



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

Controlador Estándar de Movimiento Multieje con base en FPGA

TESIS

Que como parte de los requisitos para obtener el grado de:

Maestro en Ciencias

Presenta:

Ing. Fortino Mendoza Mondragón

Dirigido por:

Dr. Juvenal Rodríguez Reséndiz

Dr. Juvenal Rodríguez Reséndiz

Presidente

Dr. Edgar Alejandro Rivas Araiza

Secretario

M. en C. José Marcelino Gutiérrez Villalobos

Vocal

M. en C. Miguel Ángel Martínez Prado

Suplente

Dr. Israel Marcos Santillán Méndez

Suplente

Dr. Aurelio Domínguez González

Director de la Facultad

Firma

Firma

Firma

Firma

Firma

Dr. Irineo Torres Pacheco

Director de Investigación y Posgrado

Controlador estándar de movimiento multieje con base en FPGA

Ing. Fortino Mendoza Mondragón

RESUMEN

El control de movimiento de motores tiene una amplia área de aplicación en la automatización industrial principalmente en el área de sistemas robóticos, el objetivo del presente trabajo es el diseño de un sistema de control de movimiento con base en la tecnología FPGA para el control de motores de DC, se desarrolló un algoritmo de control en cascada el cual puede operar en el modo velocidad y posición que además incluye ganancias de pre alimentación de velocidad y aceleración para mejorar el desempeño de seguimiento de perfiles de velocidad, el algoritmo desarrollado es altamente flexible y robusto, uno de los principales objetivos es el diseño de IP cores en VHDL que comúnmente son empleados en el control de movimiento como lo son controladores tipo PID, unidades de cuadratura y un generador de perfiles trapezoidales, además se describe una metodología para su diseño, simulación, descripción y validación utilizando MATLAB/Simulink y VHDL.

Palabras Clave: Control de movimiento, motor de DC, control en cascada feedforward, FPGA.

SUMMARY

The motion control has a wide area of application in industrial automation mainly in the control of mechatronic systems, the purpose of this work is the design of a motion control system based on FPGA technology for DC motor control, a cascade control algorithm was developed which can operate in speed and position mode and also includes acceleration and velocity feedforward gains to improve the performance of tracking in speed motion profiles, the developed algorithm is highly flexible and robust, a major objective is to design IP cores that are commonly used in motion control such as PID type controllers, encoder units and trapezoidal motion profile generator; also a methodology for the design, simulation, description and validation of the algorithm using MATLAB/Simulink and VHDL is described.

Keywords: motion control, DC motor, feedforward cascade control, FPGA.

AGRADECIMIENTOS

Cada descubrimiento y progreso, cada incremento en la riqueza de la humanidad, se origina en el conjunto del trabajo manual y cerebral de ayer y hoy. Entonces, ¿Qué derecho asiste alguien de apoderarse de una partícula de ese todo y decir: "Esto es mío y no vuestro"?

“Piotr Kropotkin”

A mi familia gracias por todo su apoyo incondicional.

ÍNDICE DE CONTENIDO

RESUMEN	i
SUMMARY	ii
AGRADECIMIENTOS	iii
ÍNDICE DE CONTENIDO	iv
ÍNDICE DE TABLAS	vii
ÍNDICE DE FIGURAS.....	viii
I INTRODUCCIÓN	1
I.1 Justificación.....	3
I.2 Planteamiento del problema.....	4
I.3 Hipótesis y objetivos	6
I.3.1 Hipótesis general.....	6
I.3.2 Objetivo general.....	6
I.3.3 Objetivos específicos	6
II REVISIÓN DE LITERATURA	7
II.1 Antecedentes	7
II.1.1 Antecedentes UAQ	7
II.1.2 Antecedentes generales.....	8
III METODOLOGÍA	11
III.1 Marco Teórico	11
III.1.1 Control de Movimiento.....	11
III.1.2 El control Automático.....	12
III.1.3 El estado actual del control de movimiento.....	14
III.1.4 El Servosistema.....	15

III.1.5	Servomotores	17
III.1.6	Motores de corriente directa o DC.....	18
III.1.7	Motor de DC con escobillas.....	19
III.1.8	Motores de DC Sin Escobillas o BLDC (Brushless DC).....	21
III.1.9	Características y desempeño de los motores de DC	22
III.1.10	Amplificadores para motores de DC con escobillas	22
III.1.11	Amplificadores Lineales	22
III.1.12	Amplificador por PWM	24
III.1.13	Amplificador para motores sin escobillas	25
III.1.14	Servoamplificadores.....	26
III.1.15	Principales algoritmos de control para motores en la industria	27
III.1.16	Decodificador Incremental (Encoder).....	30
III.1.17	Estimación de la velocidad a partir de un encoder incremental.....	35
III.1.18	Controladores de velocidad.....	39
III.1.19	Controladores de posición.....	41
III.1.20	Controlador PID discreto	46
III.1.21	Generación de Trayectorias.....	51
III.2	Metodología.....	54
III.2.1	Modelado e Identificación Del Sistema.....	54
III.2.2	Discretización y cuantificación.....	83
III.2.3	Descripción de Hardware.....	94
IV	RESULTADOS Y DISCUSIÓN	104
IV.1	Identificación del sistema	104
IV.2	Controlador PI de velocidad.....	106
IV.1	Controlador P-PI de posición	107

IV.2 Conclusiones.....	112
BIBLIOGRAFÍA	113
Anexo 1	123
Anexo 2.....	124
Anexo 3.....	130

ÍNDICE DE TABLAS

Tabla III.1 Tipos comunes de motores y sus características.....	18
Tabla III.2 Variables usualmente retroalimentadas y sus transductores (Mentz, 2000).....	29
Tabla III.3 Aproximaciones de la integral utilizando muestras presentes y pasadas.....	49
Tabla III.4 Características principales de los motores modelados.....	63
Tabla III.5. Características principales del servoamplificador B15A8.....	67
Tabla III.6. Características principales del DAC LTC2642-16	100
Tabla III.7. Características principales del ADC AD7687.	101
Tabla IV.1. Resultados de síntesis de controlador de movimiento.....	110

ÍNDICE DE FIGURAS

Figura II.1 Controlador en un chip propuesto para el control de movimiento (Cho et al., 2009). .	9
Figura II.2 Diagrama a bloques de un típico servo controlador con base en FPGA/DSP (K. Boggarpu et al., 2010).....	10
Figura III.1 Desarrollo del área de control de movimiento (Alahakoon, 2000).	11
Figura III.2. Lazo de control de un servosistema para un motor eléctrico.	15
Figura III.3 Estructura de un servosistema industrial (Nakamura et al., 2004).	16
Figura III.4 Ley de Ampere de la fuerza magnética.	19
Figura III.5. Motor de corriente directa con escobillas.....	20
Figura III.6 Motor de corriente directa con escobillas.....	20
Figura III.7 Características típicas de un motor de DC.	22
Figura III.8 Amplificador lineal para un motor de DC.....	23
Figura III.9 Circuito para el manejo de un motor por PWM (Wilamowski et al., 2011).	24
Figura III.10 Salida del PWM.....	25
Figura III.11 Diagrama a bloques de un controlador para motores sin escobillas (Wilamowski et al, 2011).	26
Figura III.12 Componentes de un codificador óptico.	31
Figura III.13 Patrón de las pistas y las señales generadas de un decodificador incremental.....	32
Figura III.14 Decodificación de las señales de cuadratura y detección del sentido de giro.	33
Figura III.15. Circuito para decodificación de las señales de cuadratura a 1X.	34
Figura III.16 Método “M”.....	36
Figura III.17 Método “T”.....	37
Figura III.18 Método “M/T”.....	38
Figura III.19 Controlador de velocidad industrial típico.	40
Figura III.20 Controlador de posición académico.	41

Figura III.21 Controlador de posición en cascada	42
Figura III.22 Controlador de posición con alimentación de velocidad.....	43
Figura III.23 Controlador de posición industrial feedforward.....	44
Figura III.24 Controlador de posición en configuración P+PDF.....	44
Figura III.25 Controlador de posición con lazo de velocidad PI y ganancias pre alimentadas de velocidad y aceleración (P-PI feedforward).	45
Figura III.26 Controlador de posición con lazo de velocidad PDF y ganancias pre alimentadas de velocidad y aceleración.....	45
Figura III.27 Comparación de los métodos para la aproximación de la integral. (a) Aproximación utilizando “Forward Euler”. (b) Aproximación utilizando “Backward Euler”. (c) Aproximación utilizando el método trapezoidal. (Levine, 2011).	48
Figura III.28 Respuesta en frecuencia de un filtro Butterworth utilizando los métodos de Tustin y Euler (Levine, 2011).	50
Figura III.29 Perfil de velocidad trapezoidal (González, 2006).	52
Figura III.30 Perfil de velocidad de curva-s (Bishop, 2011).	53
Figura III.31 Metodología de implementación de un algoritmo en FPGA.....	54
Figura III.32 Modelo del motor de imanes permanentes de DC.....	55
Figura III.33 Diagrama a bloques un motor de DC.	58
Figura III.34 Conexión estrella de un motor BLDC.	60
Figura III.35. Modelo del motor de DC en Simulink.	64
Figura III.36. Respuesta del motor de DC a una entrada escalón.....	64
Figura III.37. Respuesta de la corriente al aplicar el voltaje nominal.	65
Figura III.38. Eficiencia del motor de DC y su relacion Torque-Potencia-Velocidad.	66
Figura III.39. Modelo reducido del motor de DC incluyendo el lazo de torque.....	67
Figura III.40. Modelo reducido del motor de DC incluyendo la constante de amortiguamiento. 68	
Figura III.41. Modelo reducido del motor de DC descartando la constante de amortiguamiento.69	
Figura III.42. Control PI analógico con valores del servoamplificador B15A8N.....	70

Figura III.43. Filtro RC pasa bajas de segundo orden para el filtrado de la corriente.	71
Figura III.44. Modelo del servoamplificador B15A8.	72
Figura III.45. Modelo del servomotor y el servoamplificador.	73
Figura III.46. Modelo del sub-bloque del controlador PI.	74
Figura III.47. Controlador PI de velocidad del servomotor.	74
Figura III.48. Generador de perfiles trapezoidales.	76
Figura III.49. Modelo del controlador de movimiento.	76
Figura III.50. Respuesta a una entrada escalón de 16666.66PPR.	77
Figura III.51. Respuesta a una entrada escalón de 4000PPR.	78
Figura III.52. Respuesta del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleracion de 40000 pulsos/s ² y una posición final de 4000 PPR y sin las ganancias feedforward.	78
Figura III.53. Respuesta del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleracion de 40000 pulsos/s ² y una posición final de 4000 PPR y con la ganancia feedforward de velocidad.	79
Figura III.54. Respuesta del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleracion de 40000 pulsos/s ² y una posición final de 4000 PPR y con la ganancias feedforward de velocidad y aceleración.	80
Figura III.55. Esquema del controlador PID de posicion.	80
Figura III.56. Diagrama a bloques del controlador PID y su respuesta a una entrada escalón de 4000 PPR.	81
Figura III.57. Respuesta del controlador PID a un perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleracion de 40000 pulsos/s ² y una posición final de 4000 PPR , $\omega_c=16\text{rad/s}$ y $\Phi=70^\circ$	81
Figura III.58. Respuesta del controlador PID a un perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleracion de 40000 pulsos/s ² y una posición final de 4000 PPR, $\omega_c=160\text{rad/s}$ y $\Phi=70^\circ$	82
Figura III.59. Estimador de velocidad con LPF.	83
Figura III.60. Controlador discreto PI de velocidad.	84

Figura III.61. Generador de perfil trapezoidal de velocidad discreto.	84
Figura III.62. Modelo del controlador de movimiento en tiempo discreto.....	85
Figura III.63. Respuesta en tiempo discreto del controlador de velocidad a una entrada escalón equivalente a 500RPMs.	86
Figura III.64. Respuesta en tiempo discreto del controlador de velocidad a una entrada escalón equivalente a 500RPMs.	87
Figura III.65. Respuesta en tiempo discreto del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s ² y una posición final de 4000 PPR.	88
Figura III.66. Respuesta en tiempo discreto del perfil triangular con una velocidad de 100000 pulsos/s, una aceleración de 90000 pulsos/s ² y una posición final de 4000 PPR.	89
Figura III.67. Respuesta en tiempo discreto y cuantificada del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s ² y una posición final de 4000 PPR.	92
Figura III.68. Respuesta a un escalón de 4000 PPR en tiempo discreto y cuantificada en punto fijo.	93
Figura III.69. Comparación de la respuesta a un escalón de 4000 PPR en tiempo discreto y la respuesta del sistema cuantificado en punto fijo.....	93
Figura III.70. Arquitectura del controlador de movimiento propuesto.....	95
Figura III.71. Componentes del controlador de movimiento propuesto.....	95
Figura III.72. Diagrama a bloques del módulo QEI.	96
Figura III.73. Terminaciones para un decodificador incremental simple y diferencial.....	97
Figura III.74. Filtro digital.....	98
Figura III.75. Unidad de decodificación y detección del sentido de giro.	99
Figura III.76. Verificación del Filtro digital del core QEI.....	99
Figura III.77. Verificación del core de cuadratura QEI.....	100
Figura III.78. Verificación del proceso de escritura del DAC o “bit timing diagram”.....	101
Figura III.79. Verificación del proceso de escritura del ADC.....	102
Figura III.80. Verificación del generador de perfiles de velocidad.	103
Figura IV.1. Interfaz en LabVIEW para la identificación de sistemas.....	104

Figura IV.2. Respuesta del motor a una entrada escalón de 24V.	105
Figura IV.3. Comparación de la respuesta al escalón de modelo matemático y el modelo estimado.	105
Figura IV.4. Comparación de la respuesta al escalón del controlador de velocidad.	106
Figura IV.5. Comparación de la respuesta al escalón del controlador P-PI de posición.	107
Figura IV.6. Comparación de la respuesta al escalón del controlador P-PI de posición.	108
Figura IV.7. Respuesta del controlador en modo velocidad para el motor BLDC.	108
Figura IV.8. Respuesta del controlador en modo entrada escalon de posición para el motor BLDC.	109
Figura IV.9. Respuesta del controlador para un perfil trapezoidal el motor BLDC.	109
Figura IV.10. Estación de trabajo desarrollada.	111
Figura IV.11. Prototipo del controlador de movimiento.	111

CAPÍTULO 1:

I INTRODUCCIÓN

En la actualidad, se ha demostrado la necesidad que tiene el hombre de emplear la maquinaria en la vida cotidiana; más cuando se habla de procesos industriales donde influyen el transportamiento de piezas pesadas o maquinado de ellas, los procesos de manufactura, entre otros; por ello, se requiere el uso de instrumentos y herramientas de gran desempeño para realizar las diferentes actividades de control (velocidad, posicionamiento y aceleración), por ejemplo: bandas transportadoras, grúas móviles o fijas, tornos, empacadoras, brazos robóticos, etc., dichas herramientas de trabajo permiten obtener mejores resultados al aumentar los niveles de calidad y productividad. A la tecnología que realiza el control de dichos procesos o herramientas se le denomina servomecanismo; de acuerdo con la Real Academia Española (RAE) un servomecanismo tiene la siguiente definición:

“Sistema electromecánico que se regula por sí mismo al detectar el error o la diferencia entre su propia actuación real y la deseada”

De igual manera, los servosistemas o servomecanismos se han convertido en una parte importante de la industria electrónica y sus áreas relacionadas en la ingeniería, como son las disciplinas de computación, control e instrumentación y electrónica.

Los sistemas de control de movimiento juegan un papel importante en el control de los diferentes tipos de servosistemas. La mayoría de sistemas de control de movimiento se basan en un Procesador de Señales Digitales (DSPs), soluciones en un chip como la tecnología Field Programmable Gate Array (FPGA), también chips de uso dedicado o Application Specific Integrated Circuit (ASIC), o con base en un microprocesador (μP) y un microcontrolador (μC). Un sistema de control de movimiento multieje necesita muchas funciones con el fin de realizar de forma rápida y precisa las complejas tareas necesarias en el control de robots industriales y sistemas de automatización (Chan *et al.*, 2007), (Visioli *et al.*, 2002). Éstos incluyen la generación de perfiles como velocidad, interpolación, adquisición de datos, así como la

integración de uno o más lazos de control Proporcional Integral Derivativo (PID). Los métodos numéricos convencionales para estas funciones tienen aspectos difíciles como: complicadas multiplicaciones, divisiones y el cálculo de funciones trigonométricas de forma repetitiva (Cho *et al.*, 2009).

Bajo estas circunstancias, es necesario desarrollar controladores electrónicos con un alto poder de cómputo y, también, contar con sistemas de comunicación internos y externos de muy alta velocidad. Las soluciones On-Chip incluyen, en un solo chip, el algoritmo de control y los módulos para la interfaz con la etapa de potencia, tales como generadores de modulación por ancho de pulso (PWM), interfaces de cuadratura para encoder (QIE), entradas y salidas de propósito general (GPIO), etc. (Bueno *et al.*, 2009). Un FPGA proporciona una solución entre un ASIC y un procesador de propósito general (Gu *et al.*, 2010). Por el contrario, DSP y computadoras personales (PC) tienen arquitecturas pipelined. El tiempo de cálculo de un controlador basado en FPGA puede ser relativamente corto, sin tomar en cuenta la complejidad del algoritmo de control debido a su arquitectura de procesamiento paralelo (Cho *et al.*, 2009). Además, el controlador es más barato, robusto y compacto; por otro lado, las fallas en el proceso de ensamblaje se reducen considerablemente.

Para mejorar la operación de las tareas necesarias para la implementación de un controlador de movimiento, tanto DSPs y FPGAs se deben programar en punto fijo, que es una desventaja con respecto de los DSPs de punto flotante. Sin embargo, los FPGAs ofrecen rendimientos más altos que los DSPs en la realización de operaciones con punto fijo, reduciendo los tiempos de ejecución. En los últimos años, la tendencia a usar hardware concurrente para aplicaciones de control es cada vez mayor y los FPGAs son cada vez más populares desde que ofrecen rendimientos más altos en cálculos repetitivos y masivos (Monmasson *et al.*, 2007).

Al comparar el rendimiento entre las unidades de punto fijo y de punto flotante de los DSPs se puede notar que, aunque el costo es similar, existen algunas diferencias tecnológicas. Éstos últimos ofrecen una mayor flexibilidad matemática, exactitud, menor tiempo de ejecución y el reducido tiempo de desarrollo y flexibilidad del software hacen del DSP de punto flotante una solución atractiva para aplicaciones donde se requiere gran capacidad de cómputo aritmético (Bueno *et al.*, 2009). La principal desventaja de un DSP de coma flotante es su reducida cantidad

de periféricos; esta deficiencia es compensada con la integración de un dispositivo lógico programable (PLD) para implementar estos módulos.

En esta sección se muestran las razones y los problemas que dieron paso a la investigación que aquí se presenta, además de proporcionar los objetivos que se siguieron para la conclusión de la misma. En el capítulo 2, se muestran los antecedentes con respecto a la investigación descrita, además de realizar comparaciones entre el sistema desarrollado contra los existentes, así como una breve descripción de los sistemas estudiados y los resultados que se presentan en la literatura citada. En el capítulo 3, se muestra un marco teórico robusto y actualizado, asimismo el desarrollo de la tesis, los algoritmos y las estructuras implementadas sobre los dispositivos programables; finalmente, en el capítulo 4, se muestran los resultados obtenidos de la implementación del sistema desarrollado.

I.1 Justificación

El proyecto surgió de la necesidad de crear controladores más robustos y con mayor capacidad en los sistemas de control de movimiento de motores, utilizando la tecnología FPGA, tomando en cuenta que los procesos industriales, máquinas herramientas y robots constan de numerosos motores y el tener controles para cada uno resulta costoso. Se propuso desarrollar un sistema de control multieje que incorpore un mayor número de controladores en un solo equipo siendo más eficiente, flexible y de un costo efectivo.

En este proyecto de investigación y desarrollo tecnológico, se diseñó y construyó un controlador de movimiento para controlar 3 ejes, el desarrollo se llevó a cabo sobre una plataforma basada en FPGA con un procesador embebido, que se conoce como un soft-core. Las principales contribuciones son: el diseño del algoritmo de control P-PI feedforward, la incorporación y ejecución de múltiples lazos de control y el diseño de una arquitectura SoPC (System on Programmable Chip). También se mejoró el sistema de control y de seguridad de la arquitectura presentada en la estructura robótica controlada con una tarjeta comercial GALIL (Galil Motion Corporation, 2011), una de las firmas de controles más populares para posicionamiento de servomotores, detectando fallas el sistema de retroalimentación (Encoder).

Referente al aspecto financiero, es necesario mencionar que el controlador comercial GALIL, del modelo para manipular 6 ejes DMC-1866 (PCI Accelera), tiene un costo de \$2,795 dólares. Este precio es elevado para el pequeño empresario que comienza a solidificar la PyME. Otros tipos de dispositivos con características similares tienen un costo similar, éstos incluyen marcas como: NI, YASKAWA, ADLINK y ABB, cabe mencionar que la mayoría de estos equipos utilizan técnicas simples de control PID.

I.2 Planteamiento del problema

Los avances tecnológicos que acompañan a la industria en nuestros días, han permitido que un gran número de procesos se encuentren totalmente automatizados y la tendencia general es continuar con esa práctica. Muchos de estos cambios se han dado gracias a la amplia gama de aplicaciones de la tecnología electrónica en todas las áreas del conocimiento y de la actividad económica. Un problema en la actualidad es que los fabricantes de dispositivos de control de alto rendimiento presentan un bajo nivel de personalización y costos muy elevados (Morales, 2010). Por otro lado, es muy importante desarrollar alternativas para las pequeñas y medianas empresas (PyMES) del país y del estado; de igual manera, es de vital importancia promover el desarrollo tecnológico nacional e incentivar la inversión en la investigación científica y tecnológica del país.

Por otra parte, muchos de los servomecanismos y controladores utilizados en las empresas locales y nacionales son productos comerciales de origen extranjero; generalmente, los sistemas a controlar y los controladores son provistos por los mismos fabricantes, siendo estos productos muy costosos sin mencionar los gastos de importación y los servicios de soporte técnico, de reparación y la necesidad de usuarios especializados que cuenten con la capacitación para el manejo del equipo. Además, con los avances tecnológicos que experimentan los países desarrollados estos controladores quedan obsoletos en poco tiempo.

Es un hecho que la industria mexicana está compitiendo en mercados internacionales contra países que ofrecen diferentes opciones en incentivos gubernamentales, seguridad, estabilidad financiera, infraestructura avanzada de manufactura y personal calificado; por esta razón es necesario tomar medidas y fortalecer a México aportando soluciones reales, eficientes,

confiables, seguras, funcionales y, sobre todo, accesibles con respecto a los productos comerciales a las PyMES, dado que ellas forman el 94% del total de las empresas del ramo en la entidad. Así mismo, este sector es muy importante para la economía del estado, debido a que en su totalidad aportan el 50% del producto interno bruto (PIB) de la entidad (SEDECO, 2004). Sin embargo, en México el 80% de las PyMES fracasa antes de los 5 años y el 90% no llega a los 10 años. Son varias las causas que generan dicho fenómeno, entre las cuales se encuentra la incapacidad de los pequeños empresarios de sobrevivir a los constantes cambios y requerimientos tecnológicos que repercuten en altas inversiones económicas, en la mayoría de veces, incosteables (Osornio, 2007).

El sector de la robótica, al contrario que la máquina herramienta y otros segmentos de equipamiento automático industrial, no ha adoptado ningún estándar significativo en el área del control, por lo que cada fabricante ofrece su propia tecnología propietaria; lo que dificulta la integración de los robots en los entornos de producción. Un robot industrial está diseñado para soportar una larga vida de trabajo; sin embargo, los controladores (que incluyen la parte eléctrica y electrónica) quedan obsoletos más rápidamente que los robots que controlan (o bien la sección mecánica). Cuando un fabricante lanza al mercado una nueva generación de controladores, las mejoras e innovaciones del último modelo no suelen estar disponibles para las versiones previas. Incluso, es frecuente que las mejoras introducidas en el lenguaje de programación no puedan ser utilizadas en sistemas ya en funcionamiento; esto debido a la arquitectura cerrada de los controladores que no permiten una reconfiguración o modularidad de sus componentes. Por otro lado, la disponibilidad de recambios en tarjetas y componentes electrónicos por parte del fabricante suele limitarse a un periodo de tiempo mucho menor que la vida útil del equipo, debido a la rápida obsolescencia tecnológica de los circuitos integrados. Estos factores dificultan enormemente incorporar los avances tecnológicos en equipos ya instalados y que han supuesto, habitualmente, una fuerte inversión económica (González *et al.*, 2004).

Entonces, al proponer un controlador de movimiento que cubra las demandas de los sectores ya mencionados es posible incrementar la calidad de los procesos manipulados por estos dispositivos y, además, ofrecer a la industria un controlador confiable, de costo efectivo y flexible, mediante la utilización de circuitos FPGA que ofrecen una plataforma de desarrollo modular altamente flexible y escalable; algunas ventajas principales de esta tecnología son la

relación costo beneficio, el alto rendimiento y la habilidad del cómputo en paralelo, así como la capacidad de modificación o actualización de su arquitectura, incluso de forma dinámica; ésta es una cualidad muy importante si se quiere evitar la rápida obsolescencia tecnológica.

I.3 Hipótesis y objetivos

I.3.1 Hipótesis general

Es posible diseñar una arquitectura SoPC de un controlador de movimiento multieje que presente una mejor respuesta en el seguimiento de perfiles en comparación del controlador PID convencional.

I.3.2 Objetivo general

Diseñar una arquitectura para el control de movimiento con base en la tecnología FPGA, que permita ejecutar tareas de control de movimiento con un buen desempeño y a un costo efectivo; además de que pueda ser aplicada en sistemas que requieren el control de diversos motores, como máquinas, herramientas y robots.

I.3.3 Objetivos específicos

1. Diseñar un prototipo electrónico basado en dispositivos de alto rendimiento que permita realizar un sistema de control en lazo cerrado.
2. Diseñar un controlador en cascada robusto y flexible.
3. Diseñar un controlador capaz de operar para aplicaciones de control de posición como de velocidad.
4. Describir componentes genéricos utilizados en el control de movimiento (QEI, PI, estimador de velocidad y generador de perfiles de movimiento).

CAPÍTULO 2:

II REVISIÓN DE LITERATURA

La información y experiencias adquiridas en estas investigaciones fueron pieza fundamental para un mejor desarrollo del sistema; además, se pudo aportar y enfocar un camino a seguir sobre este desarrollo.

II.1 Antecedentes

II.1.1 Antecedentes UAQ

En lo que se refiere a los trabajos realizados por estudiantes y docentes en la FI de la UAQ, se puede mencionar que corresponden al desarrollo e investigación sobre los sistemas de control de movimiento. Entre los más destacados se encuentran: el presentado por Ronquillo (2002), quien desarrolló un control digital para servomotores sin escobillas, realizando un algoritmo de control usando un DSP e introdujo el concepto de generación de trayectorias en hardware mediante perfiles trapezoidales en velocidad; también el trabajo de Osornio (2004), este autor diseñó un controlador de 3 ejes para máquinas herramienta basado en FPGA; el controlador tiene su base teórica en un módulo PID e incluye los algoritmos de generación de perfiles básicos y es aplicado a maquinados convencionales; asimismo, Hernández (2006) implementó un controlador PID aplicado a una fresadora; con el mismo objetivo, Romero (2004) desarrolló un instrumento para la detección de desgaste y ruptura de herramientas en procesos de fresado usando el procesamiento digital de señales en hardware con FPGA; también es importante el trabajo de Franco (2007), quien desarrolló un instrumento para la detección de desgaste en herramientas para procesos de taladrado; Osornio (2007) desarrolló un controlador para maquinaria de alta velocidad basado en FPGA. Éstos han sido algunos de los trabajos desarrollados en el área de control de movimiento y servosistemas, pero en ninguno de los trabajos anteriormente mencionado se hace un desarrollo de un controlador de movimiento multieje que utilice un

algoritmo diferente del ya conocido control PID simple el cual no es suficiente para muchas aplicaciones de control en donde la regulación de la velocidad y el seguimiento de trayectorias es crítico (Nakamura *et al.*, 2004), y tampoco el diseño de una arquitectura FPGA con un procesador embebido para el control de movimiento. Así, el uso de estas tecnologías abre una ventana interesante de oportunidades para explorar las ventajas de trabajo en conjunto para obtener una mayor flexibilidad en el sistema de control y disminuir el tiempo de desarrollo, obteniendo un sistema de control más eficiente al realizar un diseño software-hardware, en el cual puedan distribuirse de manera efectiva las tareas involucradas en el control de movimiento.

II.1.2 Antecedentes generales

A continuación, se citan algunos de los trabajos realizados en el área por investigadores y científicos que han publicado sus trabajos, en donde se muestra el uso combinado de las diferentes tecnologías (μ P y DSP) con y FPGAs en los sistemas de control de movimiento.

Shao y Sun (2007) realizaron la implementación de una nueva arquitectura de control para robots, en la cual se pueden aplicar diferentes algoritmos complejos de control con un alto desempeño. El principal objetivo del artículo es eliminar el lazo de control del DSP e implementarlo en un FPGA; debido a la alta velocidad de operación, éste puede mejorar la capacidad de procesamiento y cómputo en general y aliviar la carga operacional del DSP para otras tareas. El algoritmo de control propuesto en su nueva arquitectura se divide en dos partes: la parte lineal de control y la no lineal. En la parte lineal es donde se encuentran los lazos de posición y velocidad, los cuales son implementados en el FPGA; la parte de controlador no lineal que sirve para la compensación dinámica de la parte lineal se usa para realizar el cálculo del modelo matemático del sistema y el cálculo de ganancias relacionadas al controlador; esto se realiza en el DSP. El sistema desarrollado presentó un gran desempeño especialmente en movimientos de alta velocidad.

Cho *et al.*, (2009) realizaron el diseño e implementación de un chip basado en FPGA de control de movimiento de múltiples ejes (Figura II.1); este chip fue diseñado para controlar un sistema como un robot, un brazo manipulador o una máquina de control numérico. El chip

desarrollado incluye la generación de perfiles de velocidad, cálculos de interpolación, cinemática inversa, controlador PID, el módulo de cuadratura de encoder, interfaces de conversión de datos, generación de reloj. Los componentes fueron descritos en VHDL. El sistema propuesto permite una alta tasa de muestreo, un diseño flexible de bajo consumo y de bajo costo, comparado con soluciones DSP.

A continuación se muestra el chip de control multieje y sus componentes. El controlador propuesto ofrece un arquitectura abierta para el tipo de motor a controlar utilizando servoamplificadores analógicos, lo cual proporciona modularidad al diseño.

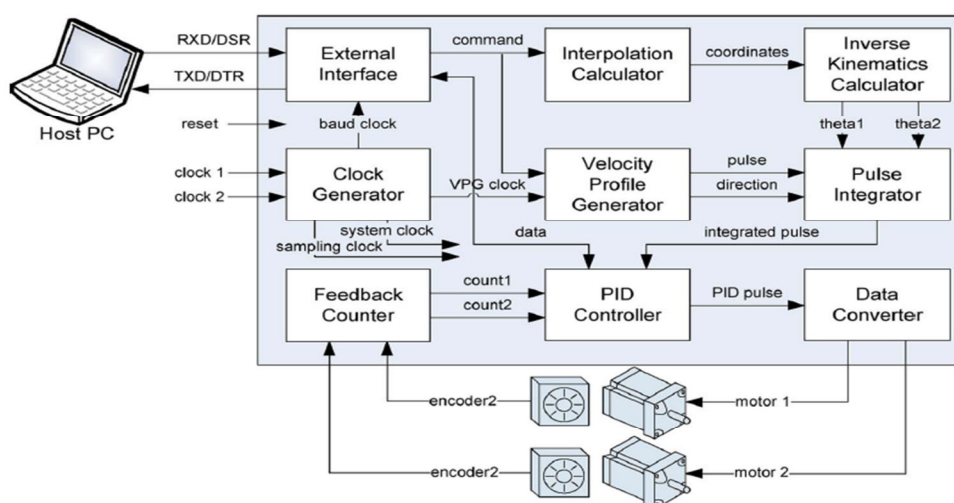


Figura II.1 Controlador en un chip propuesto para el control de movimiento (Cho et al., 2009).

Armando Astarloa *et al.* (2010) Desarrollaron un sistema de control multieje con tecnología FPGA, utilizando un controlador PID simple; la arquitectura propuesta consistía de 4 ejes de movimiento y un microprocesador (Xilinx Corp. MicroBlaze, 2011), el cual era el encargado de generar la trayectoria de movimiento, así como de la comunicación con otros módulos como protocolos de comunicación serial, Ethernet, memoria flash, memoria de acceso aleatorio (RAM), GPIO etc. Así, se obtiene un SoPC de bajo costo ofrece un controlador modular con un gran reusabilidad de sus componentes. Adicionalmente, se presenta el experimento de un SoPC auto reconfigurable utilizando reconfiguración en tiempo real, en este

caso bloque de control PID puede ser remplazado dinámicamente por otro módulo con diferentes características.

Guoying Gu *et al.* (2010) presentaron el diseño de un sistema de control distribuido multieje de control de movimiento utilizando el bus IEEE-1394. En el sistema diseñado fue utilizado el lenguaje unificado de modelado (UML) para mostrar la interacción entre los objetos requeridos en el sistema. El sistema desarrollado consiste en una serie de nodos inteligentes interconectados por el bus IEEE-1394, cada nodo contiene cuatro partes principales: el módulo de comunicación, un DSP, un FPGA y un convertidor digital analógico (DAC). El DSP realizaba las tareas del lazo de control y la comunicación en tiempo real, mientras que el FPGA fue el encargado de adquirir las señales de los transductores y de controlar los actuadores; cada nodo comparte la información con los otros módulos para compartir información acerca del estado de los sensores y actuadores.

Naveen K. y Richard C. (2010) presentaron un algoritmo para medir velocidad con gran precisión utilizando un encoder óptico de bajo costo, en el cual se utilizó un FPGA para adquirir y decodificar las señales del encoder, éste actúa como sistema esclavo de un DSP que realiza el procesamiento del algoritmo de medición de velocidad; el algoritmo fue probado en un servomotor en lazo cerrado y abierto reduciendo el error en la medición de la velocidad hasta en un 86%.

En la Figura II.2 se muestra la configuración típica maestro-esclavo de un controlador basado en FPGA/DSP, para el control de servosistemas.

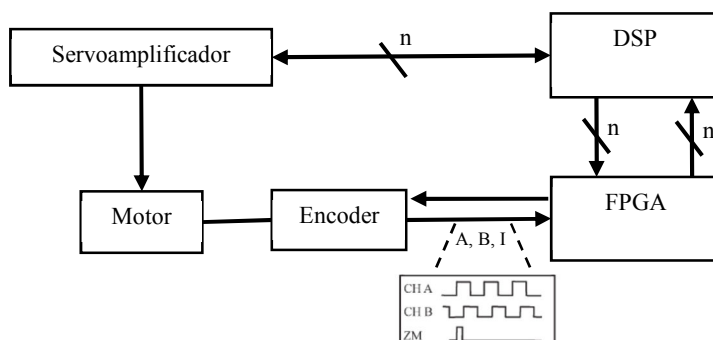


Figura II.2 Diagrama a bloques de un típico servo controlador con base en FPGA/DSP (K. Boggarpu et al., 2010).

CAPÍTULO 3:

III METODOLOGÍA

III.1 Marco Teórico

III.1.1 Control de Movimiento

Comúnmente se cree que los ingenieros especializados en el área de electrónica de potencia son quienes han desarrollado este campo, sin embargo, el control de movimiento es el resultado de aplicar la teoría de control a la electrónica de potencia. En la actualidad se ha convertido en un campo bastante maduro que tiene más de cuatro décadas. La investigación en 1970 de la aplicación de los microprocesadores en el control de equipo eléctrico estableció la base para el área de control de movimiento (Lorenz *et al.*, 1994). La evolución histórica del desarrollo del campo en el control de movimiento se muestra en la Figura III.1.

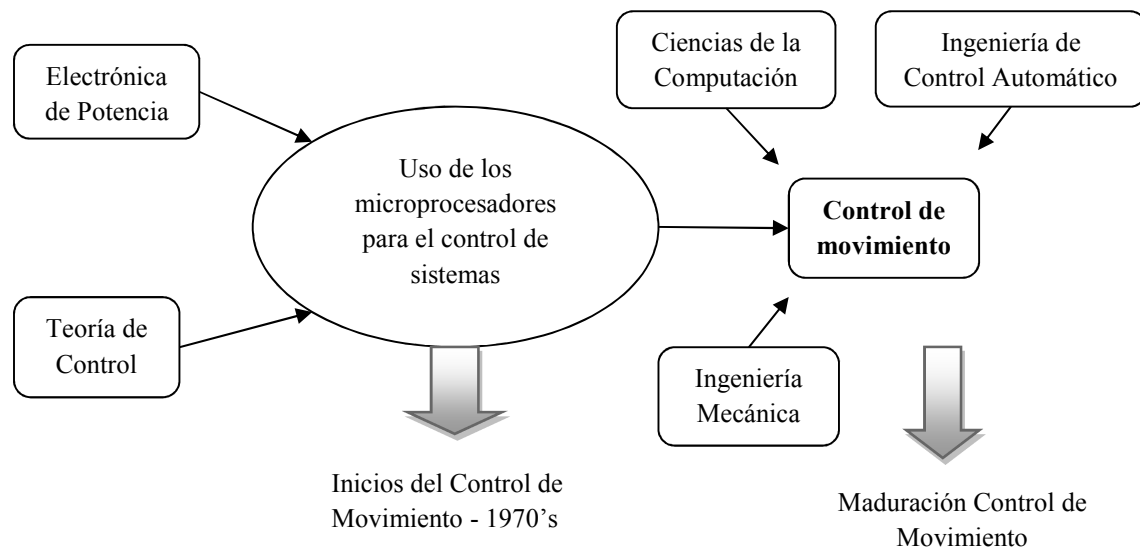


Figura III.1 Desarrollo del área de control de movimiento (Alahakoon, 2000).

Aunque el desarrollo del área fue iniciado por ingenieros en electrónica de potencia, el estado actual en la teoría de control de movimiento es tal que se requieren de cursos complementarios para entender los desarrollos en el área. Actualmente, es un campo extremadamente competitivo. Hoy en día, la electrónica de potencia y el control de movimiento son tratados como dos distintas áreas que crecen conforme a los avances de la tecnología. La electrónica de potencia principalmente estudia la transformación de la energía eléctrica, los dispositivos de alta frecuencia de conmutación, la eficiencia energética y las tecnologías amigables con el ambiente; éstas son las palabras claves en las investigaciones del área mencionada. Por su parte, el control de movimiento involucra diferentes áreas del conocimiento que ya no solo se relacionan con la parte eléctrica sino también con la mecánica.

III.1.2 El control Automático

El desarrollo del control automático está muy relacionado con la revolución industrial; algunos ejemplos típicos son: la invención de nuevas fuentes de energía (eléctrica e hidrocarburos) y de nuevas formas de producción industrial. Mientras que las nuevas formas de energía debían ser controladas propiamente, los procesos de producción debían de mantenerse operando. El control de procesos industriales y la automatización ocurrieron en paralelo a la invención de los controladores.

Las misiones espaciales, iniciadas en Rusia con el satélite artificial Sputnik en 1957, abrieron paso al control automático. Ésta fue la primera ocasión en que las computadoras fueron utilizadas para aplicaciones de control (Slemon, 1994). Por lo tanto, variables físicas como corriente, voltaje, presión, temperatura etc., que antes eran tratadas de forma analógica, ahora fueron cuantificadas y tratadas numéricamente dentro de una computadora. Esto trajo consigo un enorme flujo de ideas de matemáticos en el campo de la teoría del control automático. Estos avances crearon nuevos conceptos como la optimización, el control estocástico, la identificación, el control adaptativo, el control predictivo y varios métodos de control.

Hoy en día el área del control automático es un área de suma importancia en el ámbito industrial y social, sus áreas de aplicación se ramifican en un sin fin de aplicaciones. A continuación se mencionan algunas:

- Generación y distribución de la energía eléctrica
- Control de procesos industriales
- Manufactura y la robótica
- Medios de transporte y comunicación

Sin embargo independientemente del área de aplicación, el ingeniero de control con base en un número de disciplinas bien establecidas dentro del área de control automático es capaz de abordar cualquier problema, cómo a continuación se describe:

- 1. Teoría de sistemas:** Todos los sistemas físicos o reales que tienen que ver con el área del control automático pueden ser clasificados con la siguiente descripción:
 - Sistemas lineales
 - Sistemas no lineales
 - Sistemas estocásticos
 - Sistemas discretos
 - Sistemas con retardos en el tiempo
 - Sistemas de parámetros distribuidos
 - Sistemas descentralizados
- 2. Modelado e identificación:** Un modelo matemático de los sistemas es muy importante en las aplicaciones de control automático, existen diferentes formas ya sea a través de métodos matemáticos y la descripción de la ecuaciones diferenciales que gobiernan la dinámica de un proceso o a través de métodos experimentales, hoy en día la teoría de identificación de sistemas es una área bien establecida con gran cantidad de herramientas para la identificación de procesos.
- 3. Diseño del control:** Dado un proceso en específico se tiene que encontrar la configuración del controlador que mejor se adecue al mismo y pueda regularlo de manera satisfactoria dadas las especificaciones de control y regulación.

