un sensor con una función polinómica de grado n y con coeficientes constantes $a_0, a_1, a_2, \dots, a_n$ tal que sea más precisa, se propone la siguiente función.

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + E \tag{4.6.1}$$

$$E = y - (a_0 + a_1x + \dots + a_nx^n) \tag{4.6.2}$$

Determinando el S_r de (4.6.3) que define el cuadrado de la diferencia del error para los n valores pares observados

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

$$S_r = \sum_{i=1}^n E_i^2 = \sum_{i=1}^n \{y_1 - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_n x_i^n\}^2 \quad (4.6.3)$$

obteniendo las derivadas parciales de S_r con respecto a los coeficientes del polinomio del ajuste minimizando la correspondiente derivada.

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^n E_i \{y_1 - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_n x_i^n\} = 0 \quad (4.6.4)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^n E_i \{y_1 - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_n x_i^n\} x_i = 0 \quad (4.6.5)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum_{i=1}^n E_i \{y_1 - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_n x_i^n\} x_i^2 = 0 \quad (4.6.6)$$

·
·
·

$$\frac{\partial S_r}{\partial a_n} = -2 \sum_{i=1}^n E_i \{y_1 - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_n x_i^n\} x_i^n = 0 \quad (4.6.7)$$

Ordenando el sistema de ecuaciones (4.6.4)-(4.6.7), considerando que $\sum_{i=1}^n = \Sigma$ y

$\sum_{i=1}^n a_0 = n a_0$, donde n es el numero de pares observados

$$\begin{aligned} a_0 + a_1 \sum x_i + a_2 \sum x_i^2 + \dots + a_n \sum x_i^n &= \sum y_i \\ a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \dots + a_n \sum x_i^{n+1} &= \sum x_i y_i \\ a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \dots + a_n \sum x_i^{n+2} &= \sum x_i^2 y_i \\ &\vdots \\ a_0 \sum x_i^n + a_1 \sum x_i^{n+1} + a_2 \sum x_i^{n+2} + \dots + a_n \sum x_i^{2n} &= \sum x_i^n y_i \end{aligned}$$

Es posible determinar los coeficientes $a_0, a_1, a_2, \dots, a_n$ del polinomio de la ecuación (4.6.1) tal que el ajuste a los datos observados sea el mejor

$$\Delta = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+2} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+3} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix} \quad (4.6.8)$$

$$\Delta_0 = \begin{bmatrix} \sum y_i & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i y_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \sum x_i^2 y_i & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \sum x_i^n y_i & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix} \quad (4.6.9)$$

$\Delta_1, \Delta_2, \dots, \Delta_n$ son determinados de manera similar a (4.6.9) sustituyendo el segundo miembro del sistema en la columna n . [15]

Tal que los coeficientes quedan determinados de la forma siguiente:

$$y = \frac{\Delta_0}{\Delta} + \frac{\Delta_1}{\Delta} x + \frac{\Delta_2}{\Delta} x^2 + \cdots + \frac{\Delta_n}{\Delta} x^n \quad (4.6.10)$$

con

$$a_0 = \frac{\Delta_0}{\Delta}, a_1 = \frac{\Delta_1}{\Delta}, a_2 = \frac{\Delta_2}{\Delta}, \dots, a_n = \frac{\Delta_n}{\Delta} \quad (4.6.11)$$

4.7. Simulación digital del generador taquimétrico

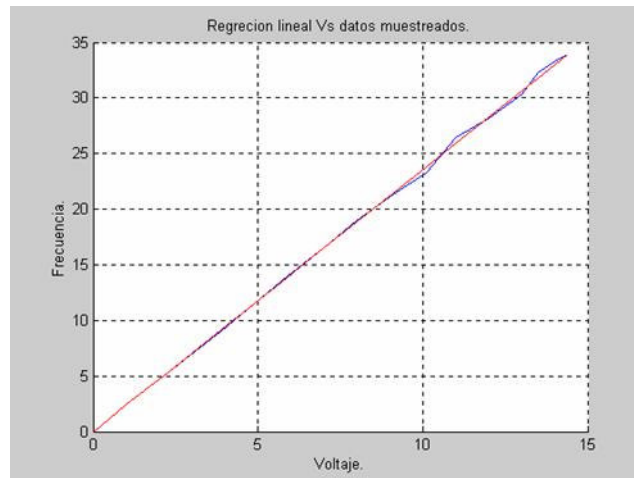


Figura 4.6: Gráfica de la regresión lineal aproximada.

4.8. Descripción de la obtención del modelo matemático de un motor de cd con generador taquimétrico.

R_m = Resistencia del devanado del inducido en ohmios.

L_m = Inductancia del devanado del inducido en henrios.

i_m = Corriente del devanado del inducido en amperios.

i_g = Corriente de campo en amperios.

V_m = Tensión aplicada a la armadura en voltios.

Vb_m = Fuerza contra- electromotriz en voltios.

w_m = Velocidad del motor.

T_m = Par desarrollado por el motor en libras-pie.

J_m = Momento de inercia equivalente del motor y carga con referencia al eje del motor en slug-pie²

$f_m = f_g$ = Coeficiente de fricción viscosa equivalente del motor y generador en libras-pie/rad/seg.

L_g = Inductancia del generador en henrios.

$w_g = w_m$ = Desplazamiento angular del eje del generador.

T_g = Par desarrollado por el generador.

J_g = Momento de inercia equivalente del generador.

kb_g = Constante.

kt = Constante.

$$L_m \frac{di_m}{dt} + R_m i_m + Vb_m = V_m \quad (4.8.1)$$

$$j_m \frac{dw_m}{dt} + f_m w_m + j_g \frac{dw_m}{dt} + f_g w_m = T_m \quad (4.8.2)$$

$$T_m = kt i_m$$

$$(j_m + j_g) \frac{dw_m}{dt} + (f_m + f_g) w_m = kt i_m \quad (4.8.3)$$

$$V_g = L_g \frac{di_g}{dt} + R_g i_g + Vb_g \quad (4.8.4)$$

donde:

$$Vb_g = Kb_g w_m \quad (4.8.5)$$

Conclusiones

De este capítulo se concluye que es importante el modelado matemático y simulación de los actuadores, no sólo para diseñar la interfaz electrónica, sino también para definir el valor adecuado de las variables utilizadas en los controles.

CAPÍTULO 5

CONTROL DE VELOCIDAD DE MOTORES DE CD

INTRODUCCIÓN

En este capítulo se describe la teoría de los distintos controles comúnmente usados en la industria.

- Esquema clásico de control.
- Estrategias clásicas de control.
- Acciones clásicas de control y controles automáticos industriales.
- Clasificación de los controles automáticos.
- Elementos de control automáticos industriales.
- Controles autoactuates.
- Acciones de control.
- Acción de dos posiciones o de si-no.
- Acción de control proporcional.
- Acción de control integral.
- Acción de control proporcional e integral.
- Acción de control proporcional y derivativo.
- Acción de control proporcional y derivativo e integral.
- Control no lineal: PID NL.
- Control PX.
- Simulaciones digitales.

5.1. Esquema clásico de control.

En la figura 5.1 se presenta el diagrama de un motor de cd. controlado en el devanado de armadura, para ello se considera un devanado de campo de imanes permanentes [2].

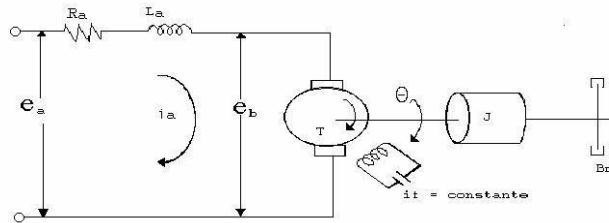


Figura 5.1 Esquema de control.

5.2. Estrategias clásicas de control.

5.2.1. Acciones clásicas de control y controles automáticos industriales.

Un control automático compara el valor efectivo de salida de una planta con el valor deseado, determina la desviación y produce una señal de control que reduce la desviación a cero o a un valor pequeño. La forma en que el control automático produce la señal de control recibe el nombre de acción de control.

En este capítulo se presentan las acciones de control básicas utilizadas comúnmente en los controles automáticos industriales. Primero se introduce el principio de operación de los controles automáticos y los diversos métodos de generación de las señales de control, como la derivada o integral de la señal de error. Luego se analizan los efectos de los distintos modos de control en el funcionamiento del sistema. A continuación se tratan brevemente los procedimientos para reducción de los efectos de las perturbaciones externas. Finalmente se introducen los amplificadores, los principios básicos de la hidráulica y las aplicaciones de los dispositivos hidráulicos.

5.2.2. Clasificación de los controles automáticos.

De acuerdo con su acción de control se pueden clasificar los controles automáticos industriales en:

- Controles de dos posiciones o control si-no.
- Controles proporcionales.
- Controles integrales.
- Controles proporcionales e integrales.
- Controles proporcionales y derivados.

Controles proporcionales y derivados e integrales.

La mayoría de los controles automáticos industriales usan como fuentes de potencia la electricidad o un fluido a presión que puede ser aceite o aire. También se pueden clasificar los controles automáticos según el tipo de fuente de energía usada en su funcionamiento, en controles neumáticos, hidráulicos, o electrónicos. Qué tipo de control usar depende de la naturaleza de la planta y sus condiciones de funcionamiento, inclusive consideraciones de seguridad, costo, disponibilidad, confiabilidad, precisión, peso y tamaño.

5.2.3. Elementos de control automáticos industriales.

Un control automático debe detectar la señal de error actuante, que habitualmente se encuentra a un nivel de potencia muy bajo y amplificarla a un nivel suficientemente alto. Por tanto, se requiere de un amplificador. La salida del control automático va a actuar sobre un dispositivo de potencia, como un motor neumático o válvula, un motor hidráulico o un motor eléctrico.

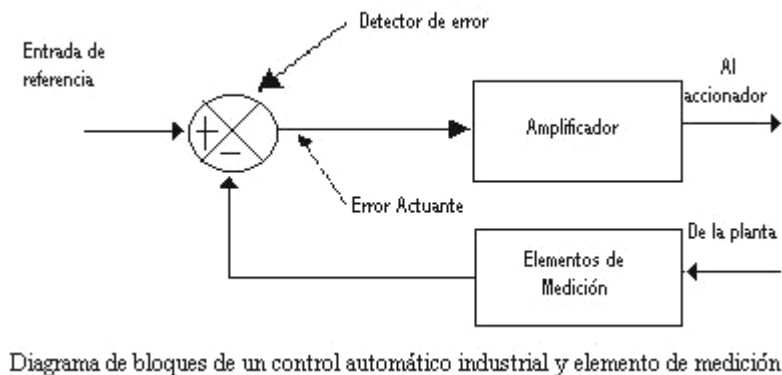


Figura 5.2: Diagrama de bloques de un control automático.

En la figura 5.2 se tiene un diagrama de bloques de un control automático industrial y un elemento de medición. El control en si consiste en un detector de error y amplificador. El elemento de medición es algún dispositivo que convierte la variable de salida en otra variable adecuada, como un desplazamiento, presión o señal electrónica, que pueda usarse como comparación de salida con la señal de entrada de referencia. Este elemento se encuentra en el camino de realimentación del sistema de lazo cerrado. Hay que convertir el punto de ajuste o regulación del control a una referencia de entrada de las mismas unidades que la señal de realimentación del electo de medición. El amplificador efectúa la amplificación de la potencia de la señal de error actuante, la que a su vez opera sobre el accionador. (Frecuentemente se usa un amplificador juntamente con algún circuito de realimentación adecuado, para modificar la señal de error actuante amplificándola y a veces derivándola o integrándola para lograr una señal de control

mejor.) El accionador es un elemento que altera la entrada a la planta de acuerdo con la señal de control, de manera que se pueda poner en correspondencia la señal de realimentación con la señal de referencia de entrada.[5]

5.2.4. Controles autoactuantes.

En la mayor parte de los controles automáticos industriales se usan unidades independientes como elementos de medición y como accionador. Sin embargo, en los muy simples, como autoactuantes, ambas unidades están reunidas en una sola. Los controles autoactuantes utilizan potencia tomada del elemento de medición y son muy simples y económicos.

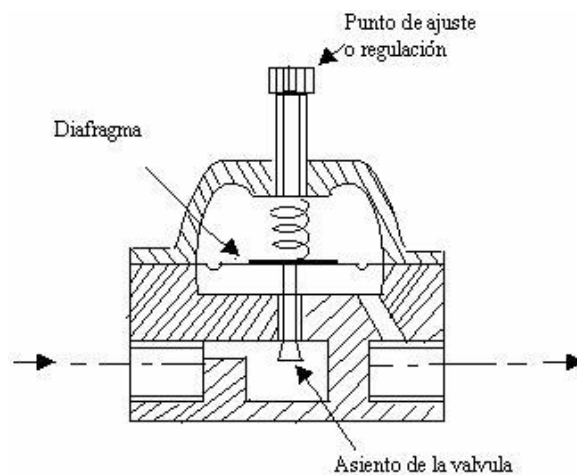


Figura 5.3: Control Automático.

En la figura 5.3 se puede ver un ejemplo de control autoactuante. El punto de ajuste queda determinado por el ajuste de la fuerza en el resorte. El diafragma mide la presión controlada. La señal de error actuante es la fuerza neta que actúa sobre el diafragma. Su posición determina la apertura de la válvula.

El funcionamiento del control autoactuante es el siguiente: supóngase que la presión de salida es inferior a la presión de referencia, según lo determina el punto de ajuste o regulación. Entonces la fuerza del resorte hacia abajo es superior a la fuerza ascendente de la presión, produciéndose un desplazamiento del diafragma hacia abajo. Esto aumenta el flujo y, por tanto, la presión de salida. Cuando la presión hacia arriba iguala a la fuerza del resorte hacia abajo, el vástago de la válvula permanece estacionario y el flujo es constante. Inversamente, si la presión de salida llega a ser mayor que la presión de referencia, se reduce la apertura de la válvula y disminuye el flujo a través de la misma. En un control como éste, el flujo a través de la apertura de la válvula es aproximadamente proporcional a la señal de error actuante.

5.2.5. Acciones de control.

En los controles automáticos industriales son muy comunes los seis tipos de acción básica de control: de dos posiciones o si-no, proporcional, integral, proporcional e integral, proporcional y derivativo y proporcional y derivativo e integral. Es importante comprender las características básicas de las diversas acciones, para que el ingeniero de control pueda elegir la más adecuada para determinada aplicación. [2]

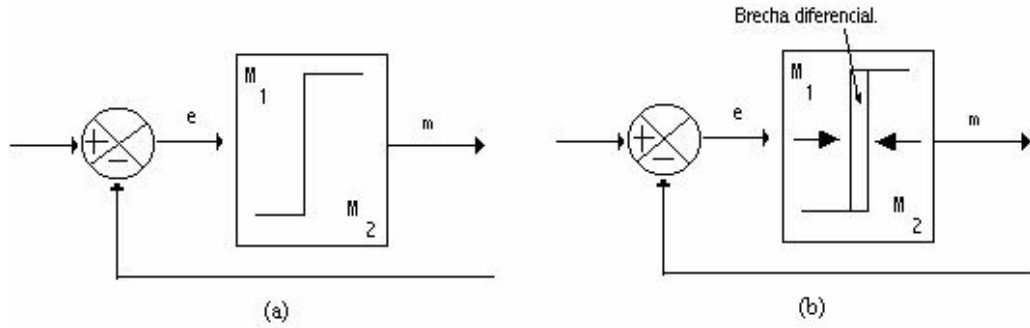
5.2.6. Acción de dos posiciones o acción si-no.

En un sistema de control de dos posiciones, el elemento accionador tiene solamente dos posiciones fijas, que en muchos casos son simplemente conectado y desconectado. El control de dos posiciones o control si-no es relativamente simple y económico, y, por esta razón, ampliamente utilizado en sistemas de control tanto industriales como domésticos. Sea la señal de salida del control $m(t)$ y la señal de error actuante $e(t)$. En un control de dos posiciones, la señal $m(t)$ permanece en un valor máximo o mínimo, según que la señal actuante sea positiva o negativa, de modo que

$$\begin{aligned} m(t) &= M_1 && \text{para } e(t) > 0 \\ &= M_2 && \text{para } e(t) < 0 \end{aligned} \tag{5.2.1}$$

donde M_1 y M_2 son constantes, generalmente el valor mínimo de M_2 es o bien cero o M_1 . los controles de dos posiciones son generalmente dispositivos eléctricos, donde habitualmente hay una válvula accionada por un solenoide eléctrico. También los controles de dos posiciones y se los denomina a veces controles neumáticos de dos posiciones.

Las figuras 5.4(a) y 5.4(b) presentan los diagramas de bloques de controles de dos posiciones. El rango en el que debe desplazar la señal de error actuante antes de que se produzca la conmutación se llama brecha diferencial. En la figura 5.4(b) se indica una brecha diferencial. La brecha diferencial hace que la salida del control $m(t)$ mantenga su valor hasta que la señal de error actuante haya pasado levemente del valor cero. En algunos casos, la brecha diferencial es un resultado de fricción no intencional y movimiento perdido, sin embargo, normalmente se lo provee deliberadamente para evitar la acción excesivamente frecuente del mecanismo del control si-no. [2]



(a) Diagrama de bloques de un control si-no; (b) diagrama de bloques de un control si-no con brecha diferencial.

Figura 5.4: Diagrama de bloques de control si-no.

Sea el sistema de control de nivel de líquido que se ve en la figura 5.5 con el control de dos posiciones, la válvula está o bien abierta o cerrada. El caudal de entrada de agua es una constante positiva o cero. Como se ve en la figura, la señal de salida se mueve continuamente entre los dos límites requeridos para que el elemento accionador se desplace de una posición fija a la otra. Se hace notar que la curva de salida sigue una de las dos curvas exponenciales; una que corresponde a llenado y la otra a vaciado. Una oscilación como ésta entre dos límites es típica respuesta característica de un sistema de dos posiciones.

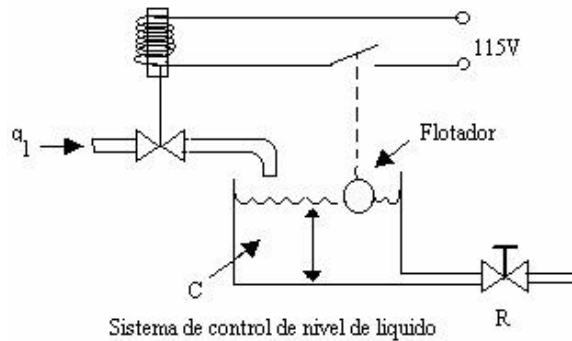


Figura 5.5: Sistema de Control de nivel de liquido.

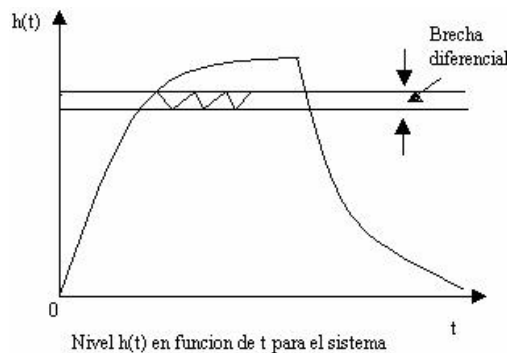


Figura 5.6: Gráfica del sistema de control.

De la figura 5.6 se ve que se puede reducir la amplitud de la oscilación de salida reduciendo la brecha diferencial. Esto, sin embargo, aumenta la cantidad de conmutaciones por minuto y reduce la vida útil del componente. Hay que determinar el valor de la brecha diferencial por consideraciones de exactitud deseada y duración de los componentes. [2]

5.2.7. Acción de control proporcional.

Para un control de acción proporcional, la relación entre la salida del controlador $m(t)$ y la señal de error actuante $e(t)$ es

$$m(t) = k_p e(t) \tag{5.2.2}$$

o en magnitudes transformadas de Laplace,

$$\frac{M(s)}{E(s)} = K_p \tag{5.2.3}$$

donde K_p se denomina sensibilidad proporcional o ganancia.

Cualquiera que sea el mecanismo en si, y sea cual sea la potencia que lo alimenta, el control proporcional esencialmente es un amplificador con ganancia ajustable. En la figura 5.7 se puede ver un diagrama de bloques de este control.[2]

5.2.8. Acción de control integral.

En un control con acción integral, el valor de la salida del controlador $m(t)$ varía proporcionalmente a la señal actuante $e(t)$. Es decir,

$$\frac{dm(t)}{dt} = k_i e(t) \tag{5.2.4}$$

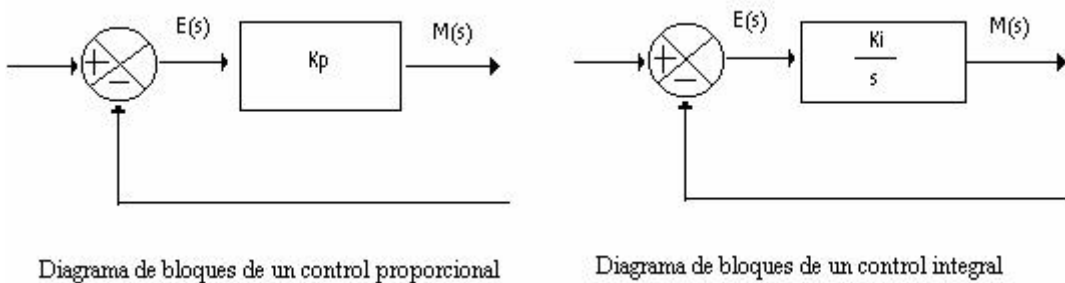


Figura 5.7: Diagrama de bloques de controles PD y PI.

o

$$m(t) = k_i \int_0^t e(t) dt \quad (5.2.5)$$

donde K_i es una constante regulable. La función transferencia del control integral es

$$\frac{M(s)}{E(s)} = \frac{K_i}{s} \quad (5.2.6)$$

Si se duplica el valor de $e(t)$, el valor de $m(t)$ varía dos veces más rápido. Para un error actuante igual a cero, el valor de $m(t)$ se mantiene estacionario. La acción de control integral recibe a veces el nombre de control de reposición. La figura 5.7 muestra un diagrama de bloques de este control. [2]

5.2.9. Acción de control proporcional e integral.

La acción de control proporcional e integral queda definida por la siguiente ecuación:

$$m(t) = k_p e(t) + \frac{k_p}{T_i} \int_0^t e(t) dt \quad (5.2.7)$$

o la función transferencia del control es

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (5.2.8)$$

donde K_p representa la sensibilidad proporcional o ganancia y T_i el tiempo integral. Tanto K_p como T_i son regulables. El tiempo integral regula la acción de control integral, mientras una modificación en K_p afecta tanto a la parte integral como a la proporcional de la acción de control. A la inversa del tiempo integral T_i se la llama frecuencia de reposición la frecuencia de reposición es el número de veces por minuto que se duplica la parte proporcional de la acción de control. La frecuencia de repetición se mide en términos de repeticiones por minutos. La figura 5.8(a) muestra un diagrama de bloques de un control proporcional e integral. Si la señal de error actuante $e(t)$ es una función escalón unitario como se ve en la figura 5.8(b), la salida del control $m(t)$ pasa a ser la indicada en la figura 5.8(c). [2]

5.2.10. Acción de control proporcional y derivativo.

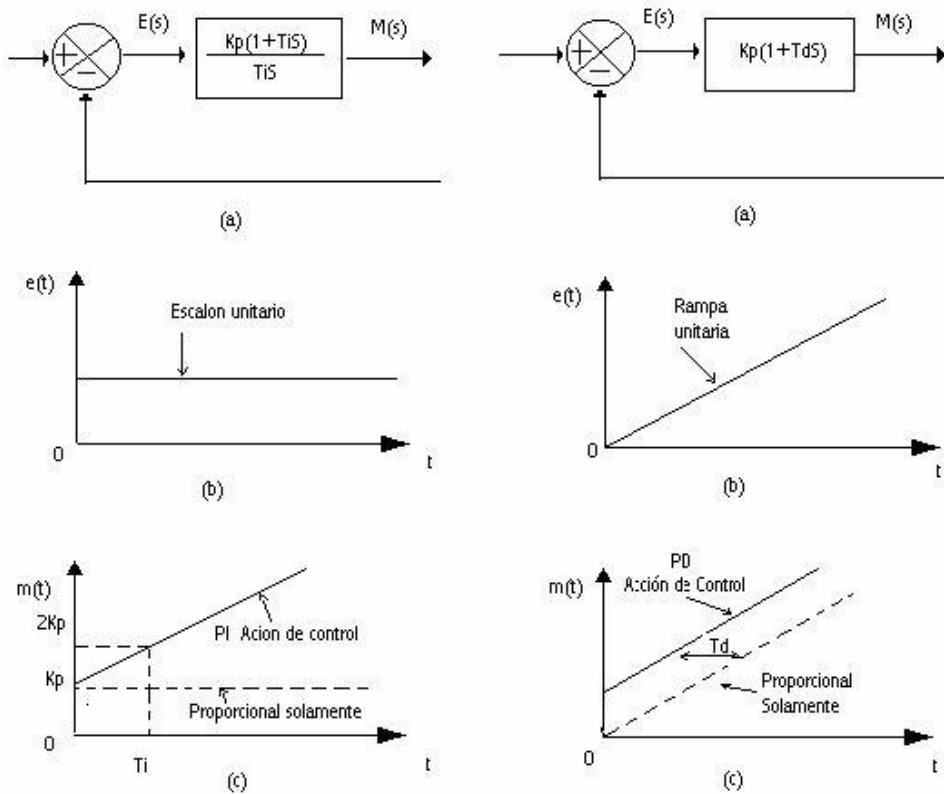
La acción de control proporcional y derivativo queda definida por la siguiente ecuación:

$$m(t) = k_p e(t) + k_p T_d \frac{de(t)}{dt} \tag{5.2.9}$$

y la función transferencia es

$$\frac{M(s)}{E(s)} = k_p (1 + T_d s) \tag{5.2.10}$$

donde K_p es la sensibilidad proporcional y T_d es el tiempo derivativo. Tanto K_p como T_d son regulables. La acción de control derivativa, a veces denominada control de velocidad, es cuando el valor de salida del control es proporcional a la velocidad de la señal de error actuante. El tiempo derivativo T_d es el intervalo de tiempo en el que la acción de velocidad se adelanta al efecto de acción proporcional.



(a) diagrama de bloques de un control proporcional e integral; (b) y (c) diagramas que muestran una entrada escalon unitario y la salida del control.

(a) diagrama de bloques de un control proporcional y derivativo; (b) y (c) diagrama que muestran una entrada rampa unitaria y la salida del control.

Figura 5.8: Diagrama del bloques de Control PI, gráficas de entrada y salida.

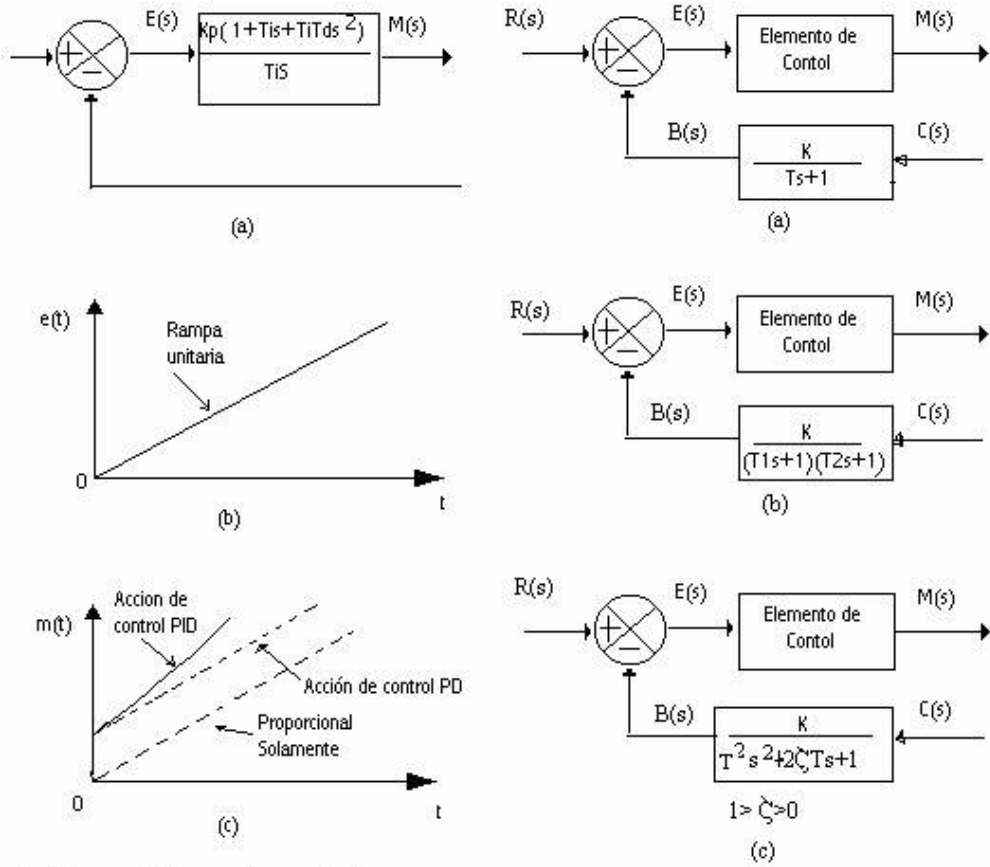
Figura 5.9: Diagrama de bloques del control PD, gráficas de entrada y salida.

La figura 5.9(a) presenta un diagrama de bloques de un control proporcional y derivativo. Si la señal de error actuante $e(t)$ es una función rampa unitaria, como se ve en la figura 5.9(b), la salida del control $m(t)$ es la que puede verse en la figura 5.9(c). Como puede verse en la figura 5.9(c), la acción de control derivativo tiene carácter de anticipación. Sin embargo, por supuesto, la acción de derivativa nunca puede anticiparse a una acción que aún no ha tenido lugar.

Mientras la acción de control derivativo tiene la ventaja de ser anticipadora, tiene las desventajas de que amplifica las señales de ruido y puede producir efecto de saturación en el accionador. Hay que notar que nunca se puede tener una acción de control derivativo sola, por que este control es efectivo únicamente durante períodos transitorios. [2]

5.2.11. Acción de control proporcional y derivativo e integral.

La combinación de los efectos de acción proporcional, acción de control derivativa y acción de control integral, se llama acción de control proporcional y derivativo e integral. Esta acción combinada tiene las ventajas de cada una de las tres acciones de control individuales.



(a) Diagrama de bloques de un control proporcional y derivativo e integral; (b) y (c) diagramas que muestran una entrada rampa unitaria y la salida del control.

Diagrama de bloques de controles automáticos con (a) elemento de medición de primer orden; (b) de segundo orden sobreamortiguado; (c) elemento de medición de segundo orden subamortiguado

Figura 5.10: Diagrama de bloques de controles PID, gráficas de entrada y salida.

La ecuación de un control con esta acción de control combinada está dada por

$$m(t) = k_p e(t) + k_p T_d \frac{de(t)}{dt} + \frac{K_p}{T_i} \int_0^t e(t) dt \tag{5.2.11}$$

o la función transferencia es

$$\frac{M(s)}{E(s)} = k_p \left(1 + T_d s + \frac{1}{T_i s} \right) \tag{5.2.12}$$

donde Kp representa la sensibilidad proporcional, T_d el tiempo derivativo y T_i el tiempo integral. En la figura 5.10(a) se puede ver el diagrama de bloques de un control proporcional y derivativo e integral. Si $e(t)$ es una función rampa unitaria como la de la figura 5.10(b), la salida de control $m(t)$ es el que puede verse en la figura 5.10(c). [2]

5.3. Acción de control no lineal: PID NL.

El control PID NO LINEAL, se basa en la siguiente ecuación:

$$u(t) = KdS + Ki \int \text{sgn}(S) ds. \quad (5.3.1)$$

$$Kd = 2Wn. \quad (5.3.2)$$

$$Ki = KdKp. \quad (5.3.3)$$

$$S = \alpha e + e' \quad (5.3.4)$$

$$\alpha = Wn / 2 \quad (5.3.5)$$

$$e = Vb - Vr \quad (5.3.6)$$

$$e' = \frac{d(Vr)}{dt} \quad (5.3.7)$$

donde:

$u(t)$.- Señal de control.

Kd .- Constante derivativa.

Ki .- Constante de integración.

Kp .- Constante proporcional.

S .-

α .- Parámetro del control.

e .- Error de velocidad.

Vb .- Velocidad deseada.

Vr .- Velocidad real.

e' .- Derivada del error.

$\text{sgn}(x)$.- Función signo.

$$\text{sgn}(x) = \begin{cases} 1; x > 0 \\ 0; x = 0 \\ -1; x < 0 \end{cases}$$

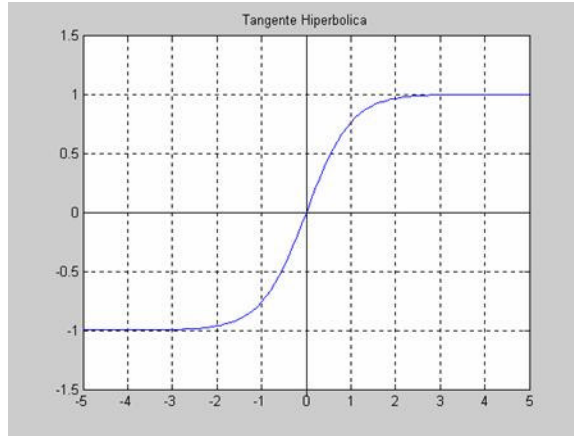


Figura 5.11: Gráfica de la función signo.

Con el fin de no llegar a una indeterminación en cero, se usa la función tangente hiperbólica de x ($\tanh(x)$). La determinación de los parámetros α y β utilizados se definen en los siguientes intervalos:

$$1 \leq \beta \leq 100$$

$$1 \leq \alpha \leq 12$$

Dado que:

$$\int \text{sgn}(s) ds \approx \int \tanh(\beta\alpha) ds$$

Por tanto tenemos que:

$$u(t) = KdS + Ki \int \tanh(\beta x) ds$$

(5.6.8)

donde:

$$\beta = 100$$

$\tanh()$.- Tangente hiperbólica.

β .- Constante.

El incremento de β permite la aproximación de la tangente hiperbólica a la función signo, de la forma siguiente:

[14]

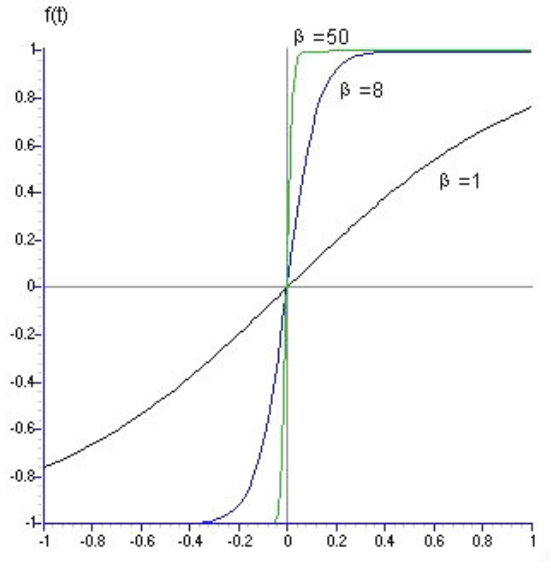


Figura 5.12: Gráfica de la tangente hiperbólica variando a β .

