



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Instrumentación y Control Automático

Control Digital de Servomotor sin Escobillas de Corriente Directa

TESIS

Que como parte de los requisitos para obtener el grado de

Maestro en Ciencias

Presenta:

Guillermo Ronquillo Lomeli

Dirigido por:

Dr. Gilberto Herrera Ruiz

SINODALES

Dr. Gilberto Herrera Ruiz
Presidente


Dr. Vladimir Rauch Sitar
Secretario

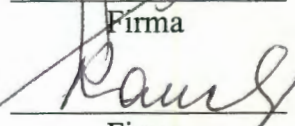
M. en I. René Romero Troncoso
Vocal

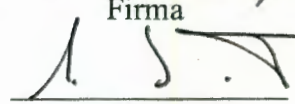
Dr. Alberto Aguado Behar
Suplente

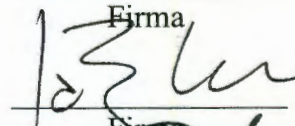
M. en I. Sergio Cervantes Pérez
Suplente

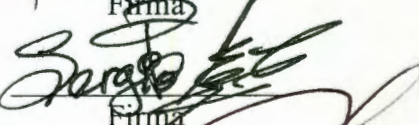
Ing. Jorge Martínez Carrillo
Director de la Facultad



Firma


Firma


Firma


Firma


Firma


Firma

Dr. Sergio Quesada Aldana
Director de Investigación y
Posgrado

Centro Universitario
Querétaro, Qro.
Mayo 2002
México

No. Adq. H 67177

No. Título _____

Clas. 629.8

R773c

Resumen

Dentro de la industria en general cada día que transcurre se tiende a un grado de automatización mayor, para lo cual es necesario importar tecnología, debido a que en México no existen desarrollos tecnológicos capaces de competir con tecnología extranjera. Debido a esto se presenta el trabajo de tesis cuyo objetivo principal es desarrollar sistemas completos y capacitar recursos humanos para competir con tecnología extranjera en servomecanismos.

Dentro de dicha automatización es necesario el control de movimientos en donde intervienen variables como aceleración, velocidad, posición, y torque los cuales pueden ser controlados por un motor, pero un motor común no es capaz por si solo de controlar con precisión estas variables, requiere de un sistema electrónico para que conjuntamente con un motor especial puedan ser controladas dichas variables.

La investigación se realizó en torno a un servomotor de corriente directa de 140 oz./in. sin escobillas con sensores de efecto de campo, con una velocidad máxima de 3200 R.P.M., comercial y se diseñó un sistema electrónico denominado servocontrol.

El sistema es capaz de comunicarse con la PC, vía puerto serial; y tiene acceso a parámetros como aceleración, velocidad y posición, tanto en observación como en orden. Posee, además, un controlador PID digital para compensar el sistema y un algoritmo de generación de trayectoria. El sistema opera en modo posición. Además se diseñó una etapa de potencia con un puente H de tres fases de transistores bipolares de compuerta aislada (IGBT's), un circuito combinacional para la conmutación de la corriente y un circuito controlador de IGBT's de alta tensión con lógica integrada para de-saturación, lógica de falla a tierra, apagado suave en condición de falla de sobre corriente, encendido y apagado independientes y retardos igualados entre fases; el cual es la interfaz entre las señales de control y el motor.

Para el desarrollo del sistema se empleó tecnología de punta como lo son módulos de IGBT's, Dispositivos Lógicos Programables (PLD's), Modulación de Ancho de Pulso (PWM), sensores de efecto de campo, microcontrolador de 16 Mhz, todos existentes en el mercado nacional, que nos dan la posibilidad de ser competitivos con los ya existentes.

Este servosistema puede ser aplicado en cualquier sistema que utilice movimiento con motores, que demanden alta precisión en velocidad y/o posición. En primera instancia se pretende aplicarlo en reconversión de máquinas-herramientas las cuales requieren movimientos de alta precisión, y es posible debido a su versatilidad y compatibilidad con sistemas digitales.

(Palabras clave: Servocontrol, PWM, PID, PLD).

Summary

Everyday industry requirements move towards an automation increase with the consequent demand for foreign goods because there is a lack of high-technology products in the national market. Due to this, the present thesis work object is to develop complete servomechanism systems that are competitive and to provide human resources development, in this country.

Automation requires control, which involves acceleration, speed, position and torque as related variables. In order to control these variables, an electronic system which acts as a feedback loop applied to the motor under control is required.

The present development was completed for a 1400 oz / in, dc brushless servomotor with field effect sensors and 3200 rpm as its maximum operating speed. The motor is controlled by an electronic servo controller.

The system is supervised by a PC via serial port to send and retrieve parameters such as acceleration, speed and position in both operation monitor and command mode. It includes a PID digital controller, to compensate the system and a trajectory generation algorithm. The system operates in position control mode. A power stage was also designed with a three-phase Isolated Gate Bipolar (IGBT) H bridge. The power interface stage is constituted by a combinational circuit for current commutation, a high-voltage IGBT controller circuit for de-saturation, a short circuit to ground, a soft turn off in over-current conditions, and an independent phase commutation with related delays.

The system development involves IGBT's modules, Programmable Logic Device (PLD), field effect sensors and a 16 Mhz microcontroller, all of them available in the national market to make them competitive with their counterparts. The system is very well suited in high-precision motion control applications especially where motor speed and/or position control is required. The system is especially devoted to automation and machine reconversion, where particular conditions like precision movement are required. It is possible due to its versatility and compatibility with digital system.

(Key words: Servocontrol, PWM, PID, PLD).

Dedicatorias

·
|
·

·
·

A mi familia por su incondicional apoyo

·
·

·
·

Agradecimientos

Al Ing. Ángel Ramírez Vázquez, director del Centro de Ingeniería y Desarrollo Industrial por su confianza y apoyo para la realización de este posgrado.

A los profesores de la facultad de Ingeniería División de Posgrado por sus enseñanzas, en especial al Dr. Gilberto Herrera Ruiz por haber dirigido la tesis y por su confianza.

Al M. en I. René Romero Troncoso por haber revisado la tesis y por sus atinados comentarios para mejorarla.

Índice

	Página
Resumen.....	i
Summary.....	ii
Dedicatorias.....	iii
Agradecimientos	iv
Índice.....	v
Índice de tablas	vi
Índice de figuras.....	vii
1. INTRODUCCIÓN.....	1
2. ESTADO DEL ARTE.....	6
3. COMPONENTES DEL SISTEMA.....	8
3.1. Introducción.....	8
3.2. Servosistema típico.....	8
3.3. Modelo del sistema.....	12
3.4. Análisis del sistema.....	19
3.5. Diseño del control.....	23
3.6. Análisis del servosistema en tiempo discreto.....	28
4. ELABORACIÓN DEL SISTEMA.....	36
4.1. Introducción.....	36
4.2. Funcionamiento del motor.....	37
4.3. El servoamplificador	41
4.4. Secuencia de conmutación de tensión.....	43
4.5. Controlador para puente de tres fases.....	45
4.6. Adquisición de velocidad y posición.....	52
4.7. Regeneración y descarga de potencia.....	54
4.8. Generación de trayectoria.....	55
4.9. El compensador	59
4.10. Instrucciones de interfaz.....	60
5. RESULTADOS.....	65
6. CONCLUSIONES.....	72
7. REFERENCIAS.....	73
8. APÉNDICE.....	74

Índice de tablas

	Página
Tabla 1.1: Evolución de los servosistemas.....	3
Tabla 3.1: Parámetros de un motor ejemplo.....	11
Tabla 3.2: Parámetros del servomotor a controlar.	21
Tabla 4.1: Potenciales causas de sobre corriente.	46
Tabla 4.2: Comparación de las soluciones del manejo de la compuerta protecciones del IGBT.....	49
Tabla 4.3: Parámetros.....	61
Tabla 4.4: Bits del registro del estado externo.	63
Tabla 4.5: Bits del registro del estado de movimientos.	63
Tabla 4.6: Variables capturadas por la instrucción c.....	64

Índice de figuras

	Página
Figura 3.1. Elementos fundamentales del control de movimiento.	9
Figura 3.2(a): Perfil de la velocidad deseada y (b): Posición correspondiente.	10
Figura 3.3: Diagrama a bloques que representa el modelo del sistema.	16
Figura 3.4: Diagrama a bloques del sistema de control de posición utilizando amplificador modo tensión.	18
Figura 3.5: Diagrama a bloques del sistema bajo diseño.	23
Figura 3.6: Diagrama a bloques del sistema a diseñar.	26
Figura 3.7: Sistema de control digital.	33
Figura 4.1: Diagrama a bloques del servocontrol.	36
Figura 4.2: Localización del sensor de efecto de campo.	38
Figura 4.3: Servomotor de tres fases con sensores de efecto de campo.	39
Figura 4.4: Secuencia de conmutación de las bobinas del motor.	40
Figura 4.5: Diagrama a bloques de la etapa de potencia o amplificador.	42
Figura 4.6: Tabla de verdad y funciones lógicas para la secuencia de conmutación.	44
Figura 4.7: Flujo de corriente durante un corto circuito de línea a línea.	50
Figura 4.8: Flujo de corriente durante una falla a tierra.	50
Figura 4.9: Flujo de corriente durante la saturación de dos transistores en serie. ...	51
Figura 4.10: Protección de sobre-corriente en un sistema de tierra flotante.	51
Figura 4.11: Protección contra sobre-corriente usando el IR2137.	51
Figura 4.12: Diagrama a bloques del IR2137.	52
Figura 4.13: Señales del encoder secuencial.	53
Figura 4.14: Sincronizador y decodificador del encoder secuencial.	54
Figura 4.15: Perfil de movimiento.	55
Figura 4.16: Perfil de velocidad para movimientos trapezoidales.	57
Figura 4.17: Perfil de velocidad para movimientos triangulares.	57
Figura 4.18: Elaboración del PID digital.	59

1. INTRODUCCIÓN.

Es evidente que el hombre de empresa dedica gran parte de su tiempo y de su esfuerzo a la obtención de una producción cada vez mayor para disminuir costos y de mejor calidad para incrementar sus utilidades.

Entre los diversos factores que inciden en ello se cuenta al diseño industrial, que puede considerarse como el resultado de utilizar los conocimientos de ingeniería en la satisfacción de las necesidades productivas. El diseño industrial puede orientarse hacia la ingeniería del producto, hacia la producción de bienes de consumo, hacia la dotación de mejores facilidades de manufactura, así como a la generación de bienes de capital, forma parte de las actividades que lleva a cabo el ingeniero en la realización de un proyecto, tales como construcción, instalación, operación y mantenimiento de equipo, producción y venta de los bienes de consumo, así como la investigación constante para mejorarla.

En México el diseño industrial como actividades de los ingenieros, representa hoy en día un escaso porcentaje respecto al resto de ellas, y su aparición data de hace poco tiempo.

En sí la ingeniería en México es reciente, dado que hace menos de un siglo empieza a surgir cuando se pretendió modificar la estructura socio-económica de nuestro país, que estaba incorporada al sistema de comercio mundial como proveedor de materias primas e importador de artículos manufacturados y bienes de capital, con industria eminentemente artesanal.

Sólo en algunas actividades como la minería, el transporte y la infraestructura urbana, se contaba con las tecnologías de la época, gran parte de éstas a través de empresarios extranjeros, encontrando en ellas su campo de acción los primeros mexicanos pero como recursos humanos en mantenimiento. Se inició entonces la pretensión de orientar la producción en nuestro país a los bienes de consumo que se importaban, etapa en la cual el ingeniero abarcó las actividades de instalación y operación de la maquinaria que se hizo necesaria, sin embargo, persistió la fuga de divisas a causa de la importación de maquinaria adquirida para este efecto, dado el alto precio de la transferencia de tecnología que representaba la misma, su equipo, insumos y contratos de licencia para fabricación.

A pesar de ello la manufactura de bienes de capital y aún la de bienes intermedios y bienes de consumo, se efectúa todavía sobre la base de diseños extranjeros cuyo alto precio de uso, así como el bajo nivel de competitividad en la producción, se reflejan en un mayor costo respecto del que se consiguen en el exterior, sin olvidar que gran parte de esa manufactura en el país la realizan empresas transnacionales con los inconvenientes de fuga de divisas.

Actualmente la actividad del diseño industrial en el ejercicio de la ingeniería en México, se concreta a algunos detalles de carácter general como adaptaciones e innovaciones de productos y procesos, pero existe ya un amplio potencial en recursos humanos con el conocimiento teórico y la experiencia necesaria para impulsar la generación de bienes de capital nacionales, que son uno de los pilares en el desarrollo tecnológico de cada país. El campo de los bienes de capital y sus componentes es muy amplio, ya que entre ellos se encuentran, equipos para construcción, calderas industriales, intercambiadores de calor, columnas de destilación, condensadores, tanques de presión, maquinaria para fabricación de cemento, fundiciones, empaques, molinos de harina, grúas y componentes de acero pesado, bombas hidráulicas, compresores, engranajes, cajas reductoras de velocidades, equipo para el manejo y transporte de materiales, motores de combustión interna y sus componentes, turbinas de gas y de vapor, máquinas para la industria alimenticia, del vestido y del calzado, transformadores y motores eléctricos, interruptores y conductores, hornos, máquinas-herramienta, sistemas electrónicos, etc.

De la amplia variedad señalada de productos, tradicionalmente se ha dado por los países desarrollados una gran atención precisamente al desarrollo de sistemas de control de movimiento ya que son la base de máquinas-herramienta tanto universales como de uso específico, ya que como es sabido son máquinas para producir más máquinas existiendo en las plantas nacionales del país un buen número de ellas las cuales tienen un control obsoleto y que sólo con tecnología extranjera es posible automatizarlas, además de que es muy costoso y ahuyenta cualquier inversión posible en México, por lo tanto es indispensable desarrollar nuestros propios sistemas de control de movimiento como lo han estado haciendo países orientales aunque de dudosa calidad, ellos se han atrevido a generar sus diseños propios, producirlos y comercializarlos a gran escala, incorporándoles solamente una pequeña parte de originalidad sobre todo de forma más no de fondo.

Por otro lado en los años recientes se ha venido incrementando el uso de los controladores digitales en sistemas de control. Los controladores digitales se usan para alcanzar un desempeño óptimo en la forma de productividad máxima, beneficio máximo, costo mínimo o la utilización mínima de energía. Recientemente, la aplicación de control por computadora ha hecho posible el movimiento inteligente en robots industriales, la optimización del combustible de los automóviles y el refinamiento en la operación de enceres y máquinas de uso doméstico, tales como los hornos de microondas y máquinas de cocer, entre otras. La capacidad en la toma de decisiones y la flexibilidad en los programas de control son las mayores ventajas en los sistemas de control digital. La tendencia actual de controlar los sistemas dinámicos en forma digital en lugar de analógica, se debe principalmente a la disponibilidad de computadores digitales de bajo costo y a las ventajas de trabajar con señales digitales en lugar de señales en tiempo continuo.

Como lo menciona (Younkin G. W. 1996), la industria de los servomecanismos tiene su inicio durante la segunda guerra mundial. Algunas aplicaciones fueron en el control de cañones militares y en el control del timón de barcos. En la tabla 1.1 se presenta la evolución de los servomecanismos a partir de la guerra.

Velocidad	Motores de corriente alterna		Servoamplificadores con SCR's Amplificadores de AC de frecuencia variable		Amplificadores para controlar velocidad en motores de AC
	Amplificadores para Motor Generador.	Servoválvulas Controladores de motores y pistones	Amplificadores con SCR's y PWM	Amplificadores para motores de corriente directa sin escobillas rotatorias.	Amplificadores para motores de corriente directa sin escobillas lineales.
	Eléctrico de corriente directa	Hidráulico	Eléctrico de corriente directa	Eléctrico de corriente alterna	
	1940-1960	1960-1970	1970-1980	1970-1980	1980-

Tabla 1.1: Evolución de los servosistemas.

El thyatron fue una forma del un conmutador eléctrico que no fue muy confiable tuvo notoriamente largos tiempos muertos o retrasos. El rectificador controlado de silicón (SCR) es la versión en estado sólido del thyatron. El SCR es muy confiable y es disponible para grandes intervalos de corriente. Éste ha sido usado extensamente en la industria de los servomecanismos. Igual que el thyatron, el SCR tiene tiempos muertos en los flujos de corriente. Estos tiempos muertos son descritos como un retraso causado en la fase del amplificador y limitando a los amplificadores de corriente directa (DC) a una respuesta lenta. El máximo intervalo de torque es del 400% de su intervalo normal, el cual puede ser usado para la aceleración.

Para muy altas potencias de servomecanismos, por ejemplo los motores de CD fueron por un tiempo manejados por generadores excitados por separado. Estos amplificadores fueron reemplazados por amplificadores con SCR. Alternamente los motores de frecuencia ajustable son usados en aplicaciones industriales donde el posicionamiento no es un requerimiento. Un ejemplo seria un ventilador de alta potencia. Los servomecanismos de AC de dos fases han sido usados en potencias fraccionales. Para aplicaciones con servomotores de alta potencia, los motores de inducción con control vectorial están siendo usados para control de velocidad y posición, con servoamplificadores de IGBT's se pueden encontrar hasta de 500 HP.

Para algunas potencias bajas de servoposicionadores los motores de corriente directa sin escobillas (motor síncrono de AC) han sido extensamente usados. Estos servomotores están disponibles en potencias menores de 30 caballos de potencia (HP). Algunos de estos servomotores son usados para posicionamiento en máquinas industriales. Estos motores aunque son de AC son tratados como motores de DC para aplicaciones. Este servomotor usa transistores en su servoamplificador lo cual da resultado un servoamplificador de alto rendimiento. La desventaja es que el intervalo de torque que maneja es limitado por los intervalos de los transistores en el amplificador. Estos amplificadores tienen un intervalo máximo del 200% del torque continuo del motor, los servomotores tienen un torque máximo del 400% pero éste es limitado a un 200% por el intervalo de corriente de los transistores. Este intervalo ha sido incrementado con la introducción de los transistores bipolares de compuerta aislada (IGBT).

Considerando lo anterior se concibió el proyecto que aquí se presenta y que pretende dar origen al dominio de una tecnología que permita la producción nacional de sistemas de control de movimientos los cuales podrán ser integrados a sistemas de control numérico computarizado, mediante la integración de resultados parciales de la infraestructura disponible en empresas de la región, tales como fabricación de circuitos impresos y otros componentes, así como diversos elementos y refacciones disponibles en el mercado nacional en cuanto a la producción se refiere, y por otra parte aprovechar la experiencia que se tiene en el diseño y realización de circuitos electrónicos para control, además de como la experiencia para desarrollar programas de uso general y específico.

Dado que el problema fundamental de un servocontrolador no es la elaboración de programas en nuestro entorno, sino la velocidad de procesamiento de la información que optimiza la aplicación de dichos programas, y dado también el grado de evolución de los microcontroladores, módulos de potencia y la disponibilidad de componentes electrónicos de tecnología de punta en el mercado local, abre la oportunidad de incursionar en este tipo de sistemas cuyo menor grado inicial de evolución en programación puede ser compensado por la alta velocidad de procesamiento de la información, y dado que se cuenta con la experiencia y conocimientos suficientes de la rama y con las herramientas de trabajo necesarias hacen totalmente factible el éxito del servocontrolador.

2. ESTADO DEL ARTE.

Los servosistemas aunque aparecieron recientemente se usan de forma generalizada en la industria de vanguardia es por eso que los servosistemas disponibles en el mercado internacional han alcanzado un grado de desarrollo muy notorio y evidente que en un principio usaban motores comunes y corrientes con escasa capacidad de trabajo y velocidad pero de gran consumo de energía y componentes dado que utilizaban tecnología obsoleta.

Al evolucionar la ingeniería electrónica y computacional, se obtuvieron nuevas herramientas para el desarrollo de nuevos sistemas de control de movimiento, totalmente amigables a sistemas de control digital; teniéndose a la fecha sistemas que no sólo controlan uno sino varios motores; ello da lugar a que con pequeñas modificaciones de programación, los servosistemas pueden ser adaptados a cualquier aplicación de control de movimientos. Esto puede dar origen a la integración de un sistema de control numérico, con componentes originados por mexicanos y competitivos con algunos sistemas existentes en el mercado.

Actualmente en la pequeña industria se usan sistemas mecánicos muy precarios como lo son embragues y frenos electro mecánicos cambio de relación de poleas manual, para el control de velocidad y posición en sistemas que así lo requieren, pero con grandes problemas de mantenimiento, precisión y repetibilidad. Aunque en el mercado existen una gran diversidad de servomecanismos el empresario no se decide al cambio debido a los grandes costos de inversión inicial y al desconocimiento de estos sistemas, sabedores de ésta problemática se desarrollo este proyecto de investigación con el objetivo de abatir costos, con el diseño y construcción de servomecanismos a la medida de las aplicaciones, generar investigación totalmente documentada en servosistemas, para capacitar recursos humanos que puedan generar sus propias aplicaciones.

Adicionalmente, cientos de universidades ofrecen cursos de teoría de control. Pero esto sólo prepara al ingeniero con los fundamentos académicos necesarios para incursionar en la industria del servocontrol. Sin embargo, siempre hay dificultades en la transición

entre la teoría y la práctica en el mundo real. Este trabajo es un ejemplo de cómo la teoría del control puede ser aplicado en problemas prácticos de control.

Se ha invertido un año que persiguen estos objetivos. Las computadoras baratas y el alto grado de componentes de control de movimiento, que fluyen fuera de la tecnología de corporaciones como la construcción de herramientas dedicadas para la robótica, se usaron en este estudio. Los costos de construir tales sistemas son bajo en términos de dólares pero las demandas de tiempo pueden ser muy grandes.

Se espera que este trabajo sirva no solamente como una vitrina del trabajo; que otros investigadores, quienes estén interesados en el uso potencial de ésta técnica, puedan obtener información práctica.

Los beneficios esperados son en principio la aceptación a considerar los resultados de este proyecto como una opción más, confiando en que sus diversas características técnico-económicas la puedan hacer competitiva. También se cree que el cambio de mentalidad originado en las personas que apoyan este proyecto sea un detonante que impulse y dé confianza en la generación de equipo en los diversos campos de la ingeniería que hasta ahora han sido vírgenes para los profesionales mexicanos.

En la actualidad los servomecanismos son digitales y tienen un servocontrol avanzado que les permite alta precisión en sus movimientos, son compactos y de apariencia moderna que les ayuda a minimizar el ruido eléctrico, filtros pasa bajas para resonancia en máquinas; tres modos de operación modo torque, modo velocidad y modo posición; diferentes intervalos de corriente, diseño robusto, comunicación serial que permite la comunicación con una computadora personal (PC) o un controlador lógico programable (PLC); software en ambiente Windows de comunicación y programación, inicio de movimientos a través de comandos seriales o digitales; entradas y salidas digitales y analógicas. Éstas funciones son las que se pretenden alcanzar con este proyecto.

3. COMPONENTES DEL SISTEMA.

3.1. Introducción.

Un control automático compara el valor real de la salida de una planta con una entrada de referencia (el valor deseado), la desviación determina y produce una señal de control que reducirá la desviación a cero o a un valor muy pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control.

En este capítulo se analiza el control típico de los servosistemas que se usan en controles industriales, para esto, inicialmente se debe obtener el modelo del sistema para posteriormente analizarlo y finalmente obtener el diseño del control del servosistema. El método de análisis es el usado por (Jacob Such a Ph D, 1994), en su obra, Step-by-Step Design of motion control systems, apoyado en (Ogata K. 1998).

3.2. Servosistema típico.

El control de movimientos son tareas inteligentes de los sistemas y son como se presente en la figura 3.1. Para propósito de claridad, las tareas del controlador pueden ser divididas en dos, generación de trayectoria R , y el seguimiento de ésta. Las dos tareas son efectuadas independientemente y ambas son requeridas para la generación del movimiento.

Hay una comunicación del control de movimiento con la computadora que enviará las órdenes de movimiento. Esta comunicación es ejecutada por un puerto serie, vía RS232.

Los requerimientos de movimiento son siempre especificados por la computadora, en términos de pocos parámetros. Tales parámetros incluyen, por ejemplo, distancia, velocidad, intervalos de aceleración y desaceleración. Estos parámetros son procesados por el control de movimiento y construyen una trayectoria continua que es la posición de referencia R .

Considere por ejemplo, el perfil de velocidad ilustrado por la figura 3.2(a). El tiempo de movimiento de 150ms es dividido igualmente entre la aceleración, giro, y desaceleración. La velocidad de cambio es de 10,000 cuentas/s y el total desplazamiento es 10,000 cuentas. Tales parámetros son típicamente especificados por el usuario. En

respuesta a esto, el controlador de movimiento genera la función de referencia $R(t)$, mostrado en la figura 3.2(b), con la correspondiente posición en el tiempo.

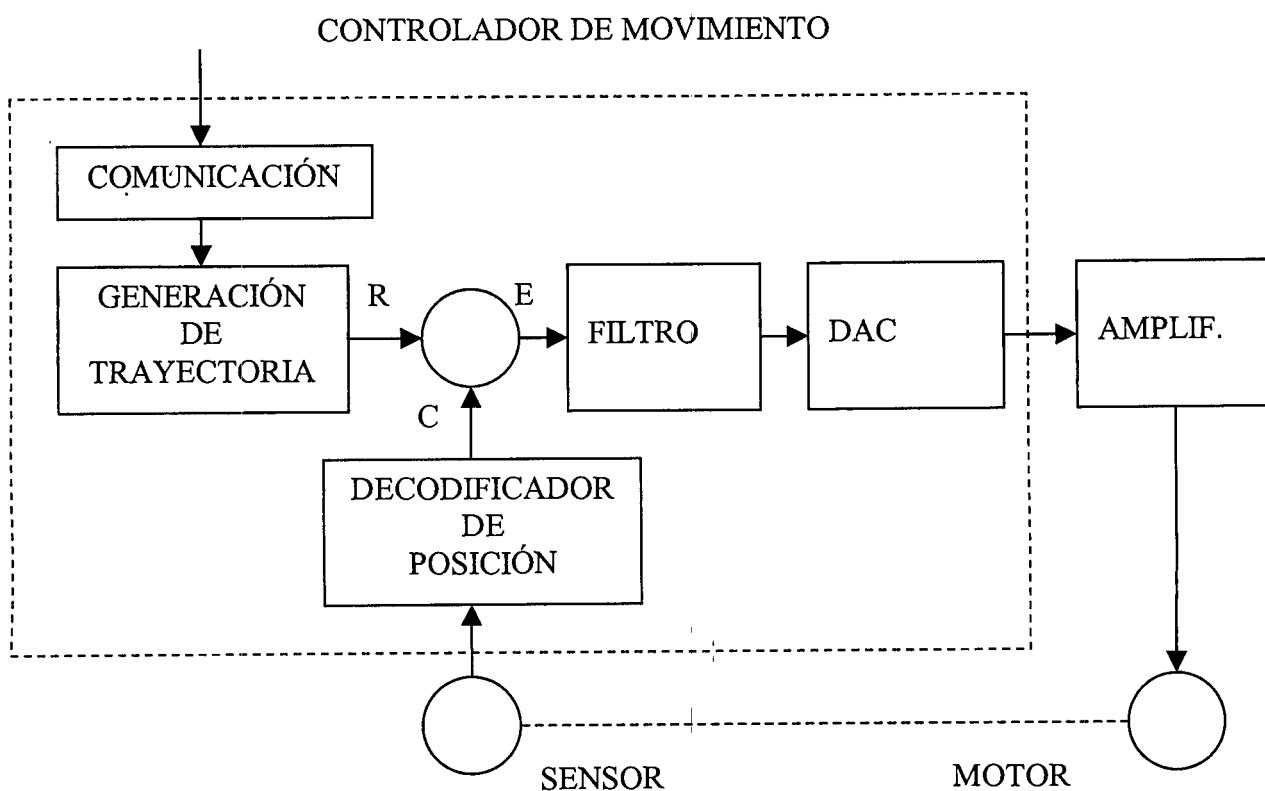


Figura 3.1. Elementos fundamentales del control de movimiento.

La segunda tarea es la de seguir la función de referencia. Comenzando con la decodificación de la señal de posición, C , y el cierre del lazo. La posición del motor C es determinada por la señal de retroalimentación (frecuentemente un encoder incremental) y comparado con la posición de referencia, R ; E , la diferencia $R - C$, es conocido como error de posición. El objetivo del controlador es reducir el valor del error de posición, E , a un mínimo sin causar oscilaciones en el sistema. Para lograrlo, el controlador frecuentemente incluye un filtro estabilizador cuya salida es aplicada al convertidor de digital a analógico (DAC) hacia el amplificador y el motor.

Para entender la operación del filtro, considere la siguiente situación. Suponga que el motor es requerido para ser operado a la posición de referencia $R=0$, y que debido a una perturbación el motor ha sido movido a la posición de $C=-100$. Se analiza el movimiento de cambio del motor.

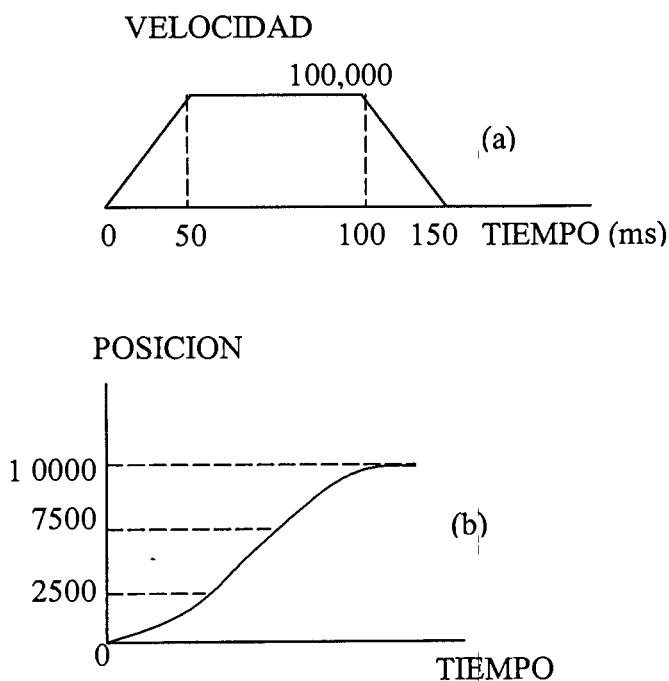


Figura 3.2(a): Perfil de la velocidad deseada y (b): Posición correspondiente.

Asume primero que la salida del controlador es directamente proporcional a la posición de error. Por lo cual esta salida forzar  la aceleraci3n del motor movi ndose entre las posiciones de -100 y cero. Como resultado, el motor alcanzara la posici3n cero a alta velocidad y puede haber sobre impulso. El proceso de sobre impulso continuar  sobre ambas posiciones, ocasionando oscilaciones en el sistema.

Una alternativa de aproximaci3n es incluir un filtro de compensaci3n, el cual se produce en dos t rminos, uno proporcional a la posici3n de error y uno proporcional a la derivada de  ste o la velocidad de cambio. Como el motor se mueve entre las posiciones menos cien y cero, el error de posici3n es positivo. En otro caso, como el error decrementa, esta derivada es negativa. La correcta mezcla de los dos t rminos asegura que durante la primera parte del movimiento la suma es positiva, dirigiendo la aceleraci3n. Durante la segunda parte del movimiento, la salida del controlador llegara ser negativa, resultando en desaceleraci3n como es requerido.

Para describir la operaci3n del sistema, considere un ejemplo con los siguientes par metros:

<u>Parámetros</u>	<u>Definición</u>
$K_t = 0.1 \text{ Nm/A}$	Torque constante del motor
$J_m = 10^{-4} \text{ Kgm}^2$	Momento de inercia
$r = 2 \Omega$	Resistencia del motor
$N = 500 \text{ líneas / rev.}$	Densidad de línea del Encoder
$K_a = 3 \text{ Amp. / voltios.}$	Ganancia del amplificador de corriente

Tabla 3.1: Parámetros de un motor ejemplo.

Ahora si se supone que el sistema requiere estar en la posición cero ($R = 0$), pero la posición actual del motor es -100 ($C = -100$). El controlador determina el error de posición determinado con la ecuación $E = R - C$, resultando en un error de cien cuentas. La señal de error, E , es entonces procesado por el filtro. Se asume, por simplicidad, que la ganancia combinada del filtro y el DAC es $0.02V$ por cuenta. Esto implica que una señal de error de cien cuentas produce una señal de movimiento de orden, X , de $2V$.

La señal X es aplicada al amplificador con una ganancia dada de tres amperes por voltio, resultando una corriente de seis amperes. Cuando la corriente es aplicada al motor, genera un torque proporcional de 0.6 Nm . El Torque positivo controla el motor hacia adelante, en dirección que reduce la posición de error.

La corriente aplicada produce un torque T_g , el cual es igual al producto de la corriente y el torque constante K_t . El torque generado acelera el motor a la velocidad α , donde

$$\alpha = \frac{T_g}{J_m}$$

Para los parámetros dados del sistema $\alpha = 6000 \text{ radianes/s}^2$

Como el motor se mueve más cerca a la posición deseada, el error E , se decrementa. Como el filtro controlador reduce el error de posición, con la derivada se produce un decremento de la señal, la cual llegara a ser negativa durante la segunda mitad

del movimiento, ocasionando la desaceleración requerida, una vez que el motor alcanza la posición de referencia; E llegará a ser cero, eliminando la corriente en el motor.

Claramente, el comportamiento dinámico del sistema del motor, depende de la correcta selección de los parámetros del filtro. El análisis del modelo del sistema expuesto y diseño es presentado en los siguientes capítulos.

3.3. Modelo del sistema.

En este punto se analiza el modelo matemático del sistema, lo cual es básico para el análisis y diseño del sistema.

Los sistemas de control de movimiento son únicos ya que incluyen sensores y elementos de diferentes tipos. El primero de ellos es el motor, un dispositivo que recibe una señal continua y opera continuamente en el tiempo. Por otro lado hay un controlador digital el cual discretiza la amplitud de las señales y opera en tiempos discretos. Un tercer tipo es el sensor de posición, el cual opera continuamente en el tiempo pero es discreto en amplitud, El cuarto tipo es el amplificador de potencia la cual produce una señal continua pero opera en tiempos discretos. Este cuarto tipo hace extremadamente difícil el desarrollo del modelo que indique el comportamiento del sistema.

Afortunadamente, sólo se necesita predecir el comportamiento del sistema sobre un intervalo de frecuencia limitado. Este intervalo de frecuencias es el que el control puede seguir, el cual es típicamente limitado a 100 Hz. La necesidad de limitar el intervalo de frecuencia simplifica el proceso de modelado, porque esto nos permite la aproximación al modelo exacto simplificado; el cual es bastante preciso sobre el intervalo de frecuencia requerido, pero puede llegar a ser inexacto a altas frecuencias. Considerando esto, se desarrolló el modelo que describe todos los elementos del sistema como si ellos tuvieran amplitudes continuas y operen en el tiempo continuamente. El resultado del modelo matemático requiere sólo del conocimiento del modelado en la transformada de Laplace para su análisis y diseño.

3.3.1. Modelo

Para el propósito de modelado se divide el sistema en cuatro elementos:

Amplificador

Motor y carga.