4. **Aprendizaje y adaptación:** Normalmente los procesos tienen la tendencia a cambiar sus propiedades con el tiempo, existen diferentes factores que intervienen en el desgaste y envejecimiento de los sistemas, lo cual requiere que los controladores necesiten ser reajustados, el aprendizaje y la adaptación son la respuesta a ese problema.
5. **Cómputo y Simulación:** El control automático está vinculado directamente con la computación y su evolución, la creación de los sistemas digitales abrió paso a un gran número de herramientas matemáticas especializadas en el análisis y diseño de sistemas de control, por ejemplo MATLAB/Simulink.
6. **Implementación:** Este sin duda es un importante aspecto en el área de control, los problemas que existen entre la teoría y la aplicación real; en la práctica sin duda existen diferentes aspectos que no siempre es posible abordar en la teoría, estos aspectos incluyen y necesitan del conocimiento de otras áreas como las ciencias de la computación, la electrónica digital y de potencia e interfaces hombre-máquina etc.
7. **Comisionamiento y operación:** Justo después del que el sistema de control ha sido diseñado e implementado tiene que ser comisionado y puesto en marcha, en esta área se presentan varios problemas que han desarrollado toda una nueva línea de investigación conocida como detección de fallas y diagnóstico.

III.1.3 El estado actual del control de movimiento

Como se mencionó anteriormente, el control de movimiento inició con dos áreas bien establecidas: el control automático y la electrónica de potencia; esto hace que el área de control de movimiento sea muy competitiva cubriendo cada sistema de movimiento, desde los micro hasta los macro. Además, hay que lidiar con diferentes tipos de actuadores principalmente eléctricos, algunos como electroimanes, servomotores de corriente alterna (AC) y directa (DC), motores lineales etc.

Dado que el movimiento es una magnitud mecánica, todos los sistemas de control de movimiento contienen partes mecánicas en el proceso donde se aplicarán; por ejemplo, aunque en un robot manipulador el motor eléctrico es el actuador, el sistema completo consiste en muchas partes mecánicas que están en movimiento y la dinámica de todas sus partes debe ser

considerada. Actualmente, en este tipo de aplicaciones se considera la dinámica del sistema completo como un solo modelo en el diseño del control, lo cual reduce y simplifica la brecha que existe entre la parte mecánica y la eléctrica. Además, los desarrollos en los métodos de identificación de sistemas facilitan encontrar la función de entrada-salida del modelo completo, teniendo así un modelo de un sistema electromecánico. Sin embargo, existen dos principales dificultades en el control de movimiento. La primera es la robustez de los algoritmos de control contra las variaciones del sistema completo, a diferencia de los actuadores eléctricos que presentan menos cambios, los sistemas mecánicos presentan parámetros como la fricción, amortiguamiento, flexibilidad etc. Éstos pueden cambiar con el tiempo o la temperatura; las variables ya mencionadas o no linealidades afectan el desempeño del controlador; para resolver estos problemas algunas de las soluciones son la estimación en línea y la adaptación. La segunda dificultad es cómo hacer el control más inteligente para que así aprenda de su propia experiencia o de la experiencia de un experto; sistemas de auto sintonizado, control adaptativo, control de lógica difusa (fuzzy) y sistemas de redes neuronales han mostrado un progreso positivo en esta área (Alahakoon, 2000).

Con esta breve introducción en la teoría y el estado actual del control de movimiento, nos enfocaremos en un tipo particular de las aplicaciones del control del movimiento que son los servosistemas eléctricos utilizados en aplicaciones de Mecatrónica.

III.1.4 El Servosistema

El servosistema (o servomecanismo) es un sistema de control retroalimentado en el que la salida es algún elemento mecánico, sea posición, velocidad o aceleración. Por tanto, los términos servosistema o sistema de control de posición, o de velocidad o de aceleración, son sinónimos.

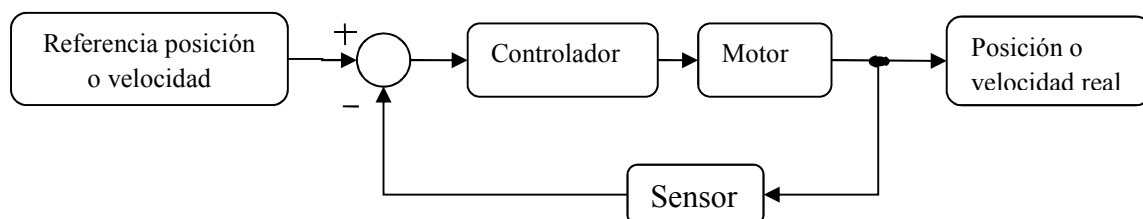


Figura III.2. Lazo de control de un servosistema para un motor eléctrico.

Nótese que a veces se denomina también servosistema a un sistema de control cuya salida debe seguir con exactitud una trayectoria determinada en el espacio (como la posición de una aeronave en el espacio en un aterrizaje automático). Los ejemplos incluyen el sistema de control de una mano de robot, en que la misma debe seguir una trayectoria determinada en el espacio, al igual que una aeronave en el sistema de control de aterrizaje. Los servosistemas adoptados en el control numérico o en robots industriales son generalmente diferentes del servosistema introducido en textos de control automático (Nakamura *et al.*, 2004). En la Figura III.3 se muestra la configuración típica de un control de movimiento utilizado en la Mecatrónica; comúnmente el control de posición es diseñado como un control de relación, de igual forma el lazo de velocidad y de corriente: además, en la estructura del servoamplificador, el amplificador por modulación de ancho de pulso (PWM) es siempre adoptado y la frecuencia de conmutación utilizada va de cientos a algunas docenas de KHz.

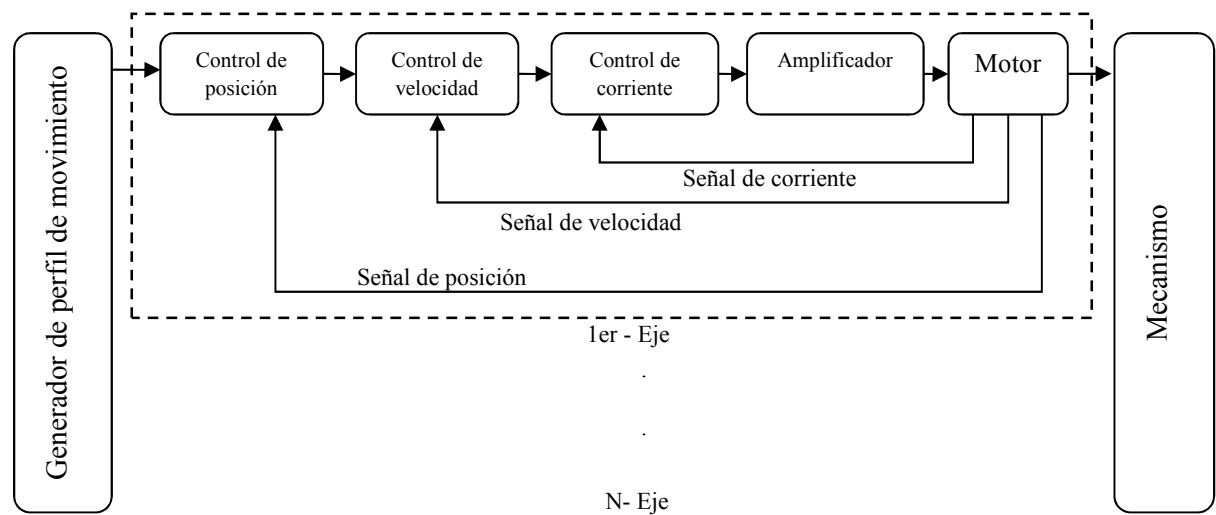


Figura III.3 Estructura de un servosistema industrial (Nakamura et al., 2004).

Con el fin de entender el uso de los servosistemas en la Mecatrónica y sus diferencias de los servosistemas en general, a continuación se enumeran algunos puntos:

1. En los servosistemas mecatrónicos, existen dos tipos de control. Uno es el control de posición punto a punto (PTP: Point To Point) que se enfoca principalmente en el tiempo de llegada y el punto final sin considerar las respuesta de la ruta seguida; por otra parte,

existe el control de contorno (CP: Contouring Path) que se centra en la trayectoria de movimiento de la posición actual a la siguiente, de igual forma en la velocidad del movimiento. Ejemplo de control PTP son los robots utilizados para aplicaciones de traslado o acomodo (pick and place) y aplicaciones de control CP se encuentran en máquinas de control numérico por computadora (CNC), robots industriales para pintar, cortar o ensamblar.

2. En control CP, el servosistema como un control de posición requiere un estricto control de la velocidad para diferentes tipos de respuesta. Un sobrepaso en el control de posición no debe ocurrir, en muchos casos, el control de velocidad es también regulado para así evitar el sobrepaso. En muchos procesos eso puede causar un defecto fatal en la producción del producto final.

III.1.5 Servomotores

Los servomotores son actuadores de alto rendimiento que por lo común se usan como generadores de movimiento mecánico en computadoras, máquinas-herramientas CNC, robots industriales, autos eléctricos y otras aplicaciones en donde el arranque y el frenado se deben hacer con rapidez y exactitud. Los servomotores son los actuadores primarios en el control de movimiento de servosistemas en Mecatrónica; además, son de peso ligero y tienen armaduras de baja inercia que responden con rapidez a los cambios en el voltaje de excitación. Asimismo, la inductancia muy baja de la armadura en estos motores da lugar a una baja constante eléctrica de tiempo (0.05 y 1.5 ms) que agudiza todavía más la respuesta del motor a las señales de voltaje (Fink y Wayne, 2001).

Las características fundamentales que se obtienen de cualquier servomotor, ya sea de DC o AC, son:

- 1) El torque de salida del motor es más o menos proporcional a la tensión de control aplicada (provista por el amplificador como respuesta a la señal de error).
- 2) El sentido del torque es determinado por la polaridad (instantánea) de tensión de control.

Tabla III.1 Tipos comunes de motores y sus características

Tipo de motor	Fuente de alimentación	Eficiencia típica ⁽¹⁾	Acoplamiento	Controlador
DC con escobillas	DC	<50%	Directo o con reductor	Simple o complejo
DC sin escobillas (BLCD)	Frecuencia variable, 3 fases AC	>90%	Directo o con reductor	Complejo
AC inducción	3 fases AC	<90%	Reductor	Simple
AC síncrono	Frecuencia variable, 3 fases AC	>90%	Directo o con reductor	Simple o complejo
Motor a pasos	DC, Conmutación digital	<5%	Directo o con reductor	Simple

(1) Las eficiencias son para motores por debajo de 3,7 KWatts. El motor aumenta la eficiencia con el tamaño para todos los tipos y es más del 90% para casi todos los motores en las decenas de kilowatts (Mentz, 2000).

La primera consideración en la selección de un motor es el tipo de alimentación disponible para robots grandes y estacionarios utilizados en la automatización y manufactura; se puede asumir que la alimentación es de AC y trifásica, pero para robots vehículos y robots pequeños a menudo se opera con una alimentación de DC. En este trabajo se orientará principalmente a los motores de DC con escobillas y sin escobillas, que a continuación son brevemente descritos.

III.1.6 Motores de corriente directa o DC

Los motores de DC, en general, operan de acuerdo a la ley de Ampere de fuerza magnética (1), la cual explica y cuantifica la fuerza que actúa sobre un cuerpo que conduce la corriente (el bobinado del motor) en presencia de un campo magnético. De acuerdo con esta ley, la fuerza **F** que actúa sobre un cuerpo es el producto vectorial de la corriente **I**, que fluye a través

del conductor y la densidad de flujo magnético \mathbf{B} , dentro del cuerpo en que se encuentra, como se muestra en la Figura III.4.

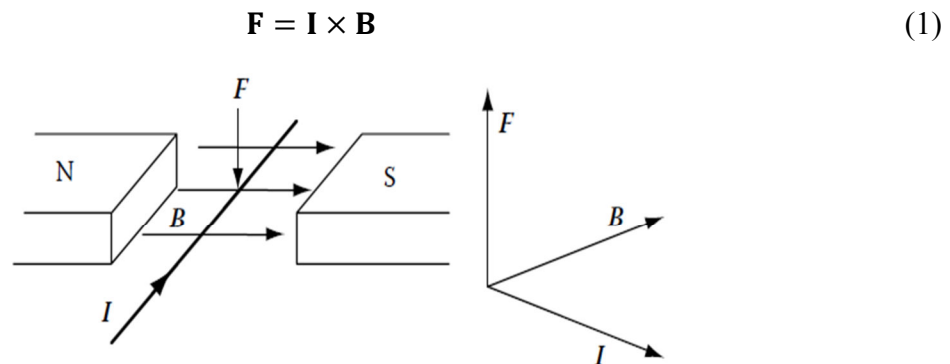


Figura III.4 Ley de Ampere de la fuerza magnética.

La corriente que fluye a través de un conductor induce una fuerza \mathbf{F} que es normal a la corriente y la densidad de flujo magnético \mathbf{B} . La corriente es la entrada eléctrica al motor de DC, la fuerza inducida se acopla al eje de salida del motor para producir el par de salida. La disposición inteligente del flujo de corriente, el flujo magnético, y la fuerza inducida resultan en una entrega suave de torque al motor (Wilamowski *et al.*, 2011).

III.1.7 Motor de DC con escobillas

Un modelo simple del motor de DC se puede ver en la Figura III.5. Se compone de un par de imanes permanentes como el estator (que sirve como la cubierta fija del motor) y una bobina como el rotor (o la flecha del motor) conectados a un conmutador o colector. Éstas forman el corazón de los motores de DC con escobillas, básicamente consisten en dos (o más) barras del conmutador (los medios cilindros en la Figura III.5) y dos escobillas, ambos de materiales conductores. Las escobillas son presionadas por resortes para mantener el contacto con el colector. Cada uno de los dos extremos de la bobina del rotor están conectados eléctricamente a una barra independiente del conmutador. Cuando el rotor gira en espiral y el cilindro del conmutador (formado por todas las barras del conmutador, aisladas unas de otras) giran juntos se produce un flujo de la corriente cambia de acuerdo al contacto realizado por las

escobillas en la barras del conmutador, lo que crea un flujo de corriente alterna en la bobina, en función de la posición de la misma. En el ejemplo mostrado en la Figura III.5, habrá un momento en que las escobillas están en contacto con ambas barras del conmutador, es decir, cuando la bobina del motor es perpendicular a las líneas de campo magnético; esto crea un corto circuito, pero el momento angular del rotor hará que la bobina se mueva de nuevo; de esta manera el rotor seguirá moviéndose, minimizando el tiempo de corto circuito.

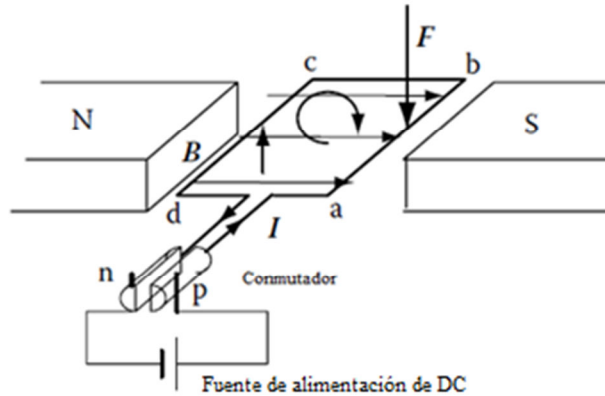


Figura III.5. Motor de corriente directa con escobillas.

Por otra parte, más de un conjunto de bobinas pueden ser utilizadas y el número de barras en conmutador aumenta, esto ayuda a reducir el tiempo del corto circuito y es una solución que hoy en día la mayoría de los motores utilizan. Con el conmutador en su lugar, el voltaje efectivo a través de la bobina del motor es equivalente a la corriente que fluye por la bobina con una resistencia fija; la corriente alternará como se muestra en la Figura III.6 para el segmento “ab” de la bobina del motor, las pequeñas secciones denotadas por “b” representan el tiempo cuando se produjo el corto circuito.

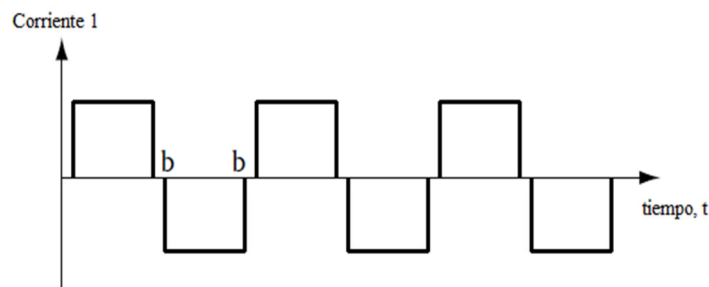


Figura III.6 Motor de corriente directa con escobillas.

III.1.8 Motores de DC Sin Escobillas o BLDC (Brushless DC)

Este tipo de motores no hacen uso de los colectores para regular la potencia o la corriente que fluye en la bobinas. En cambio, los motores sin escobillas regulan el flujo de corriente a través de semiconductores con retroalimentación de la posición del eje del motor. Una característica inherente de los motores sin escobillas es el requisito de sensores para detectar la posición angular absoluta del eje del motor, ya sea de manera directa (con sensores) o indirecta (sin sensores). La construcción de las escobillas también difiere de la de los motores de DC con escobillas. En los motores con escobillas, los imanes permanentes se utilizan como estatores, mientras que las bobinas del motor se unen al rotor. En los motores sin escobillas, el estator normalmente se compone del bobinado del motor, mientras que un imán permanente se hace cargo del rotor. Los motores sin escobillas tienen las siguientes ventajas sobre los motores con escobillas:

1. Tienen una mayor velocidad máxima y una mayor capacidad debido a la construcción del motor. La fricción del eje del rotor se reduce porque no hay necesidad de escobillas que interactúen con el colector. Por otra parte, la disipación del calor de las bobinas del estator es más eficaz a través de la carcasa del motor.
2. Trabajan en entornos menos favorables. El tamaño de éstos en comparación a los motores con escobillas es menor, haciéndolos adecuados para aplicaciones compactas, como en los brazos robóticos; los motores sin escobillas proporcionan un par mayor. La ausencia de un conmutador evita también las posibilidades de chispas o arcos eléctricos, por lo que es seguro para aplicaciones en lugares con materiales inflamables o gases como en las industrias petroquímicas.
3. Dado que no se utiliza colector, estos no requieren de mantenimiento. También producen menos ruido en comparación con los motores con escobillas.

Los motores sin escobillas son similares en su construcción a los motores de corriente alterna, pero las características de los dos son muy diferentes. Los motores con escobillas tienen un par-velocidad lineal cuando la alimentación es constante (Wilamowski *et al.*, 2011).

III.1.9 Características y desempeño de los motores de DC

Considerando que el flujo magnético de los imanes permanentes para los típicos motores de DC de imanes permanentes, sea con o sin escobillas, no varía mucho, la relación velocidad–torque y corriente–torque se mantienen y son lineales dentro de un rango de operación muy amplio.

La velocidad máxima se muestra en la Figura III.7, que sería la velocidad máxima sin carga en el motor a un voltaje de alimentación fijo, para el torque máximo la velocidad es igual a cero. La corriente en el motor es proporcional a la carga aplicada en rotor, eso simplifica de forma significativa muchas aplicaciones (Bogdan *et al.*, 2011).

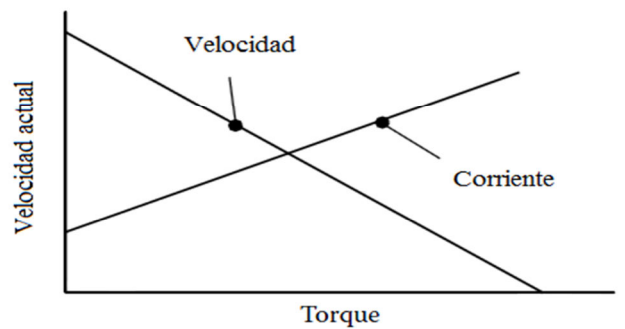


Figura III.7 Características típicas de un motor de DC.

III.1.10 Amplificadores para motores de DC con escobillas

En general, hay dos tipos populares de amplificadores o “drives eléctricos” que se utilizan para controlar la potencia de los motores de DC con escobillas: amplificadores lineales y por PWM.

III.1.11 Amplificadores Lineales

Como su nombre lo indica, éstos proporcionan un flujo de corriente continuo al motor que es linealmente proporcional a la velocidad o torque requerido por el motor. Estos controlan de

manera lineal la amplitud del voltaje o corriente que se suministra al motor. La forma más sencilla de implementar un amplificador lineal en un sistema de lazo cerrado es mediante el uso de un transistor de potencia, como se muestra en la Figura III.8.

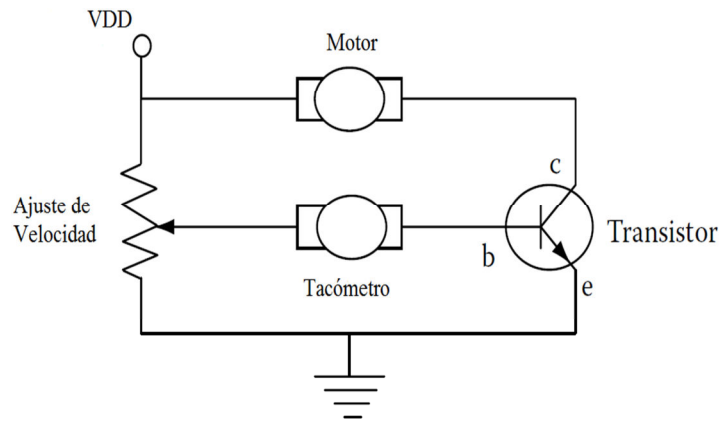


Figura III.8 Amplificador lineal para un motor de DC.

La corriente de la base del transistor es controlada y, por lo tanto, también la corriente que fluye a través del motor. La entrada a la base del transistor puede venir de un DAC o la salida de algún dispositivo programable, o como se muestra en la Figura III.8 la salida del tacómetro en un sistema de control de velocidad, los amplificadores lineales son ideales para sistemas de alto rendimiento de regulación de velocidad. Mediante la variación del potenciómetro, la velocidad del motor poder ser ajustada, ya que limita la corriente a la base del transistor, que se genera a partir de la retroalimentación del tacómetro. El ejemplo anterior se aplica sólo para la velocidad del motor de forma unidireccional. Para llevar a cabo el control de velocidad bidireccional, un transistor complementario adicional puede ser agregado al circuito. La potencia nominal de los transistores deber ser propiamente seleccionada dependiendo de la corriente del pico y nominal en el motor (Wilamowski *et al.*, 2011).

Los fabricantes de los motores también proporcionan amplificadores estándar para las diferentes marcas de motores. Las señales de entrada que se requieren, por lo general, son señales analógicas bipolares.

III.1.12 Amplificador por PWM

Dentro de los amplificadores es el método más popular para controlar la potencia en un motor. Un controlador PWM es alimentado por una sola fuente de voltaje de DC, con un amplificador que conmuta la alimentación de un estado alto a uno bajo con una determinada frecuencia y un ángulo de disparo variable (o ancho de pulso), de esta manera la potencia promedio (en términos de voltaje y corriente) es controlada. La frecuencia de salida del PWM es determinada por una red Resistencia-Capacitor (RC) como se expresa en la ecuación **¡Error! No se encuentra el origen de la referencia.** y el ancho del pulso es controlado por entrada analógica típicamente en el rango de 0-3V. Un circuito simple para PWM se muestra en la Figura III.9.

$$f_{osc} = 1.1 \div (R_T \times C_T) \quad (2)$$

La generación de la salida PWM se muestra en el diagrama de tiempo de la Figura III.9, con tres diferentes etapas de la señal de entrada, dos etapas son el mínimo y máximo ciclo de trabajo (Duty-Cycle) y la otra variando el ciclo de trabajo de la señal de control.

La frecuencia del PWM es normalmente establecida dentro del rango no-audible para mantener el sistema silencioso. Una frecuencia mayor también asegurará una distribución de la energía más equitativa y un movimiento más suave y uniforme. En el ejemplo mostrado en la Figura III.9 la frecuencia utilizada es cerca de un 1.1Khz.

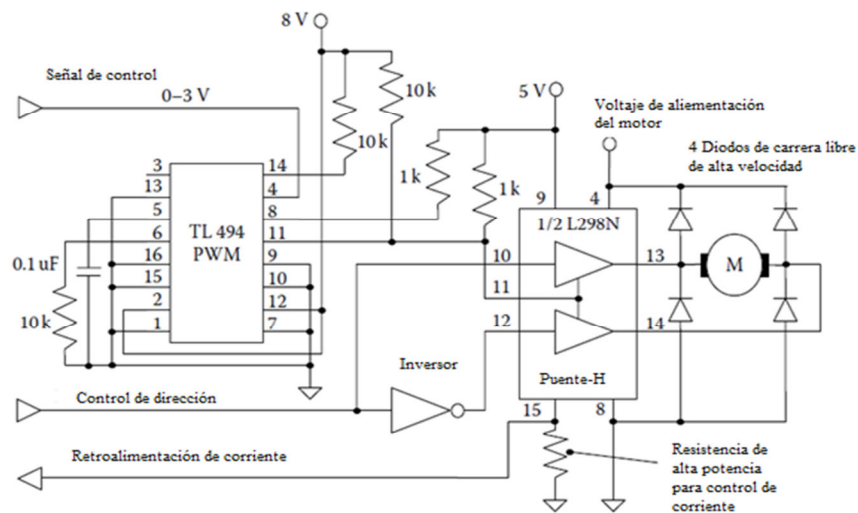


Figura III.9 Circuito para el manejo de un motor por PWM (Wilamowski et al., 2011).

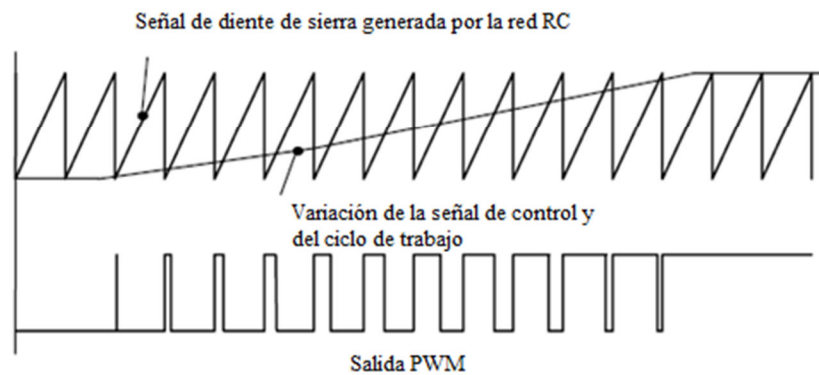


Figura III.10 Salida del PWM.

El funcionamiento general del PWM se puede ver claramente en la Figura III.10. Como la señal de control varía entre 0-3V, asimismo el ciclo de trabajo a la salida de 0-100% de forma lineal. La salida del PWM va a la entrada de un puente completo o puente H, el cual entrega la señal PWM al motor con el voltaje de tensión de motor. Con la implementación de este controlador PWM, el controlador sólo tiene que proporcionar la señal analógica de control y una señal TTL de dirección. También podemos monitorear la corriente consumida en el motor durante su operación. Actualmente, muchas aplicaciones de control por PWM se realizan por circuitos digitales como microcontroladores, ejemplo (Maxon Motor ADS 50/10,2011), donde se puede programar la frecuencia del PWM y variar el ciclo de trabajo.

III.1.13 Amplificador para motores sin escobillas

Éste consiste, normalmente, en un generador de onda sinusoidal, que es modulada por un arreglo de circuitos PWM, seguido de un puente trifásico de conmutadores de alta potencia típicamente MOSFET's (Metal-Oxide-Semiconductor Field-Effect Transistor) o IGBT's (Insulated Gate Bipolar Transistor), además de dispositivos de retroalimentación conectados al motor. El circuito de control tiene que leer la posición y (o) la velocidad del motor sin escobillas e interpretar las señales en orden para producir la salida a cada una de las fases del motor (normalmente 3 ó 4). Un diagrama general para el control de un motor sin escobillas de tres fases se muestra en la Figura III.11.

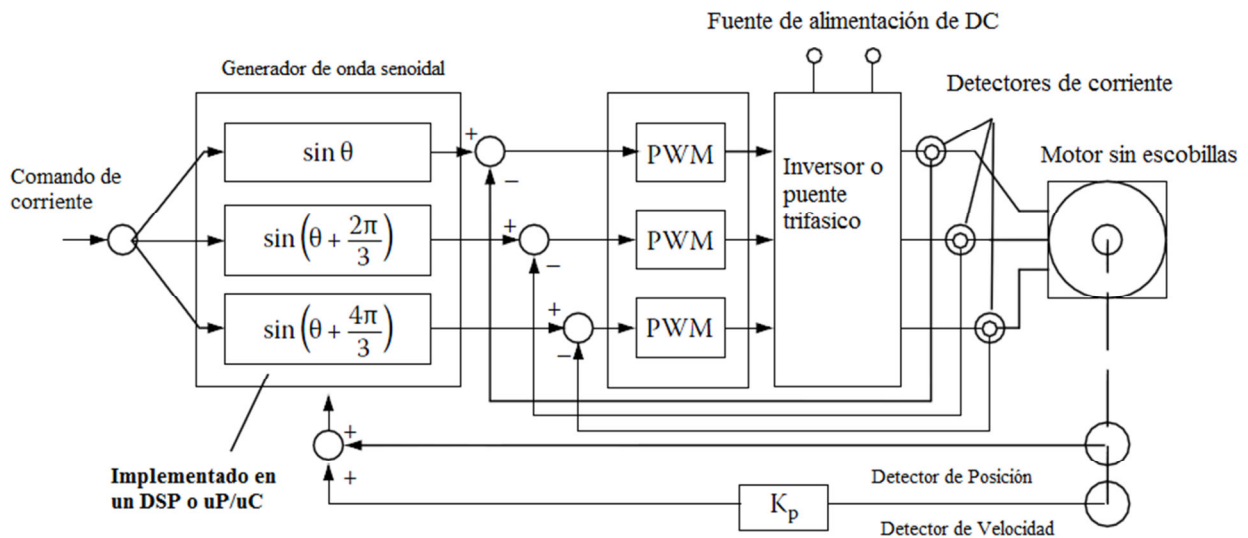


Figura III.11 Diagrama a bloques de un controlador para motores sin escobillas (Wilamowski et al, 2011).

A diferencia de un motor de AC o el motor de inducción, donde la entrada para el actuador es una fuente de AC, el motor sin escobillas utiliza una fuente de alimentación de DC para funcionar. Un generador de onda sinusoidal genera la secuencia o forma de onda para cada fase del motor después de que procesa la información de la retroalimentación sobre la posición del eje del rotor. En vista de la potencia de procesamiento que se requiere, se suelen utilizar dispositivos como un DSP, μP o un μC , a fin de modular la señal sinusoidal a través de una señal PWM y así entregar la potencia al motor. Actualmente, en mercado existen diferentes chips de uso específico que realizan todas estas tareas como MC73110 de PMD Corp. o el L6230 de ST Electronics, así como también se han desarrollado técnicas y productos para su control sin la necesidad de sensores (Mohammed et al., 2010).

III.1.14 Servoamplificadores

Los servoamplificadores, además de incluir las etapas de amplificación de corriente y voltaje (lineal o por PWM), contienen lazos de control embebidos ya sean de velocidad o corriente (control de torque); existen servoamplificadores para cada tipo de motor, sean de DC o AC, éstos incluyen numerosas características como protección contra sobre temperatura, corto circuito y sobre corrientes, normalmente tienen dos tipos de operación:

- **Modo voltaje o velocidad:** En este modo de operación el voltaje de entrada V_i o señal de comando es proporcional a la velocidad en el motor V_m , por lo tanto, el servoamplificador frecuentemente es modelado por una ganancia como se muestra en la ecuación (3).

$$\frac{V_m}{V_i} = K_V \quad (3)$$

- **Modo corriente o torque:** Cuando se configura en esta modalidad la salida de corriente I_m (corriente en el motor), se considera directamente proporcional a la señal de comando V_i , de tal manera que permanece relacionado mediante la ganancia K_I , como se muestra en la ecuación (4).

$$\frac{I_m}{V} = K_I \quad (4)$$

Normalmente, los servoamplificadores contienen un controlador tipo PI (Proporcional-Integral), se descarta el término derivativo debido al ruido en la corriente (Krishnan, 2001); el estándar industrial de la señal de comando (V_i) es $\pm 10V$ y comúnmente se necesitan señales de lógica transistor-transistor (TTL) para la señal de activación y de estado. Los servoamplificadores también pueden proporcionar información adicional, es decir, retroalimentación de la corriente, la señal de tacómetro si el sistema dispone de uno. Conocidos fabricantes que proporcionan servoamplificadores y motores son Maxon Motors de Suiza y los productos de ADVANCED Motion Controls de los Estados Unidos.

III.1.15 Principales algoritmos de control para motores en la industria

En la medida en que el área de control automático fue desarrollándose y en especial después de que los sistemas de control discretos fueron iniciados, un constante flujo de matemáticas avanzadas se introdujo en el diseño de controladores, de esta forma diferentes campos y métodos de control se fueron desarrollando de forma independiente, a continuación se mencionan algunos:

- Compensación Adelanto-Atraso
- Control PID
- Control óptimo
- Control Adaptativo
- Control robusto
- Control No-Lineal
- Modelos de control predictivo

Estas áreas están bien desarrolladas. Hoy en día es sencillo encontrar buenos libros sobre el tema y también publicaciones científicas que cubren cada aproximación al control automático, por citar algunos: Kuo y Golnaraghi (2009), Franklin et al. (1997), Astrom y Wittenmark (2008), Isidori (1995), Morari y Zafiriou (1989) y Grüne y Pannek (2011).

Es importante mencionar que los métodos de control, excepto el compensador adelanto-atraso y el PID, requieren de una plataforma digital para su implementación; de hecho en toda el área del control automático inició con los controladores analógicos (PID y compensación adelanto-atraso). Sin embargo, aún para sencillos problemas de control de movimiento, la introducción del control digital dio un gran impulso y flexibilidad y un mejor desempeño incluso con sus limitaciones. Una comparativa detallada de las ventajas y desventajas de estas dos tecnologías dando un énfasis en las aplicaciones de control de movimiento puede ser encontrada en Le-Huy (1994).

Los sistemas de control PID han sido suficientes para resolver el problema de control de muchas aplicaciones en la industria, particularmente cuando la dinámica del proceso lo permite (en general, los procesos que pueden ser descritos por sistemas de primer y segundo orden).

Una de las principales razones para que el controlador PID continúe siendo ampliamente utilizado se dio cuando la industria se trasladó del control analógico al control digital. Nuevas características como el ajuste de las ganancias a través de un puerto de comunicación (RS-232/485, USB (Universal Serial Bus), Ethernet, CAN (Controller Area Network), entre otros) tuvieron un gran impacto en las tecnologías y los métodos de control industrial, dando paso cada día a sistemas de control más complejos en comparación con los controladores estudiados en la teoría de control clásico, éstos son los sistemas de control inteligente. A pesar de los avances en

el área de control, el controlador clásico PID se encuentra en uso de manera generalizada por las siguientes ventajas: su simplicidad, ya que no requiere del conocimiento del modelo matemático exacto de la planta y existen diferentes técnicas de identificación que pueden ser aplicadas. El usuario, con un simple ajuste en tres ganancias, puede lograr la respuesta deseada del sistema; también existen un gran número de herramientas de software de diseño asistido por computador (CAD) para obtener un controlador, evaluar la estabilidad y el rendimiento, así como para poder realizar una simulación (MATLAB/Simulink), además de todos los métodos de sintonización basados en técnicas lineales sencillas y métodos heurísticos.

La industria ha desarrollado variantes e interesantes mejoras en el controlador PID para movimiento de motores. Una de las piezas del controlador que ha mostrado cambios y mejoras es el sistema de retroalimentación. En la Tabla III.2 se muestra los dispositivos de retroalimentación comúnmente disponibles para el control de motores, una comparativa en cuanto al costo, resolución y rendimiento de algunos de los transductores utilizados en el control de movimiento puede ser encontrada en (Cano, 2010).

Tabla III.2 Variables usualmente retroalimentadas y sus transductores (Mentz, 2000).

Variable retroalimentada	Fuente
Fuerza Electromotriz (FEM)	Medición a través de un convertidor analógico digital (ADC), calculada a partir de una señal PWM también o la velocidad
Corriente	Sensor de efecto Hall, resistencia shunt y toroide
Aceleración	Encoder o Resolver, odómetro
Velocidad	Encoder, Resolver o tacómetro
Posición	Encoder, Resolver, potenciómetro o un transformador diferencial variable lineal (LVDT)

Por lo general, el voltaje se manipula para controlar la corriente, que es el parámetro de retroalimentación más rápido. Un cambio en la corriente se ve obstaculizado principalmente por la inductancia de las bobinas y, en un grado mucho menor, por la fuerza contra-electromotriz (FCEM), que es proporcional a la velocidad del motor.

Todos los demás parámetros controlados (la aceleración, velocidad y posición) se ven amortiguados en su tasa de cambio por la inductancia del embobinado y la inercia del movimiento del sistema. Todos los sistemas tienen inercia positiva, por lo que un cambio en el sentido de la corriente sucederá más rápido de lo que el sistema mecánico pueda cambiar la aceleración, la velocidad o la posición. En la práctica, la rapidez de la tasa de cambio de la corriente es mayor que diez veces respecto a los demás parámetros. Esto convierte a la corriente como la variable para el modelo de todo el sistema de potencia; al amplificador de corriente (servoamplificador) y al motor, como un bloque ideal que proporciona la corriente deseada. Debido a que el torque es una constante cuando se modela el sistema, la etapa de potencia es un bloque que es tratado como el encargado de proveer el torque deseado, especialmente cuando se modela un sistema de control de velocidad o posición.

A continuación se describe a detalle el encoder o decodificador incremental que es el sensor que se utilizara como dispositivo de retroalimentación y por lo tanto es pieza clave en el sistema de control.

III.1.16 Decodificador Incremental (Encoder)

Este consiste en un codificador óptico digital, el cual es un dispositivo que convierte el movimiento en una secuencia digital de pulsos. Al contar un solo pulso o decodificar un conjunto de pulsos, estos pueden interpretarse como mediciones de la posición relativa o absoluta. Los codificadores o encoders pueden ser lineales o rotativos, pero el tipo más común es rotativo. Los codificadores rotatorios son fabricados en dos formas básicas: el codificador absoluto en donde, una sola palabra digital corresponde a la posición absoluta de un giro del eje, y el codificador incremental que produce pulsos cuando el eje gira, permitiendo la medición en relación al desplazamiento del eje. Como se ilustra en la Figura III.12, la mayoría de los codificadores rotativos esta compuesto por un disco plástico con un patrón radial organizado en

pistas. Cada pista interrumpe el haz entre un foto emisor y un foto detector de esta manera los pulsos se producen (Onwubolu, 2005).

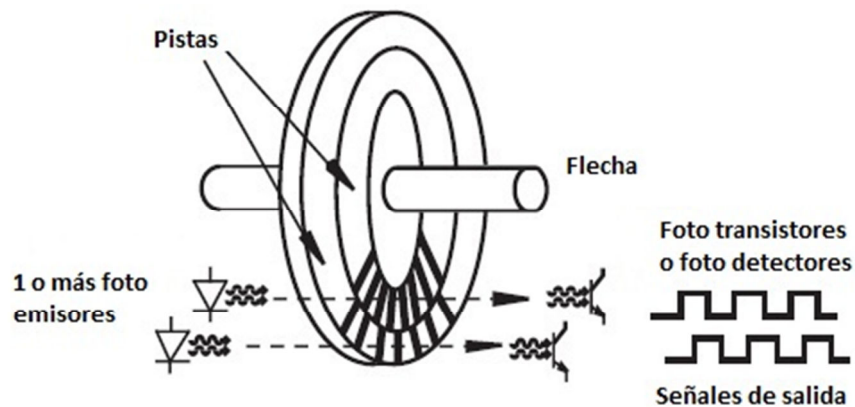


Figura III.12 Componentes de un codificador óptico.