5.4. Acción de control PX.

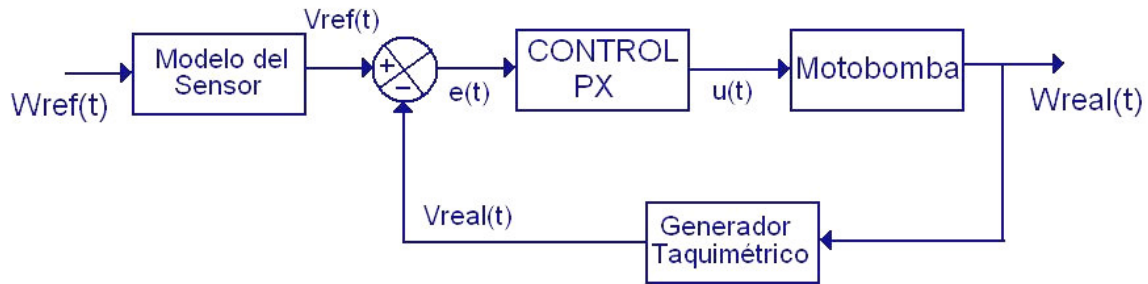


Figura 5.13: Diagrama de bloques integral del sistema de control PX.

En la figura 5.13, se presenta el diagrama de bloques correspondiente al control PX, en el que se tiene como entrada de referencia a la velocidad deseada $W_{ref}(t)$, mapeada a voltaje deseado o de referencia a través del modelo del generador taquimétrico que para este caso es el sensor de velocidad de la motobomba. Este voltaje de referencia $V_{ref}(t)$ se compara con el real, proporcionado por el sensor de velocidad y que es el representativo de la velocidad en la flecha de la motobomba. La diferencia constituye un error. Para este control PX, la única variable independiente es el error dependiente del estado de velocidad, y no de la derivada (control PD) y la integral del error (control PID), logrando beneficios significativos sobre los controles clásicos PD, PI y PID debido a que existe regulación global con solo medición del estado y no de su derivada. La representación formal de este control está descrito por el siguiente conjunto de ecuaciones:

$$u(t) = -kpe(t) - kdep(t) \quad (5.3.9)$$

$$e(t) = W_{real}(t) - W_{ref}(t) \quad (5.3.10)$$

$$ep(t) = -LW_{real}(t) + kde(t) \quad (5.3.11)$$

Como podemos apreciar, el control $u(t)$ depende exclusivamente de $e(t)$, en la figura 5.14 podemos aclarar la dependencia del control $u(t)$ en términos del estado de velocidad, tal y como se propone y se aplica experimentalmente.

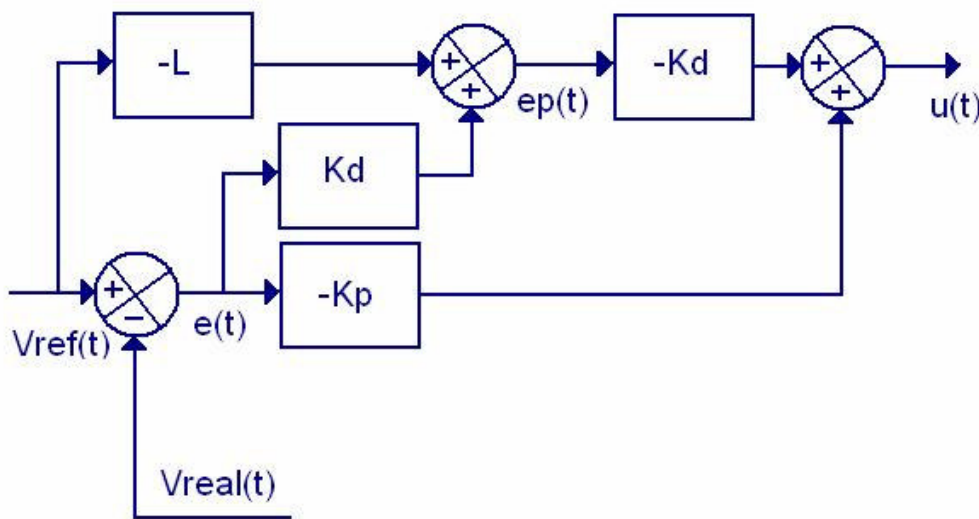


Figura 5.14: Diagrama de bloques del sistema de control PX dependiente exclusivamente del estado de velocidad.

5.5. Simulaciones digitales.

La simulación por computadora de un sistema, es una herramienta que permite conocer su comportamiento en diferentes condiciones. Para esto se requiere conocer el modelo matemático y los parámetros del sistema, el comportamiento de los estados del sistema son factor determinante para elegir un algoritmo de control.

El software utilizado para la realización de las simulaciones es Matlab versión 6 para Windows. Las simulaciones son de cada uno de los controles presentados anteriormente considerando una referencia de velocidad.

Los programas en Matlab se presentan en el apéndice de programas.

Control PD.

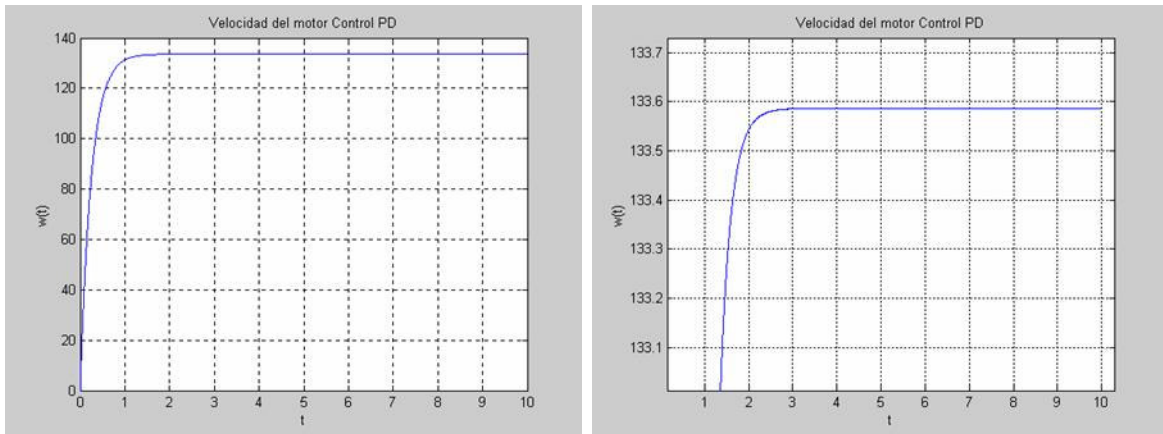


Figura 5.15: Gráfica de la velocidad del motor usando control PD. Ampliación de la Gráfica.

En esta gráfica podemos observar como la señal de velocidad trata de alcanzar el valor deseado que son las 135 rpm, pero por su baja influencia a la disminución del error solo puede alcanzar un valor aproximado a los 133.58 rpm. Es decir existe un error de estado estable.

Control PI.

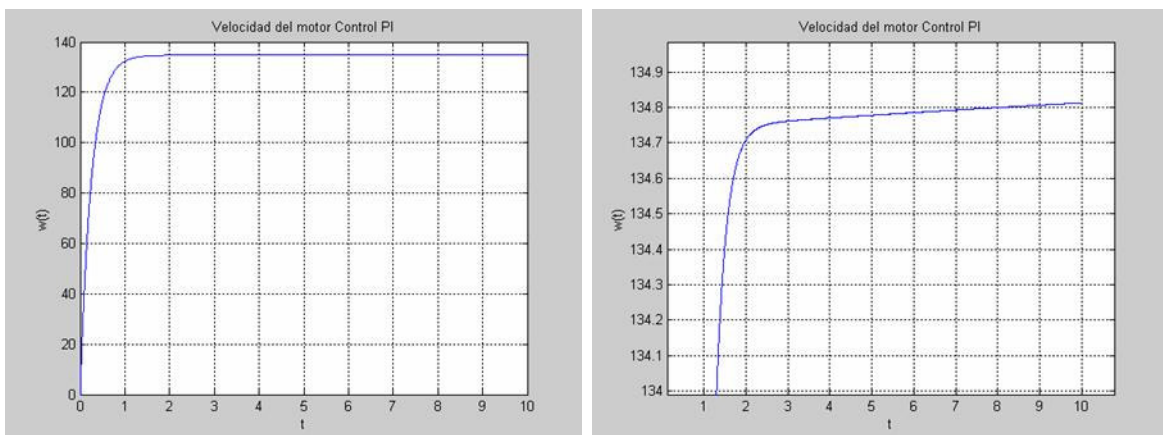


Figura 5.16: Gráfica de la velocidad del motor usando control PI. Ampliación de la Gráfica.

Este control es empleado de manera considerable y relativamente aceptable para el control de velocidad en motores de cd, debido a que las acción integral tiene la tendencia de corregir el error de estado estable, podemos apreciar en las gráficas de simulación que el valor máximo de velocidad es de 134.8 rpm, con un error de estado estable igual a 0.2 rpm.

Control PID.

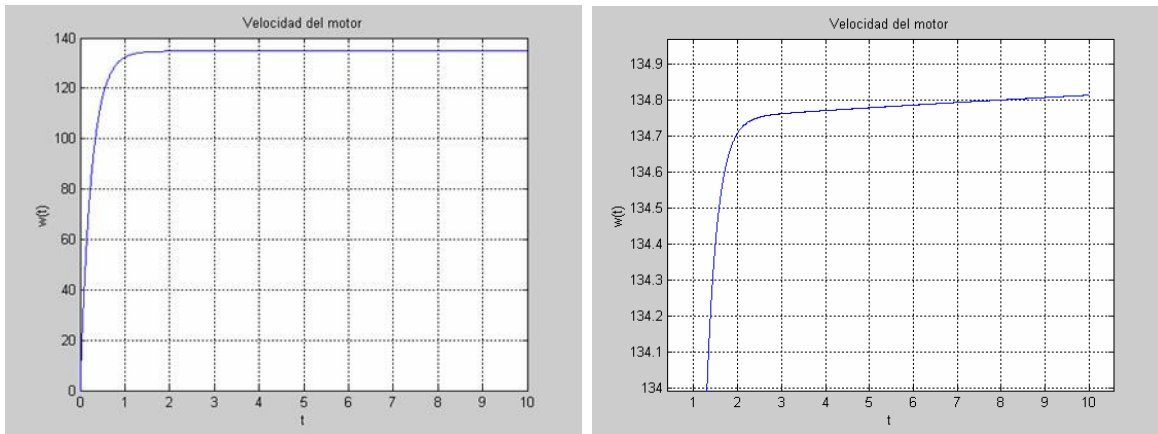


Figura 5.17: Gráfica de la velocidad del motor usando control PID. Ampliación de la Gráfica.

La acción derivativa tiene la facultad de anticiparse al error en una muestra dentro del ciclo de control, situación que permite lograr un mejor desempeño, esto de manera adicional a la acción proporcional e integral se asume un mejor resultado sobre el control de velocidad, la dificultad esta en que en los controles de velocidad con un controlador de estas características requieren de la estimación de la derivada del estado, en este caso de la aceleración, y en tiempo real implica la existencia de ruido inherente a la aproximación de la aceleración a partir de la velocidad, en esta simulación digital el error de estado estable es ligeramente menor que en los casos anteriores.

Control PX.

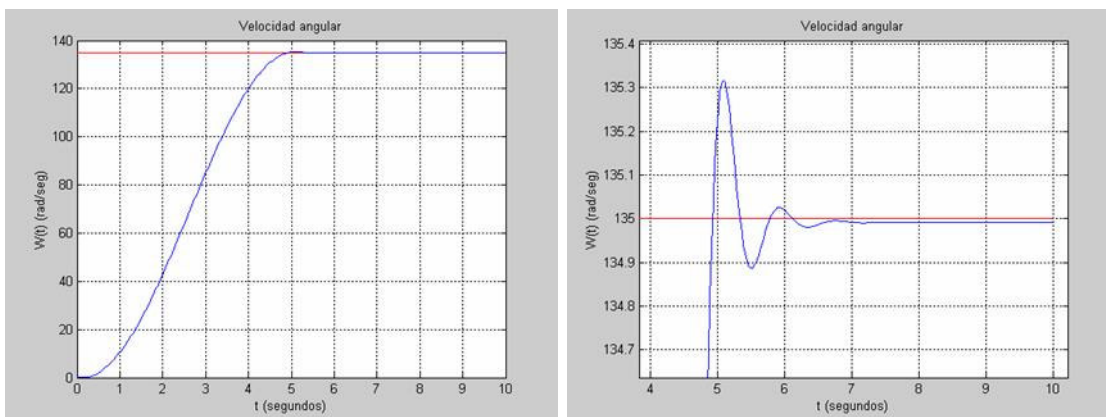


Figura 5.18: Gráfica de la velocidad del motor usando control PX. Ampliación de la Gráfica.

Podemos apreciar el error de estado estable en la aplicación de este control a nivel simulación, el error es casi nulo. Las bondades de este controlador son en que no requiere del conocimiento de la derivada del estado, ni emplear una acción de integración,

logrando resultados benéficos. Tal y como ilustra el diagrama de bloques de la figura 5.14.

Conclusiones

Del estudio anterior de simulaciones de los diferentes controles se obtienen las siguientes observaciones.

Para nuestro caso de estudio la señal requerida es la velocidad por lo que al implementar los controles PD, PI, PID, no se muestra un mejoría progresiva, sino que nos damos cuenta que el control PD nos proporciona 133.58 rpm, los controles PI y PID, 134.8 aproximadamente, esto es por la implementación de la derivada del error en los controles PD y PID, por lo que concluimos que el control PI es mejor.

Pero con la implementación del control PX, se optimiza la respuesta de salida. Superando el resultado que se obtiene con el control PI.

Por lo tanto y de acuerdo a las simulaciones, queda demostrada la eficiencia del control implementado.

CAPÍTULO 6

INTEGRACIÓN DE LA PLATAFORMA EXPERIMENTAL

INTRODUCCIÓN

En este capítulo se presenta la integración de la plataforma y las gráficas de los experimentos para cada control correspondientes a la velocidad, señal de control, error de velocidad, derivada del error y retardo del sistema, respectivamente que demuestran el comportamiento real del motor.

- El diagrama de bloques del sistema.
- La tarjeta de adquisición de datos.
- La interfaz electrónica.
- La interfaz gráfica.
- Segmentos de código de los programas en Delphi 5.
- Experimentos correspondientes a las acciones de control PD, PI, PID, PID NO LINEAL y PX con aceleración estimada, regulación y seguimiento.

6.1. Diagrama de bloques del sistema.

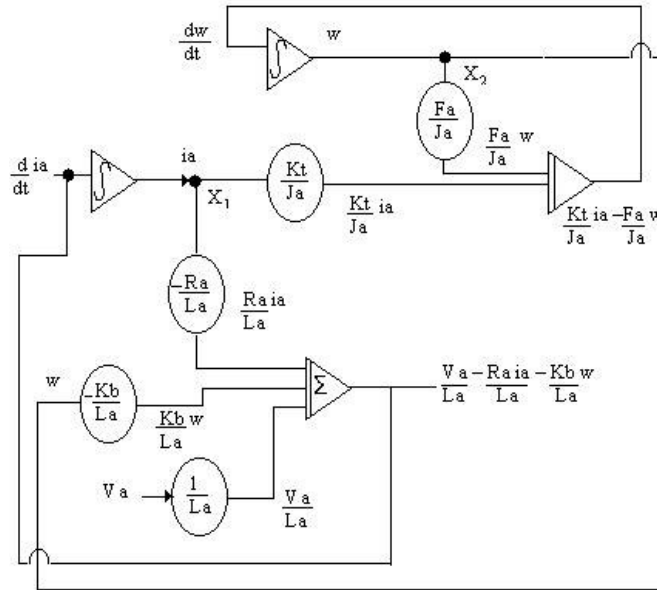


Figura 6.1: Diagrama de bloques del sistema.

La representación en variables de estado del modelo matemático del motor de cd esta dado a continuación:

$$V_a = R_a i_a + L_a \frac{di_a}{dt} + V_b \tag{6.1.1}$$

$$V_b = K_b w \tag{6.1.2}$$

$$T = J_a \frac{dw}{dt} + f_a w \tag{6.1.3}$$

$$w = \frac{d\theta}{dt} \tag{6.1.4}$$

$$T = K_t i_a \tag{6.1.5}$$

$$\frac{di_a}{dt} = \frac{V_a}{L_a} - \frac{R_a}{L_a} i_a - \frac{kb}{L_a} w \quad (6.1.6)$$

$$\frac{dw}{dt} = \frac{Kt}{J_a} i_a - \frac{f_a}{J_a} w \quad (6.1.7)$$

Donde:

- R_a = Resistencia del devanado de el inducido en ohmios.
- L_a = Inductancia del devanado de el inducido en henrios.
- i_a = Corriente del devanado de el inducido en amperios.
- V_a = Tensión aplicada a la armadura en voltios.
- V_b = Fuerza contra- electromotriz en voltios.
- θ = Desplazamiento angular del eje del motor en radianes.
- T = Par desarrollado por el motor en libras-pie.
- Ja = Inercia de armadura Henrios.
- fa = Fricción de armadura Nm/rad.

Si los estados son:

$$x_1 = i_a$$

$$x_2 = w$$

entonces se obtienen las siguientes ecuaciones.

$$\dot{x}_1 = \frac{V_a}{L_a} - \frac{R_a}{L_a} x_1 - \frac{kb}{L_a} x_2 \quad (6.1.8)$$

$$\dot{x}_2 = \frac{kt}{J_a} x_1 - \frac{f_a}{J_a} x_2 \quad (6.1.9)$$

6.2. Sistema de adquisición de datos.



Figura 6.2: Tarjeta de adquisición de datos SCB 100 de National Instruments.[22]

En la figura 6.2 se muestra la tarjeta de adquisición de datos que forma parte de nuestra interfaz, esta tarjeta nos proporciona una señal de salida analógica que habilita el modulador por ancho de pulso de nuestra interfaz electrónica, y a su vez recibe una señal analógica proporcionada por el generador taquimétrico que es el voltaje de salida que será acondicionado para obtener la velocidad por medio del control respectivo. Los datos de las terminales de conexión de la tarjeta de adquisición de datos se encuentran en el apéndice 3.

6.3. Interfaz electrónica y acondicionamiento de señales.

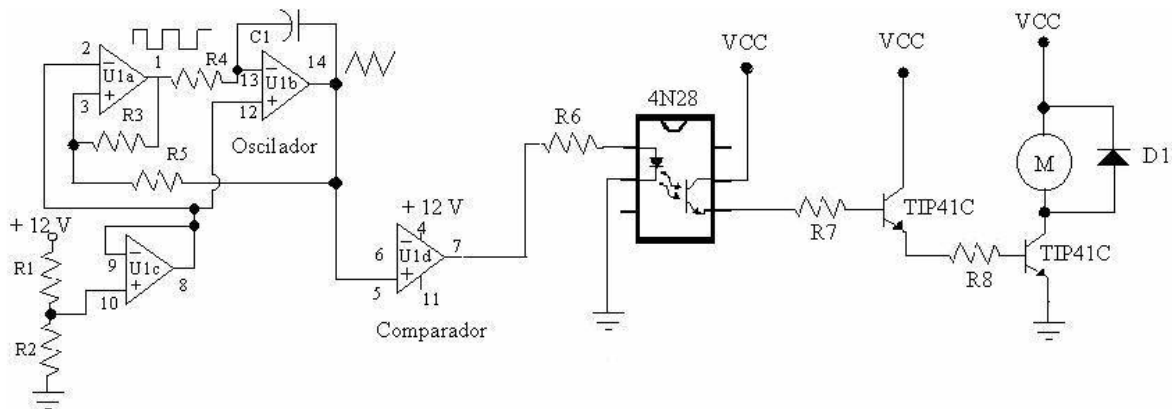


Figura 6.3: Interfaz electrónica.

En la figura 6.3 se muestra el diagrama electrónico que vamos a usar, U1a y U1b generan una forma de onda de Cuadrada / triangular con una frecuencia de alrededor 400 hertz. Uc genera una corriente que se utiliza como tierra para el oscilador. U1d realiza la configuración de la comparación y genera la variación del ancho de pulso. La Terminal de conexión 6 de U1 recibe un voltaje variable que es la señal de salida de la tarjeta de adquisición de datos. Ésta se compara a la forma de onda del triángulo de U1-14. Cuando la forma de onda esta sobre el voltaje de la entrada 6, U1 produce una salida alta. Inversamente, cuando la forma de onda está debajo del voltaje de la entrada 6, U1 produce una salida baja. La señal de salida pasa por un optoacoplador para evitar que se dañe la tarjeta en caso de una sobrecarga, llegando así a los transistores que amplifican la señal para alimentar al motor.[19]

6.4. Interfaz gráfica de usuario.



Figura 6.4: Interfaz de usuario creada en Delphi 5.

En la figura 6.4 observamos los distintos controles que componen la interfaz de usuario realizada en Delphi 5 usando algunos componentes ActiveX de la tarjeta de adquisición de datos, que son una carátula de un tacómetro en el que se representa la velocidad alcanzada por el motor, un interruptor que acciona el programa, un elemento para poder variar la velocidad deseada, otro para ajustar la frecuencia natural, y un botón para guardar el comportamiento del motor en un programa de Matlab 6. [ver apéndice2]

6.5. Segmento del código fuente.

```
procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

function fudt:Single;
var kp,kd,L:Single;
begin
L:= eL.FloatValue;
kp:=wn.FloatValue*wn.FloatValue;
kd:=2*Wn.FloatValue;
e[varN]:=avel[varN]-vd.FloatValue;
ep[varN]:=-L*avel[varN]+kd*e[varN];
result:=-kp*e[varN]-kd*ep[varN];
end;

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:= avel[varN];
```

El segmento de programa del control PX, muestra la declaración de variables, una función en la cual se realiza la acción de control, la variable donde se captura y almacena una señal que se convierte a velocidad en radianes por segundo y la visualización de la velocidad por medio de un tacómetro. El programa completo se encuentra en el apéndice 2 y esta desarrollado en lenguaje Delphi 5.

6.6. Control PD con aceleración estimada y regulación.

Los parámetros utilizados resultan de la experimentación y en base a criterios preestablecidos, estos son:

$$W_n = 4$$

$$V_d = 135 \text{ rpm.}$$

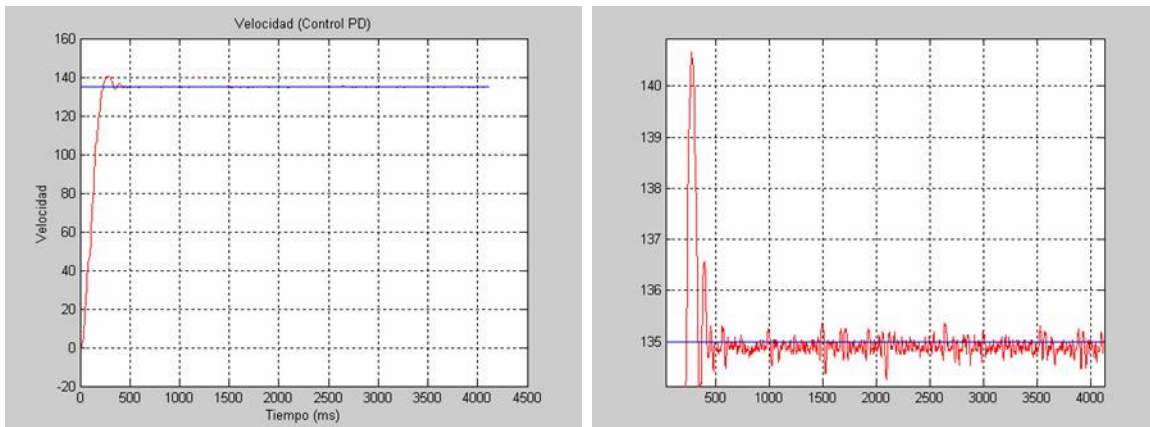


Figura 6.5: Gráfica de velocidad, ampliación de la señal, de control PD.

En la figura 6.5 a la izquierda observamos la velocidad del motor aplicando un control PD, se puede apreciar un sobre impulso al llegar al valor de referencia y luego establecerse en un valor aproximado a 135 rpm como podemos ver a la derecha en la ampliación de la misma gráfica, existe una señal con una frecuencia alta, esto es debido a la estimación de la aceleración a partir de la velocidad y más aún por el ruido propiciado por el generador taquimétrico, quien proporciona señales indeseables.

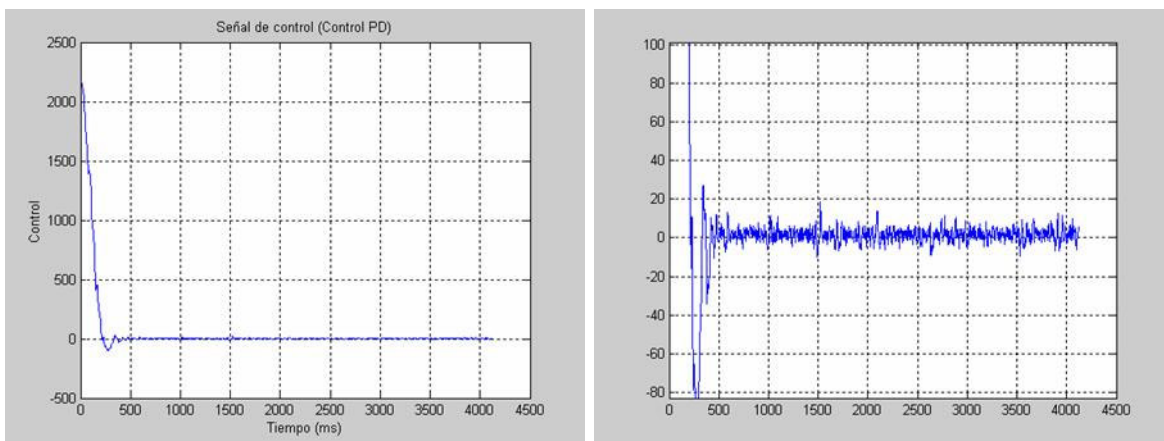


Figura 6.6: Gráfica de la señal de control, ampliación de la señal, de control PD.

En la figura 6.6 a la izquierda observamos la Señal de control aplicada al motor con un control PD, se puede apreciar como el valor de $u(t)$ es muy elevado en un principio y luego se estabiliza en un valor aproximado a cero como podemos ver a la derecha en la ampliación de la misma gráfica. El transitorio inicial de la señal de control es alto debido a que intenta superar el estado inercial del reposo y llevar al motor a la velocidad de referencia, cuando se ha alcanzado esta el control se estabiliza, únicamente compensando el ruido del generador taquimétrico.

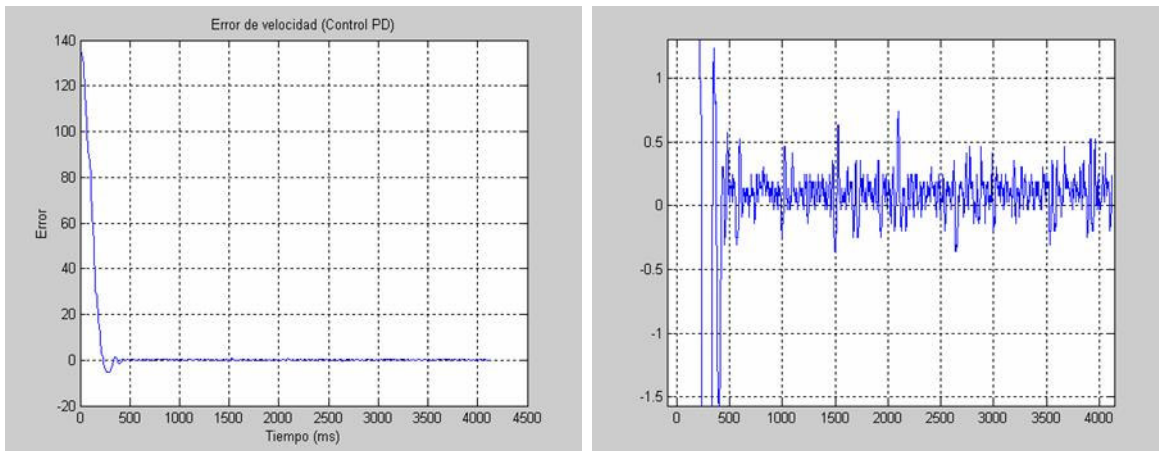


Figura 6.7: Gráfica de error de velocidad, ampliación de la señal, de control PD.

La señal de error permite apreciar la convergencia asintótica del controlador, la gráfica amplificada de la derecha es el error a una frecuencia derivado de la estimación de la derivada de la velocidad y del ruido generado en el sensor. Sin embargo es posible apreciar la contribución del controlador.

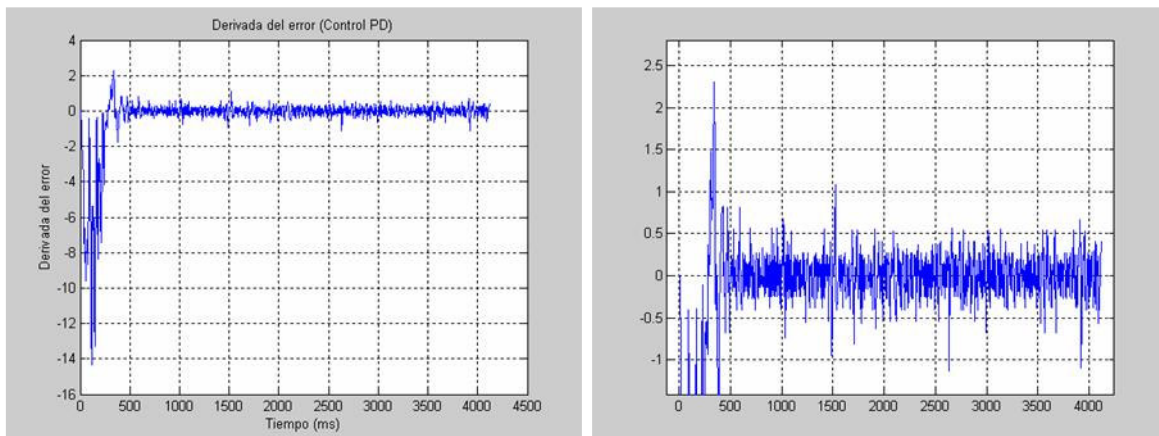


Figura 6.8: Gráfica de la derivada del error, ampliación de la señal, de control PD.

La derivada del error, implica la estimación de la derivada del estado, es decir una señal estimada a partir de una señal con ruido propiciado en el generador taquimétrico, sin

embargo es la parte de anticipación al error que intenta resolver el problema de convergencia al valor de velocidad deseado.

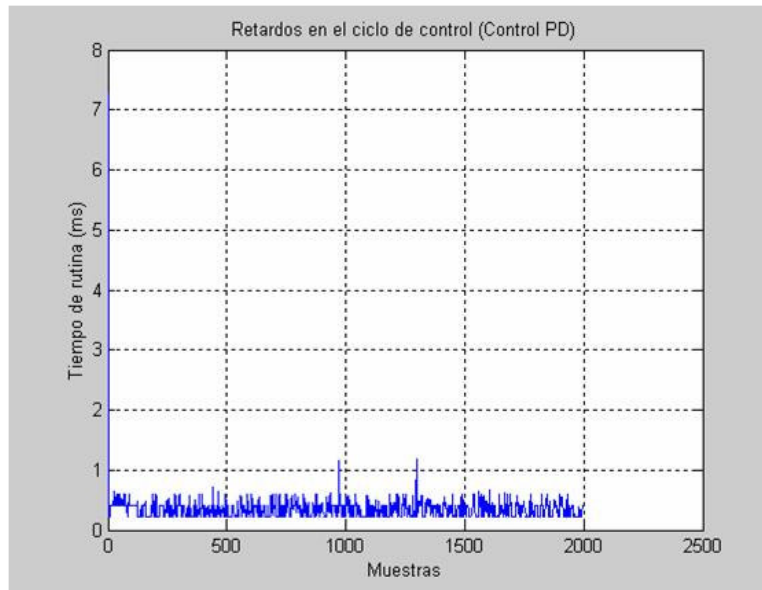


Figura 6.9: Gráfica de los retardos del sistema.

Los retardos presentados en esta gráfica son en promedio de 0.5 milisegundos, situación que beneficia en la tarea de control, propiciada a un milisegundo de tiempo de muestreo, es decir el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea aun en la plataforma de Windows.

6.7. Control PI con regulación.

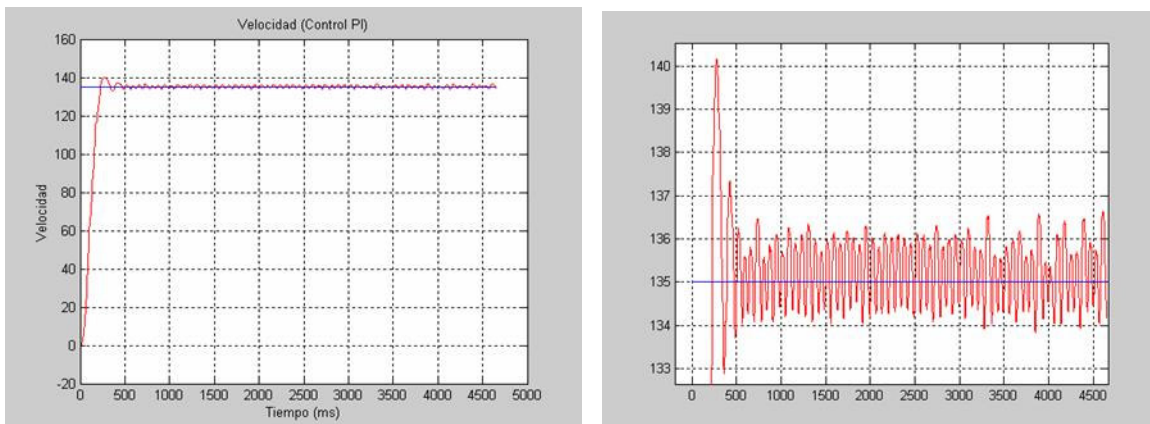


Figura 6.10: Gráfica de velocidad, ampliación de la señal, de control PI.