Sensor de posición

Controlador

El modelado de los elementos se discute a continuación:

3.3.2. Modelado del Amplificador motor.

Para iniciar el proceso de modelado, se necesita asumir que tipo de amplificador se usará. Se iniciará con el amplificador más común, el que opera en modo torque o corriente. En éste modo el amplificador tiene como salida una corriente, I , ésta es directamente proporcional la tensión de entrada, V . El factor proporcional es, K_a , conocida como la ganancia del amplificador y tiene unidades de ampere por voltio.

$$I = K_a V \quad (3-1)$$

Como la corriente es aplicada al motor, ésta genera un torque proporcional, T_g , con un factor proporcional K_t . La constante K_t es conocida como la constante de torque y sus unidades son torque por ampere.

$$T_g = K_t I \quad (3-2)$$

La generación de torque acelera el motor en un intervalo α , según la segunda ley de Newton. Si el momento total de inercia del motor y de la carga es J , y se define como la fricción opuesta como T_f . La ecuación dinámica es:

$$J\alpha + t_f = T_g \quad (3-3)$$

La velocidad del motor, w , es la integral de la aceleración α . Para simplificar el análisis, se aplica la transformada de Laplace, Obteniendo el siguiente modelo.

$$\omega = \frac{1}{s} \alpha \quad (3-4)$$

Con un manejo similar, la posición angular del motor, θ , es la integral de la velocidad. Por lo tanto, esto puede ser modelado como:

$$\theta = \frac{1}{s} \omega \quad (3-5)$$

Se pueden combinar ahora las ecuaciones (3-1) a la (3-5) para encontrar una función de transferencia, la cual relaciona el efecto de la entrada de tensión, V , en la posición del motor, θ , Este modelo represente el efecto combinado del motor y el amplificador.

Note que la posición del motor se encuentra afectado por dos entradas, la tensión, V , y la fricción T_f . Debido a la naturaleza lineal del sistema, se puede separar el efecto de las dos entradas. Por lo tanto, cuando se examina el efecto de V , se puede ignorar el efecto de T_f . El resultado combinado de la función de transferencias es:

$$\frac{\theta}{V} = \frac{K_a K_t}{Js^2} \quad (3-6)$$

La ecuación (3-6) está basada en la consideración de que el motor es rotatorio y que el amplificador opera en modo torque.

3.3.3. Modelando el sensor de posición

El sensor de posición representa la posición del motor como una señal C . Por eso, el sensor de posición puede ser modelado como un factor de proporcionalidad, K_f . El valor de K_f es igual al número de unidades de retroalimentación por un radian de rotación.

El desarrollo de un modelo específico, se hará con el sensor de posición más común, el encoder incremental. Se supone que el encoder tiene una resolución de N pulsos por revolución; implicando que las salidas del encoder, canal A y B, produce 1000 pulsos por cada giro del encoder. Como las dos señales están desviadas por un cuarto de ciclo en cuatro cuentas cuadradas, resulta una resolución efectiva de $4N$ cuentas por vuelta. Si cada vuelta es igual a 2π radianes, el resultado de la ganancia del encoder es:

$$K_f = \frac{4N}{2\pi} \quad (3-7)$$

Por ejemplo. A 1000 pulsos por revolución equivalen a 636 cuentas / radianes.

3.3.4. Modelo del controlador

Para propósito de modelado, se enfocará a las funciones del controlador que afectan la conducta dinámica. Estas funciones incluyen el lazo cerrado en posición, el filtro y el convertidor de la salida en una forma analógica.

El controlador determina siempre donde el motor debe estar. Esta posición deseada o ideal es denotada por $R(t)$ o simplemente R . Como el controlador adquiere la posición actual del motor, C , se comparan las dos y se determina la diferencia, la cual es denominada error de posición E .

$$E = R - C \quad (3-8)$$

Este error de posición es filtrado para generar una señal de salida para el amplificador del motor. El filtro puede ser dividido en partes. La primera parte es el término proporcional el cual produce el término X_p , que es proporcional a E .

$$X_p = PE \quad (3-9)$$

La segunda operación del filtro es producida por una señal de salida, X_d , la cual es proporcional a la derivada, esta parte es representada por la siguiente ecuación transformada.

$$X_d = sDE \quad (3-10)$$

La tercera parte es la salida de un integrador X_i , dada por.

$$X_i = \frac{I}{s} E \quad (3-11)$$

La señal de salida, X , es la suma de las tres componentes:

$$X = X_p + X_d + X_i \quad (3-12)$$

La función de transferencia $F(s)$, relacionada con el filtro de salida, X , y el error de posición es:

$$F(s) = \frac{X}{E} = P + sD + \frac{1}{s} \quad (3-13)$$

Este tipo de filtros es conocido como filtro PID y es el más común en sistemas de control de movimiento.

Note que el tipo de filtro dado aquí asume una operación continua. Esto es sólo un modelo equivalente del proceso digital.

La señal de salida, X , es convertida después a una forma analógica por un circuito conocido como convertidor analógico digital o simplemente DAC. Un DAC es caracterizado por su resolución, la cual varía típicamente entre ocho y dieciséis bits, y el intervalo de salida en voltios, la cual es típicamente entre -10 a 10 voltios. La ganancia efectiva del DAC, K es igual al número de voltios que éste produce por unidad de la señal X . Cuando la señal de salida varea entre -10 y 10 Voltios, y la resolución del DAC es de n bits, la ganancia del DAC es igual a:

$$K = \frac{20}{2^n} \quad (3-14)$$

El modelo matemático dado arriba, puede ser representado por el diagrama a bloques de la figura 3.3, el cual representa el sistema más común.

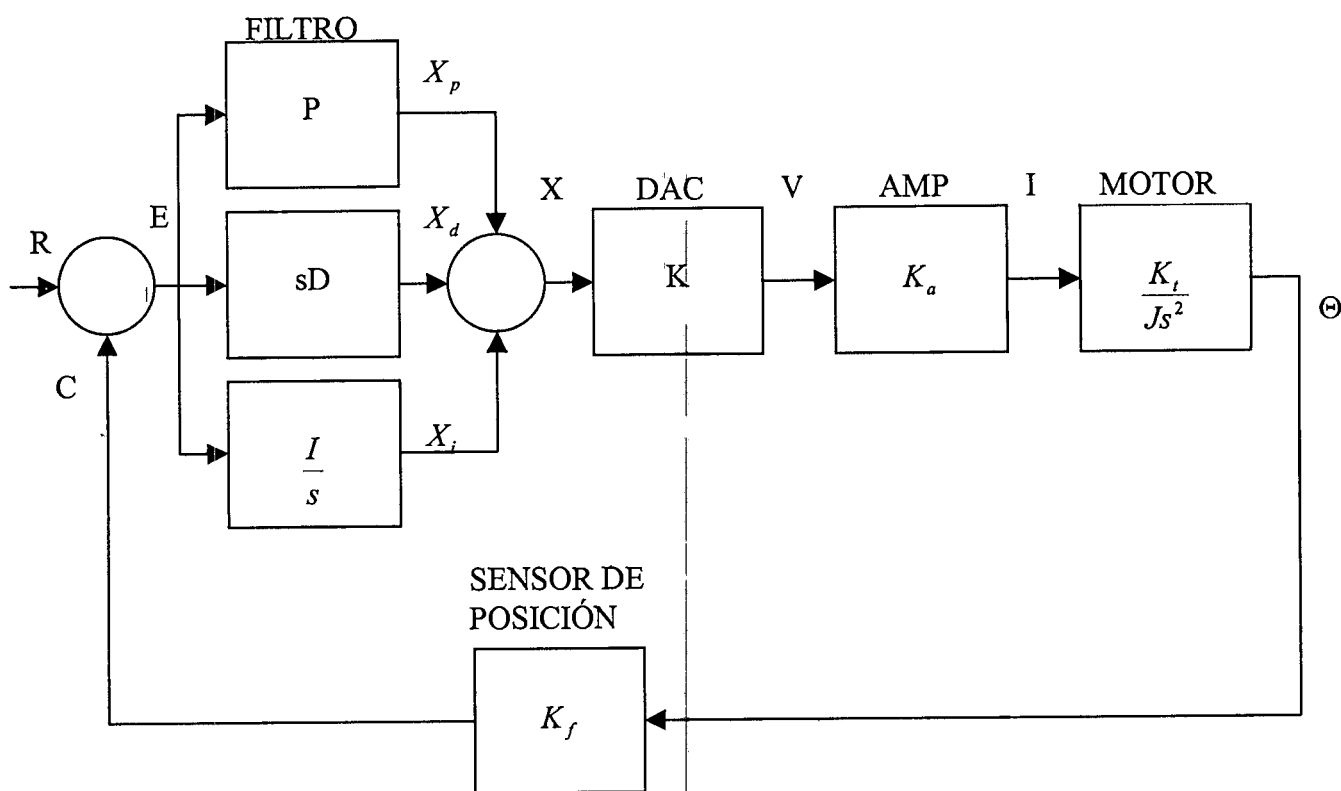


Figura 3.3: Diagrama a bloques que representa el modelo del sistema.

3.3.5. Sistemas con amplificador de tensión.

Ocasionalmente, y es nuestro caso, los amplificadores son diseñados para producir una salida proporcional de tensión V , en vez de corriente. En este caso el amplificador es modelado como un amplificador de tensión K_v .

$$U = K_v V \quad (3-15)$$

Cuando un potencial eléctrico U es aplicado al motor, éste produce una corriente I , la cual depende de la velocidad del motor ω . La ecuación del circuito del motor es:

$$U = rI + sLI + K_e \omega \quad (3-16)$$

Donde r es la resistencia de armadura del motor, L es la inductancia, y K_e es la constante de tensión. Note que la tensión en el motor incluye tres términos que determinan tres efectos físicos: el primero representa la tensión a través de la resistencia r ; el segundo representa la tensión a través de la inductancia L ; y la tercera, $K_e \omega$ es la tensión inducida por la fuerza contraelectromotriz del motor. Para producir una cierta corriente I , a una cierta velocidad, el amplificador debe alimentar una tensión U , de acuerdo con la ecuación (3-16).

La ecuación dinámica (3-3) puede ser representada en términos de ω como:

$$Js\omega + T_f = T_g \quad (3-17)$$

Note que la conducta dinámica del motor depende del modo de operación del amplificador, como es evidente por los diferentes modelos. Para desarrollar la función de transferencia del motor, se debe combinar la ecuación (3-2) con la ecuación (3-17) e ignorar la fricción T_f :

$$I = \frac{1}{K_t} sJ\omega \quad (3-18)$$

Ahora combinando las ecuaciones (3-18) y (3-16)

$$U = \frac{1}{K_t} (s^2 J\omega L + sJ\omega r + K_e K_t \omega) \quad (3-19)$$

lo cual lleva a

$$M(s) = \frac{\omega}{U} = \frac{K_t}{s^2 J L + S J r + K_e K_t} \quad (3-20)$$

esto puede ser escrito como

$$M(s) = \frac{1}{K_e (s T_m + 1)(s T_e + 1)} \quad (3-21)$$

Donde las dos constantes de tiempos, T_m y T_e

$$T_m = \frac{J r}{K_e K_t} \quad T_e = \frac{L}{r} \quad (3-22)$$

Son las constantes de tiempo y electromecánicas del motor. Los elementos del sistema están representados en diagrama a bloques del sistema de la figura 3.4.

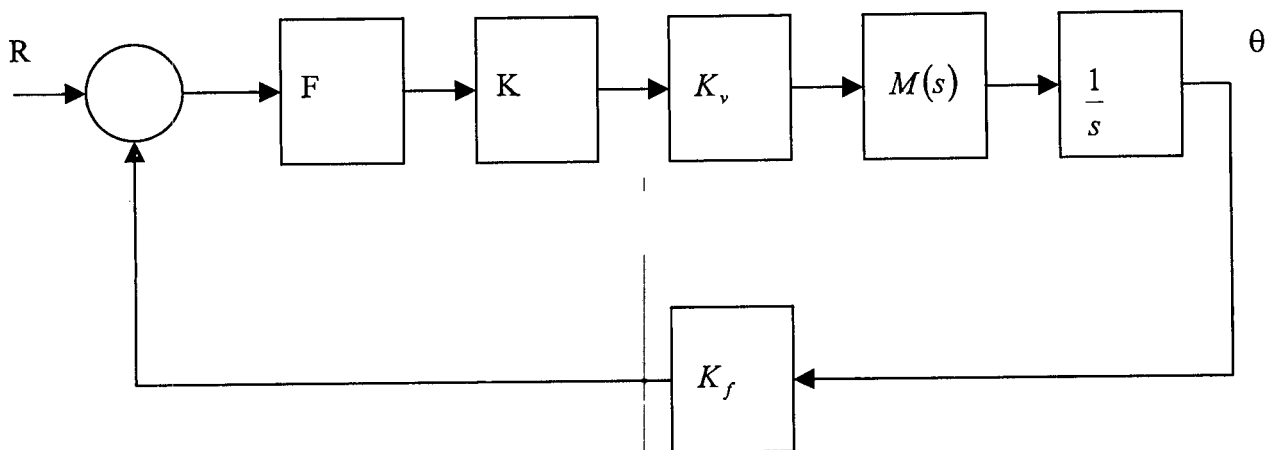


Figura 3.4: Diagrama a bloques del sistema de control de posición utilizando amplificador modo tensión.

Sobre todo la función de transferencia representa el efecto combinado del motor y el amplificador y es derivado de las ecuaciones (3-5), (3-15) y (3-21):

$$\frac{\theta}{V} = \frac{K_v}{K_e s (s T_m + 1)(s T_e + 1)} \quad (3-23)$$

Note que las ecuaciones (3-23) y (3-6) describe la operación del mismo motor con dos tipos diferentes como amplificadores.

3.4. Análisis del sistema

El análisis del sistema es un método para predecir la respuesta del sistema con base al modelo matemático. El principal objetivo del análisis del sistema es la respuesta de las siguientes dos cuestiones:

Es el sistema estable y si lo es, ¿cómo lo es?

Que tan rápida es la respuesta del sistema.

La estabilidad del sistema indica la respuesta natural. Como se mueve el motor de una posición a otra, esto puede ocasionar un sobre impulso en la tarjeta ocasionando tiempos críticos antes de estabilizarse. La importancia del sobre impulso está directamente relacionada con el grado de estabilidad, el cual es llamado amortiguamiento. Del más bajo amortiguamiento, al más largo sobre impulso. Un sistema inestable causa que el motor oscile y nunca se estabilice. El grado de estabilidad es casi siempre expresado en margen de fase; entre más grande sea el margen, mejor será la estabilidad.

La velocidad de respuesta indica que tanto le toma al sistema la ejecución de pequeñas correcciones que no cause saturación. En la mayoría de los casos, la respuesta del sistema está caracterizada por el parámetro de frecuencia de cruce, el cual está directamente relacionado con el tiempo de respuesta.

El análisis propuesto inicia con el modelo del sistema y deriva el margen de fase y la frecuencia de cruce. Esto se puede realizar con el siguiente procedimiento.

3.4.1. Procedimiento de análisis de sistema.

El procedimiento de análisis consiste de los siguiente pasos:

1. Iniciar con el modelo matemático del sistema y derivar $L(s)$, la función de transferencia en lazo abierto. En un sistema simple con un simple lazo de retroalimentación, $L(s)$ es el producto de todos los elementos en el lazo. En sistemas más complejos, derive la función de transferencia entre la entrada y la salida. Reordene los términos en el denominador para tenerla de la forma $1 + L(s)$. La función resultante, $L(s)$, es la función de transferencia en lazo abierto correspondiente.

2. Reemplace los parámetros s en $L(s)$ por los parámetros $j\omega$. La función resultante $L(j\omega)$ es llamada respuesta en frecuencia en lazo abierto.
3. Encuentre la frecuencia de cruce ω_c del sistema. La frecuencia de cruce es aquella en la cual la magnitud de $L(j\omega) = 1$.

$$|L(j\omega)| = 1 \quad \text{en} \quad \omega = \omega_c \quad (3-24)$$

La frecuencia de cruce indica la velocidad de respuesta, más alto valor de ω_c indica respuesta más rápida. En general, la constante de tiempo de la respuesta, τ es aproximadamente

$$\tau = \frac{1}{\omega_c} \quad (3-25)$$

4. Encontrar la fase de $L(j\omega_c)$. Dado que $L(j\omega)$ es una función compleja, se encontrará su argumento, y se llamará el argumento como Φ .

$$\Phi = \arg[L(j\omega_c)] \quad (3-26)$$

el margen de fase del sistema está dado por:

$$\Theta_m = 180 + \Phi \quad (3-27)$$

El parámetro Θ_m tiene la llave de la estabilidad y amortiguamiento del sistema.

El sistema es estable cuando el margen de fase Θ_m es positivo e inestable cuando Θ_m es negativo. Los valores del margen de fase en el intervalo de 30 a 45 grados indican una respuesta amortiguada; valores de margen de fase más pequeños indican respuesta de bajo amortiguamiento.

3.4.2. Sistemas de control con amplificador de tensión.

Cuando un amplificador es configurado como un alimentador de tensión, su modelo matemático cambia como ya fue discutido. A continuación se discute el efecto del amplificador de tensión en la estabilidad del lazo.

Se tiene un sistema de control de posición utilizando un amplificador de tensión. Tal como se ilustra en el sistema de la figura 3.4. El motor que se controlará tiene los parámetros que se muestran en la tabla 3.2.

PARÁMETRO	UNIDADES	DESCRIPCIÓN
$K = 0.0003052$	$V/cuentas$	Ganancia DAC
$K_v = 4$	V/V	Amplificador de tensión
$K_t = 0.24$	Nm/A	Constante de torque.
$K_e = 0.242$	$V/(rad/s)$	Constante de tensión.
$r = 2.58$	Ω	Resistencia de armadura
$J_m = 1.3 \times 10^{-4}$	Kgm^2	Momento de inercia
$L = 0.00106$	H	Inductancia.
$K_f = 636$	$Cuentas/rad$	Ganancia del encoder

Tabla 3.2: Parámetros del servomotor a controlar.

Se asume por adelantado que los parámetros del filtro son.

$$P = 100$$

$$D = 0.4$$

La función de transferencia resultante del filtro es

$$F(s) = 100 + 0.4s$$

El objetivo es determinar la frecuencia de cruce y el margen de fase. Primero se calculan las constantes de tiempo como están dadas en las ecuaciones (3-7) y (3-8)

$$T_m = \frac{rJ}{K_t^2}$$

$$T_e = \frac{L}{r}$$

Para los parámetros del sistema dados esto es igual

$$T_m = 0.0058s$$

$$T_e = 0.00041s$$

De acuerdo con la función de transferencia del motor dada por la ecuación (3-21) es

$$\frac{\omega}{V} = \frac{4.2}{(0.0058s + 1)(0.00041s + 1)}$$

La función de transferencia en lazo abierto

$$L(s) = \frac{FKK_v K_f}{K_e s (sT_m + 1)(sT_c + 1)} \quad (3-28)$$

por los parámetros dados, L(s) es igual a

$$L(s) = \frac{34400000(s + 250)}{s(s + 172)(s + 2434)} \quad (3-29)$$

Para analizar la estabilidad del sistema, se encontrará la frecuencia de cruce ω_c donde la magnitud de $L(s)$ es igual a uno.

$$|L(j\omega_c)| = 1 \quad (3-30)$$

Cuando la condición de la ecuación (3-30) es aplicada a la ecuación (3-29) resulta

$$\omega_c = 200 \text{ rad/s}$$

Lo siguiente es encontrar el margen de fase Φ , de la función de transferencia en lazo abierto en la frecuencia de cruce

$$\Phi = \text{Arg}[L(j\omega_c)]$$

Para este caso, Φ es igual

$$\Phi = \tan^{-1}\left(\frac{200}{250}\right) - 90 - \tan^{-1}\left(\frac{200}{172.4}\right) - \tan^{-1}\left(\frac{200}{2439}\right) = -139.5$$

Levando el margen de fase a

$$\Theta_m = 180 + \Phi = 40.5$$

El resultado del análisis indica que el sistema es estable. El margen de fase de 40.5 representa un buen grado de amortiguamiento y la frecuencia de cruce de 200 rad/s representa que la respuesta en el tiempo es de aproximadamente de 5 ms.

3.5. Diseño del control.

Se iniciará asumiendo que todos los elementos del sistema son fijos y que su modelo matemático es conocido. Como el amplificador, el motor y el sensor son considerados fijos en este punto, se definirá la función de transferencia combinada por $H(s)$. En el caso del motor manejado por amplificador modo corriente, $H(s)$ es

$$H(s) = \frac{K_a K_t K_f}{Js^2} \quad (3-31)$$

Los elementos del controlador que deben ser considerados ahora son el filtro, $F(s)$, y el DAC. El diagrama a bloque resultante se presente en la siguiente figura 3.5

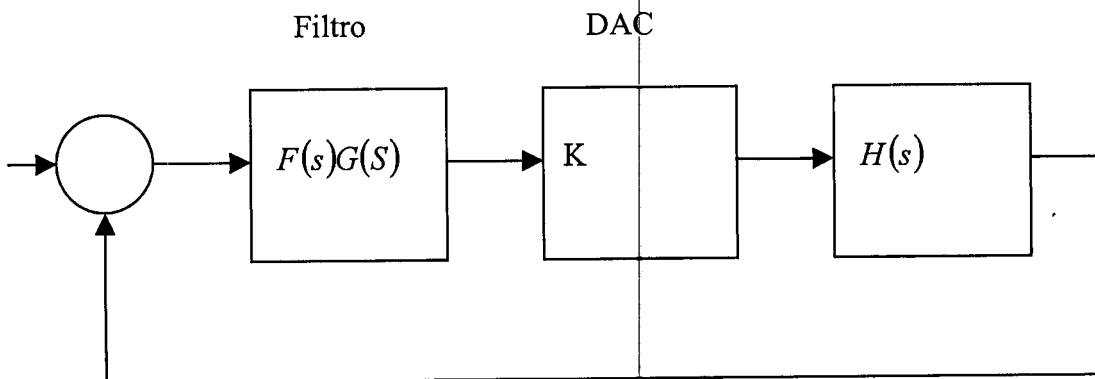


Figura 3.5: Diagrama a bloques del sistema bajo diseño.

El objetivo del diseño es seleccionar los parámetros del filtro de tal manera que produzca una respuesta deseada. La respuesta del servosistema está caracterizada por la

velocidad de respuesta y el nivel de estabilidad. La velocidad de respuesta está determinada por la frecuencia de cruce, ω_c . Ésta es la frecuencia en la cual la magnitud de la función de transferencia en lazo abierto es uno. El nivel de estabilidad está determinado por el margen de fase, Θ_m . Donde si Θ_m está entre 30 y 45 grados, la respuesta del sistema es amortiguada y estable. El margen de fase más pequeño indica respuesta de bajo amortiguamiento, y por lo tanto, grandes sobre impulsos. Pero si Θ_m es mayor de 45 grados, la respuesta puede ser sobre amortiguada y lenta.

Se asumirá que el filtro es de tipo PID con coeficientes P, I y D.

$$F(s) = P + sD + \frac{I}{s} \quad (3-32)$$

Para un diseño preciso, se considerará la naturaleza digital del controlador. Cuando el controlador opera con un periodo de muestreo T, el efecto del muestreo en el sistema dinámico es aproximado a la de un retardo de medio periodo de muestreo. Esto está expresado por G(s)

$$G(s) = e^{\frac{-sT}{2}} \quad (3-33)$$

Finalmente, el DAC es representado como una ganancia K. Resultando una función de transferencia en lazo abierto L(s)

$$L(s) = F(s)G(s)KH(s) \quad (3-34)$$

Ahora se asume que el objetivo del diseño es expresado en términos de la frecuencia de cruce ω_c , y el margen de fase Θ_m . Esto implica que

$$|L(j\omega_c)| = 1 \quad (3-35)$$

y que

$$\Theta_m = 180 + \text{Arg}[L(j\omega_c)] \quad (3-36)$$

El proceso del diseño del sistema incluye la especificación de los parámetros Θ_m , y ω_c y seleccionar los parámetros del filtro que satisfagan las ecuaciones (3-35) y (3-36).

Especificar el margen de fase es relativamente fácil. La constante de 45 grados es un parámetro bueno para todo diseño. La frecuencia de cruce, como quiere que sea, es más complicada. Idealmente, es el valor más grande posible, porque esto lleva a una respuesta más rápida. La realidad es que el modelo matemático del motor y la carga son sólo una aproximación. Esta aproximación es buena para bajas frecuencias. Pero malas cuando la frecuencia se incrementa. Por esto, la validación de la aproximación está limitada por el intervalo de frecuencia, donde el modelo es correcto. Por que el intervalo no es conocido de antemano, una buena práctica para repetir en el diseño con pocas interacciones, es iniciar con bajas frecuencias de cruce y gradualmente incrementarlas hasta que una discrepancia entre el modelo y el sistema físico se encuentre.

La segunda parte del diseño es encontrar las constantes del filtro, note que en la ecuación (3-34) puede ser escrita a detalle como

$$L(s) = \left(P + sD + \frac{I}{s} \right) e^{-sT/2} KH(s) \quad (3-37)$$

El procedimiento para el diseño propuesto es proponer dos constantes y calcular una tercera después. Como el efecto de I es de bajas frecuencias, se asumirá que I no tiene efecto en la magnitud en la ecuación (3-35), y que el efecto de I en la fase está limitado a un desvío de la fase de -5 grados.

Otra expresión de interés es el retardo asociado con el proceso de muestreo. La función de retardo tiene una magnitud unitaria por lo tanto no tiene efecto en la ecuación (3-35). Como quiera que sea el desvío de fase asociada con el retardo se encuentra como

$$G(j\omega_c) = e^{-j\omega_c T/2} \quad (3-38)$$

$$\text{Arg} \left[e^{-j\omega_c T/2} \right] = \frac{-\omega_c T}{2} \text{ rad} = \frac{-180\omega_c T}{2\pi} \text{ deg} \quad (3-39)$$

La simplificación nos permite iniciar el proceso de diseño con P y D y se escribe la ecuación (3-35) y (3-36) como

$$|(P + j\omega_c D)KH(j\omega_c)| = 1 \quad (3-40)$$

$$\Theta_m = 180 + \text{Arg}[H(j\omega_c)] + \tan^{-1}\left(\frac{\omega_c D}{P}\right) - \frac{180\omega_c T}{2\pi} - 5 \quad (3-41)$$

Porque $H(s)$ es conocida, la magnitud del argumento de $H(j\omega_c)$ puede ser encontrada, llevando estas dos simples ecuaciones algebraicas en P y D puede ser resueltas relativamente fácil.

Para sumar la constante I, se asumirá que el desvío causado por la constante integral es de -5 grados.

Se considera ahora el sistema de control para el motor, sensor y amplificador utilizado en este trabajo, se consideran los parámetros dados en la tabla 3.1.

El objetivo del diseño es lograr una frecuencia de cruce de $\omega_c = 200$ con un margen de fase de $\Theta = 45$.

Las constantes de tiempo del motor están dadas por la ecuación (3-23) y son

$$T_m = 0.058$$

$$T_e = 0.00041$$

Esto lleva a la función de transferencia del motor de

$$M(s) = \frac{4.13}{(0.058s + 1)(0.00041s + 1)} = \frac{1.7 \times 10^6}{(s + 172)(s + 2434)}$$

Los elementos del sistema resultantes se presentan en la siguiente figura 3.6

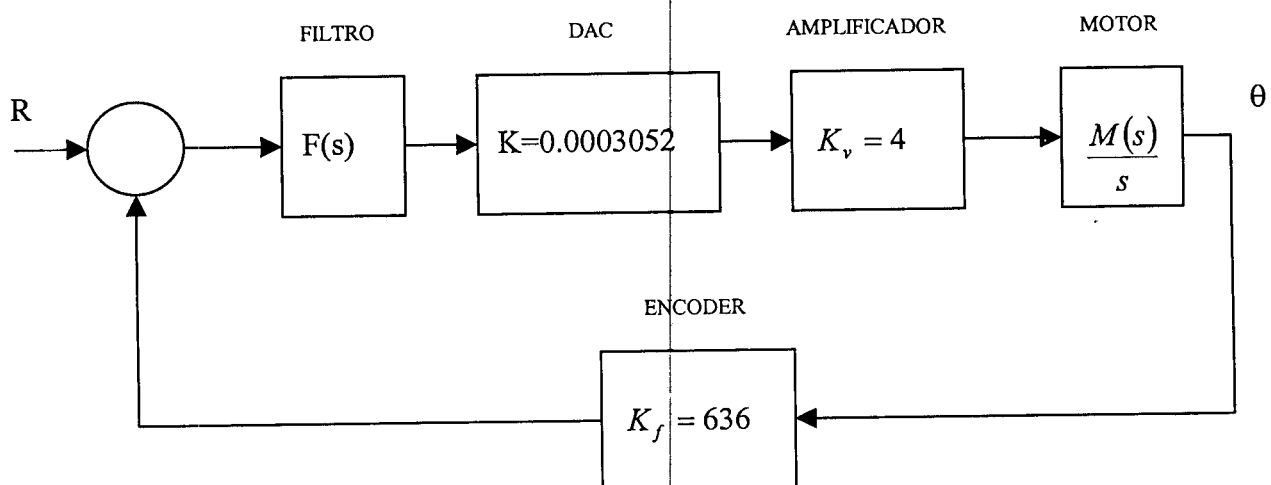


Figura 3.6: Diagrama a bloques del sistema a diseñar.

El primer paso en el diseño es encontrar la función de transferencia $H(s)$, del amplificador, motor y sensor, en el caso del amplificador es un amplificador de tensión, entonces $H(s)$ es

$$H(s) = \frac{K_v M(s) K_f}{s}$$

Con los parámetros dados esto es igual a

$$H(s) = \frac{4.4 \times 10^9}{s(s+172)(s+2434)}$$

La magnitud y el argumento de $H(s)$ a la frecuencia de cruce de 200 rad/s son encontrados de

$$H(j\omega) = \frac{4.4 \times 10^9}{j500(j500+172)(j500+2434)}$$

Resultando en

$$A = |H(j500)| = 6.7$$

$$\alpha = \arg|H(j500)| = -172.6$$

El correspondiente ángulo β

$$\beta = 57$$

Con una ganancia del DAC $K = 0.0003052$ los parámetros PID son:

$$P = \frac{1}{(6.7)(0.0003052)} \cos 57^\circ = 266$$

$$D = \frac{1}{(6.7)(0.0003052)(500)} \sin 57^\circ = 0.82$$

$$I = (500)(266) \tan 5^\circ = 11600$$

3.6. Análisis del servosistema en tiempo discreto.

Una vez diseñado el servocontrol es necesario el análisis del modelo de la planta y el controlador en el dominio discreto, para determinar si es estable y que efectos tienen la cuantización y el tiempo de muestreo en la estabilidad del sistema.

3.6.1. El controlador PID digital.

El esquema de control PID analógico ha sido usado de manera exitosa en muchos sistemas de control industrial por más de medio siglo. El principio básico del esquema de control PID, es que actúa sobre la variable a ser manipulada a través de una apropiada combinación de las tres acciones de control: acción de control proporcional; la acción de control integral y la acción de control derivativa.

En situaciones donde muchas plantas se controlan directamente mediante un sistema digital (este esquema de control puede comunicarse con cualquier otro sistema digital), la mayoría de los lazos pueden manipularse mediante un sistema de control PID.

Según (Ogata, K.1996), un PID analógico puede ser adaptado a un sistema digital de la siguiente manera.

La acción de control PID en controladores analógicos está dada por:

$$m(t) = K[e(t)] + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \quad (3-42)$$

Donde $e(t)$ es la señal al controlador, $m(t)$ es la señal de salida del controlador, K es la ganancia proporcional, T_i es en tiempo integral y T_d es el tiempo derivativo.

Para obtener la función de transferencia impulso del controlador digital PID, se puede discretizar la ecuación (3-42). Al aproximar el término integral mediante una sumatoria y el término derivativo mediante la diferencia de dos puntos, se obtiene

$$m(kT) = K \left\{ e(kT) + \frac{T}{T_i} \left[\frac{e(0) + e(T)}{2} + \frac{e(T) + e(2T)}{2} + \dots + \frac{e((k-1)T) + e(kT)}{2} \right] + T_d \frac{e(kT) - e((k-1)T)}{T} \right\}$$

o

$$m(kT) = K \left\{ e(kT) + \frac{T}{T_i} \sum_{h=1}^k \frac{e((h-1)T) + e(hT)}{2} + T_d \frac{e(kT) - e((k-1)T)}{T} \right\}$$

lo cual es válido solo si $\frac{f_s}{f_{\max}} > 20$ se define

$$\frac{e((h-1)T) + e(hT)}{2} = f(hT), \quad f(0) = 0$$

Entonces se escribe

$$Z \left[\sum_{h=1}^k \frac{e((h-1)T) + e(hT)}{2} \right] = Z \left[\sum_{h=1}^k f(hT) \right] = \frac{1}{1-z^{-1}} F(z)$$

Además se tiene que la transformada Z de $f(hT)$ es

$$F(z) = Z[f(hT)] = \frac{1+z^{-1}}{2} E(z)$$

Por lo tanto

$$Z \left[\sum_{h=1}^k \frac{e((h-1)T) + e(hT)}{2} \right] = \frac{1+z^{-1}}{2(1-z^{-1})} E(z)$$

Entonces la transformada z de la ecuación (3-42) da como resultado

$$M(z) = K \left[1 + \frac{T}{2T_i} \frac{1+z^{-1}}{1-z^{-1}} + \frac{T}{T_d} (1-z^{-1}) \right] E(z)$$

Esta última ecuación se puede describir como sigue

$$M(z) = K \left[1 - \frac{T}{2T_i} + \frac{T}{T_i} \frac{1}{1-z^{-1}} + \frac{T}{T_d} (1-z^{-1}) \right] E(z)$$

o

$$M(z) = \left[K_p + \frac{K_I}{1-z^{-1}} + K_D (1-z^{-1}) \right] E(z) \quad (3-43)$$

donde

$$K_p = K - \frac{KT}{T_i} = K - \frac{K_I}{2} = \text{Ganancia proporcional.}$$

$$K_I = \frac{KT}{T_i} = \text{Ganancia integral}$$

$$K_D = \frac{KT_D}{T} = \text{Ganancia derivativa}$$

A esta función de transferencia del PID digital se le conoce comúnmente como forma posicional del esquema PID.

Considerando los resultados de la sección anterior se tiene que

$$T_i = 0.023 \quad T_d = 0.000308$$

Si $T = 0.001$ se tiene

$$K_p = 266 - \frac{(0.001)(266)}{2(0.023)} = 260.25$$

$$K_I = \frac{(266)(0.001)}{0.023} = 11.5$$

$$K_D = \frac{(266)(0.00308)}{0.001} = 819.2$$

3.6.2. Cuantificación y errores de cuantificación.

Las principales funciones involucradas en la conversión analógico digital, son el muestreo, la cuantificación de la amplitud y la codificación. Cuando el valor de cualquier muestra cae entre dos estados de salida adyacentes, permitidos, se debe leer como el estado de salida permitido más cercano al valor real de la salida. El proceso de representación de una señal continua o analógica mediante un número finito de estados discretos se denomina cuantificación de amplitud. Esto es cuantificación, significa la transformación de una señal continua o analógica a un conjunto de estados discretos.

El estado de salida de cualquier muestra cuantificada se describe entonces mediante un código numérico. El proceso de representar el valor de una muestra mediante un código numérico, se denomina codificación. De este modo la codificación es el proceso de asignación de una palabra o código digital a cada uno de los estados discretos. El periodo de muestreo y los niveles de cuantificación afectan al desempeño de los sistemas de control digital. De manera que éstos de deben determinar cuidadosamente.

En el sistema numérico estándar utilizado para el procesamiento de señales digitales es el sistema binario. En este sistema numérico el grupo de códigos consiste en n

pulsos cada uno de los cuales indican ya sea encendido o apagado. En el caso de la cuantificación, los n pulsos pueden representarse en dos niveles de amplitud o estados de salida. El nivel de cuantificación Q se define como el intervalo entre dos puntos adyacentes de decisión y está dado mediante

$$Q = \frac{FSR}{2^n}$$

Donde FSR es el intervalo a escala completa. Observe que el bit que está más a la izquierda del código binario natural tiene el mayor peso, la mitad de la escala completa y se le conoce como el bit más significativo (MSB). El bit que está más a la derecha tiene el menor peso, y se le conoce como el bit menos significativo. De ésta manera.

$$LSB = \frac{FSR}{2^n}$$

El bit menos significativo es el nivel de cuantificación Q .