Los decodificadores ópticos son ampliamente utilizados en aplicaciones donde se requiere la medición de la posición lineal o angular, la velocidad y dirección del movimiento. Estos no solo se utilizan como parte de las maquinas computarizadas si no también en muchos dispositivos de precisión de control de movimiento en procesos industriales para asegurar la calidad y seguridad de los equipos (Shetty *et al.*, 2010). Para el control de motores síncronos donde la posición absoluta del ángulo de rotor es esencial los decodificadores absolutos han sido ampliamente utilizados, sin embargo con el desarrollo de las técnicas de control en el área de maquinas eléctricas los decodificadores incrementales con varias señales auxiliares han ganado popularidad para el control de maquinas síncronas en la industria debido a su costo efectivo; en particular para el control de maquinas de DC o AC, en donde la posición no es utilizada para la regulación de velocidad; los decodificadores incrementales son utilizados sin excepción (Sul, 2011).

El decodificador incremental, a veces llamado un encoder relativo, se compone de dos pistas y dos sensores de cuya las salidas están designadas como canal A y canal B. A medida que el eje gira, trenes de pulsos se producen en A y B con una frecuencia proporcional a la velocidad del eje, y la relación de fase de adelanto-atraso entre las señales determina la dirección de rotación como se describe en detalle a continuación. Al contar el número de pulsos y conocer

la resolución del disco se puede determinar la posición angular. El número de pulsos por revolución (P_{PR}) es un factor importante en la selección de un decodificador óptico incremental. Las señales A y B están desfasadas $1/4$ de la fase (90°) de uno con el otro y son conocidas como señales de cuadratura. A menudo, una tercera señal llamada índice (I), la cual proporciona un pulso por revolución, que es útil en el conteo de vueltas completas. También es útil para definir un de referencia o punto cero. La Figura III.13 muestra una configuración con dos pistas separadas para los canales A, B e I, a menudo los decodificadores de cuadratura industriales proporcionan las señales complementarias de estos canales \bar{A} , \bar{B} e \bar{I} ; o hacen combinación de los decodificadores relativos y absolutos como se muestra en la Figura III.13, esto comúnmente en aplicaciones de robótica, en donde las señales o pulsos (U, V, W) normalmente son proporcionados para identificar la posición absoluta los trenes de pulsos producidos por estas señales están desfasadas 120° una de la otra, por lo tanto la posición absoluta puede ser determinada con precisión de $\pm 30^\circ$. En la mayoría de las aplicaciones industriales decodificadores incrementales con menos de 8192 P_{PR} son utilizados, sin embargo para aplicaciones con gran dinámica y precisión en el control de posición y velocidad resoluciones de hasta 223 P_{PR} están actualmente disponibles, esas resoluciones usualmente son obtenidas a través de la interpolación de algunos miles de P_{PR} , utilizando esta elevada cantidad de P_{PR} es posible en ocasiones controlar la velocidad en un amplio rango que va desde algunas revoluciones por minuto (RPM) a miles de RPM (Sul, 2011).

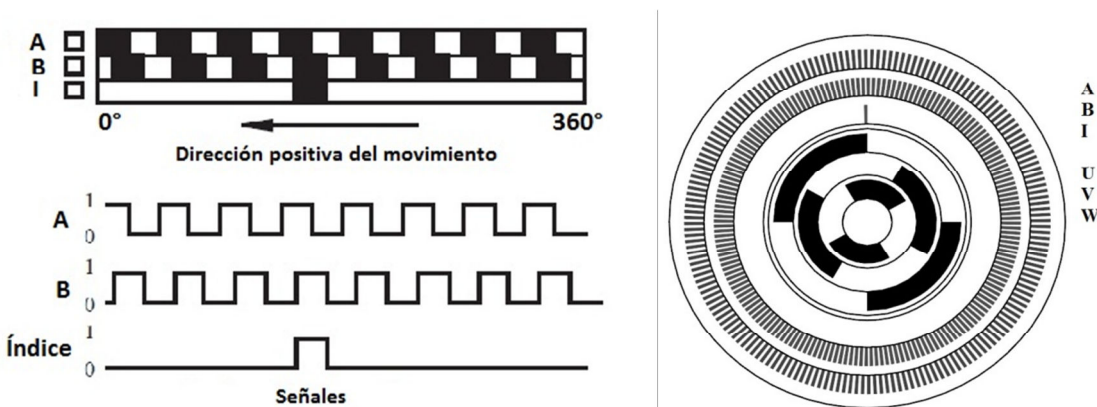


Figura III.13 Patrón de las pistas y las señales generadas de un decodificador incremental.

La cuadratura de las señales A y B pueden ser decodificadas para medir el desplazamiento angular y la dirección de rotación, como se muestra en la Figura III.14. Los pulsos aparecen en una de las dos líneas de salida que corresponden al giro en sentido de las manecillas del reloj (clockwise, CW) y la rotación en sentido anti horario (counterclockwise, CCW). La decodificación de las transiciones de A y B se realiza utilizando circuitos lógicos secuenciales estos puede ofrecer tres diferentes resoluciones: 1X, 2X y 4X. La resolución 1X proporciona un pulso de salida en cada flanco negativo de la señal A o B, lo que resulta en un solo pulso para cada ciclo o periodo de la señal de cuadratura. La resolución 2X ofrece un pulso de salida en cada flanco negativo y positivo de la señal A o B, lo que resulta en el doble de pulsos de salida. La resolución 4X ofrece un pulso de salida en cada flanco positivo y negativo de la señal A y B, dando lugar a cuatro veces el número de pulsos de salida. El sentido de giro está determinado por el nivel de una alguna de las señales de cuadratura respecto a la transición de la segunda señal. Por ejemplo, en el modo 1X, si en la señal A durante la transición de bajada ($A=\downarrow$) el nivel lógico de la señal B esta en alto ($B=1$) implica que esta girando en el sentido de las manecillas del reloj (CW), mientras que si $B=\downarrow$ y $A=1$ implica que esta girando en el sentido anti horario (CCW). De esta forma si solo se tuviera una señal en lugar de dos seria imposible determinar la dirección de rotación.

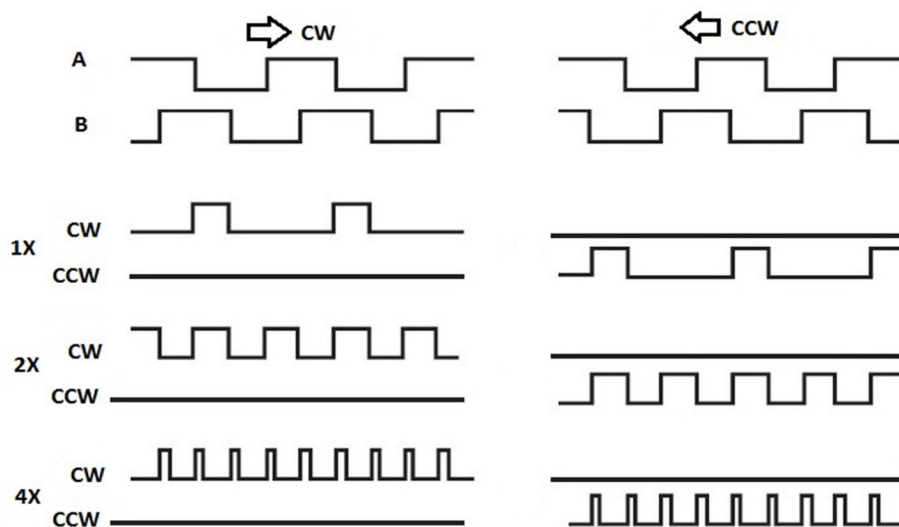


Figura III.14 Decodificación de las señales de cuadratura y detección del sentido de giro.

Diferentes circuitos puede ser diseñados para detectar los bordes de las señales de cuadratura y realizar un decodificación a 1X, 2X y 4X, estos circuitos pueden ser construidos con componentes discretos, pero también hay circuitos integrados como el HCTL-2016, HCTL-2032/2022, LS7183/4 entre otros. Además la decodificación de cuadratura también se puede realizar mediante software por medio de interrupciones en un μC o DSP. Los decodificadores incrementales proporcionan una mayor resolución a un menor costo que los decodificadores absolutos, pero sólo miden el movimiento relativo y no proporcionan la posición absoluta directamente. Sin embargo, un decodificador incremental se puede utilizar en combinación con un sensor de final de carrera (limit switch) para definir la posición absoluta con respecto a una posición de referencia definida por el sensor. Los decodificadores absolutos son elegidos en aplicaciones donde el establecimiento de una referencia posición es imposible o inadecuada. A continuación en la Figura III.15 se muestra un circuito muy sencillo para la decodificación a 1X de las señales de cuadratura, en donde los componentes U1-U4 son Flip-Flops tipo D que en adelante serán referenciados como FFD, la señal de reloj (CLK) y una señal asíncrona de restablecimiento (ARST), en la salidas CW y CCW se muestran los pulsos dependiendo del giro del motor como se muestra en la figura III.15, los componentes U1 y U2 sirven para sincronizar las señales A y B, estos pueden ser omitidos cuando se requiere de un diseño simple pero pueden presentarse fenómenos de metaestabilidad (Ashenden, 2008) debido a la naturaleza asíncrona de las señales de cuadratura.

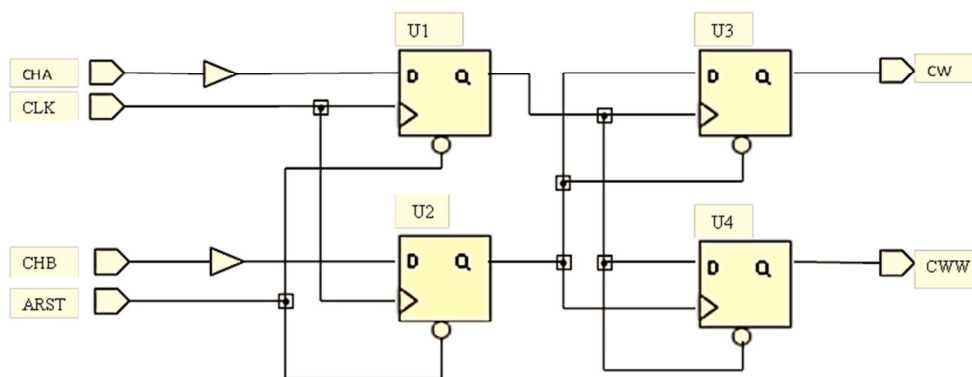


Figura III.15. Circuito para decodificación de las señales de cuadratura a 1X.

III.1.17 Estimación de la velocidad a partir de un encoder incremental

Partiendo del concepto de velocidad, definido como la razón de cambio de un desplazamiento con respecto al tiempo, es posible estimar la velocidad a partir de las mediciones de la posición en un intervalo de tiempo definido, es decir aplicando la derivada, utilizando la transformación discreta de Euler hacia atrás (“Backward Euler”) podemos estimar la velocidad, definida como:

$$\omega(k) = \frac{\theta(k) - \theta(k-1)}{T_s} \quad (5)$$

En donde $\omega(k)$ es la velocidad estimada actual, $\theta(k)$ la posición actual y T_s el tiempo de muestreo. Este método de estimación tiene la desventaja de que introduce ruido en la señal debido a la diferenciación, además factores como los la cantidad de P_{PR} es un factor importante en la precisión de la estimación y control de la velocidad, con mas P_{PR} el controlador tiene un ancho de banda mayor en la regulación (Kubo, 1990), sin embargo un decodificador con un numero mayor de P_{PR} incrementa el costo y además es sensible a golpes y ruidos mecánicos, sin embargo los P_{PR} pueden ser incrementados con las técnicas de decodificación de las señales de cuadratura. A pesar de incrementar los P_{PR} , debido a errores en el periodo entre los trenes de pulsos y de fase de las señales de cuadratura, el numero de P_{PR} en un intervalo de tiempo no son iguales, incluso a velocidad constante (Sul, 2011). Normalmente utilizando este método es necesario agregar un filtro pasa bajas (LPF) debido al efecto amplificador del ruido, un filtro de primer orden es utilizado en donde \hat{V} es la velocidad estimada a la salida del filtro, la función en tiempo continuo de un filtro de primer orden puede ser escrito como:

$$\frac{d\hat{\omega}}{dt} = \frac{1}{\tau_c} (\omega - \hat{\omega}) \quad (6)$$

En donde $\tau_c = \frac{1}{2\pi f_c}$ es la constante de tiempo del filtro pasa bajas y f_c es la frecuencia de corte, aplicando una aproximación hacia atrás tenemos que la velocidad estimada es:

$$\hat{\omega}(k) = K_1 \omega(k) + K_2 \hat{\omega}(k-1) \quad (7)$$

En donde $K_2 = \frac{\tau_c}{\tau_c + T_s}$ y $K_1 = \frac{1}{\tau_c + T_s}$

Sin embargo esta técnica tiene sus desventajas y ha motivado el desarrollo de nuevas técnicas, debido al hecho que la velocidad obtenida por medio de la posición a través de una simple diferenciación de pulsos resulta en una pobre resolución espacial debido a la cuantificación (Kavanagh, 2001).

(i) El método “M”

En este método se cuentan el número de pulsos en un intervalo de tiempo fijo, como se muestra en la Figura III.16, la velocidad estimada por medio de este método es representada como:

$$\omega_M = \frac{m\alpha}{T_{sp}} = \frac{60m}{P_{PR}T_{sp}} \text{ (RPM)} \tag{8}$$

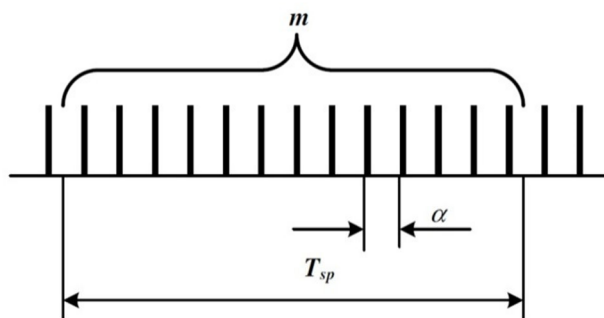


Figura III.16 Método “M”.

En donde α representa el ángulo rotacional en grados entre dos pulsos adyacentes y T_{sp} el tiempo de muestreo de la velocidad, y m el número de pulsos en el intervalo T_{sp} . El método “M” tiene la ventaja de que su implementación es simple y el tiempo de muestreo para toda velocidad es constante, sin embargo el error de cuantificación, que está expresado por $\frac{60}{P_{PR}T_{sp}}$ (RPM), es constante para cualquier velocidad, y para bajas velocidades la precisión relativa y la resolución es menor. Con un decodificador incremental con algunos miles de P_{PR} , el error de medición debería de ser de algunas decenas de RPMs con un intervalo de muestreo de varios cientos de microsegundos; y por lo tanto el desempeño de un control de velocidad sería inaceptable para la mayoría de las aplicaciones de servo control. Sin embargo, si se utiliza un

decodificador con una gran cantidad de P_{PR} por ejemplo 223 P_{PR} , el cálculo de la velocidad puede realizarse con un intervalo de muestreo de algunos cientos de microsegundos y el ancho de banda del controlador puede incrementarse varios cientos de rad/s. La precisión relativa de este método es una función que depende directamente de la velocidad y esta expresada por la siguiente fórmula, (Petrella et al., 2007).

$$\varepsilon_{\omega} = \frac{2\pi}{\omega_M P_{PR} T_{sp}} 100 \quad (\%) \quad (9)$$

(ii) El método “T”

En este método se mide el intervalo de tiempo entre dos pulsos adyacentes, como es muestra en la Figura III.17, en donde la velocidad puede ser calculada como:

$$N_T = \frac{\alpha f_c}{m} = \frac{60 f_c}{m P_{PR}} \quad (\text{RPM}) \quad (10)$$

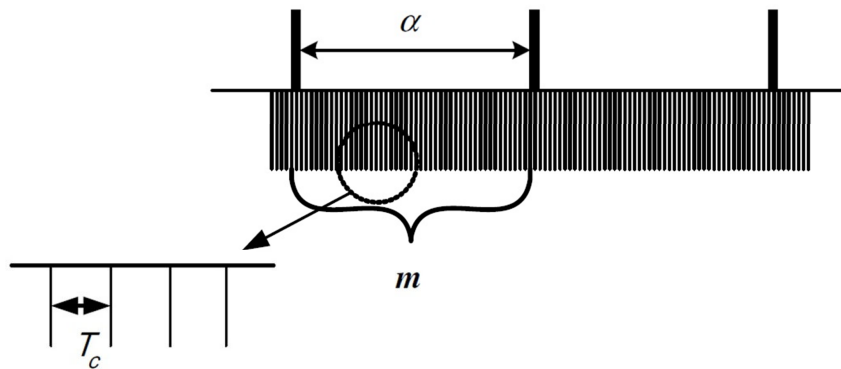


Figura III.17 Método “T”.

En donde f_c representa la frecuencia de un tren de pulsos de alta velocidad que sirven para medir el intervalo de tiempo (t_c) entre los pulsos de cuadratura, en donde $f_c = \frac{1}{t_c}$. El método “T” tiene la ventaja de poder medir la velocidad a una baja velocidad y el retardo entre las mediciones es insignificante. Sin embargo este método necesita una división aritmética para el cálculo de la velocidad lo cual normalmente es una carga para un microprocesador de bajo costo y el intervalo de muestreo varía de acuerdo a la velocidad. Además la velocidad calculada puede tener un retardo adicional respecto al tiempo de muestreo del controlador de velocidad que

es constante y esta de sincronizado la ejecución de el algoritmo de control y el calculo de la velocidad dada la naturaleza asíncrona de las señales de cuadratura. Además, para obtener una resolución alta a una muy baja velocidad, α debería ser muy grande y la cantidad de pulsos de reloj para medir el intervalo de tiempo (t_c) sería muy grande (Sul, 2011).

(iii) El método “M/T”

En la industria este método es utilizado ampliamente debido a que tiene las ventajas de ambos métodos y reduce las limitaciones que presentan los métodos por si solos. El principio de este método se muestra en la Figura III.18, en donde el intervalo de medición, T_D , es sincronizado con el pulso justo después de tiempo nominal de muestreo, T_{sp} . Mientras que T_D que es el intervalo de muestreo de la velocidad que es un poco mas grande que T_{sp} por una diferencia de ΔT para la mayoría de las velocidades de operación. La velocidad por medio de este método puede ser representada por:

$$N_{M/T} = \frac{m_1 \alpha}{P_{PR}(T_{sp} + \Delta T)} = \frac{60 f_c m_1}{P_{PR} m_2} \quad (\text{RPM}) \quad (11)$$

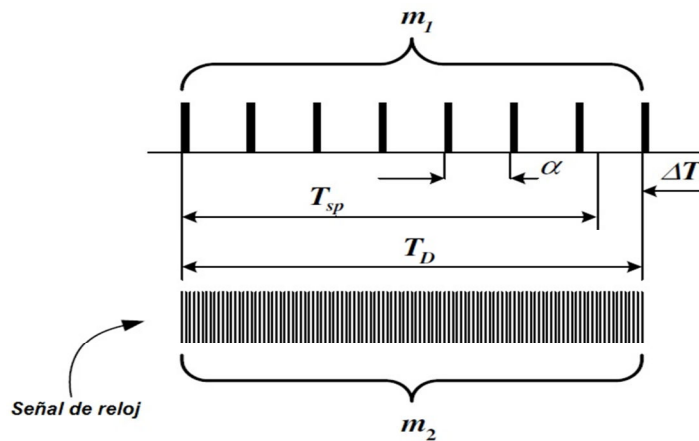


Figura III.18 Método “M/T”.

En donde m_1 y m_2 son valores enteros discretos, basados en esos valores podemos calcular la velocidad. El método “M/T” combina el método “T” el cual proporciona una

medición precisa a bajas velocidades y el método “M” para mediciones precisas en altas velocidades, de esta forma se tiene una medición razonablemente precisa en un amplio rango de velocidades, sin embargo a extremadamente bajas velocidades se comporta como el método “T” y eventualmente ΔT puede ser mas grande que el tiempo de muestro nominal T_{sp} , lo cual por debajo de una cierta velocidad el ancho de banda del controlador se reduciría de manera drástica (Sul, 2011).

La resolución de la medición de la velocidad Q_N , y el actual intervalo de muestreo T_D , pueden se representados por las siguientes ecuaciones:

$$Q_N = \frac{60f_c m_1}{P_{PR}} \left(\frac{1}{m_2 - 1} - \frac{1}{m_2} \right) = \frac{60f_c m_1}{P_{PR} m_2 (m_2 - 1)} \quad (\text{RPM}) \quad (12)$$

$$\left\{ \begin{array}{l} T_D = T_{sp} \quad (\Delta T = 0) \\ T_D = \left\{ \left[\text{Integer} \left(\frac{P_{PR} N_{M/T} T_{sp}}{60} \right) \right] + 1 \right\} \frac{60}{P_{PR} N_{M/T}} \quad (\Delta T \neq 0) \end{array} \right. \quad (13)$$

En donde $\text{Integer}(x)$ significa es la parte entera del número x .

III.1.18 Controladores de velocidad

Los controladores de velocidad, basados en estrategias de control PID simples, son comúnmente encontrados en gran cantidad de equipos comerciales (PDM Corp, 2011), en especial una configuración conocida con “feedforward” o pre alimentado, este controlador puede ser encontrado en productos como “Kollmorgen MMC Digital Smart Drives”, “Delta-Tau PMAC” (Mentz, 2000). La diferencia respecto a la configuración típica del controlador PID es que éste posee una pre alimentación de la velocidad y la aceleración, la configuración es típicamente conocida como “PID with acceleration and velocity feedforward” y es encontrada en la mayoría de los productos comerciales dedicados al control de movimiento. En la Figura III.19 se muestra en el diagrama a bloques un control de velocidad.

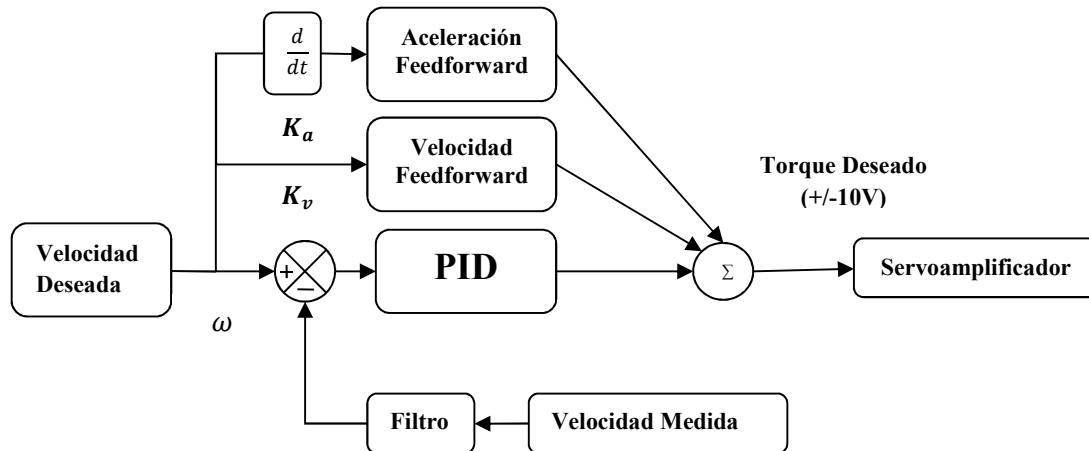


Figura III.19 Controlador de velocidad industrial típico.

- **La ganancia de velocidad ($K_{f\omega}$):** Esta ganancia es utilizada para cancelar la fricción en el motor; está es típicamente considerada y modelada como un coeficiente que es linealmente proporcional a la velocidad, de tal forma que la ganancia puede ser establecida de manera que no se requiera de una ganancia integral para mantener una velocidad constante. Un problema que se presenta al utilizar esta ganancia es que la fricción no es una variable lineal en la realidad y no incrementa linealmente conforme la velocidad, por lo que una ganancia que es correcta para una velocidad específica, será demasiado grande para una velocidad mayor y ésta rápidamente puede desestabilizar el control, de tal manera que la ganancia $K_{f\omega}$ debe ser propiamente ajustada considerando el valor máximo de la velocidad, lo cual trae como consecuencia que para velocidades pequeñas la ganancia integral será requerida para mantener la velocidad correcta, es decir para que el error en estado estacionario sea cero.
- **La ganancia de aceleración ($K_{f\epsilon}$):** Esta ganancia esta directamente relacionada con la segunda ley de newton, que en su forma rotacional es descrita por:

$$F = m * a = J * \dot{\omega} \quad (14)$$

En donde F , es la fuerza generada, m la masa, a la aceleración y de manera reciproca para un sistema rotacional, J es la inercia y ω la velocidad angular. Para sistemas puramente inerciales o sistemas en donde la fricción sea muy baja la ganancia $K_{ff\varepsilon}$ actúa siguiendo la ecuación (14) ofreciendo buenos resultados, sin embargo tiene un problema similar a $K_{ff\omega}$ y esta debe ser ajustada alrededor de la máxima velocidad de operación, la ganancia de aceleración requiere que se realice una derivación numérica de ω , de tal manera que el ruido en la señal será amplificado, por lo que un mal ajuste puede causar inestabilidad, en general las ganancias “feedforward” causan inestabilidad si son ajustadas por encima de su valor nominal por lo que un ajuste por debajo de su valor es recomendado.

III.1.19 Controladores de posición

El control de posición al igual que el de velocidad, industrialmente el controlador PID domina el mercado de estas aplicaciones, este puede ser encontrado en diferentes configuraciones que han sido ampliamente explotadas en diferentes áreas no solo en la industria si no en trabajos de investigación, a continuación mencionaran brevemente las configuraciones típicas comúnmente encontradas en productos industriales así como en trabajos de investigación en el área.

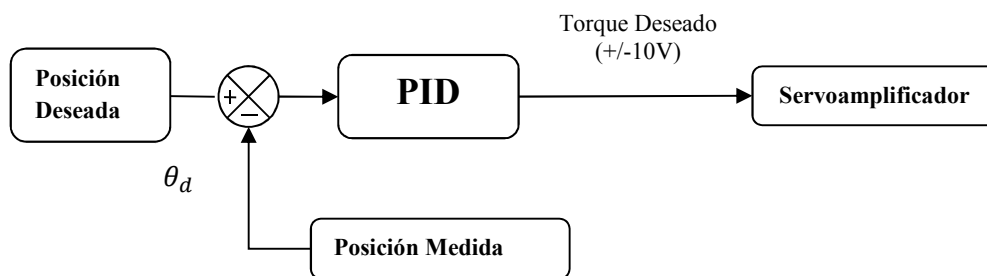


Figura III.20 Controlador de posición académico.

En la Figura III.20 se muestra la configuración del controlador PID simple que es encontrado principalmente en aplicaciones escolares y de con fines de investigación (Altayef,

2007), (Osornino, 2007), (León, 2009), (Sandoval, 2011), en donde el lazo de corriente esta embebido como parte del servoamplificador y es tratado de manera ideal como una ganancia; una de las limitaciones del controlador PID es que el termino proporcional requiere del error para producir un salida además el termino integral requiere del error y de tiempo, estas características son indeseables en muchas aplicaciones de control de movimiento, por ejemplo en aplicación punto a punto en donde se requiere una rápida respuesta; otra configuración popular utilizada ampliamente en las aplicaciones académicas es el control en cascada en donde un lazo de mayor rapidez es anidado dentro de uno de menor, en esta configuración el lazo de control de velocidad se considera de mayor rapidez de tal manera que la velocidad del motor cambia rápidamente para alcanzar la posición deseada, esta configuración se tiene un segundo orden y gracias a la acción integral en el lazo de posición y de velocidad, esto permite un error de cero ante una entrada tipo rampa a diferencia del PID simple, esta configuración a pesar de sus ventajas es muy inusual en la industria debido a que requiere que se ajusten ambos lazos de control, además existe la tendencia de que el ajuste del lazo de velocidad proporcione la respuesta mas rápida posible sin importar el sobrepaso; sin embargo la respuesta ideal para un control de posición es una respuesta críticamente amortiguada en su velocidad, de esta manera se evita cualquier sobrepaso, lo cual es critico en maquinas herramientas.

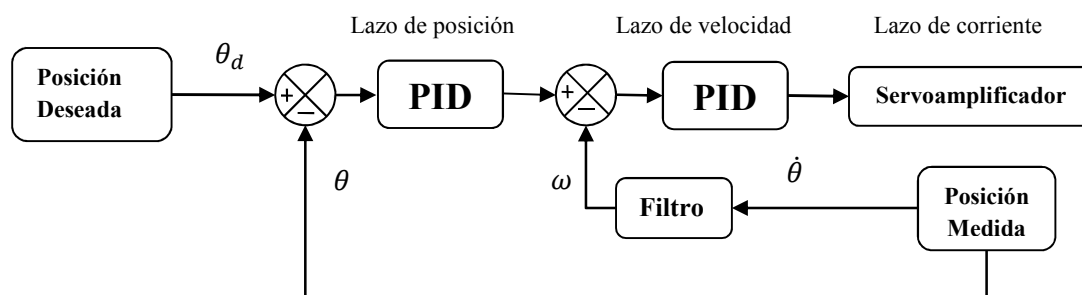


Figura III.21 Controlador de posición en cascada

En la Figura III.21 se muestra un controlador de movimiento industrial tal y como se plantea en la Figura III.3, en donde el lazo de mayor rapidez es el de la corriente o torque, seguido del control de velocidad y finalmente el lazo de posición.

En el esquema de la Figura III.22 se muestra un controlador que fue ampliamente utilizado en controladores industriales pasados, este esquema de control puede ser obtenido del esquema anterior en donde simplemente en el lazo de velocidad la ganancia proporcional es uno ($K_p = 1$), y las demás cero ($K_i = 0, K_d = 0$).

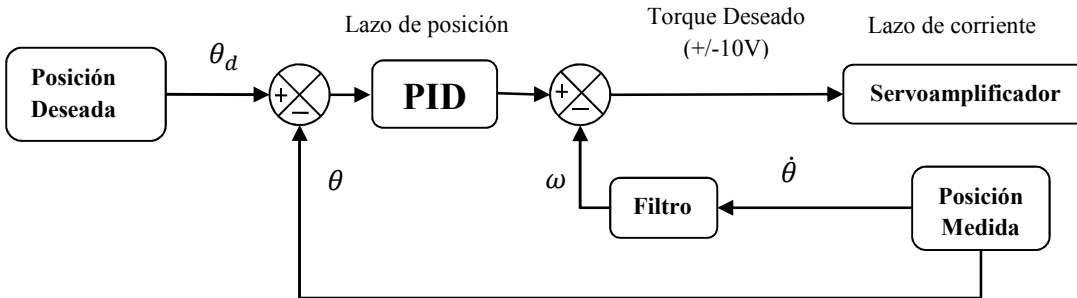


Figura III.22 Controlador de posición con alimentación de velocidad.

Uno de los esquemas PID de control de movimiento ampliamente utilizado por los controladores modernos es la configuración que se muestra en la Figura III.23, esta configuración es ampliamente recomendada cuando se requiere de cambios de posición pequeños, estos movimientos pequeños son definidos como cambios en donde la velocidad máxima del motor nunca es alcanzada; en donde la introducción de las ganancias pre alimentadas permite reducir el error de posición y velocidad ya que actúan directamente en la salida del controlador, mientras que los controladores con retroalimentación actúan con base en las mediciones pasadas, las ganancias pre alimentadas de velocidad y aceleración son elementos predictivos que utilizan la información del perfil de posición. Algunos productos comerciales en donde se puede encontrar esta configuración son EPOS Digital Motion Controller de Maxon Motors, el chip MC3110 de PMD Corp., PCI-8256 de ADLINK e incluso hay en el mercado núcleos de propiedad intelectual (IP Cores) como el “Universal Drive Controller” de ENCLUSTRA.

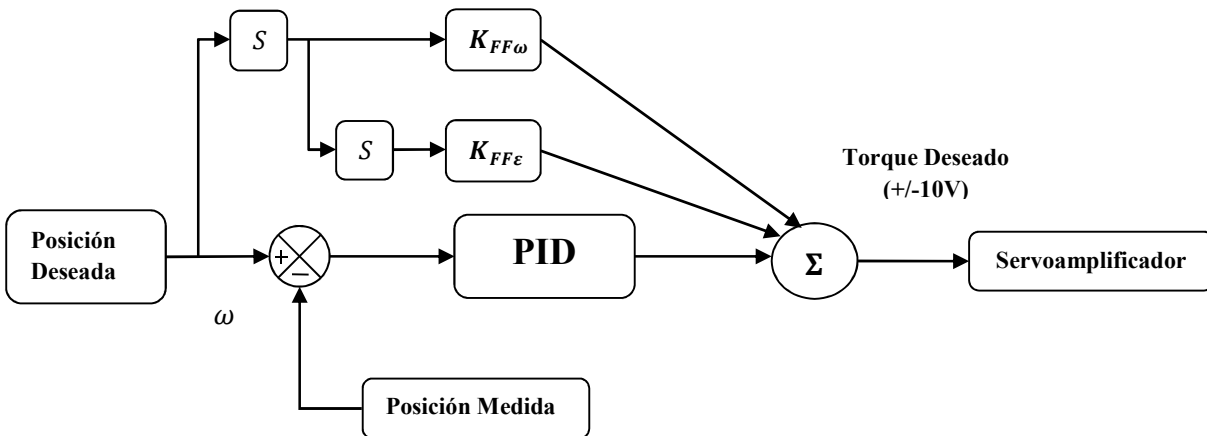


Figura III.23 Controlador de posición industrial feedforward.

De los controladores descritos existen diferentes variaciones y propuestas en diferentes trabajos de investigación (Ohm, 1994), en la Figura III.24 se muestra un controlador PDF (Pseudo-Drivative-Feedback), esta configuración ofrece un buen rechazo a las perturbaciones, mientras que la capacidades de seguimiento no son tan buenas como las del control PID.

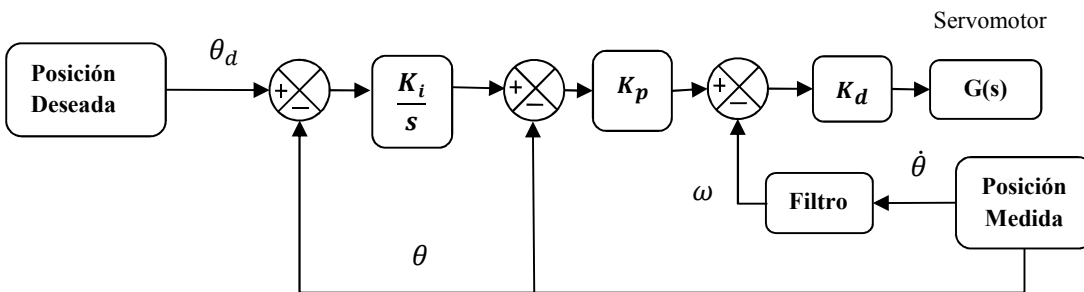


Figura III.24 Controlador de posición en configuración P+PDF.

Para mejorar las características de rechazo y de seguimiento, la introducción del control en cascada en combinación con las ganancias pre alimentadas de velocidad y aceleración, han mostrado una mejora en los sistemas de control de movimiento; en (Malek *et al.*, 2011) se presenta un análisis detallado de sus características y limitaciones.

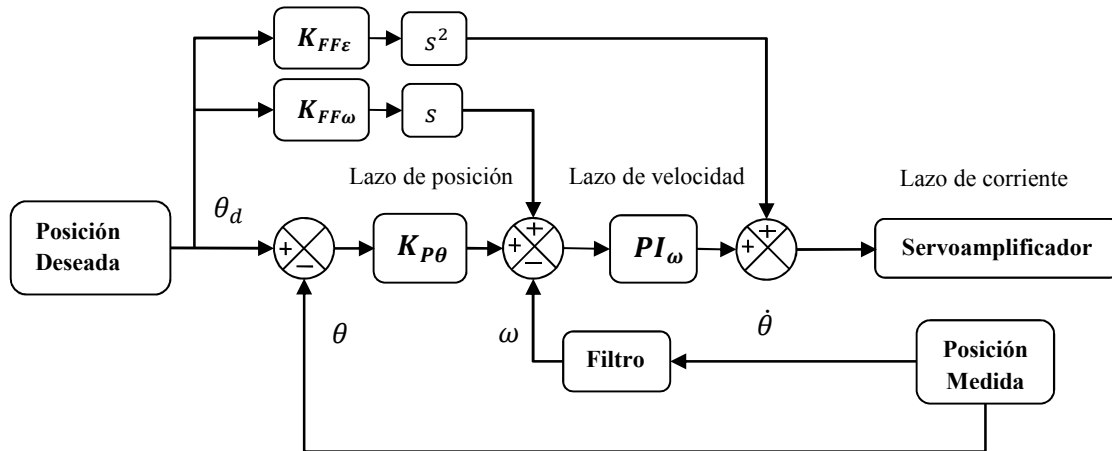


Figura III.25 Controlador de posición con lazo de velocidad PI y ganancias pre alimentadas de velocidad y aceleración (P-PI feedforward).

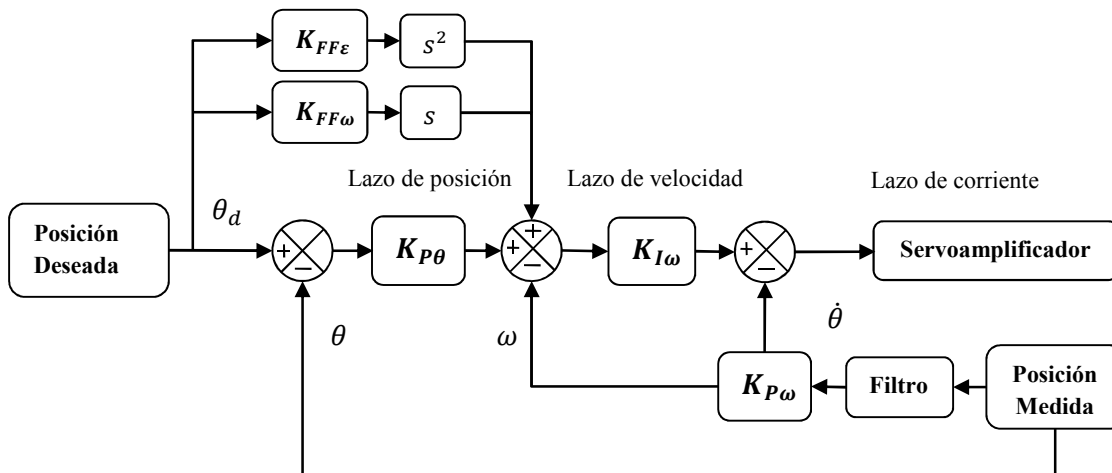


Figura III.26 Controlador de posición con lazo de velocidad PDF y ganancias pre alimentadas de velocidad y aceleración.

En (Ellis, 2004) se muestra una comparativa en cuanto al desempeño de algunas de las configuraciones de control de movimiento aquí mostradas, además presenta un análisis de sus ventajas y desventajas así como la áreas de aplicación.

III.1.20 Controlador PID discreto

El diseño del controlador PID discreto o digital, puede ser obtenido utilizando un numero de recetas bien conocidas en el campo del procesamiento digital de señales (DSP), donde son utilizadas para el diseño de filtros digitales, de hecho un controlador puede ser visto como un filtro que atenúa algunas dinámicas y acentúa otras para obtener un tiempo de respuesta deseado (Fadali, 2009), la forma mas común de aproximar un controlador analógico como el PID a uno digital consiste en convertir la función del controlador a una ecuación diferencial y después obtener una aproximación numérica de la solución de la ecuación diferencial, existen básicamente dos métodos para la aproximación numérica de la solución de una ecuación diferencial (Levine, 2011), estos son:

- La integración numérica.
- La diferenciación numérica.

La integración numérica es un importante problema computacional, la aproximación mas común es dividir el intervalo de integración en muchos subintervalos T y aproximar la contribución a la integral de cada intervalo T . Se considera la siguiente ecuación:

$$G_c(s) = \frac{F(s)}{E(s)} = \frac{1}{s} \quad (15)$$

La cual corresponde a la siguiente ecuación diferencial

$$\frac{df}{dt} = e(t) \quad (16)$$

Integrando en ambos lados de la ecuación desde t_0 a t se tiene:

$$f(t) = f(t_0) + \int_{t_0}^t e(t)dt, \quad t \geq t_0 \quad (17)$$

Para un intervalo de muestras tenemos, $t = kT, k = 0,1,2, \dots$ donde para un intervalo de muestreo tenemos, $t_0 = kT$ a $t = kT + T$, resolviendo (17), tenemos:

$$f(kT + T) = f(kT) + \int_{kT}^{kT+T} e(t)dt \quad (18)$$

A continuación se muestra la solución a este problema utilizando diferentes aproximaciones.