En la figura 6.10 a la izquierda observamos la velocidad del motor aplicando un control PI, se puede apreciar un sobre impulso al llegar al valor de referencia y luego establecerse en un rango que varía sobre los 135 rpm (134.3 rpm a 136.3 rpm), como podemos ver a la derecha en la ampliación de la misma gráfica. La acción de integración de la señal nos proporciona una salida un poco más estable pero aun con variaciones considerables.

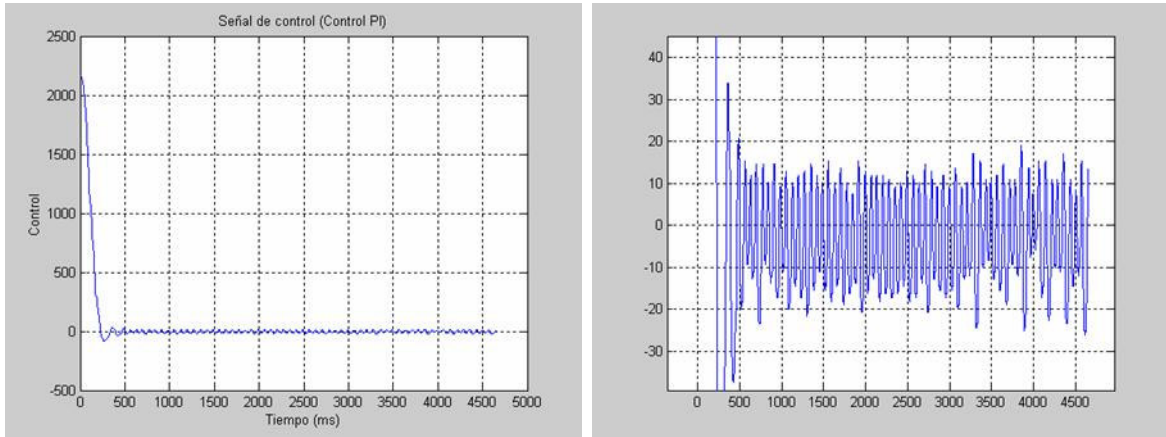


Figura 6.11: Gráfica de señal de control, ampliación de la señal, de control PI.

En la figura 6.11 a la izquierda observamos la señal de control aplicada al motor con un control PI, el valor de $u(t)$ es muy elevado en un principio y luego se estabiliza en un rango que varía sobre cero como podemos ver a la derecha en la ampliación de la gráfica. Debido a que se encuentra en reposo, el estado transitorio inicial de la señal de control es alto y trata de ajustarse a la velocidad de referencia, cuando este la alcanza el control se estabiliza, por medio de la acción de integración.

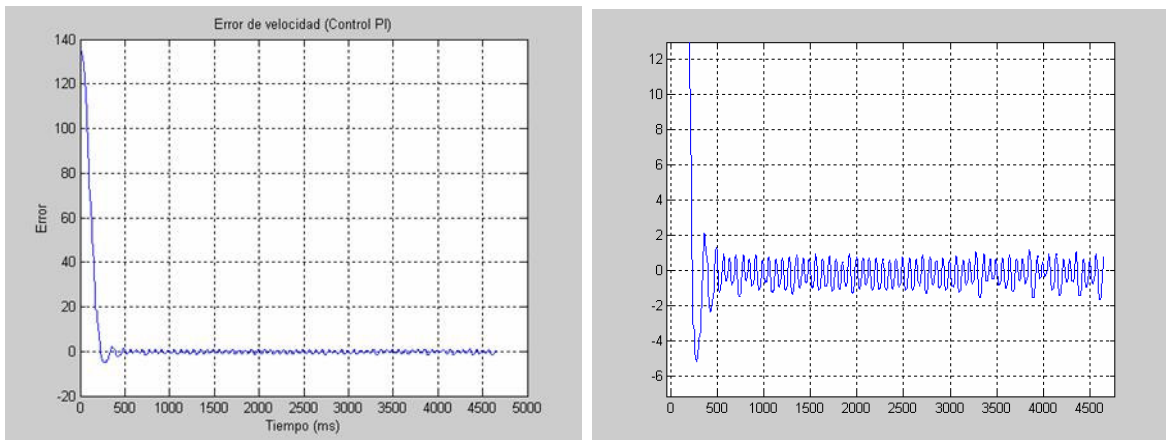


Figura 6.12: Gráfica del error de velocidad, ampliación de la señal, de control PI.

Podemos observar que la acción integral suprime el error en estado estable, la gráfica ampliada de la derecha es el error a una frecuencia derivado de la estimación de la integral de la velocidad y del ruido generado en el sensor.

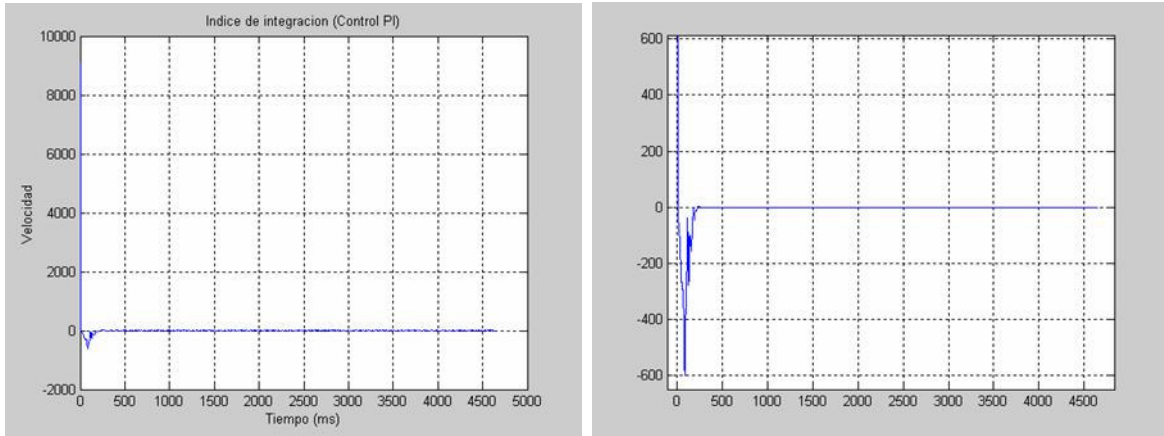


Figura 6.13: Gráfica del índice de integración, ampliación de la señal, de control PI.

La figura 6.13 muestra el índice de integración de la señal de control, se puede observar la variación que tiene la señal en un principio y la estabilidad en cero después de los 500 ms.

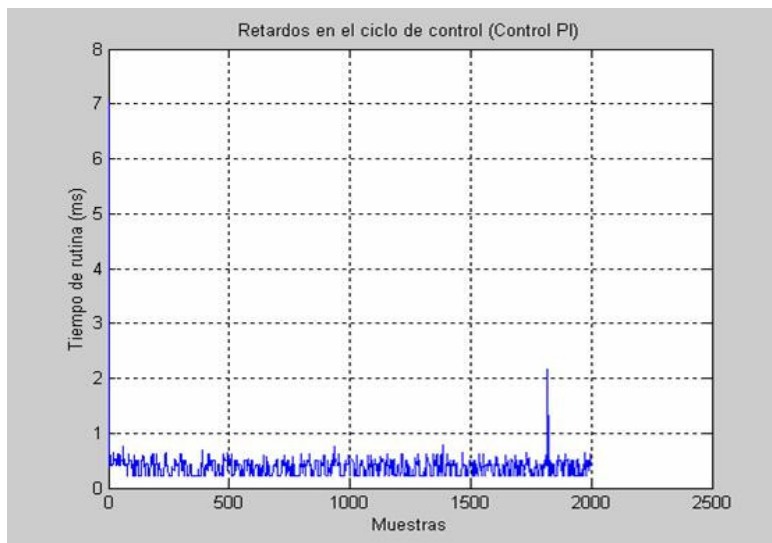


Figura 6.14: Gráfica de los retardos del sistema.

Los retardos del sistema son en promedio de 0.5 milisegundos, esto beneficia a la tarea de control, que es propiciada a 1 milisegundo de tiempo de muestreo, esto es que el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea.

6.8. Control PD con aceleración estimada y regulación basada en seguimiento.

Los parámetros utilizados en el control PD con Seguimiento, resultan de la experimentación y en base a criterios preestablecidos, estos son:

$$W_n = 4$$

$$V_d = 135 \text{ rpm.}$$

$$T_b = 2 \text{ Seg.}$$

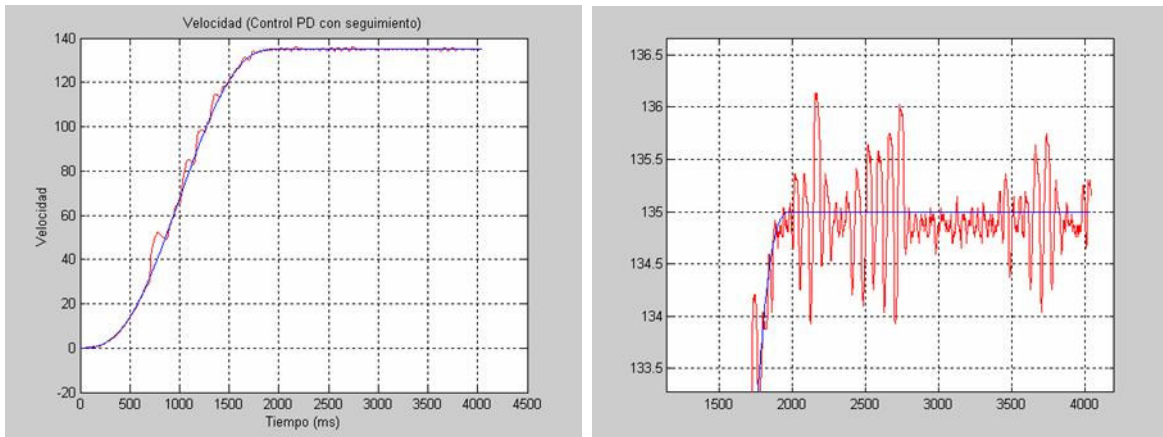


Figura 6.15: Gráfica de velocidad, ampliación de la señal de control PD con seguimiento.

En la figura 6.15 a la izquierda observamos la velocidad con seguimiento del motor con un control PD, a la derecha en la ampliación de la misma gráfica, observamos que el seguimiento es bueno con algunas perturbaciones (134 rpm a 136 rpm).

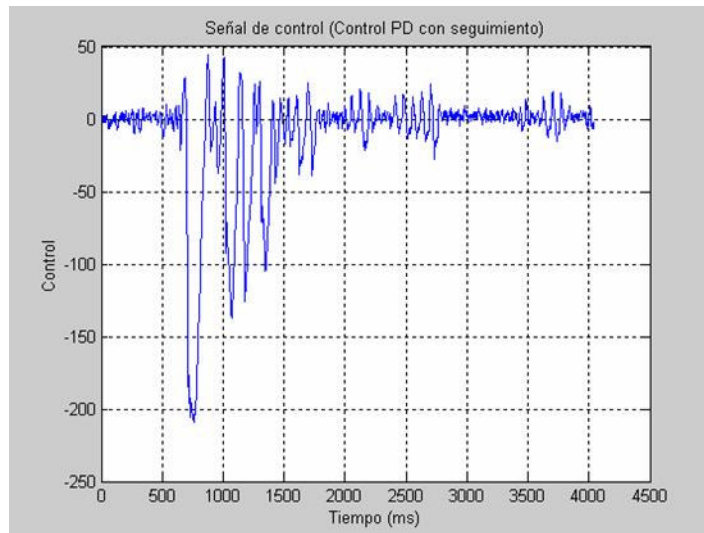


Figura 6.16: Gráfica de la señal de control; control PD.

En la figura 6.16 observamos la señal de control aplicada al motor con un control PD, se puede apreciar como actúa el control cuando el valor de $u(t)$ es bajo en un principio y luego alto, esto se puede observar también en la gráfica de la velocidad se puede ver como a los 500 ms existe un descontrol que se puede observar en las dos gráficas.

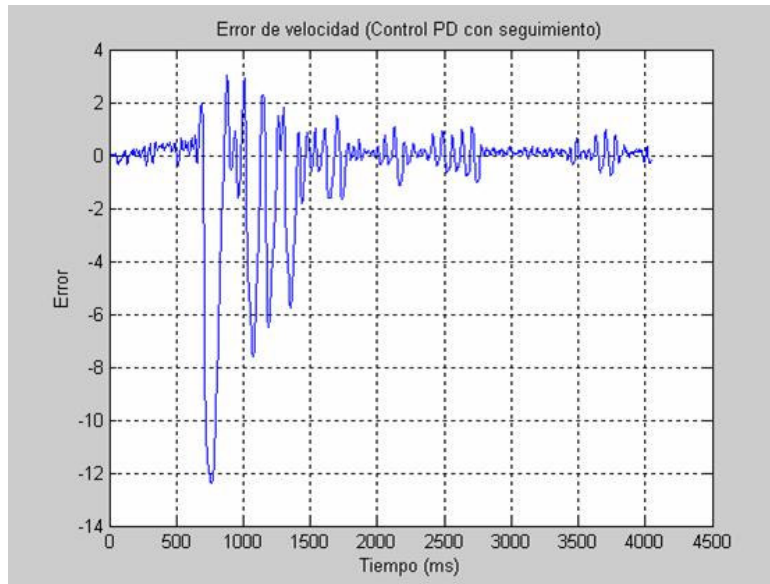


Figura 6.17: Gráfica del error basada en seguimiento, de control PD.

La señal de error permite apreciar la convergencia asintótica del controlador, la gráfica muestra el error de la señal al momento de realizar el seguimiento y después al alcanzar el valor de referencia.

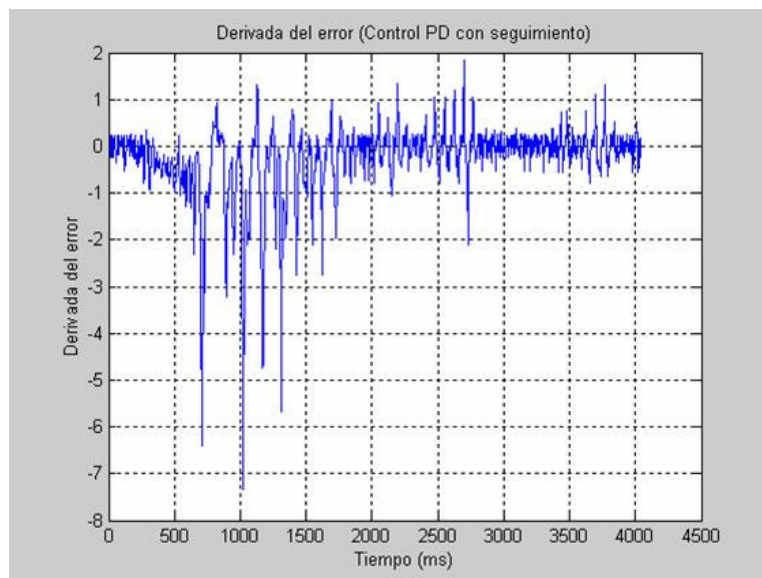


Figura 6.18: Gráfica de la derivada del error basada en seguimiento, de control PD.

La figura 6.18 muestra la derivada del error que implica la derivada del estado, sin embargo es la parte de anticipación al error que intenta resolver el problema de convergencia al valor de velocidad y seguimiento deseado.

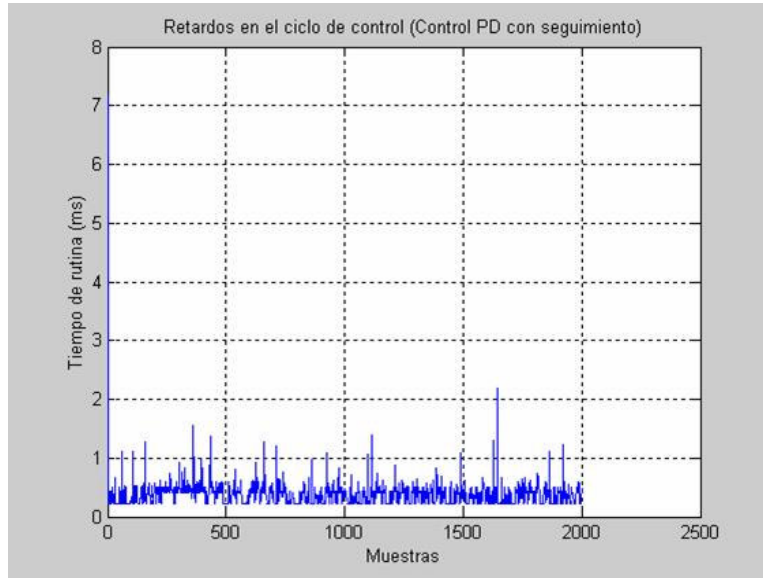


Figura 6.19: Gráfica de los retardos del sistema.

La gráfica nos muestra mayores retardos en el sistema, esto afecta a la tarea de control que es realizada a 1 milisegundo de tiempo de muestreo, esto es que el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea.

6.9. Control PI con regulación basada en seguimiento.

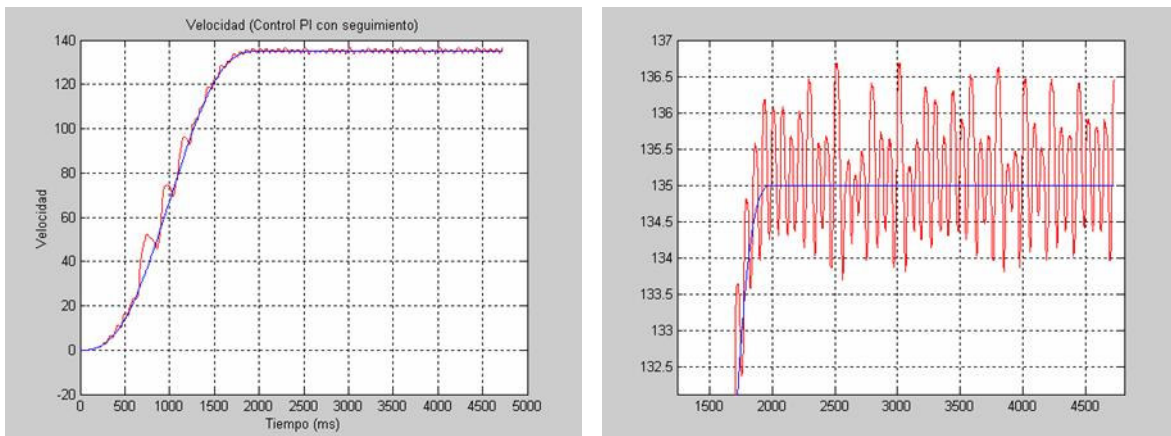


Figura 6.20: Gráfica de velocidad basada en seguimiento, de control PI.

En la figura 6.20 a la izquierda observamos la velocidad con seguimiento del motor con un control PI, a la derecha en la ampliación de la misma gráfica observamos las fallas de la

señal. Podemos observar que la acción integral suprime el error en estado estable pero no en su totalidad las variaciones (133.8 rpm a 136.6 rpm) son producto del ruido generado por el generador taquimétrico.

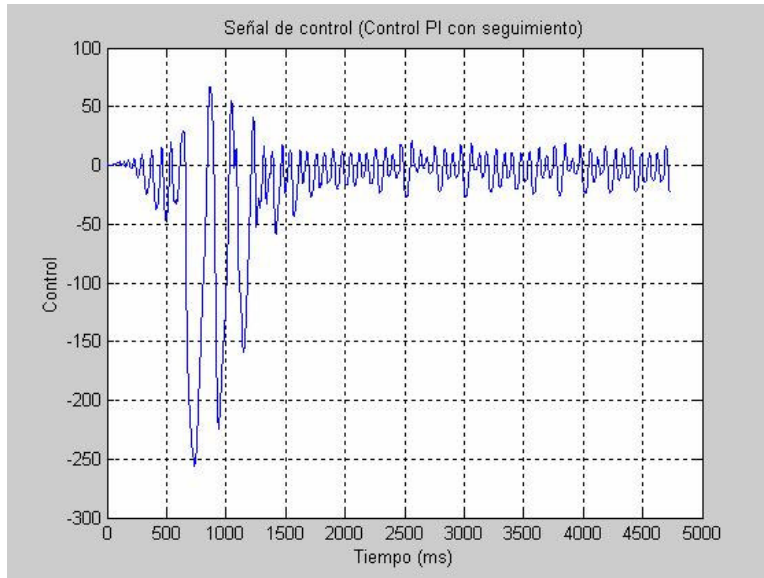


Figura 6.21: Gráfica de la señal de control; control PI con seguimiento.

En la figura 6.21 observamos la señal de control aplicada al motor con un control PI, se puede apreciar como actúa el control cuando el valor de $u(t)$ es bajo en un principio y luego alto, esto se puede observar también en la gráfica de la velocidad se puede ver como a los 750 ms aproximadamente existe un descontrol que se puede observar en las dos gráficas.

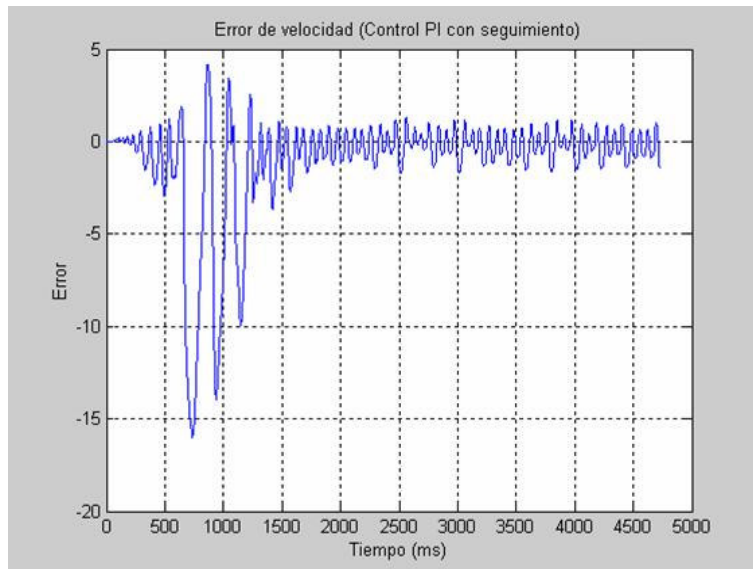


Figura 6.22: Gráfica del error, de control PI con seguimiento.

El error, implica una señal estimada a partir de una señal con ruido propiciado en el generador taquimétrico, sin embargo es la parte integral la que intenta resolver el problema de convergencia al valor de velocidad deseado.

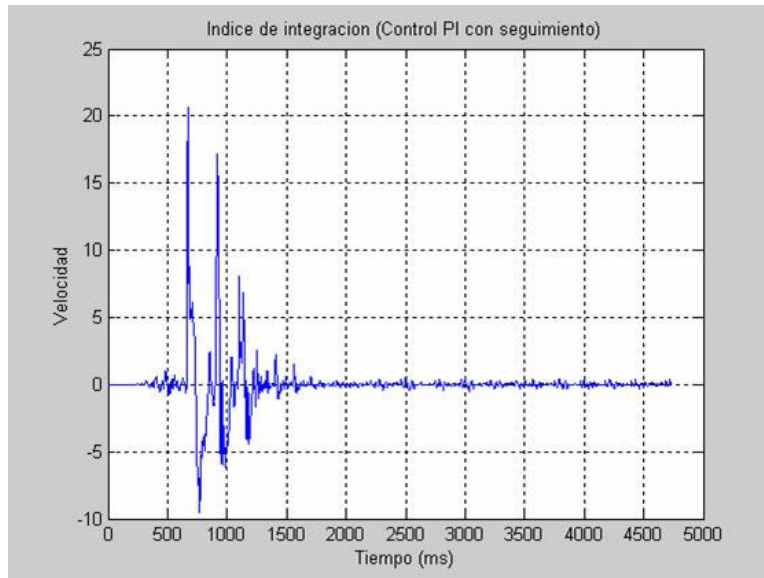


Figura 6.23: Gráfica del índice de integración, de control PI con seguimiento.

La figura nos muestra el índice de integración de la señal de control, se puede observar la variación que tiene la señal en el momento que realiza el seguimiento y la estabilidad en valores cercanos a cero después de los 1500 ms.

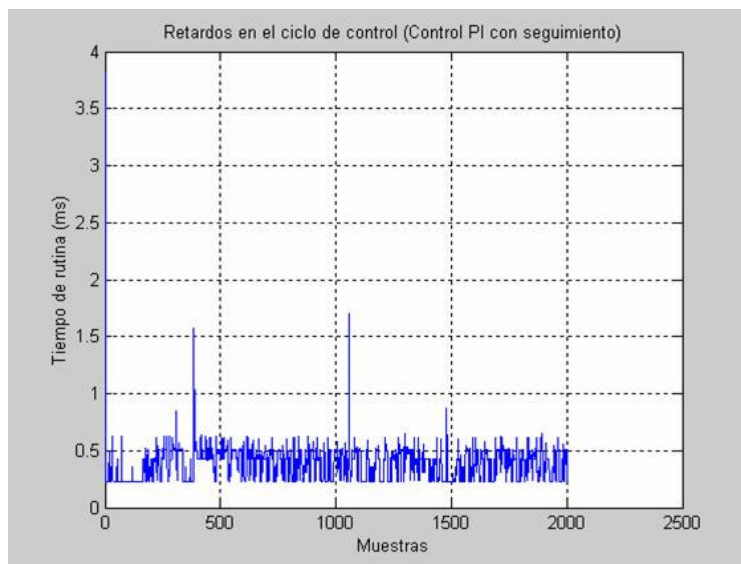


Figura 6.24: Gráfica de los retardos del sistema.

Los retardos del sistema son en promedio de 0.5 milisegundos aproximadamente, esto beneficia a la tarea de control, que es propiciada a 1 milisegundo de tiempo de muestreo, esto es que el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea.

6.10. Control PID con aceleración estimada y regulación.

Los parámetros utilizados en el control PID, resultan de la experimentación y en base a criterios preestablecidos, estos son:

$$W_n = 4$$

$$V_d = 135 \text{ rpm.}$$

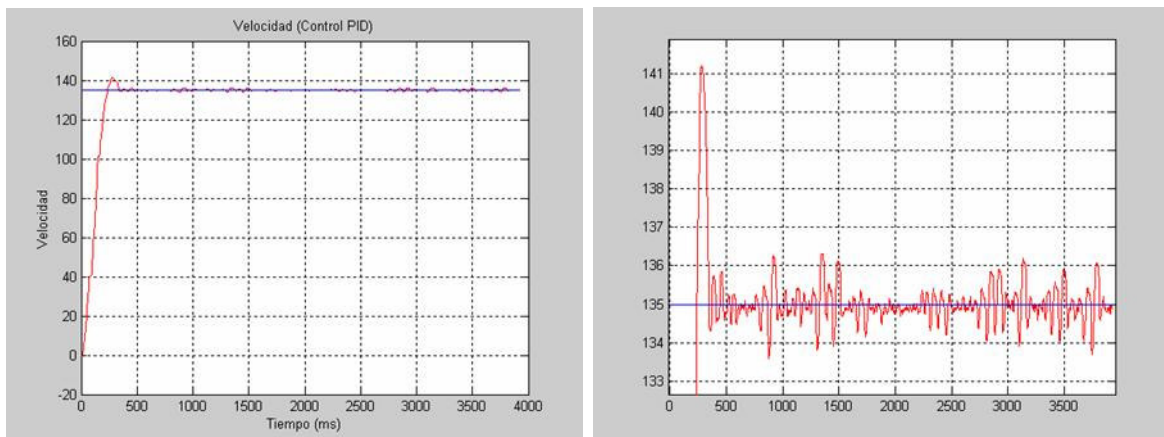


Figura 6.25: Gráfica de velocidad, ampliación a 135 rpm, de control PID.

En la figura 6.25 a la izquierda observamos la velocidad del motor aplicando un control PID, se puede apreciar un sobre impulso al llegar al valor y luego establecerse en un valor aproximado a 135 rpm con variaciones irregulares (134 rpm a 136 rpm) pero menores que en los controles anteriores, como podemos ver a la derecha en la ampliación de la misma gráfica. Esto es el reflejo de la implementación de la derivación e integración del error.

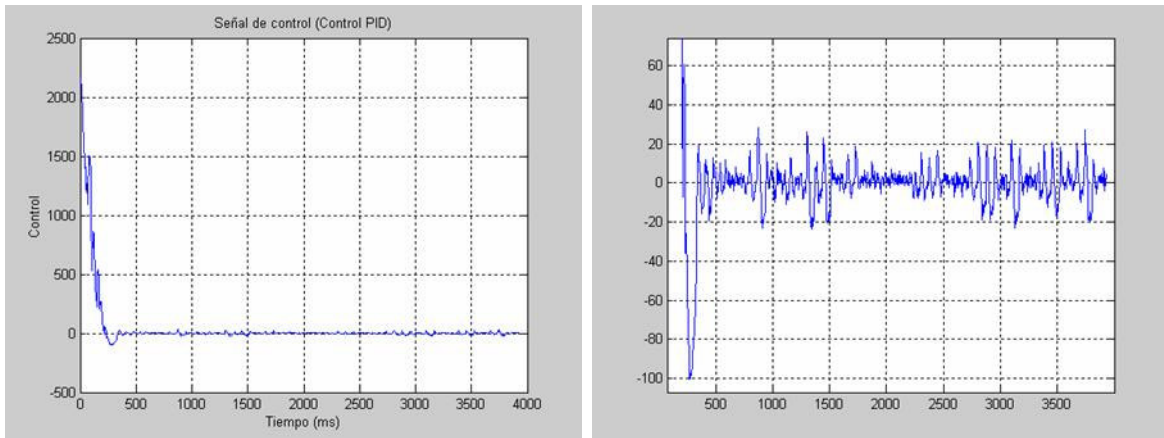


Figura 6.26: Gráfica de la señal de control; control PID.

En la figura 6.26 a la izquierda observamos la señal de control aplicada al motor con un control PID, se puede apreciar como el valor de $u(t)$ es muy alta al iniciar y después varía alrededor a cero, como podemos ver a la derecha en la ampliación de la misma gráfica.

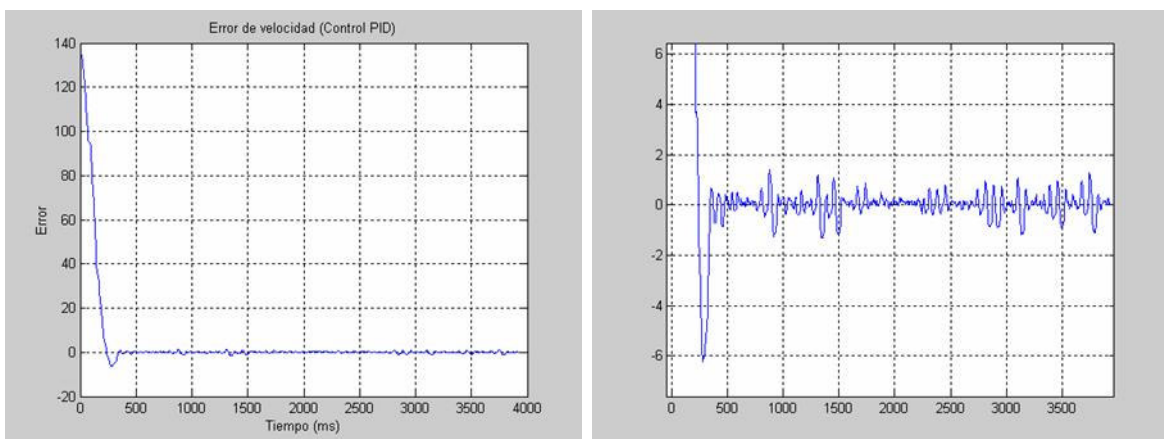


Figura 6.27: Gráfica del error de velocidad; control PID.

Podemos observar que la acción integral y la acción derivativa en conjunto suprimen el error en estado estable, como podemos ver en la ampliación de la grafica a la derecha es considerablemente menor que en los controles PD o PI.

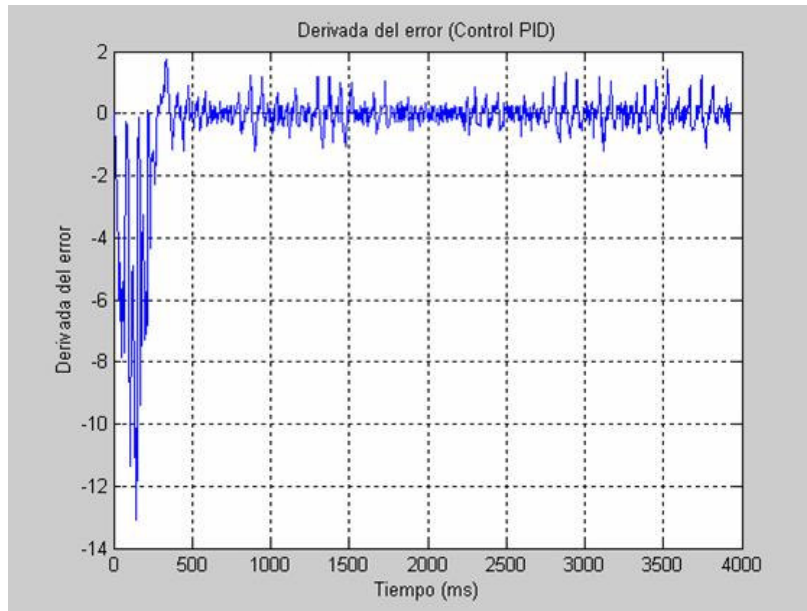


Figura 6.28: Gráfica de la Derivada del error; control PID.

La derivada del error, implica la estimación de la derivada del estado, es decir una señal estimada a partir de una señal con ruido propiciado en el generador taquimétrico, sin embargo es la parte de anticipación al error y aunado a esto la acción integral que intentan resolver el problema de convergencia al valor de velocidad deseado.

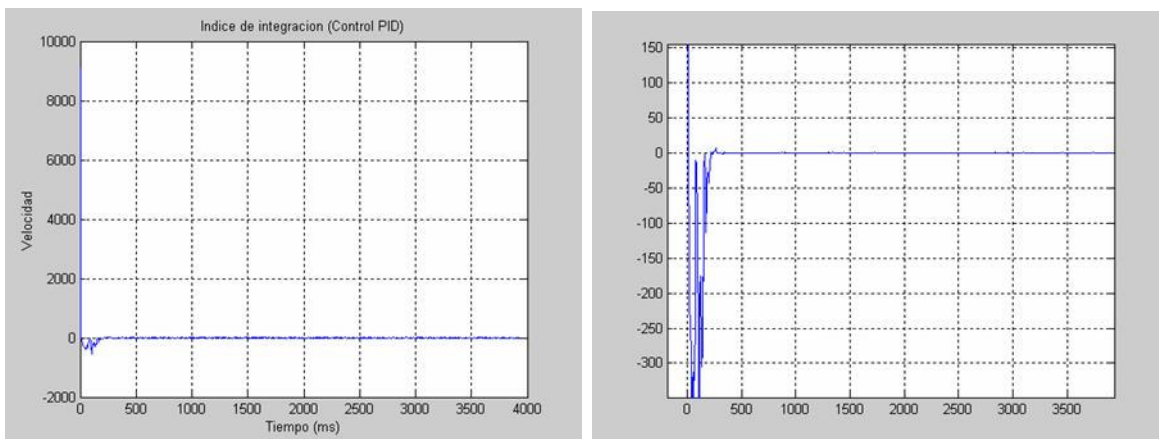


Figura 6.29: Gráfica del índice de integración; control PID.