Puesto que el número de bits de la palabra digital es finito. La conversión analógico-digital da como resultado una resolución finita. Esto es, la salida digital puede adoptar solamente un número finito de niveles, y por lo tanto, un número analógico se debe redondear al nivel digital más cercano. Por consiguiente, toda conversión analógico digital involucra un error de cuantificación. Dicho error de cuantificación varía entre 0 y $\pm 1/2Q$. Este error depende de la fineza del nivel de cuantificación y se puede hacer tan pequeño como se desee haciendo más pequeño el nivel de cuantificación. En la práctica, existe un máximo para el número n de bits. Y de ese modo siempre existe algún error debido a la cuantificación. La incertidumbre presente en el proceso de cuantificación se conoce como ruido de cuantificación.

En el análisis numérico, el error resultante de despreciar los dígitos remanentes se denomina error de redondeo. Debido a que el proceso de cuantificación es un proceso de aproximación en el que la cantidad analógica se aproxima mediante un número digital finito, el error de cuantificación es un error de redondeo. Es claro que mientras más fino sea el nivel de cuantificación, más pequeño será el error de redondeo. Para un nivel de cuantificación pequeño Q la naturaleza del error de cuantificación es similar a la del ruido aleatorio. Y en efecto, el proceso de cuantificación actúa como una fuente de ruido

aleatorio. Según (Ogata, K.1996), si el nivel de cuantificación Q es pequeño comparado con la amplitud promedio de la señal de entrada, entonces la varianza del ruido de cuantificación es un doceavo del cuadrado de cuantificación. En nuestro caso se tiene que esta varianza es igual a un doceavo de cuenta, es decir 0.00013 radianes de varianza del error de la posición, el cual no es significativo para el intervalo que se maneja.

3.6.3. Análisis de estabilidad del sistema en z .

A continuación, se analiza la estabilidad del sistema de control en tiempo discreto, para sistemas lineales e invariantes en el tiempo de una entrada y una salida; como lo es nuestro caso. Se considerará el siguiente sistema con función de transferencia pulso en lazo cerrado

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + GH(z)} \quad (3-44)$$

La estabilidad del sistema que define la ecuación (3-44), así como la de otros tipos de sistemas de control en tiempo discreto, puede determinarse por las localizaciones de los polos en lazo cerrado en el plano z , o por las raíces de la ecuación característica

$$P(z) = 1 + GH(z) = 0$$

como sigue

- 1 Para que el sistema sea estable, los polos en lazo cerrado o las raíces de la ecuación característica deben presentarse en el plano z dentro del círculo unitario. Cualquier polo en lazo cerrado exterior al círculo unitario hace inestable al sistema.
- 2 Si un polo simple se presenta en $z=1$, entonces el sistema se convierte en críticamente estable. También el sistema se convierte en críticamente estable si sólo un par de polos complejos conjugados se presenta sobre el círculo unitario en el plano z . Cualquier polo múltiple en lazo cerrado sobre el círculo unitario hace al sistema inestable.
- 3 Los ceros en lazo cerrado no afectan la estabilidad absoluta y por lo tanto pueden quedar en cualquier punto del plano z .

Entonces, un sistema de control en lazo cerrado en tiempo discreto lineal e invariante con el tiempo de una entrada y una salida se vuelve inestable si cualquiera de los polos en lazo cerrado se presenta por fuera del círculo unitario y/o cualquier polo múltiple en lazo cerrado se presenta sobre un círculo unitario del plano z .

En el servosistema mostrado en la figura 3.7.

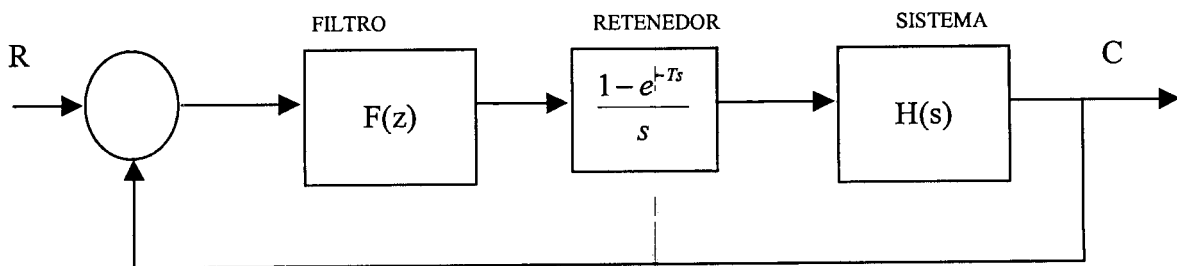


Figura 3.7: Sistema de control digital.

Donde $F(z)$ y $H(s)$ son

$$F(z) = \left[K_p + \frac{K_I}{1-z^{-1}} + K_D(1-z^{-1}) \right] E(z)$$

$$H(s) = \frac{1319929}{s(s+172)(s+2434)}$$

Se define $G_p(s)$ como

$$G_p(s) = \frac{1-e^{-Ts}}{s} \frac{4.4 \times 10^9}{s(s+172)(s+2434)}$$

Se determinará la estabilidad del servosistema para $K_p=260.25$ $K_i=11.5$ y $K_d=819.2$. Primero se obtendrá la función de transferencia en z del retenedor de orden cero y el servosistema $G_p(z)$.

$$Z[G_p(s)] = Z \left[\frac{1-e^{-Ts}}{s} \frac{1319929}{s(s+172)(s+2434)} \right]$$

$$Z[G_p(s)] = G_p(z) = (z-1) \left(\frac{0.0197}{z-0.84} - \frac{9.8 \times 10^{-5}}{z-0.0876} + \frac{0.0031}{(z-1)^2} - \frac{0.0196}{z-1} \right)$$

Entonces según (3-44)

$$G(z) = F(z)G_p(z) \quad \text{y} \quad H(z) = 1$$

La función de transferencia en lazo cerrado para el servosistema es:

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + G(z)}$$

la ecuación característica es

$$1 + G(z) = 0$$

la cual mediante el uso del programa (Derive) se convierte en

$$0.7942z^5 - 2.2498z^4 + 2.4917z^3 - 1.2865z^2 + 0.2344z + 0.0202 = 0$$

Usando el programa (MATLAB) se encontraron sus raíces de la ecuación anterior y son:

$$z_1 = 0.5969 + j0.4646$$

$$z_2 = 0.5969 - j0.4646$$

$$z_3 = 0.9462$$

$$z_4 = 0.7550$$

$$z_5 = -0.0622$$

En vista que las magnitudes de los polos del servosistema en lazo cerrado están dentro del círculo unitario del plano z el sistema es estable.

Se debe de tener especial cuidado con los errores de cuantificación, errores por redondeo, dado que estos pueden mover la posición de las raíces y ubicarlas fuera del círculo unitario e inestabilizar el sistema. En este caso no se tiene este problema debido a la resolución que se maneja.

Las características de la respuesta transitoria del sistema de control en tiempo discreto dependen del periodo de muestreo T . Un periodo de muestreo grande tiene efectos dañinos sobre la estabilidad relativa del sistema.

Como se ve en el análisis precedente, para los valores dados de las constantes del PID digital, aumentar el periodo de muestreo T hará que el sistema de control en tiempo discreto sea menos estable y que eventualmente se convierta en inestable. De manera alternativa, al reducir el periodo de muestreo T , se permite que los valores críticos de las

ganancias respecto a la estabilidad sean mayores. De hecho, reducir el periodo de muestreo más y más tiende a hacer que el sistema se comporte muy parecido a un sistema en tiempo continuo.

En el sistema mostrado en la figura 3.7 se puede observar que las raíces del sistema dependen del periodo T . Si se aumenta el periodo T a 0.01 y se aplica el procedimiento anterior se obtienen las raíces.

$$z_1 = -5.873$$

$$z_2 = -0.918$$

$$z_3 = 0.711$$

$$z_4 = 0.221$$

$$z_5 = -0.001$$

En donde, se puede ver que z_1 está fuera del círculo unitario del plano z y por lo tanto el sistema es inestable en lazo cerrado.

4. ELABORACIÓN DEL SISTEMA.

4.1. Introducción.

El servomecanismo se diseñó de forma digital como se muestra en la figura 4.1. Es un sistema de control de motor con un dispositivo incremental de retroalimentación, conocido como encoder incremental. El sistema consiste en un contador de encoder, un procesador, un convertidor analógico a digital, una interfaz de órdenes digitales y un amplificador de potencia la cual es una interfaz entre el control digital y el motor.

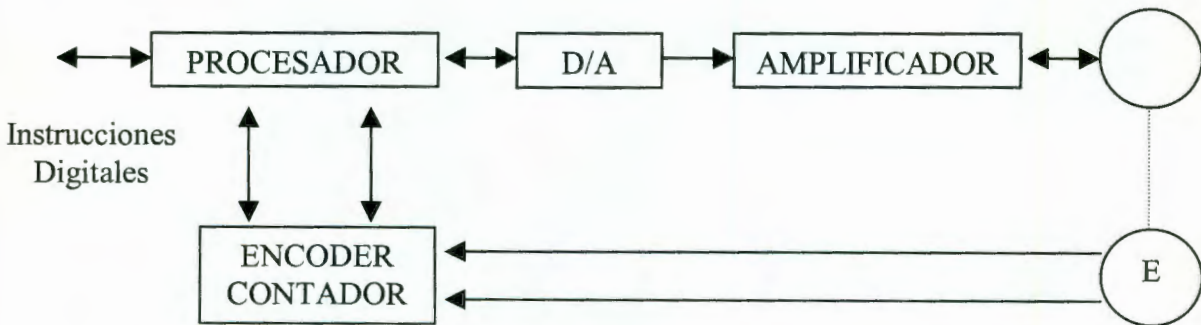


Figura 4.1: Diagrama a bloques del servocontrol.

En el servomecanismo se realizó un algoritmo servocompensador y otro generador de trayectoria. El algoritmo generador de trayectoria es necesario para un óptimo movimiento y es tan importante como el servocompensador. El servocompensador puede ser realizado como un filtro digital tradicional o como un algoritmo PID simple. Los cálculos combinados del servocompensador y la generación de trayectoria generan significativas demandas en el procesador.

El convertidor digital analógico puede ser manejado por un convencional DAC o usando modulación de ancho de pulso (PWM), en cualquiera de los casos puede ser adaptado al servoamplificador o etapa de potencia.

La salida PWM puede en una sola señal llevar toda la información combinada de la señal de dirección y de orden, en este caso se separa la dirección de la orden con la finalidad de darle más resolución al PWM.

El encoder secuencial produce un tren de pulsos cuadrados, desde los cuales la posición, velocidad y dirección pueden ser deducidos. La frecuencia es proporcional a la velocidad y cada transición representa un incremento en posición. La fase de las señales es usada para determinar la dirección de rotación.

La señal del encoder es decodificada usando una máquina de estados en dos señales de pulsos, estos pulsos son leídos e interpretados por dos contadores los cuales los guardan en forma digital y representan la posición del motor. El contador-decodificador es realizado en una combinación de sistemas físicos y programas y es una parte fundamental del sistema por ser la retroalimentación del movimiento. En los apéndices B y C se presentan los diagramas eléctricos generales del sistema.

4.2. Funcionamiento del motor.

Para mover el rotor en la dirección de la orden, el amplificador enviará corriente a través de dos de las fases del motor. Esta corriente produce un campo electromagnético que desarrolla un torque en el rotor, y el rotor gira. El rotor se parará si éste puede alcanzar una posición, donde su imán permanente esté próximo al campo magnético que los atrae. Antes que el rotor pueda alcanzar esta posición, el amplificador conmuta la corriente a una nueva combinación de las bobinas del estator y crea un nuevo conjunto de campo electromagnético, que causa que el rotor continúe su movimiento.

El proceso de cambiar continuamente la corriente a diferentes bobinas del motor produce torque en el motor y esto substituye a las escobillas del motor de CD común.

Si el amplificador conoce la posición del imán permanente del rotor, éste puede instalar campos magnéticos en el estator que tiene una correcta posición y polaridad y causa que el rotor gire. Con tres sensores de efecto de campo localizados en el motor son afectados por el imán permanente del rotor. Los tres sensores transmiten una única combinación de señales para cada posición del rotor. El amplificador usa estas señales para determinar la posición del rotor.

Hay tres sensores de efecto de campo dentro del motor. En la figura 4.2 se presenta un dibujo conceptual del interior del motor, y tres sensores.

Por claridad el estator está representado en forma simplificada, sin sus embobinados. Los sensores de efecto de campo están localizados en un extremo del estator, cerca de los polos del rotor. Estos están posicionados aproximadamente como se presenta en la figura 4.2.

Se tiene disponibilidad de cinco cables para conectar los sensores de efecto de campo. Tres son para salidas individuales, la cuarta y quinta son para la alimentación de los sensores. Los sensores están disponibles en una gran variedad de salidas de tensiones y polaridades. El tipo de sensor utilizado en el motor utilizado para este trabajo, se enciende con un polo magnético sur, y cuando se acerca un polo magnético norte la salida es apagada.

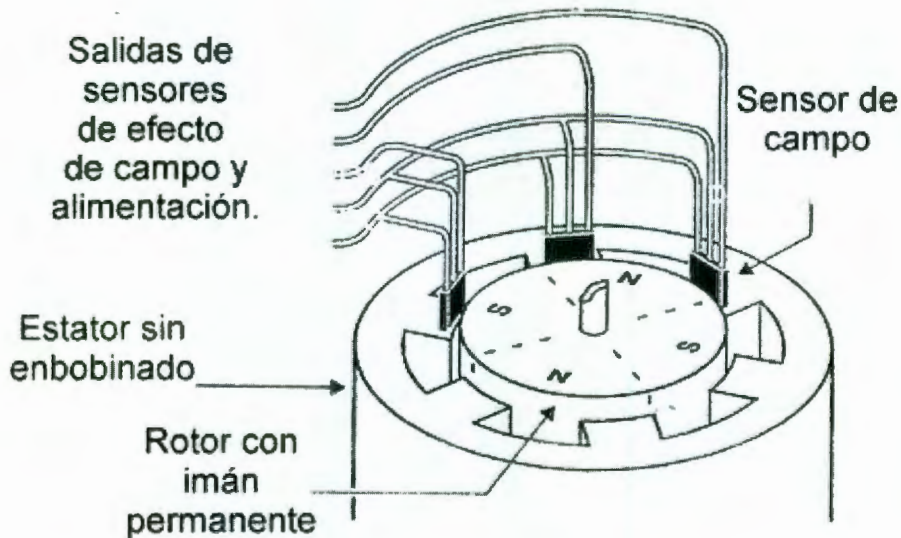


Figura 4.2: Localización del sensor de efecto de campo.

En la figura 4.3 se representa una vista final del motor, con las fases separadas; se presentan sus posiciones relativas de alrededor del estator, las tres fases forman parte de una conexión central como se presenta a detalle en las líneas punteadas de la figura 4.3.

La separación física entre los sensores de efecto de campo es muy importante. Se nota que un polo del rotor puede afectar dos sensores al mismo tiempo. En la figura 4.3, el polo norte del rotor está adyacente a dos sensores, y el otro sensor está sobre un polo magnético norte apagando su salida, esto nos presenta una combinación única en las salidas de los sensores de efecto de campo.

El amplificador enviará una corriente por una fase y saldrá por otra, la tercera fase no recibirá corriente. Cuando la corriente fluye a través de las fases, dos polos magnéticos del mismo signo son formados en lado opuesto del motor. Si se usa la convención de la figura 4.3, cuando la corriente fluye del amplificador hacia la bobina, ésta produce un polo norte, y cuando fluye de la bobina hacia el amplificador, producirá un polo sur.

El amplificador verifica sus tres entradas de los sensores de efecto de campo y usa un circuito combinacional para asignar una posición al rotor por cada uno de los seis estados de los sensores de efecto de campo, y entonces dirige al motor una corriente que resulta en un movimiento del rotor en la dirección de la orden.

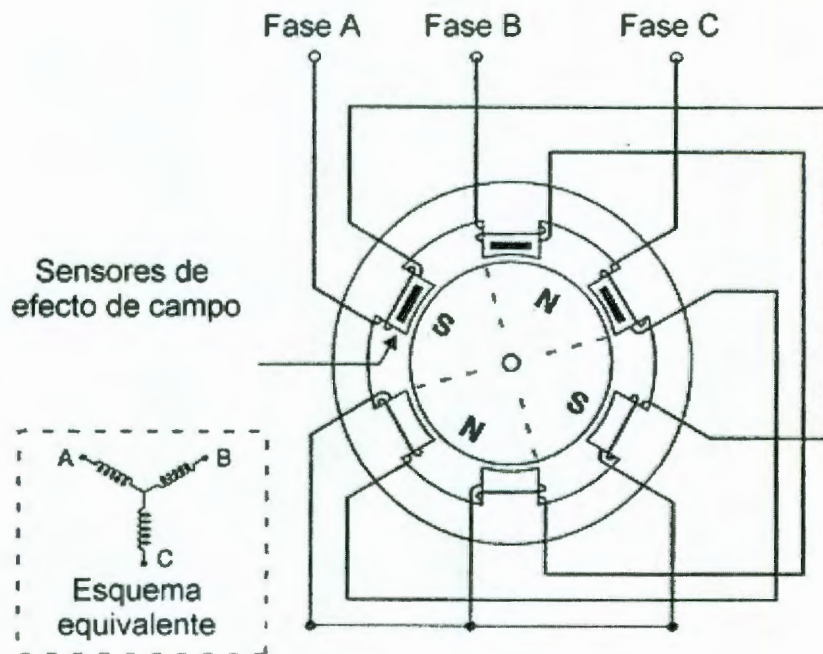


Figura 4.3: Servomotor de tres fases con sensores de efecto de campo.

Las tres señales de los sensores de efecto de campo producen una rotación positiva en la flecha como se presenta en la figura 4.4. El estado de los sensores es presentado y a lo largo de la figura las corrientes de las fases del amplificador usado para cada estado de los sensores de efecto de campo.

Para una rotación negativa, se deben hacer dos cambios: Primero, como el rotor se mueve negativamente, éste pasa a través del mismo estado de los sensores, pero en orden

opuesto (ver la secuencia de abajo hacia arriba de la figura 4.4). Segundo, las corrientes deben cambiar de sentido y de esa manera el movimiento es negativo.

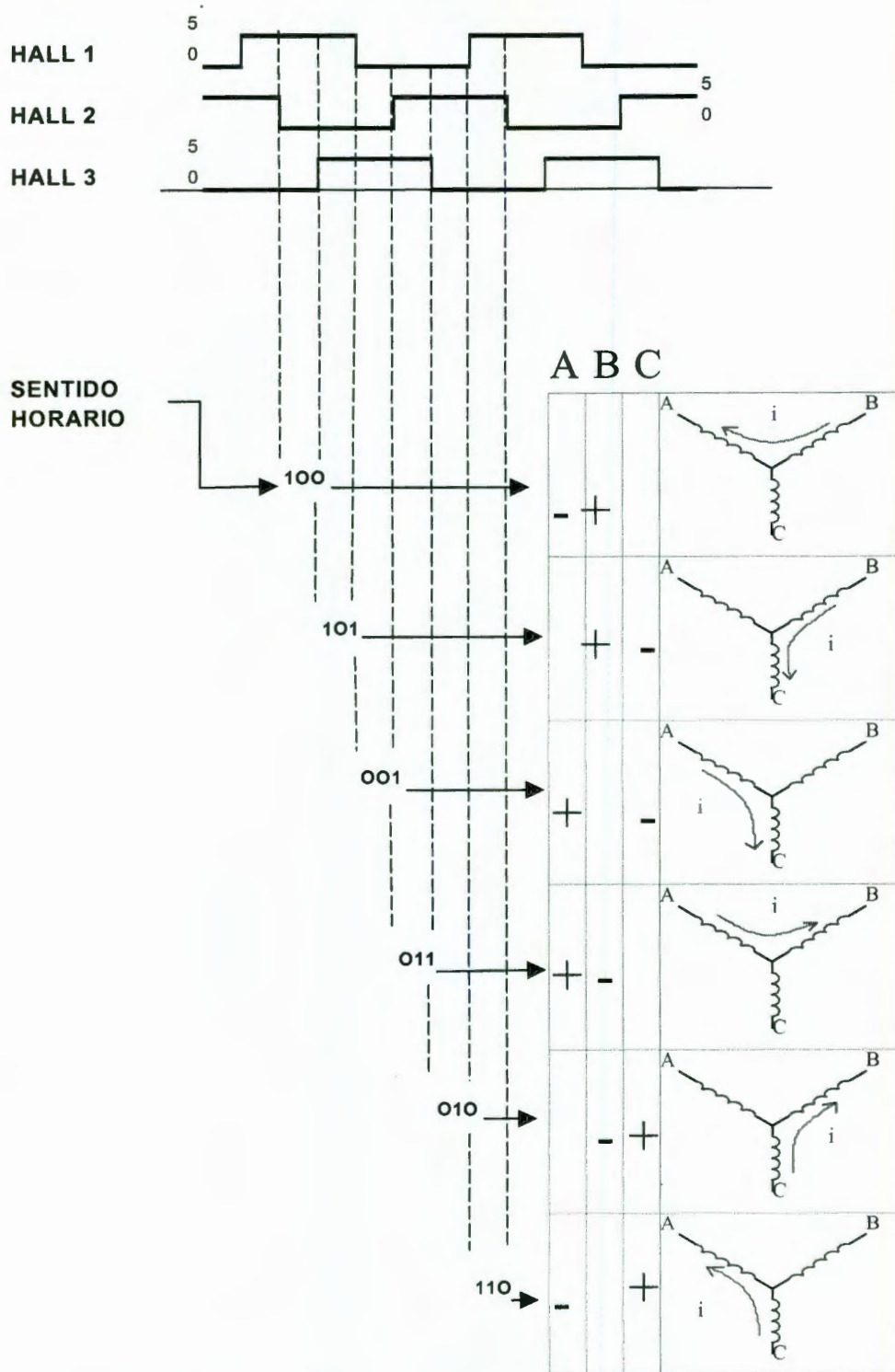


Figura 4.4: Secuencia de conmutación de las bobinas del motor.

El amplificador envía corriente a través de las mismas bobinas, pero la dirección de la corriente es inversa a la que se presenta en la figura 4.4. Esto resulta en la generación de un torque por cada estado y causa que el rotor gire inversamente.

En la figura 4.4, se muestra la secuencia de conmutación basada en los estados de campos del rotor para generar un giro en la flecha del motor. Esta secuencia está realizada en el amplificador de potencia en el bloque codificador de señales.

4.3. El servoamplificador

El microcontrolador contiene un subsistema de modulación de ancho de pulso de alta resolución. Ésta es una forma muy eficiente para realizar un convertidor de digital a analógico cuando se realiza con un puente de potencia de conmutación. Cuando la frecuencia de conmutación es de 15.6 KHz. El PWM mantiene una alta resolución y esto permite un control efectivo de la tensión mientras se mantiene la modulación por arriba de la sensibilidad del oído humano. Esto tiene una especial importancia dado que en determinados lugares se requiere un mínimo de ruido.

El amplificador opera en modo tensión, lo cual significa que la señal ordenada es amplificada y aplicada al motor. Esta corriente causa un movimiento en el motor.

La entrada al amplificador es una señal llamada entrada de la orden y es una señal digital la cual proviene de un PWM y tiene un intervalo de 5% al 95 %, la salida de voltios es escalada por este porcentaje del potencial eléctrico de la alimentación del amplificador que es de 72 voltios; es decir, cuando la señal ordenada está en un 50% la tensión aplicada al motor es de 36 voltios. Adicionalmente, el amplificador recibe una señal de dirección, la cual nos indica en que dirección se debe aplicar la orden de tensión, por lo tanto es el signo de la orden. A su vez el amplificador responde con una señal de falla cuando existe una operación anormal en el amplificador.

La orden de entrada que proviene del PWM es una señal digital de 0 a 5 voltios, la cual en combinación con los sensores de efecto de campo y la línea de dirección generan la secuencia de pulsos necesaria para mover el motor. En donde se aplica baja tensión si el ancho del pulso es delgado y alta tensión si el ancho del pulso es amplio, y el sentido de giro será positivo si la línea de dirección está apagada y negativo si está encendida.

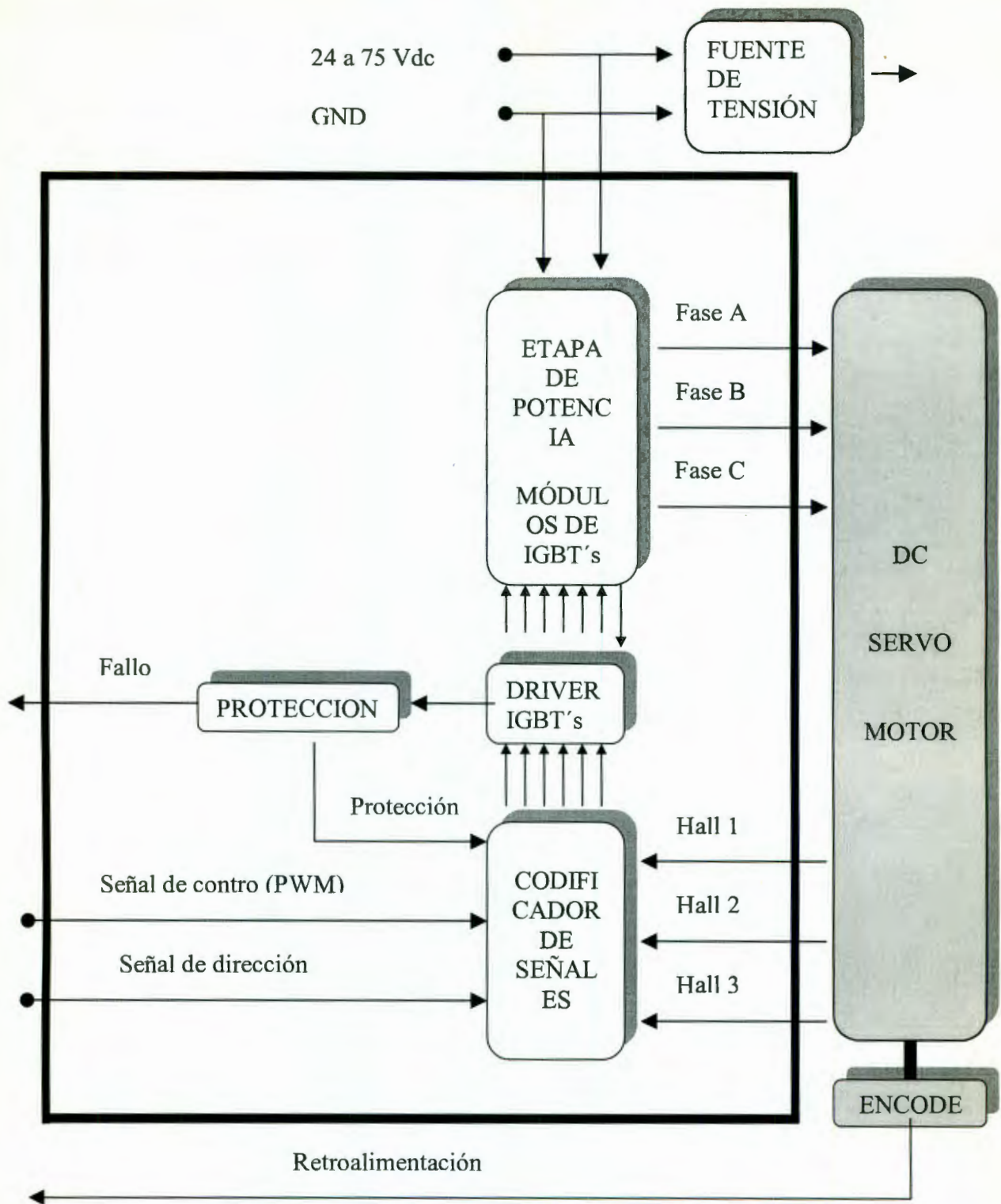


Figura 4.5: Diagrama a bloques de la etapa de potencia o amplificador.

El amplificador tiene un sistema de protección y falla, el cual verifica sobre corrientes, corto circuito y temperatura (incluidos en el IR2137J, apéndice B). Estos pueden

detener el amplificador si se exceden sus límites, más adelante se tratará este tema con detalle.

En el sistema que se presenta en la figura 4.5 se describe de manera general el sistema de control de potencia o amplificador.

4.4. Secuencia de conmutación de tensión.

El módulo de IGBT's (CM15TF-12H de POWEREX apéndice B) tiene tres salidas, cada una está conectada a una fase del motor. El amplificador obtiene señales desde los sensores de efecto de campo del motor, y determina en cual de las seis posiciones del rotor se encuentra. A su vez por medio de un circuito combinacional y un circuito de acoplamiento con el módulo de IGBT's aplica tensión en dos de sus bobinas, la tercera bobina queda en tercer estado. Esta tensión crea una corriente y al mismo tiempo un torque en el rotor, el cual gira a su siguiente posición. El amplificador lee la nueva posición de los sensores de efecto de campo, y conmuta la tensión a diferente combinación de bobina. El rotor gira nuevamente y el proceso se repite causando un movimiento continuo en el motor.

En la figura 4.6 se presenta el diseño del bloque de codificador de señales mostrado en la figura 4.5 en donde se toman las señales de los sensores de efecto de campo del motor y la señal de orden que provienen de controlador, que son PWM, DIR y EN. Con esta información se debe generar los disparos adecuados para los seis IGBT's que están conectados directamente en las fases del motor.

Este circuito combinacional fue realizado en un dispositivo lógico programable (GAL16V8D mostrado como U1 en el apéndice C), específicamente en un arreglo lógico genérico, por su gran versatilidad, disponibilidad en el mercado, bajo costo, y tamaño.

LOGICA DE CONMUTACION

	DIR	H1	H2	H3	Q1	Q2	Q3	Q4	Q5	Q6
HORARIO	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	1	0	0	0	0	1
	0	0	1	0	0	0	1	0	1	0
	0	0	1	1	1	1	0	0	0	1
	0	1	0	0	0	0	1	0	1	0
	0	1	0	1	0	0	1	0	0	1
	0	1	1	0	0	0	0	1	1	0
	0	1	1	1	0	0	0	0	0	0
ANTHORARIO	1	0	0	0	0	0	0	0	0	0
	1	0	0	1	0	0	1	1	0	0
	1	0	1	0	0	1	0	0	0	1
	1	0	1	1	0	1	0	1	0	0
	1	1	0	0	1	0	0	0	1	0
	1	1	0	1	0	0	1	0	1	0
	1	1	1	0	1	1	0	0	0	1
	1	1	1	1	0	0	0	0	0	0

SENSOR	1	2	3	4	5	6
H1	1	1	0	0	0	1
H2	0	1	1	1	0	0
H3	0	0	0	1	1	1

FUNCIONES

$$Q1 = \overline{H1} \cdot H3 \cdot \overline{ID} + H1 \cdot \overline{H3} \cdot D$$

$$Q2 = H1 \cdot \overline{H2} \cdot \overline{ID} + \overline{H1} \cdot H2 \cdot D$$

$$Q3 = H2 \cdot \overline{H3} \cdot \overline{ID} + \overline{H2} \cdot H3 \cdot D$$

$$Q4 = \overline{H1} \cdot H3 \cdot D + H1 \cdot \overline{H3} \cdot \overline{ID}$$

$$Q5 = H1 \cdot \overline{H2} \cdot D + \overline{H1} \cdot H2 \cdot \overline{ID}$$

$$Q6 = H2 \cdot \overline{H3} \cdot D + \overline{H2} \cdot H3 \cdot \overline{ID}$$

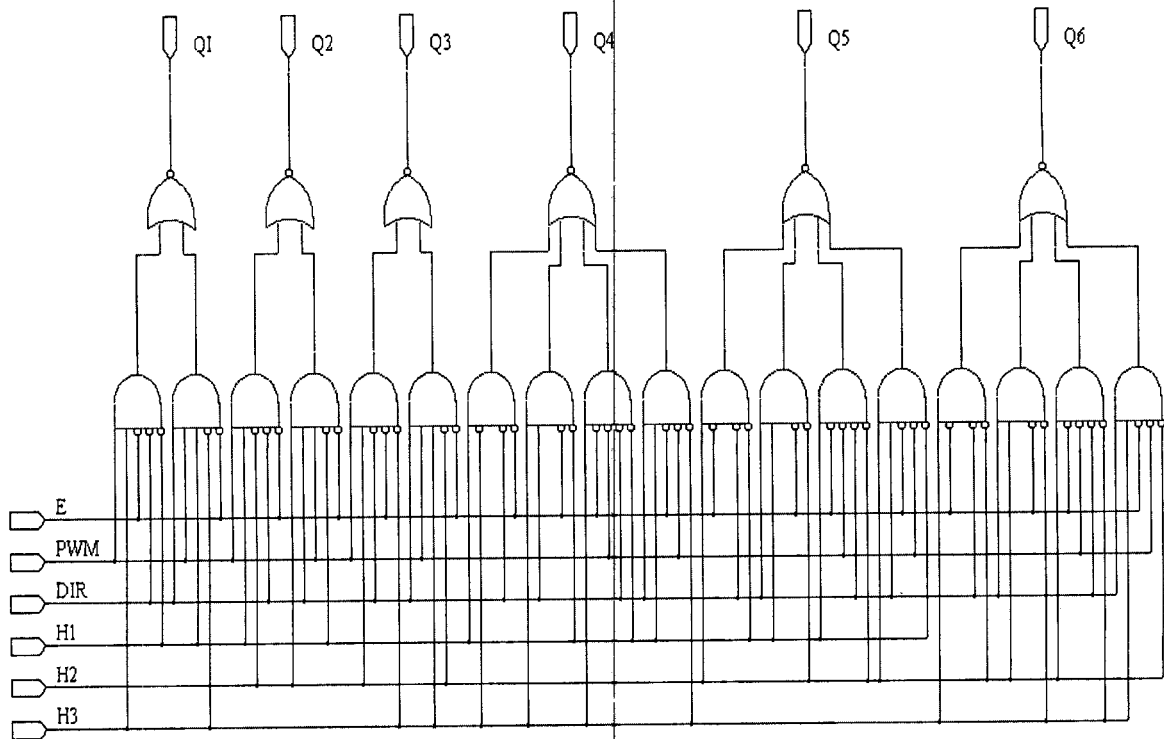


Figura 4.6: Tabla de verdad y funciones lógicas para la secuencia de conmutación.

4.5. Controlador para puente de tres fases.

4.5.1. Características.

Canal flotante hasta 600 Voltios.

Suave apagado por sobre-corriente en las seis salidas.

Control de encendido suave.

Controlador para IGBT integrado para freno dinámico.

Tres controladores independientes con pines comunes.

Amplificador de salidas de encendido y apagado separadas.

Retardos igualados en las salidas.

Suspensión por baja tensión con banda de histéresis.

4.5.2. Descripción.

El IR2137 es un controlador para IGBT de tres fases, para aplicaciones con motores de AC de alta tensión. Tiene lógica integrada de de-saturación, protección de falla a tierra, tan buena como cualquier modo de protección contra sobre-corriente. Suave apagado que inicia en un evento de alguna condición de falla de sobre-corriente o tierra y todas las salidas son apagadas. Las salidas del controlador tienen pines de encendido y apagado separadas para facilitar la manipulación de la impedancia del controlador de compuerta con un suave encendido. Óptimos retardos igualados entre las fases y los flancos de subida o bajada; habilitados pequeños tiempos muertos, de ese modo se improvisa la ejecución de baja velocidad. El controlador para el freno elimina circuitos adicionales.

4.5.3. Introducción.

En la pasada década, controladores de corriente alterna industriales han llegado a ser más avanzados y en dimensiones más compactas. Esto es doblemente bueno por el rápido desarrollo de la tecnología de la potencia de silicón, incluyendo el crecimiento de los IGBT's y los circuitos integrados de alta tensión. Adicionalmente el crecimiento de la tecnología ha sido particularmente acelerado para controladores de AC de baja potencia. Aún con protección de falla a tierra, son disponibles ampliamente sólo en últimos modelos con ayuda de sensores y otros circuitos involucrados. El mercado continuó demandando

total protección, en sistemas más económicos. Por esta razón se ha introducido el IR2137, una nueva generación de controladores para IGBT de tres fases, caracterizando toda protección de IGBT con una característica adicional de suave apagado.