(i) El Método de Euler hacia adelante (Euler’s Forward)

Este es el método más simple para la aproximación de la integral de la ecuación (18), en donde es sencillo aproximarse a la integral utilizando una constante igual al valor del integrando por la izquierda al final del punto de cada subintervalo T y multiplicarlo por el intervalo de tiempo T, como se muestra en la Figura III.27, por lo tanto la solución a la ecuación (18) es:

$$\hat{f}(kT + T) = \hat{f}(kT) + Te(kT) \tag{19}$$

Aplicando la transformada $-Z$ en ambos lados de la ecuación:

$$z\hat{F}(z) - \hat{F}(z) = TE(z) \tag{20}$$

Por lo tanto tenemos que:

$$G_z(z) = \frac{\hat{F}(z)}{E(z)} = \frac{T}{z-1} \tag{21}$$

Esto implica que el equivalente discreto de un controlador analógico puede ser calculado con el método de “Euler’s Forward” simplemente reemplazando cada s en el controlador analógico con la función de la ecuación (22), como se muestra en la ecuación (23), por lo que tenemos una aproximación para la integral y la derivada en tiempo continuo utilizando la ecuación (21).

$$\frac{1}{s} \rightarrow \frac{T}{z-1} \quad s \rightarrow \frac{z-1}{T} \tag{22}$$

$$G_c(z) = G_c(s)|_{s=z-1/T} \tag{23}$$

(ii) El Método de Euler hacia atrás (Euler’s Backward)

Ahora en lugar de aproximarse por el lado izquierdo del final del intervalo, el método de “Euler’s Backward” se aproxima al integrando por el valor del lado derecho al final del subintervalo T y se multiplica por su intervalo de muestreo T como es muestra en la Figura III.27, por que la ecuación (18) se expresa como:

$$\hat{f}(kT + T) = \hat{f}(kT) + Te(kT + T) \tag{24}$$

Aplicando la transformada z se tiene:

$$G_z(z) = \frac{\hat{F}(z)}{E(z)} = \frac{z^T}{z-1} \quad (25)$$

De igual forma el equivalente discreto del controlador analógico puede ser encontrado utilizando:

$$\frac{1}{s} \rightarrow \frac{z^T}{z-1} \quad S \rightarrow \frac{z-1}{z^T} \quad (26)$$

$$G_c(z) = G_c(s)|_{s=z-1/z^T} \quad (27)$$

Normalmente los métodos “Euler’s Forward” y “Euler’s Backward” son llamados métodos rectangulares, debido a que durante el intervalo de muestreo, el área bajo la curva es aproximada mediante un rectángulo, además estos métodos son llamados de primer orden debido a que solo usan una muestra en cada intervalo de tiempo.

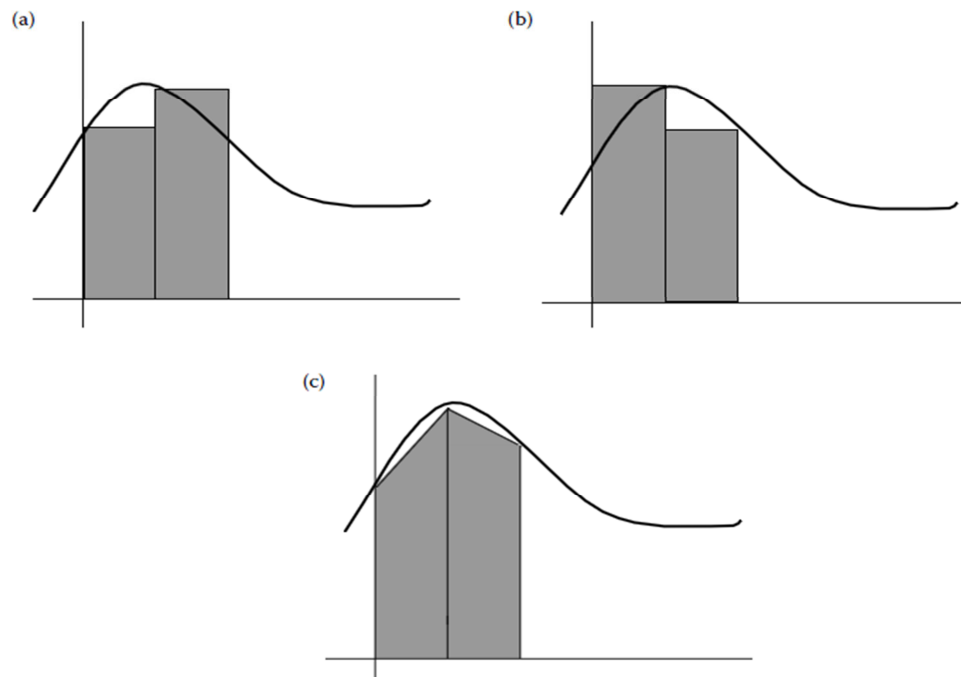


Figura III.27 Comparación de los métodos para la aproximación de la integral. (a) Aproximación utilizando “Forward Euler”. (b) Aproximación utilizando “Backward Euler”. (c) Aproximación utilizando el método trapezoidal. (Levine, 2011).

(iii) El Método de Trapezoidal (Tustin’s o Bilinear)

El desempeño de un controlador digital puede ser mejorado si durante su intervalo de muestreo se utiliza más de una muestra para actualizar la aproximación a un controlador analógico; como se muestra en la Figura III.27, la aproximación trapezoidal se aproxima utilizando una línea recta entre cada intervalo de muestreo, por que la ecuación (18) puede ser calculada a través de una aproximación trapezoidal, por lo que tenemos:

$$\hat{f}(kT + T) = \hat{f}(kT) + \frac{T}{2}\{e(kT) + e(kT + T)\} \tag{28}$$

Aplicando la transformada z se tiene:

$$G_z(z) = \frac{\hat{F}(z)}{E(z)} = \frac{T}{2} \left[\frac{z+1}{z-1} \right] \tag{29}$$

De igual forma el equivalente discreto del controlador analógico puede ser encontrado utilizando:

$$\frac{1}{s} \rightarrow \frac{T}{2} \left[\frac{z+1}{z-1} \right] \quad s \rightarrow \frac{2}{T} \left[\frac{z-1}{z+1} \right] \tag{30}$$

$$G_c(z) = G_c(s) \Big|_{s=\frac{2}{T} \left[\frac{z-1}{z+1} \right]} \tag{31}$$

La integral puede ser aproximada utilizando polinomios de mayor orden, sin embargo para la solución numérica de la integral, se utiliza una ecuación recursiva en donde solo debe involucrar valores presentes y pasados, no futuros. A continuación en la Tabla III.3 se muestran las aproximaciones para la integral (Levine, 2011).

Tabla III.3 Aproximaciones de la integral utilizando muestras presentes y pasadas.

Método	Ecuación de diferencias	Función en Z
Euler’s Forward	$\hat{f}[(k + 1)T] = \hat{f}(kT) + Te(kT)$	$\frac{z - 1}{T}$
Euler’s Backward	$\hat{f}[(k + 1)T] = \hat{f}(kT) + Te((k + 1)T)$	$\frac{z - 1}{zT}$
Tustin’s	$\hat{f}[(k + 1)T] = \hat{f}(kT) + \frac{T}{2}[e(k + 1)T] + \frac{T}{2}[e(k)T]$	$\frac{T}{2} \left[\frac{z - 1}{z + 1} \right]$

A continuación en la Figura III.28 se muestra la respuesta en frecuencia de un filtro el cual fue aproximado con los diferentes métodos presentados y donde se puede observar sus diferencias y como la aproximación Bilineal ofrece una mejor respuesta de un control o filtro analógico.

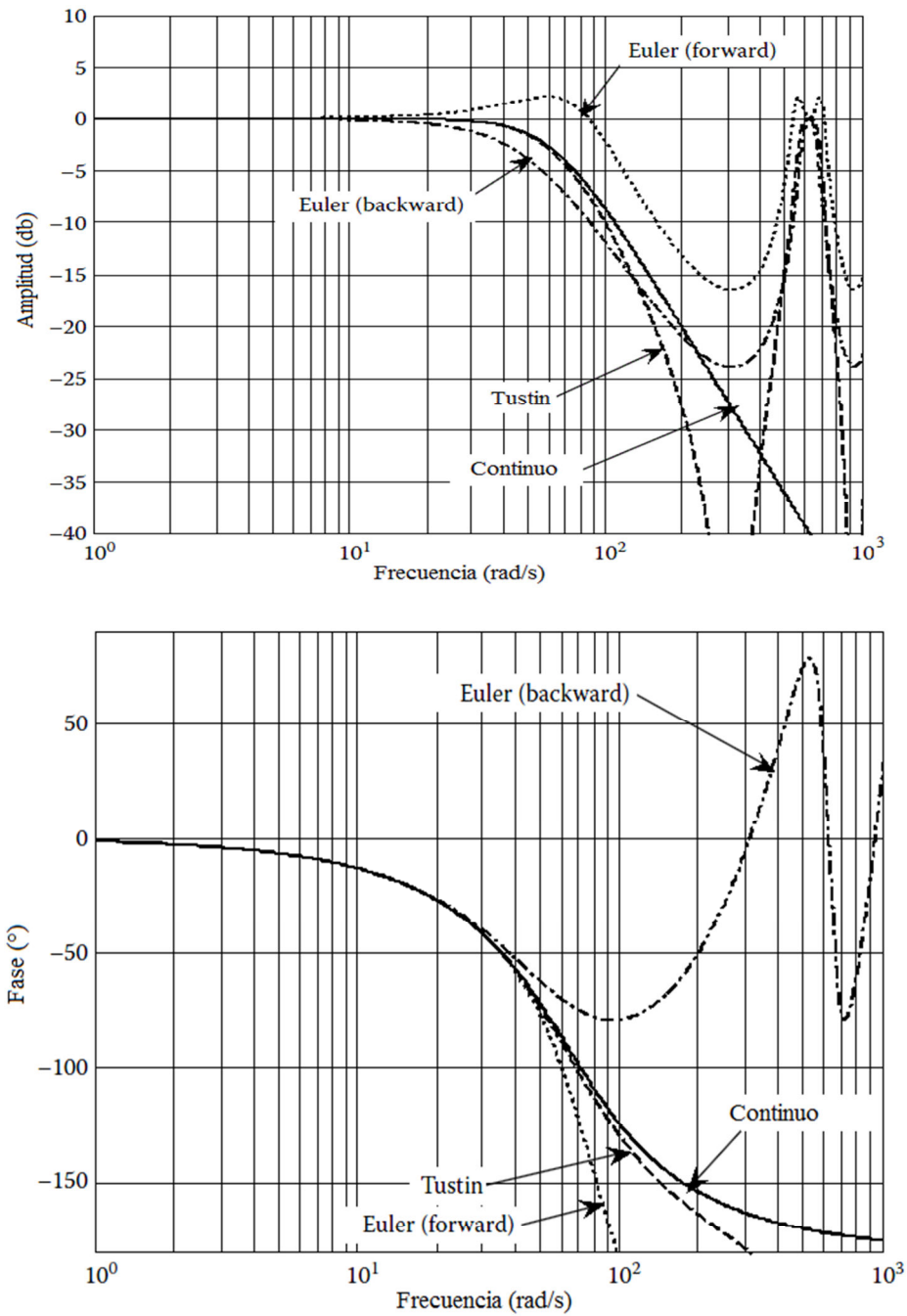


Figura III.28 Respuesta en frecuencia de un filtro Butterworth utilizando los métodos de Tustin y Euler (Levine, 2011).

III.1.21 Generación de Trayectorias

El generador de trayectoria en un controlador de movimiento es el responsable por la planeación de trayectorias basado en una geometría especificada, existen diferentes tipos de geometrías las cuales típicamente consisten en: líneas rectas, arcos circulares, arcos helicoidales y arcos esféricos, además de geometrías predefinidas, el pueden ser especificadas por medio de un conjunto de puntos y el controlador de movimiento utiliza algoritmos de interpolación para seguir esa geometría particular, esto se conoce como movimiento de contorno.

Además existen los perfiles de velocidad que serán utilizados para trazar la geometrías deseadas, estos perfiles son creados especificando parámetros del movimiento como lo son la máxima velocidad, máxima aceleración y la máxima desaceleración y el jerk o jaloneo (que es la tasa de cambio de la aceleración). Estos parámetros son utilizados para generar perfiles trapezoidales o curvas-s. El generador de trayectorias en un controlador de movimiento asegura que estas condiciones nunca son violadas mientras se realiza un movimiento. Además también se pueden generar perfiles donde se especifique una posición una velocidad y un tiempo deseado, perfiles (PVT). La salida del generador de trayectorias son los valores de referencia que alimentan al lazo de control, que actúa en base a las necesidades del perfil deseado. Normalmente los generadores de perfiles permiten cambios sobre la marcha, es decir este genera los valores de referencia en tiempo real, lo que permite detener y reiniciar un movimiento que esta en progreso.

(i) Perfiles de velocidad (Trapezoidal y Curvas-S)

El generador de trayectorias utiliza los valores de la posición, aceleración y velocidad de un movimiento en particular y determina cuando tiempo empleará en los tres principales segmentos del movimiento (aceleración, velocidad constante y desaceleración), estos definen el perfil de velocidad.

1) *Perfil Trapezoidal de velocidad*

Cuando se utiliza un perfil trapezoidal, el movimiento acelera a la tasa especificada y después mantiene una velocidad máxima, bajo la limitante que debido al tipo movimiento y la distancia cubierta puede resultar imposible alcanzar la máxima velocidad especificada, al final de su posición el movimiento desacelera, como se muestra en la Figura III.29.

Este perfil de movimiento puede se calculado utilizando las cuatro ecuaciones siguientes de cinemática para una aceleración constante:

$$v = v_o + at \tag{32}$$

$$x = x_o + \frac{1}{2}(v_o + v)t \tag{33}$$

$$x = x_o + v_o t + \frac{1}{2}at^2 \tag{34}$$

$$v^2 = v_o^2 + 2ax \tag{35}$$

En donde los parámetros de las ecuaciones anteriores son, x la posición o distancia, v la velocidad, a la aceleración y t el tiempo.

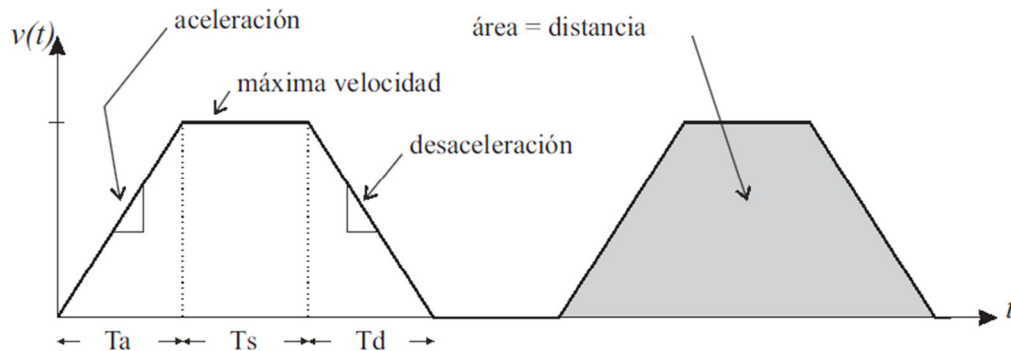


Figura III.29 Perfil de velocidad trapezoidal (González, 2006).

En donde se tiene que la velocidad, la distancia y la aceleración están dadas por las siguientes ecuaciones:

$$v(t) \begin{cases} \frac{v_{max}t}{T_a} \\ v_{max} \\ v_{max}(1 - \frac{t-(T_s+T_a)}{T_d}) \end{cases} \begin{cases} 0 < t < T_a \\ T_a < t < T_a + T_s \\ T_a + T_s < t < T \end{cases} \tag{36}$$

$$a(t) \begin{cases} \frac{v_{max}}{T_a} & \left\{ \begin{array}{l} 0 < t < T_a \\ T_a < t < T_a + T_s \\ T_a + T_s < t < T \end{array} \right. \\ 0 \\ -\frac{v_{max}}{T_d} \end{cases} \quad (37)$$

$$x_f = v_{max} \left(\frac{T_a}{2} + T_s + \frac{T_d}{2} \right) \quad (38)$$

En donde T_a es el tiempo de aceleración, T_s el tiempo de velocidad máxima, T_d el tiempo de desaceleración y T el tiempo total del perfil.

2) Perfil de curva-s de velocidad

Los segmentos de la aceleración y desaceleración en un perfil de curva-s son suaves, lo que resulta en una transición menos abrupta entre el segmento de aceleración\desaceleración y el segmento de velocidad constante, como se muestra en la Figura III.30, estas transiciones disminuyen el jerk en controlador de movimiento, sin embargo incrementa el ciclo de tiempo del movimiento; el valor con el cual el perfil es suavizado se llama máximo jerk deseado o el valor de la curva-s (Bishop, 2008). Las ecuaciones 32-35 no son suficientes para el calculo de estos perfiles debido a que asumen una aceleración constante con un jerk igual a cero, estas derivan de tomar la aceleración $a = dv/dt$ e integrar dos veces respecto al tiempo, en cambio iniciando con el jerk que es, $j = da/dt$ e integrando tres veces respecto al tiempo se obtienen las siguientes ecuaciones:

$$a = a_o + jt \quad (39)$$

$$v = v_o + a_o t + \frac{1}{2}jt^2 \quad (40)$$

$$x = x_o + v_o t + \frac{1}{2}a_o t^2 + \frac{1}{6}jt^3 \quad (41)$$

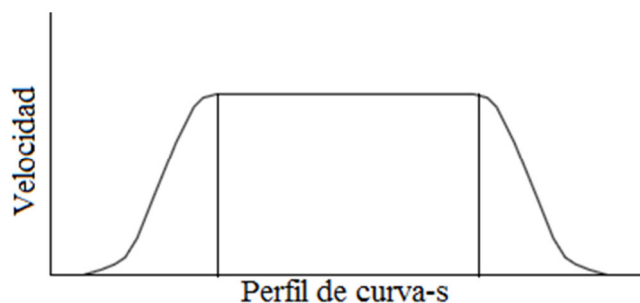


Figura III.30 Perfil de velocidad de curva-s (Bishop, 2011).

III.2 Metodología

En esta sección se abordaran y detallaran los pasos que mencionan en el capítulo anterior sobre como abordar un problema de control automático y sus pasos a seguir, uno de los problemas mas importantes en la ingeniería de control es la brecha que existe entre la teoría y la aplicación practica por lo que en esta sección se describirá con detalle muchos de estos aspectos, en el diagrama de flujo siguiente se describe la metodología que se emplea para la implementación en FPGA de cualquier algoritmo en general.

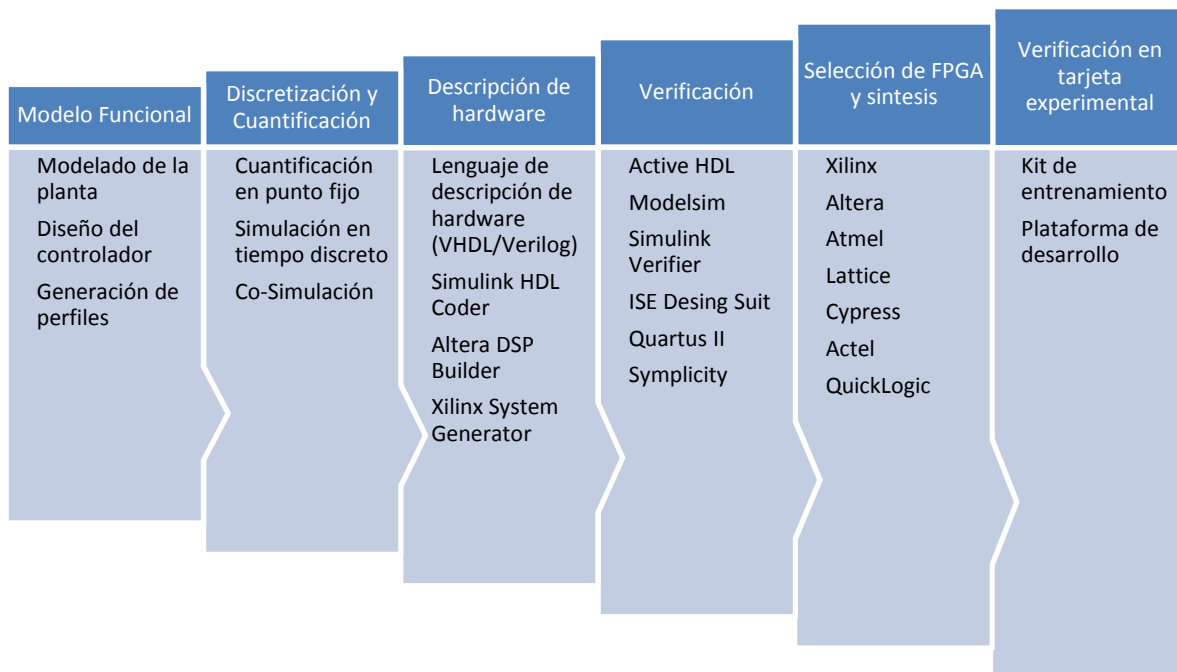


Figura III.31 Metodología de implementación de un algoritmo en FPGA.

III.2.1 Modelado e Identificación Del Sistema

En esta sección se abordaran diferentes métodos de identificación teóricos y prácticos; primeramente se identificara el sistema de manera analítica describiendo las ecuaciones diferenciales del modelo, utilizando una segunda aproximación se utilizaran un método de identificación experimental aplicando una entrada escalón y midiendo su respuesta, se comparan

resultados, el abordar diferentes métodos ayuda a tener herramientas poder realizar la identificación del sistema ya sea que se conozcan las propiedades físicas del motor (que se cuente con un hoja de datos del fabricante) o no.

(i) Sistema Físico Y Su Modelado Matemático

Para el desarrollo del controlador de movimiento nos enfocaremos en dos tipos de actuadores para el control de movimiento lo que son los motores de DC de imanes permanentes con escobillas y BLDC, a continuación describen las ecuaciones diferenciales que describen su dinámica, se obtiene su función de transferencia en el dominio de Laplace y su equivalente discreto, para su modelado y análisis en MATLAB/Simulink. En la figura III.32 se muestra el diagrama eléctrico de la armadura y de las partes móviles (rotor).

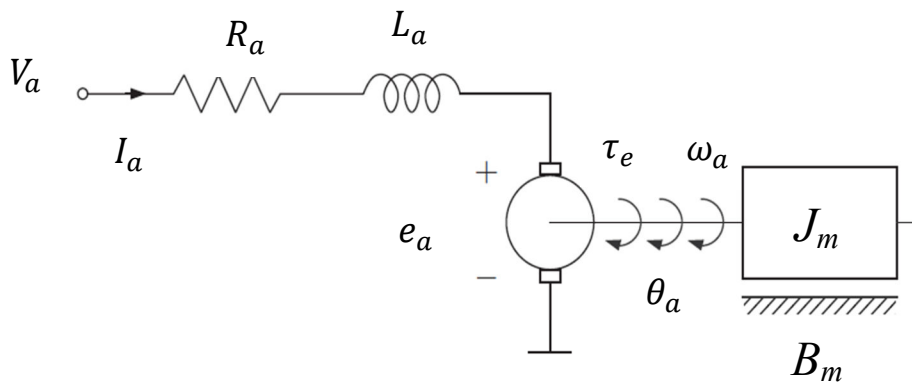


Figura III.32 Modelo del motor de imanes permanentes de DC.

En donde:

V_a – Voltaje en la armadura (V)

e_a – Voltaje inducido (V)

I_a – Corriente en la armadura (A)

τ_e – Torque electrico (Nm)

R_a – Resistencia electrica (Ohm)

ω_a – Velocidad del rotor (rad/s)

L_a – Inductancia electrica (H)

θ_a – Posición del rotor (rad)

(ii) Características Eléctricas

Las ecuaciones diferenciales para el circuito equivalente pueden ser encontradas utilizando las leyes de Kirchhoff, en donde tenemos que:

$$V_a - V_{R_a} - V_{L_a} - e_a = 0 \quad (42)$$

En donde:

$$V_{R_a} = i_a R_a \quad (43)$$

$$V_{L_a} = L_a \frac{d i_a}{dt} \quad (44)$$

$$e_a = k_e \omega_a \quad (45)$$

En donde k_e es la constante de la fuerza contra electromotriz y ω_a es la velocidad rotacional de la armadura, ahora sustituyendo (43), (44), (45) en (42) y despejando $\frac{d i_a}{dt}$, obtenemos la ecuación diferencial de la corriente en la armadura.

$$\frac{d i_a}{dt} = -\frac{R_a}{L_a} i_a - \frac{k_e}{L_a} \omega_a + \frac{V_a}{L_a} \quad (46)$$

(iii) Características Mecánicas

Realizando el balance de energía siguiendo la segunda ley de Newton, donde la suma de las fuerzas o torques debe ser igual a cero, por lo tanto.

$$T_e - T_{\dot{\omega}} - T_{\omega} - T_L = 0 \quad (47)$$

Donde T_e es el torque electromagnético, $T_{\dot{\omega}}$ es el torque generado debido a la aceleración en el rotor, T_{ω} es el torque generado por la velocidad en el rotor y T_L es el torque ejercido por la carga, en donde se tiene que:

$$T_e = k_T i_a \quad (48)$$

Donde, T_e es directamente proporcional a la corriente en la armadura y una constante de torque k_T , además:

$$T_{\dot{\omega}} = J_m \frac{d \omega_a}{dt} \quad (49)$$

Donde J_m es la inercia del motor, que equivale a una carga mecánica en el rotor. Y el torque asociado a la velocidad es escrito como:

$$T_{\omega} = B_m \omega_a \quad (50)$$

Donde B_m es el amortiguamiento o fricción asociado al sistema rotacional mecánico, ahora sustituyendo (48), (49) y (50) en (47) y despejando $\frac{d \omega_a}{dt}$, tenemos la ecuación diferencial de la velocidad de la armadura.

$$\frac{d \omega_a}{dt} = \frac{k_t}{J_m} i_a - \frac{B_m}{J_m} \omega_a - \frac{T_L}{J_m} \quad (51)$$

Aplicando la transformada de la Laplace, considerando condiciones iniciales iguales a cero, en (46) y (51), tenemos:

$$sI_a(s) = -\frac{R_a}{L_a} I_a(s) - \frac{K_e}{L_a} \Omega_a(s) + \frac{V_a(s)}{L_a} \quad (52)$$

$$s\Omega_a(s) = \frac{K_t}{J_m} I_a(s) - \frac{B_m}{J_m} \Omega_a(s) - \frac{T_L(s)}{J_m} \quad (53)$$

Las ecuaciones (52) y (53) pueden ser fácilmente representadas en un diagrama a bloques descritos por las siguientes ecuaciones:

$$I_a(s) = \frac{-K_e \Omega_a(s) + V_a(s)}{sL_a + R_a} \quad (54)$$

$$\Omega_a(s) = \frac{K_t I_a(s) - T_L(s)}{sJ_m + B_m} \quad (55)$$

En la Figura III.33 se muestra el diagrama a bloques que representa las ecuaciones (54) y (55), además se agregó un integrador a la salida de la velocidad para así poder conocer su posición, y tener un modelo completo de las variables de proceso en el control de movimiento que son corriente, velocidad y posición.

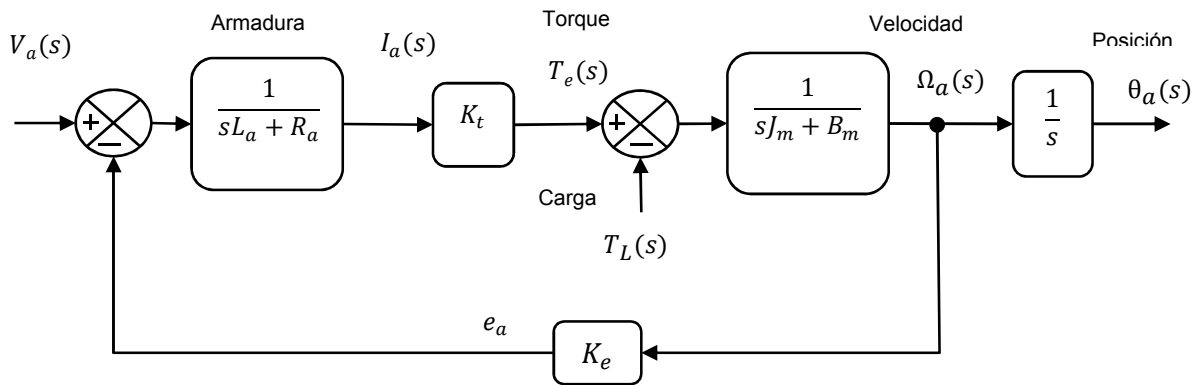


Figura III.33 Diagrama a bloques un motor de DC.

De este modelo, con $T_L(s) = 0$ y sin considerar el integrador a la salida, se tiene la siguiente función de transferencia entre el voltaje de entrada y la velocidad salida.

$$\frac{\Omega_a}{V_a} = \frac{\frac{K_t}{J_m L_a}}{s^2 + \left(\frac{B_m}{J_m} + \frac{R_a}{L_a}\right)s + \left(\frac{K_t K_e + B_m R_a}{J_m L_a}\right)} \quad (56)$$

Para propósitos de modelado el sistema planteado es también valido para motores de DC sin escobillas (BLDC), debido a que comparten las mismas características (Mentz, 2002), el modelo planteado nos permite tener acceso a todas las variables involucradas en el motor, pero esta aproximación puede ser aun reducida si despreciamos la fricción, en general en la ecuaciones que describen los servomotores (AC o DC) podemos encontrar dos constantes de tiempo, una es la eléctrica y la otra la mecánica (George, 2011), estas constantes de tiempo nos ayudan a representar el modelo de motor, normalmente estas son proporcionadas por el fabricante pero estas constantes son sin tomar en cuenta la carga en el rotor, de tal manera que considerando la ecuación (48), tenemos que :

$$T_e(s) = K_T I_a = J_T \alpha = J_T \Omega_a s \quad (57)$$

En donde J_T es la inercia total (inercia del motor más la inercia de la carga), y α es la aceleración (rad/s^2), despejando i_a de (57) y sustituyendo (52), tenemos:

$$I_a = \frac{J_T \Omega_a s}{K_T} \quad (58)$$

$$s \left(\frac{J_T \Omega_a s}{K_T} \right) = -\frac{R_a}{L_a} \left(\frac{J_T \omega_a s}{K_T} \right) - \frac{K_e}{L_a} \Omega_a + \frac{V_a}{L_a} \quad (59)$$

$$\left(\frac{J_T \Omega_a s}{K_T} \right) \left(s + \frac{R_a}{L_a} \right) = \frac{1}{L_a} (-K_e \Omega_a + V_a) \quad (60)$$

$$R_a \left(\frac{J_T \Omega_a s}{K_T} \right) \left(\frac{L_a}{R_a} s + 1 \right) = (-K_e \Omega_a + V_a) \quad (61)$$

A partir de (61) se encuentra la función de transferencia entre $\frac{\Omega_a}{V_a}$.

$$\frac{\Omega_a}{V_a} = \frac{K_T}{R_a J_T s \left(\frac{L_a}{R_a} s + 1 \right) + K_e K_T} \quad (62)$$

Dividiendo (62) entre $K_v K_e$, tenemos:

$$\frac{\Omega_a(s)}{V_a(s)} = \frac{\frac{1}{K_e}}{\left(\frac{R_a J_T}{K_e K_T} \right) \left(\frac{L_a}{R_a} \right) s^2 + \left(\frac{R_a J_T}{K_e K_T} \right) s + 1} \quad (63)$$

De la ecuación (63), tenemos que la constante de tiempo mecánica es igual a:

$$t_m = \frac{R_a J_T}{K_e K_T} \quad (64)$$

En donde tenemos que J_T es la suma de las inercias, de igual forma R_a es la suma de las resistencias en el circuito por lo que podemos decir que:

$$t_m = \frac{\sum R_a J_T}{K_v k_T} \quad (65)$$

De igual la manera la constante de tiempo eléctrica:

$$t_e = \frac{L_a}{\sum R_a} \quad (66)$$

Por lo tanto tenemos que la función de transferencia de un servomotor de DC de puede ser representada por:

$$\frac{\Omega_a(s)}{V_a(s)} = \frac{\frac{1}{K_e}}{t_m t_e s^2 + t_m s + 1} \quad (67)$$

Las constantes, mecánica y eléctrica, para un motor BLDC, tienen la misma estructura básica que (65) y (66), pero con algunas modificaciones, para un motor BLDC con conexión estrella, el circuito es el siguiente:

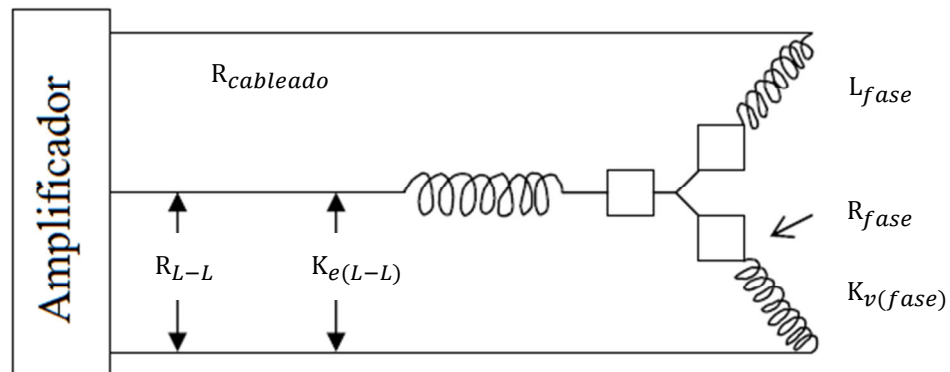


Figura III.34 Conexión estrella de un motor BLDC.

La constante de tiempo mecánica para el motor BLDC de la figura III.34, es:

$$t_m = \frac{\sum R_{fases} J_T}{K_{e(fase)} K_T} \quad (68)$$

En donde:

$$K_{e(fase)} = \frac{K_{e(L-L)}}{\sqrt{3}} \quad (69)$$

$$\sum R_{M(L-L)} = 1.35 R_{M(L-L)} \quad (70)$$

$$\sum R_{M(fase)} = 0.5 * \sum R_{M(L-L)} \quad (71)$$

La mayoría de las hojas de datos de los motores, proporcionan los valores entre línea-línea, como se mostró en las ecuaciones anteriores esos valores deben ser cambiados a sus valores por fase, por lo que podemos decir que la constante de tiempo mecánica puede ser escrita como:

$$t_m = \frac{\sqrt{3}}{2} \left(\frac{\sum R_{M(L-L)} J_T}{K_{e(L-L)} k_T} \right) \quad (72)$$

La constante de tiempo eléctrica puede ser descrita por:

$$t_e = \frac{L_{L-L}}{\sum R_{M(L-L)}} \quad (73)$$

De las ecuaciones (56) o (67) podemos describir el modelo la planta del motor DC y BLCD, estos modelos pueden ser reducidos a un sistema de primer orden tomando en cuenta la rapidez del sistema eléctrico en donde $t_e \ll t_m$, por lo que puede ser despreciada, un desarrollo a detalle puede ser encontrado en (Ogata, 1998).

$$\frac{\Omega_a(s)}{V_a(s)} = \frac{\frac{K_t}{(K_t K_e + B_m R_a)}}{\left(\frac{J_m R_a}{(K_t K_e + B_m R_a)} \right) s + 1} \approx \frac{\frac{1}{K_e}}{t_m s + 1} \quad (74)$$

Ahora para determinar la relación torque-velocidad se utilizan las condiciones del equilibrio del sistema; es decir cuando al motor se le aplica un voltaje determinado y llega a una velocidad constante, partiendo de las ecuaciones (46) y (51), en el punto de equilibrio la derivada de la corriente y la velocidad son cero por lo que podemos escribir que las ecuaciones de equilibrio son:

$$V_a = R_a i_a + K_e \omega_a \quad (75)$$

$$T_L = K_t i_a - B_m \omega_a \quad (76)$$

Despejando i_a de (75) y (76) e igualando tenemos:

$$V_a = \frac{R_a T_L}{k_t} + \frac{R_a \omega_a B_m}{k_t} + k_e \omega_a \quad (77)$$

En donde despejamos las ecuaciones para el equilibrio de velocidad y torque respectivamente.

$$\omega_a = \frac{V_a K_t + R_a T_L}{R_a B_m + K_t K_e} \quad (78)$$

$$T_L = \frac{V_a K_t}{R_a} - \omega_a \left(B_m + \frac{K_t K_e}{R_a} \right) \quad (79)$$

De la ecuación (79) se puede determinar el torque máximo a rotor bloqueado, donde $\omega_a = 0$, este es conocido como “stall torque”.

$$T_s = \frac{V_a K_t}{R_a} \quad (80)$$

Por lo tanto la corriente máxima o rotor bloqueado esta dada por:

$$I_s = \frac{T_s}{K_t} = \frac{V_a}{R_a} \quad (81)$$

La máxima velocidad es alcanzada cuando no se le aplica ninguna carga $T_L = 0$, de la ecuación (78) tenemos que:

$$\omega_{a_max} = \frac{T_s}{B_m + \frac{K_t K_e}{R_a}} \quad (82)$$

La potencia mecánica P_m entregada al motor esta dada por el torque aplicado y su velocidad angular, de la ecuación (79), tenemos:

$$P_m = T_L \omega_a = T_s \omega_a - \omega_a^2 \left(B_m + \frac{K_t K_e}{R_a} \right) \quad (83)$$

La eficiencia del motor esta dada por la relación entre la potencia eléctrica suministrada y la potencia mecánica producida, para obtener la eficiencia en función del equilibrio, se expresa la corriente y el torque en términos de la velocidad, utilizando la ecuación (75) y (79) tenemos:

$$\eta = \frac{P_m}{V_a I_a} = \frac{T_L \omega_a}{V_a I_a} = \frac{T_s \omega_a R_a - \omega_a^2 (B_m R_a + K_t K_e)}{V_a^2 + K_e \omega_a} \quad (84)$$

Encontradas las principales ecuaciones que describen el comportamiento de un motor de DC; se selecciono un motor Maxon-148867 de DC y el motor de BLDC de Galil-BLM-N23-50-1000-B, en Tabla.III.4 se muestran sus principales características.

Tabla III.4 Características principales de los motores modelados.

Parámetros	Variable	Motor de DC	Motor BLDC
Resistencia (Ω)	R_a	0.32	1.2
Inductancia (H)	L_a	$82.3 \cdot 10^{-6}$	0.0026
Constante de torque (Nm/A)	K_t	0.0302	0.08
Constante de voltaje (V/(rad/s))	K_e	0.0302	0.08
Inercia del rotor (Kg m^2)	J_m	$13.8 \cdot 10^{-6}$	$25 \cdot 10^{-6}$
Constante de amortiguamiento ($\text{Kg m}^2/\text{s}$)	B_m	$18.89 \cdot 10^{-6}$	$222.22 \cdot 10^{-6}$
Constante de tiempo mecánica (ms)	T_m	4.81	4.5
Voltaje nominal (V)	VDD	24	48
Corriente nominal (A)	I_{max}	5.77	4.6
Velocidad nominal sin carga (RPM)	ω_{a_max}	7580	5000
Corriente de arranque (A)	I_s	75.7	10.4
Resolución Encoder (Pulsos/rev)	P_{PR}	500	1000

En ocasiones algunos parámetros del motor como la constante de tiempo eléctrica o la constante de amortiguamiento, no son incluidos en la hoja de datos pero pueden ser calculados por la ecuación (64) y (85) respectivamente.

$$B_m = \frac{J_m}{tm} - \frac{K_t * K_e}{R_a} \quad (85)$$

Una vez convertido las constantes al Sistema Internacional de unidades (SI), se introducen a un modelo creado en MATLAB/Simulink, primeramente se realizara todo el análisis y modelado utilizando el motor de DC y posteriormente en la sección de resultados se aplicara la misma metodología para el control del motor BLDC, a continuación en la Figura III.35 se observa el modelo completo del motor de DC.

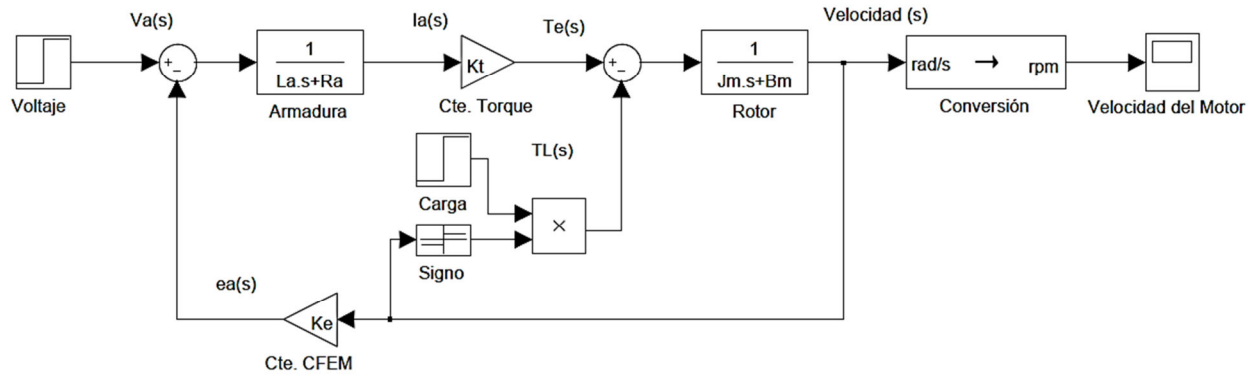


Figura III.35. Modelo del motor de DC en Simulink.