La figura nos muestra el índice de integración de la señal de control, se puede observar la variación tan alta que tiene la señal al intentar superar el estado inercial del reposo y llevar al motor a la velocidad de referencia y la estabilidad en valores cercanos a cero después de los 500 ms.

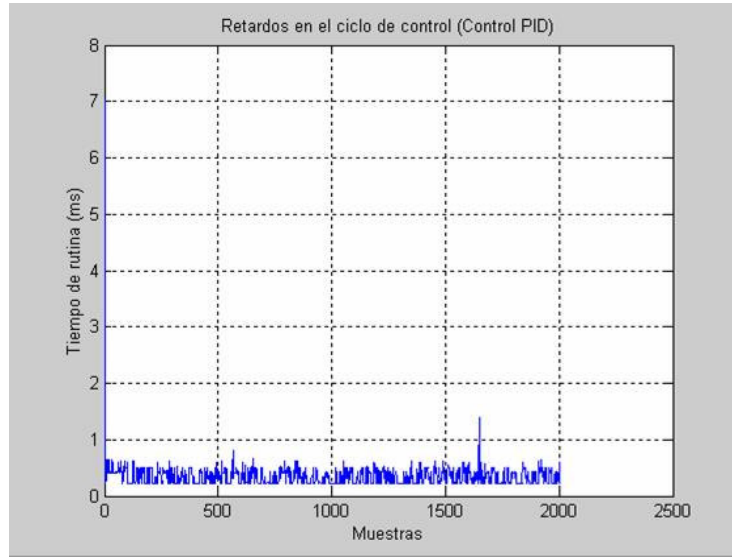


Figura 6.30: Gráfica de los retardos del sistema.

Los retardos presentados en esta gráfica son en promedio de 0.5 milisegundos, situación que beneficia en la tarea de control, propiciada a 1 milisegundo de tiempo de muestreo, es decir el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea aun en la plataforma de Windows.

6.11. Control PID con regulación basada en seguimiento.

Los parámetros utilizados en el control PID con Seguimiento, resultan de la experimentación y en base a criterios preestablecidos, estos son:

- $W_n = 4$
- $V_d = 135 \text{ rpm.}$
- $T_b = 2 \text{ Seg.}$

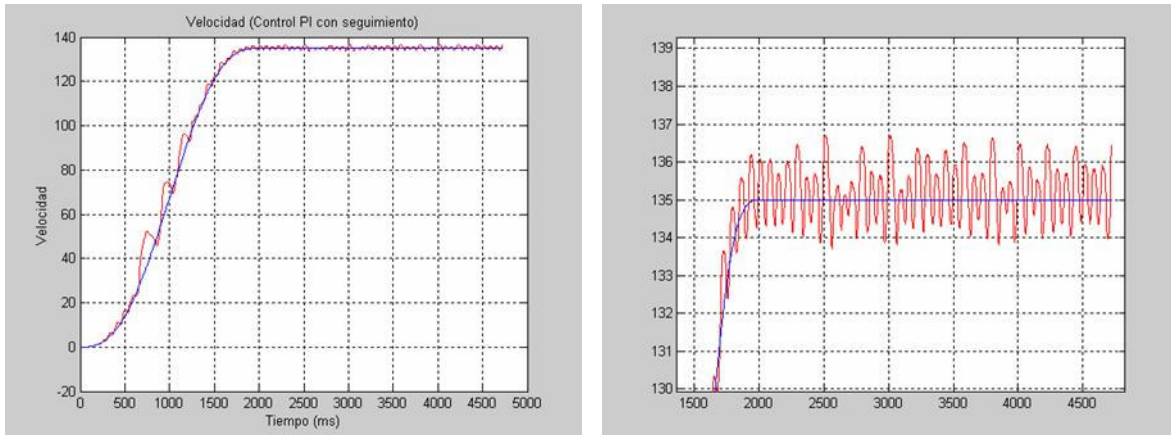


Figura 6.31: Gráfica de velocidad, ampliación a 135 rpm, de control PID con seguimiento.

En la figura 6.31 a la izquierda observamos la velocidad con seguimiento del motor con un control PID, a la derecha en la ampliación de la misma gráfica observamos las fallas de la señal y los rangos (134 rpm a 136 rpm) en la señal al momento de realizar el seguimiento y los beneficios que proporciona el control.

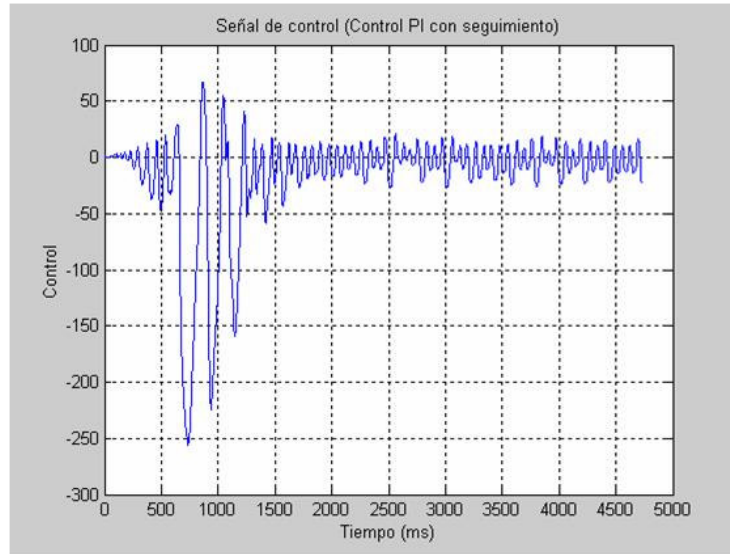


Figura 6.32: Gráfica de la señal de control; control PID con seguimiento.

En la figura 6.32 observamos la Señal de control aplicada al motor con un control PID con seguimiento, se puede apreciar como actúa el control cuando el valor de $u(t)$ es bajo en un principio y luego alto, esto se puede observar también en la gráfica de la velocidad se puede ver como a los 750 ms aproximadamente existe un descontrol que se puede observar en las dos gráficas.

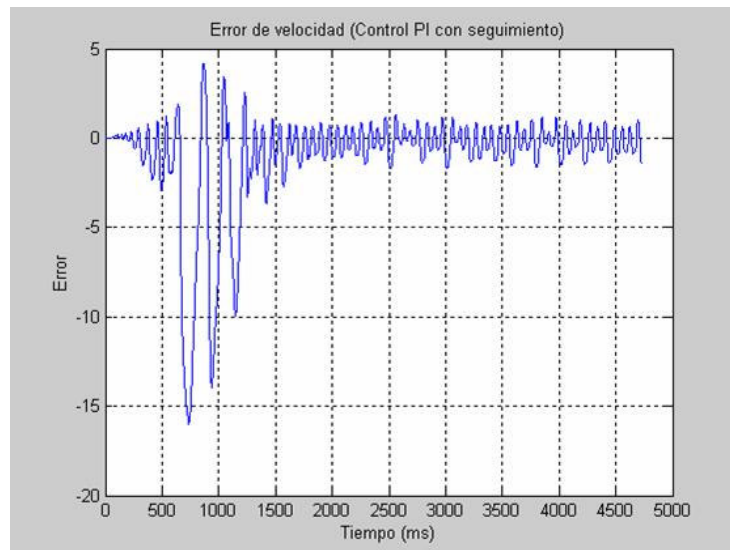


Figura 6.33: Gráfica del error de velocidad; control PID con seguimiento.

Podemos observar que la acción integral y la acción derivativa en conjunto suprimen el error en estado estable, como podemos ver en la grafica el comportamiento del error de estado estable es considerablemente menor que en los controles PD o PI.

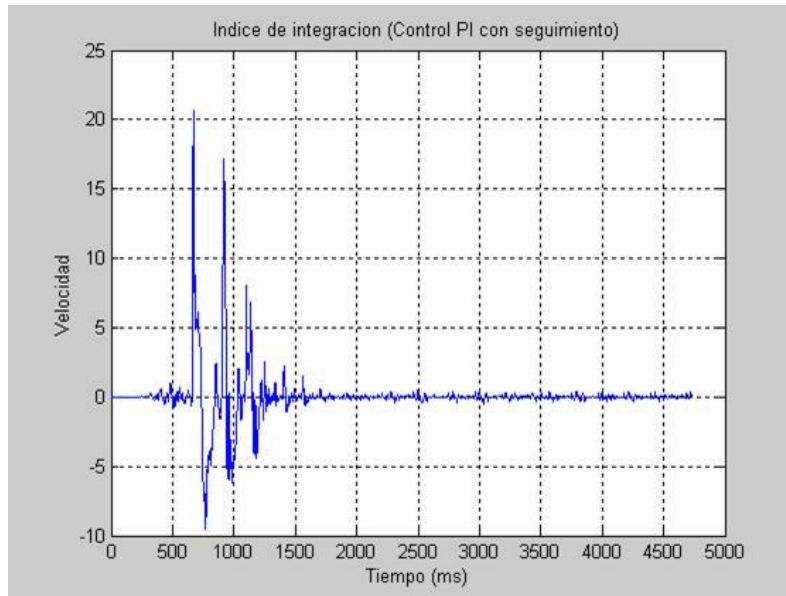


Figura 6.34: Gráfica del índice de integración; control PID con seguimiento.

La figura nos muestra el índice de integración de la señal de control al momento de realizar un seguimiento, se puede observar la variación tan alta que tiene la señal al realizar el seguimiento y llevar al motor a la velocidad de referencia y la estabilidad en valores cercanos a cero después de los 1500 ms.

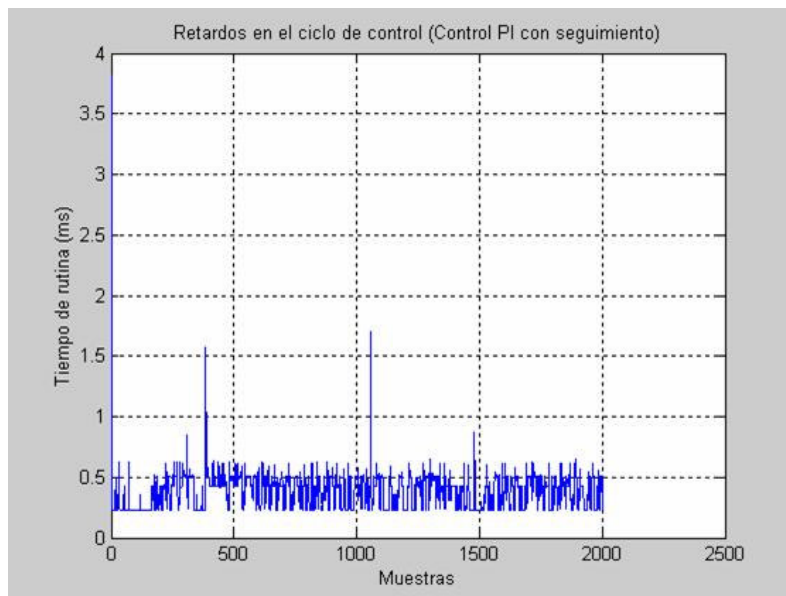


Figura 6.35: Gráfica de los retardos del sistema.

Los retardos presentados en esta gráfica son en promedio de 0.6 milisegundos, situación que beneficia en la tarea de control, propiciada a 1 milisegundo de tiempo de muestreo, es decir el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea aun en la plataforma de Windows.

6.12. Control PID NO LINEAL con aceleración estimada y regulación.

Los parámetros utilizados en el control PID NL resultan de la experimentación y en base a criterios preestablecidos, estos son:

$$W_n = 1.84$$

$$V_d = 135 \text{ rpm.}$$

$$K_i = 100$$

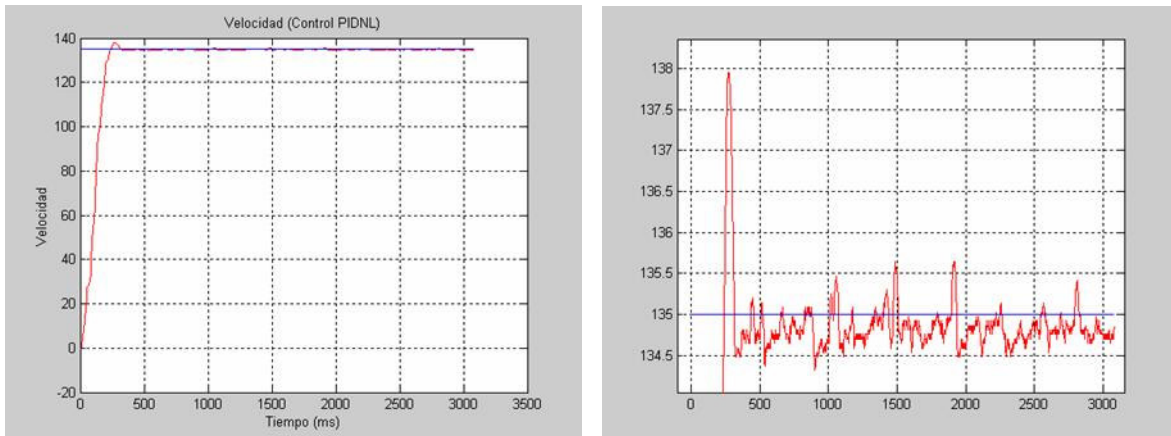


Figura 6.36: Gráfica de velocidad, ampliación a 135 rpm, de control PID NL.

En la figura 6.36 a la izquierda observamos la velocidad del motor aplicando un control PID NL, se puede apreciar un sobre impulso al llegar al valor y luego establecerse en un valor aproximado a 135 rpm con variaciones irregulares (de 134.5 rpm a 135.5 rpm) pero que varían en un rango menor que en los otros controles, como podemos ver a la derecha en la ampliación de la misma gráfica.

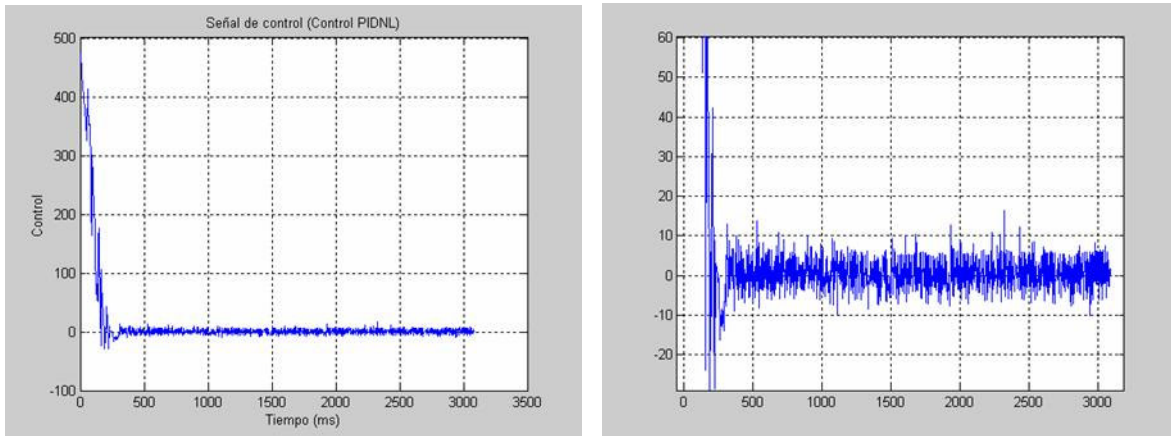


Figura 6.37: Gráfica de la señal de control; control PID NL.

En la figura 6.37 a la izquierda observamos la señal de control aplicada al motor con un control PID NL, se puede apreciar como el valor de $u(t)$ es muy elevado en un principio y luego varía alrededor a cero con mucho mayor inestabilidad en la señal, como podemos ver a la derecha en la ampliación de la misma gráfica. Este es en parte por el ruido propiciado por el generador taquimétrico, quien proporciona señales indeseables.

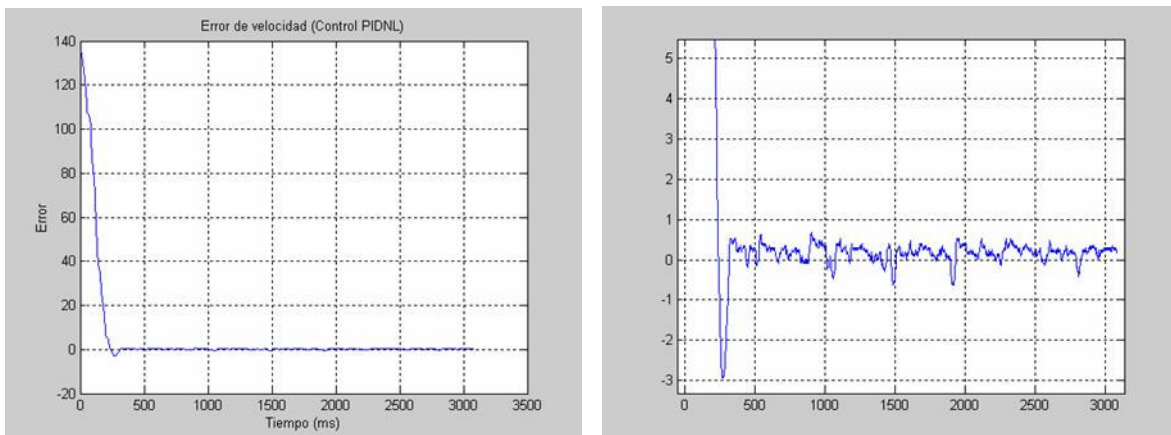


Figura 6.38: Gráfica del error de velocidad; control PID NL.

Podemos observar que la acción integral suprime el error en estado estable, además el producto $\beta\alpha$ compensa a k_p con un error en estado estable considerablemente menor que en el PID convencional.

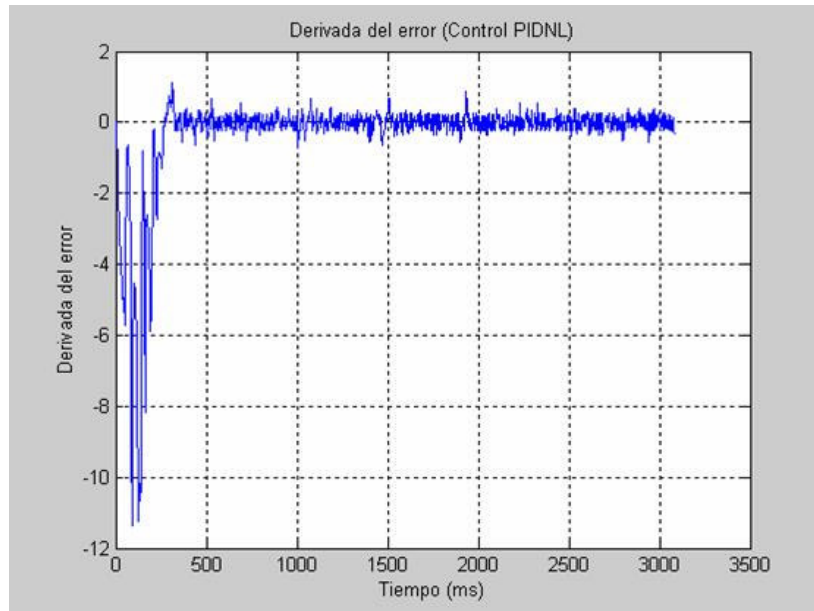


Figura 6.39: Gráfica de la derivada del error; control PID NL.

La derivada del error, implica la estimación de la derivada del estado, es decir una señal estimada a partir de una señal con ruido propiciado en el generador taquimétrico, sin embargo la señal se estabiliza en comparación con la obtenida en al control PID, es la parte de anticipación al error y aunado a esto la acción integral que intentan resolver el problema de convergencia al valor de velocidad deseado.

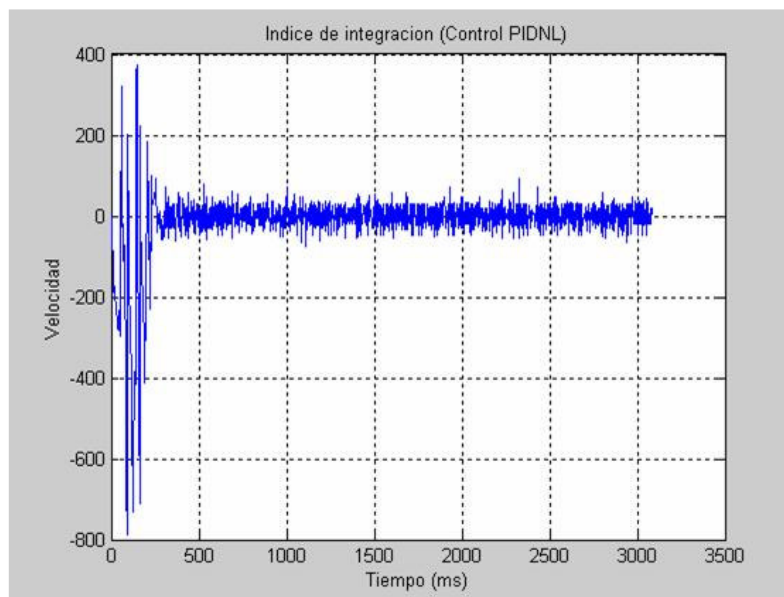


Figura 6.40: Gráfica del índice de integración; control PID NL.

La figura nos muestra el índice de integración de la señal de control, se puede observar la variación que tiene la señal en el momento que realiza el seguimiento. Existe una señal con una frecuencia alta, esto es debido a la estimación de la aceleración a partir de la

velocidad y más aún por el ruido propiciado por el generador taquimétrico, quien proporciona señales indeseables.

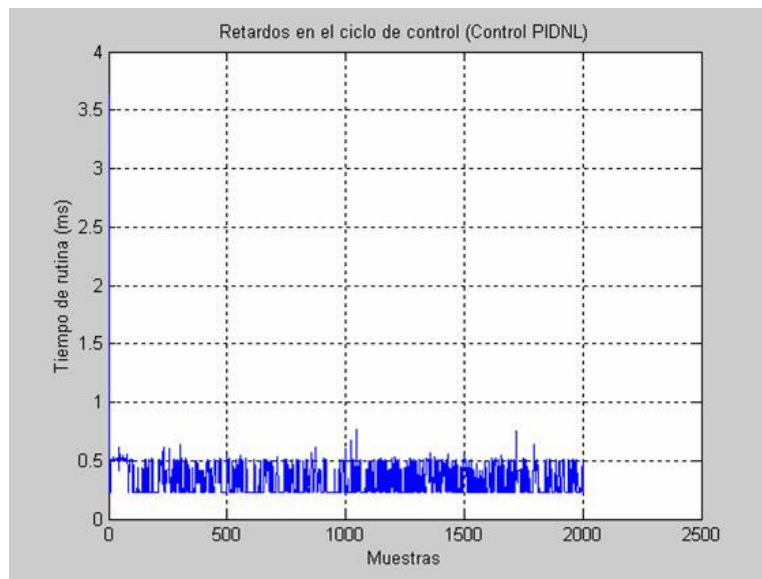


Figura 6.41: Gráfica de los retardos del sistema.

Los retardos presentados en esta gráfica son en promedio de 0.5 milisegundos, situación que beneficia en la tarea de control, propiciada a 1 milisegundo de tiempo de muestreo, es decir el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea aun en la plataforma de Windows.

6.13. Control PID NO LINEAL basada en seguimiento.

Los parámetros utilizados en el control PID NL con Seguimiento resultan de la experimentación y en base a criterios preestablecidos, estos son:

$$W_n = 1.84$$

$$V_d = 135 \text{ rpm.}$$

$$K_i = 100$$

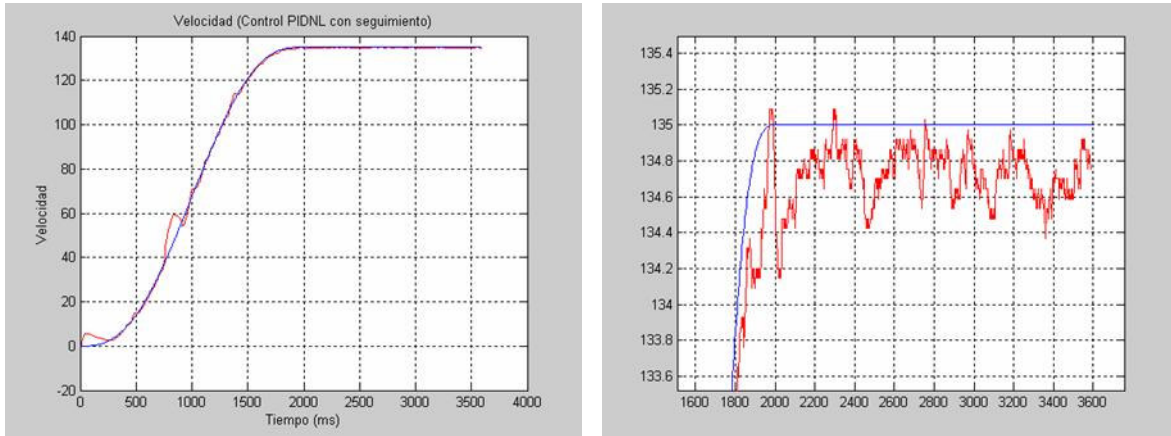


Figura 6.42 Gráfica de velocidad, ampliación a 135 rpm, del control PID NL con seguimiento.

En la figura 6.42 a la izquierda observamos la velocidad con seguimiento del motor con un control PID NL, a la derecha en la ampliación de la misma gráfica. Observamos que el seguimiento es bueno con un pequeño sobre impulso al principio y un después aproximadamente a los 800 ms, pero hay variaciones de 0.6 rpm (134.4 rpm a 135 rpm) cuando se alcanza el valor de referencia.

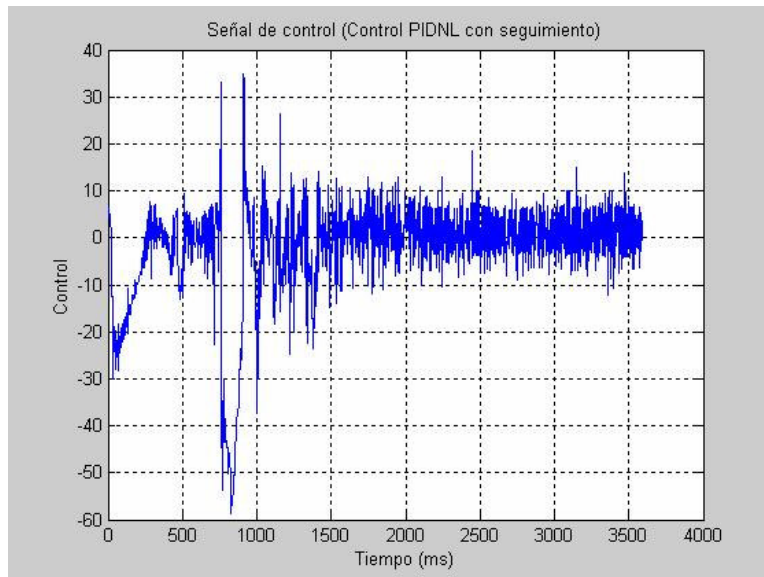


Figura 6.43: Gráfica de la señal de control; control PID NL con seguimiento.

En la figura 6.43 observamos la señal de control aplicada al motor con un control PID NL con seguimiento, se puede apreciar como actúa el control ya que el valor de $u(t)$ es menor que con los otros controles. La señal nos muestra un variación al momento de realizar el seguimiento y al llegar al valor de referencia se aproxima a cero. La gran variación en la señal es propiciada por el ruido del generador taquimétrico.

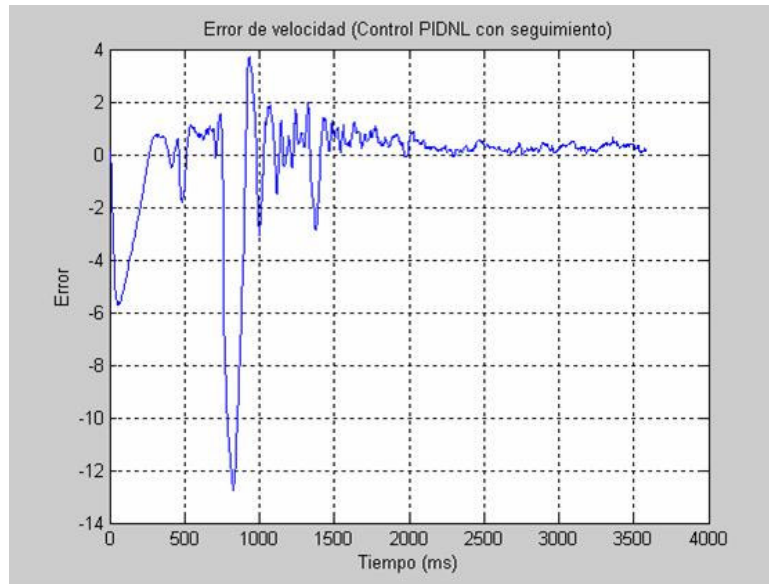


Figura 6.44: Gráfica del error de velocidad; control PID NL con seguimiento.

La señal de error permite apreciar la convergencia asintótica del controlador, es posible apreciar la contribución del controlador. Como vemos los rangos de variación son menores que en cualquier otro control.

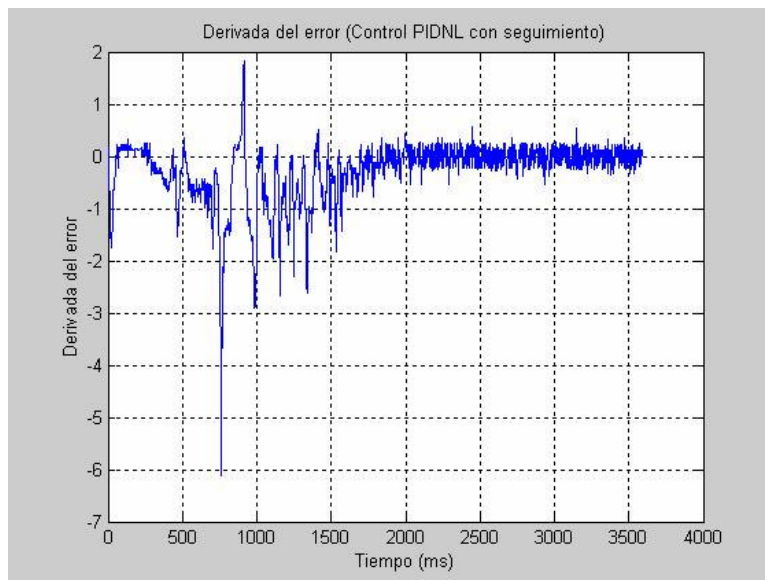


Figura 6.45: Gráfica de la derivada del error; control PID NL con seguimiento.

La derivada del error, implica la estimación de la derivada del estado, es decir una señal estimada a partir de una señal con ruido propiciado en el generador taquimétrico.

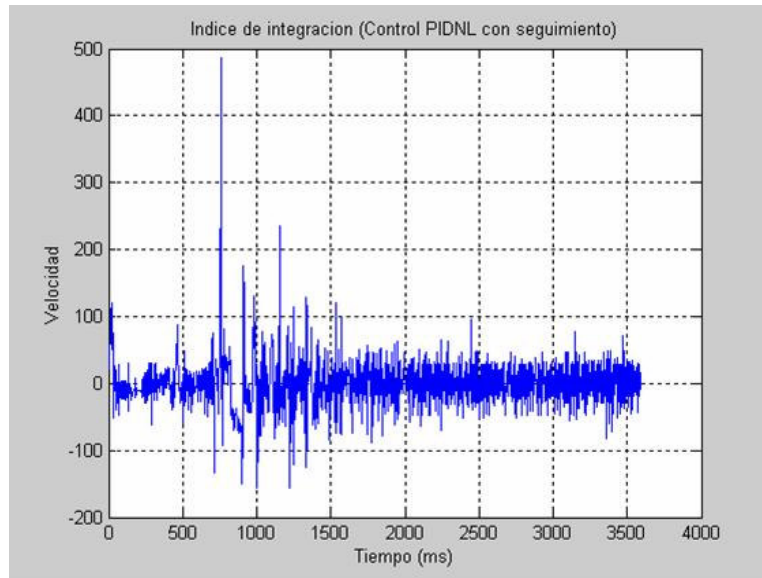


Figura 6.46: Gráfica del índice de integración; control PID NL con seguimiento.

La figura nos muestra el índice de integración de la señal de control, se puede observar la variación que tiene la señal entre los 800 ms y los 1500 ms y la estabilidad en valores cercanos a cero al alcanzar el valor de referencia.

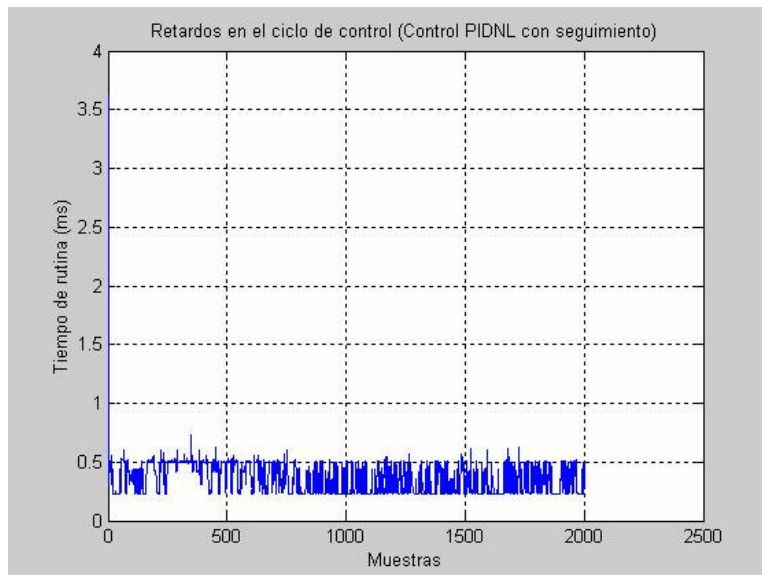


Figura 6.47: Gráfica de los retardos del sistema.

Los retardos presentados en esta gráfica son en promedio de 0.5 milisegundos, situación que beneficia en la tarea de control, propiciada a 1 milisegundo de tiempo de muestreo, es decir el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea aun en la plataforma de Windows.