4.5.4. Teoría de falla para protección.

Una de las más comunes y fatales fallas de los controladores de AC con IGBT es la condición de sobre-corrientes. En la tabla 4.1 se enlista las tres causas más comunes de modos de sobre-corrientes.

Modos de sobre-corrientes	Causa potencial
Corto de línea a línea	Extraño cableado. Las terminales del motor en corto. El aislamiento de las fases del motor roto.
Falla de poner a tierra.	Aislamiento de las fases del motor roto.
Disparos continuos	Falso encendido de IGBT

Tabla 4.1: Potenciales causas de sobre corriente.

En orden para proteger efectivamente los dispositivos IGBT bajo una condición de sobre-corriente, se necesita el enfoque en dos factores.

Primero se necesita detectar el modo de sobre-corriente y detener el sistema. En modo corto de línea a línea y disparos en serie, la corriente de corto circuito fluye en el bus del capacitor, figuras 4.7 y 4.9. Por lo tanto, un resistor de cierre a tierra puede detectar esa condición de cierre a tierra. Como siempre, en el modo de falla de corto a tierra, el flujo de corriente desde la entrada de la línea de AC continúa en el bus de DC positivo en la parte alta de los IGBT a tierra, (figura 4.8). Así como una condición de sobre-corriente debe ser detectado con una resistencia de cierre en el bus positivo o detectando la corriente en las líneas de salida.

Segundo, se necesita ver la arquitectura del controlador de AC y su influencia sobre el método. Para esta instancia, la figura 4.10 presenta la típica arquitectura de referencia a tierra flotante, en el cual el microcontrolador está referenciado a una tierra flotante, y por tanto el circuito de detección de sobre corriente necesita ser aislado. Una arquitectura alterna puede reducir o no aislar.

4.5.5. Métodos tradicionales de detección de sobre-corriente.

Método 1.

Se pueden detectar corrientes de cortos de línea a línea y de disparo en serie insertando sensores de efecto de campo o un acoplador óptico lineal a través de la resistencia de cierre. El dispositivo estaría en serie con la línea del bus negativo. Para protección de falla a tierra un sensor adicional de corriente de efecto de campo en la línea de entrada de AC o en el bus de DC. El circuito de protección es realizado entonces usando comparadores rápidos. Las salidas de estos comparadores se realizan con un microcontrolador o con un generador de PWM para iniciar el apagado de las señales de las compuertas.

Método 2.

Si se instala en la salida de la fase del motor, sensores de efecto de campo usando dos comparadores. Esto se debe a que las corrientes positivas y negativas fluyen durante la condición de corto circuito de línea a línea. Otra consideración importante es el retardo total de propagación para el apagado. El tiempo de retardo asociado con un acoplador óptico en el controlador de la compuerta y el sensor de efecto de campo es típicamente más de 2 microsegundos. Por lo tanto estimando la realización del circuito de protección, este retardo podría ser sumado al retardo del circuito incrementando el tiempo de duración del corto circuito en los IGBT's. Como se ve en la Figura 4.10, la configuración requiere dos sensores de efecto de campo o aisladores ópticos además del circuito de protección. El circuito de comparación requiere de tensiones de referencia, capacitores y resistores.

Método 3.

Otro método de protección es el uso de un circuito de de-saturación en un IGBT. Este circuito discreto puede ser construido con un controlador óptico de la compuerta (estos circuitos pueden ser encontrados en el mercado). Estos circuitos detectan la tensión generada que cruza a través del colector y emisor mientras el circuito está totalmente encendido. Si esta tensión excede los límites especificados, la señal de la compuerta asociado es apagada. Este circuito discreto requiere de un comparador con la tensión de referencia, y un diodo de alta tensión además de varios resistores y capacitores.

4.5.6. Métodos tradicionales de cese de sobre corriente.

Para cortar la conducción de un IGBT, cuando una condición de sobre corriente es detectada, un apagado suave es preferido porque esto reduce picos de alta tensión a través del colector y emisor del dispositivo IGBT en falla. De ese modo se provee de un margen durante la condición de corto circuito y el circuito de la red snuber puede ser minimizada o eliminada. Cuando se realiza en un circuito discreto basado en una arquitectura de referencia a tierra flotante, el circuito de suave apagado puede tener varias complejidades. Cada controlador de compuerta del IGBT requiere un acoplador óptico rápido y un circuito de apagado suave con un transistor amplificador tótem pole. Esto da costo y complejidad al circuito, una red snuber es preferida que el circuito discreto de apagado suave. Para dispositivos de baja potencia los circuitos snubers pueden ser realizados con un capacitor de alta frecuencia a través del bus de DC, cerca de los IGBT's.

4.5.7. Problemas con métodos tradicionales.

Problemas con el circuito de control.

En un esquema de apagado suave el apagado simultaneo de los seis IGBT's es requerido para prevenir un posible encendido mientras el IGBT está en modo apagado suave. Si un IGBT de la parte alta es apagado suavemente mientras el otro IGBT está a la mitad de la transición de la conmutación dentro de un periodo del PWM, el IGBT puede intencionalmente ser encendido. Esto es difícil de sincronizar el apagado simultaneo usando arquitectura tradicional.

Problema de número de partes, tamaño, y costo.

Si el sistema requiere la protección, ahora los seis IGBT's, incluyendo protección contra falla de tierra, y si está basada en la arquitectura de referencia a tierra flotante, entonces el controlador de la compuerta y las demandas del circuito de protección requiere de muchos componentes, esto sin la característica de apagado suave:

6 acopladores ópticos rápidos.

2 sensores de efecto de campo rápidos.

2 comparadores.

4 fuentes de alimentación de 15 voltios.

Los sensores de efecto de campo son grandes y estorbosos estos requieren más espacio que un circuito integrado. Además, si en el sistema es necesaria la función de apagado suave, son requeridos seis acopladores ópticos adicionales y seis circuitos amplificadores. Esto para sistemas de baja potencia de AC.

El costo total, incluyendo el ensamble del sistema inversor, es grande debido al tamaño y el número de los componentes.

4.5.8. IR2137: Controlador para compuerta de tres fases con protección para IGBT integrado.

Se ha introducido ahora por parte de International Rectifier el IR2137 (U2 en el apéndice B) al mercado. Este circuito integrado de alta tensión provee una solución al manejo de las seis compuertas de los IGBT's, mientras que simultáneamente suministra toda la protección de los IGBT's y el apagado suave, en contraste con la arquitectura de referencia a tierra flotante, el circuito IR2137 habilita una estructura simple. La diferencia entre ambas arquitecturas es resumida en la tabla 4.2.

	Sistema de tierra flotante	Solución con IR2137
Manejo de la compuerta	Seis acopladores ópticos rápidos	IR2137
Protección del IGBT	Dos sensores de efecto de campo y dos comparadores.	Función integrada al IR2137 con un resistor de cierre y un integrador.
Circuitos adicionales	Cuatro fuentes de tensión flotantes.	Eliminado
	Circuito snuber circuito discreto de apagado suave	Eliminados por la integración de la función de apagado suave
	Circuito manejador para IGBT con un aislador óptico.	Eliminado por la integración del circuito manejador del IGBT del freno.

Tabla 4.2: Comparación de las soluciones del manejo de la compuerta protecciones del IGBT.

Solución con el circuito IR2137, como se ve en la figura 4.11, el IR2137 y el microcontrolador.

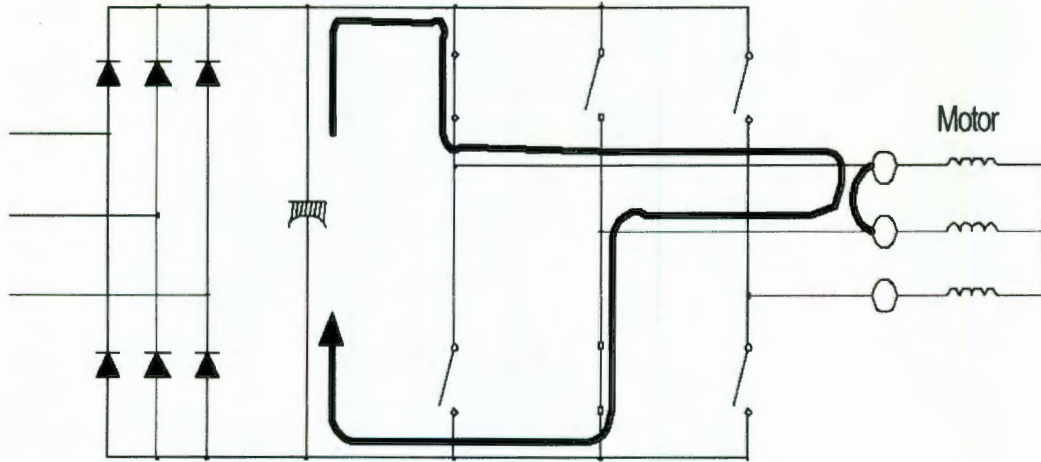


Figura 4.7: Flujo de corriente durante un cortocircuito de línea a línea.

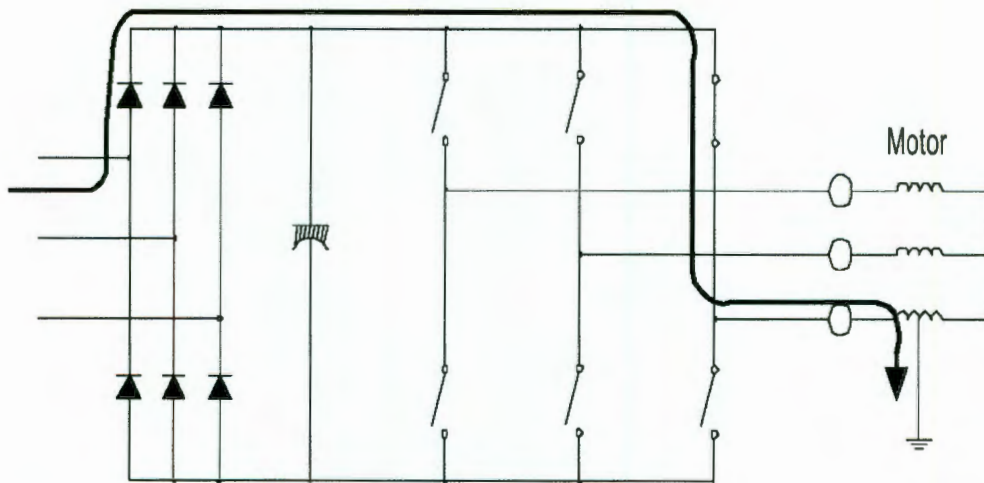


Figura 4.8: Flujo de corriente durante una falla a tierra.

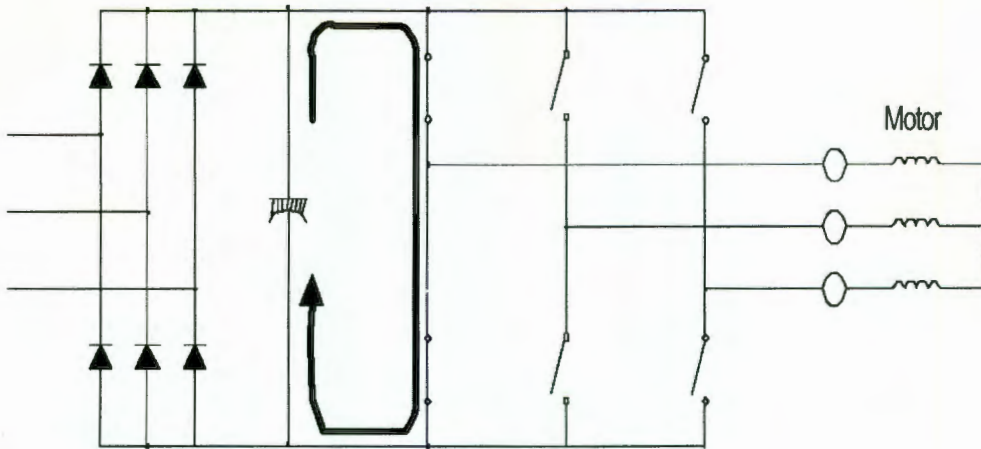


Figura 4.9: Flujo de corriente durante la saturación de dos transistores en serie.

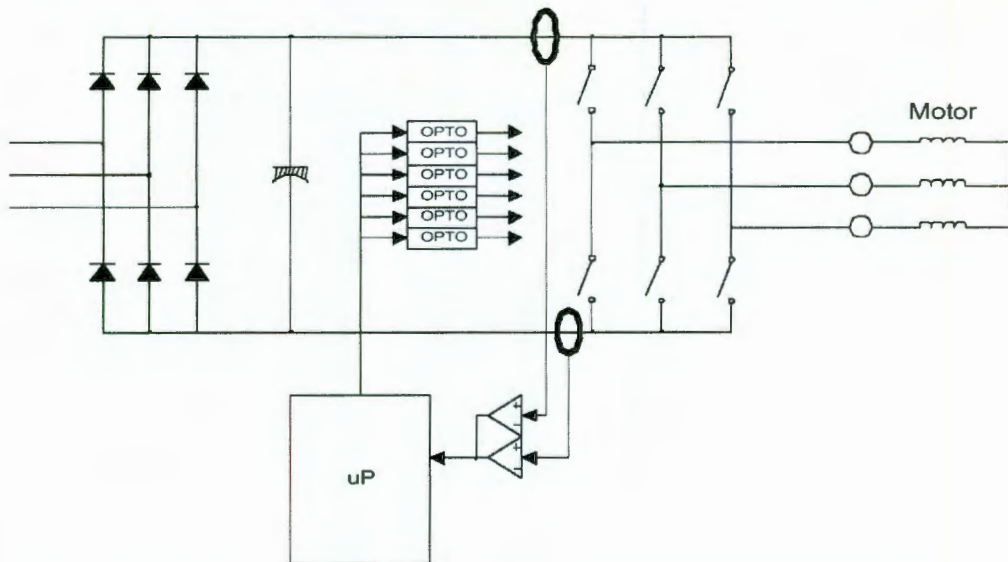


Figura 4.10: Protección de sobre-corriente en un sistema de tierra flotante.

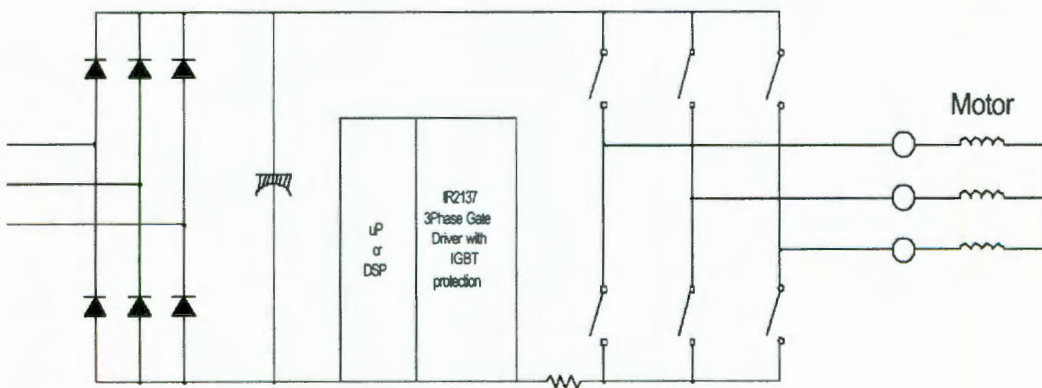


Figura 4.11: Protección contra sobre-corriente usando el IR2137.

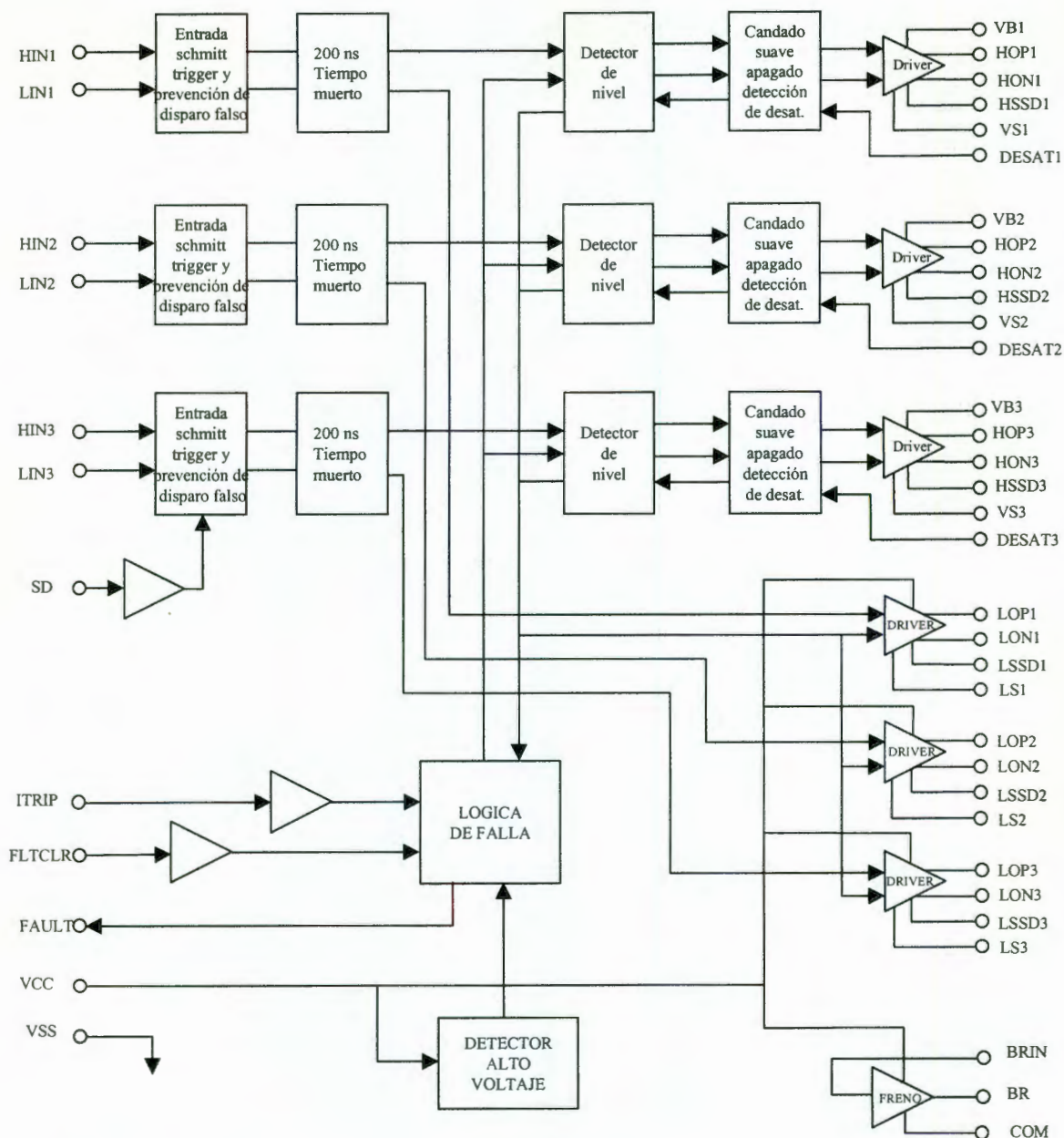


Figura 4.12: Diagrama a bloques del IR2137.

4.6. Adquisición de velocidad y posición.

En el sistema es necesario obtener la posición y la velocidad del motor para poder controlar al mismo, para lo cual se dispone de un encoder incremental el cual está integrado con el motor, El encoder genera dos señales de pulsos cuadrados, desfasadas entre sí para determinar la dirección, en las cuales la frecuencia determina la velocidad y la cantidad de pulsos la posición. Pero esa señal no puede ser recibida directamente por el

microcontrolador, por lo cual esta señal es procesada por un dispositivo lógico programable (GAL16V8D U7 en el apéndice C), generando dos señales de pulsos que son cuenta hacia arriba y cuenta hacia abajo como se muestra en la figura 4.13.

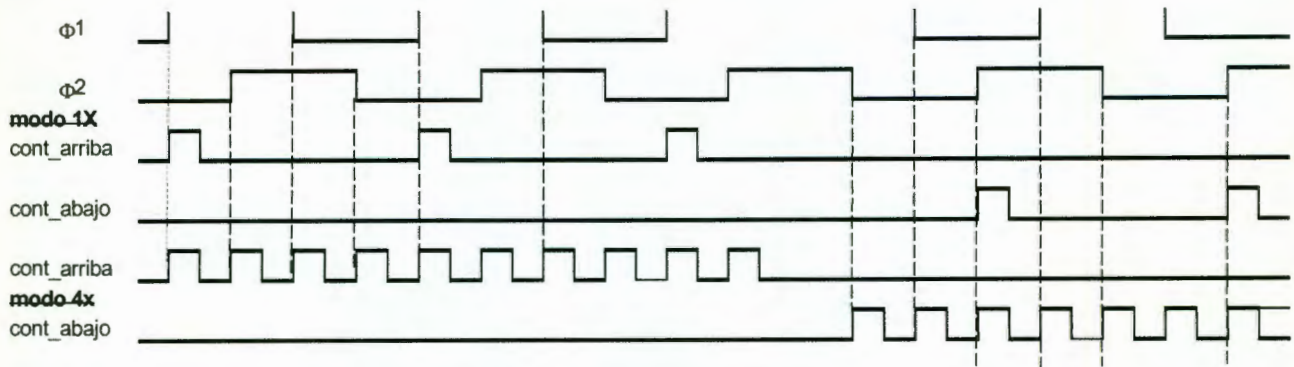


Figura 4.13: Señales del encoder secuencial.

El controlador leerá estas dos señales en dos contadores cada tiempo de muestreo y la diferencia de ambos será la distancia recorrida por el motor, y la posición actual del motor se obtiene sumando la posición previa con la posición incrementada. Ambos contadores del microcontrolador son de 16 bits, por lo cual se debe cuidar que no excedan la cuenta de 32767 y evitar el sobre flujo de los contadores, si se tiene un tiempo de muestreo de 1 ms. La frecuencia de los encoder no debe ser mayor a 3.2767 Mhz., en otras palabras si el encoder del motor es de 1000 pulsos por revolución, nuestra velocidad máxima es de 3200 revoluciones por segundo.

La posición se mantendrá en 24 bits y se incrementa cuando gira en el sentido de las manecillas del reloj y decrementa en sentido contrario a las manecillas del reloj.

La figura 4.14 se representa el circuito que adapta las señales del encoder secuencial a los dos contadores internos del procesador. En este circuito no hay posibilidad de falla por retardos y zonas de incertidumbre, debido a la naturaleza del circuito, la primera etapa es de sincronización en donde las señales del encoder se hacen coincidir con el reloj (U9 del apéndice C), cuya frecuencia determina el ancho del pulso de entrada a los contadores (250 ns.), estas señales sincronizadas son manipuladas por un circuito combinacional para generar los pulsos de salida. Este circuito combinacional está constituido por compuertas AND y OR, pero las entradas de cada una de ellas sólo puede cambiar una a la vez, evitando zonas de incertidumbre. En el peor de los casos habrá un

retardo de 2.5 ns, por el dispositivo que se está usando, el cual es insignificante para el control. La entrada X4 sólo está permitido cambiarla de estado cuando el sistema está deshabilitado.

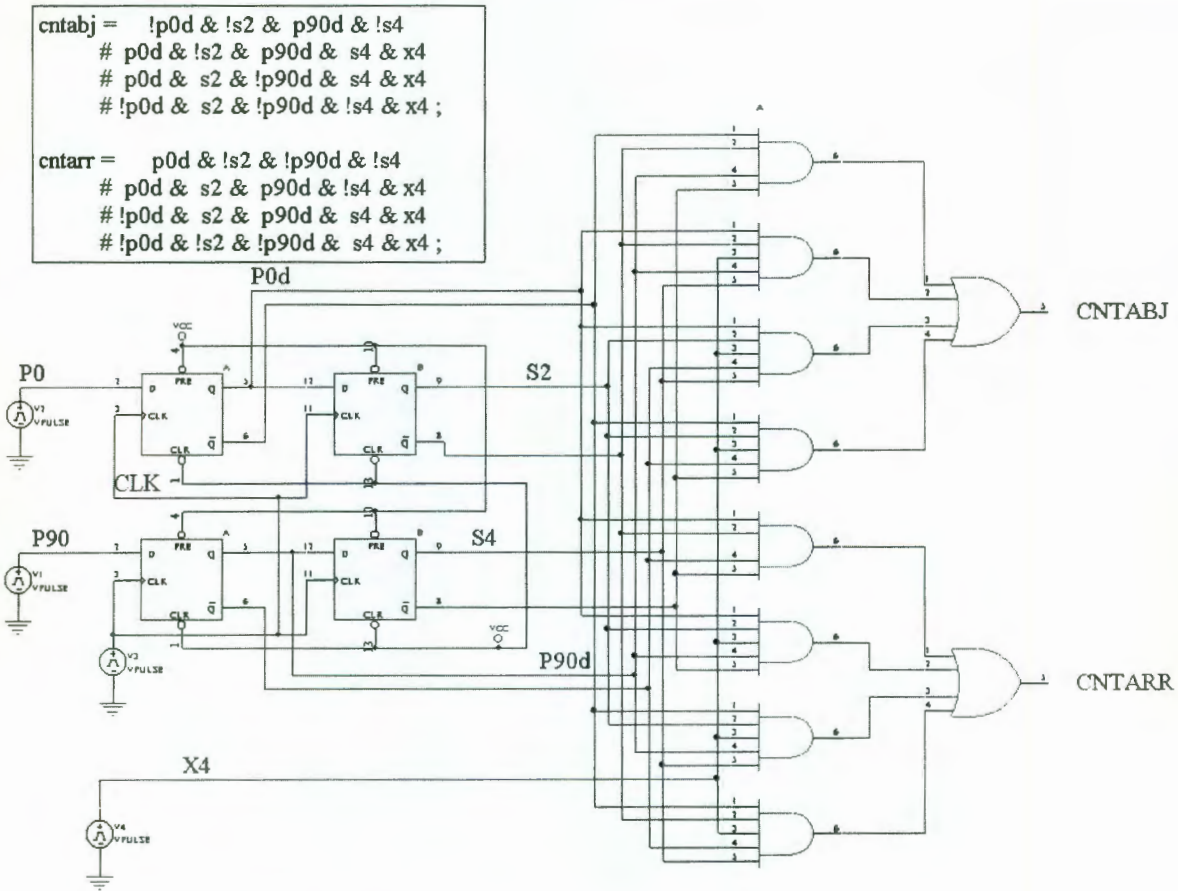


Figura 4.14: Sincronizador y decodificador del encoder secuencial.

4.7. Regeneración y descarga de potencia.

Igual que otras máquinas de imán permanente, los servomotores sin escobillas de CD actúan como un generador; cuando la flecha es empujada mecánicamente. Esto significa que la energía entregada por la inercia de la carga durante la aceleración es regresada al amplificador durante la desaceleración. Se tiene ahora regeneración de corriente que se regresa al capacitor de la fuente, donde ésta tiene el efecto de incrementar el potencial de la fuente. El circuito para la descarga de potencia se presenta en el apéndice B, el dispositivo U1 es un generador de PWM. D2 y C2 generan una tensión de referencia igual a la tensión de la fuente de alimentación de AC y es comparada con la tensión de la fuente directa al puente de IGBT's. En condiciones normales la salida de U1 es 0%, pero

cuando la tensión de la fuente de potencia es mayor al de referencia el PWM se incrementa proporcionalmente a esta diferencia. Esta señal actuará directamente sobre un transistor que conectará una resistencia de 33 ohms en la fuente de alimentación, forzando a disminuir el potencial de la fuente de potencia, logrando de esta manera disipar la potencia regenerada.

4.8. Generación de trayectoria.

Un algoritmo de generación de trayectoria, figura 4.15, es necesario para un óptimo control del motor. Una trayectoria lineal del motor se realizará. Para un movimiento de posición, la velocidad se incrementará por una aceleración constante hasta la velocidad máxima especificada. La velocidad máxima es mantenida por cierto tiempo y después es disminuida con la misma aceleración (desaceleración) hasta la velocidad de cero. La trayectoria de velocidad tendrá una forma de trapecio y si el movimiento es corto la trayectoria tendrá una forma de triángulo. Una rutina de servicio a interrupción procesa el servocontrol y hace los cálculos de generación de trayectoria, mientras el programa principal se usa para realizar la interfaz con el usuario mediante comunicación serial. La rutina de servicio a interrupción tendrá una estructura simple. El orden para ejecutar el servocontrol se necesita leer el encoder, calcular el nuevo punto de trayectoria y el valor del PID, y habilitar el PWM, todo en un intervalo constante predefinido.

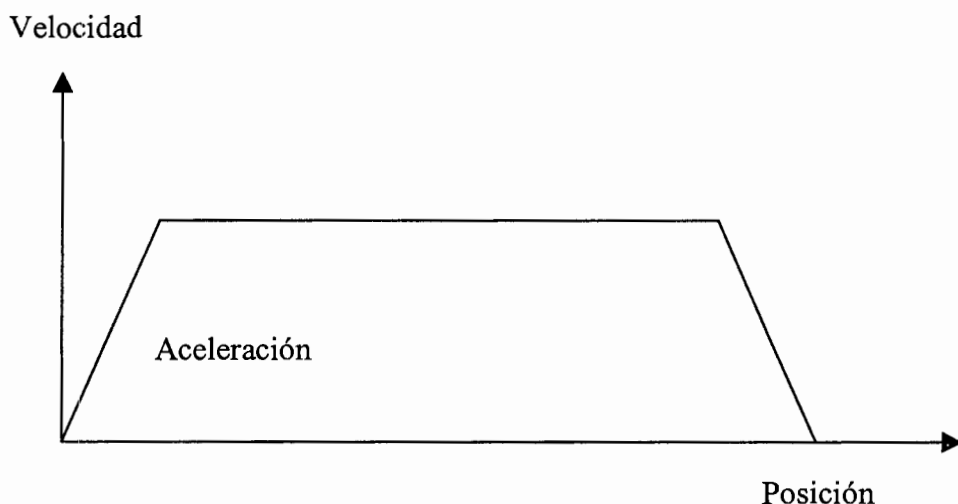


Figura 4.15: Perfil de movimiento.

4.8.1. Premovimiento.

Esta rutina es ejecutada una vez al inicio de cada movimiento. Primero algunas variables y banderas son inicializados y se ejecuta una prueba para el modo de movimiento. En modo de posición el movimiento mínimo es triangular y consiste en dos pasos:

Por lo tanto si $\text{abs}(\text{VALMOV}) > 2$, un movimiento inmediato es ejecutado. En otro caso la generación de un movimiento normal es posible con el signo del movimiento en SIGNMOV y los límites de velocidad y aceleración con signo apropiado en V y en A y $\text{VALMOV} / 2$ en VALMOV.M .

En modo velocidad el signo del movimiento es calculado en SIGNMOV , la señal de velocidad con el signo apropiado y límites de aceleración son fijados en V y A .

Finalmente, en **ModoListo** a VALMOV se da signo para aritmética de alta precisión y el algoritmo del servocontrol es habilitado.

En modo torque, VALMOV es salida directamente al PWM y el servocontrol es deshabilitado y la rutina movimiento no es ejecutada.

4.8.2. Movimiento.

La generación de movimiento está basado en un fragmento de modelo de constante de aceleración. Durante una aceleración constante, se tiene las ecuaciones para posición y velocidad que son:

$$X(t) = X_0 + V_0 t + \frac{at^2}{2} \quad V(t) = V_0 + at$$

Con unidades de t en tiempos de muestreo, el tiempo de incremento entre el tiempo de muestreo subsiguiente es 1, produciendo la ecuación interactiva, para actualizar la posición y velocidad y está realizada en **hVelPos**, dada por:

$$P(k) = P(k-1) + V(k-1) + \frac{A}{2} \quad V(k) = V(k-1) + A$$

Donde A es el límite de aceleración calculada con signo en **Premovimiento**, la ecuación inversa de esta interacción es necesaria para desacelerar el motor y detenerlo, esta rutina se encuentra en **dVelPos** y dada por:

$$P(k-1) = P(k) - V(k-1) - \frac{A}{2} \quad V(k-1) = V(k) - A$$

En el modo de posición, la forma actual de perfil de velocidad, depende de los valores de V , A y el tamaño de movimiento. Si cada uno de los límites de velocidad es alcanzado antes de que medio movimiento sea completado, resulta un perfil de velocidad trapezoide, (figura 4.16) o si la mitad de movimiento es alcanzado antes que haya llegado a su límite de velocidad, resulta un perfil de velocidad triangular, (figura 4.17).

En el algoritmo empleado aquí, el límite de velocidad sigue al límite de velocidad actual, con eso se permite exactamente el mismo número de pasos durante las secciones de aumento de velocidad (**Aumenta Velocidad**) y decremento de velocidad (**Baja Velocidad**). La **fase1** está definida como una sección de movimiento donde la orden de posición es menor que la mitad de movimiento y la **fase2** es la parte sobrante de movimiento. $T1$ es el tiempo cuando el límite de velocidad actual es alcanzado y $T2$ es el tiempo de fin de la fase1.

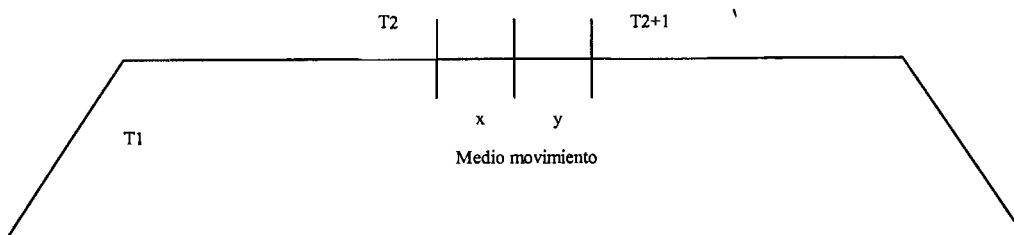


Figura 4.16: Perfil de velocidad para movimientos trapezoidales.



Figura 4.17: Perfil de velocidad para movimientos triangulares.

Además, si X es la cantidad de bajo tiro y Y la cantidad de sobre tiro de medio movimiento en $T2$, el error por discretización es minimizado por el uso de los valores de X y Y , si uno pesa más, reducirá el tamaño del movimiento inmediato final durante el último paso de ese movimiento. Para un movimiento triangular, la discretización de error está dada

por min. $(2X, 2Y)$, resultando la condición que si $2X > 2Y$, entonces se hace un paso más de aumento de velocidad (**AumentaVelocidad**). En el caso de un movimiento trapezoide, el error por discretización está dado por min. $(2X, X-Y)$, produciendo la condición que si $(3X > Y)$, se hace un paso más durante la sección plana de la fase2 (**Plana**).

En el inicio de **Movimiento**, TIEMPOMOV es incrementado y **hVelPos** es llamado para la evaluación del próximo valor propuesto por la orden de posición y velocidad bajo el nivel actual de la A. En el modo de posición, en **fase1**, la posición original más la mitad de movimiento menos la nueva orden de posición propuesto es calculado y fijado en MOVDEL con el previo MOVDEL salvado en MOVTMP. Como la mitad de movimiento podría ser pasado $MOVTMP = -X$ y $MOVDEL = Y$, con $Y < 0$ para el primer tiempo indicando que la **fase1** está cerca de ser completada. Por lo tanto si $Y < 0$, se continúa en la **fase1**, donde si la máxima velocidad no ha sido alcanzada, la nueva orden de posición propuesta es ejecutada. Por otra parte, cuando el movimiento propuesto exceda la velocidad máxima, se deshace el movimiento propuesto, se pondrá la aceleración en cero y se reevalúa la ecuación interactiva; con la nueva aceleración, se pondrá $T1 = TIEMPOMOV - 1$, y se ejecuta el movimiento.