Para validar el modelo se aplica una entrada escalón con el voltaje nominal del motor y sin carga ($T_L = 0$), en la Figura III.36 se muestra la respuesta al escalón.

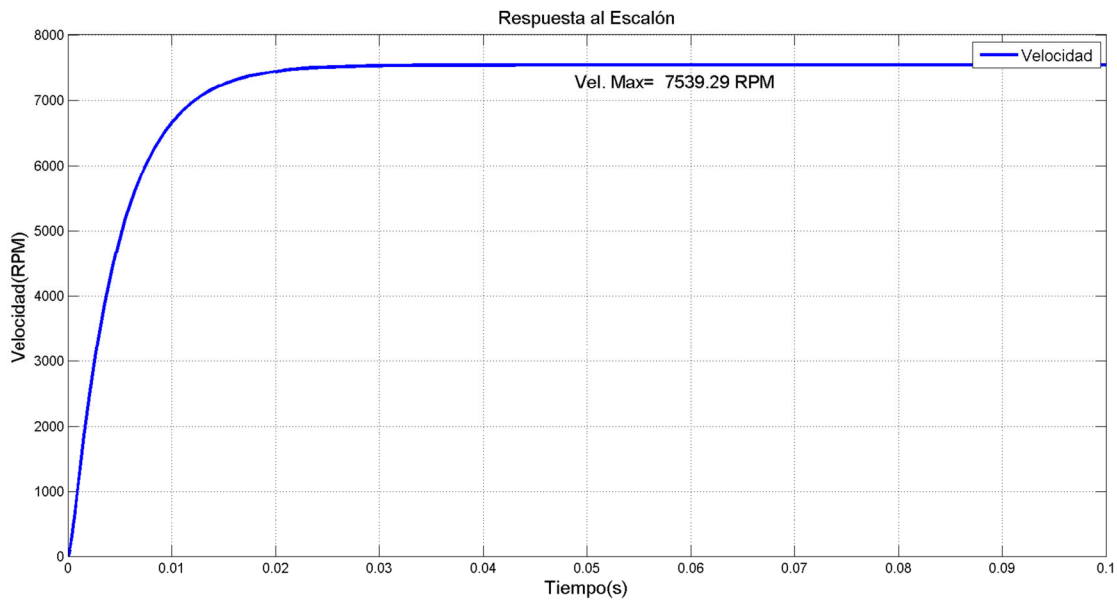


Figura III.36. Respuesta del motor de DC a una entrada escalón.

En la Figura III.36 podemos observar que se tiene un valor máximo de 7539.29 RPM que es la velocidad nominal del motor sin carga; con respecto al valor nominal de la Tabla III.4 tenemos un error del 0.5%, de igual manera en la Figura III.22 podemos observar el comportamiento de la

corriente en el arranque del motor y ver como el pico de corriente máximo en la simulación de 66.27A que es cercano al valor de la Tabla III.3.

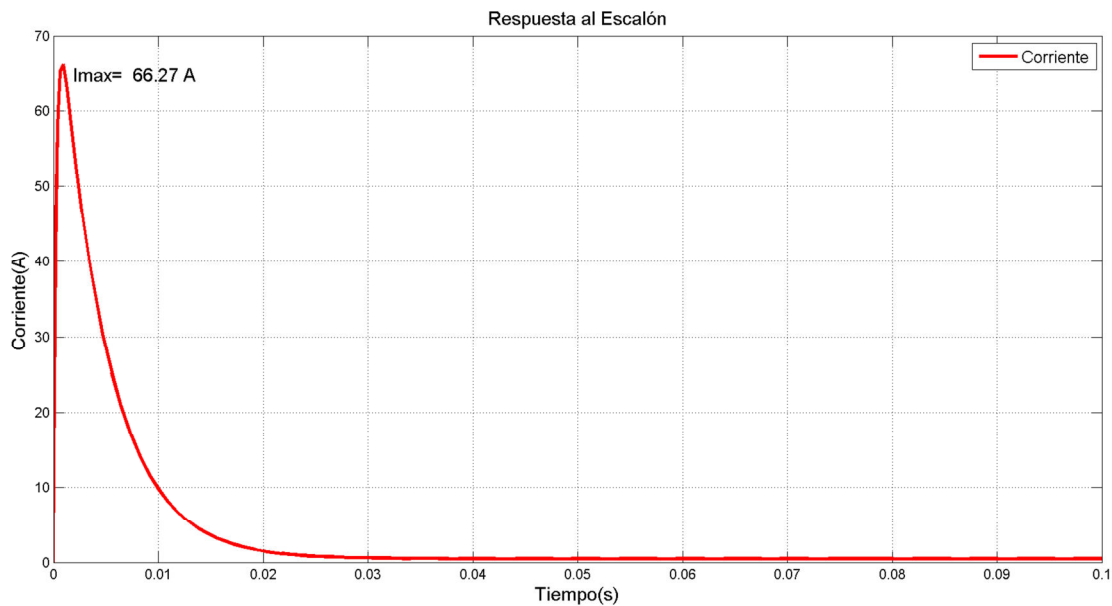


Figura III.37. Respuesta de la corriente al aplicar el voltaje nominal.

Utilizando las ecuaciones (80-83), se obtiene que la función del torque del motor respecto a su velocidad así como la relación de la potencia y la velocidad de igual forma en la eficiencia de motor respecto a su velocidad, en la Figura III.38 se muestra la relación de la velocidad - torque, potencia y eficiencia.

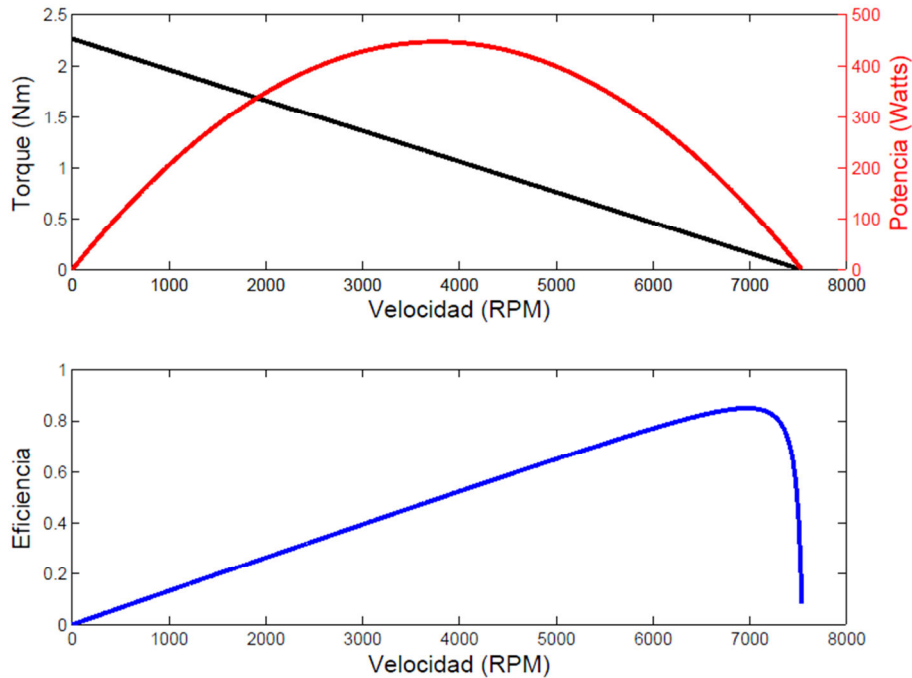


Figura III.38. Eficiencia del motor de DC y su relacion Torque-Potencia-Velocidad.

Como se mencionó en el capítulo anterior los servoamplificadores son ampliamente utilizados en la industria sobre todo el manejo de dispositivos electromecánicos, ya que simplifican el orden y complejidad del modelo al incluir el control de torque o lazo de corriente, de tal manera que el sistema electromecánico se reduce a un sistema meramente mecánico, en el modelo de la Figura III.35 la parte eléctrica puede ser reducida y vista como una ganancia (Parker, 2011); Gracias a los servoamplificadores es que se puede contar con un controlador estándar, el cual con una señal de $\pm 10V$ puede integrarse con diferentes servoamplificadores de diversos fabricantes (Maxon Motor, Copley Controls, Advanced Motion Controls) y para diferentes tecnologías (Motores de DC o BLDC) inclusive con motores síncronos de DC o PMSM, esto representa una gran ventaja en control de movimiento, ya que el mismo controlador puede ser aplicado en diferentes motores evitando así su obsolescencia y reduciendo costos, es por ello que el controlador propuesto se diseñara bajo estos criterios. Tomando en cuenta esta consideración el modelo del motor con el servoamplificador normalmente es reducido para simplificar el modelo. De tal manera que la función de transferencia $G_{Ia}(s)$ del control de corriente puede ser vista como una ganancia K_{i_a} (V/A).

$$G_{Ia}(s) = K_{i_a} \tag{86}$$

Y la función de transferencia del servomotor con un servoamplificador en modo corriente, quedaría de la siguiente forma:

$$\frac{\theta(s)}{I_a(s)} \approx \left(\frac{K_{i_a} K_e}{J_m s + B_m} \right) \left(\frac{1}{s} \right) = \frac{K_f K_{i_a} K_e}{J_m s^2 + B_m s} \tag{87}$$

$$K_f = \frac{P_{PR}}{2\pi} \tag{88}$$

En donde θ representa la posición de la flecha del motor, y se introduce K_f que es la ganancia de retroalimentación (Encoder), en donde para P_{PR} se toman en cuenta el número de pulsos totales, esto es después de la decodificación (1X, 2X y 4X) de las señales de cuadratura. El modelo reducido de la ecuación (87) muestra en la Figura III.39.

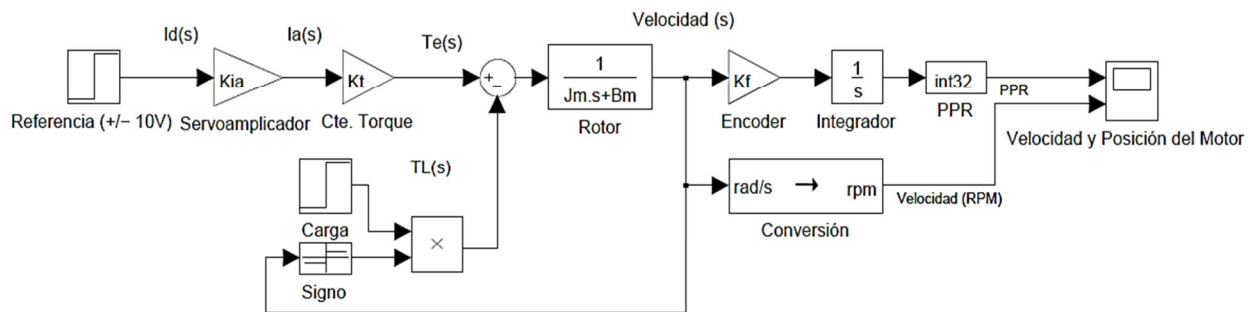


Figura III.39. Modelo reducido del motor de DC incluyendo el lazo de torque.

Para el modelado y características del servoamplificador se utilizo el modelo B15A8 de Advanced Motion Controls, que como particularidad que posee es que puede trabajar con ambos tipos de motores DC y BLDC (A-M-C, 2011), en Tabla III.5 se muestran sus principales características.

Tabla III.5. Características principales del servoamplificador B15A8

Voltaje de alimentación (V)	20-80
Máxima corriente de salida pico (A)	15

Máxima corriente de salida continua (I_{max})	7.5
Frecuencia de conmutación (KHz)	33
Inductancia mínima de la carga (H)	$200 \cdot 10^{-6}$

Normalmente el servoamplificador se configura de tal manera que la relación es de uno a uno es decir, que la señal de control o referencia V_{ref} (+/- 10V) sea directamente proporcional a la corriente, esto para simplificar aun mas el modelo.

$$K_{i_a} = \frac{I_{max}}{V_{ref}} = 1 \quad (89)$$

En la Figura III.40 y Figura III.41 se aprecia la respuesta del sistema de la ecuación (87), con referencia de $0.5V = 0.5A$, en la Figura III.40 se toma en cuenta la constante de amortiguamiento B_m , y en la Figura III.41 la constante B_m se descarta, esta simplificación es utilizada por diferentes autores (Osornio, 2007), (Wang et al., 2007), (Yokote, 1990).

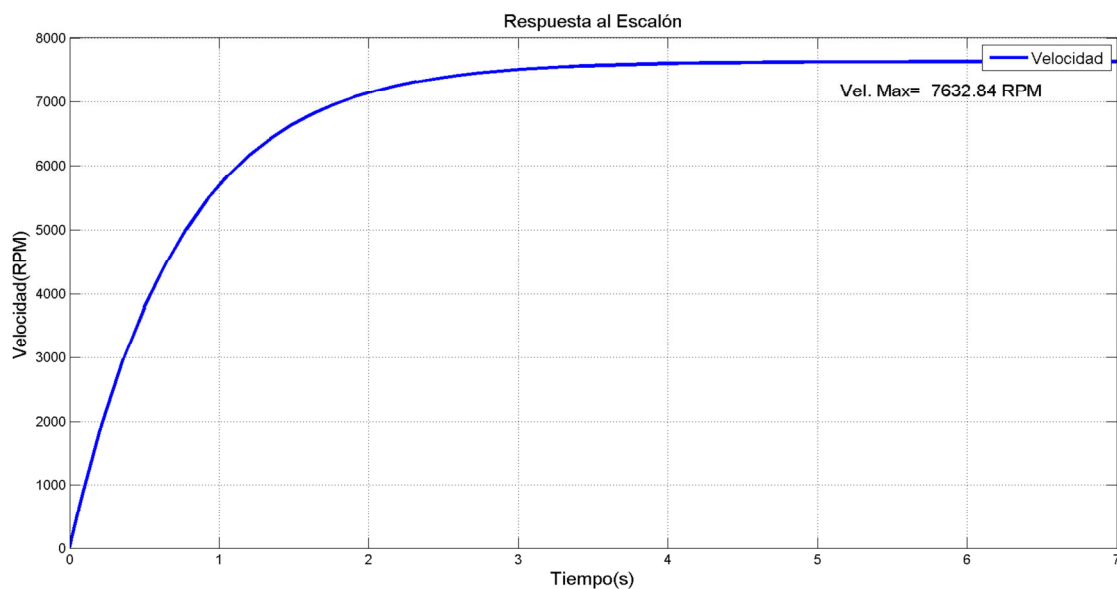


Figura III.40. Modelo reducido del motor de DC incluyendo la constante de amortiguamiento.

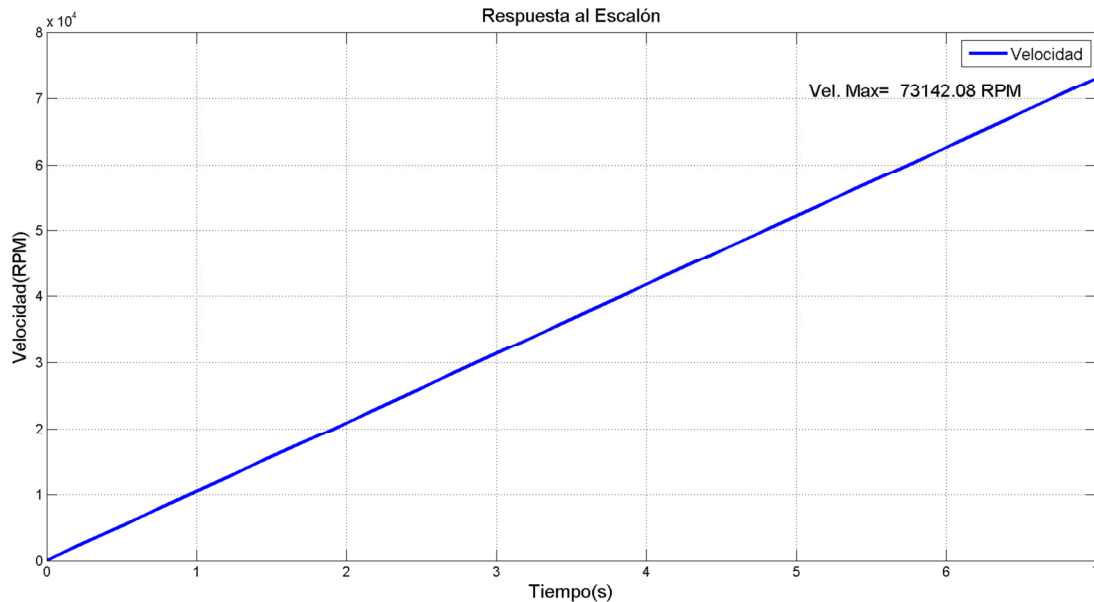


Figura III.41. Modelo reducido del motor de DC descartando la constante de amortiguamiento.

El modelo de la ecuación (87) presenta una respuesta acorde a la velocidad nominal del motor, sin embargo cuando se descarta el factor de amortiguamiento el sistema no tiende a converger a un valor final, debido a este comportamiento, se opta por utilizar el modelo del motor dado por ecuaciones (54-55) y realizar el modelado del servoamplificador esto para tener una simulación confiable.

(iv) Diseño y modelado del controlador

1) Control de corriente

Primeramente se redujo el modelo de la Figura III.35 a un sub-bloque llamado “Servomotor” donde se incluyeron las ganancias de la función de transferencia de la ecuación (87). Partiendo de la información del servoamplificador se elaboro un modelo simplificado del control de corriente, las ecuaciones siguientes se infirieron del hoja de datos del mismo y de la medición experimental de la respuesta al escalón; primeramente el servoamplificador cuenta con un controlador tipo PI analógico, para su representación en tiempo continuo se utilizan los valores de sus componentes pasivos y se calcularon sus ganancias utilizando las siguientes expresiones:

$$u_{I_a}(t) = K_p e(t) + K_i \int_0^t e(t) dt \rightarrow G_I(s) = K_p + \frac{K_i}{s} = K_p \left(1 + \frac{1}{T_i}\right) \quad (90)$$

$$K_p = \frac{R_2}{R_1} \quad (91)$$

$$T_i = R_2 C \quad (92)$$

$$K_i = \frac{K_p}{T_i} \quad (93)$$

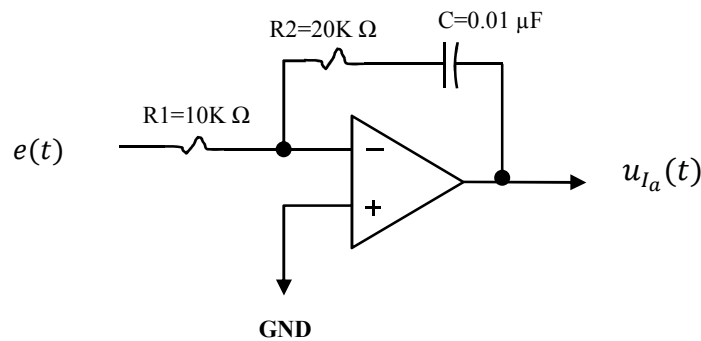


Figura III.42. Control PI analógico con valores del servoamplificador B15A8N.

El PWM puede ser modelado como un retardo por medio de una aproximación Pade de primer orden, o como un retardo en el tiempo (Buso et al., 2006), es decir:

$$PWM(s) = e^{-s \frac{1}{F_{PWM}}} \cong \frac{1 - s \frac{1}{4 F_{PWM}}}{1 + s \frac{1}{4 F_{PWM}}} \quad (94)$$

En donde F_{PWM} es la frecuencia de conmutación, otra parte importante del servoamplificador es el filtrado de la corriente, este normalmente se hace por medio de filtros pasivos por que son la solución mas barata, sin embargo los filtros activos ofrecen mejores resultados, para la selección y modelado del LPF no se recurrió en algún proceso de ingeniería inversa, éste se modelo considerando las recomendaciones de diseño planteadas en (Adams, 2003), donde se afirma que la frecuencia de fundamental la corriente en los drivers para motores esta normalmente en el rango de los 8-10KHz y una solución muy común para su

acondicionamiento es utilizar un filtro RC de dos polos, como se muestra en la Figura III.43, la primera etapa de diseño para tener una frecuencia de corte (F_c) de 10KHz y la segunda etapa se utiliza una frecuencia mucho mayor en este caso para atenuar el efecto del PWM, que tiene una frecuencia de 33KHz, para cuestiones practicas $R2 \gg R1$ para no sobrecargar la primer sección del filtro.

$$F_c = \frac{1}{2\pi RC} \quad (95)$$

$$F_{RC}(s) = \frac{1}{\left(\frac{1}{2\pi F_c}\right)^2 s^2 + 1} \quad (96)$$

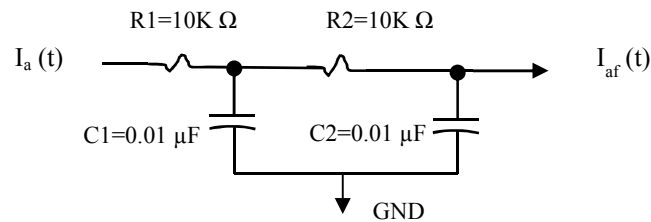


Figura III.43. Filtro RC pasa bajas de segundo orden para el filtrado de la corriente.

Se modelaron las no-linealidades o saturaciones de los componentes del servoamplificador, es decir los niveles máximos de voltaje de referencia (97), máxima corriente (98), los valores máximos a la salida del control PI (99), los niveles de referencia para la señal PWM (100), la ganancia del voltaje aplicado al motor (101), la ganancia de retroalimentación de la corriente (102) y finalmente la ganancia del voltaje de referencia (103).

$$sat(V_{ref}) = \begin{cases} 10V & si V_{ref} > 10V \\ V_{ref} & si |V_{ref}| \leq 10V \\ -10V & si V_{ref} < -10V \end{cases} \quad (97)$$

$$sat(I_{max}) = \begin{cases} 7.5A & si I_{max} > 7.5A \\ I_{max} & si |I_{max}| \leq 7.5A \\ -7.5A & si I_{max} < -7.5A \end{cases} \quad (98)$$

$$sat(u_{I_a}) = \begin{cases} 15V & si u_{I_a} > 15V \\ u_{I_a} & si |u_{I_a}| \leq 15V \\ -15V & si u_{I_a} < -15V \end{cases} \quad (99)$$

$$sat(PWM_{ref}) = \begin{cases} 5V & si PWM_{ref} > 5V \\ PWM_{ref} & si |PWM_{ref}| \leq 5V \\ -5V & si PWM_{ref} < -5V \end{cases} \quad (100)$$

$$K_{vdd} = \frac{VDD}{5V} \quad (101)$$

$$K_{if} = \frac{1V}{2V} \quad (102)$$

$$K_{ia} = 1 \quad (103)$$

En la Figura III.44 se muestra el modelo del sub-bloque del servoamplificador con cada uno de sus componentes y ganancias, en donde u_{va} representa el voltaje aplicado al motor, en la Figura III.45 se muestra el modelo del servomotor de DC con el servoamplificador, estos modelos son la base para el diseño del control de velocidad y de posición.

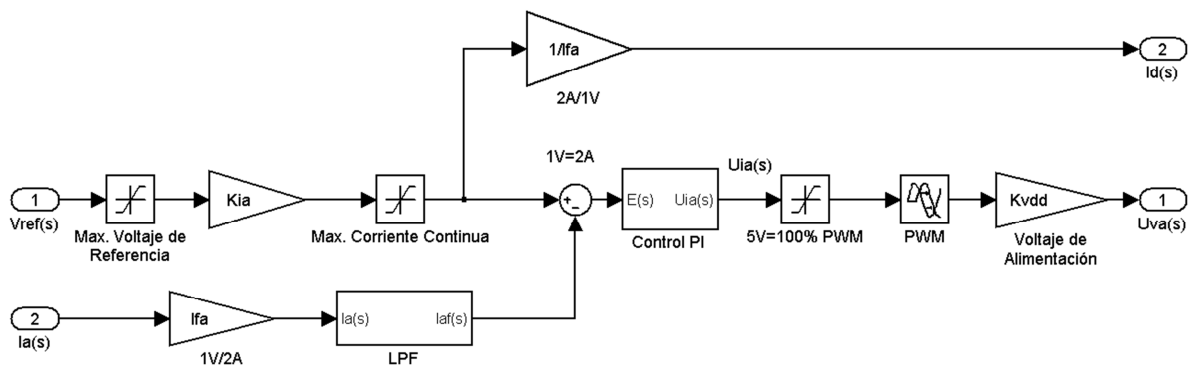


Figura III.44. Modelo del servoamplificador B15A8.

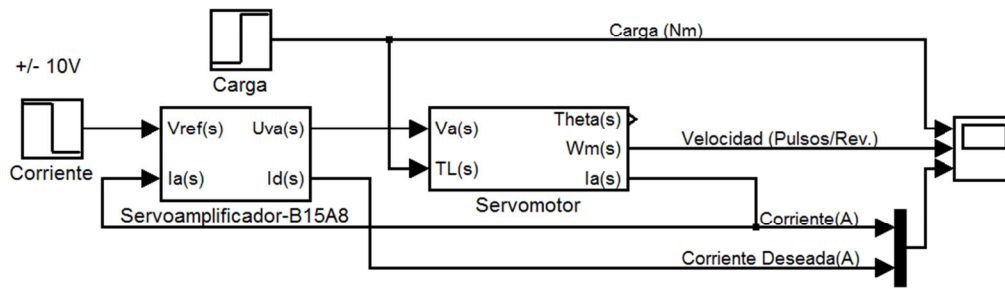


Figura III.45. Modelo del servomotor y el servoamplificador.

2) Control de velocidad

Para el controlador de movimiento se utilizó la configuración propuesta por (Malek et al., 2011) mostrado en la Figura III.25, debido a su rendimiento y flexibilidad, primeramente se inició con el diseño del controlador PI de velocidad el cual está dado por las siguientes ecuaciones:

$$u_{\omega}(t) = K_{p\omega}e(t) + K_{i\omega} \int_0^t e(t)dt \rightarrow R_{\omega}(s) = K_{p\omega} + \frac{K_{i\omega}}{s} \quad (104)$$

$$F_{\omega}(s) = e^{-s\left(\frac{T_{s\omega}}{2}\right)} \quad (105)$$

En donde la función $F_{\omega}(s)$ representa un retardo en tiempo continuo y $T_{s\omega}$ es el tiempo de muestreo. La ecuación (104) representa la función de transferencia del controlador de velocidad en lazo abierto y K_{DAC} la ganancia del DAC en donde V_{max} y V_{min} representan el voltaje máximo y mínimo de salida y R la resolución en bits, las unidades del controlador de velocidad serán pulsos/s.

$$L_{\omega}(s) = F_{\omega}(s)R_{\omega}(s)K_{DAC} \quad (106)$$

$$K_{DAC} = \frac{V_{max} - V_{min}}{2^R} \quad (107)$$

El controlador PI de velocidad incluye una función de saturación debido a la resolución del DAC la cual esta dada por siguiente expresión:

$$sat(u_\omega) = \begin{cases} 2^{R-1} - 1 & \text{si } u_\omega > 2^{R-1} - 1 \\ u_\omega & \text{si } |u_\omega| \leq 2^{R-1} - 1 \\ -2^{(R-1)} & \text{si } u_\omega < -2^{(R-1)} \end{cases} \quad (108)$$

Para la retroalimentación de la velocidad se aplicó un LPF de primer orden, en donde $F_{c\omega}$ es la frecuencia de corte.

$$H_\omega(s) = \frac{1}{\left(\frac{1}{2\pi F_{c\omega}}\right)s + 1} \quad (109)$$

A continuación se muestra el diseño el sub-bloque del controlador PI de velocidad que muestra en la Figura III.46 y en la Figura III.47 se muestra el diagrama controlador de velocidad en lazo cerrado.

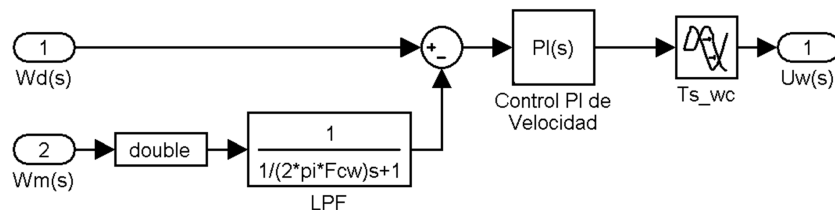


Figura III.46. Modelo del sub-bloque del controlador PI.

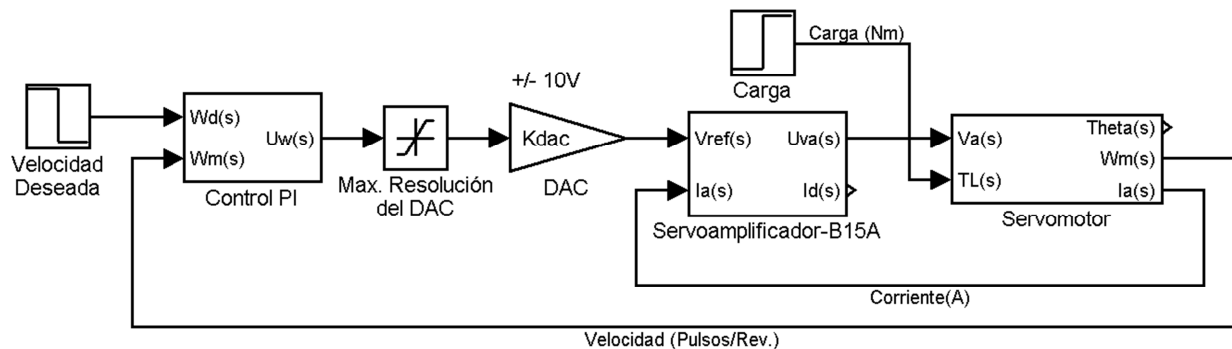


Figura III.47. Controlador PI de velocidad del servomotor.

3) Control de posición

Finalmente para tener todo el modelo completo del controlador de movimiento, se incluyo el controlador P de posición (110) y la introducción de las ganancias de pre alimentación, en un inicio la ganancias de pre alimentación son totalmente despreciadas debido a la ausencia de un perfil de velocidad, como se menciona en el capítulo anterior estas ganancias de pre alimentación ayudan a reducir el error en el seguimiento de trayectorias, a continuación en la Figura III.48 se muestra el controlador P de posición que esta dado por:

$$u_{\theta}(t) = K_{p\omega}e(t) \rightarrow R_{\theta}(s) = K_{p\theta} \quad (110)$$

$$F_{\theta}(s) = e^{-s\left(\frac{T_{s\theta}}{2}\right)} \quad (111)$$

$$L_{\theta}(s) = F_{\theta}(s)R_{\theta}(s) \quad (112)$$

La salida del controlador de posición esta limitada por la velocidad máxima (P_{max}) del servomotor que esta dada en pulsos/s y esta determinada por la velocidad máxima del motor en RPM y la resolución del encoder P_{PR} .

$$P_{max} = \frac{\omega_{a_max}P_{PR}}{60} \quad (113)$$

$$u_{\theta}(t) = \begin{cases} P_{max} & si \ u_{\theta} > P_{max} \\ u_{\theta} & si \ |u_{\theta}| \leq P_{max} \\ -P_{max} & si \ u_{\theta} < -P_{max} \end{cases} \quad (114)$$

En la Figura III.49 se muestra el controlador de movimiento modelado, para su simulación se utilizan los parámetros del motor de DC de Tabla III.4, para cumplir con los requerimientos de funcionamiento (inductancia mínima) del servoamplificador se añadieron en serie al motor dos inductancias con una resistencia e inductancia equivalente de 0.2Ω y $263\mu H$ respectivamente, se diseño un generador de perfiles de velocidad trapezoidal partiendo de las ecuaciones (32-25), el cual se muestra en la Figura III.48, uno de los objetivos de realizar el modelado en

MATLAB/Simulink son la herramientas que ofrecen para el diseño y sintonía de controladores el cual por medio de una linealización de los componentes nos permite ajustar de manera fácil y sencilla las ganancias del controlador ya sea seleccionando el margen de fase y frecuencia o por el tiempo de respuesta, anteriormente se menciono que una de la desventajas del control en cascada y su aplicación en la industria es su elaborado proceso de sintonía sin embargo con el modelo y método aquí presentado nos permite de manera rápida y sencilla realizar este proceso, sin embargo para complementar en su desarrollo en el Anexo 3 se muestra el desarrollo del proceso de sintonía utilizando el método de margen de fase.

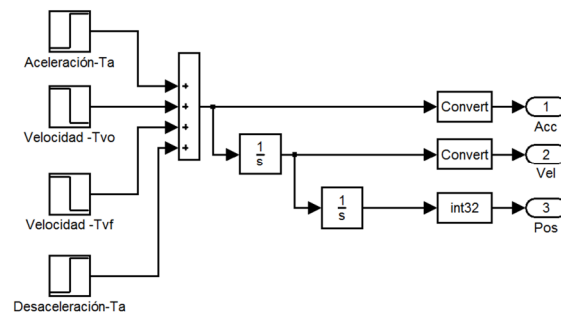


Figura III.48. Generador de perfiles trapezoidales.

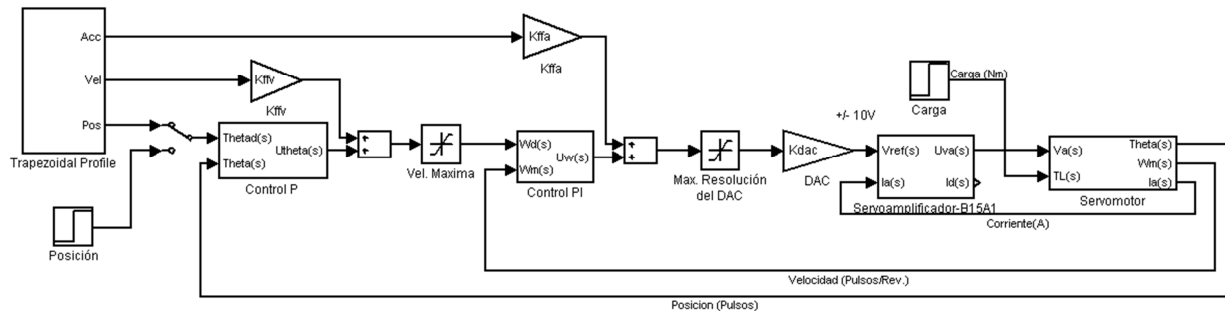


Figura III.49. Modelo del controlador de movimiento.

Las ganancias de pre-alimentación proporcionan la parte necesaria para el seguimiento de la trayectoria con el mínimo error, debido a que proporcionan una señal de referencia inmediata sobre la velocidad deseada y la corriente deseada, mientras que el controlador P-PI regula el error, para la sintonía de la ganancias $K_{ff\omega}$ y K_{ffE} se siguieron las recomendaciones y ecuaciones propuestas en (Dessaint et al., 1990) y (Ellis, 2004).

El periodo de muestro para el controlador de velocidad se selecciono tomando en cuenta el siguiente criterio (Fadali, 2009); para el sistema de primer orden de la ecuación (74), se tiene que la respuesta en frecuencia dada por:

$$H(j\omega) = \frac{K}{j\omega + \omega_b} \quad (115)$$

En donde K es la ganancia de DC y ω_b es el ancho de banda del sistema, la amplitud de la respuesta en frecuencia del sistema cae por debajo del nivel de DC por un factor de 10 cerca de $7\omega_b$, por lo que se considera que la frecuencia de muestreo puede ser seleccionada como:

$$\omega_s = k * \omega_b, \quad 35 \leq k \leq 70 \quad (116)$$

Utilizando un factor de $k = 60$ y considerando la constantes de tiempo mecánicas de Tabla III.4 tenemos que el tiempo de muestreo seleccionado es de $500\mu s$ y $F_{c\omega} = 200Hz$, primeramente se sintonizo el lazo de velocidad con un frecuencia de $160rad/s$ y un margen de fase de 70° con un sobrepaso del 12.1%, se sometió a un entrada escalón de $16666 PPR$ que equivale a $500 RPMs$ y con una perturbación de $0.015Nm$.

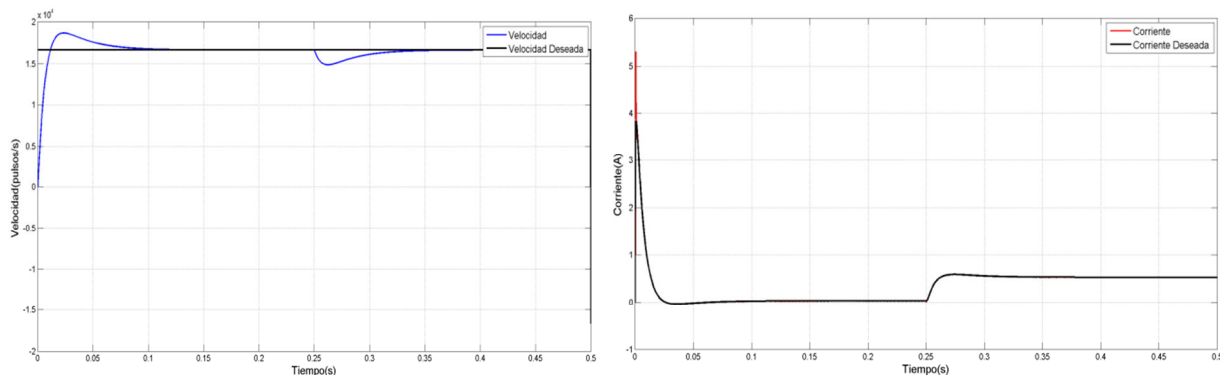


Figura III.50. Respuesta a una entrada escalón de 16666.66PPR.

A continuación se muestra la respuesta del controlador de posición en tiempo continuo a una entrada escalón y se muestra el seguimiento de la referencia de velocidad.

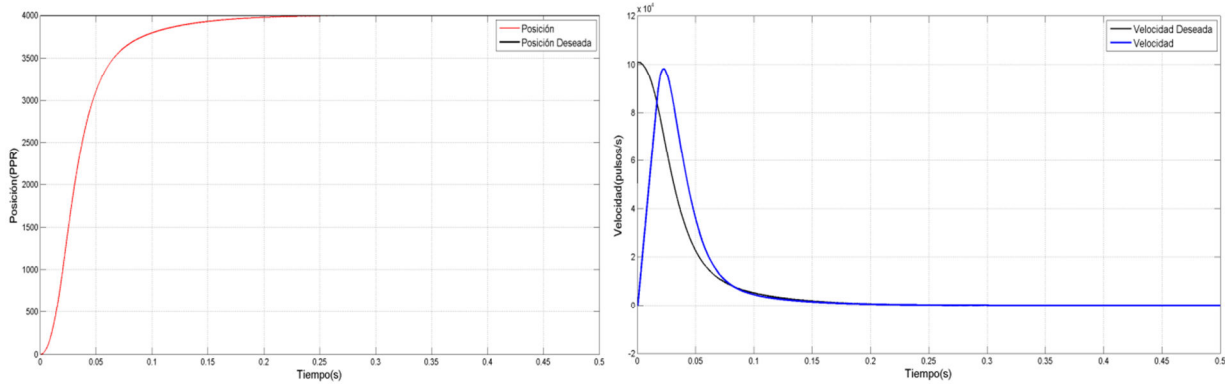


Figura III.51. Respuesta a una entrada escalón de 4000PPR.

Las Figuras III.52-54 muestran la respuesta utilizando un perfil de velocidad trapezoidal, primeramente sin realizar la pre-alimentación de la velocidad y la aceleración y posteriormente con las ganancias $K_{FF\omega}$ y $K_{FF\epsilon}$, se observó una clara mejora en el desempeño y disminución en error.