6.14. Control PX con regulación.

Los parámetros utilizados en el control PX, resultan de la experimentación y en base a criterios preestablecidos, estos son:

$$W_n = 0.12$$

$$L = 0.087$$

$$V_d = 135 \text{ rpm.}$$

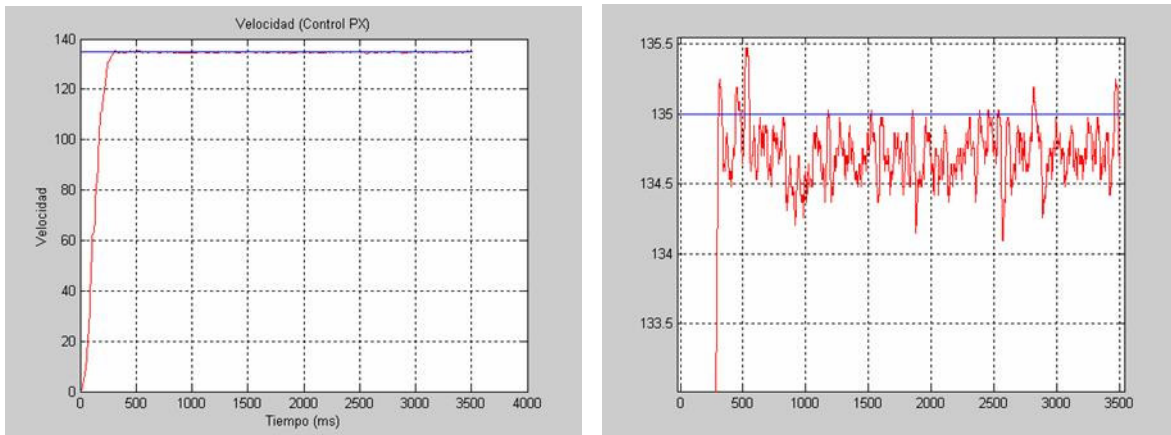


Figura 6.48: Gráfica de velocidad, ampliación a 135 rpm, de control PX.

La figura 6.48 nos muestra la gráfica del control PX de velocidad, como podemos observar el sobre impulso es menor o casi nulo y la señal tiene mejor estabilidad como podemos ver a la derecha en la ampliación de la gráfica. Existen variaciones (134.5 rpm 135 rpm) en la señal de control por la provocadas por el ruido del generador taquimétrico.

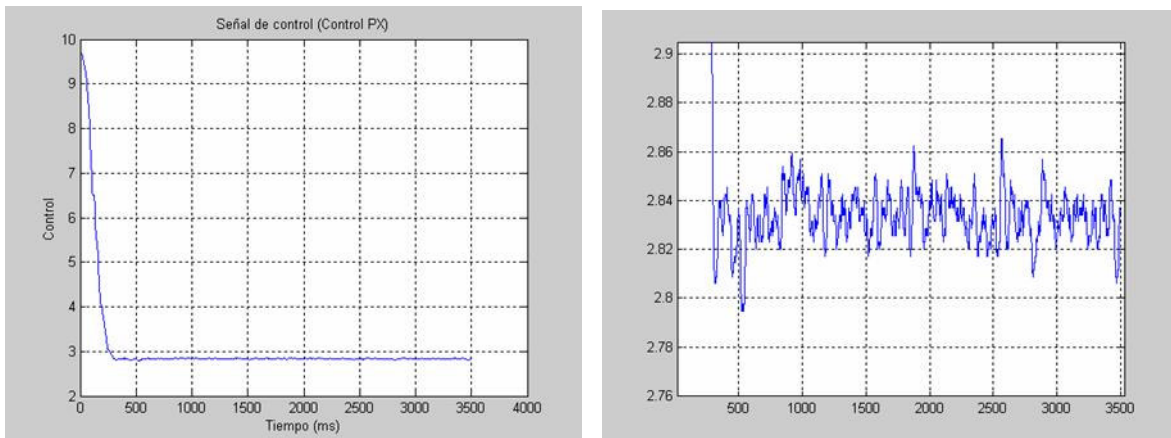


Figura 6.49: Gráfica del control PX de la señal de control, ampliación de la gráfica.

En la figura 6.49 a la izquierda observamos la señal de control aplicada al motor con un control PX, se puede apreciar como el valor de $u(t)$ es menor que con los otros controles, y

se estabiliza en un valor relativamente bajo, como podemos ver a la derecha en la ampliación de la misma gráfica.

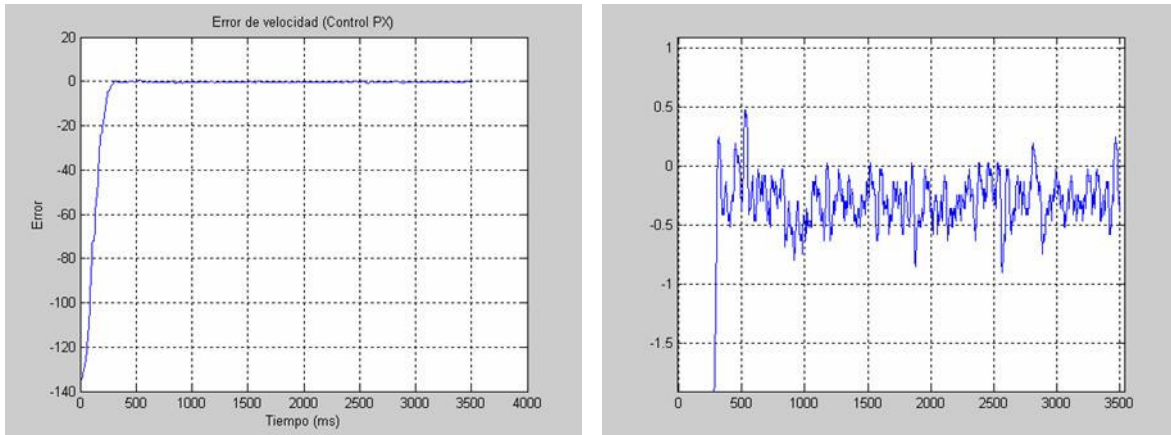


Figura 6.50: Gráfica del error del control PX, ampliación de la gráfica.

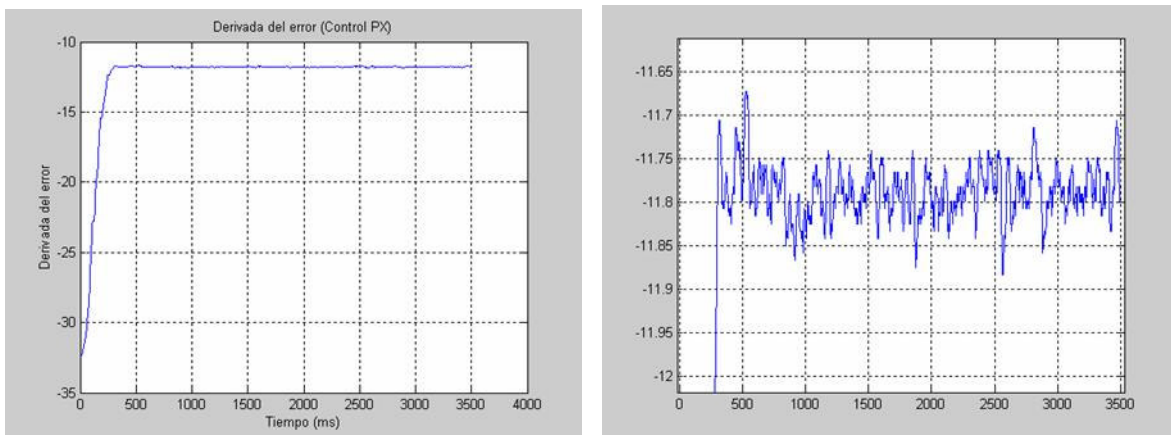


Figura 6.51: Gráfica de la derivada del error del control PX, ampliación de la gráfica.

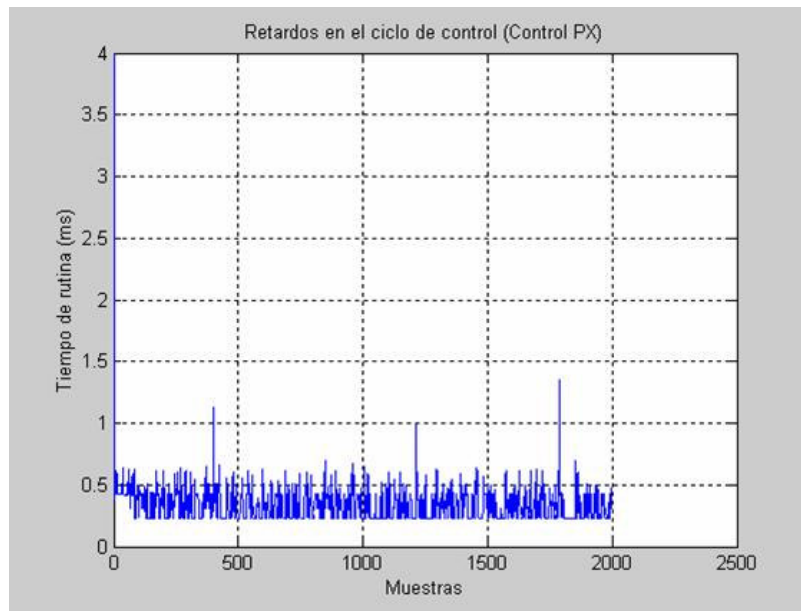


Figura 6.52: Gráfica de los retardos del sistema.

Los retardos presentados en esta gráfica son en promedio de 0.6 milisegundos, situación que beneficia en la tarea de control, propiciada a 1 milisegundo de tiempo de muestreo, es decir el ancho de banda del sistema empleado es suficiente para la ejecución de esta tarea aun en la plataforma de Windows.

6.15 Control PX basada en seguimiento.

Los parámetros utilizados en el control PX con Seguimiento resultan de la experimentación y en base a criterios preestablecidos, estos son:

$$W_n = 0.23$$

$$V_d = 135 \text{ rpm.}$$

$$T_b = 2 \text{ Seg.}$$

$$L = 0.0446$$

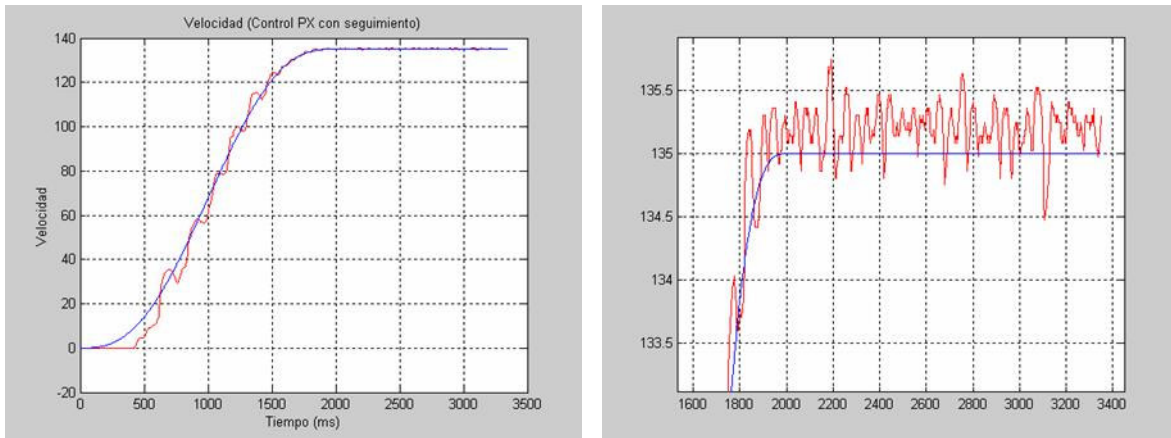


Figura 6.53: Gráfica de velocidad, ampliación a 135 rpm, de control PX con seguimiento.

En la figura 6.53 a la izquierda observamos la Velocidad con seguimiento del motor con un control PX, a la derecha en la ampliación de la misma gráfica observamos que el seguimiento no es bueno, pero es muy bueno al alcanzar al velocidad.

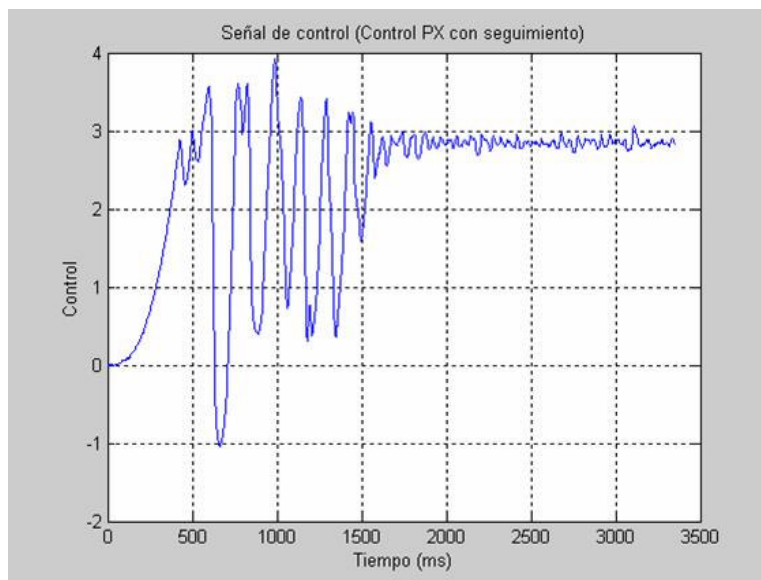


Figura 6.54: Gráfica de la señal de control del control PX con seguimiento.

En la figura 6.54 observamos la Señal de control aplicada al motor con un control PX con seguimiento, se puede apreciar como actúa el control con un valor de $u(t)$ mas bajo.

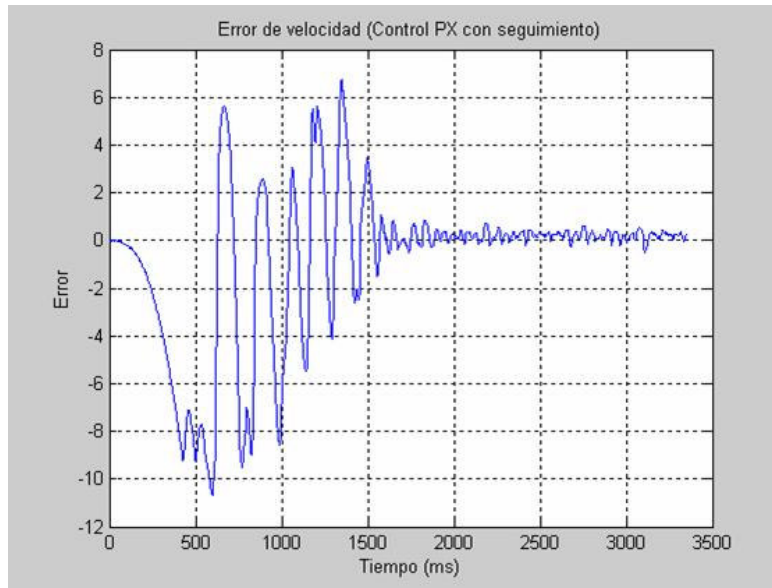


Figura 6.55: Gráfica del error de velocidad del control PX con seguimiento.

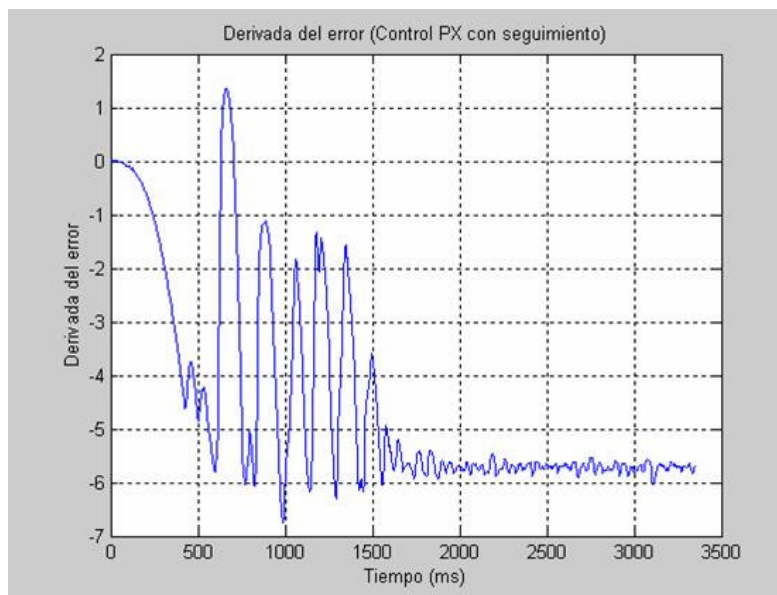


Figura 6.56: Gráfica de la derivada del error del control PX con seguimiento.

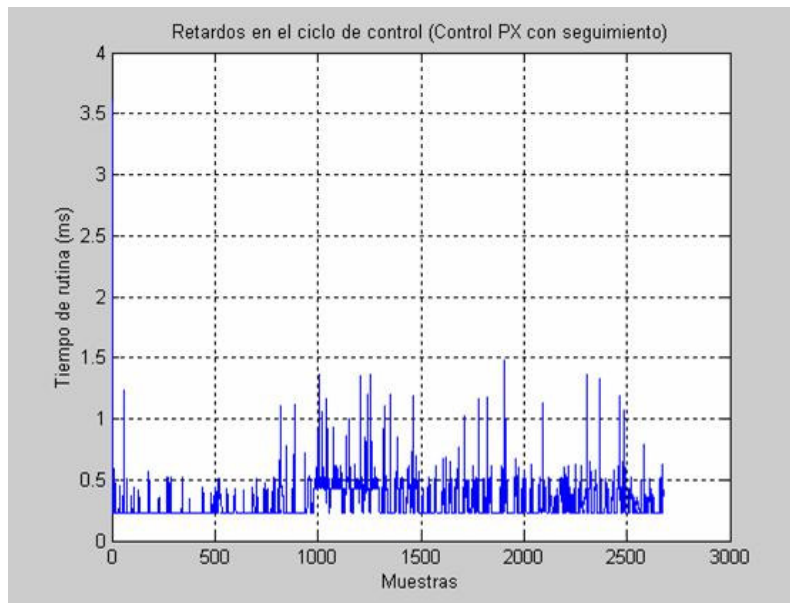


Figura 6.57: Gráfica de los retardos del sistema.

Los retardos presentados en esta gráfica son mayores al promedio, situación que no beneficia en la tarea de control, propiciada a 1 milisegundo de tiempo de muestreo.

6.16 Graficas de los controles aplicando perturbación.

A continuación se muestran las graficas de velocidad, señal de control, error de velocidad, derivada del error, índice de integración y los retardos del sistema de los controles PD, PI, PID, PID NL, y PX en condiciones de perturbación, es decir sujetando la flecha del motor (rotor bloqueado).

La frecuencia alta de las señales es por el ruido propiciado por el generador taquimétrico, quien proporciona señales indeseables.

6.17 Control PD con perturbación.

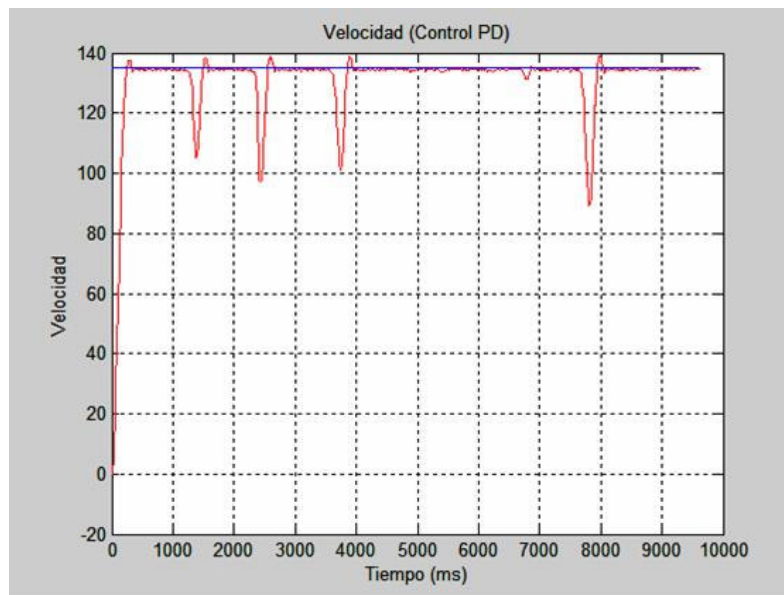


Figura 6.58: Gráfica de la velocidad, control PD con perturbaciones.

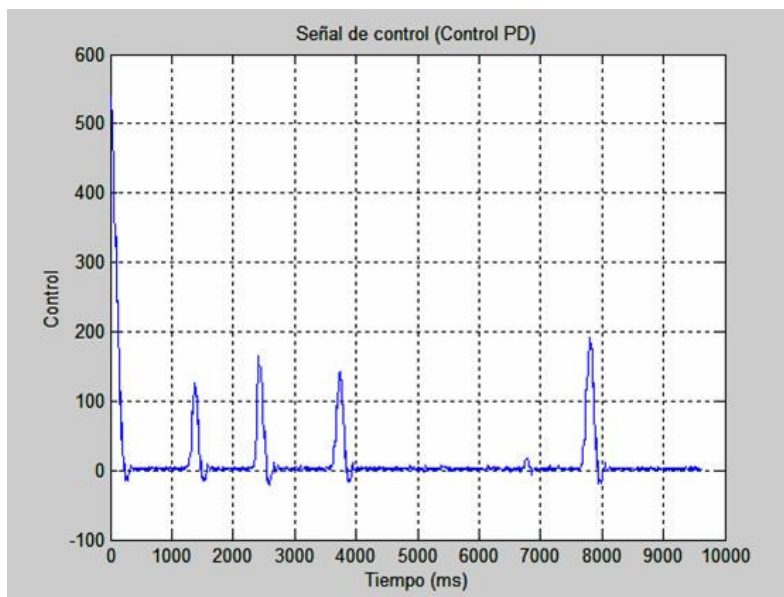


Figura 6.59: Gráfica de la señal de control, control PD con perturbaciones.

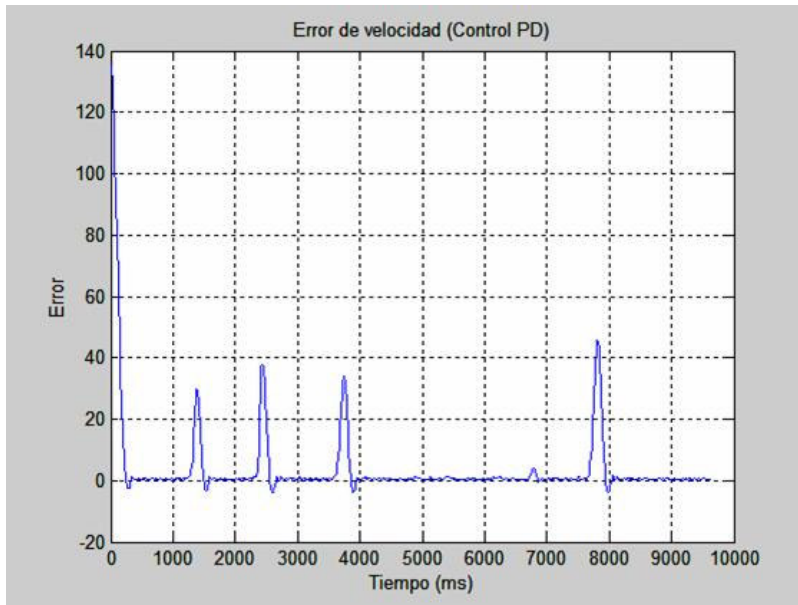


Figura 6.60: Gráfica del error, control PD con perturbaciones.

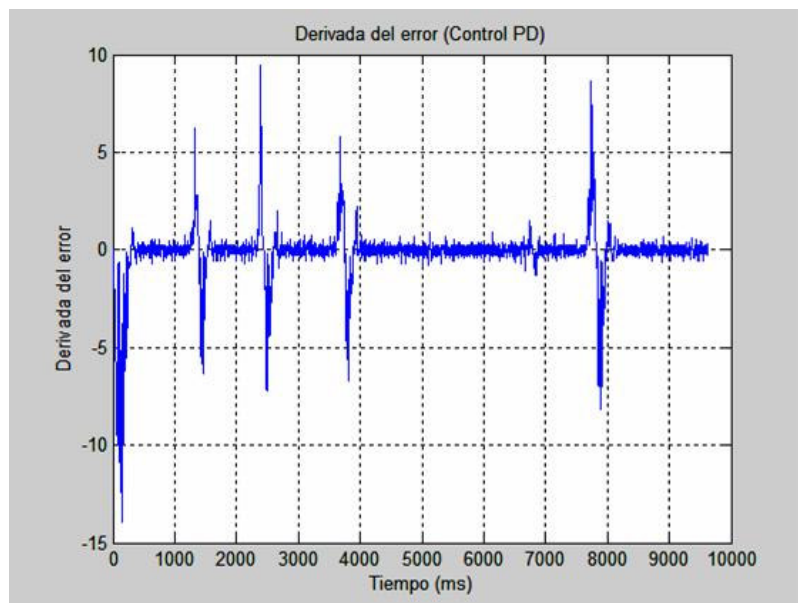


Figura 6.61: Gráfica de la derivada del error, control PD con perturbaciones.

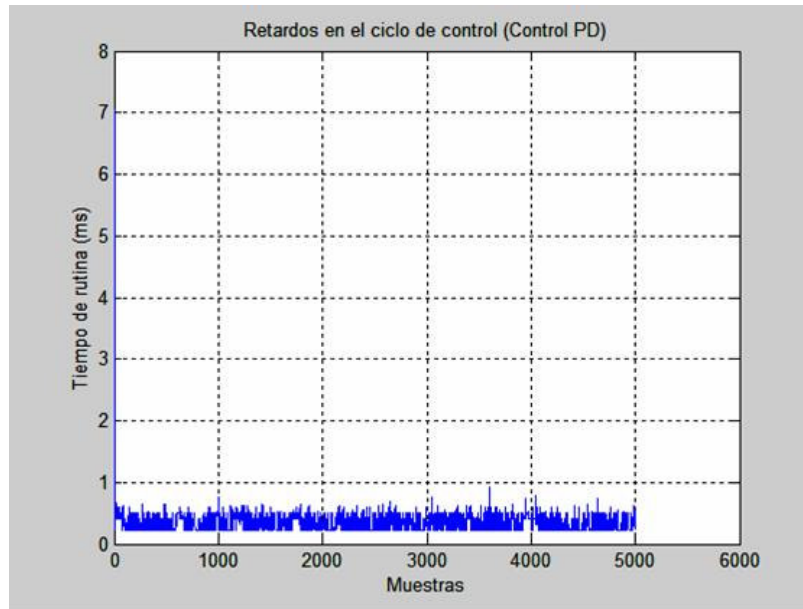


Figura 6.62: Gráfica de los retardos del sistema, control PD con perturbaciones.

6.18 Control PI con perturbación.

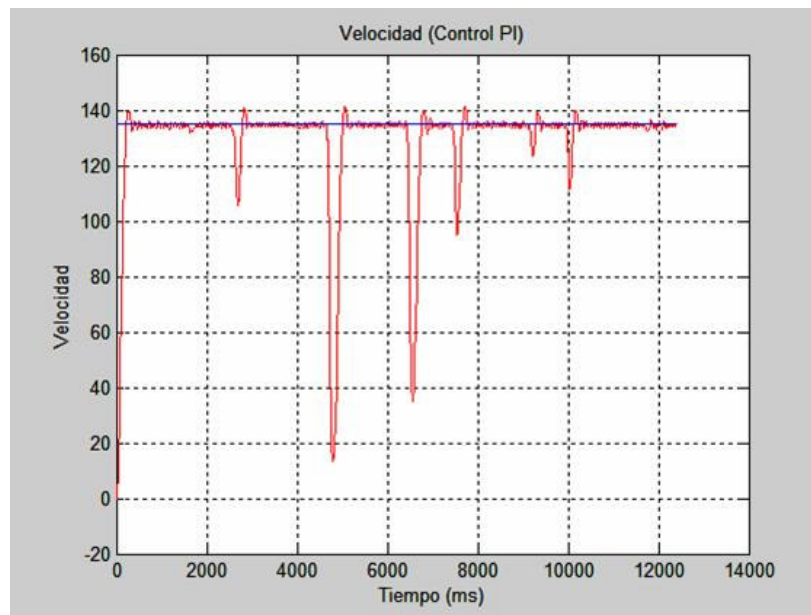


Figura 6.63: Gráfica de la velocidad, control PI con perturbaciones.

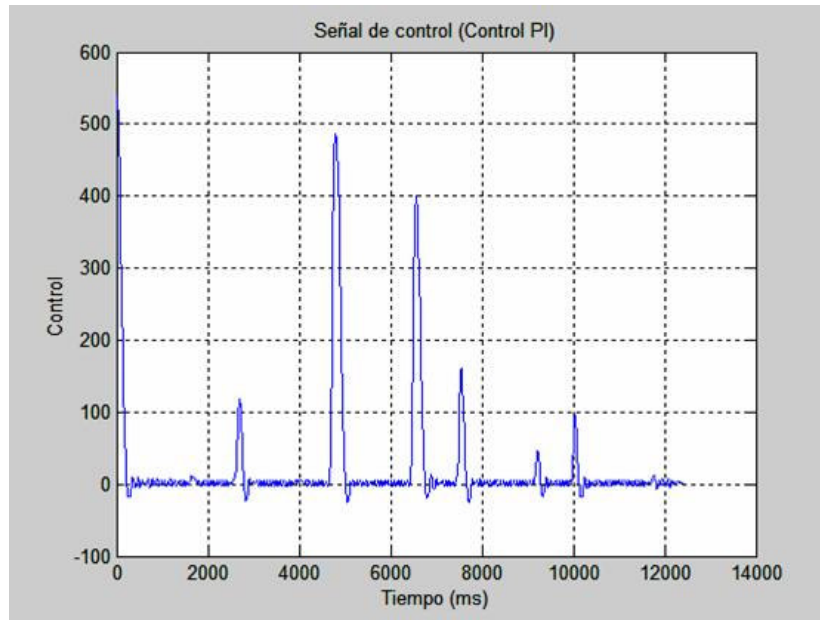


Figura 6.64: Gráfica de la señal de control, control PI con perturbaciones.

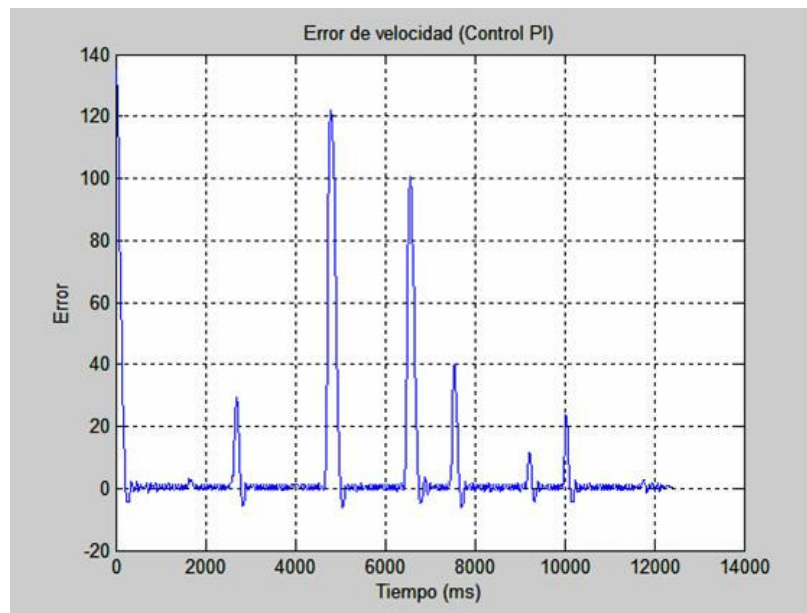


Figura 6.65: Gráfica del error, control PI con perturbaciones.

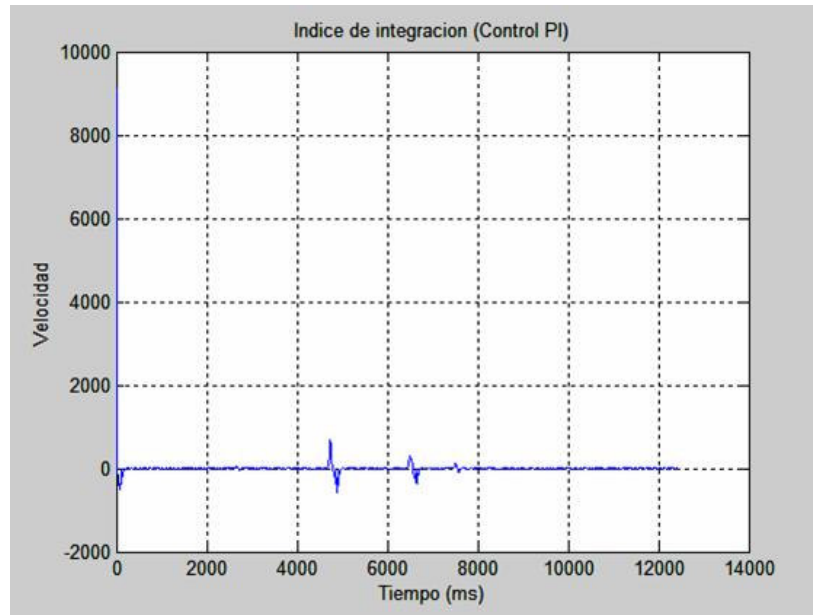


Figura 6.66: Gráfica del índice de integración, control PI con perturbaciones

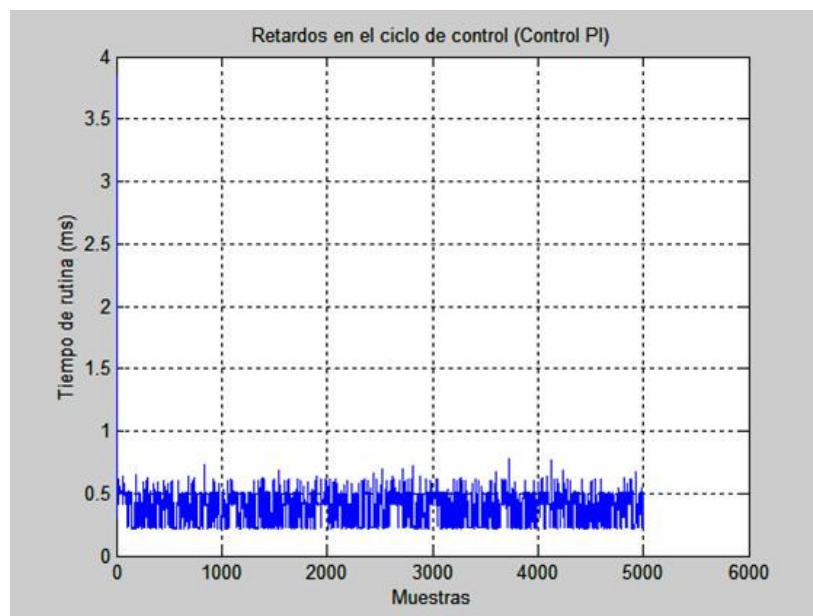


Figura 6.67: Gráfica de los retardos del sistema, control PI con perturbaciones.