Entonces la variable T1 es limpiada en **PreMovimiento**, ésta es usada como una bandera para indicar si la esquina del perfil de velocidad ha sido alcanzada, entonces se encontrará que $Y > 0$, se avanzará en el código, esto será ejecutado sólo una vez, con la fase2 se inicia el próximo paso. Si $T1 = 0$, la máxima velocidad no ha sido alcanzada aún, como $T1 = T2$ y el perfil de velocidad es triangular, en este caso, A es negada para bajar la velocidad y si $X > Y$, un paso más es necesario para minimizar el error por discretización. Como A es negado, el paso propuesto se deshizo, A es nuevamente negado para bajar la velocidad y el paso es recalculado y ejecutado con $T2 = T1 = TIEMPOMOV - 1$.

Si T1 no es cero, indica que se está en la sección plana de la fase 1, entonces se va a **t2net1**, donde $T2 = TIEMPOMOV - 1$ y si $3X > Y$, entonces un paso más en la parte plana de la fase2 es necesario para minimizar el error por discretización, FAS2PLANA es definida como el número de pasos en la sección plana de la fase2 y es usada como un contador mientras ésta es completada. SI $3X > Y$, entonces $FAS2PLANA = T2 - T1$, de otra manera $FAS2PLANA = T2 - T1 - 1$ y la fase1 es finalmente completada. Todos los subsecuentes pasos procederán a través de la fase2. Primero se decidirá si la sección plana

está terminada probando si PH2FLAT alcanza cero, si no, se va al plano donde PH2FLAT es disminuida y probada si es cero, si también en la sección de bajar la velocidad es iniciada calculando una señal apropiada de aceleración signada y ejecutando el último movimiento de la sección plana. Para todos los siguientes pasos, FAS2PLANA=0, permite el final de la prueba con la orden de velocidad cero, esto indica el final de movimiento. Esto siempre ocurrirá desde la velocidad máxima actual, ligado abajo por el límite dado por el usuario, el cual es siempre un múltiplo entero de límite de aceleración dado por el usuario, con exactamente el mismo número de pasos hechos durante el incremento y decremento de velocidad.

En el modo velocidad, sólo se ejecutará un movimiento indefinido, con un perfil de velocidad en forma de rampa. Si la velocidad final no ha sido alcanzada, el movimiento continúa a su máxima aceleración. Si su máxima velocidad ha sido alcanzada la aceleración se pone en cero y se genera un movimiento se ordena posición y velocidad continuamente hasta que la velocidad ordenada sea cero.

4.9. El compensador

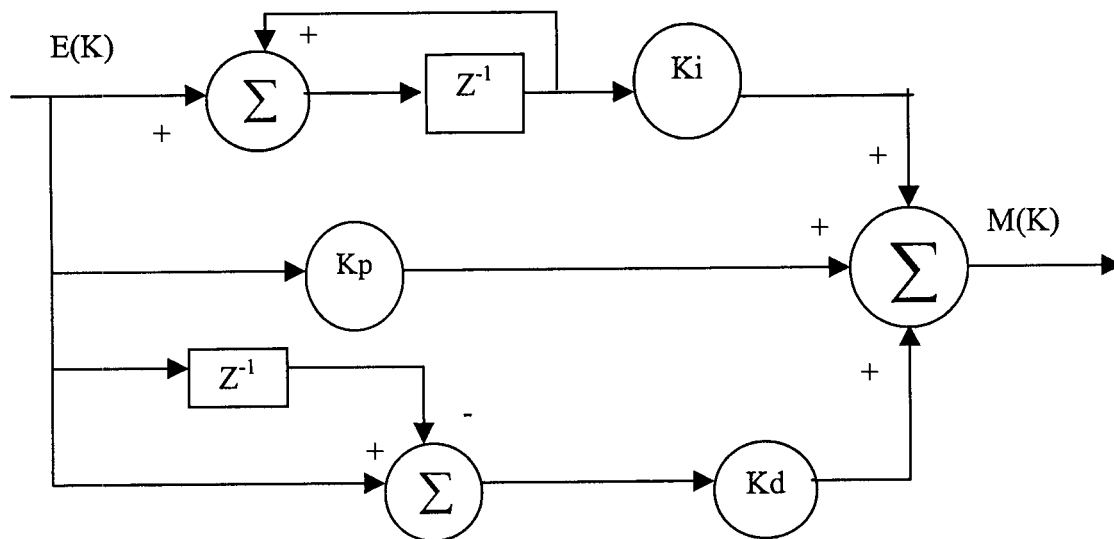


Figura 4.18: Elaboración del PID digital.

El controlador PID discreto es el algoritmo más usado para el control de motores por su fácil manejo y sintonización. En la figura 4.18 se presenta un algoritmo de un PID digital común, es la realización de la ecuación (3-43), en donde $E(k)$ es el error de posición

y $M(k)$ es la salida. Este algoritmo a sido realizado en un PIC17C42 (U2 apéndice B). Sólo con 800 instrucciones, la ejecución del PID se logra con un tiempo de 0.2 ms. con un cristal de 16 MHz. Aplicando algunas consideraciones de robustez que sugiere (Aguado, A 2000) el sistema tiene el sistema es realizable y con buenas características.

4.10. Instrucciones de interfaz.

El sistema tiene previsto una interfaz hombre máquina, la cual está constituida por una comunicación serial (RS232), a una velocidad de 19200 bits por segundo, entre el servomecanismo (U5 y U6 del apéndice C) y una computadora personal (PC), en la cual se desarrollo un programa que maneja uno de sus puertos serie. Cuando el servomecanismo recibe por su puerto serie un caracter y éste está dentro de su base de datos es reconocido como una instrucción y ejecuta una función específica.

Las siguientes instrucciones están realizadas y son reconocidas por la interfaz del usuario.

Deshabilitar d

Deshabilita el amplificador. Al ejecutar esta instrucción el amplificador no reproduce las señales que vienen del control, simplemente las ignora y el motor no se moverá ante la ejecución de una instrucción **M**.

Habilitar h

Habilita el amplificador. Inicialmente el amplificador está deshabilitado, al ejecutar la instrucción **h** el amplificador comienza a ejecutar las señales de control, provenientes del controlador.

Movimiento M[-8,388,608 a 8,388,607]

Comanda al eje para moverse a una nueva posición o velocidad, el dato de posición es relativo y el dato de velocidad es absoluto, el dato de posición es en pulsos del encoder, el dato de velocidad está dado en pulsos del encoder por muestreo multiplicado por 256, todos los movimientos serán ejecutados por el procesador respetando los parámetros almacenados en la memoria. Todos los movimientos ordenados son almacenados en una variable FIFO (primero en entrar primero en salir), las instrucciones en

la variable son ejecutadas tan pronto como la ejecución anterior sea completada. Si no existe una instrucción M en ejecución el movimiento iniciará inmediatamente.

Modo de operación O[P,V,T]

Si el argumento es P causará que el siguiente movimiento sea en modo normal es decir el movimiento se genera en modo trapezoide o triangular. Un argumento V causará que el movimiento sea de modo indefinido acelerando de cero a la velocidad programada por la instrucción M. En un argumento T, el movimiento será en modo manual, en donde el valor escrito con la instrucción M se transferirá directamente sobre el amplificador del motor.

Programación de Parámetros S[00 a FF,-8,388,608 a 8,388,607]

Envía los parámetros al controlador dándoles un valor, los parámetros aceptados son los presentados en la tabla 1.

PARAMETROS	NÚMERO EN HEX	INTERVALO
LIMITE DE VELOCIDAD	00	0 a 8,833,607
ACELERACION	01	0 a 8,688,607
KP: GANANCIA PROPORCIONAL	02	-32,768 a 32,767
KD: GANANCIA DIFERENCIAL	03	-32,768 a 32,767
KI GANANCIA INTEGRAL	04	-32,768 a 32,767

Tabla 4.3: Parámetros.

Lectura de parámetros R[00 a FF]

Regresa el valor actual de los parámetros.

Tiempo C

Regresa el tiempo desde que inicia el presente movimiento hasta la ejecución de esta instrucción y captura la velocidad y la posición al tiempo de ejecución de esta instrucción.

Lectura de la posición ordenada P

Regresa la posición creada por la rutina de generación de trayectoria, la cual fue capturada en la ejecución de la instrucción C. El dato entregado es la posición deseada que el motor tenía cuando éste fue capturado.

Lectura de la velocidad ordenada V

Regresa la velocidad creada por la rutina de generación de trayectoria, capturada durante la ejecución de la instrucción C. El dato entregado es la velocidad deseada que el motor tenía cuando éste fue capturado.

Lectura de la posición actual. p

Regresa la posición actual del contador de encoder, la cual fue capturada durante la ejecución de la instrucción C. El dato entregado es la posición real que el motor tenía cuando éste fue capturado.

Lectura de la velocidad actual v

Regresa la velocidad actual multiplicada por 256, la cual fue capturada durante la ejecución de la instrucción C. El dato entregado es la velocidad real que el motor tenía cuando éste fue capturado.

Estado externo X

Regresa dos dígitos en hexadecimal los cuales definen el estado externo del controlador. Esta instrucción limpiará todos los bits del registro de estado externo que se presentan en la tabla 2.

Bit 7	Índice marcado detectado
Bit 6	Límite positivo encontrado
Bit 5	Límite negativo encontrado

Bit 4	Entrada correcta
Bit 3-0	No realizados

Tabla 4.4: Bits del registro del estado externo.

Modo del encoder $x[1,4]$

Cambia el modo de operación del encoder. Si se ejecuta $x1$ se tiene una cuenta por cada pulso del encoder, si se ejecuta $x4$ se tiene cuatro cuentas por cada pulso del encoder.

Mover estado Y

Regresa dos dígitos en hexadecimal el cual define el estado del movimiento. Esta instrucción limpiará todos los bits del registro del estado del movimiento de la tabla 4.5.

Bit 7	Localidad del movimiento vacío
Bit 6	Movimiento completo
Bit 5-0	No realizados

Tabla 4.5: Bits del registro del estado de movimientos.

Restablece Z

Ejecuta una reiniciación del sistema por programa. Con la ejecución de esta instrucción el sistema limpia todas las variables internas del servosistema y cargando los valores por defecto.

Inhabilitación del sistema s

Esta instrucción inhabilita el funcionamiento del servomecanismo. El sistema se habilitará con la ejecución de una instrucción M .

Captura $c[0,4]$

Activa la función de captura de datos en donde el número que sigue de la instrucción c significa la variable a ser capturada como se indica en la tabla 4.6.

Si se trata de ejecutar una instrucción desconocida por el servosistema éste responde con el carácter ¿ , lo cual significa que hubo un error en la transmisión o que la

instrucción no existe. Si la transmisión y la instrucción son correctas el servosistema regresara la misma instrucción seguida de un carácter ;, lo cual significa que la instrucción se ejecuto satisfactoriamente.

Número de variable	Variable (16 bits)	Descripción
0	Ninguno	Desactiva la instrucción captura.
1	POSICION	Captura y envía la variable que contiene la posición ordenada.
2	VELOCIDAD	Captura y envía la variable que contiene la velocidad ordenada.
3	MPOSICION	Captura y envía la variable que contiene la posición medida del encoder.
4	MVELOCIDAD	Captura y envía la variable que contiene la velocidad medida del encoder.

Tabla 4.6: Variables capturadas por la instrucción c.

5. RESULTADOS.

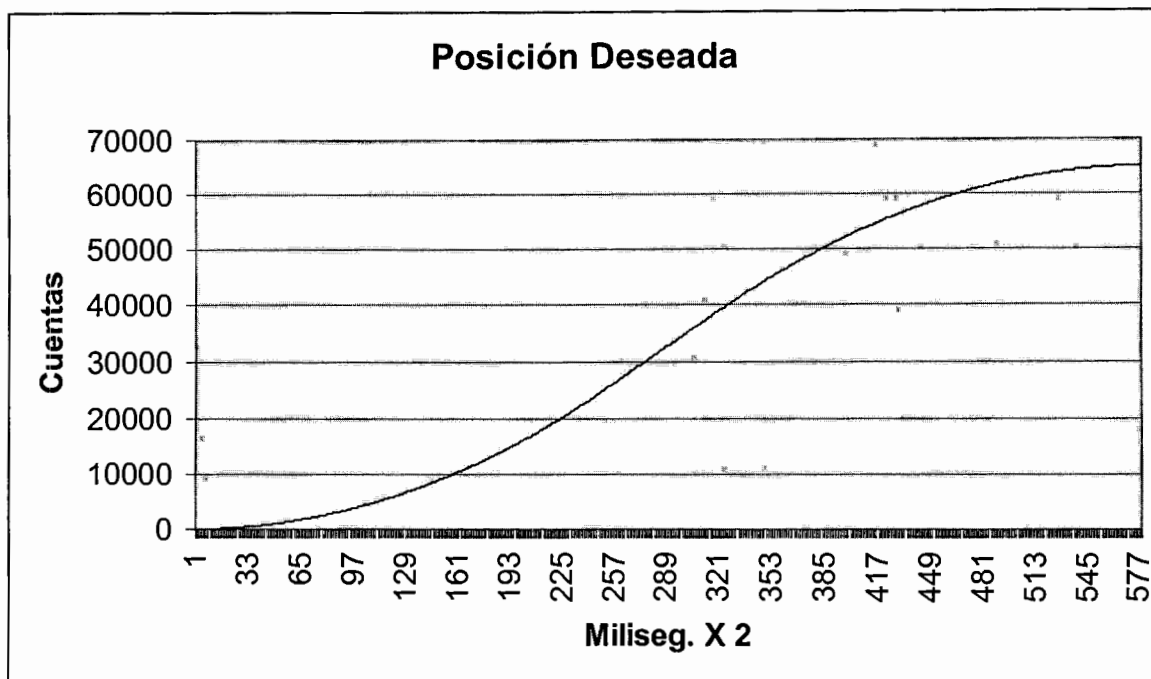
Como no se dispone de equipo especializado para medir posición y velocidad, que esté calibrado, las lecturas se tomaron directamente del encoder del motor lo cual nos da una referencia válida para la evaluación de los resultados. En un futuro se pretende hacer estas mediciones con equipo adecuado y cotejarlos con los resultados obtenido hasta ahora.

Para obtener los resultados, se programó una función especial (ver la rutina *EnviaDatos* del apéndice F) en el servosistema. Como se tiene una velocidad de transmisión de 19200 bits por segundo, sólo es posible enviar una palabra de 16 bits en dos milisegundos, con el periodo de control es de un milisegundo se enviará sólo en los periodos de control pares. La instrucción consiste en capturar, en la interfaz (en nuestro caso una PC), una variable que contiene la información del movimiento ejecutado. Cuando la instrucción `c[0,4]` es ejecutada y un movimiento es ordenado, el servosistema enviará, durante el lapso del movimiento, la información contenida en las variables especificadas; ver tabla 4.6. Esta información enviada por el servosistema es recibida por el puerto serie de la computadora, la cual se manipula y almacena en un archivo compatible con Excel, y de esta manera se obtienen los resultados.

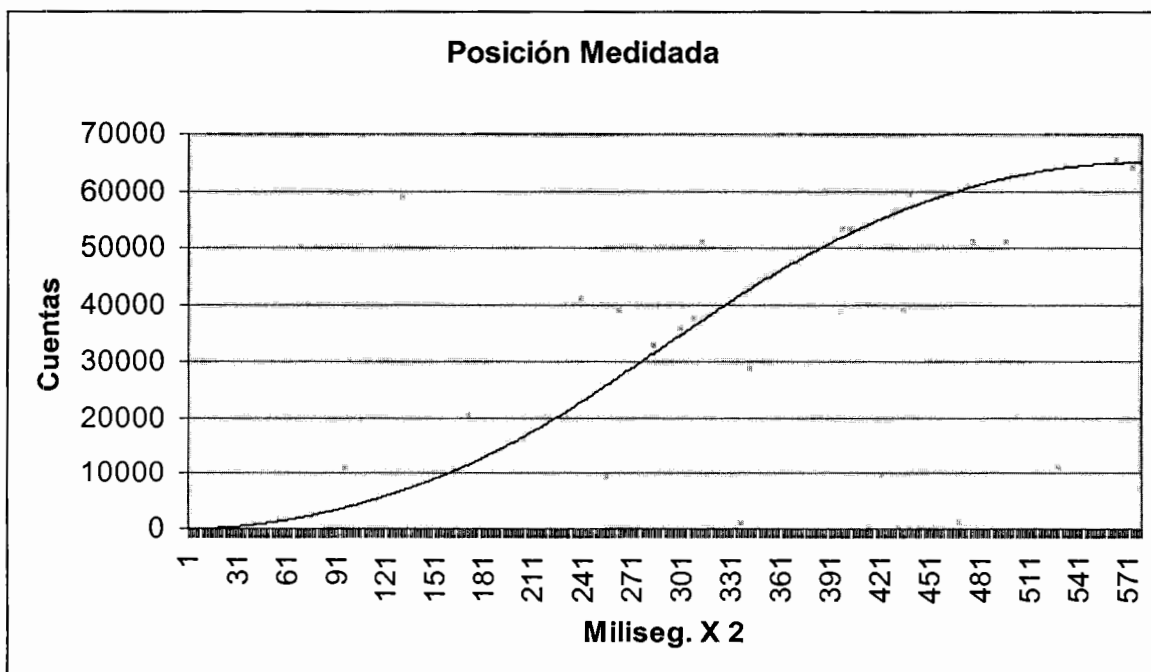
Se ejecutaron movimientos de 65,000 cuentas a una velocidad de 100 cuentas / ms a diferentes aceleraciones con constantes del PID de $K_p = 1800$, $K_d = 15600$, $K_i = 52$. Con una carga cilíndrica de 2,5 centímetros de diámetro y aproximadamente 50 gramos de peso.



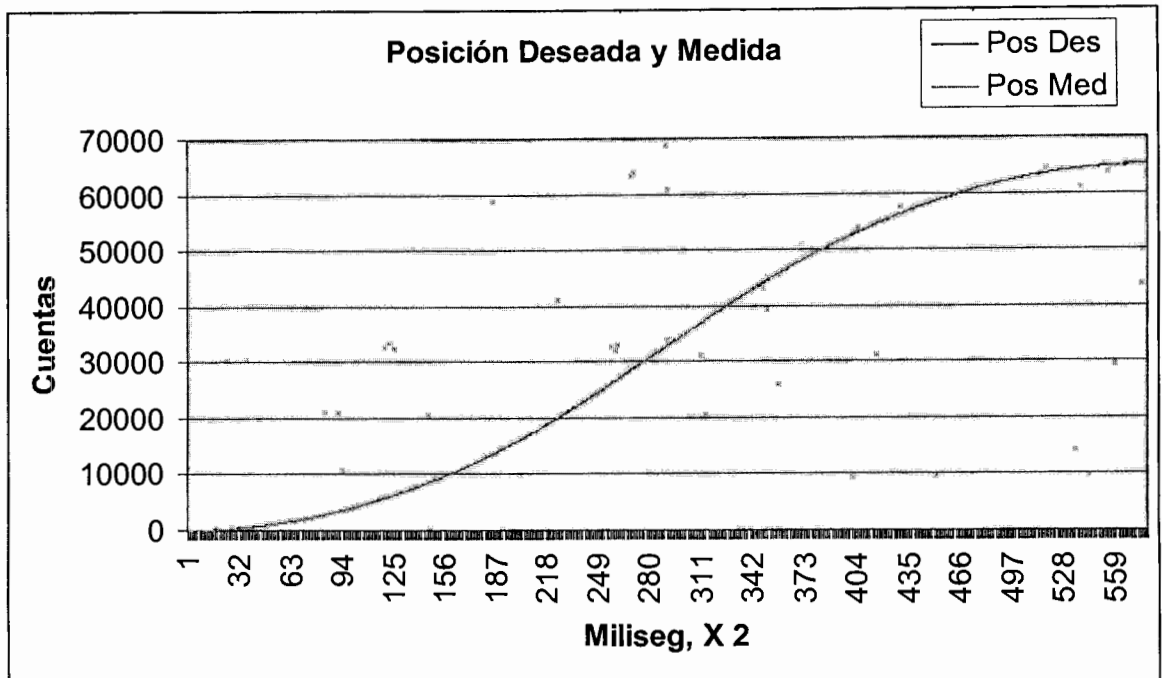
Con una aceleración de $0.2 \frac{\text{cuentas}}{\text{ms}^2}$ se obtuvieron los siguientes resultados.



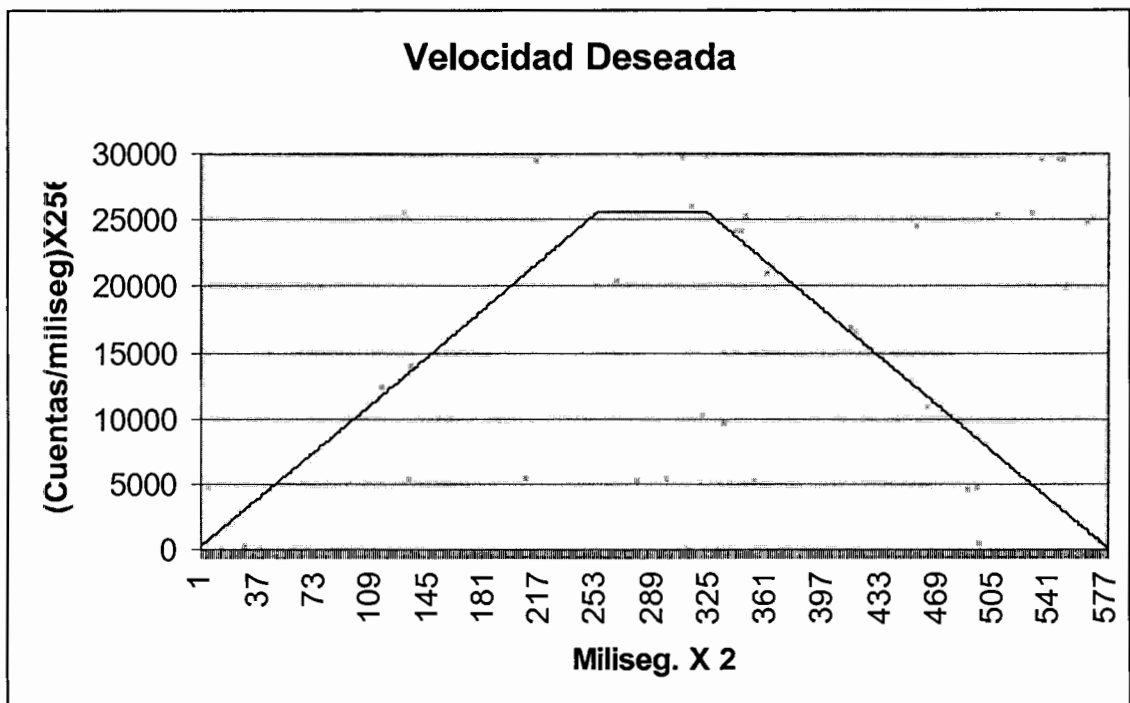
Respuesta en posición del algoritmo de generación de trayectoria.



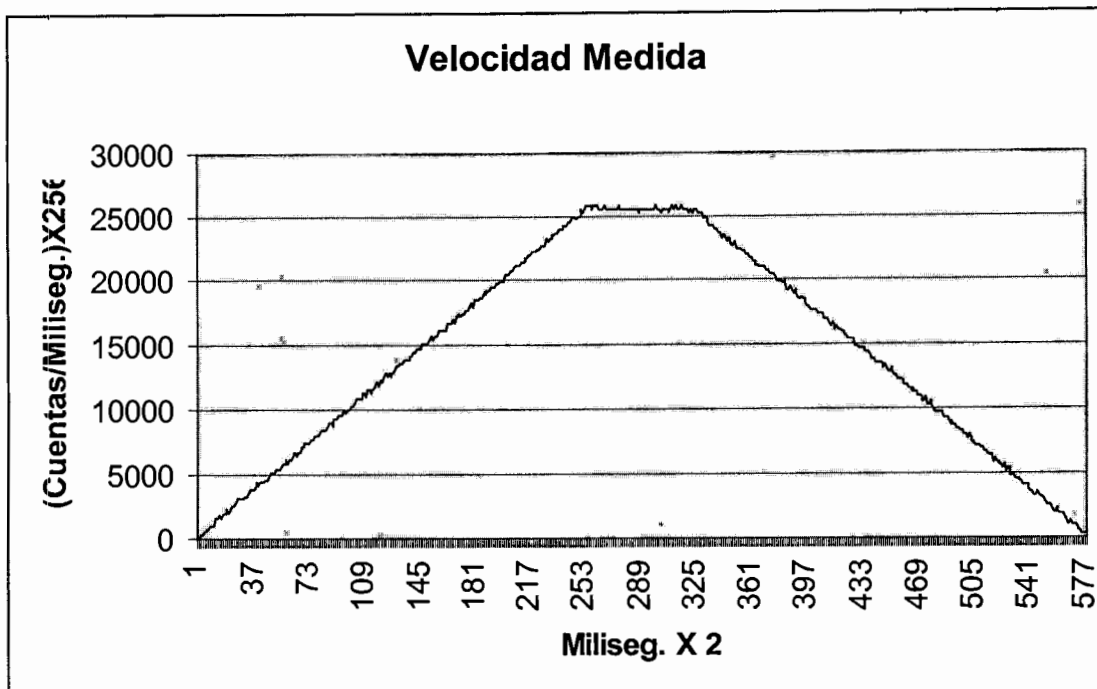
Respuesta en posición del algoritmo compensador.



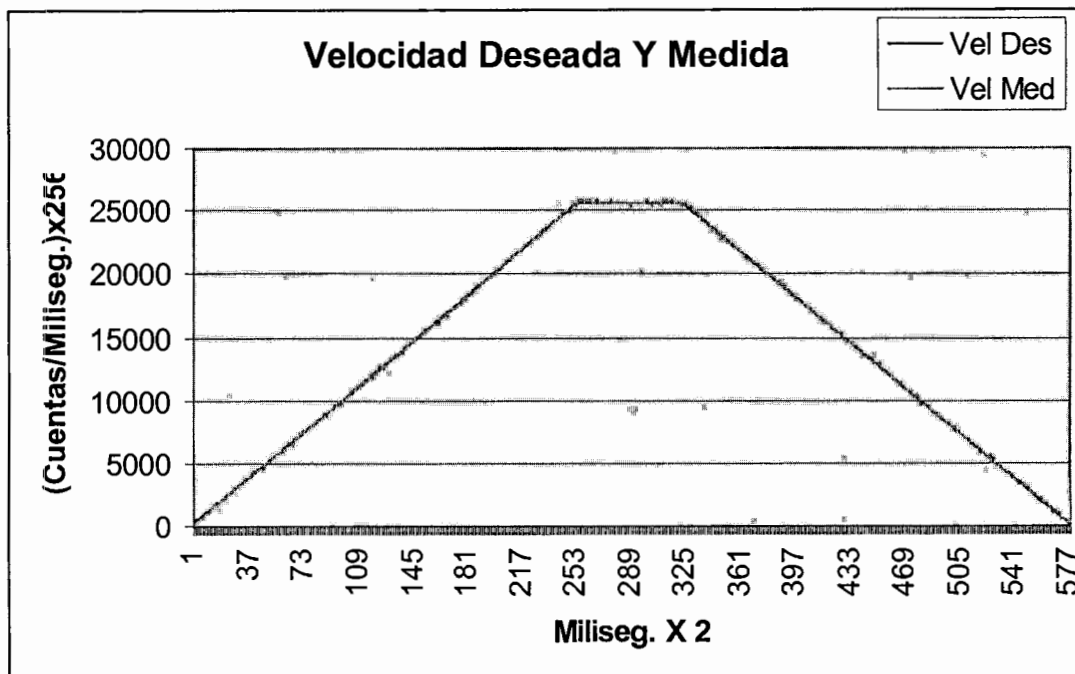
Comparación de las respuestas en posición.



Respuesta en velocidad del algoritmo de generación de trayectoria.

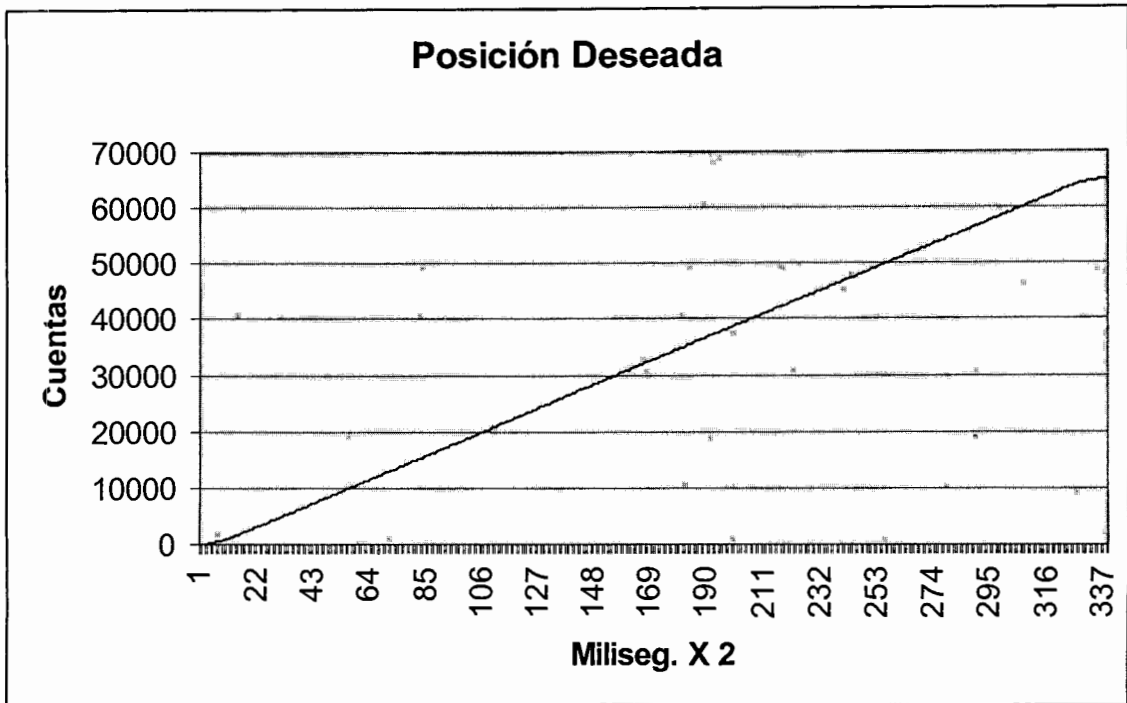


Respuesta en velocidad del algoritmo compensador.

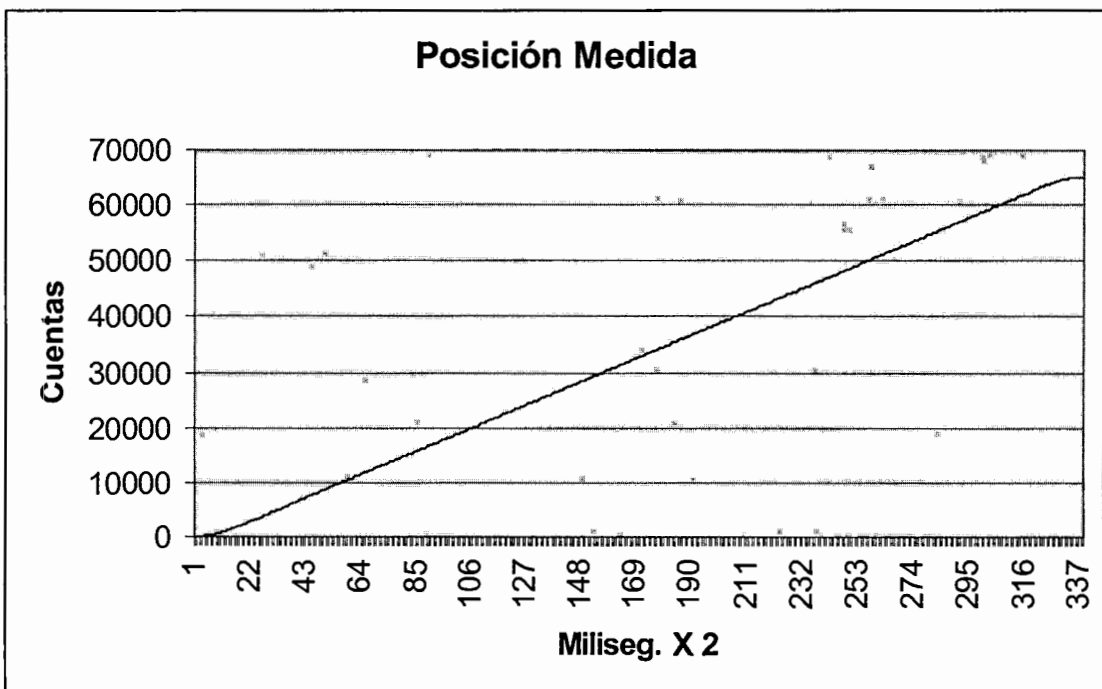


Comparación de las respuestas en velocidad.

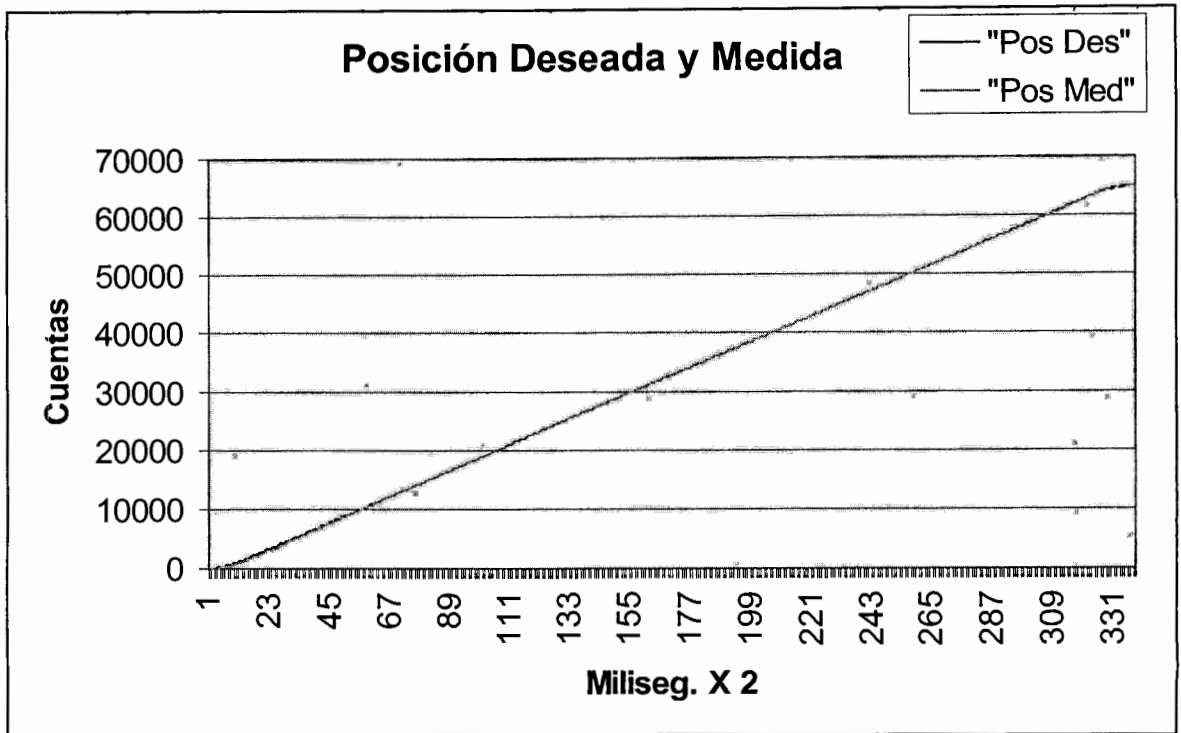
Con una aceleración de $3.125 \text{ cuentas/ms}^2$ se obtuvieron los siguientes resultados.