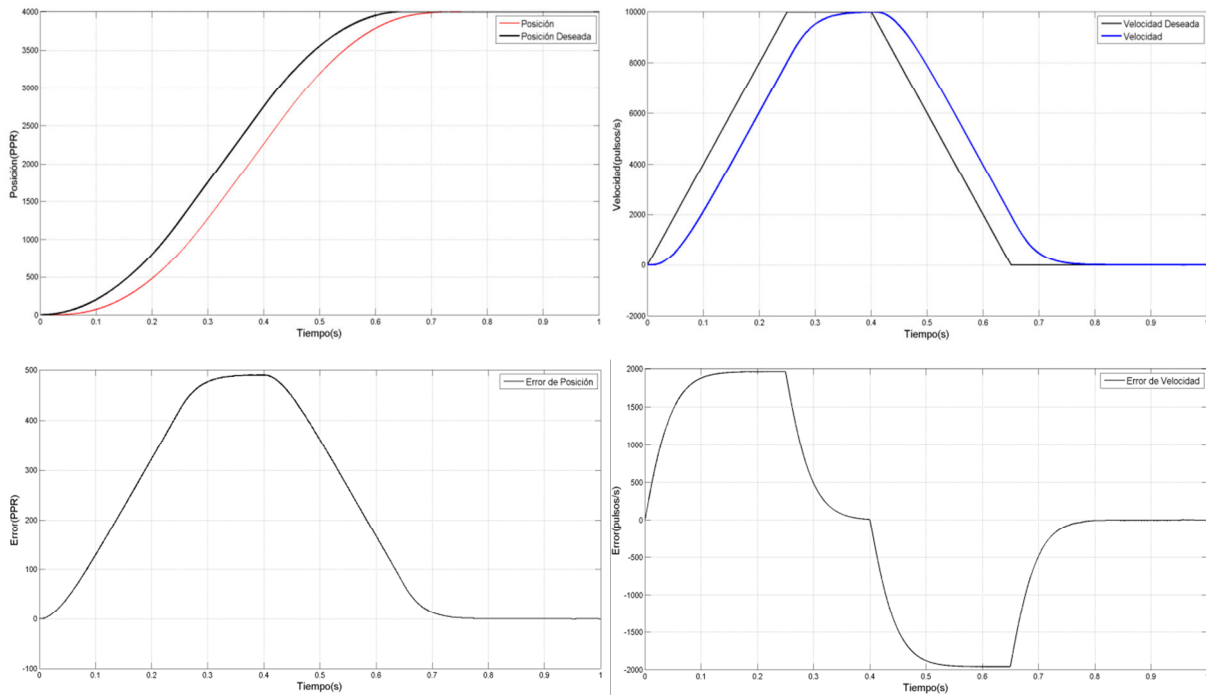


Figura III.52. Respuesta del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s² y una posición final de 4000 PPR y sin las ganancias feedforward.

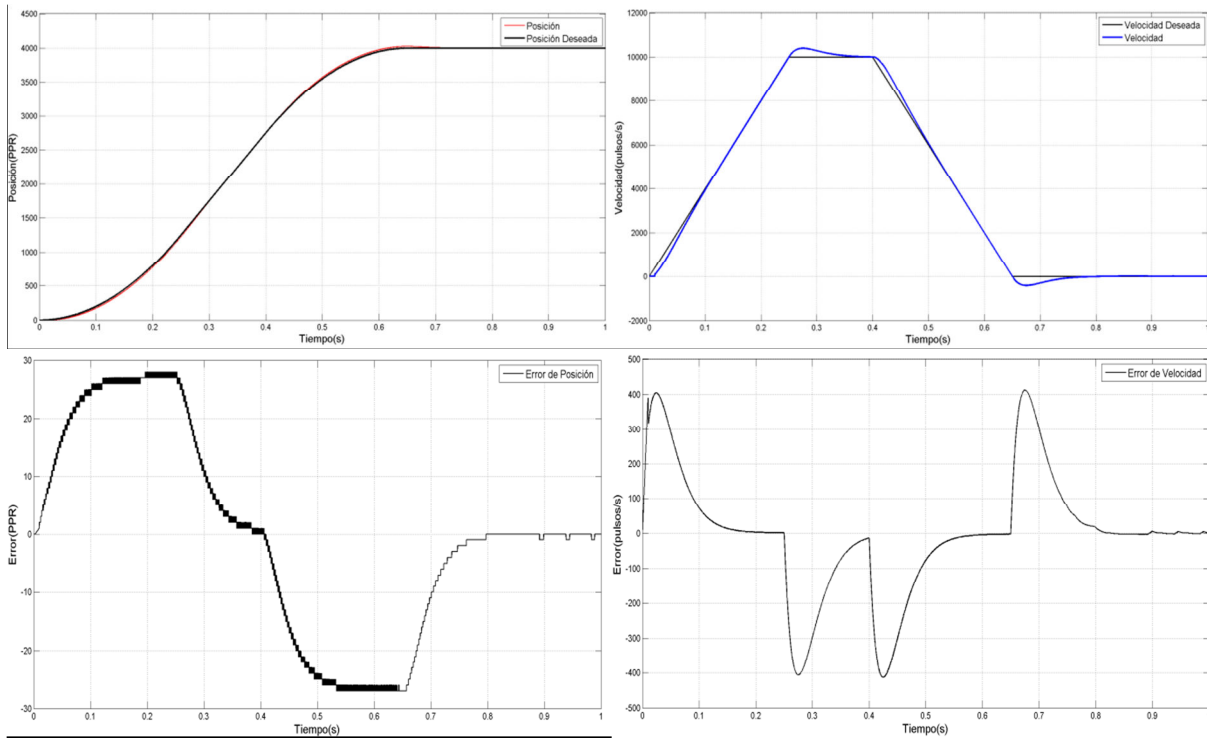
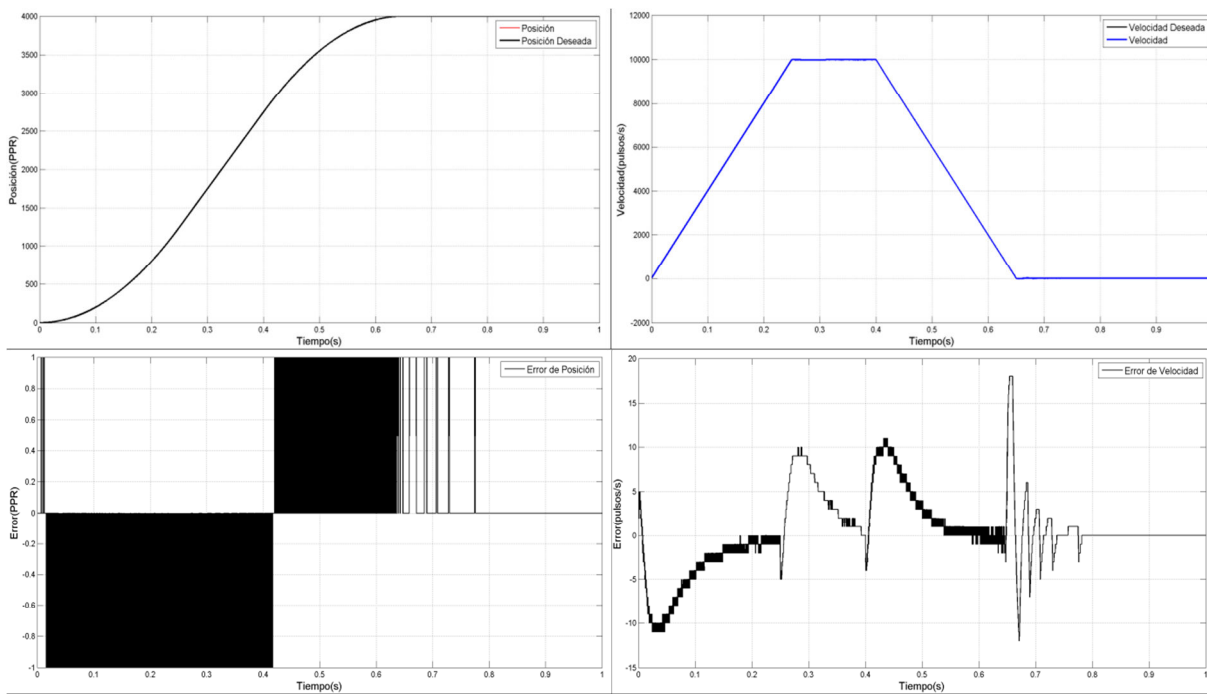


Figura III.53. Respuesta del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s² y una posición final de 4000 PPR y con la ganancia feedforward de velocidad.



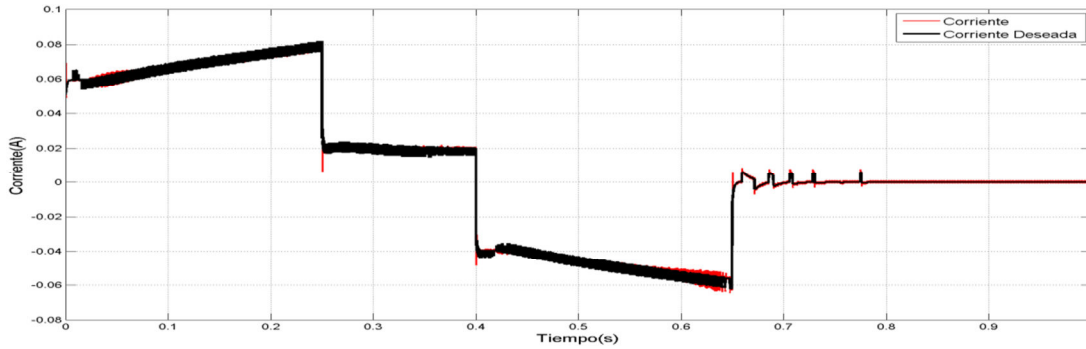


Figura III.54. Respuesta del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s² y una posición final de 4000 PPR y con la ganancias feedforward de velocidad y aceleración.

A continuación se muestra la respuesta del controlador PID convencional mostrado en la Figura III.55, el cual esta dado por la siguiente expresión:

$$u_{\theta p}(t) = K_p \theta_p + \frac{K_i \theta_p}{s} + K_d \frac{s}{s+1} \quad (117)$$

Se sintonizo con una frecuencia de cruce de 16rad/s y un margen de fase de 70°, se aplica una entrada escalón de 4000PPR, en donde el tiempo de respuesta es cercano al de la Figura III.51.

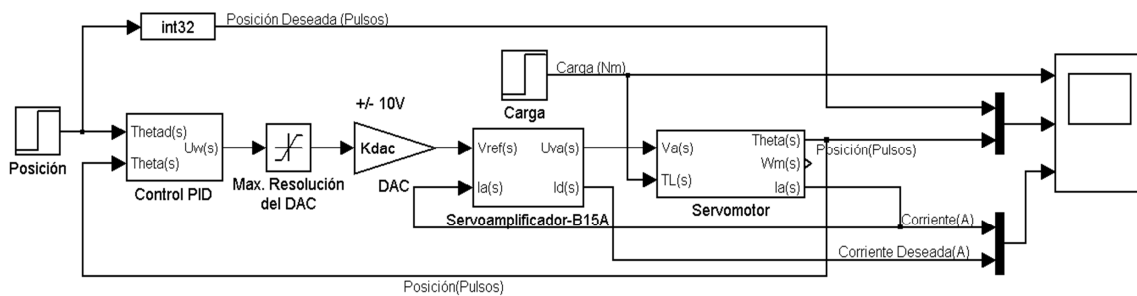


Figura III.55. Esquema del controlador PID de posición.

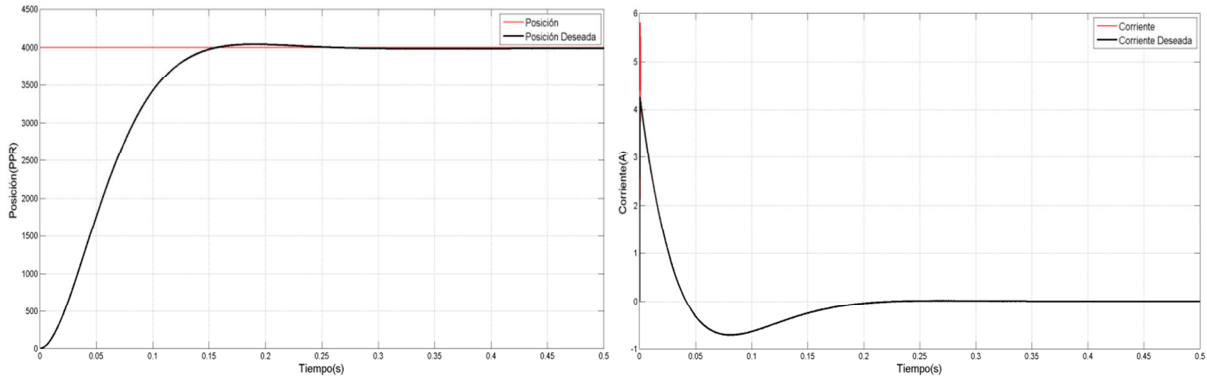


Figura III.56. Diagrama a bloques del controlador PID y su respuesta a una entrada escalón de 4000 PPR.

Utilizando la misma sintonización se realiza el mismo movimiento agregando un perfil trapezoidal con las mismas condiciones de aceleración y velocidad utilizadas en controlador P-PI feedforward, a continuación se muestran los resultados obtenidos:

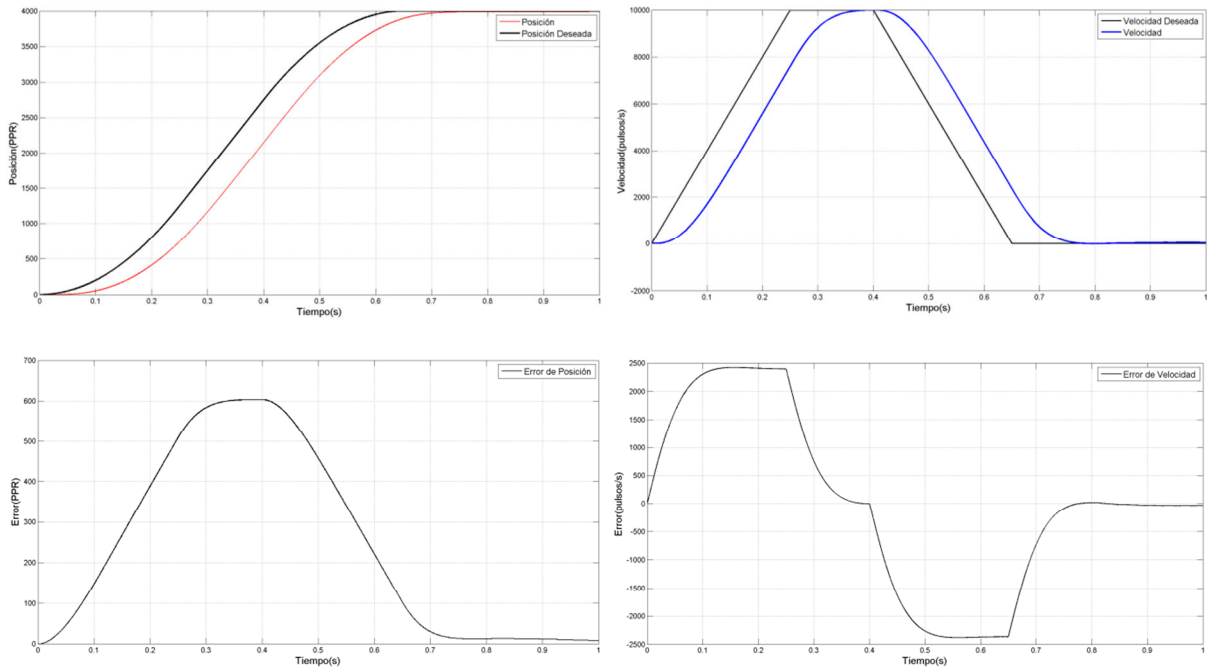


Figura III.57. Respuesta del controlador PID a un perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s² y una posición final de 4000 PPR, $\omega_c=16\text{rad/s}$ y $\Phi=70^\circ$.

Para disminuir el error se realizó una sintonía del controlador PID para tener una frecuencia de cruce de 160rad/s y un margen de fase de 70° , se muestran los resultados obtenidos de la simulación y se agrega la respuesta de la corriente.

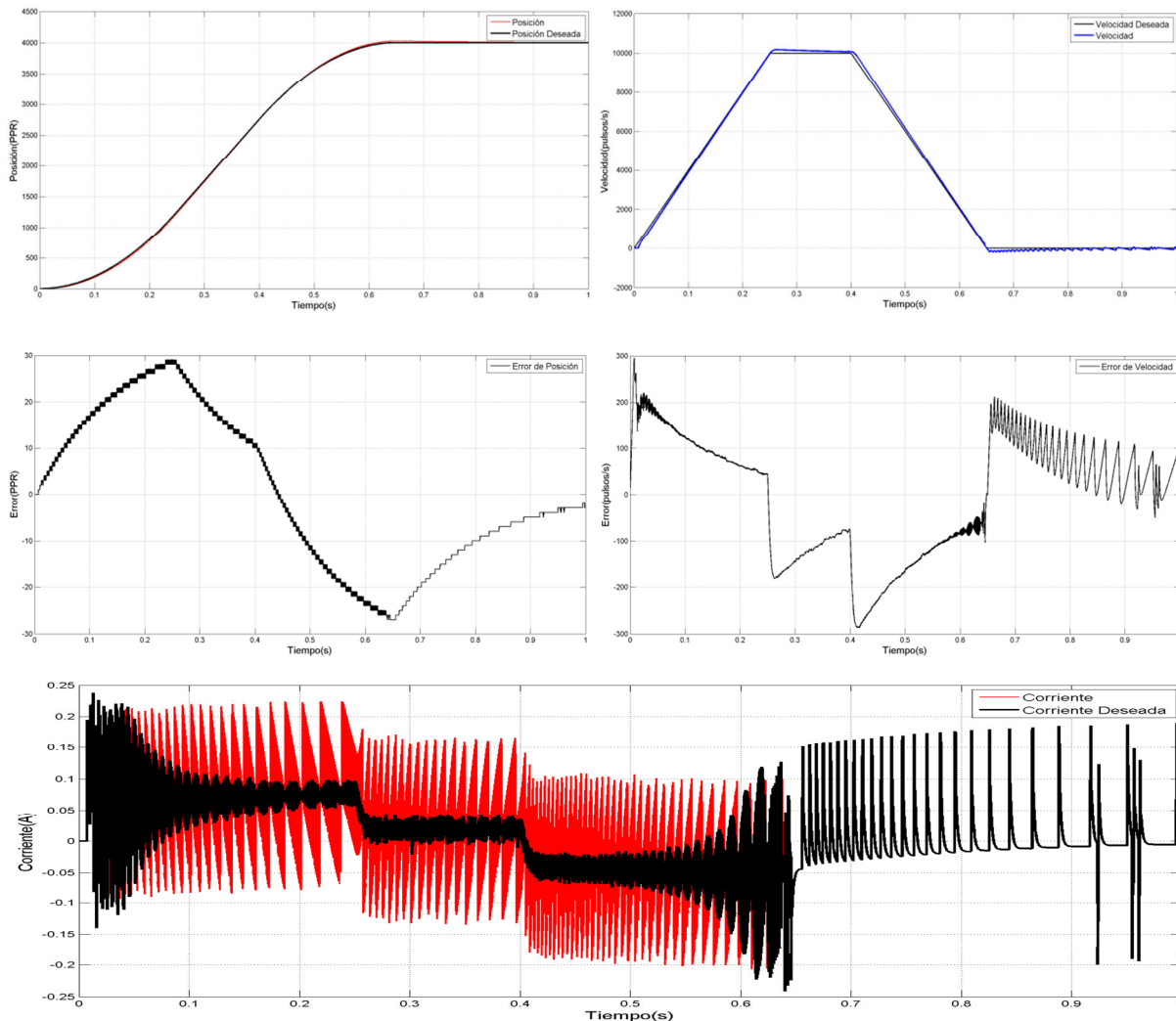


Figura III.58. Respuesta del controlador PID a un perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s² y una posición final de 4000 PPR, $\omega_c=160\text{rad/s}$ y $\Phi=70^\circ$.

Como se puede observar de las Figuras III.54 y Figura III.58 el controlador P-PI feedforward tiene un error menor en el seguimiento del perfil de velocidad además que ofrece un mayor confort para el controlador como se observa en la graficas de la corrientes deseadas, lo cual somete a menos estrés a los componentes del sistema, por lo que se selecciona la topología mostrada en la Figura III.25 y se procede con su discretización.

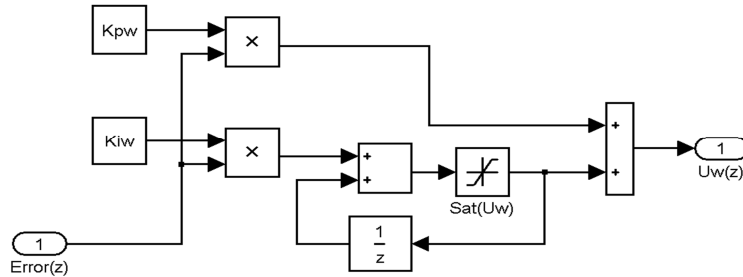


Figura III.60. Controlador discreto PI de velocidad.

Para el controlador de posición como solamente es del tipo proporcional su ecuación es:

$$u_{\theta}(k) = K_{p\omega}e(k) \tag{120}$$

(iii) Perfil trapezoidal de velocidad

El perfil de velocidad esta construido con unidades de integración, el método utilizado fue el método de “Backward Euler”, el método de “Tustin” mostro un mejora muy pequeña pero sin embargo requiere de un número mayor de operaciones, en Figura III.61 se muestra el generador de perfiles en tiempo discreto con un tiempo de muestreo igual a 1ms, lo valores de aceleración (pulsos/s²), velocidad (pulsos/s²) y posición (PPR) son valores enteros con signo.

$$\frac{1}{s} \rightarrow \frac{z T_{s\theta}}{z-1} \rightarrow y(k) = T_{s\theta}x(k) + y(k - 1) \tag{121}$$

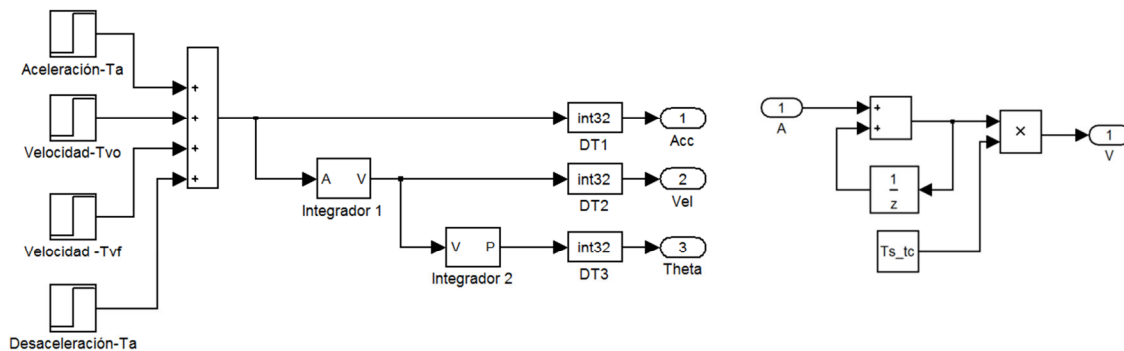


Figura III.61. Generador de perfil trapezoidal de velocidad discreto.

El perfil se genera a partir de entradas tipo escalón que tienen la magnitud de la aceleración del movimiento, estas entradas generan el perfil de aceleración el cual se integra y genera el perfil de velocidad y posición, estos escalones se generan calculando los tiempos necesarios, en donde el perfil tiene tiempos iguales para la aceleración y desaceleración T_a mientras la duración del movimiento a velocidad constante tiene una duración $T_v = T - 2T_a$, en donde T es el tiempo total, dependiendo del valor de aceleración y velocidad el perfil de velocidad generado puede ser un perfil trapezoidal o triangular, a continuación se muestran los resultados de la simulación del controlador de movimiento en tiempo discreto bajo las condiciones antes planteadas. En la Figura III.63 se muestra el controlador de movimiento en tiempo discreto.

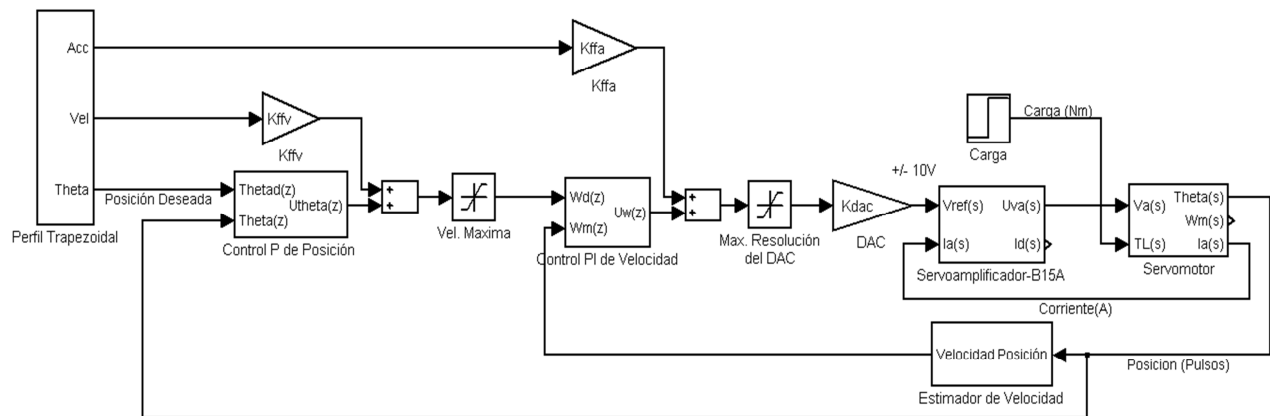


Figura III.62. Modelo del controlador de movimiento en tiempo discreto.

Se realizó primeramente la simulación del lazo de velocidad bajo las mismas condiciones de la Figura III.50, y posteriormente se presenta la respuesta del controlador de posición a una entrada escalón, un perfil de velocidad trapezoidal y triangular.

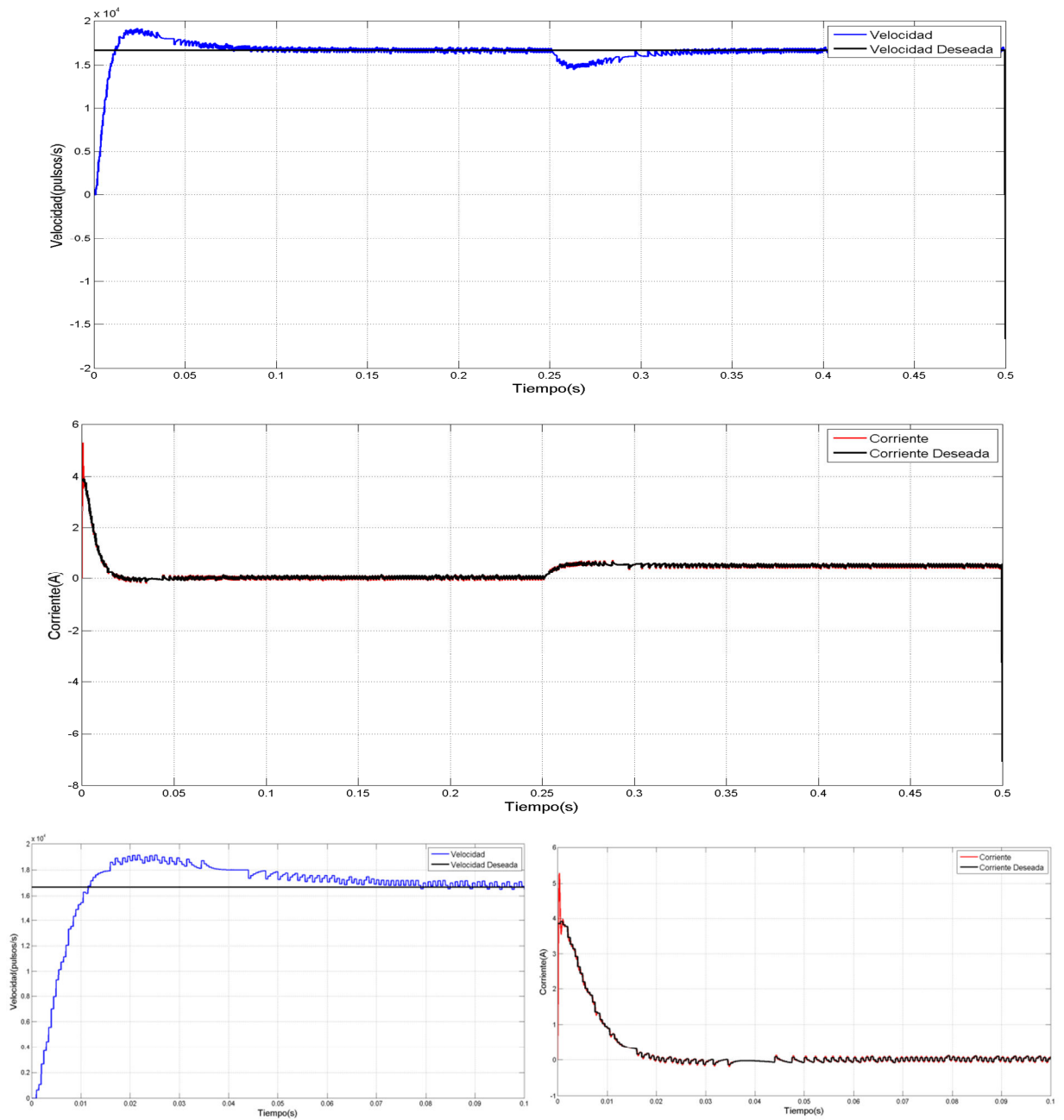


Figura III.63. Respuesta en tiempo discreto del controlador de velocidad a una entrada escalón equivalente a 500RPMs.

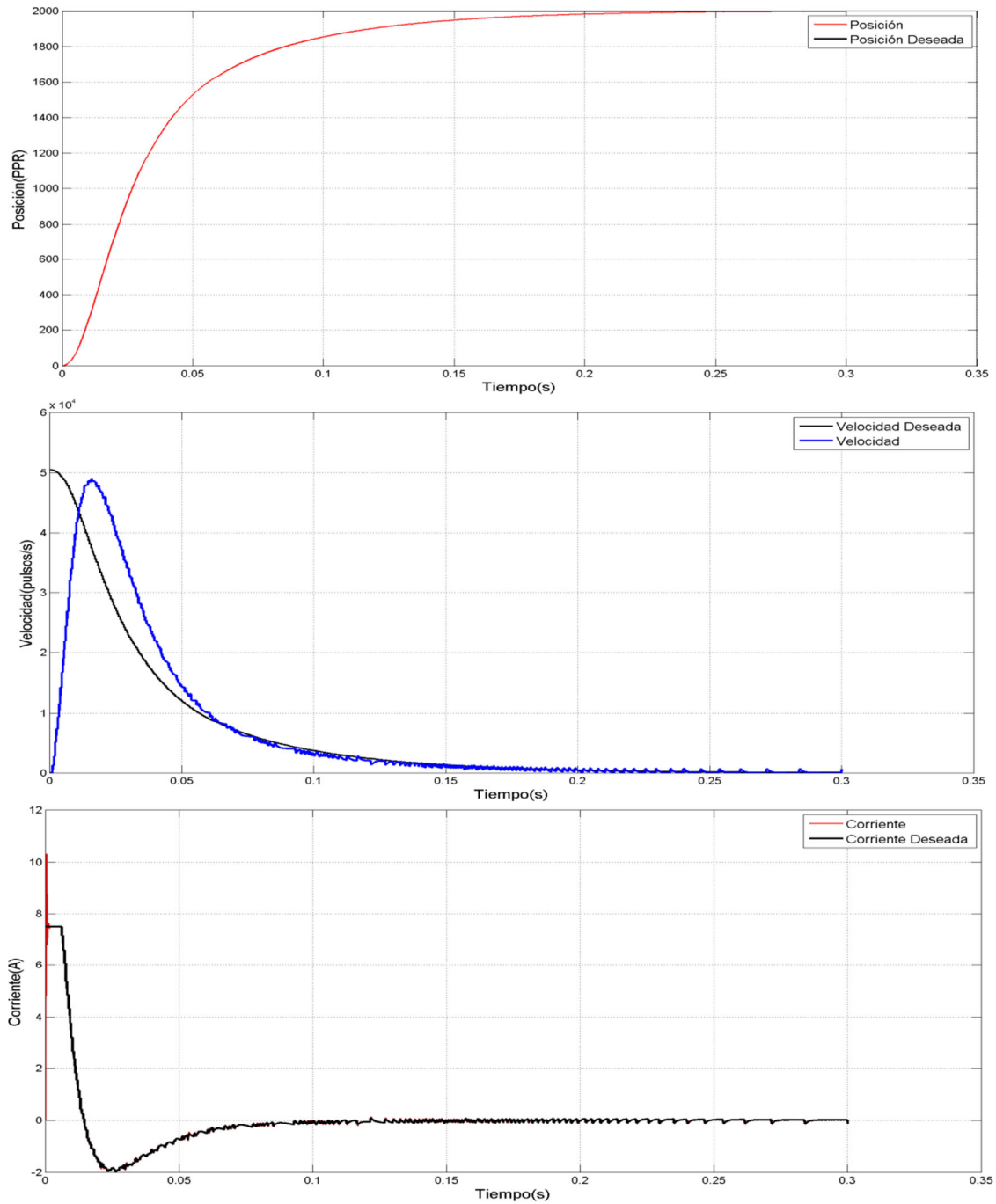


Figura III.64. Respuesta en tiempo discreto del controlador de velocidad a una entrada escalón equivalente a 500RPMs.

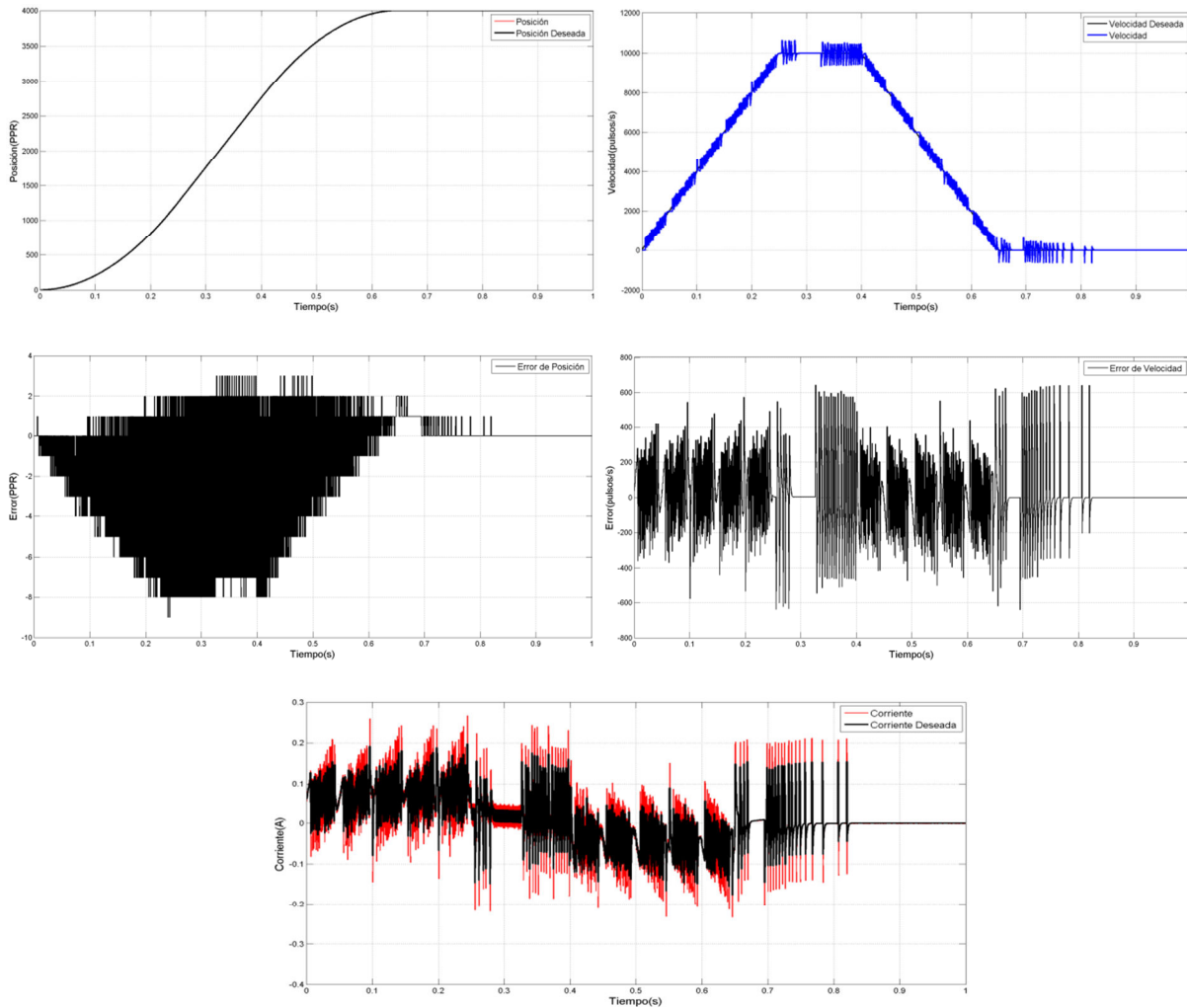


Figura III.65. Respuesta en tiempo discreto del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s² y una posición final de 4000 PPR.

En la Figura III.63 se observa el efecto de la diferenciación numérica de la posición la cual introduce ruido, este método de estimación de la velocidad por medio de las mediciones de la posición reduce el ancho de banda del controlador, sin embargo la respuesta del controlador es estable y converge la posición deseada. A continuación se muestra la respuesta del controlador para un perfil de velocidad triangular, este perfil ocurre cuando dada una velocidad (V) y aceleración (A) y la posición dada por $\frac{V^2}{A}$ es menor a la posición deseada.

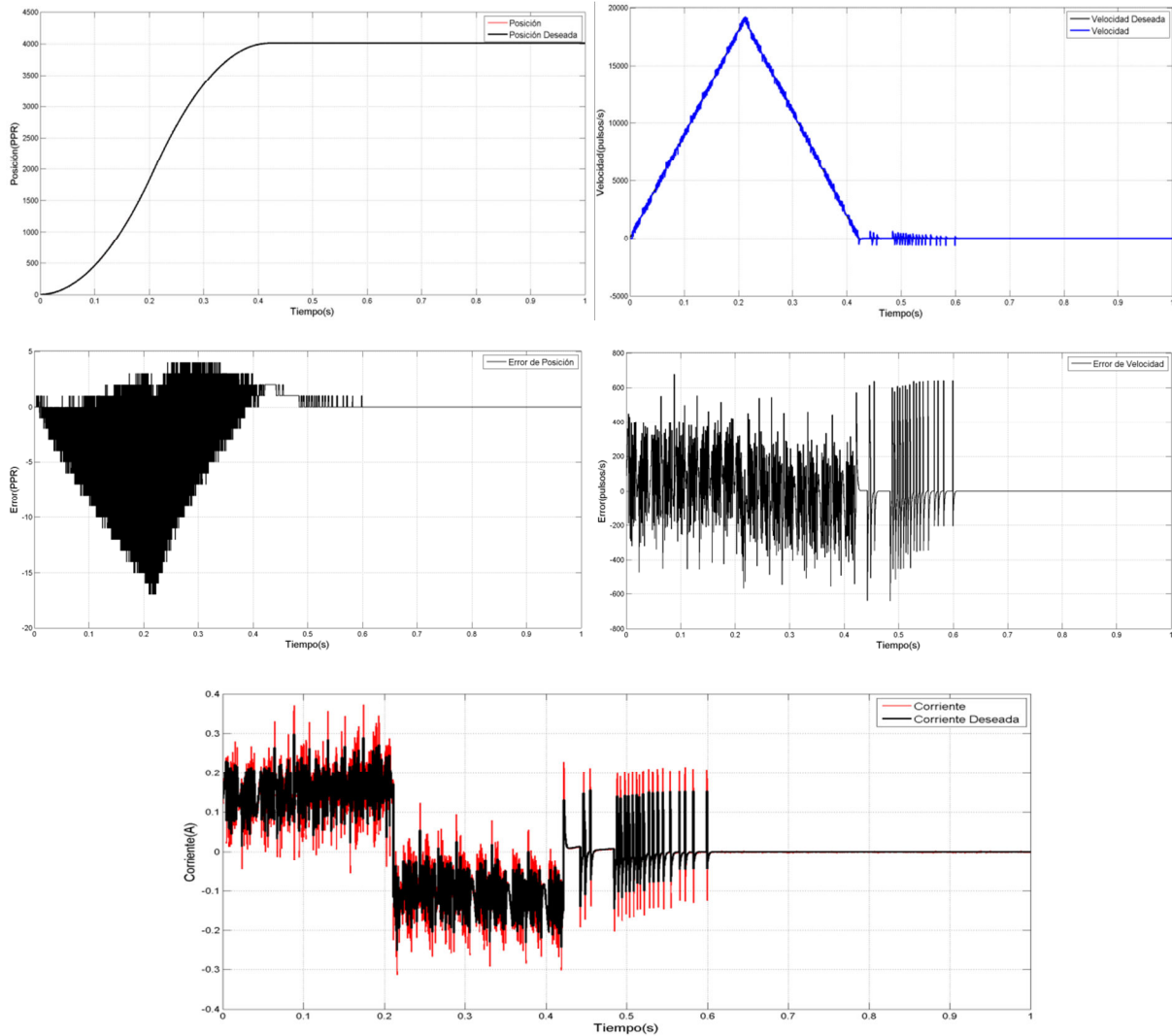


Figura III.66. Respuesta en tiempo discreto del perfil triangular con una velocidad de 100000 pulsos/s, una aceleración de 90000 pulsos/s² y una posición final de 4000 PPR.