6.19 Control PID con perturbación.

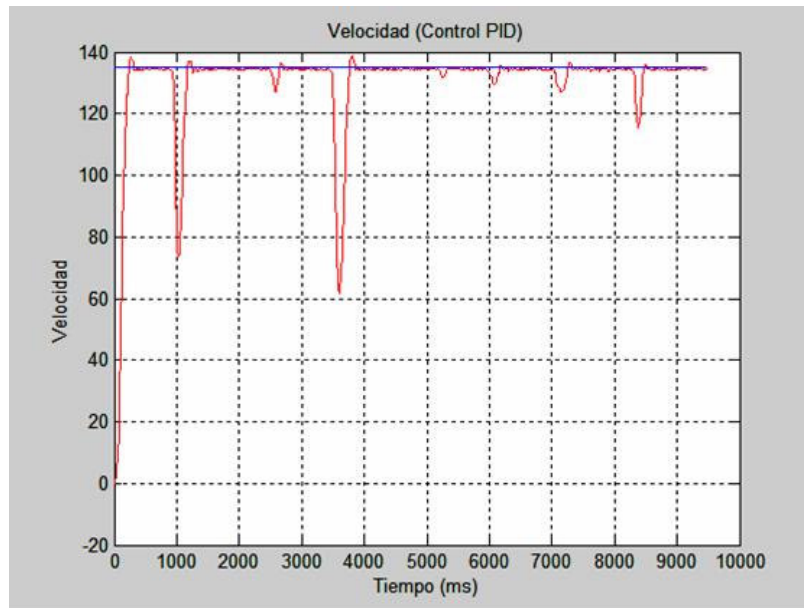


Figura 6.68: Gráfica de la velocidad, control PID con perturbaciones.

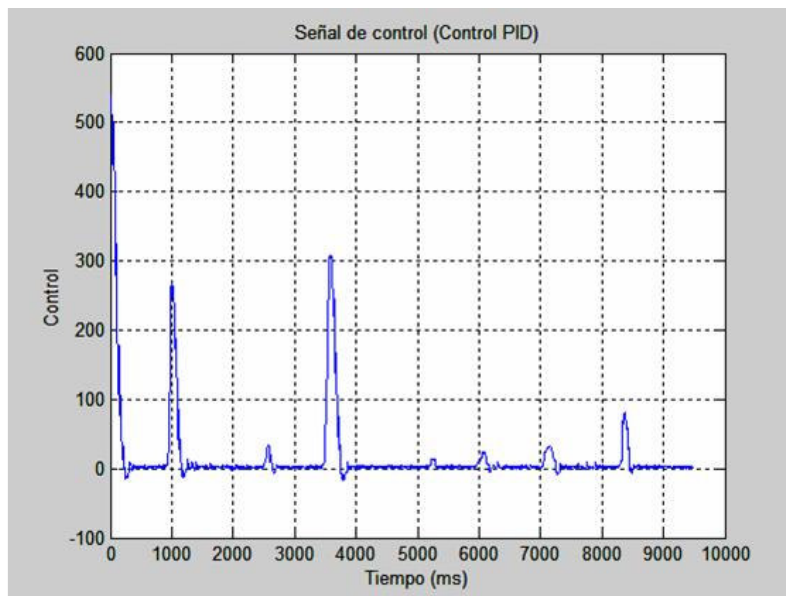


Figura 6.69: Gráfica de la señal de control, control PID con perturbaciones.

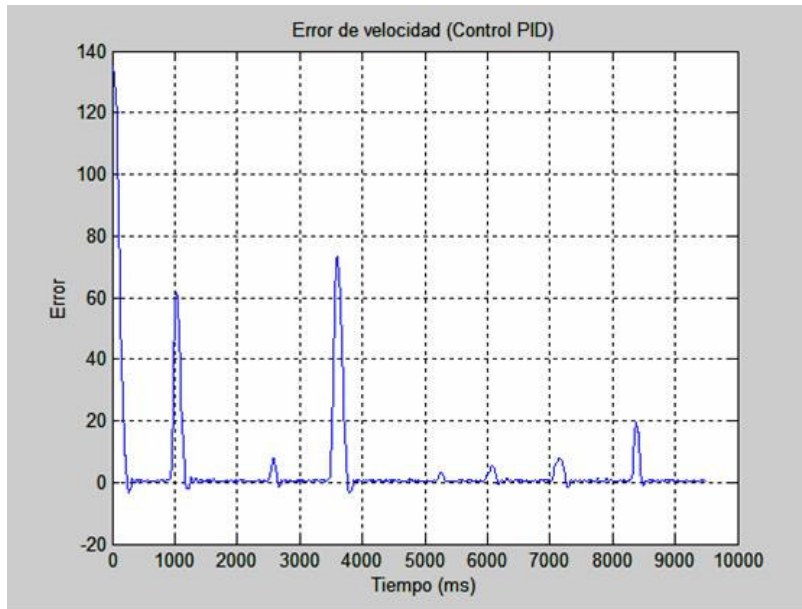


Figura 6.70: Gráfica del error, control PID con perturbaciones

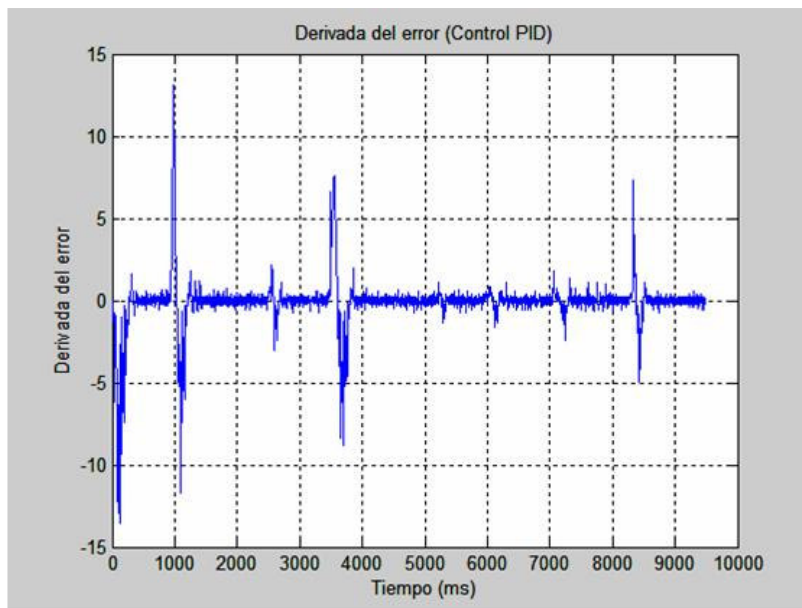


Figura 6.71: Gráfica de la derivada del error, control PID con perturbaciones.

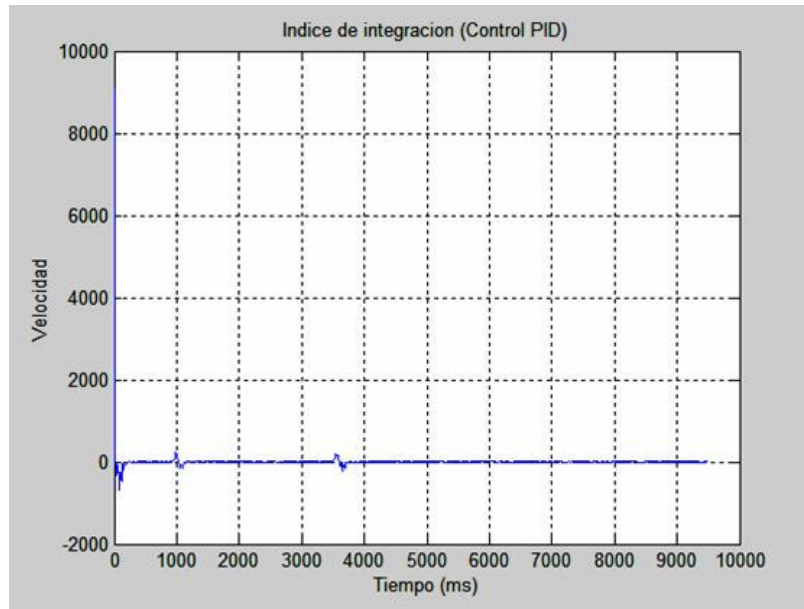


Figura 6.72: Gráfica del índice de integración, control PID con perturbaciones

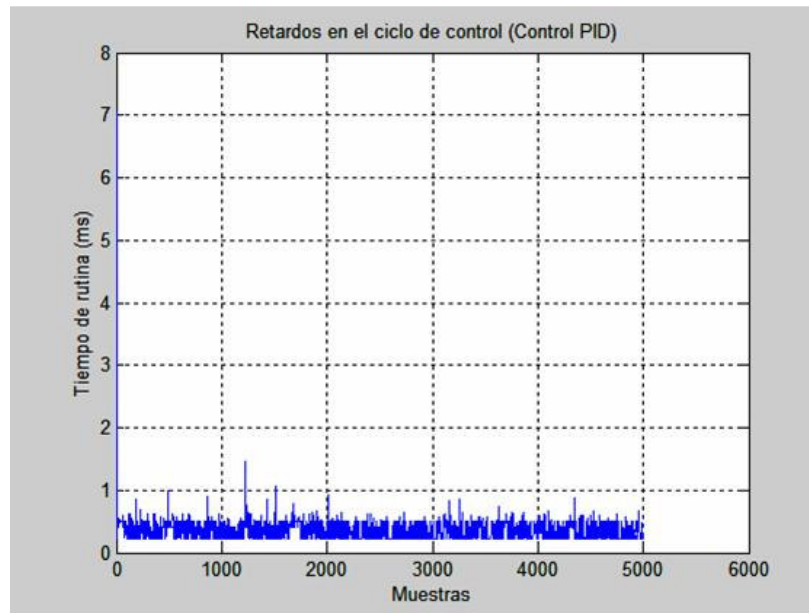


Figura 6.73: Gráfica de los retardos del sistema, control PID con perturbaciones

6.20 Control PID NL con perturbación.

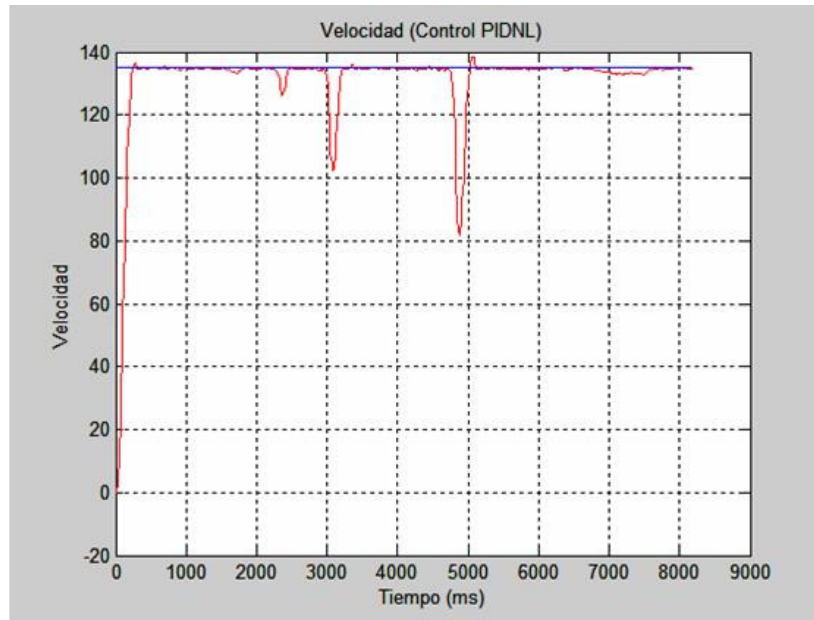


Figura 6.74: Gráfica de la velocidad, control PID NL con perturbaciones.

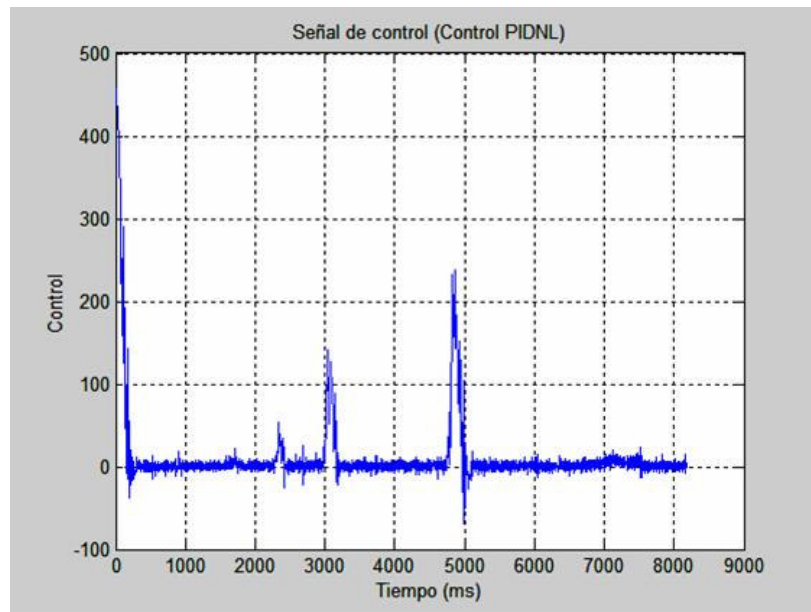


Figura 6.75: Gráfica de la señal de control, control PID NL con perturbaciones.

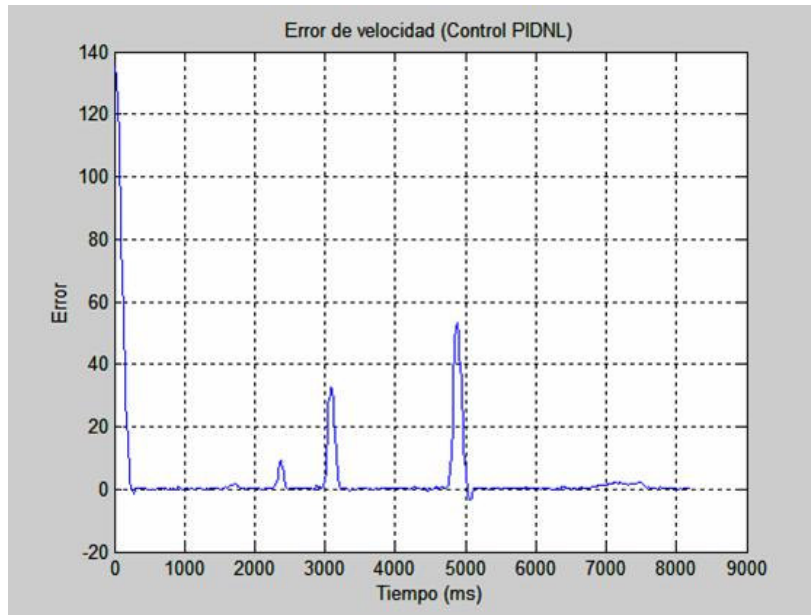


Figura 6.76: Gráfica del error, control PID NL con perturbaciones.

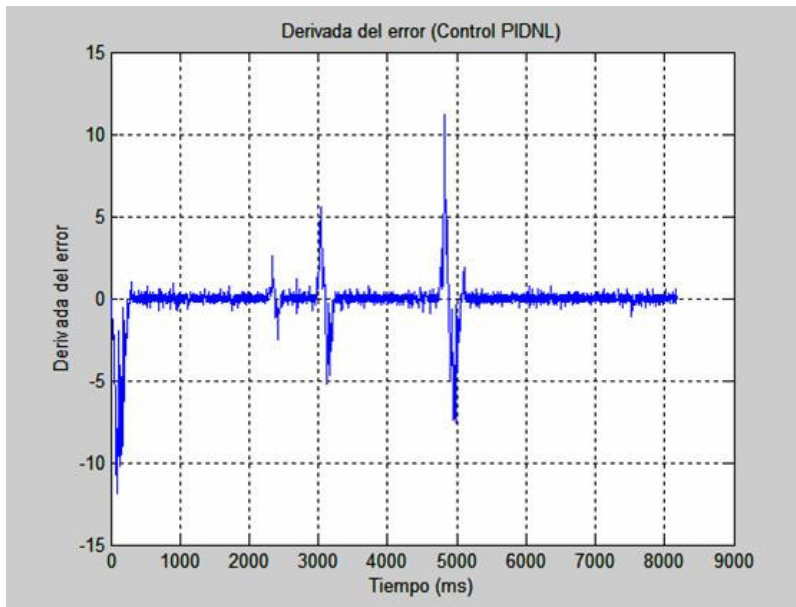


Figura 6.77: Gráfica de la derivada del error, control PID NL con perturbaciones.

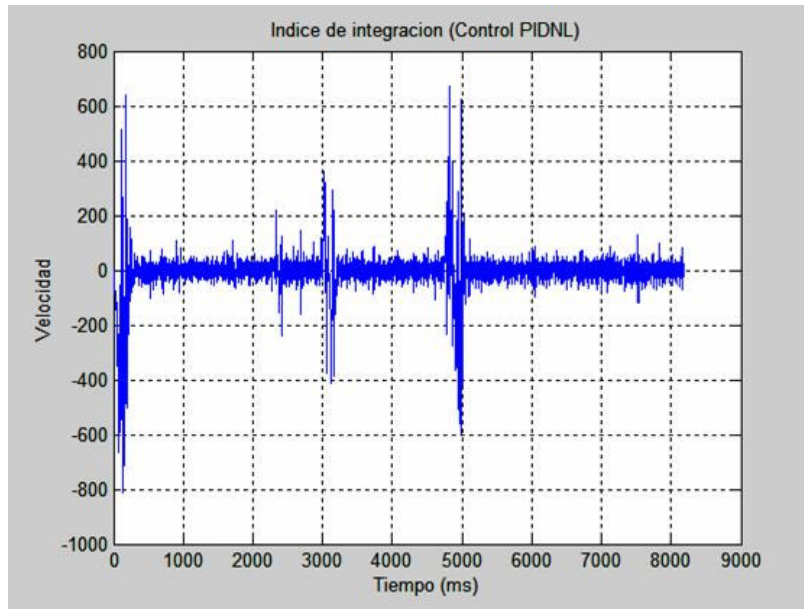


Figura 6.78: Gráfica del índice de integración, control PID NL con perturbaciones.

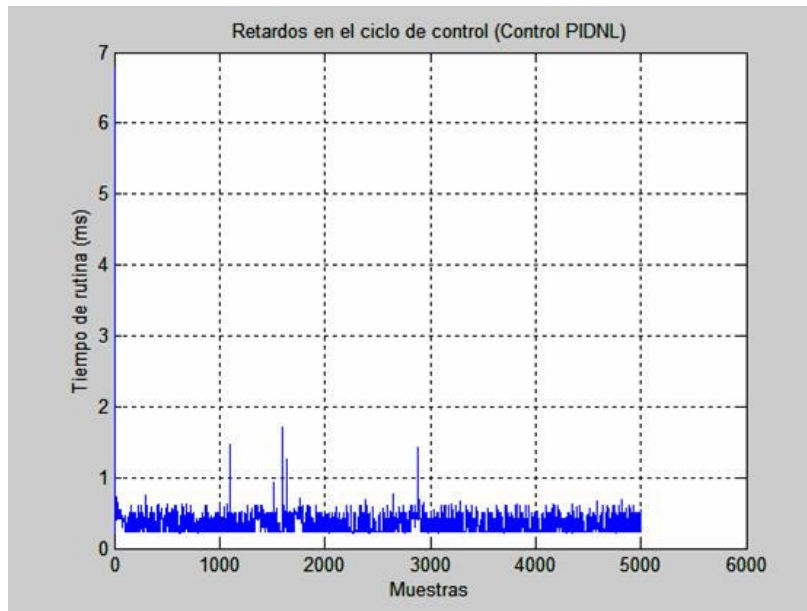


Figura 6.79: Gráfica de los retardos del sistema, control PID NL con perturbaciones.

6.21 Control PX con perturbación.

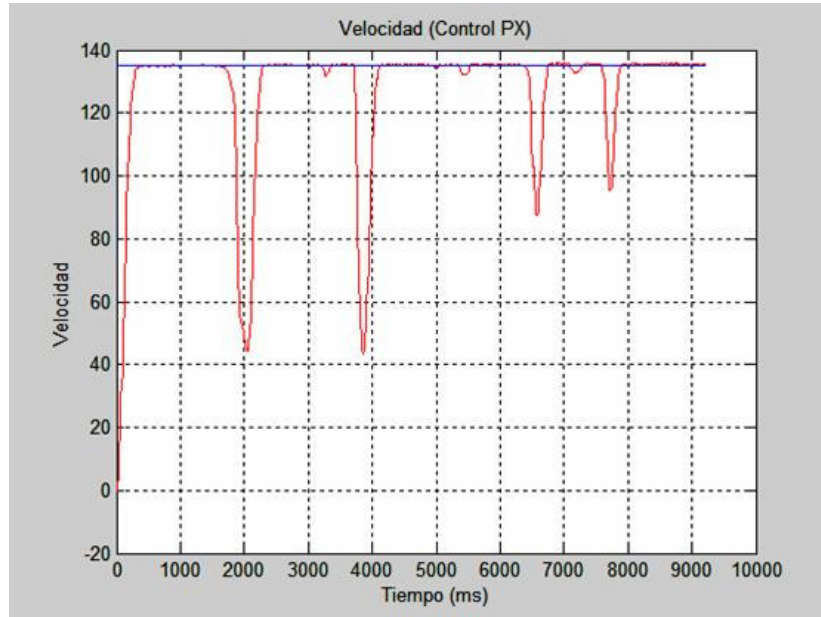


Figura 6.80: Gráfica de la velocidad, control PX con perturbaciones.

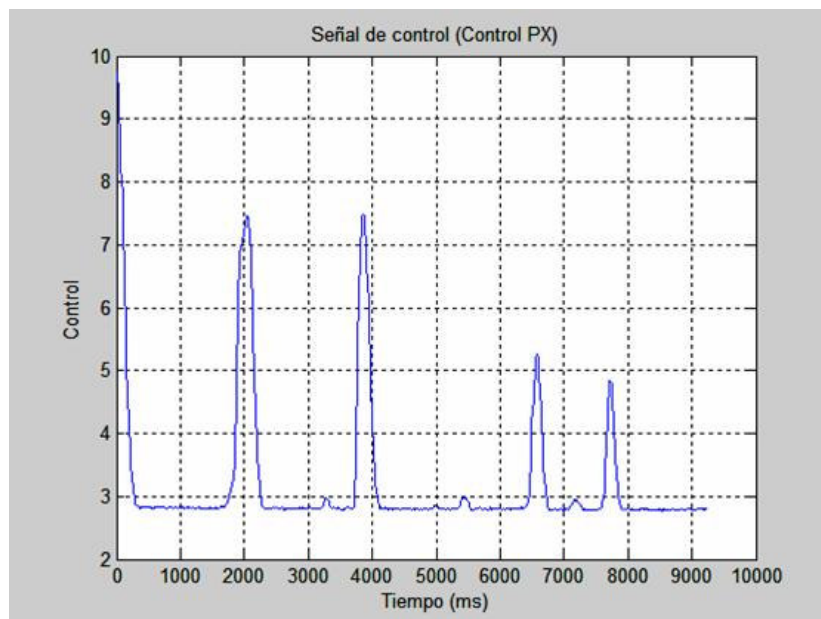


Figura 6.81: Gráfica de la señal de control, control PX con perturbaciones.

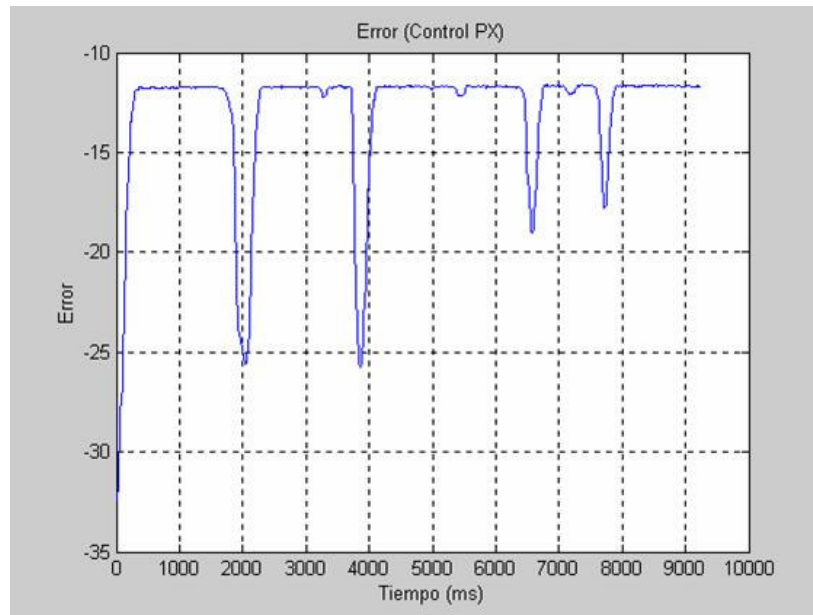


Figura 6.82: Gráfica del error, control PX con perturbaciones

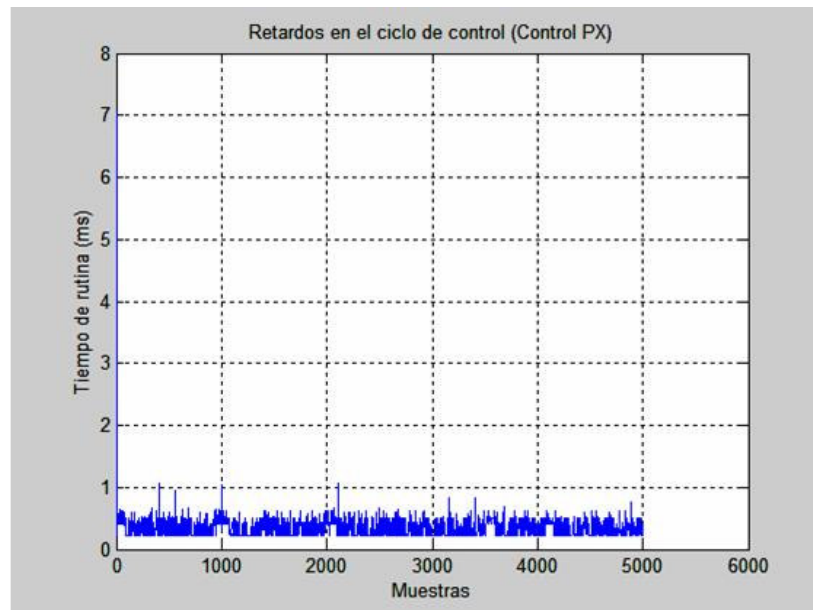
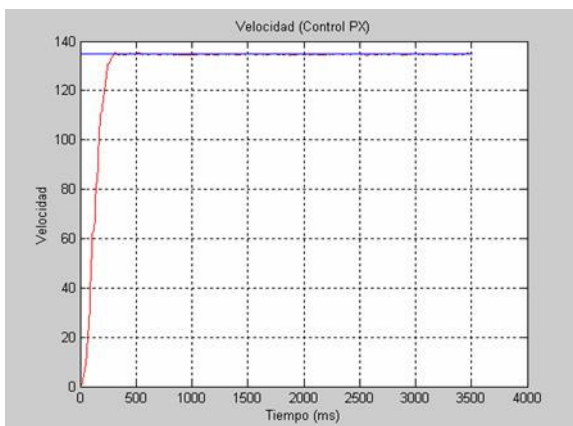
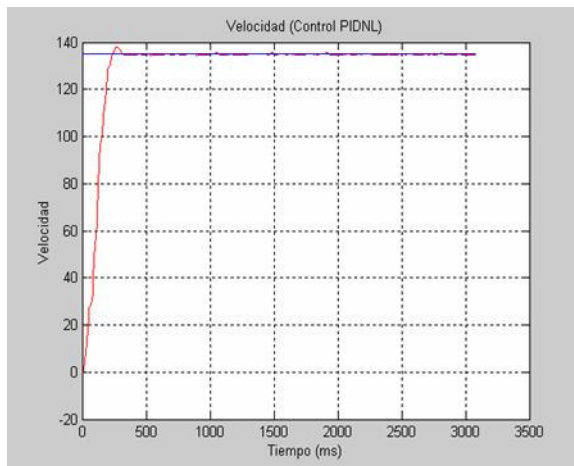
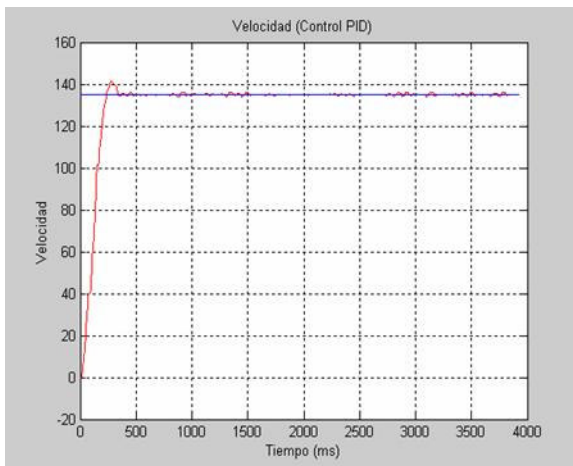
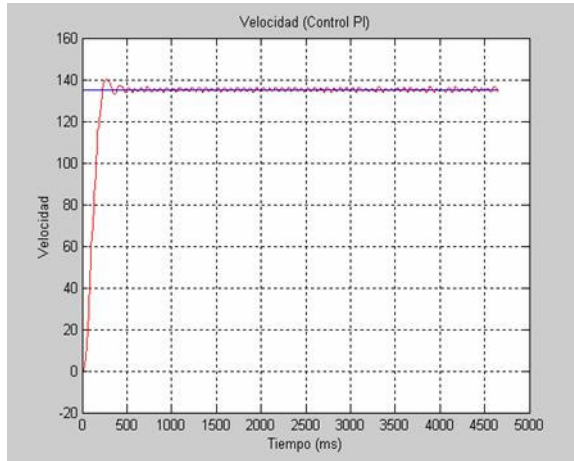
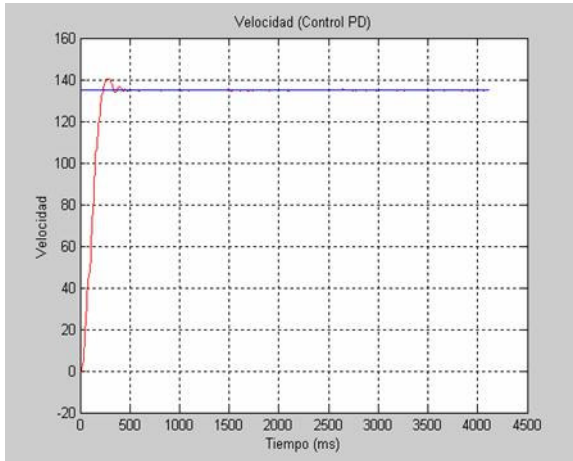


Figura 6.83: Gráfica de los retardos del sistema, control PX con perturbaciones.

Como nos muestran las graficas de velocidad, el resultado optimo lo obtenemos del el control PX. Ya que en todas y cada una de las perturbaciones el valor de la señal cae y se estabiliza en el valor de referencia. Para el caso de la señal de control nos damos cuenta que con el control PX el valor de $u(t)$ es el mas bajo al momento de ajustar cada una de las perturbaciones. Para el caso de la señal de error el control PX nos muestra un intervalo de -30 a -10 en el valor de $e(t)$.

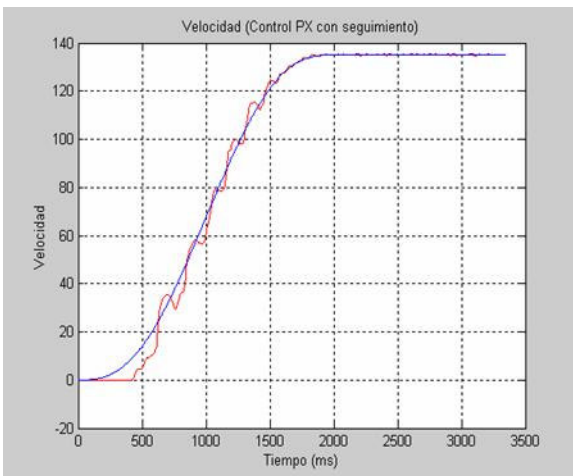
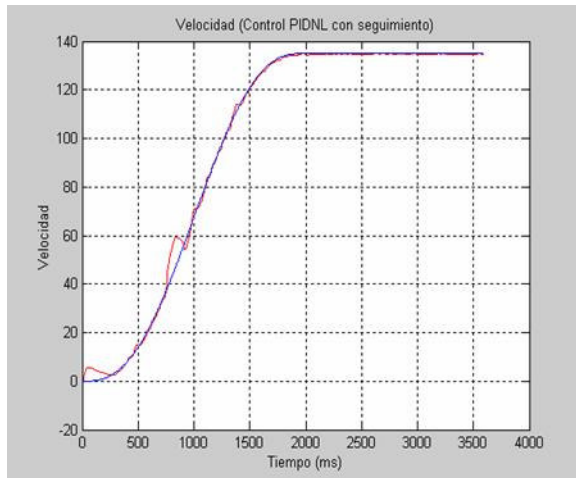
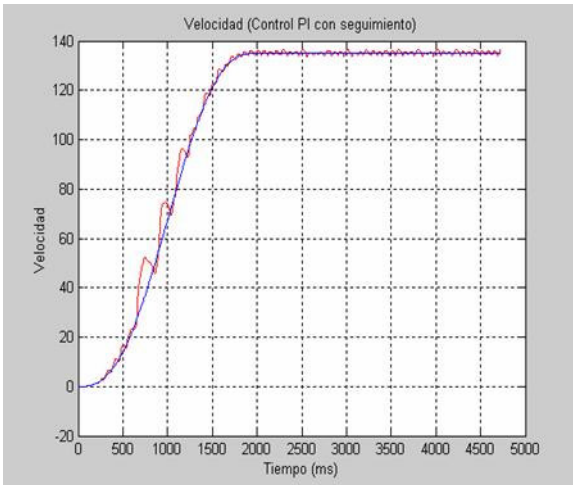
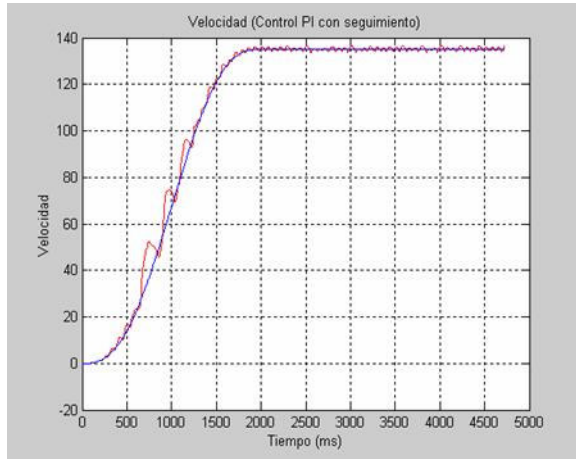
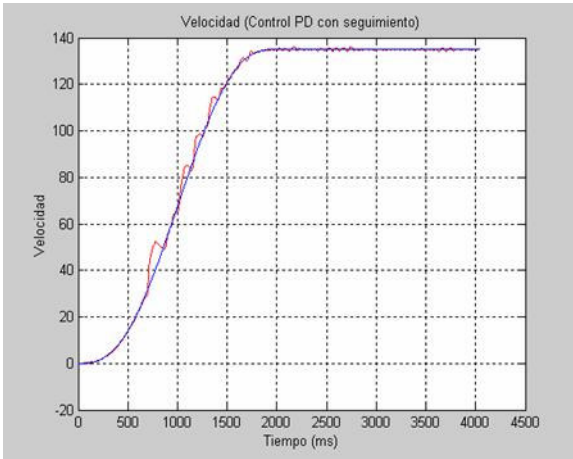
6.22 Comparación final de la gráficas de velocidad.