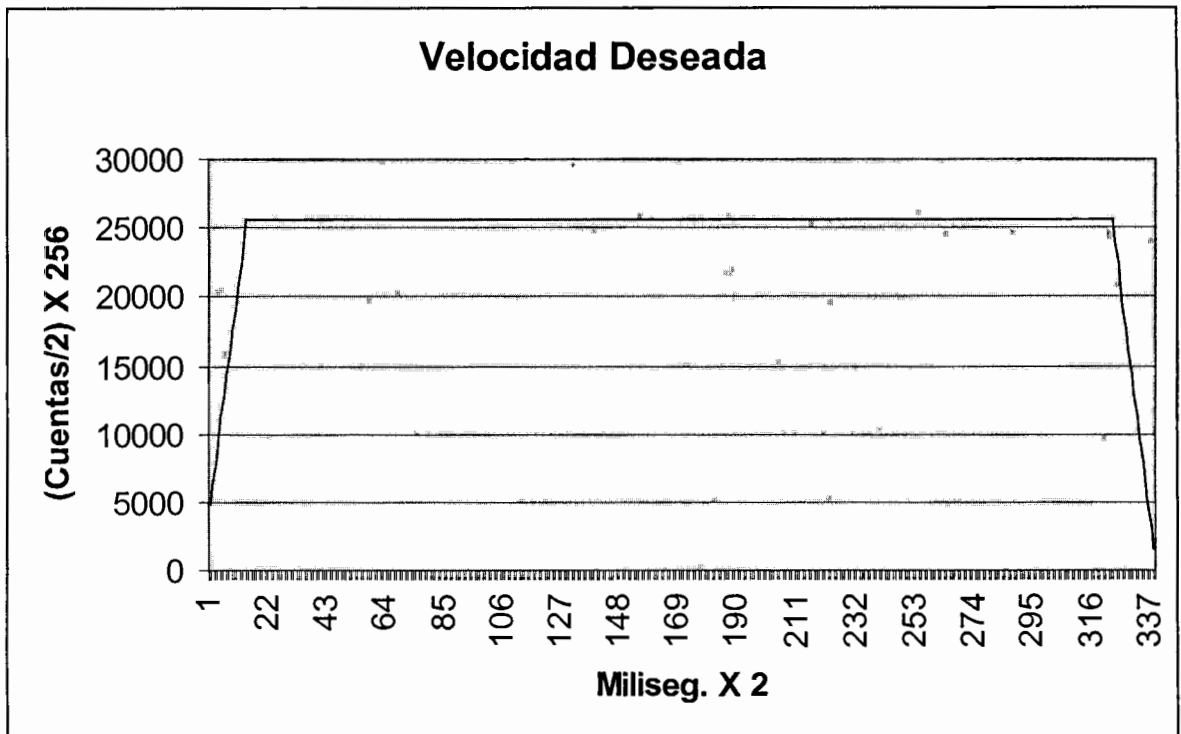
Respuesta en posición del algoritmo de generación de trayectoria.



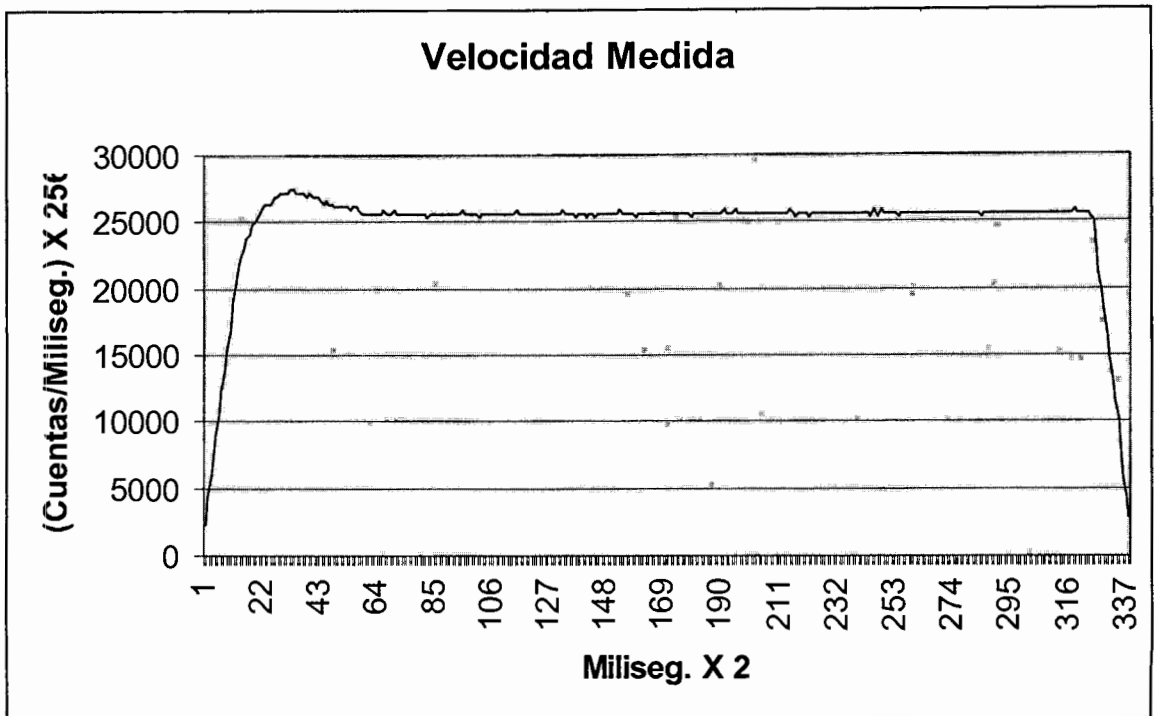
Respuesta en posición del algoritmo compensador.



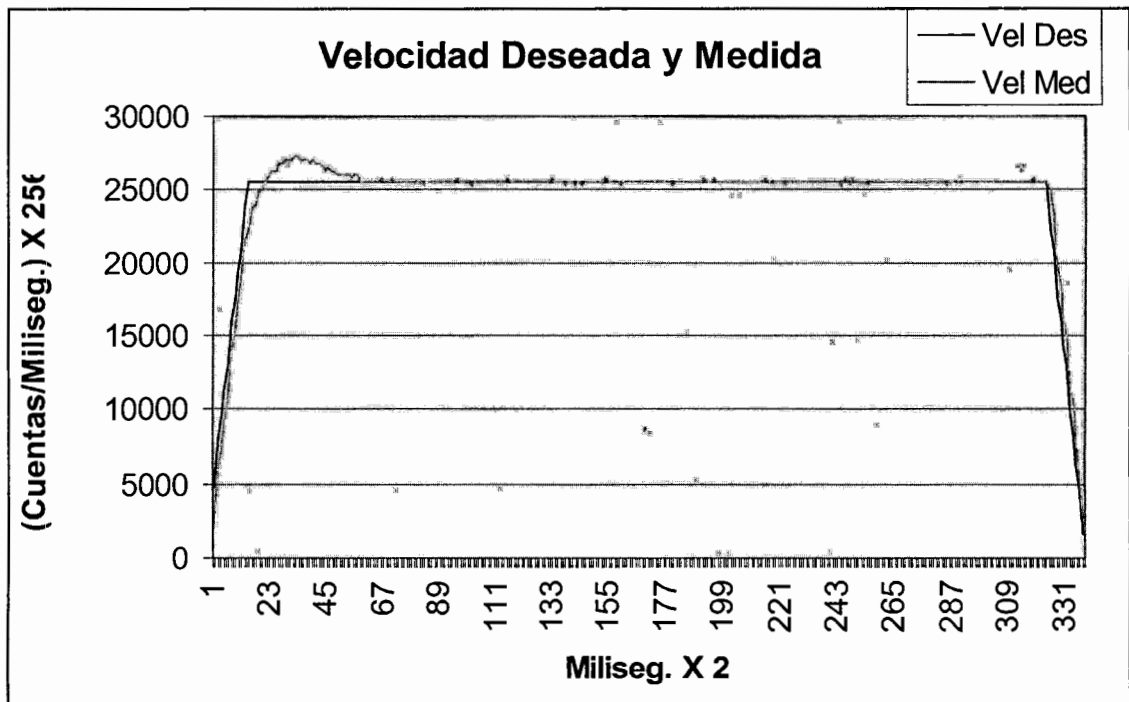
Comparación de las respuestas en posición.



Respuesta en velocidad del algoritmo de generación de trayectoria.



Respuesta en velocidad del algoritmo compensador.



Comparación de las respuestas en velocidad.

6. CONCLUSIONES.

Al final del proyecto se logró asimilar la tecnología del servocontroladores observando a detalle su comportamiento, ventajas y desventajas, teniendo el respaldo suficiente para producirlos a bajo costo así como la experiencia necesaria para decidir sobre sus aplicaciones.

Se tiene un servocontrolador compacto de pocos componentes con características similares a algunos comerciales aunque limitado en cuanto al número de funciones, pero continuando la investigación se puede seguir desarrollando para darle más versatilidad.

Usando un microcontrolador de alta integración es una solución la cual requiere pocos componentes externos lo cual se traduce a un costo efectivo, esto aunado a la alta tecnología en lógica combinacional y a manejadores de módulos de transistores de efecto de campo el sistema es compacto y de buena eficiencia.

El procesador usado tiene las características suficientes y puede perfectamente con el control pero puede usarse cualquier otro dependiendo de las demandas de la aplicación y de la disponibilidad en el mercado, esto queda abierto para investigaciones futuras.

Sin embargo, de acuerdo a los resultados alcanzados en el servocontrolador se puede utilizar en la modernización o reconversión de cualquier máquinas-herramienta con propósitos de incidir en el aumento de su producción. Mientras que la parte tecnológica y didáctica tiene la ventaja de que se puede observar a detalle el funcionamiento del controlador dando una herramienta a los maestros para la enseñanza de estos sistemas ya que actualmente existe muy poca documentación tanto teórica como práctica en servomecanismos en el ámbito universitario.

Se logró asimilar esta tecnología generando documentación completa y ahora se esta en la posibilidad de desarrollar aplicaciones a la medida, según los requerimientos de la máquina o equipo; eliminando gastos innecesarios y de esa manera reducir costos, lo cual alentará la inversión en estos sistemas y de esa manera resolver el problema de rezago tecnológico.

7. REFERENCIAS.

Aguado, A 2000. Temas de identificación y Control Adaptable. Instituto de Cibernética Matemática y Física. Habana Cuba.

Jacob Tal Ph D, 1994. Step-by-Step Design of motion control systems.

Ogata, K.1998 Ingeniería de control moderna. Universidad de Minnesota.

Ogata, K.1996 Sistemas de Control en Tiempo Discreto. Universidad de Minnesota.

Savant - Roden – Carpen Diseño Electrónico, Ctos y Sistemas. Segunda Edición. Addison-Wesley, Iberoamericana.

George C. Chryssis, 1984 HIGH FREQUENCY SWITCHING POWER SUPPLIES

Younkin G. W. 1996 Industrial Servo Control Systems Fundamentals and Applications. Inc. Fond du Lac, Wisconsin MOTION CONTROL, Noviembre/December 1991

MICROCHIP 1996 PIC16/17 MICROCONTROLLER DATA BOOK.

MICROCHIP 1994 EMBBEDED CANTROL HAND BOOK.

8. APÉNDICE.

A Publicación.

Brushless DC Digital Servo Motor Control.

Guillermo Ronquillo Lomeli*, Gilberto Herrera Ruiz.**

***Centro de Ingeniería y Desarrollo Industrial.**

Pie de la cuesta # 702 Desarrollo San Pablo.

C.P. 76130 Querétaro Qro Mexico.

****Universidad Autónoma de Querétaro.**

Cerro de las campanas S/N Centro Universitario.

C.P. 76010 Querétaro Qro. México.

Emails: autyctrl@cidesi.mx

gherrera@sunserver.uaq.mx

(Keywords: Servo Control, PWM, PID, PLD).

SUMMARY

This system is a position control with a servo motor brushless, Hall effect sensors and an incremental encoder. Which consists on an encoder, a microcontroller, a digital to analog converter, a power amplifier that operates as voltage amplifier and a serial interface; everything with modern components. In the microcontroller compensation routines (PID) are implemented, as well the generation of trapezoidal trajectories, the combination of both algorithms generates a good movement. The communication between the processor and the power amplifier is through the technique of pulse-width modulation (PWM).

1. INTRODUCTION.

Everyday industry requirements move towards an automation increase with the consequent demand for foreign goods because there is a lack of high-technology products in the national market. Due to this, the present thesis work object is to develop complete servo mechanism systems that are competitive and to provide human resources development, in this country.

Automation requires control, which involves acceleration, speed, position and torque as related variables. In order to control these variables, an electronic system which acts as a feedback loop applied to the motor under control is required.

The present development was completed for a 140 oz / in, dc brushless servo motor with field effect sensors and 3200 rpm as its maximum operating speed. The motor is controlled by an electronic servo controller.

The system is supervised by a PC via serial port to send and retrieve parameters such as acceleration, speed and position in both operation monitor and command mode. It includes a PID digital controller, to compensate the system and a trajectory generation algorithm. The system operates in position control mode. A power stage was also designed with a 3-phase Isolated Gate Bipolar (IGBT) H bridge. The power interface stage is constituted by a combinational circuit for current commutation, a high-voltage IGBT controller circuit for de-saturation, a short circuit to ground, a soft turn off in over-current conditions, and an independent phase commutation with related delays.

The system development involves IGBT's modules, Programmable Logic Device (PLD), field effect sensors and a 16 MHz microcontroller, all of them available in the national market to make them competitive with their counterparts. The system is very well suited in high-precision motion control applications especially where motor speed and/or position control is required. The system is especially devoted to automation and machine reconversion, where particular conditions like precision movement are required. It is possible due to its versatility and compatibility with digital system.

2. BASIC CONCEPT.

A 3-phases brushless servo motor 140 oz-in with hall effect sensors and an 1000 pulses encoder was available, the motor uses 3-phases and three hall effect sensors to locate the shaft position and to commute currents in motor phases. A servo driver prototype was designed and built which amplifies a voltage command.

Modern digital servo systems are formed as shown in figure 1. These systems control a motor with an incremental feedback device like incremental encoder. They consist of an encoder counter, a microcontroller, some form of D/A (Digital-to-Analog) converter, and a power amplifier, which delivers current or voltage to the motor.

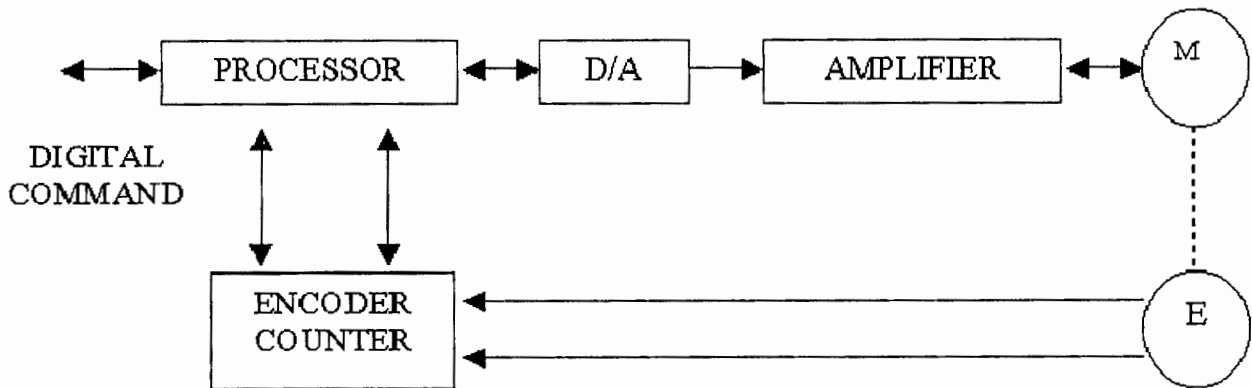


Figure 1: Blocks Diagram Servo System.

The PIC17C42 implements both the servo compensator algorithm and the trajectory profile (trapezoidal) generation. A trajectory generation algorithm is necessary for optimum motion and its implementation is as important as the servo compensator itself. The servo compensator can be implemented as a traditional digital filter, a fuzzy logic algorithm, or a simple PID algorithm (as implemented in this application note). The combination of servo compensator and trajectory calculations can take significant demands on the microcontroller. The D/A conversion can be handled by a conventional DAC or by using the PIC17C42's PWM. In either case the output signal is feed to a power stage which translates the analog signal(s) into usable voltages and currents to drive the motor. PWM output can be a duty-cycle signal in combination with a direction signal or a single signal which carries both pieces of information. In the latter case a 0% duty cycle commands a null output, a 100% duty cycle commands maximum output, and the direction as independent signal. The amplifier is configured to supply a controlled voltage to the motor. Most embedded systems use voltage output because it is simpler and cheaper. Sequential encoders produce quadrature pulse trains, from which position, speed, and direction of the motor rotation can be derived. The frequency is proportional to speed and each transition represents an increment on position. The phase of the signals is used to determine rotation direction. These encoder signals are usually decoded into Count Up and Count Down pulses, using a small state machine. These pulses

are then routed to an N-bit, up/down counter whose value corresponds to the position of the motor shaft. The decoder/counter may be implemented in hardware, software, or a combination of the two.

3. PERFORMANCE OF THE MOTOR.

The microcontroller has a PWM subsystem of high resolution. This is a very efficient form to implement a convertor of digital to analog signals when it is implemented with a bridge commutation power. When the commutation frequency is 15.6 KHz. the PWM maintains a high resolution and this allows an effective control of tension while it stays the modulation above the sensibility of the human hearing. This has a special importance since in certain places a minimum of noise is required. The figure 2 shows the power control system.

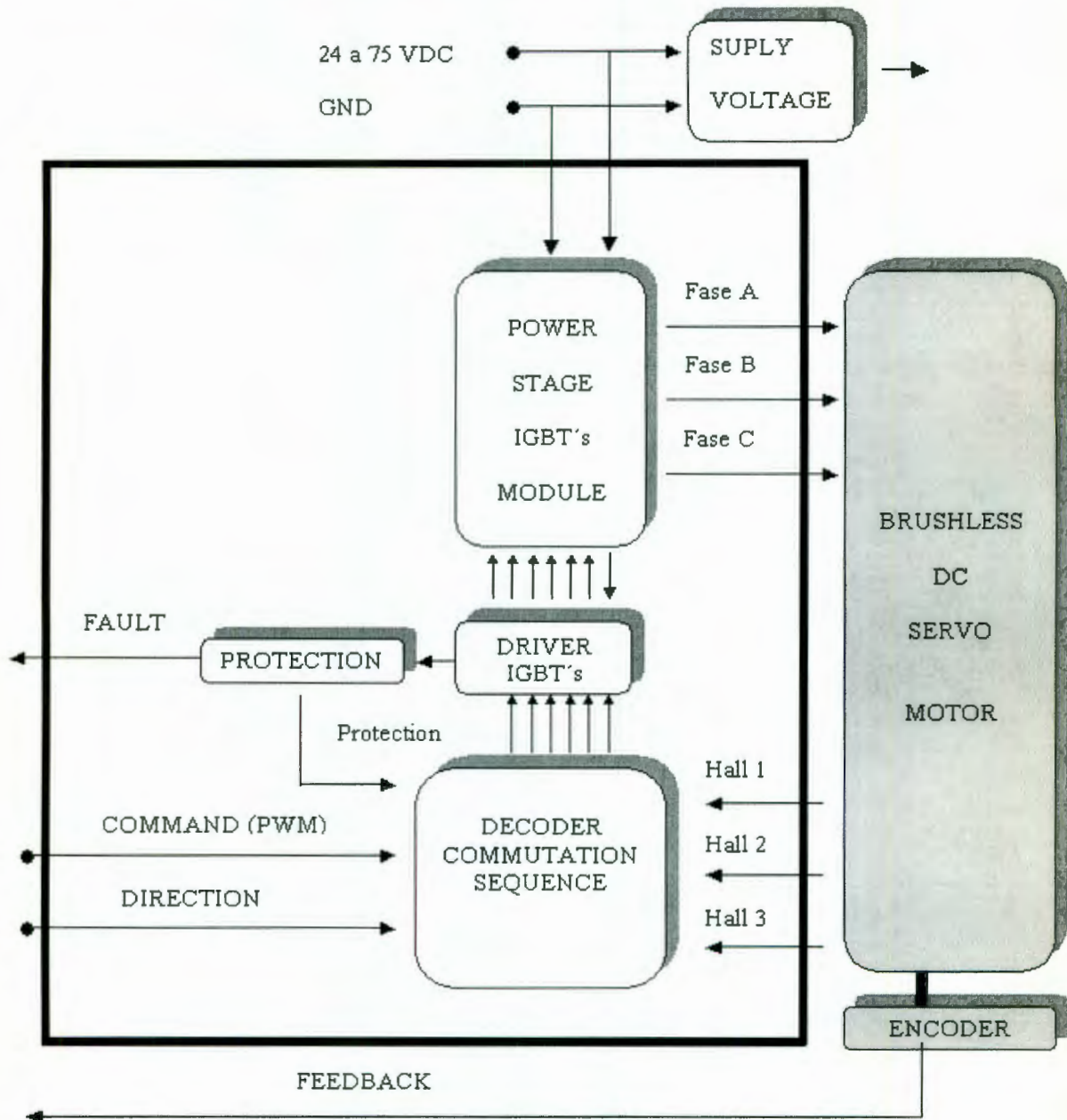


Figure 2: Block Diagram of Power Stage.

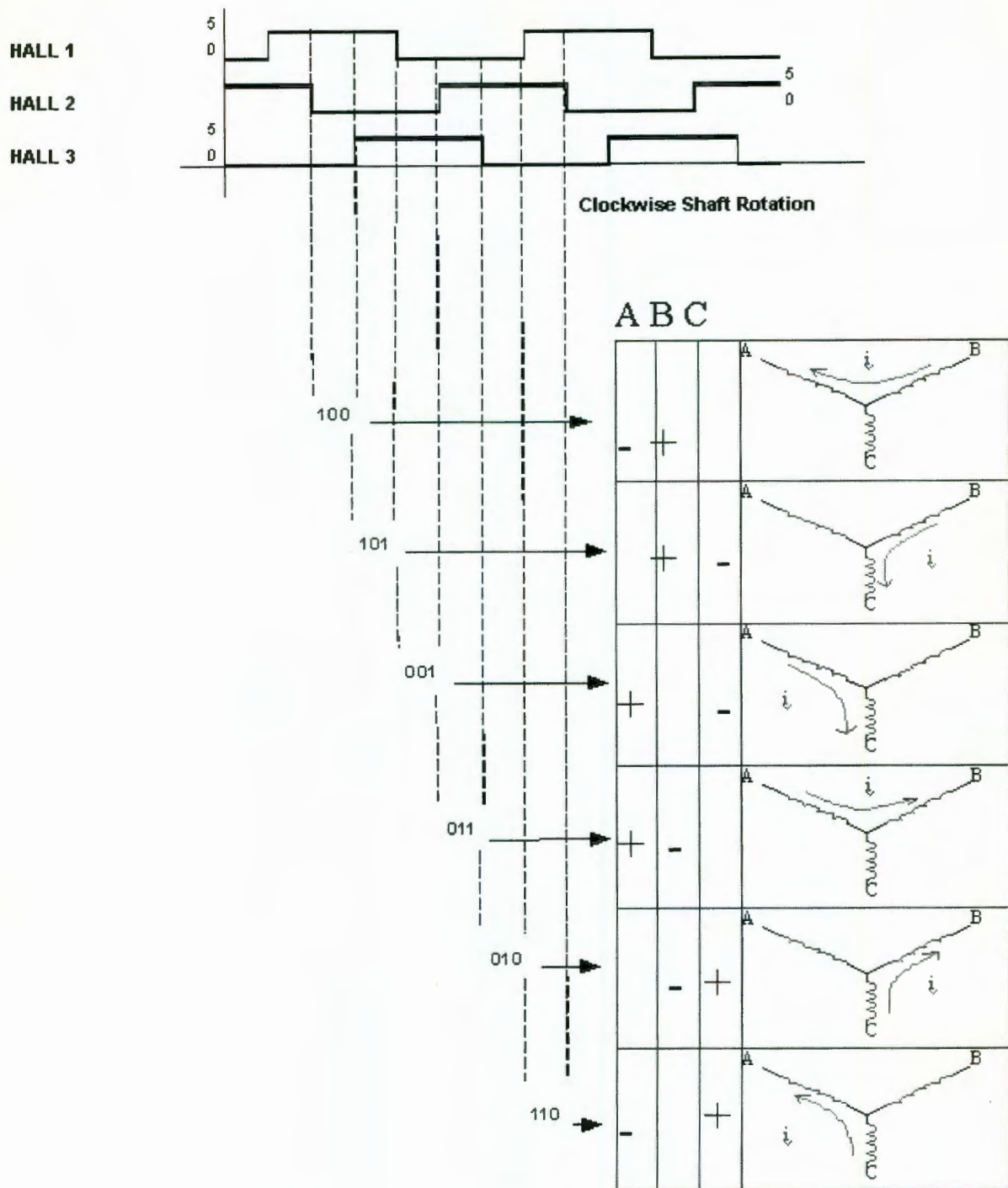


Figure 3: Commutation for Clockwise Shaft Rotation-based on Hall Sensors.

The servo driver monitors the three hall inputs. It uses internal logic circuitry to assign a rotor position to each of the six hall states, and then sends a motor current that results in the rotor movement in the commanded direction.

The three hall signals that produced clockwise shaft rotation are show in figure 3. The hall states are also listed, along with the table of phase currents the servo driver uses for each hall state.

For counterclockwise rotation, two changes are made. First, as the rotor moves counterclockwise, it passes through the same hall states but in the opposite order. (In the figure 3, read the hall states from the bottom up for counterclockwise rotation). The servo driver sends currents through the same coils show in this figure, but the direction of the current is reversed from that shown. As a result, a torque is produced in each state that causes the rotor to turn counterclockwise.

For which we use a programmable logical device where a combinational circuit with the commutation field states rotor sequences were programmed; which has as input command signals a PWM, a digital signal, an address bit, another bit to enable outputs; and finally another to disable the amplifier and to protect the system. As output signals, it has six signals one for each IGBT of the 3-phase bridge.

These signals have an interface which drive the IGBT's and which will control the devices commutation, they are connected to the motor phases and then it will apply the exact current for the wanted movement of the motor shaft.

4. CONTROLLER FOR BRIDGE OF 3-PHASES.

4.1 Description.

The IR2137 is a high voltage, high speed 3-phase IGBT driver best suited for AC motor drive applications. Integrated de-saturation logic provides ground fault protection as well as other mode of over current protection. Soft shutdown is initiated in the event of any overcurrent/ground fault conditions, and all six outputs are simultaneously turned off. Output drivers have separate turn on/off pins to facilitate independent gate drive impedance with electromagnetic interference EMI soft turn on. Optimum matched delays between phases, and between high/low side enables small deadtime, thus improving low speed performance. The brake driver eliminates additional circuits.

4.2 Introduction.

In the last decade, industrial AC drives have become more advanced in performance and more compact in size. This is largely due to the rapid development of power silicon technology, including the enhancement of IGBTs and of High Voltage Integrated Circuits. Additionally, this technological growth has been particularly accelerated for low horsepower (< 5 Hp) AC Drives. Yet the ground fault protection, a feature sought increasingly in motor drives of all power levels, is available widely only in the high-end models due to the sensors and other circuitry involved. The market, however, continues to demand a fully protected, yet inexpensive, system. To meet this demand, International Rectifier has introduced the IR2137: a new generation 3-phase IGBT gate driver, featuring full IGBT protection and advanced soft shutdown features.

4.3 Background theory for protection against failure.

According to the document [Takahashi Toshio 2001], published by International Rectifier, one of the most common, and fatal, AC drive faults for IGBTs is the Over-Current condition. Table 1 outlines the causes for the three most common Over-Current modes.

Over-Current Mode	Potential Causes
Line-to-Line Short	Mis-wiring, Motor leads shorting, Motor phase-to-phase insulation breaking down
Ground Fault	Motor phase-to-phase insulation breaking down
Shoot Through	False IGBT turn-on

Table 1: Potential Causes of Over Current.

In order to protect IGBT devices effectively during an Over-Current condition, one needs to focus on two factors.

First, one needs to detect the type of Over-Current mode and shut the system down. In both Line-to-Line and Shoot Through mode, the short-circuit current flows from and to the DC bus capacitors. Therefore, a shunt resistor in the ground path can detect these Over-Current conditions. However in the Ground Fault mode, the current flows from the AC line input, through the positive DC bus and high side IGBT, to the earth ground. Therefore, the Ground Fault condition has to be detected either with a shunt resistor on the positive bus line or by detecting the current in the output lines.

Second, one needs to look at the AC drive architecture. Protection circuitry needs to be built in a manner that does not disrupt the drive system. For instance, a typical Floating Ground Reference Architecture in which the microcontroller is on the Floating Ground reference. Therefore, any Over-Current detection circuit needs to be isolated. Alternate architecture may require reduced or no isolation.

4.4 IR2137: 3-Phase date driver with integrated IGBT protection.

The IR2137 circuit of high tension provides a solution to the handling of the six floodgates of the IGBT, while simultaneously it provides of the whole protection of the IBGT and the out one soft, in contrast with the reference architecture to floating earth, the circuit IR2137 enables a simple structure. The difference among both architectures is summarized in the table 2.

	Floating Ground System	IR2137 solution
Gate driver	Six Fast Optical Isolators	IR2137
IGBT Protection	Two Hall Effect sensors . Two comparators	IR2137 integrated function with a shunt resistor and a comparator
Additional circuits	Four floating power supplies	Eliminated (boot strap power)
	Snubber circuit or discrete soft shutdown circuits	Eliminated by integrated soft shutdown function
	Brake IGBT drive circuit with an optical isolator	Eliminated by integrated brake IGBT driver circuit

Table 2: Solution Comparison of Gate Drive and IGBT Protection.

5. SPEED AND POSITION SENSOR.

Position feedback for the system is derived from a quadrature encoder mounted on the motor shaft. Both incremental position and direction can be derived from this inexpensive device. The quadrature encoder signals are processed by a 16V8-type Programmable Logical Device PLD device. The PLD converts the quadrature pulses into two pulse streams: Count Up and Count Down (Figure 4). These signals are then fed to two 16-bit timers of the PIC17C42 (Timer3 and Timer0).

The PIC17C42 keeps track of the motor shaft's incremental position by differencing these two 16-bit timers. This operation is performed each servo sample time and the current position is calculated by adding the incremental position to the previous position. Since both timers are 16-bits, keeping track of the overflow is unnecessary, unless the encoder signals frequency is greater than 32767 times the sample frequency. For example, at a servo sample time of 1 ms, the maximum encoder rate would be 3.2767 MHz. Counter wraparound is not a concern because only the difference between the two counters is used. Two's-complement subtraction takes care of this automatically. Position is maintained as a three-byte, 24-bit quantity. However, there is no limit to the size of the internal position register. By adding the 16-bit incremental position each sample time to an N-byte software register, an N-byte position may be maintained.

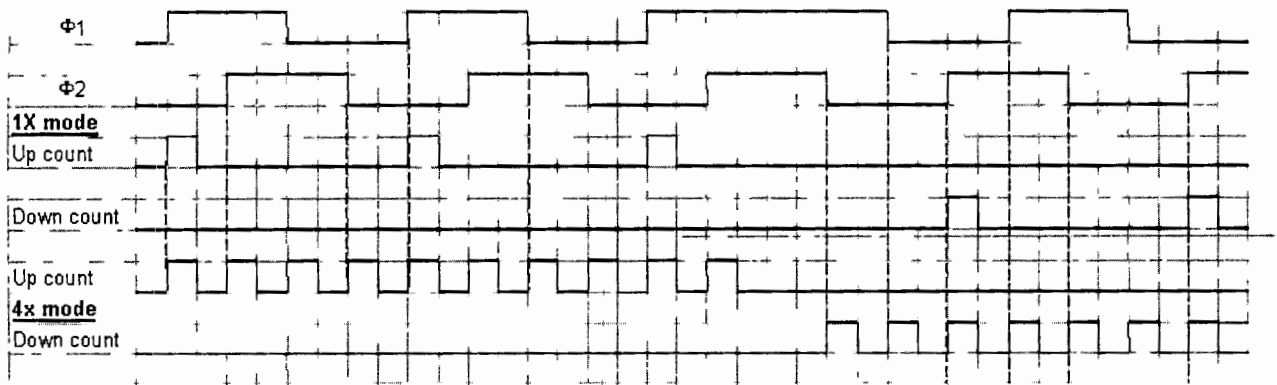


Figure 4: Sequential Encoder Signals.

6. TRAJECTORY GENERATION.

A trajectory generation algorithm is essential for optimum motion control. A linear piecewise velocity trajectory is implemented in this application. For a position move, the velocity is incremented by a constant acceleration value until a specified maximum velocity is reached. The maximum velocity is maintained for a required amount of time and then decremented by the same acceleration (deceleration) value until zero velocity is attained. The velocity trajectory is therefore trapezoidal for a long move and triangular for a short move where maximum velocity was not reached. The pre-move subroutine is invoked once at the beginning of a move to calculate the trajectory limits. The move routine is then invoked at every sample time to calculate new "desired" velocity and position values as follows:

$$V(K) = V(K-1) + A \quad (A = \text{Acceleration})$$

$$P(K) = P(K-1) + V(K-1) + A/2$$

7. THE COMPENSADOR.

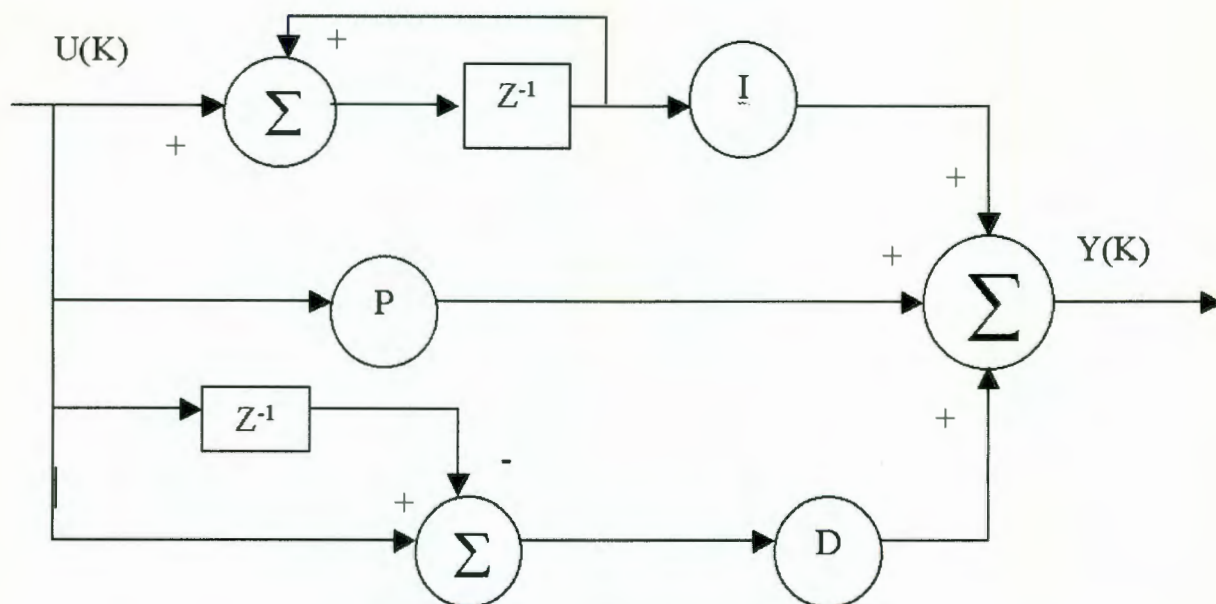


Figure 5: Digital PID Implementation.