Para su implementación en FPGA se realizó una simulación tomando en cuenta los efectos de cuantificación, esto dio pauta para su correcta descripción en hardware y su validación, además de la depuración e identificación de desbordamientos en las señales

(iv) Cuantificación

Para el proceso de cuantificación se eligió una representación en punto fijo; a continuación se presenta su modelado y simulación. Para la representación del formato en punto fijo se utiliza la siguiente notación:

$$F_x = [\text{Signo}, \text{Tamaño de la palabra}, N^\circ \text{ bits para las fracciones}]$$

Por ejemplo se tiene $F_x = [1,32,0]$ lo que quiere decir es que utilizara un entero de 32-bits con signo y cero bits para la parte decimal.

Los formatos elegidos parten de tipo de entrada de cada sub-bloque, y de los valores máximos de referencia de aceleración (23-bits), velocidad (27-bits) y posición (32-bits), además se considero el uso completo de los multiplicadores del FPGA que normalmente son de 9x9 o de 18x18 bits esto proporciono una mayor precisión, se consideraron el número de iteraciones necesarias en cada sub-bloque y se agregaron los bits de guarda necesarios para evitar cualquier desbordamiento, en especial las unidades de integración. A continuación se describen brevemente los módulos del controlador de movimiento los formatos de entrada, de salida y de las constantes o ganancias involucradas en cada sub-bloque, internamente se realizan recortes en la longitud de palabra para obtener el formato de salida deseado.

- **Generador de perfiles trapezoidales:** El formato para la a velocidad, aceleración y posición máxima del perfil tienen un formato $F_{x_A} = [1,23,0]$, $F_{x_V} = [1,24,0]$ y $F_{x_P} = [1,32,0]$ respectivamente, para uno bloques de integración del perfil se tiene de base el tiempo el periodo de muestreo del controlador de posición $T_{s\theta}$ el cual tiene un formato $F_{x_{T_{s\theta}}} = [1,18,17]$.
- **Estimador de velocidad:** El formato de entrada de las mediciones de posición es $F_{x_\theta} = [1,32,0]$, el formato de salida de la velocidad estimada en pulsos/s es $F_{x_{\omega_e}} = [1,27,0]$, en la estimación de la velocidad intervienen dos constantes la $C_1 = K_1$ y $C_2 = K_2$ estas se representaron utilizando un formato $F_{x_{c1}} = [1,18,5]$ y $F_{x_{c2}} = [1,18,15]$ respectivamente.

- **Controlador P de posición:** Se tienen los formatos de entrada que son F_{x_θ} y F_{x_p} la diferencia de estos que es el error se le hizo una extensión signo para tener un formato $F_{x_{e_\theta}} = [1,36,0]$ y la ganancia proporcional tiene un formato $F_{x_{K_{p\theta}}} = [1,18,10]$, dándonos como salida un formato $F_{x_{\theta u}} = [1,54,10]$ que fue recortado a $F_{x_{\theta u}} = [1,38,10]$.
- **Ganancia feedforward de velocidad:** Para el calculo de la ganancia de pre-alimentación de velocidad , primeramente la velocidad de referencia del perfil (V) tiene un formato $F_{x_{V_{ff}}} = [1,27,0]$ y para la ganancia de velocidad se usa un formato $F_{x_{K_{ff\omega}}} = [1,18,14]$ donde se tiene un formato de salida $F_{x_{Ku_{ff\omega}}} = [1,45,10]$ que se recorta a $F_{x_{Ku_{ff\omega}}} = [1,38,10]$, la cual se suma con la salida del controlador de posición $F_{x_{\theta u}}$ y posteriormente se aplica la función de saturación descrita por (111) en donde P_{max} es un numero de 36-bits con signo que es formato de salida en el estimador de velocidad $F_{x_{\omega e}}$.
- **Controlador PI de velocidad:** El formato de la velocidad deseada es el mismo formato de salida del estimador de velocidad es decir $F_{x_{\omega d}} = F_{x_{\omega e}}$; para el formato de salida del control PI es $F_{x_{\omega u}} = [1,27,10]$ y la ganancias del controlador $K_{p\omega}$ y $K_{i\omega}$ tienen un formato $F_{x_{K_{p\omega}}} = [1,18,10]$ y $F_{x_{K_{i\omega}}} = [1,18,10]$ respectivamente.
- **Ganancia feedforward de aceleración:** Para el calculo de la ganancia de pre-alimentación de aceleración , primeramente a la aceleración de referencia del perfil (A) se aplicó una extensión de signo $F_{x_{A_{ff}}} = [1,27,0]$ y para la ganancia de aceleración tiene un formato $F_{x_{K_{ff\varepsilon}}} = [1,18,17]$ y se tiene un formato de salida $F_{x_{Ku_{ff\varepsilon}}} = [1,45,17]$, el cual se suma con la salida del controlador de velocidad $F_{x_{\omega u}}$ y posteriormente se aplica la función de saturación descrita por (106) para finalmente escribir el valor de referencia en el DAC.

A continuación se muestran los resultados de la simulación considerando los efectos de la cuantificación y se comparan resultados con el modelo en tiempo discreto.

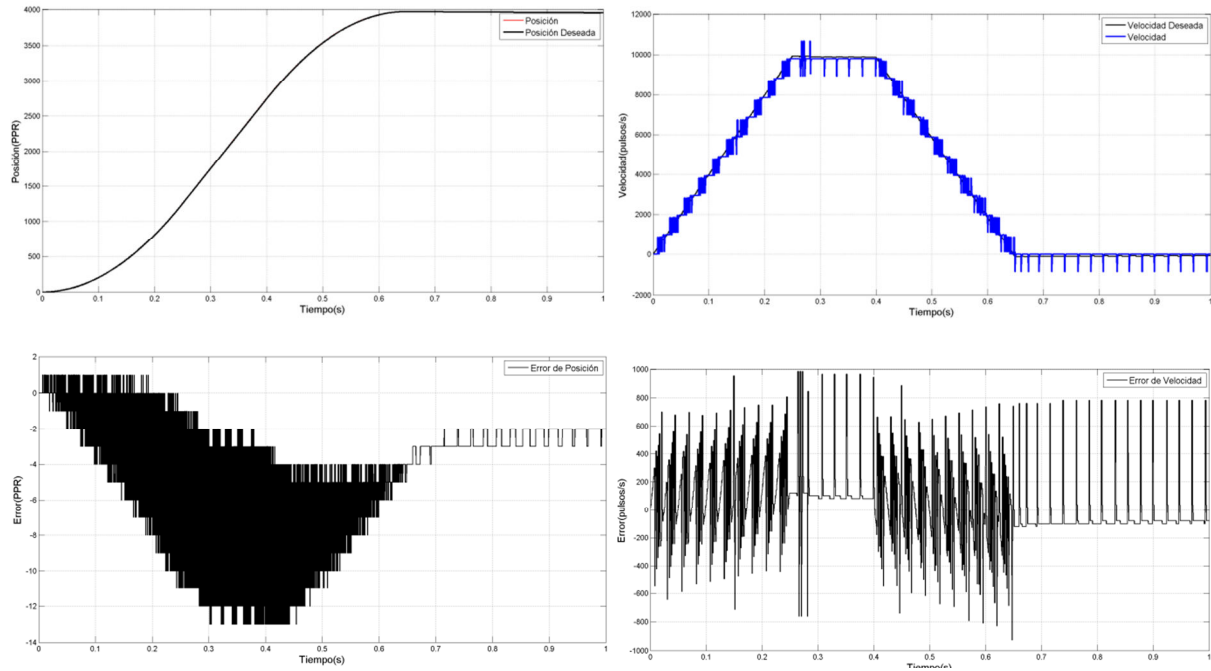
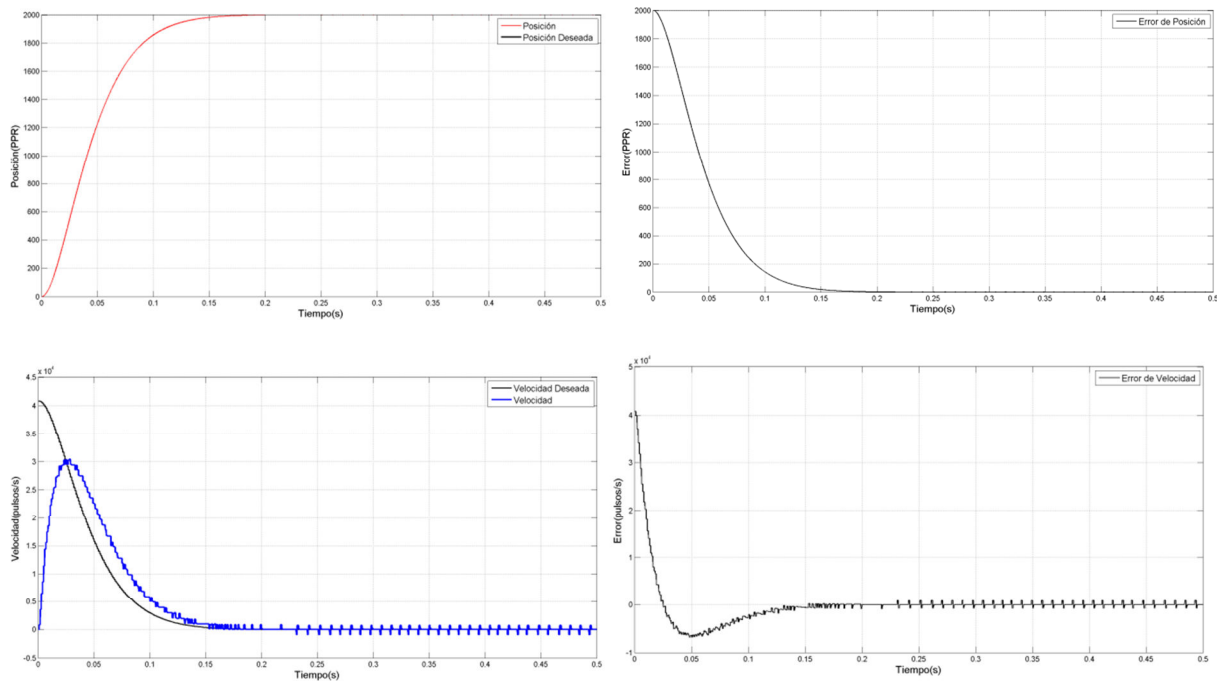


Figura III.67. Respuesta en tiempo discreto y cuantificada del perfil trapezoidal con una velocidad de 10000 pulsos/s, una aceleración de 40000 pulsos/s² y una posición final de 4000 PPR.



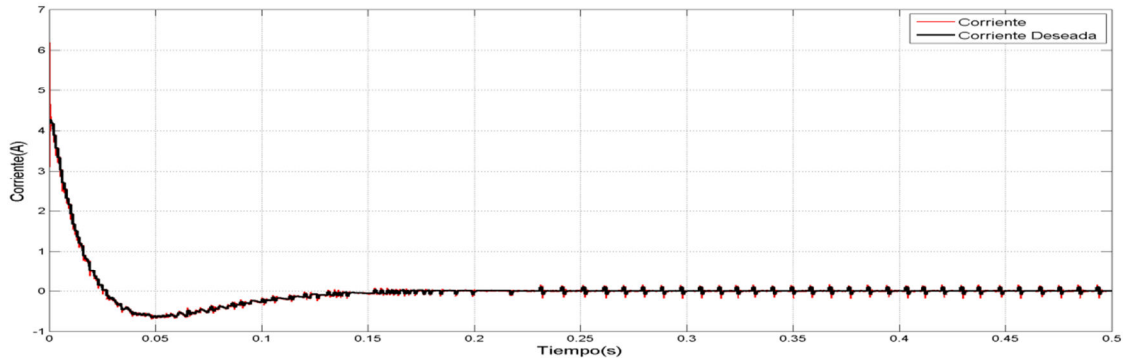


Figura III.68. Respuesta a un escalon de 4000 PPR en tiempo discreto y cuatificada en punto fijo.

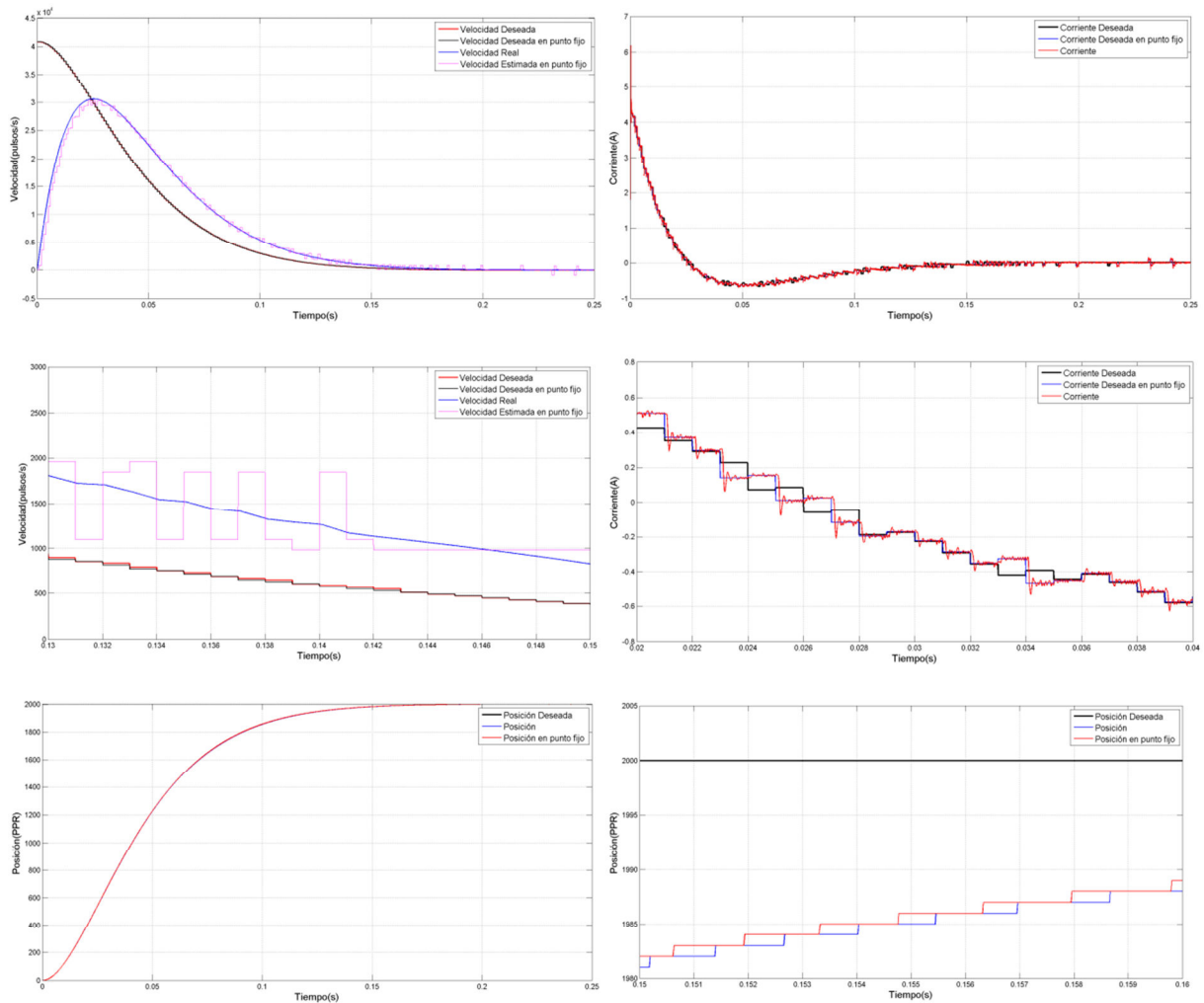


Figura III.69. Comparación de la respuesta a un escalon de 4000 PPR en tiempo discreto y la respuesta del sistema cuatificado en punto fijo.

Como se puede observar en la Figura III.69 la diferencia entre el modelo en tiempo discreto que utiliza una representación en punto flotante de doble precisión (64-bits) y la respuesta del modelo cuantificado en punto fijo difieren con un mínimo error, lo cual no afecta de forma significativa la respuesta del algoritmo de control; el modelado de los diferentes componentes y su equivalente discreto así como su simulación en punto fijo fueron la base para su correcta implementación y validación en hardware, el uso de herramientas de software para validar y diseñar un sistema en hardware es conocida como metodología de co-diseño, y su objetivo principal es retrasar la elección e implementación en hardware para reducir riesgos y tiempos de desarrollo.

III.2.3 Descripción de Hardware

Para la implementación de controlador de movimiento planteado se eligió un FPGA de Altera de la familia Cyclone IV debido a la densidad de integración (el número de elementos lógicos (LE)), el número de multiplicadores y las herramientas de desarrollo que ofertan (Quartus II y Nios II) y la disponibilidad de licencias de desarrollo completas, el lenguaje empleado para de descripción de hardware fue VHDL en su estándar 2008, se siguieron las recomendaciones y metodologías de diseño planteadas en (Baese, 2001) , (Wilson, 2007), (Ashenden, 2008), (Smith, 2010) y (Kafig, 2011), el diseño realizado se hizo a nivel de registro de transferencia (RTL) evitando en lo posible circuitos diseñado a nivel de comportamiento, esto con la finalidad de aprovechar los recursos en hardware del FPGA y crear código completamente sintetizable. En la Figura III.70 un diagrama a bloques del sistema propuesta de control de movimiento multieje.

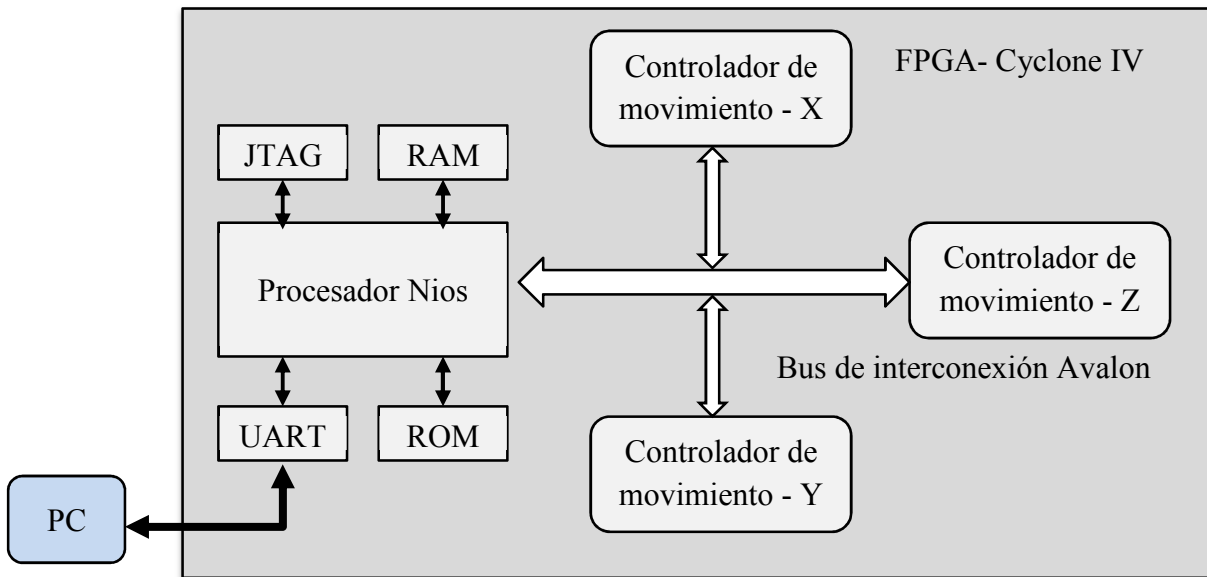


Figura III.70. Arquitectura del controlador de movimiento propuesto.

A continuación se describen los componentes del controlador de movimiento estándar, para el diseño propuesto y la elección de los componentes se basó en la información técnica de marcas comerciales como Galil, NI, Maxon Motor, Performance Motion Device, Faulhaber y ADLINK entre otras.

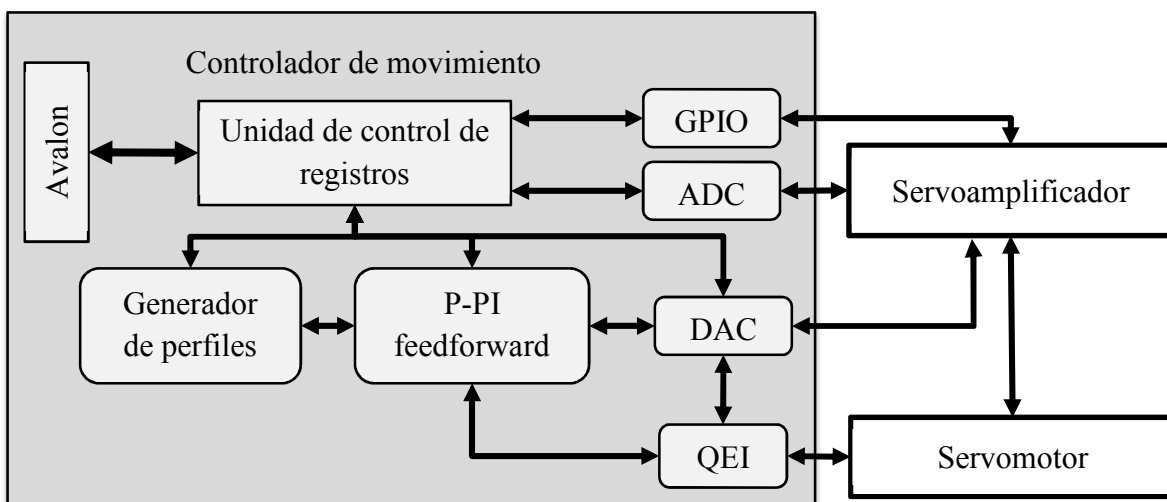


Figura III.71. Componentes del controlador de movimiento propuesto.

(i) Interfaz de Cuadratura de Encoder (QEI)

Para la implementación de el circuito de tomaron en cuenta algunas de características y consideraciones mencionadas en (Boggarpu et al., 2010), (Petrella et al., 2007), (Faccio et al., 2004) y (Tsai et al., 2002), para mejorar la confiabilidad y seguridad del modulo QEI. A continuación se muestra un diagrama a bloques de la unidad desarrollada.

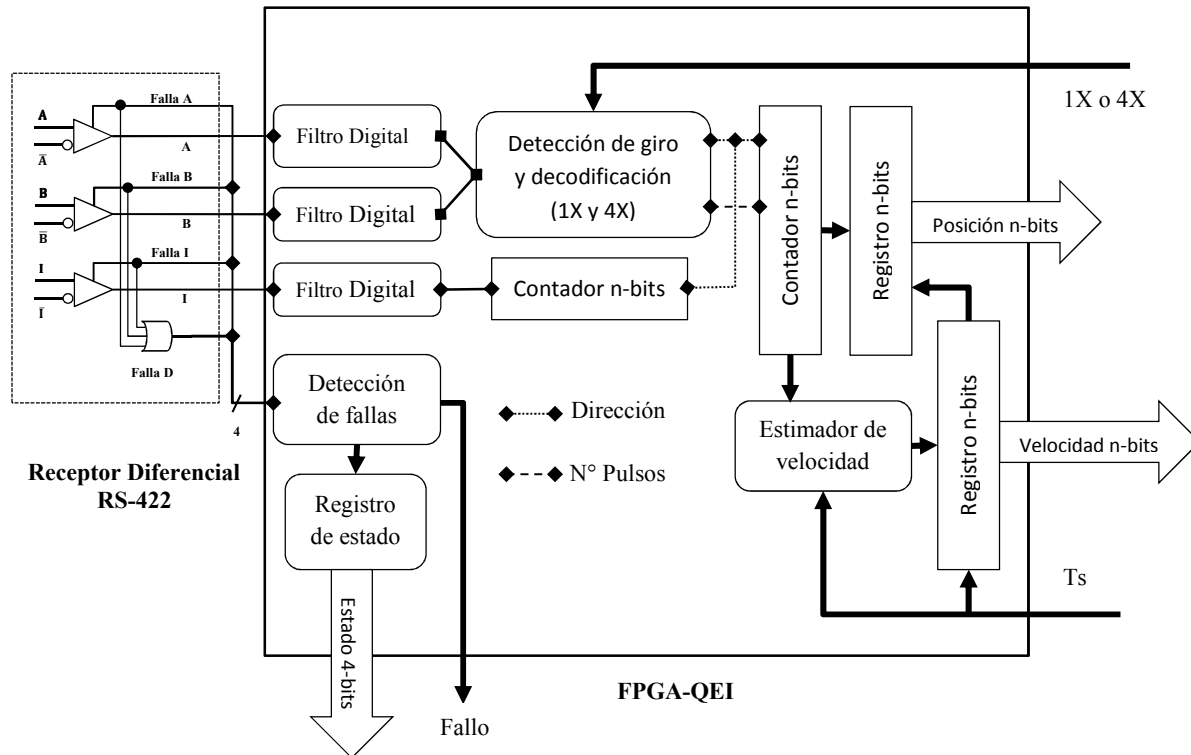


Figura III.72. Diagrama a bloques del modulo QEI.

Primeramente en la parte eléctrica, las entradas de la circuitería consisten en un receptor diferencial RS-422 especial (MAX3098E) para su uso en aplicaciones robustas de control de motores (MAXIM IC, 2001), sin embargo este viene diseñado solo trabajar con señales de tipo diferencial; la entrada estándar del RS-422 es normalmente terminada con una resistencia de 120Ω, pero en si depende del valor de resistencia del cableado y normalmente se encuentra en un rango de 75-330Ω. Para compensar su diseño se tuvo que agregar resistencias de terminación para poder utilizar entradas con niveles de lógica transistor-transistor (TTL) y conexiones de terminación simple o “Single-Ended”, como se muestra en la Figura III.73, de tal forma que solo removiendo la resistencia de terminación de la red (R1) podemos tener una interfaz robusta para

ambos tipos de decodificadores incrementales con terminaciones diferenciales o simples. Las resistencias R2 y R3 son utilizadas para soportar terminaciones “single-ended”, la carga equivalente es pequeña para la terminación RS-422 y no provoca un desbalanceo significativo en la entrada, debido a esto R2 y R3 deben de mantenerse incluso si una la entradas en modo RS-422. Cuando se use una terminación simple TTL no se deben dejar las entradas flotadas, esto produce una salida inconsistente en el comparador y puede producir errores de operación la unidad QEI.

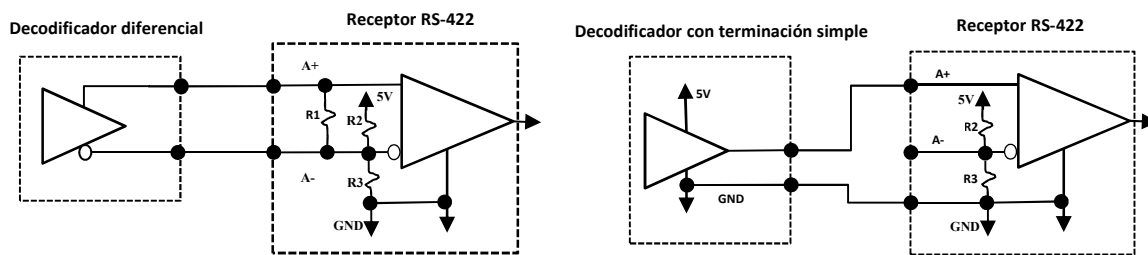


Figura III.73. Terminaciones para un decodificador incremental simple y diferencial.

Adicionalmente el circuito utilizado nos proporciona señales de alarma que indican diferentes tipos de fallos en el sistema de retroalimentación del controlador de movimiento, algunas de las fallas posibles para detectar son:

- Desconexión para cada una de las señales A, B e I.
- Exceso en el rechazo al modo común.
- Detección de corto circuito entre canales.
- Detección de bajo nivel de tensión en modo diferencial (<200mV).

A continuación se describen los principales componentes para el diseño del modulo QEI en el FPGA:

Filtro digital: Esta unidad se utiliza principalmente para sincronizar y evitar problemas de meta-estabilidad ocasionados por los eventos asíncronos de las señales de cuadratura, por que se utilizan FFD para sincronizar las señales y además se añade un filtro para eliminar los pulsos de pequeña duración que normalmente ocurren en aplicaciones de control de motores, sin embargo

este tipo de filtros genera latencia, el circuito utilizado para el filtrado de la señales puede ser encontrado en soluciones ASIC como el IC HCTL-2032, se muestra en la Figura III.74.

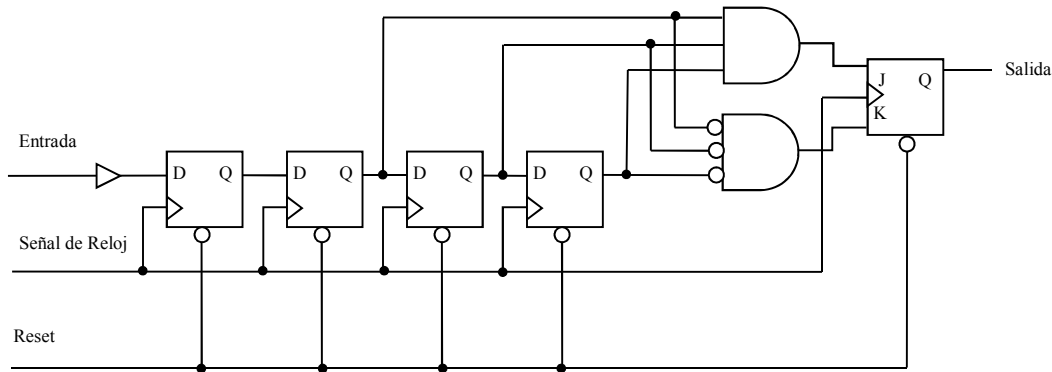


Figura III.74. Filtro digital.

Se debe tomar en consideración la duración mínima de un pulso, T_{min} , de las señales de cuadratura que esta dada por la siguiente ecuación:

$$F_{max} = \omega_{max} P_{PR} \quad (122)$$

$$T_{min} = \frac{F_{max}}{2} \quad (123)$$

En donde F_{max} es la frecuencia máxima del tren de pulsos de la señal, ω_{max} la velocidad máxima de rotación en RPMs. En este caso la máxima frecuencia de operación de la unidad QEI y tiempo mínimo del ancho de pulso esta dado por la frecuencia de operación del FPGA (F_{CLK}) y el numero de retardos de la señal para que sea un dato valido, en este caso son tres FFD conectados en cascada por lo que la frecuencia máxima de la señales de cuadratura, $F_{QEI_{max}}$, y la duración mínima de pulso están dados por:

$$T_{QEI_{min}} = \frac{3}{F_{CLK}} \quad (124)$$

$$F_{QEI_{max}} = \frac{1}{T_{min}} \quad (125)$$

Decodificador de cuadratura: Para decodificación de las señales de cuadratura se creó una unidad que sea capaz de trabajar con al menos dos tipos de decodificación 1X y 4X; esta unidad incluye el detector de cambio de giro del motor. A continuación en la Figura III.75 se muestra el circuito descrito.

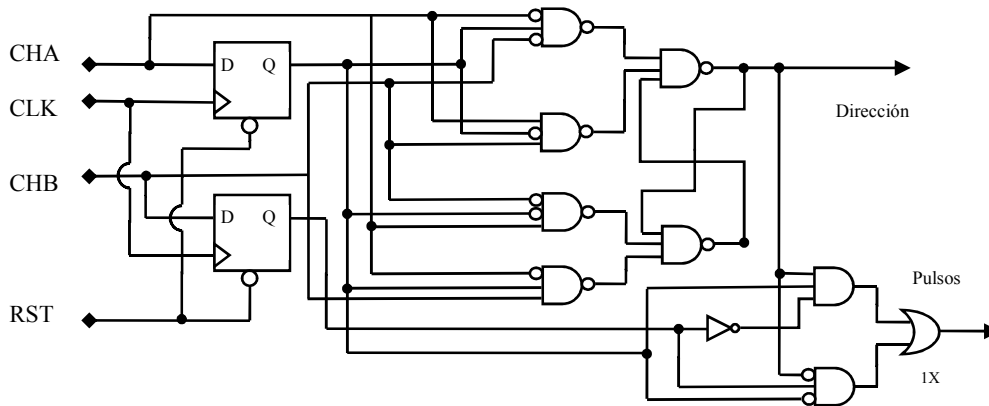


Figura III.75. Unidad de decodificación y detección del sentido de giro.

Para la verificación de los componentes se crearon bancos de pruebas y describir los IP cores para el sistema de control de movimiento a continuación se muestran los resultados del filtro digital y de la unidad de cuadratura. Se muestra el IP core para la unidad de cuadratura la cual incluye el estimador de velocidad descrito anteriormente.

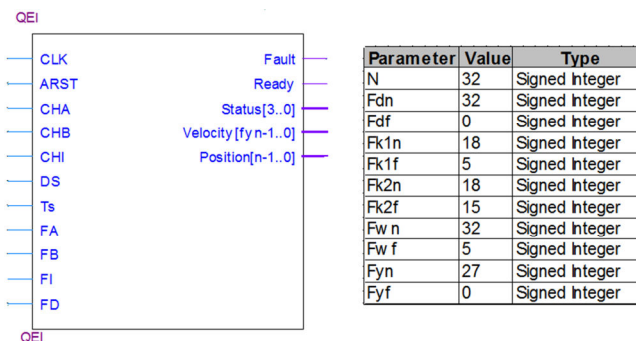


Figura III.76. Verificación del Filtro digital del core QEI.

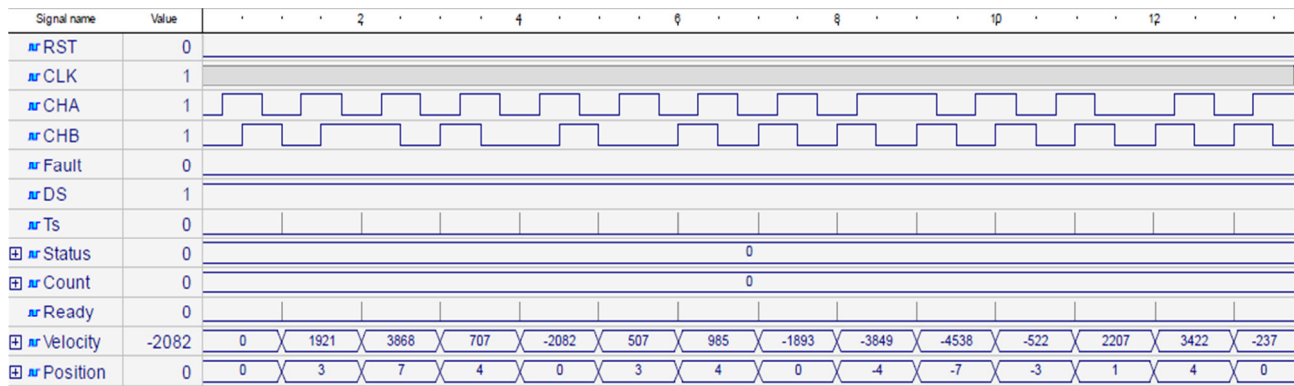


Figura III.77. Verificación del core de cuadratura QEI.

(ii) DAC y ADC

El DAC utilizado fue el LTC2642-16- Linear Technology, se selecciono por que esta diseñado para trabajar con sistemas de adquisición y control industrial de rápida respuesta, es bipolar (+/- 10V) y tiene un restablecimiento a media escala (0V) en el encendido “Power on reset to midscale” el cual nos ayuda a mantener la señal de control en cero durante el encendido del sistema, en la siguiente tabla se muestran las características principales:

Tabla III.6. Características principales del DAC LTC2642-16

Resolución	16-bits
Interfaz de comunicación	Serial Peripheral Interface (SPI) hasta 50MHz
Error de linealidad no integral	+/- 0.5 LSB
Error de linealidad diferencial	+/- 0.5LSB
Rango de Voltaje	+/- 5V

La velocidad de operación resulta conveniente para trabajar con el FPGA ya que pueden operar a la misma frecuencia y no se necesita de divisores de frecuencia o contadores para generar la señal de reloj lo que ayuda a disminuir el uso de recursos en el diseño. La ecuación característica del DAC esta dada por la siguiente expresión:

$$V_{out_Bipolar} = V_{ref} \left(\frac{k}{2^{N-1}} - 1 \right) \quad (126)$$

El formato de entrada del DAC es radiométrico por lo que tenemos que negar el MSB para tenerlo en complemento a dos, en la Figura III.78 se muestra el diagrama de tiempos de escritura y se muestra el bloque de la entidad descrita, esta entidad contiene una maquina de tres estados y convertidor de paralelo a serial.

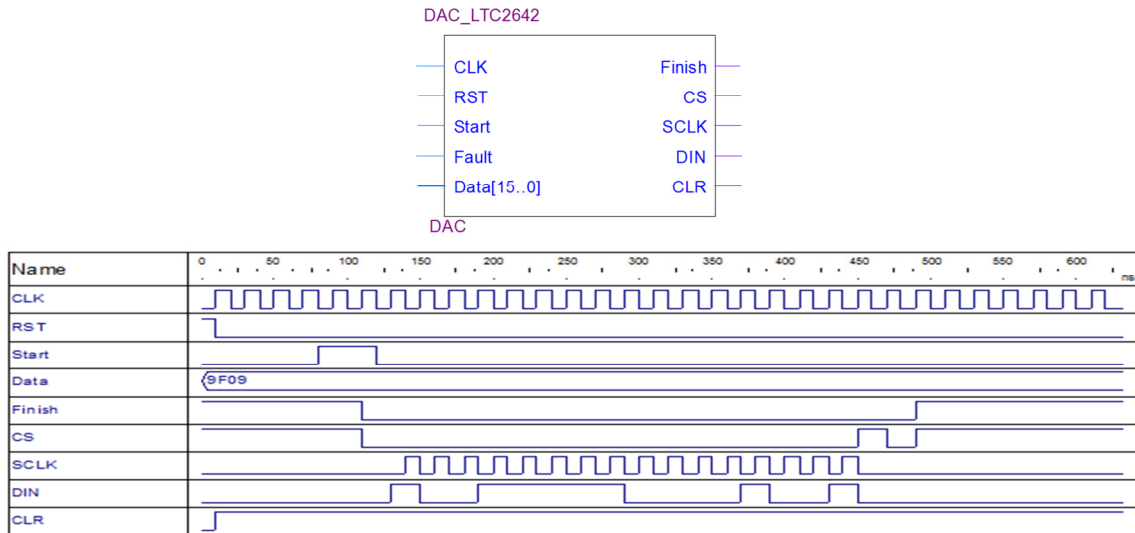


Figura III.78. Verificación del proceso de escritura del DAC o “bit timing diagram”.

De igual manera se selecciono un ADC del tipo bipolar basándonos en las características de productos comerciales y el tipo de aplicación se selecciono el ADC AD7687 de Analog Devices a continuación se muestran las características principales:

Tabla III.7. Características principales del ADC AD7687.

Resolución	16-bits
Interfaz de comunicación	SPI hasta 50MHz
Error de linealidad no integral	+/- 0.4 LSB
Error de linealidad diferencial	+/- 0.4LSB
Rango de Voltaje	+/- 10V
Tasa de conversión	250KSPS

A continuación se muestra el la entidad descrita la cual consiste en una maquina de estado y convertidor de serial a paralelo.

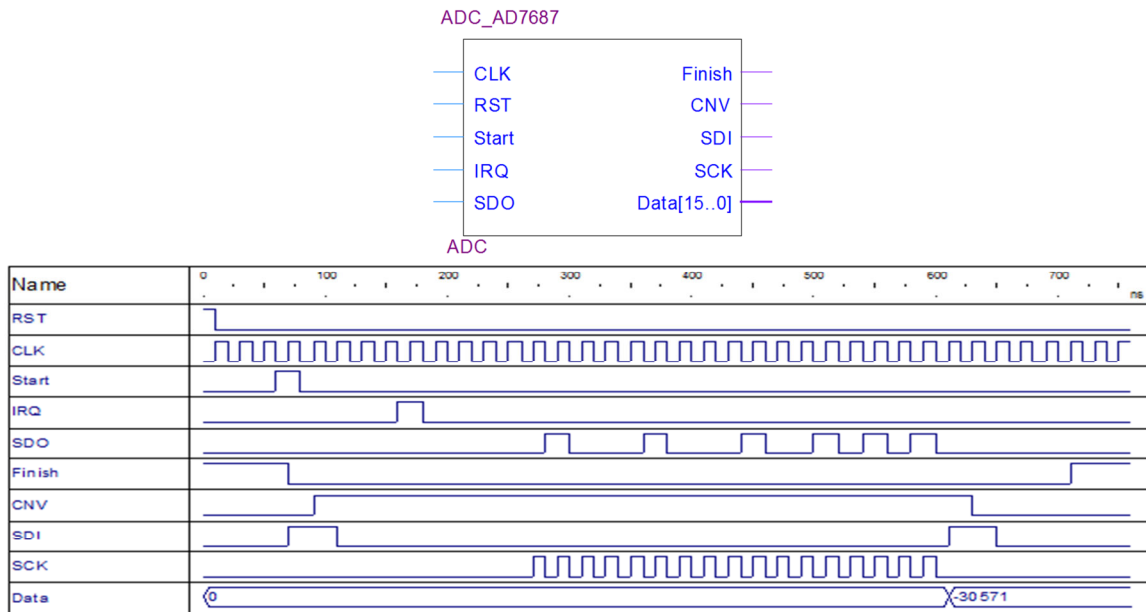
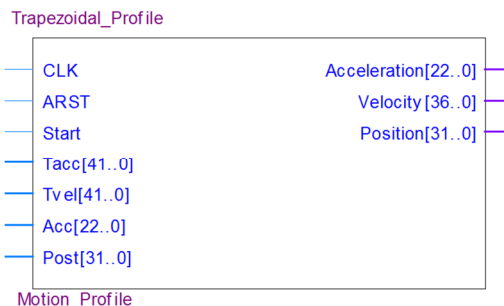


Figura III.79. Verificación del proceso de escritura del ADC.