En esta comparación se muestran las gráficas obtenidas con los controles PD, PI, PID, PID NL, PX, a la misma escala y a 135rpm.



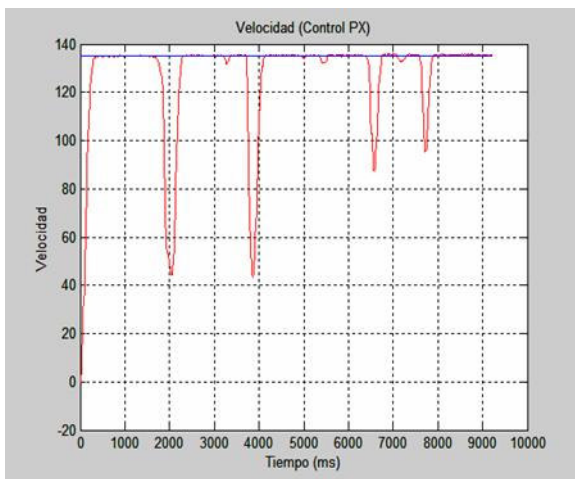
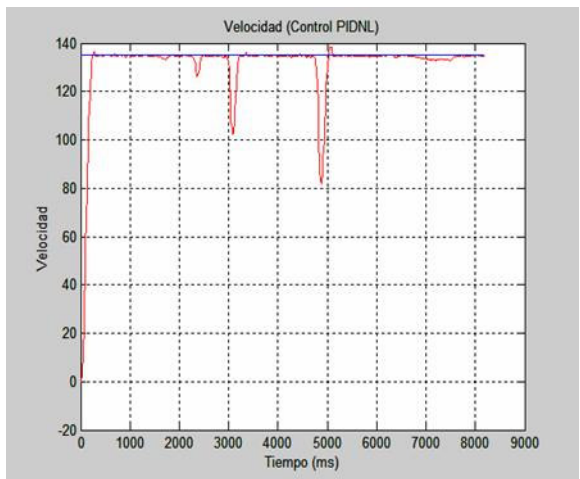
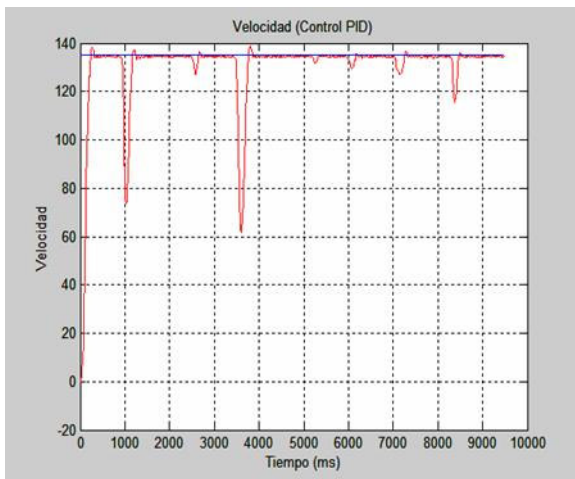
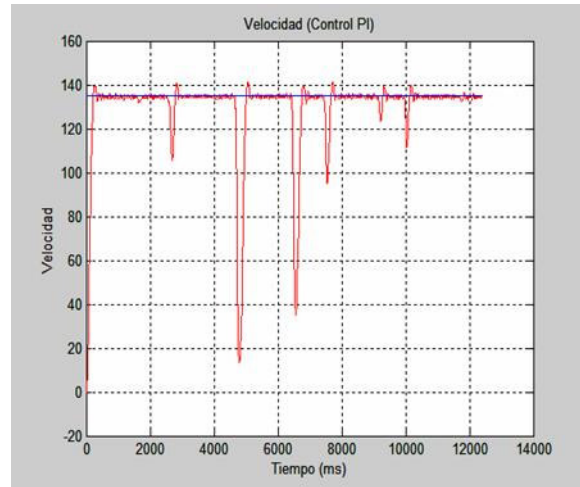
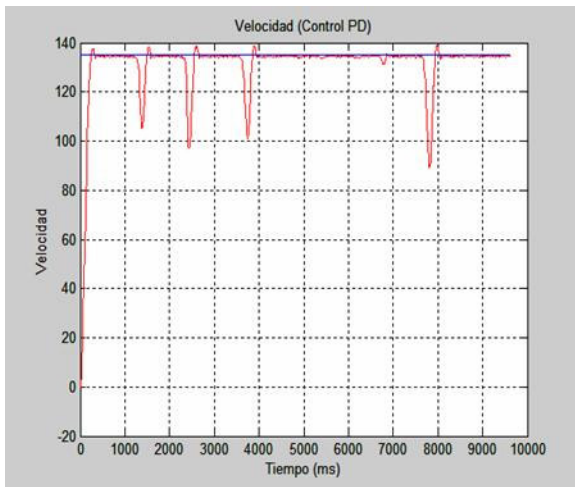
El control PX proporciona la mejor respuesta al ajustar la señal al valor deseado.

En esta comparación se muestran las gráficas obtenidas con los controles PD, PI, PID, PID NL, PX, con seguimiento a la misma escala y a 135rpm.



El control PID NL proporciona la mejor respuesta al realizar el seguimiento.

En esta comparación se muestran las gráficas obtenidas con los controles PD, PI, PID, PID NL, PX, con perturbación a la misma escala y a 135rpm.



El control PX proporciona la mejor respuesta al controlar las perturbaciones.

Conclusiones

Para cada control se realizaron mediciones de velocidad, señal de control, error y retardos en el sistema, en los controles PI, PID, PID NL se muestra también la gráfica de índice de integración y en PD, PID, PID NL la de derivada del error.

De acuerdo a los experimentos realizados y como se puede observar en las gráficas los distintos controles analizados se puede demostrar que para nuestro caso de estudio “regulación global suave de velocidad”, el control PX se muestra con un óptimo desempeño tanto en los experimento de control con regulación como los obtenidos en control con perturbación, ya que lo que se busca es el control fino de la velocidad, de tal forma que no exista el sobre impulso que se muestra con los otros controles.

CAPÍTULO 7

PERSPECTIVAS Y CONCLUSIONES

7.1 Perspectivas.

El control PX permitirá en un futuro realizar estudios comparativos de otros controladores con el afán de determinar las virtudes de cada uno de ellos en diferentes circunstancias y aplicaciones.

Para un trabajo futuro se sugiere utilizar microcontroladores, esto simplifica el acondicionamiento de señales, son muy accesibles en la programación y la implementación.

Así como también la implementación de filtros para eliminar los distintos ruidos ocasionados por el generador taquimétrico y el sistema de interfaz.

7.2 Conclusiones.

En el desarrollo de este trabajo se realizaron las siguientes etapas:

- Diseño y construcción de la interfaz electrónica para realizar los experimentos de control que consta de un circuito a base de amplificadores operacionales para generar una señal modulada en el ancho del pulso a partir de la comparación de la señal de control y una señal triangular de amplitud y frecuencia constante, dispositivos optoaisladores y transistores para amplificar la señal de control y controlar al motor.
- Estudio de la caracterización del motor.
- Estudio de la caracterización del sistema de retroalimentación a base del generador taquimétrico acoplado a la flecha del motor.
- El desarrollo de programas de simulación en MatLab 6 para obtener referencias y ajustar las variables requeridas.
- La sintonización de los controladores PD, PI, PID, PID NO LINEAL y PX con y sin compensación en regulación, seguimiento y perturbaciones.
- El desarrollo de un software en lenguaje Delphi 5 para el experimento de cada control, en el que se lee un arreglo multidimensional de las variables, archivado durante la ejecución de la acción de control en un archivo de Matlab 6 para mostrar el comportamiento del motor en tiempo real.

De los resultados experimentales obtenidos podemos concluir que:

- El controlador mas eficiente es el control PX ya que el error en estado estable es considerablemente menor que el error obtenido con la aplicación de la acción de control PD, PI, PID y PID NL.
- La sintonización de los controles en forma analítica es importante pero no es la sintonización definitiva ya que esta se define mas adecuadamente en forma experimental, sin embargo la obtención de las ganancias en forma analítica permite tener un criterio importante para la designación definitiva.
- Se logró la implementación del control con resultados óptimos ofreciendo una nueva opción para los sistemas de control.

Referencias

- [1] Domínguez Ramírez, O. A. and V. Parra – Vega, "Constrained Lagrangian-Based Force –Position Control for Haptic Guidance", Proceedings of Eurohaptics 2003, Incorporating the PHANTOM Users Research Symposium. Trinity College Dublin and Media Lab Europe, Dublin, Ireland, UK, pp. 444–450, 2003.
- [2] Ogata. K. "Ingeniería de Control Automático", Séptima edición. Ed. Prentice Hall Hispanoamericana, S.A. de C. V. 1996.
- [3] Malvino. "Principios de electrónica", Ed. Mc Graw Hill.
- [4] Maloney T. J. "Electrónica Industrial", Ed. Prentice Hall.
- [5] Kuo B. C. "Sistemas de Control Automático", Séptima edición. Ed Prentice Hall Hispanoamericana, S.A. de C. V. 1996..
- [6] Smith, C. A. Corripio (1996). "Control Automático de Procesos. Teoría y Práctica", Ed. Limusa Noriega Editores.
- [7] Alerich W. N., "Control de motores eléctricos teoría y aplicaciones." Ed. Diana, México, 1972.
- [8] Franklin, Gene. Powell, David. Emami-Naeine, Abbas (1991). "Control de Sistemas Dinámicos con Retroalimentación" Ed. Addison-Wesley Iberoamericana.
- [9] Thaler G. J. Wilcox M. L. " Máquinas eléctricas", Ed. Limusa, México, 1984.
- [10] Burden Richard L.. D. "Análisis Numérico", Ed. Thomson Learning; Septima edición. México, (2002); ISBN 970-686-134-3.
- [11] Pérez García M. A. Labres Antón J. C., Campo Rodríguez J. C., Ferrero Martín F., Grillo Ortega G. J., "Instrumentación Electrónica", Ed. Thomson, Madrid (2004).
- [12] Juncar J. A., Cortés J. L., Pérez Quiroz D., "Delphi 5", Ed. Marcombo S.A. 2000 AlfaOmega.
- [13] Martínez Rodríguez A. "Diseño y Construcción del Subsistema Electrónico Empleado en Robots Manipuladores" Mayo de 2002.
- [14] Domínguez Ramírez O. A., "Diseño, construcción y evaluación de algoritmos de control en un robot planar de 2grados de libertad", Agosto de 1999.
- [15] Díaz Montiel M. G. Galván Zavala V.H. Aranda López N.J. "Caracterización de un sensor".
- [16] Bartho F. T., Windibank, Martínez García Salvador, "Motores eléctricos industriales y dispositivos de control", Ed Urmo, España 1968.
- [17] Adkins B. Leon A. L. "Teoría general de las maquinas eléctricas" Ed. Urmo. España, 1967.
- [18] Philips Consumer Electronic Company, ECG Products, Master Replacement Guide.[19] <http://www.solorb.com/elect/solarcirc/pwm1/index.html>[20] <http://www.bombas y sus aplicaciones.htm>[21] http://cgi.ebay.com/PMI-KOLLMORGEN-U12M4-SERVODISC-DC-MOTOR-00-01202-090_W0QQitemZ7545467023QQcategoryZ92078QQssPageNameZWDVWQrdZ1QQcmdZViewlte
- [22] <http://personales.com/peru/arequipa/compupari/>
- [23] <http://www.redeya.com>
- [24] <http://sine.ni.com/nips/cds/view/p/lang/en/nid/1181>
- [25] <http://pdf.alldatasheet.com/datasheet-pdf/view/63496/MOTOROLA/LM324N.html>
- [26] <http://pdf.alldatasheet.com/datasheet-pdf/view/2849/MOTOROLA/4N28.html>
- [27] <http://www.cibernautica.com.ar/temasutiles/sistemashidraulicos/loselementos.htm>
- [28] <http://pdf.alldatasheet.com/datasheet-pdf/view/5771/MOTOROLA/TIP41C.html>
- [29] <http://html.rincondelvago.com/generador-de-corriente-continua.html>

Glosario de términos.

Actuador: Es el dispositivo que proporciona el movimiento en este caso el motor.

Automatización: Tecnología que está relacionada con el empleo de sistemas mecánicos, electrónicos y basados en las computadoras para la operación y control de la producción

Componentes ActiveX: Son formularios Delphi que actúan hacia fuera como controles ActiveX.

Convertor análogo/digital: es un dispositivo que convierte una señal analógica en una señal digital (1 y 0).

Convertor digital/análogo: es un dispositivo que convierte una señal digital en una señal analógica (corriente o voltaje).

Error: es la diferencia entre la señal de referencia y la señal de salida real.

Estación de trabajo. Minicomputadora de un solo usuario generalmente con una capacidad de gráficos de alta resolución, a una velocidad que puede poner en marcha las aplicaciones en forma independiente conjuntamente con otras computadoras por medio de una red.

Generador taquimétrico: Generador eléctrico del tipo de dínamo que al adjuntarse a un dispositivo giratorio, entrega una tensión proporcional a la velocidad de giro.

Interfaz. Es una conexión e interacción entre el Hardware, Software, Electrónica y el usuario, la cual hace capaz que llegue a existir la comunicación entre el hombre y la máquina;

Par motor: expresa la fuerza de actuación y depende de la potencia que sea capaz de desarrollar el motor, así como de la velocidad de rotación del mismo.

Perturbación: es una señal que tiende a afectar la salida del sistema, desviándola del valor deseado.

Planta: es el elemento físico que se desea controlar. Planta puede ser: un motor, un horno, un sistema de disparo, un sistema de navegación, un tanque de combustible, etc.

Proceso: operación que conduce a un resultado determinado.

Sensor: es un dispositivo que convierte el valor de una magnitud física (presión, flujo, temperatura, etc.) en una señal eléctrica codificada ya sea en forma analógica o digital. También es llamado

Simulación. Un proceso o aparato para generar condiciones de ensayo que se aproximan a las condiciones reales u operacionales. Un sistema de imitación o que se usa para simular algo distinto.

Simulador. El simulador es un programa que imita las acciones de otro programa, dispositivo de hardware u otra clase de dispositivo.

Tiempo real. Es el tiempo de respuesta idóneo(el menor tiempo posible), entre los movimientos de el usuario y las respuestas del sistema, disminuyendo en lo más posible retardos generados por estos.

Transductor. Los sensores, o transductores, analógicos envían, por lo regular, señales normalizadas de 0 a 5 voltios, 0 a 10 voltios o 4 a 20 mA.

Sensor analógico: es aquel que puede entregar una salida variable dentro de un determinado rango.

Sensor digital: es aquel que entrega una salida del tipo discreta

Señal análoga: es una señal continua en el tiempo.

Señal de control: es la señal que produce el controlador para modificar la variable controlada de tal forma que se disminuya, o elimine, el error.

Señal digital: es una señal que solo toma valores de 1 y 0. El PC solo envía y/o recibe señales digitales.

Señal de salida: es la variable que se desea controlar (posición, velocidad, presión, temperatura, etc.). También se denomina **variable controlada**.

Señal de referencia: es el valor que se desea que alcance la señal de salida.

Servomotor: Es un tipo de motor que posee tres terminales dos de ellas para alimentación y otra para control de posición del motor. Un servomotor además de un motor de cc, incorpora un conjunto de engranajes limitadores de fin de recorrido un potenciómetro para realimentar la posición del motor y un circuito integrado para el control de posición del motor.

Sistema: consiste en un conjunto de elementos que actúan coordinadamente para realizar un objetivo determinado.

Sistema de control en lazo cerrado: es aquel en el cual continuamente se está monitoreando la señal de salida para compararla con la señal de referencia y calcular la señal de error, la cual a su vez es aplicada al controlador para generar la señal de control y tratar de llevar la señal de salida al valor deseado. También es llamado **control realimentado**.

Sistema de control en lazo abierto: en estos sistemas de control la señal de salida no es monitoreada para generar una señal de control.

Apéndice 1

Apéndice de circuitos.

TIP41C [28]

Transistor de Silicio, conmutador amplificador de potencia de audio.

Características:

Corriente de colector máxima I_C	Voltaje máximo de ruptura			Ganancia de corriente inversa típica h_{FE}	Potencia máxima de disipación de colector P_D	Frecuencia típica f_T
	Colecto a Base BV_{CBO}	Colector a Emisor BV_{CEO}	Emisor a Base BV_{EBO}			
15 A	100 V	100 V	5 V	40 Min	<i>90 W</i>	3 min Mhz

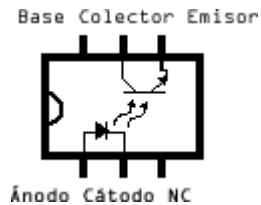
4N28 [26]

Opto-acoplador con salida de transistor NPN, la entrada es un fotodiodo compatible con TTL.

Características:

Valores de dispositivo totales			Valores máximos del LED			Valores máximos del fototransistor		
Sobre tensión de voltaje de aislamiento V_{ISO}	Potencia total P_T	Proporción de transferencia de corriente CD C_{TR}	Corriente hacia delante I_F	Voltaje hacia delante V_F	Voltaje Inverso V_R	Voltaje de Colector a Base BV_{CBO}	Voltaje de Colector a Emisor BV_{CEO}	Corriente de colector en la oscuridad I_C
7500 V	250 mW	20 %	80 mA	1.5 V	3 V	70 V	30 V	100 nA

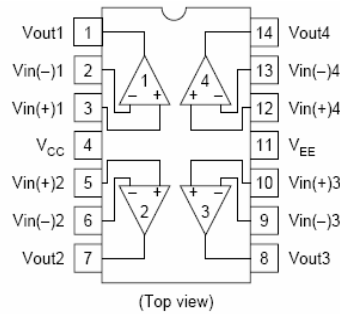
Diagrama de Circuito:



Fuente:

NTE replacements.

LM324N quad op-amp [24]




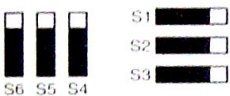
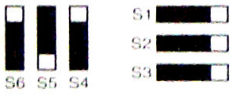

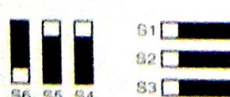
DC ELECTRICAL CHARACTERISTICS (Continued)

V_{CC}=5V, T_A=25°C unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LM324A			UNIT
			Min	Typ	Max	
V _{OS}	Offset voltage ¹	R _S =0Ω		±2	±3	mV
		R _S =0Ω, over temp.			±5	
ΔV _{OS} /ΔT	Temperature drift	R _S =0Ω, over temp.		7	30	μV/°C
I _{BIAS}	Input current ²	I _{IN} (+) or I _{IN} (-)		45	100	nA
		I _{IN} (+) or I _{IN} (-), over temp.		40	200	
ΔI _{BIAS} /ΔT	Temperature drift	Over temp.		50		pA/°C
I _{OS}	Offset current	I _{IN} (+) - I _{IN} (-)		±5	±30	nA
		I _{IN} (+) - I _{IN} (-), over temp.			±75	
ΔI _{OS} /ΔT	Temperature drift	Over temp.		10	300	pA/°C
V _{CM}	Common-mode voltage range ³	V _{CC} ≤30V	0		V _{CC} -1.5	V
		V _{CC} ≤30V, over temp.	0		V _{CC} -2	V
CMRR	Common-mode rejection ratio	V _{CC} =30V	65	85		dB
V _{OUT}	Output voltage swing	R _L =2kΩ, V _{CC} =30V, over temp.	26			V
V _{OH}	Output voltage high	R _L ≤10kΩ, V _{CC} =30V, over temp.	27	28		V
V _{OL}	Output voltage low	R _L ≤10kΩ, over temp.		5	20	mV
I _{CC}	Supply current	R _L =∞, V _{CC} =30V, over temp.		1.5	3	mA
		R _L =∞, over temp.		0.7	1.2	
A _{VOL}	Large-signal voltage gain	V _{CC} =15V (for large V _O swing), R _L ≥2kΩ	25	100		V/mV
		V _{CC} =15V (for large V _O swing), R _L ≥2kΩ, over temp.	15			V/mV
	Amplifier-to-amplifier coupling ⁵	f=1kHz to 20kHz, input referred		-120		dB
PSRR	Power supply rejection ratio	R _S =0Ω	65	100		dB
I _{OUT}	Output current source	V _{IN} + = +1V, V _{IN} - = 0V, V _{CC} =15V	20	40		mA
		V _{IN} + = +1V, V _{IN} - = 0V, V _{CC} =15V, over temp.	10	20		
	Output current sink	V _{IN} - = +1V, V _{IN} + = 0V, V _{CC} =15V	10	20		mA
		V _{IN} - = +1V, V _{IN} + = 0V, V _{CC} =15V, over temp.	5	8		
		V _{IN} - = +1V, V _{IN} + = 0V, V _O =200mV	12	50		
I _{SC}	Short-circuit current ⁴		10	40	60	mA
V _{DIFF}	Differential input voltage ³				V _{CC}	V
GBW	Unity gain bandwidth			1		MHz
SR	Slew rate			0.3		V/μs
V _{NOISE}	Input noise voltage	f=1kHz		40		nV/√Hz

Apéndice 2.

Tabla de referencia de las terminales de la tarjeta de adquisición de datos SCB-100.

SCB-100 Quick Reference Label  AT-MIO-64E-3 (AT-MIO-16DE-10) P/N 183022A-01		PIN #	SIGNAL				
FACTORY DEFAULT SETTING  * TEMP. SENSOR DISABLED * ACCESSORY POWER ON		1	AIGND	33	DGND	67	ACH32 (PB7)
		2	AIGND	34	+5V	68	ACH40 (GND)
		3	ACH0	35	+5V	69	ACH33 (PB6)
		4	ACH8	36	SCANCLK	70	ACH41 (GND)
		5	ACH1	37	EXTSTROBE*	71	ACH34 (PB5)
		6	ACH9	38	PFI0/TRIG1	72	ACH42 (GND)
		7	ACH2	39	PFI1/TRIG2	73	ACH35 (PB4)
 * TEMP. SENSOR ENABLED ON SINGLE ENDED CH 0 * ACCESSORY POWER ON		8	ACH10	40	PFI2/CONVERT*	74	ACH43 (GND)
		9	ACH3	41	PFI3/GPCTR1 SOURCE	75	AISENSE2 (PB3)
		10	ACH11	42	PFI4/GPCTR1 GATE	76	AIGND (GND)
		11	ACH4	43	GPCTR1_OUT	77	ACH36 (PB2)
		12	ACH12	44	PFI5/UPDATE*	78	ACH44 (GND)
		13	ACH5	45	PFI6/WFTRIG	79	ACH37 (PB1)
		14	ACH13	46	PFI7/STARTSCAN	80	ACH45 (GND)
 * TEMP. SENSOR ENABLED ON DIFFERENTIAL CH 0 * ACCESSORY POWER ON		15	ACH6	47	GPCTR0 SOURCE	81	ACH38 (PB0)
		16	ACH14	48	GPCTR0 GATE	82	ACH46 (GND)
		17	ACH7	49	GPCTR0_OUT	83	ACH39 (PA7)
		18	ACH15	50	FREQ_OUT	84	ACH47 (GND)
		19	AISENSE	51	ACH16 (PC7)	85	ACH48 (PA6)
		20	DAC0OUT	52	ACH24 (GND)	86	ACH56 (GND)
		21	DAC1OUT	53	ACH17 (PC6)	87	ACH49 (PA5)
 * 100 GENERIC TERMINALS (TEMP. SENSOR AND ACCESSORY POWER OFF)		22	EXTREF	54	ACH25 (GND)	88	ACH57 (GND)
		23	AOGND	55	ACH18 (PC5)	89	ACH50 (PA4)
		24	DGND	56	ACH26 (GND)	90	ACH58 (GND)
		25	DIO0	57	ACH19 (PC4)	91	ACH51 (PA3)
		26	DIO4	58	ACH27 (GND)	92	ACH59 (GND)
		27	DIO1	59	ACH20 (PC3)	93	ACH52 (PA2)
		28	DIO5	60	ACH28 (GND)	94	ACH60 (GND)
29	DIO2	61	ACH21 (PC2)	95	ACH53 (PA1)		
30	DIO6	62	ACH29 (GND)	96	ACH61 (GND)		
31	DIO3	63	ACH22 (PC1)	97	ACH54 (PA0)		
32	DIO7	64	ACH30 (GND)	98	ACH62 (GND)		
		65	ACH23 (PC0)	99	ACH55 (+5V)		
		66	ACH31 (GND)	100	ACH63 (GND)		

Apéndice 3

Apéndice de Programas en lenguaje Delphi 5.

Programa del control PD.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
  SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls;

type
  TForm1 = class(TForm)
    CEA: TCWAIPoint;
    Taco: TCWKnob;
    BEnc: TCWButton;
    CSA: TCWAOPoint;
    EVd: TSCFloatSpinEdit;
    Wn: TSCFloatSpinEdit;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    Memo1: TMemo;
    SCButton1: TSCButton;
    procedure BEncClick(Sender: TObject);
    procedure SCButton1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    procedure Control;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  e,ep,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);
  for i:=0 to 2000 do
    begin
      Control;
    end;
  end;

```

```

Application.ProcessMessages;
if GetKeyState(VK_ESCAPE)and 128=128 then break;
end;
BEnc.Value:=False;
CSA.SingleWrite(0, True);
end;

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

function fudt:Single;
var kp,kd:Single;
begin
kp:=wn.FloatValue*wn.FloatValue;
kd:=2*Wn.FloatValue;
e[varN]:=Evd.FloatValue-avel[varN];
ep[varN]:=-(avel[varN]-avel[varN-1])/lapso[varN-1];
result:=kp*e[varN]+kd*ep[varN];
end;

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:=avel[varN];

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;
if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
//

QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1 Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do

```



```

Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
    FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+' '+FloatToStr(i));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\VPD.mat');
end;

```

```

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

```

```

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PD con regulación y seguimiento.

```

unit Unit1;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
    SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls, Math;

type
TForm1 = class(TForm)
    CEA: TCWAIPoint;
    Taco: TCWKnob;
    BEnc: TCWButton;
    CSA: TCWAOPoint;
    Vd: TSCFloatSpinEdit;
    Wn: TSCFloatSpinEdit;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    Memo1: TMemo;
    SCButton1: TSCButton;
    ETb: TSCFloatSpinEdit;
    StaticText4: TStaticText;
    EL: TSCFloatSpinEdit;
    StaticText3: TStaticText;
    procedure BEncClick(Sender: TObject);
    procedure SCButton1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
    procedure Control;
public
    { Public declarations }
end;

```

```

var
  Form1: TForm1;
  Tref,e,ep,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);
  for i:=0 to 20000 do
    begin
      Control;
      Application.ProcessMessages;
      if GetKeyState(VK_ESCAPE)and 128=128 then break;
    end;
    BEnc.Value:=False;
    CSA.SingleWrite(0, True);
  end;

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

function fudt:Single;
var L,tao,taoDer,Tb,kp,kd:Single;
begin
  L:=EL.FloatValue;
  kp:=wn.FloatValue*wn.FloatValue;
  kd:=2*Wn.FloatValue;
  Tb:= ETb.FloatValue*1000;

  tao:=(10*power(tiempo[varN],3)/power(Tb,3) //Splina a seguir
    - 15*power(tiempo[varN],4)/power(Tb,4)
    + 6*power(tiempo[varN],5)/power(Tb,5));
  if tiempo[varN]>Tb then tao:=1; //Condición para cuando llegue

  taoDer:=(30*power(tiempo[varN],2)/power(Tb,3) //Derivada de la Splina
    -60*power(tiempo[varN],3)/power(Tb,4)
    +30*power(tiempo[varN],4)/power(Tb,5));
  if tiempo[varN]>Tb then taoDer:=0; //Condición de la derivada para cuando llegue

  Tref[varN]:=vd.FloatValue*tao;
  e[varN]:=avel[varN]-Tref[varN];
  ep[varN]:=-L*avel[varN]+kd*e[varN];
  result:=-kp*e[varN]-kd*(ep[varN]);
end;

```

```

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:= avel[varN];

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;
if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do
Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+' '+FloatToStr(i)+' '+FloatToStr(Tref[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPXSeg.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PI.