A PID routine is the most widely used algorithm for servo motor control. Although it may not be the optimum controller for all applications, it is easy to understand and tune. The standard digital PID algorithm's [Ogata, K.1996] form is shown in Figure 5. $U(k)$ is the position or velocity error and $Y(k)$ is the output. This algorithm has been implemented using the PIC17C42's math library. Only 800 instruction cycles are required, resulting in a 0.2 ms PID execution time at 16 MHz. Integrator windup is a condition which occurs in PID controllers when a large following error is present in the system, for instance when a large step disturbance is encountered. The integrator continually builds up during this following error condition even though the output is saturated. The integrator then "unwinds" when the servo system reaches its final destination causing excessive oscillation. The PID implementation shown in Figure 5 avoids this problem by stopping the action of the integrator during output saturation.

The controller discret PID is an algorithm more used for the motor control by its easy handling and tune. The figure 5 an algorithm of a common digital PID is presented where $U(k)$ it is the position error or speed and $Y(k)$ it is the exit. This algorithm had been implemented in a PIC17C42. Alone with 800 instructions this is at one time of 0.2 ms the execution of the PID with a crystal of 16 MHz.

8. RESULTS.

Once the algorithms were implemented, the PID was tuned trying to achieve the faster response with little overshoot using the phase margin and crossover frequency technique for design system [Jacob Such a Ph D, 1994] and [Younkin G. W. 1996], we obtained the values of $K_p=1800$, $K_d=15600$ and $K_i=52$ as a result.

The figure 6 shows the system response position for $3.125 \text{ pulses} / \text{ms}^2$ as acceleration and a speed of $100 \text{ pulses} / \text{ms}$. The figure 7 shows the system response velocity for the same values.

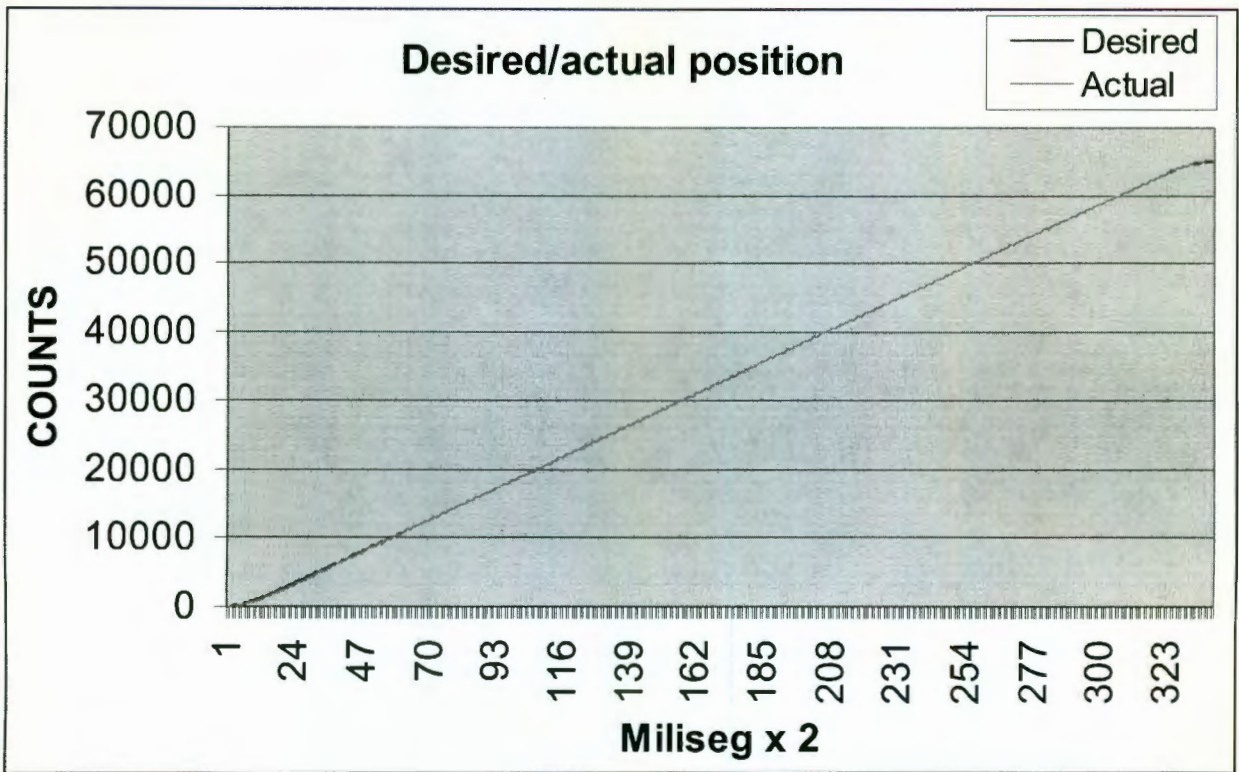


Figure 6: Position Response.

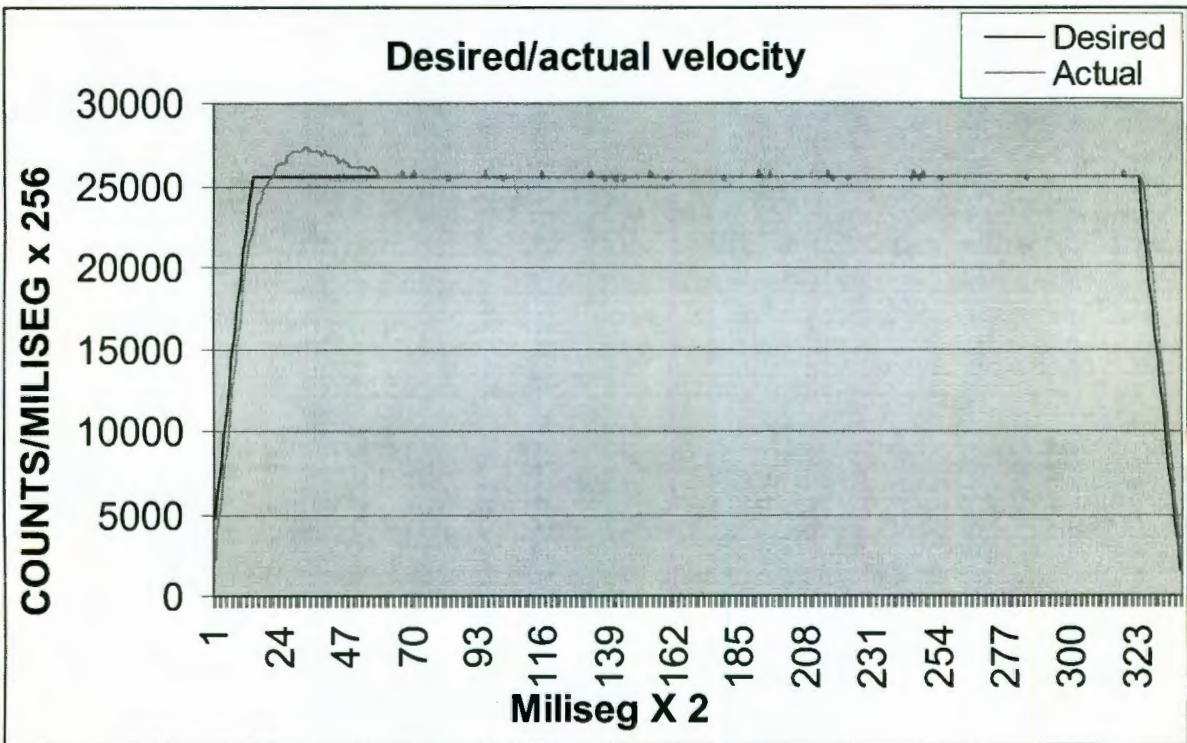


Figure 7: Velocity Response.

9. CONCLUSIONS

With this project was possible to assimilate the technology of servo controls looking at detail their behavior, advantages and disadvantages, having the enough knowledge to produce them at low cost as well as the necessary experience to decide on their applications.

A compact servo controller is available with few components and similar characteristic to some commercial although functions are limited. But the initial expectations have been completed and in this moment we have the experience to design servo systems less than one horse power for specific applications. If the research continues it is possible to develop systems with higher versatility.

The use of a high integrated microcontroller, is a good solution because few external components are required and this is translated in an effective solution, this joined to the high technology in combinational logic and drivers for modules with insulated gate bipolar transistors, the system is compact and good efficiency.

The processor that was used has a good performance and it works perfectly with the control algorithm but any other can be used depending on the application demands and market demands, this is open for investigations in the future.

10. REFERENCES

Jacob Such a Ph D, 1994. Step-by-Step Design of motion control systems.

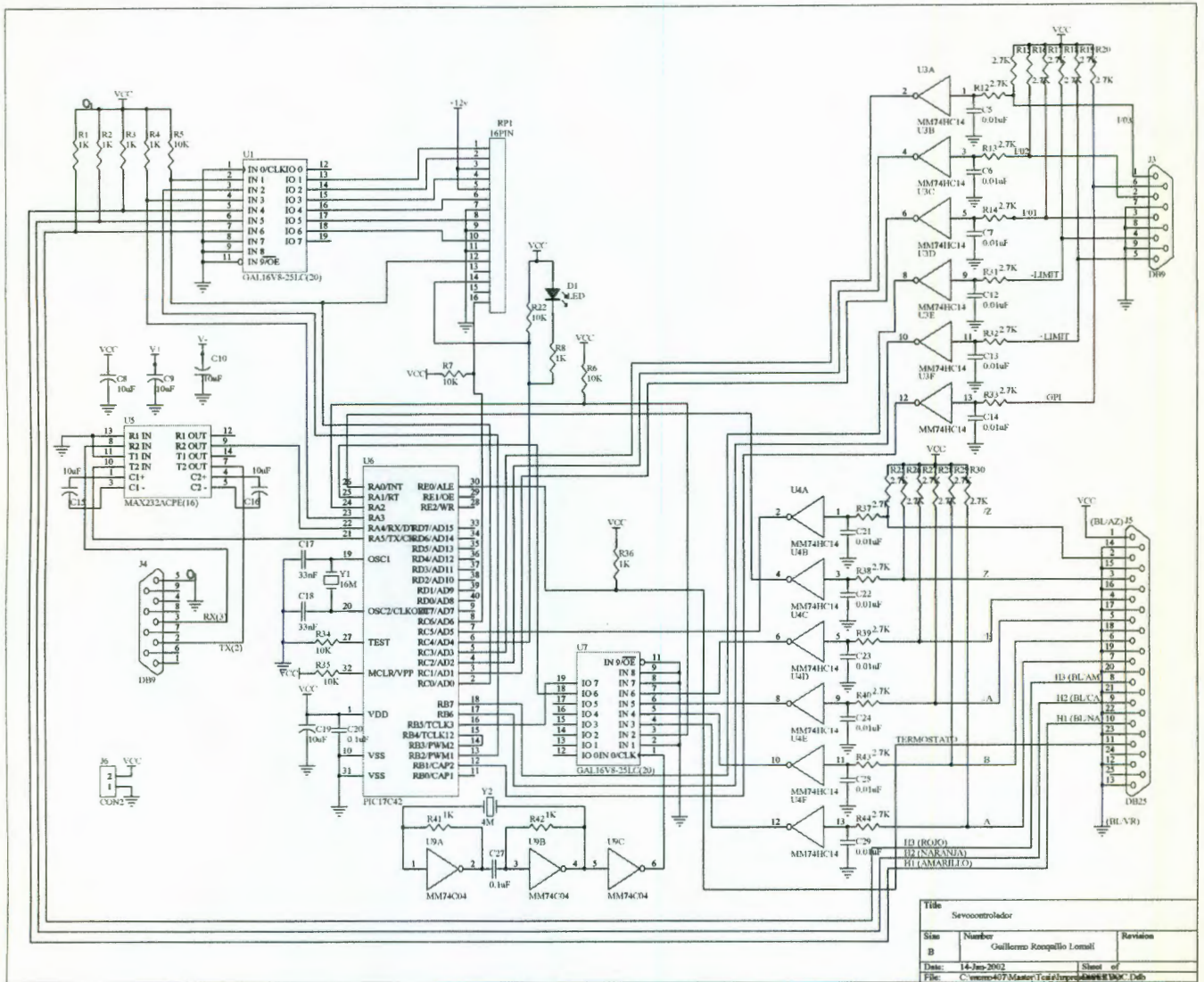
Ogata, K. 1996 Systems of Control in Discreet Time. University of Minnesota.

Takahashi Toshio 2001. Solving IGBT Protection in AC or BLDC Motor Drive. International Rectifier.

Younkin G. W. 1996 industrial Servo Control Systems Fundamentals and Applications. Inc. Fond du Lac, Wisconsin.

C Circuito de control del servosistema.

Circuito desarrollado y realizado por el alumno apoyado en (Savant – Roden).



D. Listado en CUPL del PLD de la secuencia de conmutación.

```

/*****/
/* Advanced PLD99: Schematic to CUPL translator */
/*****/
/* Translated at 16:07:03 */
/* on 14-May-2001 */
/*****/
NAME SERVOCNM;
DATE 14/05/01;
PARTNO UNO;
REVISION PRIMERA;
DESIGNER GUILLERMO RONQUILLO LOMELI;
COMPANY TESIS;
ASSEMBLY GRL;
LOCATION QUERETARO QUERETARO;
DEVICE g16v8;
/*****/
/* All Parts Connected */
/* - [OPAD,U1] */
/* - [OPAD,U2] */
/* - [OPAD,U3] */
/* - [OPAD,U4] */
/* - [OPAD,U5] */
/* - [OPAD,U6] */
/* - [AND5B3,U7] */
/* - [NOR2,U8] */
/* - [AND5B2,U9] */
/* - [AND5B3,U10] */
/* - [NOR2,U11] */
/* - [AND5B2,U12] */
/* - [AND5B3,U13] */
/* - [NOR2,U14] */
/* - [AND5B2,U15] */
/* - [AND4B3,U16] */
/* - [AND4B2,U17] */
/* - [NOR4,U18] */
/* - [AND5B4,U19] */
/* - [AND5B3,U20] */
/* - [AND4B3,U21] */
/* - [AND4B2,U22] */
/* - [NOR4,U23] */
/* - [AND5B4,U24] */
/* - [AND5B3,U25] */
/* - [AND4B3,U26] */
/* - [AND4B2,U27] */
/* - [NOR4,U28] */
/* - [AND5B4,U29] */
/* - [AND5B3,U30] */
/* - [IPAD,U31] */
/* - [IPAD,U32] */
/* - [IPAD,U33] */
/* - [IPAD,U34] */
/* - [IPAD,U35] */
/* - [IPAD,U36] */
/*****/

```



```

/*****/
/* Input and Output Pin Declarations */
/*****/
PIN 13 = Q1;
/* Part = (U1) */
PIN 14 = Q2;
/* Part = (U2) */
PIN 15 = Q3;
/* Part = (U3) */
PIN 16 = Q4;
/* Part = (U4) */
PIN 17 = Q5;
/* Part = (U5) */
PIN 18 = Q6;
/* Part = (U6) */
PIN 2 = E;
/* Part = (U31) */
PIN 3 = PWM;
/* Part = (U32) */
PIN 4 = DIR;
/* Part = (U33) */
PIN 5 = H1;
/* Part = (U34) */
PIN 6 = H2;
/* Part = (U35) */
PIN 7 = H3;
/* Part = (U36) */
/*****/
/*****/
/* Logic Equations */
/*****/
NetU7_O = PWM & H3 & !E & !DIR & !H1;
/* Part = (U7) */
Q1 = !(NetU7_O # NetU8_I0);
/* Part = (U8) */
NetU8_I0 = DIR & PWM & H1 & !H3 & !E;
/* Part = (U9) */
NetU10_O = PWM & H1 & !H2 & !DIR & !E;
/* Part = (U10) */
Q2 = !(NetU10_O # NetU11_I0);
/* Part = (U11) */
NetU11_I0 = H2 & DIR & PWM & !H1 & !E;
/* Part = (U12) */
NetU13_O = PWM & H2 & !DIR & !E & !H3;
/* Part = (U13) */
Q3 = !(NetU13_O # NetU14_I0);
/* Part = (U14) */
NetU14_I0 = DIR & PWM & H3 & !H2 & !E;
/* Part = (U15) */
NetU16_O = H1 & !H3 & !DIR & !E;
/* Part = (U16) */
NetU17_O = H3 & DIR & !E & !H1;
/* Part = (U17) */
Q4 = !(NetU16_O # NetU17_O # NetU18_I1 # NetU18_I0);
/* Part = (U18) */

```



```

NetU18_I1 = H3 & !E & !PWM & !DIR & !H1;
/* Part = (U19) */
NetU18_I0 = DIR & H1 & !E & !PWM & !H3;
/* Part = (U20) */
NetU21_O = H2 & !E & !DIR & !H1;
/* Part = (U21) */
NetU22_O = DIR & H1 & !E & !H2;
/* Part = (U22) */
Q5 = !(NetU21_O # NetU22_O # NetU23_I1 # NetU23_I0);
/* Part = (U23) */
NetU23_I1 = H1 & !E & !PWM & !DIR & !H2;
/* Part = (U24) */
NetU23_I0 = H2 & DIR & !E & !PWM & !H1;
/* Part = (U25) */
NetU26_O = H3 & !E & !DIR & !H2;
/* Part = (U26) */
NetU27_O = DIR & H2 & !E & !H3;
/* Part = (U27) */
Q6 = !(NetU26_O # NetU27_O # NetU28_I1 # NetU28_I0);
/* Part = (U28) */
NetU28_I1 = H2 & !E & !PWM & !DIR & !H3;
/* Part = (U29) */
NetU28_I0 = DIR & H3 & !E & !PWM & !H2;
/* Part = (U30) */
/*****

```

E. Listado en CUPL del PLD para el encoder.

```
Name      Encoder      ;
Partno    Dos          ;
Revision  1            ;
Date      21/06/01    ;
Designer  Guillermo Ronquillo Lomeli ;
Company   UAQ          ;
Assembly  ;
Location  Querétero   ;
Device    g16v8;
Format    ;

/*****
/* This PLD design (Revision 1) created on 21/06/01      */
/* para          UAQ          */
/* y es almacenado como Eacoder          */
*****/

/** Entradas **/
Pin 1 = clk ;
Pin 2 = reset ;
Pin 3 = x4 ;
Pin 4 = p0 ;
Pin 5 = p90 ;
Pin 11 = oe ;

/** Salidas **/
Pin 12 = s2 ;
Pin 13 = s4 ;
Pin 14 = p0d ;
Pin 15 = p90d ;
Pin 18 = cntup;
Pin 19 = cntdn;

/** Declaración de variables intermedias **/
s2.d = p0d & !reset ;
s4.d = p90d & !reset ;
p0d.d = p0 & !reset ;
p90d.d = p90 & !reset ;

/** Ecuaciones lógicas **/
cntdn = !p0d & !s2 & p90d & !s4
        # p0d & !s2 & p90d & s4 & x4
        # p0d & s2 & !p90d & s4 & x4
        # !p0d & s2 & !p90d & !s4 & x4 ;

cntup = p0d & !s2 & !p90d & !s4
        # p0d & s2 & p90d & !s4 & x4
        # !p0d & s2 & p90d & s4 & x4
        # !p0d & !s2 & !p90d & s4 & x4 ;
```

F. Código completo del sistema.

Estos algoritmos están referenciados en la notas de aplicación AN532 de [MICROCHIP 1996] y [MICROCHIP 1994] fueron aumentadas y adaptadas por el alumno.

```
*****
;
;   Programa principal
;   -----
;   Adaptado por el alumno
;
*****
PROCESSOR    PIC17C42
LIST         COLUMNS=120,XREF=YES,NOWRAP,LINES=255,R=DEC

RELOJ_MAESTRO SET    16000000
RANGO_MUESTREO SET    1000
RANGO_ENCODER SET    2000
RANGO_VELOCIDAD SET    6000

RANGOTRANS    SET    19200

    include "p17c42.inc"

    include "help.mac"

#define _PICMASTER_DEBUG VERDADERO
#define _SERVO_PID        VERDADERO
#define DECIO             VERDADERO
#define PUERTOSERIAL      VERDADERO

#define HABRANGOTRANS(bps) ((10*RELOJ_MAESTRO/(64*bps))+5)/10-1
RELOJSAL      set    RELOJ_MAESTRO>>2

INIC_TCON1 SET    0X06
INIC_TCON2 SET    0X3F
INIC_PR1     SET    0XFF
INIC_PR2     SET
((10*RELOJ_MAESTRO/(4*(INIC_PR1+1)*RANGO_MUESTREO))+5)/10

INIC_PW1DCH  SET    0X00
INIC_PW1DCL  SET    0X00
INIC_PW2DCH  SET    0X7F
INIC_PW2DCL  SET    0XC0

INIC_T0STA   SET    0X80
INIC_RCSTA   SET    0X90
INIC_TXSTA   SET    0X20
INIC_SPBRG   SET    HABRANGOTRANS(RANGOTRANS)

INIC_DDRB   SET    0XF3
INIC_DDRD   SET    0X00
INIC_DDRC   SET    0XBE ;nuevo

PWMINL      SET    0X40
PWMINH      SET    0X01
PWMAXL      SET    0X80
PWMAXH      SET    0XFE
```

;VARIABLES GLOBALES

CBLOCK 0X18
DPX,DPX1,DPX2,DPX3
AARG,AARG1,BARG,BARG1
ENDC

CBLOCK 0X18
TMP,TMP1,TMP2,TMP3
MOVTMP,MOVTMP1,MOVTMP2,MOVTMP3
ENDC

CBLOCK 0X20
VL,VL1,VL2
AL,AL1,AL2

KP,KP1
KV,KV1
KI,KI1

IM
FV,FV1
FA,FA1

VALBUF,VALBUF1,VALBUF2

DVALBUF,DVALBUF1,DVALBUF2
ISRBSR,ISRWREG
CMDCHAR,CMDTEMP,CMDPTRH,CMDPTRL
PARTEMP,PARLEN,PARPTR

CPOSICION,CPOSICION1,CPOSICION2
CVELOCIDAD,CVELOCIDAD1,CVELOCIDAD2
CMPOSICION,CMPOSICION1,CMPOSICION2
CMVELOCIDAD,CMVELOCIDAD1,CMVELOCIDAD2

VALARREGLOA,VALARREGLOB
HEXVAL,HEXTMP,HEXSTAT

OPOSICION,OPOSICION1
OPOSICION2,OPOSICION3
POSICION,POSICION1,POSICION2
VELOCIDAD,VELOCIDAD1,VELOCIDAD2
NMOVVAL,NMOVVAL1,NMOVVAL2,NMOVVAL3
MOVVAL,MOVVAL1,MOVVAL2,MOVVAL3
HMOVVAL,HMOVVAL1,HMOVVAL2,HMOVVAL3
TIEMPOMOV,TIEMPOMOV1

SIGNOMOV

T1,T11
T2,T21
TAU,TAU1
NMODO
MODO

MPOSICION,MPOSICION1,MPOSICION2
MVELOCIDAD,MVELOCIDAD1
MVELOCIDAD2,MVELOCIDAD3
ERRORPOS,ERRORPOS1,ERRORPOS2
ERRORVEL,ERRORVEL1,ERRORVEL2

SIGNO

Y,Y1,Y2,Y3
U0,U01,U1,U11

YPWM,YPWM1,YPWM2,YPWM3
YPWMIN,YPWMIN1,YPWMAX,YPWMAX1

SERVOBANDERA
MODOTYPE
ESTADOEXT
ESTADOMOV
BANDERAMOV
BANDERASAT
INTEGRAL,INTEGRAL1

DECVAL,DECSTAT,DECTMP,DECSIGNO

A,A1,A2,A3
V,V1,V2,V3
MOVPCBUF,MOVPCBUF1,MOVPCBUF2,MOVPCBUF3
MOVVBUF,MOVVBUF1,MOVVBUF2,MOVVBUF3

CUENTARRIBA,CUENTARRIBA1
CUENTABAJO,CUENTABAJO1

MOVDEL,MOVDEL1,MOVDEL2,MOVDEL3
PH2FLAT,PH2FLAT1
INDEXPOS,INDEXPOS1,INDEXPOS2

NUMCORRIMIENTO
DIR

NUMDATO
BANDERAENVIA
ENVIALISTO
BUFENVIA,BUFENVIA1

CERO,UNO
tblptrlTemp,tblptrhTemp
TbLatLo,TbLatHi
alustaTemp

ENDC

AUTONO MACRO

bsf ALUSTA,FS0
bsf ALUSTA,FS1

```

        bsf     ALUSTA,FS2
        bsf     ALUSTA,FS3
    ENDM

AUTOINC MACRO
        bsf     ALUSTA,FS0
        bcf     ALUSTA,FS1
        bsf     ALUSTA,FS2
        bcf     ALUSTA,FS3
    ENDM

AUTODEC MACRO
        bcf     ALUSTA,FS0
        bcf     ALUSTA,FS1
        bcf     ALUSTA,FS2
        bcf     ALUSTA,FS3
    ENDM

LOADAB MACRO    A,B
        MOVFP A+B0,AARG+B0
        MOVFP A+B1,AARG+B1
        MOVFP B+B0,BARG+B0
        MOVFP B+B1,BARG+B1
    ENDM

CR    SET    0X0D
CAN  SET    0X18
BS   SET    0X08
SP   SET    0X20
LF   SET    0X0A
MN   SET    '.'

CHARACTERLISTO    SET    0X01

NUMPAR            SET    0X08

CMD_OK           SET    '!'
CMD_MALO SET     '?'

HEX_SP          SET    0X00
HEX_MN          SET    0X01
HEX_CR          SET    0X02
HEX_CAN         SET    0X03

DEC_SP          SET    0X00
DEC_MN          SET    0X01
DEC_CR          SET    0X02
DEC_CAN         SET    0X03

NULO            SET    CR
MOVER           SET    'M'
CAMBMODO        SET    'O'
ESTABPARAMETROS SET    'S'
LEERPARAMETROS SET    'R'
DO_SHUTTER      SET    'C'
LEERCOMPOSICION SET    'P'

```

```

;LEERCOMVELOCIDAD      SET  'V'
LEERPOSACTUAL          SET  'p'
;LEERVELACTUAL         SET  'v'
LEERESTEXT             SET  'X'
LEERESTADOMOV         SET  'Y'
;LEERPOSINDICE         SET  'I'
;ESTABLECEPOS          SET  'H'
REESTABLECER          SET  'Z'
PARO                   SET  's'
CAPTURA               SET  'c'
RESOLUCION             SET  'x'
HABILITA               SET  'h'
DESHABILITA           SET  'd'

```

```

CMD_DEF MACRO  FUNC,ROOT
              DATA ROOT
              DATA FUNC
          ENDM

```

```

CMD_ENTRY_LEGTH      SET  2

```

```

CMD_START MACRO      LABEL
LABEL
          ENDM

```

```

CMD_END MACRO
          DATA 0X00
          ENDM

```

```

#define      _KP      1800
#define      _KI      52
#define      _KV      15600

```

```

VELOCIDADLIMITE SET  4000
ACCLIMITE SET  100

```

```

      if _SERVO_PID
      _NUMCORRIMIENTO EQU 4
      endif

```

```

;*****
;
;      Inicio
;      -----
;      Aumentada y adaptada
;*****

```

```

      ORG 0x0
      goto inicio

      ORG 0x20
      goto InterrutPoll

```

```

inicio
      bsf  CPUSTA,GLINTD

```

```

      AUTONO

```

```

movlw 0x18
movpf WREG,FSR0

lazomem
clrf INDF0,F ;Limpia toda la RAM de proposito general
incfsz FSR0,F
goto lazomem

incf UNO,F
movlb BANK3
movlw INIC_TCON2
movfp WREG,TCON2

movlw INIC_PW1DCH
movfp WREG,PW1DCH

movlw INIC_PW1DCL
movfp WREG,PW1DCL

movlw INIC_PW2DCH
movfp WREG,PW2DCH

movlw INIC_PW2DCL
movfp WREG,PW2DCL

movlw INIC_TCON1
movfp WREG,TCON1

movlb BANK2

movlw INIC_PR1
movfp WREG,PR1
movlw INIC_PR2
movfp WREG,PR2

movlb BANK0

movlw INIC_T0STA
movfp WREG,T0STA

movlw 0xf8
movwf PORTA

if PUERTOSERIAL
    movlw INIC_RCSTA
    movfp WREG,RCSTA

    movlw INIC_TXSTA
    movfp WREG,TXSTA

    movlw INIC_SPBRG
    movfp WREG,SPBRG
endif

movlw INIC_DDRB

```



```

movfp  WREG,DDRB

movlb  BANK1

clrf   PORTC           ;nuevo
movlw  INIC_DDRC      ;nuevo
movwf  DDRC           ;nuevo
bsf    PORTC,bit0     ;nuevo
bsf    PORTC,bit6     ;nuevo

if (_SERVO_PID == VERDADERO)
    MOVK16  _KP,KP
    MOVK16  _KI,KI
    MOVK16  _KV,KV
endif

MOVK16  ACCLIMITE,AL
MOVK16  VELOCIDADLIMITE,VL

movlw  _NUMCORRIMIENTO
movwf  NUMCORRIMIENTO

movpf  PW1DCH,YPWM+B1

movlw  PWMAXL
movpf  WREG,YPWMAX+B0
movlw  PWMAXH
movpf  WREG,YPWMAX+B1
movlw  PWMINL
movpf  WREG,YPWMIN+B0
movlw  PWMINH
movpf  WREG,YPWMIN+B1

clrf   PIR,F
clrf   INTSTA,F
bsf    PIE,TMR2IE
bsf    INTSTA,PEIE

bcf    CPUSTA,GLINTD

movlb  BANK2

zeroctr
clrf   TMR0L,F
clrf   TMR0H,F

clrf   TMR3L,F
clrf   TMR3H,F

movlb  BANK1
bcf    PORTC,bit6

movlw  0xFF
retardodecsz WREG,F
goto   retardo
bsf    PORTC,bit6

```

```

movlb  BANK2          ;nuevo

movfp  TMR0L,WREG
iorwf  TMR0H,W
iorwf  TMR3L,W
iorwf  TMR3H,W
tstfsz WREG
goto   zeroctr

goto   PollingLoop

```

```

;*****
; Rutina de servicio a la interrupción
; -----
; Aumentada y adaptada
;*****
InterrutPoll

```

```

movwf  ISRWREG
movfp  ALUSTA,WREG
movwf  alustaTemp
movpf  BSR,ISRBSR
movpf  TBLPTRL,tblptrlTemp
movpf  TBLPTRH,tblptrhTemp
tlrd   0,TblLatLo
tlrd   1,TblLatHi

movlb  BANK1

call   MEDIRPOSYVEL

call   leerEstadoExt

rlncf  ESTADOMOV,W
andlw  0x01
subwf  BANDERAMOV,W
btfsz  WREG,MSB
call   ejecutarPremovimiento

btfsz  ESTADOMOV,bit6
call   movimiento

call   calcularError

tstfsz SERVOBANDERA
call   ejecutarServo

tstfsz BANDERAENVIA
call   EnviaDatos

movlb  BANK1

clrf  PIR,F

```

```

tlwt    0,TblLatLo
tlwt    1,TblLatHi
movfp   tblptrlTemp,TBLPTRL
movfp   tblptrhTemp,TBLPTRH
movfp   ISRBSR,BSR
movfp   alustaTemp,ALUSTA
movfp   ISRWREG,WREG

```

```
retfie
```

```

;*****
;
;   Calculo del error de posición y velocidad
;   -----
;   Sin modificación
;*****

```

```
calcularError
```

```

MOV24 POSICION,ERRORPOS
SUB24 MPOSICION,ERRORPOS

```

```

btfsc   ERRORPOS+B2,MSB
goto    pneg

```

```
ppos
```

```

movfp   ERRORPOS+B1,WREG
andlw   0x80
iorwf   ERRORPOS+B2,F
clrf    WREG,F
cpfsgt  ERRORPOS+B2
goto    psatok
clrf    ERRORPOS+B2,F
movlw   0X7F
movpf   WREG,ERRORPOS+B1
setf    ERRORPOS,F
goto    psatok

```

```
pneg
```

```

movfp   ERRORPOS+B1,WREG
iorlw   0x7F
andwf   ERRORPOS+B2,F
setf    WREG,F
cpfslt  ERRORPOS+B2
goto    psatok
setf    ERRORPOS+B2,F
clrf    ERRORPOS+B1,F
bsf     ERRORPOS+B1,MSB
clrf    ERRORPOS,F

```

```
psatok
```

```

MOV24 VELOCIDAD,ERRORVEL
SUB24 MVELOCIDAD,ERRORVEL

```

```

btfsc   ERRORVEL+B2,MSB
goto    vneg

```

```
vpos
```

```
movfp   ERRORVEL+B1,WREG
```

```

    andlw 0x80
    iorwf ERRORVEL+B2,F
    clrf  WREG,F
    cpfsgt ERRORVEL+B2
    goto  vsatok
    clrf  ERRORVEL+B2,F
    movlw 0X7F
    movpf WREG,ERRORVEL+B1
    setf  ERRORVEL,F
    goto  vsatok
vneg
    movfp ERRORVEL+B1,WREG
    iorlw 0x7F
    andwf ERRORVEL+B2,F
    setf  WREG,F
    cpfslt ERRORVEL+B2
    goto  vsatok
    setf  ERRORVEL+B2,F
    clrf  ERRORVEL+B1,F
    bsf  ERRORVEL+B1,MSB
    clrf  ERRORVEL,F
vsatok
    return

```

```

;*****
;
;   Lectura de los contadores del encoder
;   -----
;
;   Sin modificación.
;*****
MEDIRPOSYVEL

```

```

MOVFP16    CUENTARRIBA,TMP+B0

```

```

leerArriba

```

```

    movpf TMR0H,WREG
    movpf TMR0L,CUENTARRIBA+B0
    cpfseq TMR0H
    goto  leerArriba
    movpf WREG,CUENTARRIBA+B1

```

```

    clrf  MVELOCIDAD+B0,F

```

```

MOV16 CUENTARRIBA,MVELOCIDAD+B1
SUB16 TMP+B0,MVELOCIDAD+B1

```

```

MOVFP16    CUENTABAJO,TMP+B0

```

```

leerAbajo

```

```

    movlb BANK2
    movpf TMR3H,WREG
    movpf TMR3L,CUENTABAJO+B0
    cpfseq TMR3H
    goto  leerAbajo
    movpf WREG,CUENTABAJO+B1

```

```

MOVFP16 CUENTABAJO+B0,TMP+B2
SUB16 TMP+B0,TMP+B2
SUB16 TMP+B2,MVELOCIDAD+B1

```

```

clrf MVELOCIDAD+B3,F
btfsc MVELOCIDAD+B2,MSB
setf MVELOCIDAD+B3,F

```

```

ADD24 MVELOCIDAD+B1,MPOSICION

```

```

return

```

```

;*****
;
; Lectura del estado externo del sistema
;
; -----
;
; Sin modificación.
;*****

```

```

leerEstadoExt

```

```

btfss INTSTA,bit4
goto otherbits

```

```

MOV24 MPOSICION,INDEXPOS

```

```

bcf INTSTA,bit4
bsf ESTADOEXT,MSB

```

```

otherbits

```

```

movlb BANK0
movfp PORTB,WREG
rrcf WREG,F
andlw 0x61
movpf WREG,TMP
swapf TMP,F
iorwf TMP,W
iorwf ESTADOEXT,F

```

```

movlb BANK1
btfsc PORTC,bit4
return
bsf PORTC,bit0
clrf SERVOBANDERA,F
clrf BANDERAMOV,F
clrf ESTADOMOV,F
movlb BANK3
movlw 0x00
movwf PW1DCL
movwf PW1DCH

```

```

return

```

```

;*****
;
; Ciclo de pregunta por el puerto serie
;
; -----
;
; Aumentada y adaptada