```

unit Unit1;
interface

```

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
 OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
 SCMaskEdit, SCAAdvEdits, SCStdControls, StdCtrls;

type

```
TForm1 = class(TForm)
  CEA: TCWAIPoint;
  Taco: TCWKnob;
  BEnc: TCWButton;
  CSA: TCWAOPoint;
  Vd: TSCFloatSpinEdit;
  Wn: TSCFloatSpinEdit;
  StaticText1: TStaticText;
  StaticText2: TStaticText;
  Memo1: TMemo;
  SCButton1: TSCButton;
  eki: TSCFloatSpinEdit;
  StaticText3: TStaticText;
  procedure BEncClick(Sender: TObject);
  procedure SCButton1Click(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  procedure Control;
public
  { Public declarations }
end;
```

var

```
Form1: TForm1;
e,ep,Int,audt,avel,tiempo,lapso:Array[0..20000] of Single;
varN:Integer=1;
start1,stop1:TLargeInteger;
```

implementation

```
{ $R *.DFM }
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);
  for i:=0 to 2000 do
    begin
      Control;
      Application.ProcessMessages;
      if GetKeyState(VK_ESCAPE)and 128=128 then break;
    end;
  BEnc.Value:=False;
  CSA.SingleWrite(0, True);
end;
```

```

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

function Integral(a,b:Single):Single;
var h,k1,k2,k4:Single;
begin
    h:= b-a;
    k1:=a;
    k2:=a+h/2;
    k4:=a+h;
    result:=h*(k1+4*k2+k4)/6;
end;

function fudt:Single;
var kp,kd,ki:Single;
begin
    kp:=wn.FloatValue*wn.FloatValue;
    //kd:=2*Wn.FloatValue;
    ki:=kp*kd/eki.FloatValue;
    e[varN]:=vd.FloatValue-avel[varN];
    //ep[varN]:=-(avel[varN]-avel[varN-1])/lapso[varN-1];
    Int[varN]:=Integral(e[varN-1],e[varN]);
    //result:=kp*e[varN]+kd*ep[varN]+ki*Int[varN];
    result:=kp*e[varN]+ki*Int[varN];
end;

begin
    QueryPerformanceFrequency(freq);
    QueryPerformanceCounter(Start);
    QueryPerformanceCounter(Stop1);
    tiempo[varN]:=(Stop1-Start1)/freq*1000;

    // Se captura la velocidad
    CEA.SingleRead(vol);
    avel[varN]:= (2.3662*vol-0.013)*30/pi;

    //Visualización
    Taco.Value:= avel[varN];

    // Control
    audt[varN]:=fudt;
    udt:=0.4224*audt[varN]+0.0089;
    if udt>5 then udt:=5;
    if udt<0 then udt:=0;
    CSA.SingleWrite(udt, True);
    QueryPerformanceCounter(Stop);
    lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
    inc(varN);
end;

```

```

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do
  Memo1.Lines.Add(FloatToStr(tiempo[i]+' '+FloatToStr(lapso[i]+' '+FloatToStr(avel[i])+
    '+FloatToStr(audt[i]+' '+FloatToStr(e[i]+' '+IntToStr(i)+' '+FloatToStr(Int[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPI\CPI.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PI con regulación y seguimiento.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
  SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls, Math;

type
  TForm1 = class(TForm)
    CEA: TCWAIPoint;
    Taco: TCWKnob;
    BEnc: TCWButton;
    CSA: TCWAOPoint;
    Vd: TSCFloatSpinEdit;
    Wn: TSCFloatSpinEdit;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    Memo1: TMemo;
    SCButton1: TSCButton;
    eki: TSCFloatSpinEdit;
    StaticText3: TStaticText;
    ETb: TSCFloatSpinEdit;
    StaticText4: TStaticText;
    procedure BEncClick(Sender: TObject);
    procedure SCButton1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  end;

```

```

private
  procedure Control;
public
  { Public declarations }
end;

var
  Form1: TForm1;
  Tref,e,ep,Int,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);
  for i:=0 to 2000 do
    begin
      Control;
      Application.ProcessMessages;
      if GetKeyState(VK_ESCAPE)and 128=128 then break;
    end;
  BEnc.Value:=False;
  CSA.SingleWrite(0, True);
end;

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

function Integral(a,b:Single):Single;
var h,k1,k2,k4:Single;
begin
  h:= b-a;
  k1:=a;
  k2:=a+h/2;
  k4:=a+h;
  result:=h*(k1+4*k2+k4)/6;
end;

function fudt:Single;
var tao,kp,kd,ki,Tb:Single;
begin
  kp:=wn.FloatValue*wn.FloatValue;
  ki:=kp*kd/eki.FloatValue;
  Tb:= ETb.FloatValue*1000;

  tao:=(10*power(tiempo[varN],3)/power(Tb,3) //Splina a seguir

```

```

- 15*power(tiempo[varN],4)/power(Tb,4)
+ 6*power(tiempo[varN],5)/power(Tb,5));
if tiempo[varN]>Tb then tao:=1;           //Condición para cuando llegue

Tref[varN]:=vd.FloatValue*tao;
e[varN]:=Tref[varN]-avel[varN];
Int[varN]:=Integral(e[varN-1],e[varN]);
result:=kp*e[varN]+ki*Int[varN];
end;

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:= avel[varN];

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;
if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do
Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+IntToStr(i)+' '+FloatToStr(Tref[i])+' '+FloatToStr(Int[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPISeg.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin

```



```

CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PID.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
  SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls;

type
  TForm1 = class(TForm)
    CEA: TCWAIPoint;
    Taco: TCWKnob;
    BEnc: TCWButton;
    CSA: TCWAOPoint;
    Vd: TSCFloatSpinEdit;
    Wn: TSCFloatSpinEdit;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    Memo1: TMemo;
    SCButton1: TSCButton;
    eki: TSCFloatSpinEdit;
    StaticText3: TStaticText;
    procedure BEncClick(Sender: TObject);
    procedure SCButton1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    procedure Control;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  e,ep,Int,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);

```

```

for i:=0 to 2000 do
  begin
  Control;
  Application.ProcessMessages;
  if GetKeyState(VK_ESCAPE)and 128=128 then break;
  end;
BEnc.Value:=False;
CSA.SingleWrite(0, True);
end;

```

```

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

```

```

function Integral(a,b:Single):Single;
var h,k1,k2,k4:Single;
begin
h:= b-a;
k1:=a;
k2:=a+h/2;
k4:=a+h;
result:=h*(k1+4*k2+k4)/6;
end;

```

```

function fudt:Single;
var kp,kd,ki:Single;
begin
kp:=wn.FloatValue*wn.FloatValue;
kd:=2*Wn.FloatValue;
ki:=kp*kd/eki.FloatValue;
e[varN]:=vd.FloatValue-avel[varN];
ep[varN]:=- (avel[varN]-avel[varN-1])/lapso[varN-1];
Int[varN]:=Integral(e[varN-1],e[varN]);
result:=kp*e[varN]+kd*ep[varN]+ki*Int[varN];
end;

```

```

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

```

```

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

```

```

//Visualización
Taco.Value:= avel[varN];

```

```

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;

```

```

if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do
Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+' '+IntToStr(i)+' '+FloatToStr(Int[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPID.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PID con regulación y seguimiento.

```

unit Unit1;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls, Math;

type
TForm1 = class(TForm)
CEA: TCWAIPoint;
Taco: TCWKnob;
BEnc: TCWButton;
CSA: TCWAOPoint;
Vd: TSCFloatSpinEdit;
Wn: TSCFloatSpinEdit;
StaticText1: TStaticText;
StaticText2: TStaticText;
Memo1: TMemo;
SCButton1: TSCButton;

```

```

eki: TSCFloatSpinEdit;
StaticText3: TStaticText;
ETb: TSCFloatSpinEdit;
StaticText4: TStaticText;
procedure BEncClick(Sender: TObject);
procedure SCButton1Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  procedure Control;
public
  { Public declarations }
end;

var
  Form1: TForm1;
  S,Int,Tref,e,ep,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);
  for i:=0 to 20000 do
    begin
      Control;
      Application.ProcessMessages;
      if GetKeyState(VK_ESCAPE)and 128=128 then break;
    end;
  BEnc.Value:=False;
  CSA.SingleWrite(0, True);
end;

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

function integralTanh(sv,sn:Single):Single;
var xi,h,k1,k2,k4:Single;
begin
  xi:= Sv*100;
  h:= Sn*100-xi;
  k1:=tanh(xi);
  k2:=tanh(xi+h/2);
  k4:=tanh(xi+h);
  result:=h*(k1+4*k2+k4)/6;
end;

```

```

function fudt:Single;
var alfa,tao,taoDer,Tb,kp,kd,ki:Single;
begin
kp:=wn.FloatValue*wn.FloatValue;
kd:=2*Wn.FloatValue;
ki:=kp*kd/eki.FloatValue;
alfa:=Wn.FloatValue/2;
Tb:= ETb.FloatValue*1000;
tao:=(10*power(tiempo[varN],3)/power(Tb,3) //Splina a seguir
- 15*power(tiempo[varN],4)/power(Tb,4)
+ 6*power(tiempo[varN],5)/power(Tb,5));
if tiempo[varN]>Tb then tao:=1; //Condición para cuando llegue
taoDer:=(30*power(tiempo[varN],2)/power(Tb,3) //Derivada de la Splina
-60*power(tiempo[varN],3)/power(Tb,4)
+30*power(tiempo[varN],4)/power(Tb,5));
if tiempo[varN]>Tb then taoDer:=0; //Condición de la derivada para cuando llegue

Tref[varN]:=vd.FloatValue*tao;
e[varN]:=Tref[varN]-avel[varN];
ep[varN]:=taoDer-(avel[varN]-avel[varN-1])/lapso[varN-1];
s[varN]:= alfa*e[varN]+ep[varN];
Int[varN]:= integralTanh(s[varN-1],s[varN]);
Result:= kd*s[varN]+ki*Int[varN];
end;

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:= avel[varN];

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;
if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;

```

```

begin
Memo1.Clear;
for i:=1 to varN-1 do
  Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
    FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+
    '+'+FloatToStr(i)+' '+FloatToStr(Tref[i])+' '+FloatToStr(Int[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPIDNLSeg.mat');
end;
procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PIDNL.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
  SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls, Math;

type
TForm1 = class(TForm)
  CEA: TCWAIPoint;
  Taco: TCWKnob;
  BEnc: TCWButton;
  CSA: TCWAOPoint;
  Vd: TSCFloatSpinEdit;
  Wn: TSCFloatSpinEdit;
  StaticText1: TStaticText;
  StaticText2: TStaticText;
  Memo1: TMemo;
  SCButton1: TSCButton;
  eki: TSCFloatSpinEdit;
  StaticText3: TStaticText;
  procedure BEncClick(Sender: TObject);
  procedure SCButton1Click(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  procedure Control;
public
  { Public declarations }
end;

```

```

var
  Form1: TForm1;
  S,Int,Tref,e,ep,audt,avel,tiempo,lapso:Array[0..2000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

```

implementation

```
{$R *.DFM}
```

```
procedure TForm1.BEncClick(Sender: TObject);
```

```
var i:Integer;
```

```
begin
```

```
varN:=1;
```

```
QueryPerformanceCounter(Start1);
```

```
for i:=0 to 2000 do
```

```
begin
```

```
Control;
```

```
Application.ProcessMessages;
```

```
if GetKeyState(VK_ESCAPE)and 128=128 then break;
```

```
end;
```

```
BEnc.Value:=False;
```

```
CSA.SingleWrite(0, True);
```

```
end;
```

```
procedure TForm1.Control;
```

```
var udt,vol: OleVariant;
```

```
freq, start, stop: TLargeInteger;
```

```
function integralTanh(sv,sn:Single):Single;
```

```
var xi,h,k1,k2,k4:Single;
```

```
begin
```

```
xi:= Sv*100;
```

```
h:= Sn*100-xi;
```

```
k1:=tanh(xi);
```

```
k2:=tanh(xi+h/2);
```

```
k4:=tanh(xi+h);
```

```
result:=h*(k1+4*k2+k4)/6;
```

```
end;
```

```
function fudt:Single;
```

```
var alfa,kp,kd,ki:Single;
```

```
begin
```

```
kp:=wn.FloatValue*wn.FloatValue;
```

```
kd:=2*Wn.FloatValue;
```

```
ki:=kp*kd/eki.FloatValue;
```

```
alfa:=Wn.FloatValue/2;
```

```
Tref[varN]:=vd.FloatValue;
```

```
e[varN]:=Tref[varN]-avel[varN];
```

```
ep[varN]:=- (avel[varN]-avel[varN-1])/lapso[varN-1];
```

```
s[varN]:= alfa*e[varN]+ep[varN];
```

```

Int[varN]:= integralTanh(s[varN-1],s[varN]);
Result:= kd*s[varN]+ki*Int[varN];
end;

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:= avel[varN];

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;
if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do
Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+' '+
'+FloatToStr(i)+' '+FloatToStr(Tref[i])+' '+FloatToStr(Int[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPIDNL.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```


Programa del control PIDNL con regulación y seguimiento.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
  SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls, Math;

type
TForm1 = class(TForm)
  CEA: TCWAIPoint;
  Taco: TCWKnob;
  BEnc: TCWButton;
  CSA: TCWAOPoint;
  Vd: TSCFloatSpinEdit;
  Wn: TSCFloatSpinEdit;
  StaticText1: TStaticText;
  StaticText2: TStaticText;
  Memo1: TMemo;
  SCButton1: TSCButton;
  eki: TSCFloatSpinEdit;
  StaticText3: TStaticText;
  ETb: TSCFloatSpinEdit;
  StaticText4: TStaticText;
  procedure BEncClick(Sender: TObject);
  procedure SCButton1Click(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  procedure Control;
public
  { Public declarations }
end;

var
  Form1: TForm1;
  S,Int,Tref,e,ep,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);
  for i:=0 to 20000 do

```

```

begin
Control;
Application.ProcessMessages;
if GetKeyState(VK_ESCAPE)and 128=128 then break;
end;
BEnc.Value:=False;
CSA.SingleWrite(0, True);
end;

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

function integralTanh(sv,sn:Single):Single;
var xi,h,k1,k2,k4:Single;
begin
xi:= Sv*100;
h:= Sn*100-xi;
k1:=tanh(xi);
k2:=tanh(xi+h/2);
k4:=tanh(xi+h);
result:=h*(k1+4*k2+k4)/6;
end;

function fudt:Single;
var alfa,tao,taoDer,Tb,kp,kd,ki:Single;
begin
kp:=wn.FloatValue*wn.FloatValue;
kd:=2*Wn.FloatValue;
ki:=kp*kd/eki.FloatValue;
alfa:=Wn.FloatValue/2;
Tb:= ETb.FloatValue*1000;
tao:=(10*power(tiempo[varN],3)/power(Tb,3) //Splina a seguir
- 15*power(tiempo[varN],4)/power(Tb,4)
+ 6*power(tiempo[varN],5)/power(Tb,5));
if tiempo[varN]>Tb then tao:=1; //Condición para cuando llegue
taoDer:=(30*power(tiempo[varN],2)/power(Tb,3) //Derivada de la Splina
-60*power(tiempo[varN],3)/power(Tb,4)
+30*power(tiempo[varN],4)/power(Tb,5));
if tiempo[varN]>Tb then taoDer:=0; //Condición de la derivada para cuando llegue
Tref[varN]:=vd.FloatValue*tao;
e[varN]:=Tref[varN]-avel[varN];
ep[varN]:=taoDer-(avel[varN]-avel[varN-1])/lapso[varN-1];
s[varN]:= alfa*e[varN]+ep[varN];
Int[varN]:= integralTanh(s[varN-1],s[varN]);
Result:= kd*s[varN]+ki*Int[varN];
end;

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);

```

```

tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:= avel[varN];

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;
if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do
Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+
'+FloatToStr(i)+' '+FloatToStr(Tref[i])+' '+FloatToStr(Int[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPIDNLSeg.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PX.

```

unit Unit1;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls;

```

```

type
  TForm1 = class(TForm)
    CEA: TCWAIPoint;
    Taco: TCWKnob;
    BEnc: TCWButton;
    CSA: TCWAOPoint;
    Vd: TSCFloatSpinEdit;
    Wn: TSCFloatSpinEdit;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    Memo1: TMemo;
    SCButton1: TSCButton;
    eL: TSCFloatSpinEdit;
    StaticText3: TStaticText;
    procedure BEncClick(Sender: TObject);
    procedure SCButton1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    procedure Control;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  e,ep,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1 ,stop1 :TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);
var i:Integer;
begin
  varN:=1;

  QueryPerformanceCounter(Start1);
  for i:=0 to 2000 do
    begin
      Control;
      Application.ProcessMessages;
      if GetKeyState(VK_ESCAPE)and 128=128 then break;
    end;
  BEnc.Value:=False;
  CSA.SingleWrite(0, True);
end;

procedure TForm1.Control;
var udt,vol: OleVariant;
    freq, start, stop: TLargeInteger;

```

```

function fudt:Single;
var kp,kd,L:Single;
begin
  L:= eL.FloatValue;
  kp:=wn.FloatValue*wn.FloatValue;
  kd:=2*Wn.FloatValue;
  e[varN]:=avel[varN]-vd.FloatValue;
  ep[varN]:=-L*avel[varN]+kd*e[varN];
  result:=-kp*e[varN]-kd*ep[varN];
end;

begin
  QueryPerformanceFrequency(freq);
  QueryPerformanceCounter(Start);
  QueryPerformanceCounter(Stop1);
  tiempo[varN]:=(Stop1-Start1)/freq*1000;

  // Se captura la velocidad
  CEA.SingleRead(vol);
  avel[varN]:= (2.3662*vol-0.013)*30/pi;

  //Visualización
  Taco.Value:= avel[varN];

  // Control
  audt[varN]:=fudt;
  udt:=0.4224*audt[varN]+0.0089;
  if udt>5 then udt:=5;
  if udt<0 then udt:=0;
  CSA.SingleWrite(udt, True);

  QueryPerformanceCounter(Stop);
  lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
  inc(varN);
end;

procedure TForm1.SCButton1 Click(Sender: TObject);
var i:Integer;
begin
  Memo1.Clear;
  for i:=1 to varN-1 do
    Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
      FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+' '+FloatToStr(i));
  Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPX.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
  avel[0]:=0;
  lapso[0]:=0;
end;

```

```

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programa del control PX con regulación y seguimiento.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  OleCtrls, CWDAQControlsLib_TLB, CWUIControlsLib_TLB, SCControl, SCEdits,
  SCMaskEdit, SCAdvEdits, SCStdControls, StdCtrls, Math;

type
  TForm1 = class(TForm)
    CEA: TCWAIPoint;
    Taco: TCWKnob;
    BEnc: TCWButton;
    CSA: TCWAOPoint;
    Vd: TSCFloatSpinEdit;
    Wn: TSCFloatSpinEdit;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    Memo1: TMemo;
    SCButton1: TSCButton;
    eL: TSCFloatSpinEdit;
    StaticText3: TStaticText;
    ETb: TSCFloatSpinEdit;
    StaticText4: TStaticText;
    procedure BEncClick(Sender: TObject);
    procedure SCButton1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    procedure Control;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  Tref,e,ep,audt,avel,tiempo,lapso:Array[0..20000] of Single;
  varN:Integer=1;
  start1,stop1:TLargeInteger;

implementation
{$R *.DFM}
procedure TForm1.BEncClick(Sender: TObject);

```

```

var i:Integer;
begin
varN:=1;

QueryPerformanceCounter(Start1);
for i:=0 to 2000 do
begin
Control;
Application.ProcessMessages;
if GetKeyState(VK_ESCAPE)and 128=128 then break;
end;
BEnc.Value:=False;
CSA.SingleWrite(0, True);
end;

procedure TForm1.Control;
var udt,vol: OleVariant;
freq, start, stop: TLargeInteger;

function fudt:Single;
var
Tb,tao,kp,kd,L:Single;
begin
L:= eL.FloatValue;
kp:=wn.FloatValue*wn.FloatValue;
kd:=2*Wn.FloatValue;
Tb:= ETb.FloatValue*1000;

tao:=(10*power(tiempo[varN],3)/power(Tb,3) //Splina a seguir
- 15*power(tiempo[varN],4)/power(Tb,4)
+ 6*power(tiempo[varN],5)/power(Tb,5));
if tiempo[varN]>Tb then tao:=1; //Condición para cuando llegue

Tref[varN]:=vd.FloatValue*tao;
e[varN]:=avel[varN]-Tref[varN];
ep[varN]:=-L*avel[varN]+kd*e[varN];
result:=-kp*e[varN]-kd*ep[varN];
end;

begin
QueryPerformanceFrequency(freq);
QueryPerformanceCounter(Start);
QueryPerformanceCounter(Stop1);
tiempo[varN]:=(Stop1-Start1)/freq*1000;

// Se captura la velocidad
CEA.SingleRead(vol);
avel[varN]:= (2.3662*vol-0.013)*30/pi;

//Visualización
Taco.Value:= avel[varN];

```

```

// Control
audt[varN]:=fudt;
udt:=0.4224*audt[varN]+0.0089;
if udt>5 then udt:=5;
if udt<0 then udt:=0;
CSA.SingleWrite(udt, True);
QueryPerformanceCounter(Stop);
lapso[varN]:=(Stop-Start)/freq*1000; //en milisegundos
inc(varN);
end;

procedure TForm1.SCButton1Click(Sender: TObject);
var i:Integer;
begin
Memo1.Clear;
for i:=1 to varN-1 do
Memo1.Lines.Add(FloatToStr(tiempo[i])+' '+FloatToStr(lapso[i])+' '+FloatToStr(avel[i])+' '+
FloatToStr(audt[i])+' '+FloatToStr(e[i])+' '+FloatToStr(ep[i])+ ' '+FloatToStr(i)+' '+FloatToStr(Tref[i]));
Memo1.Lines.SaveToFile('C:\Tesis\Febrero14\Programa\CPD\CPXSeg.mat');
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
avel[0]:=0;
lapso[0]:=0;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
CSA.SingleWrite(0, True);
end;
end.

```

Programas de Simulación en Matlab 6.

Simulación del Comportamiento del Servomotor

```

clc % Borra pantalla
clear all % Borra variables
close all % Cierra ventanas (Gráficas)

global dx
dx=zeros(2,1);
options=odeset('MaxStep',0.1,'InitialStep',0.1);
[t,x]=ode45('motor',[0 10],[0 0],options);

figure(1)
plot(t,x(:,1),'b');
title('Intensidad de corriente de armadura');
xlabel('t');
ylabel('i(t)');

```



```

grid

figure(2)
plot(t,x(:,2),'b');
title('Velocidad del motor ');
xlabel('t');
ylabel('w(t)');
grid
end

```

```

function dx = motor(t,x)
global dx
La = 0.04;
Ra = 3.5;
kb = 0.1815;
kt = 1.2;
ja = 0.04;
fa = 0.17;
va = 12;

dx(1,1)=(va/La)-((Ra/La)*x(1))-((kb/La)*x(2));
dx(2,1)=((kt/ja)*x(1))-((fa/ja)*x(2));

```

Simulación del comportamiento del sensor.

```

f = [0 2.45 5.4 9.1 14 16.5 19 21.3 23.3 26.5 28.2 30.3 32.3 33.3 33.9]
v = [0 1 2.3 3.9 5.9 7 8 9.1 10.1 11 12 13 13.5 14 14.4]

```

```

n = length(v);

for i = 1:n
    sumxy = v(i)*f(i);
    sumx2 = v(i)^2;
    sumx = v(i);
    sumy = f(i);
end

a1 = (n*(sumxy)-sumx*sumy)/(n*sumx2-sumx^2);
a0 = 1/n*sumy-a1/n*sumx;
fa = a0 + a1*v;
hold on;
plot (v,f,'b',v,fa,'r');
hold off;

```

Simulación del control P.

```

clc                % Borra pantalla
clear all         % Borra variables
close all        % Cierra ventanas (Gráficas)

```

```

global dx
dx=zeros(2,1);

options=odeset('MaxStep',0.1,'InitialStep',0.1);
[t,x]=ode45('motor',[0 10],[0 0],options);

figure(1)
plot(t,x(:,1),'b');
title('Intensidad de corriente de armadura');
xlabel('t');
ylabel('i(t)');
grid

figure(2)
plot(t,x(:,2),'b');
title('Velocidad del motor ');
xlabel('t');
ylabel('w(t)');
grid
end

function dx = motor(t,x)
global dx
La = 0.04;
Ra = 3.5;
kb = 0.1815;
kt = 1.2;
ja = 0.04;
fa = 0.17;
kp=64;
wd=135;
w=x(2);
error=wd-w;

va = kp*error;

dx(1,1)=(va/La)-((Ra/La)*x(1))-((kb/La)*x(2));
dx(2,1)=((kt/ja)*x(1))-((fa/ja)*x(2));

```

Simulación del control PD.

```

clc % Borra pantalla
clear all % Borra variables
close all % Cierra ventanas (Gráficas)

global dx
dx=zeros(2,1);

options=odeset('MaxStep',0.1,'InitialStep',0.1);
[t,x]=ode45('motor',[0 10],[0 0],options);

figure(1)

```

```

plot(t,x(:,1),'b');
title('Intensidad de corriente de armadura');
xlabel('t');
ylabel('i(t)');
grid

```

```

figure(2)
plot(t,x(:,2),'b');
title('Velocidad del motor ');
xlabel('t');
ylabel('w(t)');
grid
end

```

```

function dx = motor(t,x)
global dx
La = 0.04;
Ra = 3.5;
kb = 0.1815;
kt = 1.2;
ja = 0.04;
fa = 0.17;
kp=64;
kd=16;
wd=135;
w=x(2);
wp=((kt/ja)*x(1))-((fa/ja)*x(2));
error=wd-w;

va = kp*error-kd*wp;

dx(1,1)=(va/La)-((Ra/La)*x(1))-((kb/La)*x(2));
dx(2,1)=((kt/ja)*x(1))-((fa/ja)*x(2));

```

Simulación del control PID.

```

clc % Borra pantalla
clear all % Borra variables
close all % Cierra ventanas (Gráficas)

global dx
dx=zeros(3,1);

options=odeset('MaxStep',0.1,'InitialStep',0.1);
[t,x]=ode45('motor',[0 10],[0 0 0],options);

figure(1)
plot(t,x(:,1),'b');
title('Intensidad de corriente de armadura');
xlabel('t');
ylabel('i(t)');

```

```

grid

figure(2)
plot(t,x(:,2),'b');
title('Velocidad del motor' );
xlabel('t');
ylabel('w(t)');
grid
end

function dx = motor(t,x)
global dx
La = 0.04;
Ra = 3.5;
kb = 0.1815;
kt = 1.2;
ja = 0.04;
fa = 0.17;
kp=64;
kd=16;
ki=2.2;
wd=135;
w=x(2);
wp=((kt/ja)*x(1))-((fa/ja)*x(2));
error=wd-w;

va = kp*error-kd*wp+ki*x(3);

dx(1,1)=(va/La)-((Ra/La)*x(1))-((kb/La)*x(2));
dx(2,1)=((kt/ja)*x(1))-((fa/ja)*x(2));
dx(3,1)=error;

```

Simulación del Control PX.

```

clc % Borra pantalla
clear all % Borra variables
close all % Cierra ventanas (Gráficas)

global dx
dx=zeros(2,1);

options=odeset('MaxStep',0.1,'InitialStep',0.1);
[t,x]=ode45('PX',[0 10],[0 0],options);

vecn=size(x(:,1));
vecn2=ones(vecn);
Wref=135;
Wr=Wref*vecn2;

figure(1)
plot(t,Wr,'r',t,x(:,1),'b');

```

```

title('Velocidad angular');
xlabel('t (segundos)');
ylabel('W(t) (rad/seg)');
grid
end

function dx = PX(t,x)
global dx
kt=.1;
kb=.1;
j=.25;
f=.1;
ra=10;
la=2;
tb = 5;

Si = 1/2+1/2*sin(pi/tb*(t-tb/2));
dSi = 1/2*(pi/tb)*cos(pi/tb*(t-tb/2));
ddSi = -1/2*(pi/tb)^2*sin(pi/tb*(t-tb/2));

if t >= tb
    Si = 1;
    dSi = 0;
    ddSi = 0;
end

kp=64;
kd=16;
Wref=135*Si;
Wreal=x(1);
Ew=Wreal-Wref;
L=0.63;
Ewp=-L*x(1)+kd*Ew;
va=-kp*Ew-kd*Ewp;

dx(1,1)=(kt/j)*x(2)-(f/j)*x(1);
dx(2,1)=(1/la)*va-(ra/la)*x(2)-(kb/la)*x(1);

```

Programa de control PD.

```

clc % Borra pantalla
clear all % Borra variables
close all % Cierra ventanas (Gráficas)

load VPDP.mat -ascii % Lee archivo de datos

Tiempo = VPDP(:,1); % Tiempo
Lapsos = VPDP(:,2); % Lapsos
Vel = VPDP(:,3); % Velocidad
Control = VPDP(:,4); % Control
Error = VPDP(:,5); % Error
EP = VPDP(:,6); % Derivada del Error

```

```

Muestras = VPDP(:,7);           % Muestras
%Tref     = VPDP(:,8);         % Teta de referencia

```

```

n=length(Tiempo);
for i=1:n
    Tref(i)=135;
end

```

```

figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PD)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid

```

```

figure(2)
plot(Tiempo , EP)
title('Derivada del error (Control PD)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid

```

```

figure(3)
plot(Tiempo , Error)
title('Error de velocidad (Control PD)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid

```

```

figure(4)
plot(Tiempo , Control)
title('Señal de control (Control PD)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid

```

```

figure(5)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PD)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
gris

```

Programa control PD con Seguimiento.

```

clc           % Borra pantalla
clear all    % Borra variables
close all    % Cierra ventanas (Gráficas)

load CPDSeg.mat -ascii      % Lee archivo de datos

```

```
Tiempo = CPDSeg(:,1);      % Tiempo
Lapsos  = CPDSeg(:,2);    % Lapsos
Vel     = CPDSeg(:,3);    % Velocidad
Control = CPDSeg(:,4);    % Control
Error   = CPDSeg(:,5);    % Error
EP      = CPDSeg(:,6);    % Derivada del Error
Muestras = CPDSeg(:,7);  % Muestras
Tref    = CPDSeg(:,8);    % Teta de referencia
```

```
figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PD con seguimiento)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid
```

```
figure(2)
plot(Tiempo , EP)
title('Derivada del error (Control PD con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid
```

```
figure(3)
plot(Tiempo , Error)
title('Error de velocidad (Control PD con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid
```

```
figure(4)
plot(Tiempo , Control)
title('Señal de control (Control PD con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid
```

```
figure(5)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PD con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

Programa control PI.

```
clc          % Borra pantalla
clear all   % Borra variables
close all   % Cierra ventanas (Gráficas)

load CPIP.mat -ascii      % Lee archivo de datos
```

```
Tiempo = CPIP(:,1); % Tiempo
Lapsos = CPIP(:,2); % Lapsos
Vel = CPIP(:,3); % Velocidad
Control = CPIP(:,4); % Control
Error = CPIP(:,5); % Error
Muestras = CPIP(:,6); % Muestras
Integral = CPIP(:,7); % Integral
```

```
n=length(Tiempo);
for i=1:n
    Tref(i)=135;
end
```

```
figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PI)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid
```

```
figure(2)
plot(Tiempo , Error)
title('Error de velocidad (Control PI)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid
```

```
figure(3)
plot(Tiempo , Control)
title('Señal de control (Control PI)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid
```

```
figure(4)
plot(Tiempo , Integral)
title('Indice de integracion (Control PI)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

```
figure(5)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PI)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```


Programa de control PI con seguimiento.

```
clc                % Borra pantalla
clear all         % Borra variables
close all        % Cierra ventanas (Gráficas)
```

```
load CPISeg.mat -ascii      % Lee archivo de datos
```

```
Tiempo  = CPISeg(:,1);      % Tiempo
Lapsos  = CPISeg(:,2);      % Lapsos
Vel      = CPISeg(:,3);      % Velocidad
Control = CPISeg(:,4);      % Control
Error    = CPISeg(:,5);      % Error
Muestras = CPISeg(:,6);      % Muestras
Tref     = CPISeg(:,7);      % Teta de referencia
Integral = CPISeg(:,8);      % Integral del error
```

```
figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PI con seguimiento)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid
```

```
figure(2)
plot(Tiempo , Error)
title('Error de velocidad (Control PI con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid
```

```
figure(3)
plot(Tiempo , Control)
title('Señal de control (Control PI con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid
```

```
figure(4)
plot(Tiempo , Integral)
title('Indice de integracion (Control PI con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

```
figure(5)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PI con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
```

grid

Programa PID.

```

clc                                % Borra pantalla
clear all                          % Borra variables
close all                          % Cierra ventanas (Gráficas)

load CPIDP.mat -ascii              % Lee archivo de datos
Tiempo = CPIDP(:,1);               % Tiempo
Lapsos = CPIDP(:,2);               % Lapsos
Vel = CPIDP(:,3);                  % Velocidad
Control = CPIDP(:,4);              % Control
Error = CPIDP(:,5);                % Error
EP = CPIDP(:,6);                   % Derivada del Error
Muestras = CPIDP(:,7);             % Muestras
Integral = CPIDP(:,8);              % Teta de referencia

n=length(Tiempo);
for i=1:n
    Tref(i)=135;
end

figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PID)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid

figure(2)
plot(Tiempo , EP)
title('Derivada del error (Control PID)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid

figure(3)
plot(Tiempo , Error)
title('Error de velocidad (Control PID)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid

figure(4)
plot(Tiempo , Control)
title('Señal de control (Control PID)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid

figure(5)

```

```

plot(Tiempo , Integral)
title('Indice de integracion (Control PID)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid

```

```

figure(6)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PID)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid

```

Programa PID con seguimiento.

```

clc                % Borra pantalla
clear all          % Borra variables
close all          % Cierra ventanas (Gráficas)

load CPIDSeg.mat -ascii      % Lee archivo de datos
Tiempo = CPIDSeg(:,1);      % Tiempo
Lapsos = CPIDSeg(:,2);      % Lapsos
Vel = CPIDSeg(:,3);         % Velocidad
Control = CPIDSeg(:,4);     % Control
Error = CPIDSeg(:,5);       % Error
EP = CPIDSeg(:,6);          % Derivada del Error
Muestras = CPIDSeg(:,7);    % Muestras
Tref = CPIDSeg(:,8);        % Teta de referencia
Integral = CPIDSeg(:,9);    % Integral del error

```

```

figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PID con seguimiento)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid

```

```

figure(2)
plot(Tiempo , EP)
title('Derivada del error (Control PID con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid

```

```

figure(3)
plot(Tiempo , Error)
title('Error de velocidad (Control PID con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid

```

```

figure(4)

```

```
plot(Tiempo , Control)
title('Señal de control (Control PID con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid
```

```
figure(5)
plot(Tiempo , Integral)
title('Indice de integracion (Control PID con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

```
figure(6)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PID con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

Programa PIDNL.

```
clc % Borra pantalla
clear all % Borra variables
close all % Cierra ventanas (Gráficas)

load CPIDNLP.mat -ascii % Lee archivo de datos
Tiempo = CPIDNLP(:,1); % Tiempo
Lapsos = CPIDNLP(:,2); % Lapsos
Vel = CPIDNLP(:,3); % Velocidad
Control = CPIDNLP(:,4); % Control
Error = CPIDNLP(:,5); % Error
EP = CPIDNLP(:,6); %Derivada del Error
Muestras = CPIDNLP(:,7); % Muestras
Tref = CPIDNLP(:,8); % Teta de referencia
Integral = CPIDNLP(:,9); % Integral del error
```

```
figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PIDNL)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid
```

```
figure(2)
plot(Tiempo , EP)
title('Derivada del error (Control PIDNL)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid
```

```
figure(3)
```

```
plot(Tiempo , Error)
title('Error de velocidad (Control PIDNL)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid
```

```
figure(4)
plot(Tiempo , Control)
title('Señal de control (Control PIDNL)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid
```

```
figure(5)
plot(Tiempo , Integral)
title('Indice de integracion (Control PIDNL)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

```
figure(6)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PIDNL)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

Programa PIDNL con seguimiento.

```
clc                % Borra pantalla
clear all         % Borra variables
close all        % Cierra ventanas (Gráficas)

load CPIDNLSeg2.mat -ascii      % Lee archivo de datos
Tiempo    = CPIDNLSeg2(:,1);    % Tiempo
Lapsos    = CPIDNLSeg2(:,2);    % Lapsos
Vel       = CPIDNLSeg2(:,3);    % Velocidad
Control   = CPIDNLSeg2(:,4);    % Control
Error     = CPIDNLSeg2(:,5);    % Error
EP        = CPIDNLSeg2(:,6);    % Derivada del Error
Muestras  = CPIDNLSeg2(:,7);    % Muestras
Tref      = CPIDNLSeg2(:,8);    % Teta de referencia
Integral  = CPIDNLSeg2(:,9);    % Integral del error
```

```
figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PIDNL con seguimiento)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid
```

```
figure(2)
```

```

plot(Tiempo , EP)
title('Derivada del error (Control PIDNL con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid

```

```

figure(3)
plot(Tiempo , Error)
title('Error de velocidad (Control PIDNL con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid

```

```

figure(4)
plot(Tiempo , Control)
title('Señal de control (Control PIDNL con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid

```

```

figure(5)
plot(Tiempo , Integral)
title('Indice de integracion (Control PIDNL con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid

```

```

figure(6)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PIDNL con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid

```

Programa PX.

```

clc                % Borra pantalla
clear all         % Borra variables
close all         % Cierra ventanas (Gráficas)

load CPX.mat -ascii      % Lee archivo de datos
Tiempo  = CPX(:,1);      % Tiempo
Lapsos  = CPX(:,2);      % Lapsos
Vel     = CPX(:,3);      % Velocidad
Control = CPX(:,4);      % Control
Error   = CPX(:,5);      % Error
EP      = CPX(:,6);      % Derivada del error
Muestras = CPX(:,7);    % Muestras
n=length(Tiempo);
for i=1:n
    Tref(i)=135;
end

```

```
figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PX)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid
```

```
figure(2)
plot(Tiempo , EP)
title('Derivada del error (Control PX)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid
```

```
figure(3)
plot(Tiempo , Error)
title('Error de velocidad (Control PX)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid
```

```
figure(4)
plot(Tiempo , Control)
title('Señal de control (Control PX)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid
```

```
figure(5)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PX)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid
```

Programa PX con seguimiento.

```
clc                % Borra pantalla
clear all          % Borra variables
close all           % Cierra ventanas (Gráficas)

load CPXSeg.mat -ascii % Lee archivo de datos
Tiempo = CPXSeg(:,1); % Tiempo
Lapsos = CPXSeg(:,2); % Lapsos
Vel = CPXSeg(:,3); % Velocidad
Control = CPXSeg(:,4); % Control
Error = CPXSeg(:,5); % Error
EP = CPXSeg(:,6); % Derivada del Error
Muestras = CPXSeg(:,7); % Muestras
Tref = CPXSeg(:,8); % Teta de referencia
```

```

figure(1)
plot(Muestras , Lapsos)
title('Retardos en el ciclo de control (Control PX con seguimiento)');
xlabel('Muestras');
ylabel('Tiempo de rutina (ms)');
grid

```

```

figure(2)
plot(Tiempo , EP)
title('Derivada del error (Control PX con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Derivada del error');
grid

```

```

figure(3)
plot(Tiempo , Error)
title('Error de velocidad (Control PX con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Error');
grid

```

```

figure(4)
plot(Tiempo , Control)
title('Señal de control (Control PX con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Control');
grid

```

```

figure(5)
plot(Tiempo , Vel,'r',Tiempo , Tref,'b')
title('Velocidad (Control PX con seguimiento)');
xlabel('Tiempo (ms)');
ylabel('Velocidad');
grid

```