```

```

;*****
PollingLoop
    if PUERTOSERIAL
        call    IdleFunction

        call    GetChk
        cpfseq UNO
        goto    PollingLoop

        call    GetChar
        movpf   WREG,CMDCHAR
        goto    ejecutarComando

    else
        clrwdt
        goto    PollingLoop
    endif

```

```

;*****
; Multiplicación de 16 bits*16 bits = 32bits
; -----
; Sin modificación
;*****

```

```

Dmult
    btfss    BARG+B1,MSB
    goto     argsok
    NEG16   AARG+B0
    NEG16   BARG+B0
argsok
    clrf    DPX+B3,F
    clrf    DPX+B2,F

    MULMAC

    rlc     AARG+B1,W    ;since BARG is always made
    rrc     DPX+B3,F    ;the last bit known to be
    rrc     DPX+B2,F
    rrc     DPX+B1,F
    rrc     DPX+B0,F

    return

```

```

;*****
; Generación de Trayectoria
; -----
; Sin modificación
;*****
ejecutarPremovimiento

```

```

CLR16 INTEGRAL

MOV24 NMOVVAL,MOVVAL

```

```

bcf     ESTADOMOV,bit7
bsf     ESTADOMOV,bit6
bsf     ESTADOMOV,bit5
movfp   UNO,WREG
movpf   WREG,BANDERAMOV

clrf    OPOSICION+B0,F
MOV24   POSICION,OPOSICION+B1
MOV32   OPOSICION,MOVVBUF

clrf    BANDERASAT,F
CLR16   TIEMPOMOV
CLR16   T1
CLR16   T2
CLR16   TAU
CLR32   MOVDEL
CLR16   PH2FLAT

tstfsz  MODOTYPE
goto    vmode

```

pmode

```

MOVFP24 MOVVAL,TMP
btfsz   TMP+B2,MSB
goto    mvpos

NEG24   TMP

```

mvpos

```

clrf    MOVTMP+B0,F
clrf    MOVTMP+B1,F

clrf    MOVTMP+B2,F
bsf     MOVTMP+B0,bit0
bsf     MOVTMP+B0,bit1

SUB24   MOVTMP,TMP

btfsz   TMP+B2,MSB
goto    nonzero
setf    SERVOBANDERA,F
clrf    BANDERAMOV,F
bcf     ESTADOMOV,bit5
bcf     ESTADOMOV,bit6

ADD24   MOVVAL,POSICION

return

```

nonzero

```

CLR32   MOVVBUF

movfp   MOVVAL+B2,WREG
andlw   0x80

```

movpf WREG,SIGNOMOV

clrf V+B3,F
MOV24 VL,V
clrf A+B3,F
MOV24 AL,A

clrf WREG,F
cpfsgt SIGNOMOV
goto minc

NEG32 V
NEG32 A

minc

clrf HMOVVAL+B0,F
MOV24 MOVVAL,HMOVVAL+B1
RRC32 HMOVVAL

goto modeready

vmode

btfsc MODOTYPE,MSB
goto tmode
clrf HMOVVAL+B3,F

MOV24 MOVVAL,HMOVVAL

btfsc MOVVAL+B2,MSB
setf HMOVVAL+B3,F

SUB32 MOVVBUF,HMOVVAL
movfp HMOVVAL+B3,WREG
andlw 0x80
movpf WREG,SIGNOMOV

clrf V+B3,F
MOV24 VL,V

clrf A+B3,F
MOV24 AL,A

clrf WREG,F
cpfsgt SIGNOMOV
goto modeready
NEG32 V
NEG32 A

modeready

clrf MOVVAL+B3,F
btfsc MOVVAL+B2,MSB
setf MOVVAL+B3,F

setf SERVOBANDERA,F

return


```

tmode
MOV16 MOVVAL+B1,YPWM

clrf  SERVOBANDERA,F
call  ejecutarTorque
clrf  BANDERAMOV,F
bcf   ESTADOMOV,bit5

return

```

```

;*****
;
;   Algoritmo de generación de trayectoria y cálculo orden de velocidad y posición
;   -----
;   Sin modificación
;*****

```

movimiento

```
INC16 TIEMPOMOV
```

```
call  doPosVel
tstfsz MODOTYPE
goto  vmove

```

pmove

```
movfp UNO,WREG
cpfseq BANDERAMOV
goto  phase2

```

phase1

```
MOVFP32  MOVDEL,MOVTMP
MOV32 OPOSICION,MOVDEL
ADD32 HMOVVAL,MOVDEL
SUB32 MOVPPBUF,MOVDEL
btfs  SIGNOMOV,MSB
goto  mpos1
NEG32 MOVDEL

```

mpos1

```
btfs  MOVDEL+B3,MSB
goto  speedup
TFSZ16 T1

```

```
goto  t2net1
NEG32 A
ADD32 MOVDEL,MOVTMP
btfs  MOVTMP+B3,MSB
goto  triok
setf  T2+B0,F
setf  T2+B1,F
NEG32 A
call  undoPosVel
NEG32 A
call  doPosVel

```

triok

```
ADD16 TIEMPOMOV,T2
MOV16 T2,T1
incf  BANDERAMOV,F

```

```

        goto    mvok
t2net1
        setf    T2+B0,F
        setf    T2+B1,F
        ADD16  TIEMPOMOV,T2
        MOVFP32    MOVTMP,TMP
        RLC32  MOVTMP
        ADD32  TMP,MOVTMP
        ADD32  MOVDEL,MOVTMP
        btfss  MOVTMP+B3,MSB
        goto   trapok
        setf    PH2FLAT+B0,F
        setf    PH2FLAT+B1,F
trapok
        ADD16  T2,PH2FLAT
        SUB16  T1,PH2FLAT
        incf   BANDERAMOV,F
        goto   mvok
speedup
        MOVFP32    V,MOVTMP
        SUB32  MOVVBUF,MOVTMP
        btfss  SIGNOMOV,MSB
        goto   mpos
        NEG32  MOVTMP
mpos
        btfss  MOVTMP+B3,MSB
        goto   mvok
        TFSZ16 T1

        goto   mvok
        call   undoPosVel
        CLR32  A
        call   doPosVel
        setf   T1+B0,F
        setf   T1+B1,F
        ADD16  TIEMPOMOV,T1
mvok
        MOV24  MOVVBUF+B1,POSICION
        MOV24  MOVVBUF+B0,VELOCIDAD
        return

phase2
        TFSZ16 PH2FLAT
        goto   flat
        TFSZ32 MOVVBUF
        goto   mready
        clrf   BANDERAMOV,F
        bcf    ESTADOMOV,bit6
        bcf    ESTADOMOV,bit5
        CLR32  A
        MOV16  TIEMPOMOV,TAU
        MOV32  OPOSICION,MOVVBUF
        ADD24  MOVVAL,MOVVBUF+B1
        goto   mready
flat
        setf   MOVTMP+B0,F

```

```

setf    MOVTMP+B1,F
ADD16  MOVTMP,PH2FLAT
TFSZ16 PH2FLAT
goto   mready
clrf   A+B3,F
MOV24  AL,A
clrf   WREG,F
cpfseq SIGNOMOV
goto   mready
NEG32  A
mready
MOV24  MOVVBUF+B1,POSICION
MOV24  MOVVBUF+B0,VELOCIDAD
return
vmove
MOVFP32    MOVVAL,MOVTMP
SUB32     MOVVBUF,MOVTMP
btfss    SIGNOMOV,MSB
goto     vmpos
NEG32    MOVTMP
vmpos
btfss    MOVTMP+B3,MSB
goto     vmoveok
CLR32   A
MOV32   MOVVAL,MOVVBUF

clrf    BANDERAMOV,F
bcf     ESTADOMOV,bit5
MOV16   TIEMPOMOV,TAU
TFSZ32  MOVVAL
goto    vmoveok
bcf     ESTADOMOV,bit6

vmoveok
MOV24  MOVVBUF+B1,POSICION
MOV24  MOVVBUF+B0,VELOCIDAD
return

```

```

;*****
;
;   Aceleración
;   -----
;   Sin modificación
;*****
doPosVel

```

```

ADD32  MOVVBUF,MOVVBUF
ADD32  A,MOVVBUF
MOVFP32    A,MOVTMP
RRC32  MOVTMP
ADD32  MOVTMP,MOVVBUF

return

```

undoPosVel

```
SUB32 A,MOVVBUF
SUB32 MOVVBUF,MOVPBUF
MOVFP32 A,MOVTMP
RRC32 MOVTMP
SUB32 MOVTMP,MOVPBUF
```

return

if _SERVO_PID

```
.*****
;
;   Calculo del PID
;   -----
;   Modificada y adaptada
;*****
ejecutarServo
```

MOV16 ERRORPOS,U0

```
LOADAB U0,KP
call Dmult ;179 ciclos de reloj
MOVFP32 DPX,Y
clrf WREG,F
cpfsgt BANDERASAT
call doIntegral
```

```
LOADAB INTEGRAL,KI
call Dmult ;179 ciclos de reloj
```

ADD32 DPX,Y

MOVFP16 U0,AARG

SUB16 U1,AARG

```
MOVFP16 KV,BARG
call Dmult ;179 ciclos de reloj
ADD32 DPX,Y
MOV16 U0,U1
```

;fin del calculo

```
clrf WREG,F
cpfsgt NUMCORRIMIENTO
goto grabok
movfp NUMCORRIMIENTO,TMP
```

grabloop

```
RLC32 Y
decfsz TMP,F
goto grabloop
```

grabok

```
clrf BANDERASAT,F
btfsc Y+B3,MSB
goto negs
```

```

poss
    movfp Y+B2,WREG
    andlw 0x80
    iorwf Y+B3,F
    clrf WREG,F
    cpfsgt Y+B3
    goto zero6bits
    incf BANDERASAT,F
    clrf Y+B3,F
    movlw 0x7F
    movpf WREG,Y+B2
    setf Y+B1,F
    setf Y+B0,F
    goto zero6bits

negs
    movfp Y+B2,WREG
    iorlw 0x7F
    andwf Y+B3,F
    setf WREG,F
    cpfslt Y+B3
    goto zero6bits

    setf BANDERASAT,F
    setf Y+B3,F
    clrf Y+B2,F
    bsf Y+B2,MSB
    clrf Y+B1,F
    clrf Y+B0,F

zero6bits
    MOV24 0X80+B1,YPWM+B0 ;0X80=Y

ejecutarTorque
    movlw 0xE0
    andwf YPWM+B0,F
    btfsc YPWM+B1,MSB
    goto tmlimit

tplimit
    bcf DIR,bit3
    btfss ESTADOEXT,bit6
    goto mplimitok
    CLR32 YPWM
    goto mplimitok

tmlimit
    bsf DIR,bit3
    btfss ESTADOEXT,bit5
    goto mplimitok
    CLR32 YPWM

mplimitok
    btfss YPWM+B1,MSB
    goto testmax
    comf YPWM+B0,F
    comf YPWM+B1,F
    movlw 0xE0
    andwf YPWM+B0,F

testmax
    RLC16 YPWM
    clrf TMP+B2,F

```

```

    clrf    YPWM+B2,F
    clrf    YPWM+B3,F
    MOVFP16    YPWMAX,TMP
    SUB24    YPWM,TMP
    btfss   TMP+B2,MSB
    goto    testmin
    MOV16   YPWMAX,YPWM
    goto    limitok

```

testmin

```

    clrf    TMP+B2,F
    clrf    YPWM+B2,F
    clrf    YPWM+B3,F
    MOVFP16    YPWMIN,TMP
    SUB24    YPWM,TMP
    btfsc   TMP+B2,MSB
    goto    limitok
    MOV16   YPWMIN,YPWM

```

limitok

```

    movlb   BANK3
    movfp   YPWM+B0,PW1DCL
    movfp   YPWM+B1,PW1DCH

```

```

    movlb   BANK0
    btfss   DIR,bit3
    goto    cero
    bsf     PORTA,bit3
    return

```

cero

```

    bcf     PORTA,bit3
    return

```

```

;*****
;
;   Evalua la Integral de los Servocalculos
;   -----
;
;   Sin modificaciones
;*****

```

doIntegral

```

    ADD16  U0,INTEGRAL

```

```

    return

```

```

    endif

```

```

    if PUERTOSERIAL

```

```

;*****
;
;   Utilerias para la comunicaci3n serial
;   -----
;
;   I/O Seriales y Funciones de Utilerias
;*****

```

IdleFunction

```

    clrwdt

```

```

return

;*****
; Busca en la tabla de instrucciones y las ejecuta
; -----
; aumentada y adaptada
;*****
ejecutarComando

    movlw  CMD_TABLE & 0xFF
    movpf  WREG,TBLPTRL
    movlw  PAGE_CMD_TABLE
    movpf  WREG,TBLPTRH

    tablrd 1,1,CMDTEMP
tryNextCmd

    tablrd 0,1,CMDTEMP
    tlrld 1,CMDPTRH
    tablrd 0,1,CMDPTRL

    movfp  CMDTEMP,WREG
    cpfslt CERO

    goto  noCommand

    cpfseq CMDCHAR
    goto  tryNextCmd

    call  PutChar

    movfp  CMDPTRH,PCLATH
    movfp  CMDPTRL,PCL
    NOP
cmdFinish
    call  PutChar

    movlw  CR
    call  PutChar
    goto  PollingLoop

noCommand
    movlw  CMD_MALO
    goto  cmdFinish

;*****
; Esta instrucción no ejecuta nada es usado para probar la comunicación
; -----
; sin modificación
;*****
nulo

    movlw  CMD_OK
    goto  cmdFinish

```

```

*****
;
;   M [-8388608,8388607]
;
;   -----
;
;   sin modificación
*****
mover
    if      DECIO
        call  GetDecVal
    else
        call  GetVal
    endif

    btfsc   ESTADOMOV,bit7
    goto    bufoverflow
MOV24 VALBUF,NMOVVAL
    bsf     ESTADOMOV,bit7
    movlw   CMD_OK
    goto    cmdFinish

bufoverflow
    movlw   CMD_MALO
    goto    cmdFinish

*****
;
;   O [P,V]
;
;   -----
;
;   sin modificación
*****
cambModo

    call    IdleFunction
    call    GetChk
    cpfseq  UNO
    goto    cambModo
    call    GetChar
    movpf   WREG,VALARREGLOB
    clrf    MODOTYPE,F

testP
    movlw   'P'
    cpfseq  VALARREGLOB
    goto    testV
    goto    modeok

testV
    movlw   'V'
    cpfseq  VALARREGLOB
    goto    testT
    incf    MODOTYPE,F
    goto    modeok

testT
    movlw   'T'
    cpfseq  VALARREGLOB
    goto    modeerror
    setf    MODOTYPE,F
    clrf    SERVOBANDERA,F

```



```

        goto    modeok
modeerror
        movlw  CMD_MALO
        goto   cmdFinish
modeok
        movfp  VALARREGLOB,WREG
        call   PutChar

        movlw  CMD_OK
        goto   cmdFinish

```

```

;*****
;
;   Establece parámetros
;   -----
;
;   sin modificación
;*****
estabParametros

```

```

        call   GetPar

        movlw  NUMPAR
        cpfst  VALBUF+B0
        goto   Serror

        movlw  PAR_TABLE & 0xFF
        movpf  WREG,TBLPTRL
        movlw  PAGE PAR_TABLE
        movpf  WREG,TBLPTRH

        tablrd 1,1,PARTEMP

```

setNextPar

```

        tlrd   1,PARTEMP
        tablrd 0,1,PARLEN
        tablrd 0,1,PARPTR

        movlw  NUMPAR
        cpfst  PARTEMP
        goto   Serror

        movfp  PARTEMP,WREG
        cpfseq VALBUF+B0
        goto   setNextPar

        movfp  PARPTR,WREG
        movfp  WREG,FSR1

        if     DECIO
            call  GetDecVal
        else
            call  GetVal
        endif

        movlw  VALBUF

```

```
movfp WREG,FSR0
AUTOINC
```

```
setGetMore
```

```
movfp INDF0,INDF1
decf  PARLEN,F
tstfsz PARLEN
goto  setGetMore
```

```
AUTONO
```

```
movlw CMD_OK
goto  cmdFinish
```

```
Serror
```

```
movlw CMD_MALO
goto  cmdFinish
```

```
.*****
;
;   R [O,FF]      Regresa el valor del parametro indicado
;   -----
;   sin modificación
;
;*****
```

```
leerParametros
```

```
call  GetPar
```

```
movlw NUMPAR
cpfst VALBUF+B0
goto  Rerror
```

```
movlw PAR_TABLE & 0XFF
movfp WREG,TBLPTRL
movlw PAGE PAR_TABLE
movfp WREG,TBLPTRH
```

```
tablrd 1,1,PARTEMP
```

```
readNextPar
```

```
tlrd 1,PARTEMP
tablrd 0,1,PARLEN
tablrd 0,1,PARPTR
```

```
movlw NUMPAR
cpfst PARTEMP
goto  Rerror
```

```
movfp PARTEMP,WREG
cpfseq VALBUF+B0
goto  readNextPar
```

```
movfp PARPTR,WREG
movfp WREG,FSR1
```

```

movlw VALBUF
movfp WREG,FSR0

AUTOINC

CLR24 VALBUF

readGetMore
movfp INDF1,INDF0
decf PARLEN,F
tstfsz PARLEN
goto readGetMore

AUTONO

if DECIO
call PutDecVal
else
call PutVal
endif

movlw CMD_OK
goto cmdFinish

Rerror
movlw CMD_MALO
goto cmdFinish

;*****
; C Regresa el tiempo del presente movimiento
; -----
; sin modificación
;*****
do_shutter
MOV24 POSICION,CPOSICION
MOV24 VELOCIDAD,CVELOCIDAD
MOV24 MPOSICION,CMPOSICION
MOV24 MVELOCIDAD,CMVELOCIDAD
clrf VALBUF+B2,F
MOV16 TIEMPOMOV,VALBUF

if DECIO
call PutDecVal
else
call PutVal
endif

movlw CMD_OK
goto cmdFinish

;*****
; P Regresa la última orden de posición
; -----
; sin modificación
;*****

```

```

*****
;
leerComPosicion

    MOV24 CPOSICION,VALBUF

    if    DECIO
        call    PutDecVal
    else
        call    PutVal
    endif

    movlw  CMD_OK
    goto   cmdFinish

```

```

*****
;
;    V    Regresa la última orden de velocidad
;    -----
;    sin modificación
*****
;
leerComVelocidad

```

```

MOV24 CVELOCIDAD,VALBUF

```

```

    if    DECIO
        call    PutDecVal
    else
        call    PutVal
    endif

    movlw  CMD_OK
    goto   cmdFinish

```

```

*****
;
;    p    Regresa la ultima posición medida
;    -----
;    sin modificación
*****
;
leerPosActual

```

```

MOV24 CMPOSICION,VALBUF

```

```

    if    DECIO
        call    PutDecVal
    else
        call    PutVal
    endif

```

```

    movlw  CMD_OK
    goto   cmdFinish

```

```

*****
;
;    v    Regresa la ultima velocidad medida
;    -----
;    sin modificación
*****
;

```

```

leerVelActual
    MOV24 CMVELOCIDAD,VALBUF
    if    DECIO
        call    PutDecVal
    else
        call    PutVal
    endif

    movlw  CMD_OK
    goto   cmdFinish

.*****
;
;   X      El registro de estado externo es regresado
;   -----
;
;   sin modificación
.*****
leerEstExt
    bsf    CPUSTA,GLINTD
    movfp  ESTADOEXT,WREG
    clrf   ESTADOEXT,F
    bcf    CPUSTA,GLINTD
    call   PutHex

    movlw  CMD_OK
    goto   cmdFinish

.*****
;
;   Y      El registro de estado de movimiento es regresado
;   -----
;
;   sin modificación
.*****
leerEstadoMov

    movfp  ESTADOMOV,WREG
    call   PutHex

    movlw  CMD_OK
    goto   cmdFinish

.*****
;
;   I      Regresa la ultima posición del indice capturado
;   -----
;
;
.*****
leerPosIndice
    MOV24 INDEXPOS,VALBUF
    if    DECIO
        call    PutDecVal
    else
        call    PutVal
    endif

    movlw  CMD_OK

```

```

        goto    cmdFinish

;*****
;      H [-8388608,8388607]
;      -----
;      sin modificación
;*****
establecePos
    if        DECIO
        call   GetDecVal
    else
        call   GetVal
    endif

    MOV24 VALBUF,POSICION
    MOV24 VALBUF,MPOSICION
    CLR32 Y

    movlw    CMD_OK
    goto     cmdFinish

;*****
;      Z      Ejecuta un reestablecimiento por programa
;      -----
;      sin modificación
;*****
reestablecer
    movlw    CMD_OK
    call     PutChar
    goto     inicio

;*****
;      s      Detiene el servosistema
;      -----
;      sin modificación
;*****
paro
    clrf     SERVOBANDERA,F
    movlw    CMD_OK
    goto     cmdFinish

;*****
;      Captura trayectoria
;      -----
;      Realizado totalmente por el alumno
;*****
captura
    movlb    BANK0

capturaciclo
    call     IdleFunction
    call     GetChk
    cpfseq   UNO

```

```

    goto    capturaciclo
    call    GetChar
    movpf  WREG,VALARREGLOB
captura1
    movlw  '1'
    cpfseq VALARREGLOB
    goto   captura2
    movlw  0x01
    movwf  NUMDATO
    goto   capturaok
captura2
    movlw  '2'
    cpfseq VALARREGLOB
    goto   captura3
    movlw  0x02
    movwf  NUMDATO
    goto   capturaok
captura3
    movlw  '3'
    cpfseq VALARREGLOB
    goto   captura4
    movlw  0x03
    movwf  NUMDATO
    goto   capturaok
captura4
    movlw  '4'
    cpfseq VALARREGLOB
    goto   capturaerror
    movlw  0x04
    movwf  NUMDATO
    goto   capturaok

capturaerror
    clrf   BANDERAENVIA
    movlw  CMD_MALO
    goto   cmdFinish
capturaok
    setf   BANDERAENVIA
    movfp  VALARREGLOB,WREG
    call   PutChar

    movlw  CMD_OK
    goto   cmdFinish

```

```

;*****
;
; Rutina nueva para configurar la resolución
; -----
;
; Realizado totalmente por el alumno
;*****
resolucion
    movlb  BANK0

resolucionciclo
    call   IdleFunction
    call   GetChk

```

```

    cpfseq UNO
    goto resolucionciclo
    call GetChar
    movpf WREG,VALARREGLOB
prueba1
    movlw '1'
    cpfseq VALARREGLOB
    goto prueba4
    bcf PORTA,bit2
    goto resolucionok
prueba4
    movlw '4'
    cpfseq VALARREGLOB
    goto resolucionerror
    bsf PORTA,bit2
    goto resolucionok

resolucionerror
    movlw CMD_MALO
    goto cmdFinish
resolucionok
    movfp VALARREGLOB,WREG
    call PutChar

    movlw CMD_OK
    goto cmdFinish

```

```

,*****
;
; Rutina para habilitar el servomecanismo
; -----
; Realizado totalmente por el alumno
,*****
;

```

```

habilita
    movlb BANK1
    bcf PORTC,bit0

    movlw 0x2F
retar decfsz WREG,F
    goto retar

    bcf PORTC,bit6 ;nuevo
    movlw 0x0F
retar1 decfsz WREG,F
    goto retar1
    bsf PORTC,bit6 ;nuevo

    movlw CMD_OK
    goto cmdFinish

```

```

,*****
;
; Rutina para deshabilitar el servomecanismo
; -----
; Realizado totalmente por el alumno
,*****
;

```



```

deshabilita
    movlb BANK1
    bsf   PORTC,bit0

    clrf  SERVOBANDERA,F
    clrf  BANDERAMOV,F
    clrf  ESTADOMOV,F

    movlb BANK3
    movlw 0x00
    movwf PW1DCL
    movwf PW1DCH

    movlw CMD_OK
    goto  cmdFinish

```

```

;*****
;
;   Obtiene el número de parámetros de [0,FF] por el
;   -----
;   el puerto serie y lo fija en VALBUF+B0
;*****

```

GetPar

```

CLR24 VALBUF
call   GetHex
movfp  HEXVAL,WREG
andlw  0x0f
movpf  WREG,VALBUF+B0
swapf  VALBUF+B0,F

call   GetHex
movfp  VALBUF+B0,WREG

addwf  HEXVAL,W
movpf  WREG,VALBUF+B0

```

return

```

;*****
;
;   Obtiene el caracter del registro del puerto serie
;   -----
;   Sin modificaciones
;*****

```

GetChar

```

movlb  BANK0
movpf  RCREG,WREG
return

```

```

;*****
;
;   Envía caracter por el puerto serie
;   -----
;   Sin modificaciones
;*****

```

PutChar

```

        movlb  BANK1

bufwait
    btfss   PIR, TXIF      ;?Está el registro de transmisión vacío;
    goto   bufwait       ;No

        movlb  BANK0      ;Si

shfwait
    btfss   TXSTA, TRMT   ;Está el registro de turno vacío
    goto   shfwait       ;No

        movpf  WREG, TXREG ;Si

        return

```

```

;*****
;
;   Checa si fue recibido un caracter por el puerto serie
;
;   -----
;
;   Sin modificaciones
;*****
GetChk
    movlb  BANK1
    movpf  PIR, WREG
    andlw  CHARACTERLISTO ;Regresa status en wreg
    return ;Variable de recibido lleno o vacío

```

```

;*****
;
;   Convierte un valor hex [0,F] en WREG a ASCII
;   El nibble alto es asumido como cero
;
;   -----
;
;   Sin modificaciones
;*****
    if     DECIO
PutDec
        addlw  0x30
        call   PutChar
        return

    endif

```

```

;*****
;
;   Convierte el valor de wreg de ASCII a hex y lo
;   envía por el puerto serie
;
;   -----
;
;   Sin modificaciones
;*****
PutHex
    movpf  WREG, HEXVAL
    swapf  WREG, F
    andlw  0X0F
    movpf  WREG, HEXTMP
    negw   WREG, F

```

```

    addlw 0x09
    btfss WREG,MSB
    goto puth20
    movlw 'A'-0x0A
    addwf HEXTMP,W
    goto puth25
puth20
    movlw '0'
    addwf HEXTMP,W
puth25
    call PutChar
    movfp HEXVAL,WREG
    andlw 0X0F
    movfp WREG,HEXTMP
    negw WREG,F
    addlw 0x09
    btfss WREG,MSB
    goto putl20
    movlw 'A'-0x0A
    addwf HEXTMP,W
    goto putl25
putl20
    movlw '0'
    addwf HEXTMP,W
putl25
    call PutChar

    return

```

```

,*****
;
;   Envia una cadena de caracteres por el puerto serie
;   -----
;
;   Sin modificaciones
,*****
PutStr
    tablrd 1,1,VALARREGLOA
GetNextPair

    tlrd 1,VALARREGLOA
    tablrd 0,1,VALARREGLOB

    movfp VALARREGLOA,WREG
    cpfseq CERO
    goto putH
    return
putH
    call PutChar

    movfp VALARREGLOB,WREG
    cpfseq CERO
    goto putL
    return
putL
    call PutChar
    goto GetNextPair

```

```

;*****
;
;   Recibe un caracter ASCII por el puerto serie y lo
;   convierte en un valor numerico
;   -----
;   Sin modificaciones
;*****

```

GetHex

getnxt

```

call   IdleFunction
call   GetChk
cpfq   UNO
goto   getnxt

clrf   HEXSTAT,F
call   GetChar
movpf  WREG,HEXVAL
call   PutChar
movlw  CR
subwf  HEXVAL,W
tstfsz WREG
goto   gth10
goto   gthCR

```

gth10

```

movfp  HEXVAL,WREG
sublw  '9'
btfss  WREG,MSB
goto   gth20

```

gth20

```

movlw  0x0f
andwf  HEXVAL,F
clrf   HEXSTAT,F
return

```

gthCR

```

movlw  0x01
movpf  WREG,HEXSTAT
return

```

```

;*****
;
;   Obtiene el valor de [800000,7FFFFFF] desde el puerto
;   -----
;   serie y lo fija en VALBUF
;*****

```

```

if     DECIO
else

```

GetVal

```
CLR24 VALBUF
```

getnext

```
call   GetHex
```

```

        movlw 0x01
        cpfseq HEXSTAT
        goto shift
        return
shift
        swapf VALBUF+B2
        movfp VALBUF+B2,WREG
        andlw 0x0f
        movpf WREG,VALBUF+B2
        swapf VALBUF+B1
        movfp VALBUF+B1,WREG
        andlw 0x0f
        addwf VALBUF+B2
        movfp VALBUF+B1,WREG
        andlw 0xf0
        movpf WREG,VALBUF+B1
        swapf VALBUF+B0
        movfp VALBUF+B0,WREG
        andlw 0x0f
        andwf VALBUF+B1
        movfp VALBUF+B0,WREG
        andwf 0xf0
        addwf HEXVAL,W
        movpf WREG,VALBUF+B0

        goto getnext

```

endif

```

*****
;
;   Obtiene un valor de [-8388608,8388607] por el puerto
;   serie y lo fija en VALBUF
;   -----
;   Sin modificaciones
*****
        if DECIO
GetDecVal
        CLR24 VALBUF

        call GetDec
        setf DECSIGNO,F
        movlw DEC_MN
        cpfseq DECSTAT
        clrf DECSIGNO,F

getdecnext
        call GetDec

        movlw DEC_CR
        cpfseq DECSTAT
        goto mul10
        goto fixsigno
mul10

```

```

RLC24 VALBUF
MOV24 VALBUF,DVALBUF
RLC24 VALBUF
RLC24 VALBUF
ADD24 DVALBUF,VALBUF
CLR24 DVALBUF
movfp  DECVAL,WREG
movpf  WREG,DVALBUF+B0
ADD24 DVALBUF,VALBUF

goto   getdecnext
fixsigno
clrf   WREG,F
cpfsgt DECSIGNO
return
NEG24 VALBUF
return

endif

;*****
;
;   Recibe un ASCII decimal por el puerto serie y lo
;   -----
;
;   convierte en un valor numerico
;*****
if     DECIO
GetDec

getdecnxt
call   IdleFunction
call   GetChk
cpfseq UNO
goto   getdecnxt

call   GetChar
movpf  WREG,DECVAL
call   PutChar

movlw  CR
subwf  DECVAL,W
cpfslt CERO
goto   gtdCR
movlw  MN
subwf  DECVAL,W
cpfslt CERO
goto   gtdMN
movlw  SP
subwf  DECVAL,W
cpfslt CERO
goto   gtdSP
gtd09
movlw  0x0f
andwf  DECVAL,F
clrf   DECSTAT,F
return

```

```

gtdCR
    movlw DEC_CR
    movpf WREG,DECSTAT
    return
gtdMN
    movlw DEC_MN
    movpf WREG,DECSTAT
    return
gtdSP
    movlw DEC_SP
    movpf WREG,DECSTAT
    return

    endif

```

```

,*****
;      Envia el valor de VALBUF [800000,7FFFFFFF] por el
;      puerto serie
;      -----
;      Sin modificaciones
,*****
,
    if      DECIO
    else

```

PutVal

```

    movfp VALBUF+B2,WREG
    call  PutHex
    movfp VALBUF+B1,WREG
    call  PutHex
    movfp VALBUF+B0,WREG
    call  PutHex

    return

    endif

```

```

,*****
;      Envia el valor de VALBUF [-8388608,8388607] por el
;      puerto serie
;      -----
,*****
,
    if      DECIO

```

PutDecVal

```

    btfss VALBUF+B2,MSB
    goto  pdpos
    NEG24 VALBUF

    movlw MN
    call  PutChar
    goto  pddigits
pdpos

```

```

        movlw SP
        call  PutChar

pddigits
        movlw (DEC_TABLE & 0XFF)
        movpf WREG,TBLPTRL
        movlw PAGE DEC_TABLE
        movpf WREG,TBLPTRH

        tablrd 0,1,DVALBUF+B0
readNextDec

        tlrld 0,DVALBUF+B0
        tablrd 1,1,DVALBUF+B1
        tablrd 0,1,DVALBUF+B2

        setf  WREG,F
        cpfseq DVALBUF+B0
        goto  getdigit
        goto  unitsposicion
getdigit
        incf  DVALBUF+B0,F
        setf  DECVAL,F
inc
        incf  DECVAL,F
        SUB24 DVALBUF,VALBUF
        btfss VALBUF+B2,MSB
        goto  inc

        ADD24 DVALBUF,VALBUF

        movfp DECVAL,WREG
        call  PutDec

        goto  readNextDec

unitsposicion
        movfp VALBUF+B0,WREG
        call  PutDec

        return

        endif

```

```

;*****
;
;   TABLAS
;   -----
;   Modificada y aumantada
;*****
;
;   CMD_START CMD_TABLE

```

```

CMD_TABLE

```

```

        CMD_DEF nulo,NULO
        CMD_DEF mover,MOVER

```



```
CMD_DEF cambModo,CAMBMODO
CMD_DEF estabParametros,ESTABPARAMETROS
CMD_DEF leerParametros,LEERPARAMETROS
CMD_DEF do_shutter,DO_SHUTTER
CMD_DEF leerComPosicion,LEERCOMPOSICION
;CMD_DEF leerComVelocidad,LEERCOMVELOCIDAD
CMD_DEF leerPosActual,LEERPOSACTUAL
;CMD_DEF leerVelActual,LEERVELACTUAL
CMD_DEF leerEstExt,LEERESTEXT
CMD_DEF leerEstadoMov,LEERESTADOMOV
;CMD_DEF leerPosIndice,LEERPOSINDICE
;CMD_DEF establecePos,ESTABLECEPOS
CMD_DEF reestablecer,REESTABLECER
CMD_DEF paro,PARO
CMD_DEF resolucion,RESOLUCION
CMD_DEF habilita,HABILITA
CMD_DEF deshabilita,DESHABILITA
      CMD_DEF captura,CAPTURA
```

```
CMD_END
```

```
PAR_TABLE DATA 0X0003
          DATA VL
          DATA 0X0103
          DATA AL
          DATA 0X0202
          DATA KP
          DATA 0X0302
          DATA KV
          DATA 0X0402
          DATA KI
          DATA 0X0501
          DATA IM
          DATA 0X0602
          DATA FV
          DATA 0X0702
          DATA FA
          DATA NUMPAR
```

```
if      DECIO
```

```
DEC_TABLE DATA 0X423F
          DATA 0X000F
          DATA 0X869F
          DATA 0X0001
          DATA 0X270F
          DATA 0X0000
          DATA 0X03E7
          DATA 0X0000
          DATA 0X0063
          DATA 0X0000
          DATA 0X0009
          DATA 0X0000
          DATA 0XFFFF
```

```
endif
```

endif

```
.*****  
;  
;   Escribe la trayectoria en el puerto serial  
;   -----  
;   Realizada por el alumno  
.*****
```

```
EnviaDatos  
    btfsc  ESTADOMOV,bit6      ;Esta en movimiento?  
    goto   EnviaDatos1       ;Si continúa  
    return ;No regresa
```

```
EnviaDatos1  
    btfsc  TIEMPOMOV+B0,bit0  
    goto   Manda1  
  
    setf   ENVIALISTO  
  
    movlw  0x01  
    cpfseq NUMDATO  
    goto   EnviaDatos2  
    MOV16 POSICION,BUFENVIA  
    goto   Manda
```

```
EnviaDatos2  
    movlw  0x02  
    cpfseq NUMDATO  
    goto   EnviaDatos3  
    MOV16 VELOCIDAD,BUFENVIA  
    goto   Manda
```

```
EnviaDatos3  
    movlw  0x03  
    cpfseq NUMDATO  
    goto   EnviaDatos4  
    MOV16 MPOSICION,BUFENVIA  
    goto   Manda
```

```
EnviaDatos4  
    MOV16 MVELOCIDAD,BUFENVIA
```

```
Manda  
    movfp  BUFENVIA+B0,WREG  
    call   PutChar  
    return
```

```
Manda1  
    movfp  BUFENVIA+B1,WREG  
    call   PutChar  
    return
```

END