(iii) Generador de perfiles trapezoidal

A continuación se muestra el generador de perfiles trapezoidales y triangulares, como se menciona la generación de uno u otro perfil esta en función de la posición, velocidad y aceleración deseada, los parámetros de entrada son el tiempo de aceleración y velocidad que son previamente calculados. En la Figura III.80 y III.81 se muestra la generación de un perfil trapezoidal y una secuencia de una trayectoria la cual genera primeramente un perfil triangular seguido de uno trapezoidal.



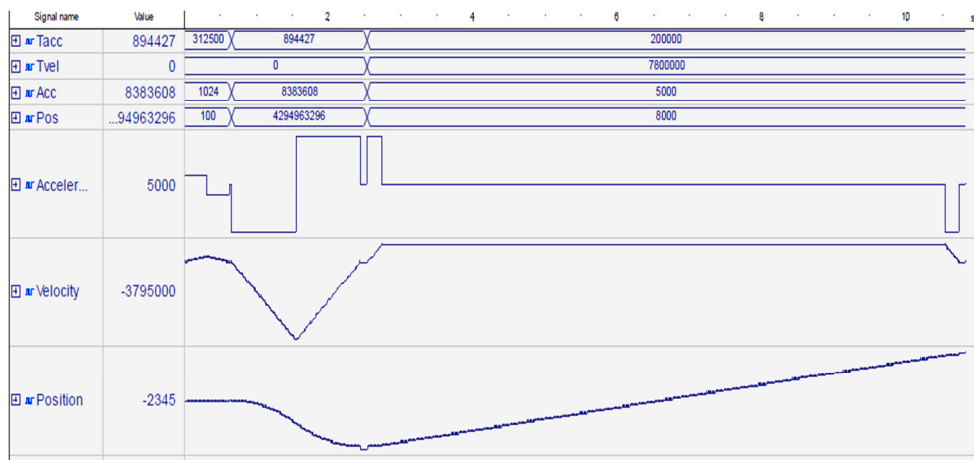
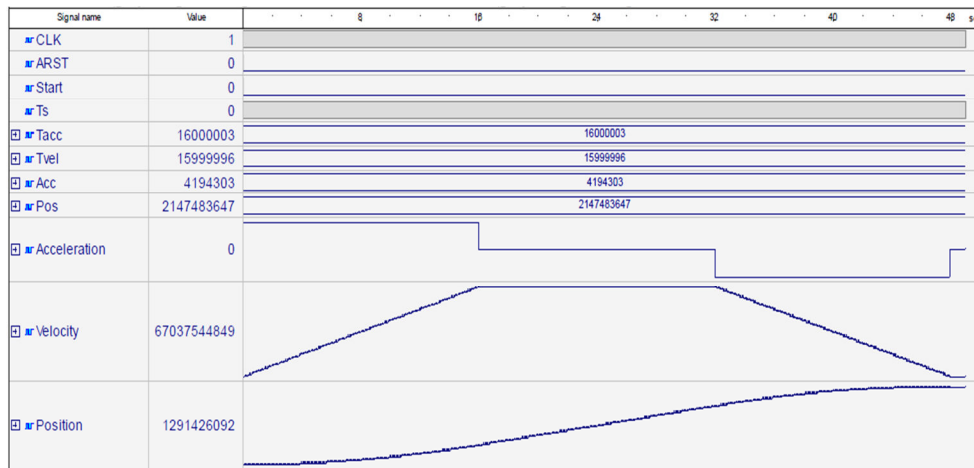


Figura III.80. Verificación del generador de perfiles de velocidad.

Una vez descritos y verificados todos los componentes del controlador de movimiento se procedió con la experimentación en la cual se probaron físicamente los perfiles anteriormente descritos y se comparan los resultados simulados y reales.

CAPÍTULO 4:

IV RESULTADOS Y DISCUSIÓN

IV.1 Identificación del sistema

Se muestran los resultados obtenidos de la identificación del sistema de manera experimental, a continuación se muestra la interfaz desarrollada en LabVIEW para la identificación de sistemas como modelos de primer orden con retardo en el tiempo.

$$\frac{\Omega_a}{V_a} = \frac{Ke^\Theta}{Ts+1} \quad (127)$$

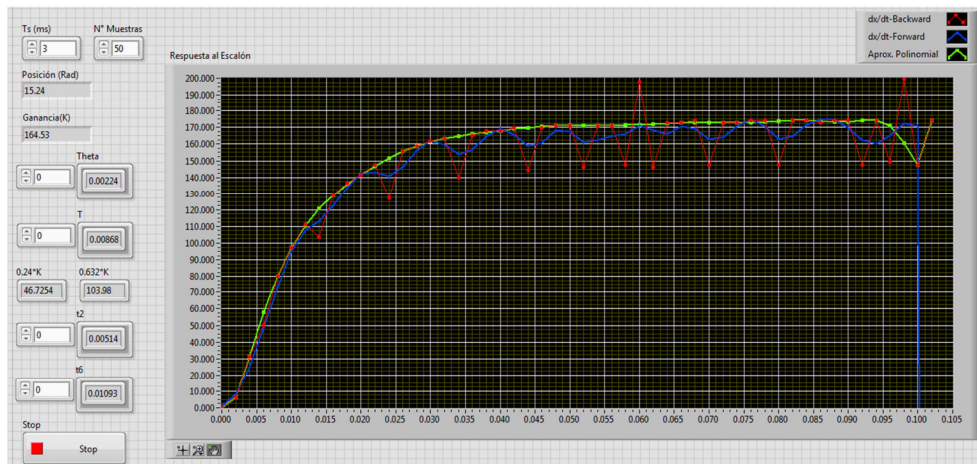


Figura IV.1. Interfaz en LabVIEW para la identificación de sistemas.

Para identificar el motor de DC de la Tabla III.4 se aplicó una entrada escalón con una magnitud de 24V, en la figura siguiente se muestra la respuesta medida y se estima un modelo de la planta y se compara con el modelo matemático desarrollado; para tener una mejor resolución se aplica una aproximación polinomial de la respuesta medida.

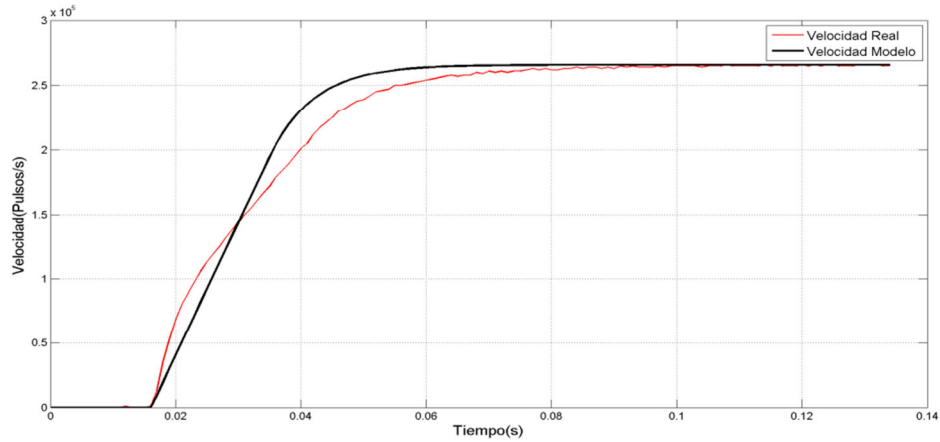


Figura IV.2. Respuesta del motor a una entrada escalón de 24V.

Para la identificación del motor se aplicó una entrada escalón de 24V con una fuente con corriente limitada a 15A, el periodo de muestreo fue de un milisegundo; El coeficiente de correlación de Pearson entre las señales de Figura IV.2 es de $\rho = 0.98$. Los valores encontrados para el modelo de primer orden fueron $K = 34.81$, $T = 0.0195$ y un retardo $\Theta = 0.0015$, por lo que se tiene la siguiente función de transferencia, para esta función es multiplicada por la ganancia del decodificador que es 318.3.

$$\frac{\Omega_a}{V_a} = Kf * \frac{34.81e^{0.0015s}}{0.0195s+1} \quad (128)$$

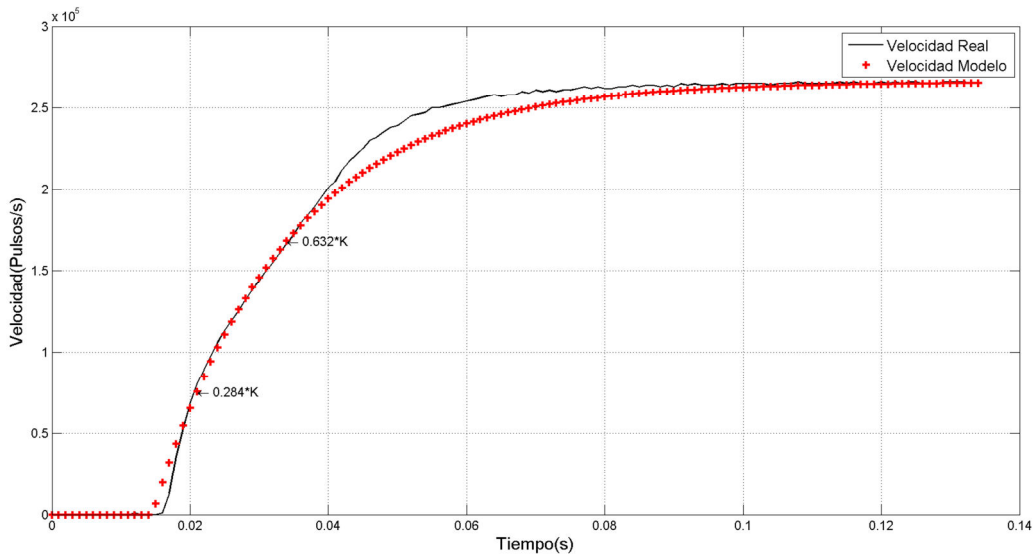


Figura IV.3. Comparación de la respuesta al escalón de modelo matemático y el modelo estimado.

De los resultados anteriores podemos concluir que el modelo matemático utilizando en la metodología de diseño del controlador de movimiento representa fielmente al modelo físico del motor y ambas funciones de transferencia convergen a valor nominal de velocidad del motor, este resultado es muy importante debido a que valida la modelación y simulaciones presentadas.

IV.2 Controlador PI de velocidad

A continuación se muestra la respuesta al escalón del lazo de velocidad presentado en la Figura III.63 y se compara con la respuesta de modelo simulado, el periodo de muestreo se mantuvo a 1ms, la señales adquiridas tienen un factor de correlación de $\rho = 0.95$, la ganancias del controlador son $K_{pw}=0.37$ y $K_{iw}=0.005$ se calcularon utilizando el modelo en tiempo continuo con el método de margen de fase.

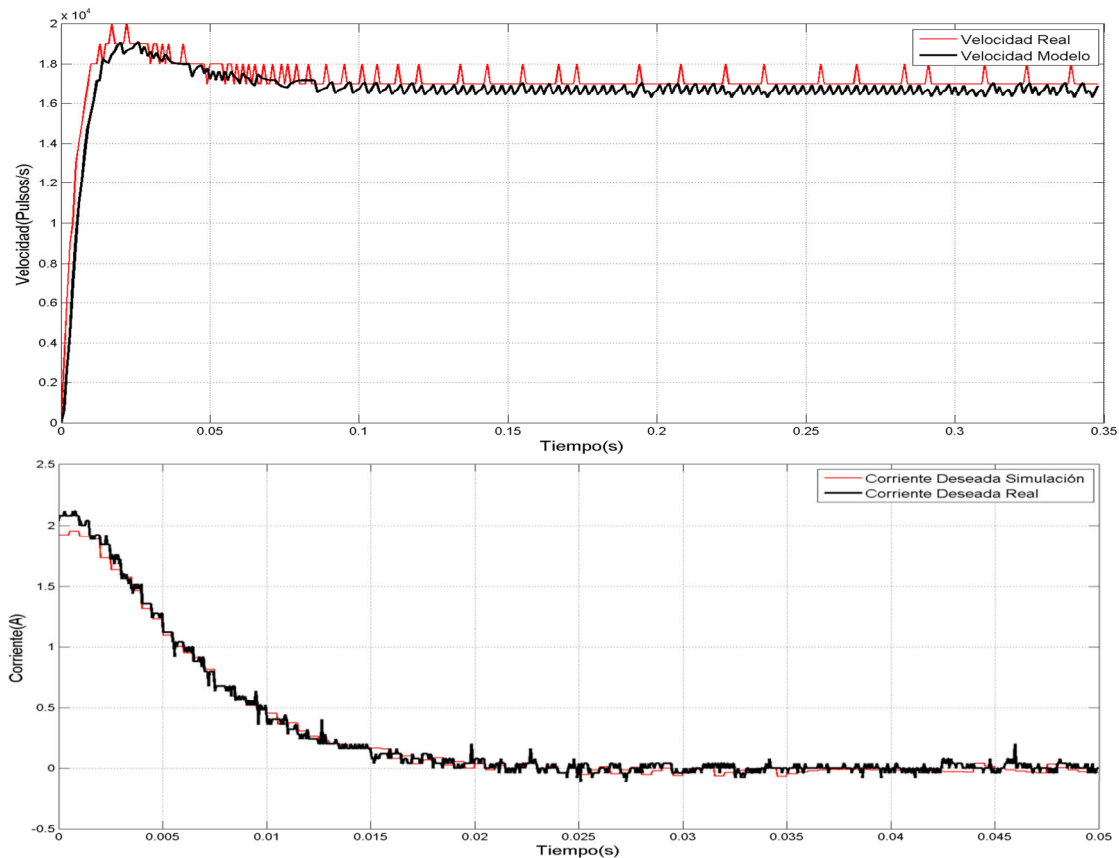


Figura IV.4. Comparación de la respuesta al escalón del controlador de velocidad.

IV.1 Controlador P-PI de posición

A continuación se muestra la respuesta al escalón del controlador en modo posición utilizando una entrada escalón de 4000PPR presentado en la Figura III.64 y se compara con la respuesta de modelo simulado, el periodo de muestreo se mantuvo a 1ms, la señales adquiridas tienen un factor de correlación de $\rho = 0.93$, la ganancias del controlador son $K_{pw}=0.37$ y $K_{iw}=0.005$, y $K_{pt}=25.4$.

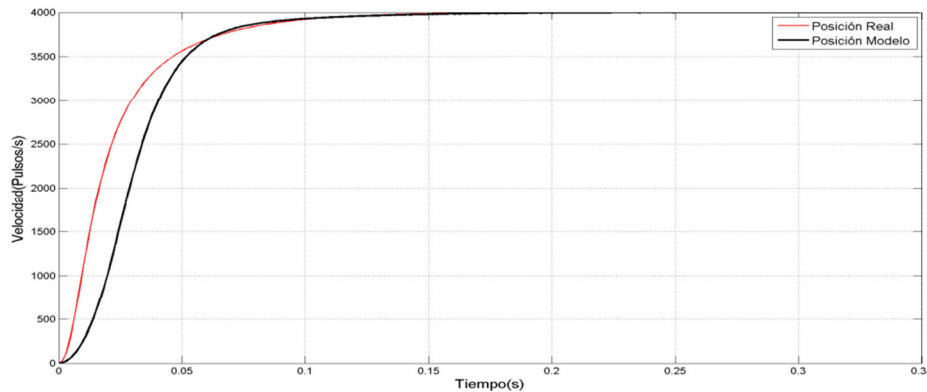
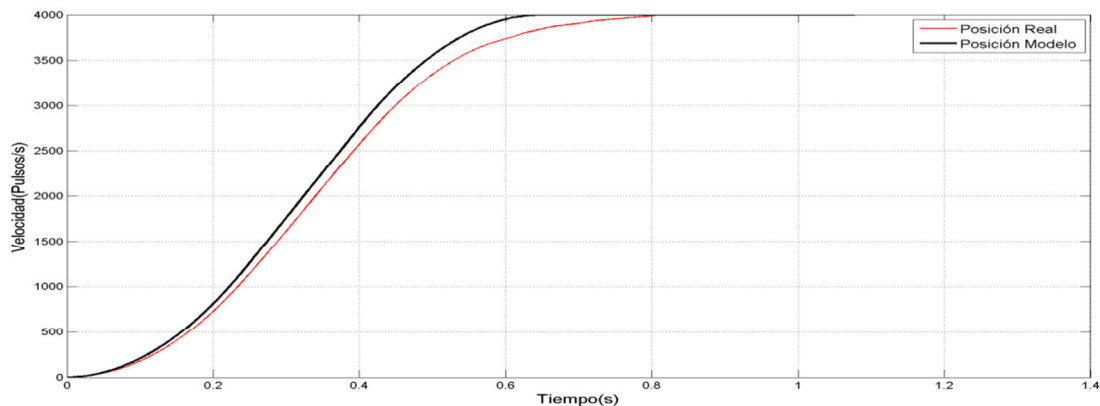


Figura IV.5. Comparación de la respuesta al escalón del controlador P-PI de posición.

A continuación se muestra la respuesta del controlador trabajando con el generador de perfiles de movimiento el cual genera una posición final de 4000PPR y una velocidad máxima de 10000 pulsos/s² , mismas condiciones que modelo presentado en la Figura III.65, las ganancias de controlador se mantuvieron iguales, las ganancias de pre alimentación son $K_{FF\omega} = 1$ y $K_{FF\epsilon} = 0.002$.



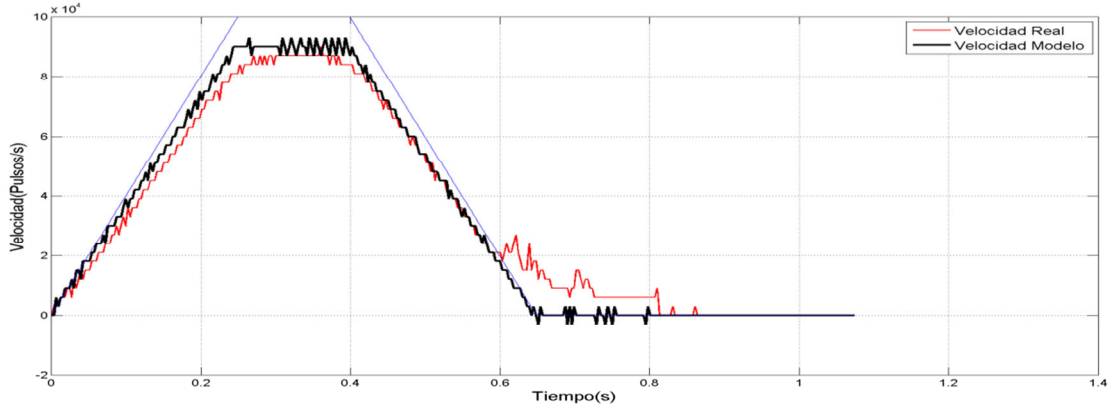


Figura IV.6. Comparación de la respuesta al escalón del controlador P-PI de posición.

Utilizando la misma metodología se aplicó el controlador de movimiento en el motor BLDC descrito en la tabla III.4, se selecciono el mismo método de sintonización con una frecuencia de cruce de 160rad/s y un margen de fase de 70°, a continuación se muestran los resultados de cada uno de los modos de operación de controlador de movimiento, los cuales son:

- Modo de velocidad
- Modo de posición con entrada escalón
- Modo de posición con generador de perfiles de movimiento

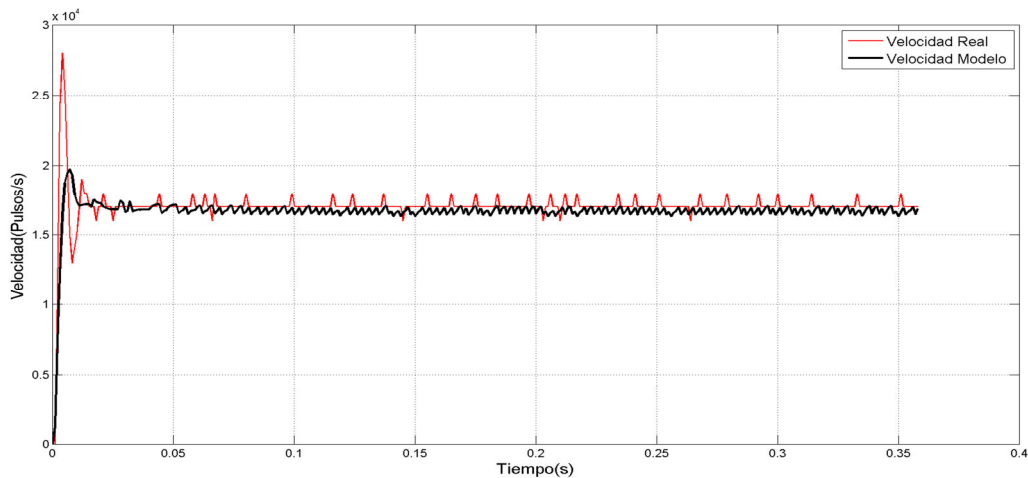


Figura IV.7. Respuesta del controlador en modo velocidad para el motor BLDC.

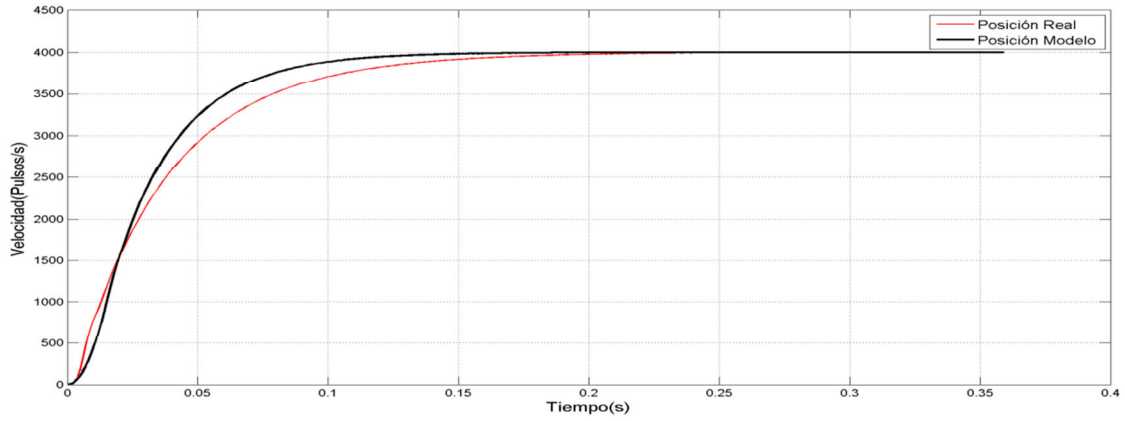


Figura IV.8. Respuesta del controlador en modo entrada escalon de posición para el motor BLDC.

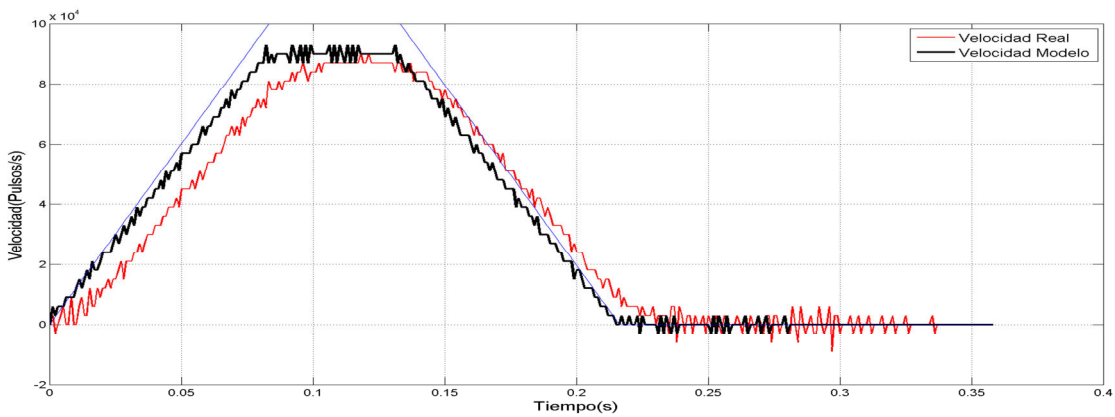
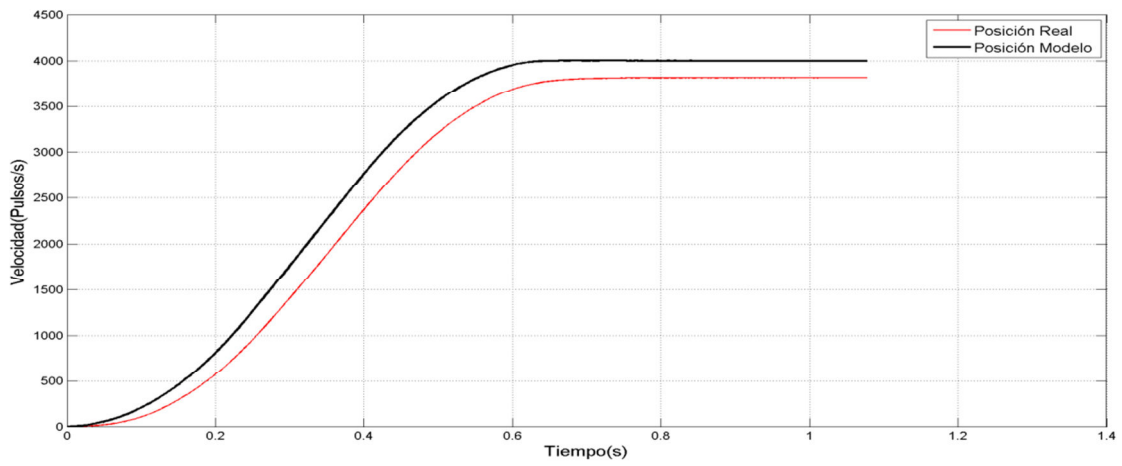


Figura IV.9. Respuesta del controlador para un perfil trapezoidal el motor BLDC.

Como se puede observar los resultados obtenidos la metodología planteada ofrece una buena representación del modelado de sistemas de control de movimiento donde se utilicen motores de DC la metodología planteada facilita el diseño, depuración, sintonización e implementación de un algoritmo de control de movimiento en un dispositivo FPGA, utilizando las herramientas de punto fijo de MATLAB/Simulink podemos identificar posibles perdidas y sobre flujos en la variables medidas y de control. Además, los cores genéricos para punto fijo descritos VHDL proporcionan una base para el desarrollo de algoritmos de control mas complejos o adecuados para una aplicación particular, como se menciona en el capítulo III existen diferentes tipos de topologías para el control de movimiento sin embargo muchas de ellas se basan en estrategias PID y sus variantes por lo que los cores descritos pueden ser usados para futuros trabajos. A continuación se muestran los resultados de síntesis del controlador propuesto, la síntesis se realizó contando con una licencia completa académica del software Quartus II 12.0 SP2 que incluye la licencia para el procesador NIOS II, el FPGA utilizado fue el Cyclone IV E número EP4CE115F29C7 el cual viene por defecto en la tarjeta de entrenamiento DE2-115.

Tabla IV.1. Resultados de síntesis de controlador de movimiento

Número de controladores	Nº PLLs	Nº Funciones combinaciones	Nº de Registros	Nº de Multiplicadores	Porcentaje de Utilización (114480)	Nº de bits de memoria
1	2	3992	2641	29/532	4,398 / (4 %)	95552
2	2	4985	3315	54/532	5,534 / (5 %)	95552
3	2	5,989	3989	79/532	6,667 / (6 %)	95552
4	4	6,985	4675	104/502	7,825 / (7 %)	95552
5	4	7,987	5383	129/502	9,000 / (8 %)	95552

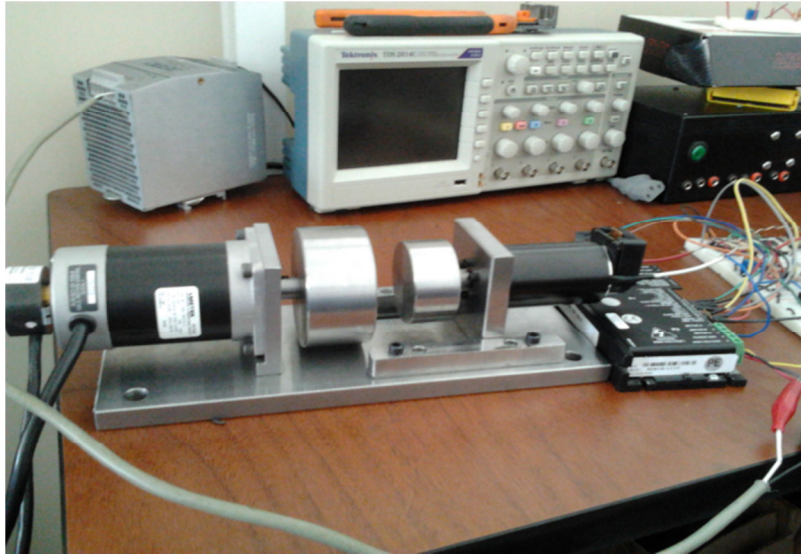


Figura IV.10. Estación de trabajo desarrollada.

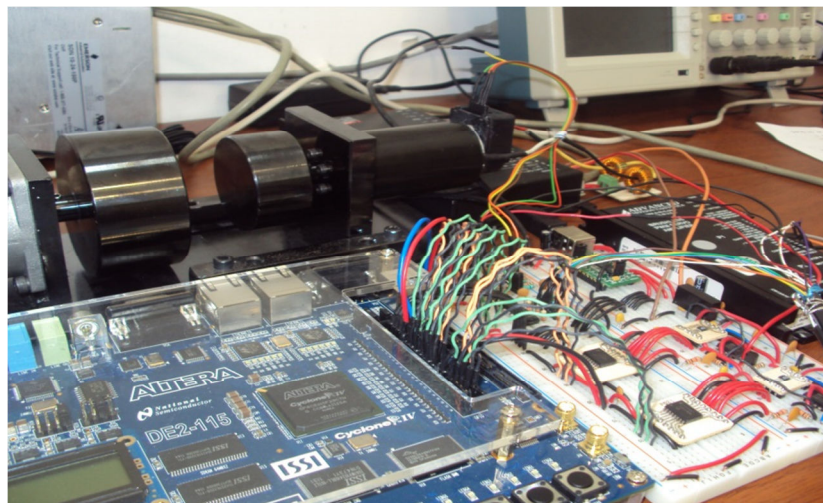


Figura IV.11. Prototipo del controlador de movimiento.

IV.2 Conclusiones

La metodología planteada para el desarrollo y descripción de un algoritmo de control en FPGA se llevo de manera satisfactoria pudiendo llevar a cabo su correcta implementación la cual mostro una clara similitud entre la teoría y la practica, o lo simulado y lo real una brecha que normalmente es considerada algo que no es de mucha utilidad o la cual no ayuda o no valida un algoritmo de control. La incursión en de la nuevas características del lenguaje de VHDL estándar 2008 ayudo a acelerar el diseño y verificación de los componentes descritos.

La ganancia de pre alimentación de velocidad y aceleración mostraron gran ayuda para evitar, por ejemplo la dinámica del control de posición y actuar directamente sobre el lazo interno de velocidad, así como reducir de manera significativa el error en el seguimiento de trayectorias, además, la ganancia de aceleración se anticipa en los puntos de donde se presenta el jerk en el perfil, el control en cascada P-PI feedforward que se presento presenta una robustez y rechazo frente a perturbaciones, además de la flexibilidad de operar en diferentes modos, lo cual es una característica que ofrecen una gran ventaja frente a los controladores comerciales con controladores PID clásicos, se diseño un sistema de detección de fallos en la unidad de retroalimentación la cual es critica en la seguridad de sistemas robóticos, característica con la cual no cuentan firmas comerciales como Galil.

Como se vio en la sección anterior el numero de elementos de control que pueden estar trabajando en paralelo en un FPGA depende de gran medida de la densidad de integración del chip que se utilizó, y esta limitada en gran medida por el número de multiplicadores que se utilicen, la metodología de co-diseño de utilizar soft-cores o procesadores embebidos dentro del FPGA, para tareas no criticas como monitoreo e interfaz ayudan a disminuir el tiempo de desarrollo, y ofrecen ventajas interesantes en el diseño de sistemas mas complejos que estén embebidos en un solo chip, lo cual ayuda a disminuir costos y el numero de componentes electrónicos utilizados.

BIBLIOGRAFÍA

A-M-C, ADVANCED Motion Controls. 2011. AN-007- Using a Brushed Motor with Brushless Drive. <http://www.a-m-c.com/>

Astarloa, A., Lázaro, J., Bidarte, U., Jiménez, J. y Zuloaga, A. 2009. “FPGA technology for multi-axis control systems”. *Mechatronics*. vol.19. pp. 258-268.

Colín, J. 2006. “Descripción VHDL de los Bloques Funcionales de un Controlador Digital PID”. Tesis de Maestría. Universidad Autónoma de Querétaro.

Contreras, C. A., Robles, G. y Carlos, J. 2001. “Control de movimiento PID digital para servomotor de DC 1.5KW”. Instituto Tecnológico de Chihuahua.

Monmasson, E. and Cirstea, M. 2007. “FPGA design methodology for industrial control systems—A review”. *IEEE Transactions on Industrial Electronics*. vol. 54. No. 4.

Bueno, E. J., Hernández, Á., Rodríguez, F. J., Girón, C., Mateos, R. y Cobreces, S. 2009. “A DSP- And FPGA-Based Industrial Control with High-Speed Communication Interfaces for Grid Converters Applied To Distributed Power Generation Systems”, *IEEE Transactions on Industrial Electronics*, vol. 56, No. 3.

Franco Gasca, L.A. 2007. “Desarrollo de un sistema de monitoreo para máquina-herramienta basada en FPGA”. Tesis de doctorado. Universidad Autónoma de Querétaro.

Fink, D. G. y Wayne, B. H., 2001. *Manual de ingeniería eléctrica*. Mc Graw Hill. 13ª Edición. México. ISBN 970-10-0871-5.

Galil, Motion Corporation. 2011. DMC-1866 (PCI 6-axis Accelera) Data Sheet. Galil Motion Corporation. www.galilmc.com.

Gu, G., Zhu, L., Xiong, Z. y Ding, H. 2010. "Design of a Distributed Multiaxis Motion Control System Using the Ieee-1394 Bus". IEEE Transactions on Industrial Electronics. vol. 57. No. 12.

González, J., Baeyens, E., Gayubo, F., Pérez J., Fraile, J. y García, F. 2004. "Desarrollo de un controlador abierto para un robot industrial tipo SCARA". Revista Iberoamericana de Automática e Informática Industrial. vol. 1. No. 1.

Hernández, D. 2005. "Métodos de Identificación y Control Adaptable Aplicados a una Tarjeta Controladora de Movimiento". Tesis de Maestría. Universidad Autónoma de Querétaro.

Hernández Vargas, M. 2006. "Diseño para implementación en hardware de un control PID". Tesis maestría. Universidad Autónoma de Querétaro.

Mentz, J. 2000." Motion Control Theory Needed in the Implementation of Practical Robotic Systems". Tesis de Maestría. Virginia Polytechnic Institute.

Morales V. 2010. "Diseño de plataforma hardware-software para el desarrollo de aplicaciones industriales basadas en FPGA". Tesis de Doctorado. Universidad Autónoma de Querétaro.

Cho, J. U., Le, Q. N. y Jeon, J. W. 2009. "An FPGA-Based Multiple-Axis Motion Control Chip", IEEE Transactions on Industrial Electronics. vol. 56, No. 3.

León, L. Y. B. 2009. "Optimización de referencias y control en los actuadores de un robot PUMA para el mejoramiento de su dinámica". Tesis de Maestría. Universidad Autónoma de Querétaro.

Nakamura, M., Goto, S. and Kyura, N. 2004. Mechatronic Servo System Control- Problems in Industries and their Theoretical Solutions. Springer-Verlag.

Mussoi, F. L. 2002. Máquinas eléctricas, Brasil.

Boggarpu, N. K. y Kavanagh, R. C. 2010. “New Learning Algorithm for High-Quality Velocity Measurement and Control When Using Low-Cost Optical Encoders”. IEEE Transactions On Instrumentation And Measurement. vol. 59. No. 3.

Aliane, N. 2010. “A Matlab/Simulink-Based Interactive Module For Servo Systems Learning”, IEEE Transactions on Education. vol. 53. No. 2.

Osornio, R. 2004. “Diseño y Construcción de una Tarjeta Controladora de 3 Ejes”. Tesis de Maestría. Universidad Autónoma de Querétaro.

Osornio, R. 2007. “Diseño de sistema de control para CNC de alta velocidad”. Tesis de Doctorado. Universidad Autónoma de Querétaro.

Romero, R. 2004. “Procesamiento de Señales para la Detección de Ruptura de Herramienta en Sistemas de Manufactura por Control Numérico Computarizado”. Tesis de Doctorado. Universidad Autónoma de Querétaro.

Ronquillo, G. 2002. “Controlador Digital para Servomotores sin Escobillas de Corriente Directa”. Tesis de Maestría. Universidad Autónoma de Querétaro.

Trejo Hernández, M. 2006. “Módulo de maquinado y monitoreo, aplicando control de difusión en un proceso de torneado”. Tesis maestría. Universidad Autónoma de Querétaro.

Visioli, A. y Legnani, G. 2002. “On the trajectory tracking control of industrial SCARA robot manipulators,” IEEE Trans. Ind. Electron. vol. 49, No. 1.

Shao, X., Sun, D. y Mills, J. K. 2006. “A new motion control hardware architecture with FPGA-based IC design for robotic manipulators”. IEEE Int. Conf. Robot. Autom.

Shao, X. y Sun, D. 2007. "Development of a New Robot Controller Architecture with FPGA-Based IC Design for Improved High-Speed Performance". IEEE Transactions On Industrial Informatics. vol. 3. No. 4.

Xilinx Corp. 2011. MicroBlaze™ soft processor core, Xilinx processor central, www.xilinx.com/products/design_resources/proc_central/index.htm.

Chan, Y. F., Moallem, M. y Wang, W. 2007. "Design and implementation of modular FPGA Based PID controllers". IEEE Transactions on Industrial Electronics. vol. 54. No. 4.

Tzou, Y. Y. y Kuo, T. S. 1997. "Design and implementation of an FPGA-based motor control IC for permanent magnet AC servo motors". in Proc. Int. Conf. Ind. Electron., Control, Instrum. vol. 2. No 3.

Lorenz, R. D., Lipo, T. A. y Novotny, D. W. 1994. "Motion Control with Induction Motors", Proceedings of the IEEE. vol. 82. No. 8.

Alahakoon S. 2000. "Digital Motion Control Techniques for Electrical Drives", Tesis Doctorado, School of Electrical Engineering and Information Technology, Stockholm.

Slemon, G.R. 1994. "Electrical Machines for Variable-Frequency Drives", Proceedings of the IEEE. vol. 82. No. 8.

Maxon Motor. 2011. "ADS 50/10, Servoamplifier pulsed (PWM) 4-Q-DC ", <http://shop.maxonmotor.com/ishop/article/article/201583.xml>.

Mohammed, M. et al. 2010. "Improved speed operation of sensorless BLDC motor drives using IIR digital filter". Power and Energy (PECon), 2010 IEEE International Conference.

Wilamowski, B. M. y Irwin, J. D. 2011, The Industrial Electronics Handbook - Control and Mechatronics, Taylor and Francis Group. 2da Edición. Cap. 35. pp. 3-8.

Kuo, B. C. y Golnaraghi, F. 2009. "Automatic Control Systems". Ninth edition, Prentice-Hall International, Inc.

Franklin, G.F., Powell, J.D. y Workman, M.L. 1997. "Digital Control of Dynamical Systems". Third Edition. Addison-Wesley Publishing Company, Inc.

Astrom, K.J. y Wittenmark, B. 2008. "Adaptive Control", Second Edition, Addison-Wesley Publishing Company.

Isidori, A. 1995. "Nonlinear Control Systems". Third Edition, Springer - Verlag London Ltd.

Morari, M. y Zafiriou, E. 1989. "Robust Process Control", Prentice-Hall International, Inc., Englewood Cliffs, N.J.

Grüne, L. y Pannek, J. 2011. "Nonlinear Model Predictive Control: Theory and Algorithms (Communications and Control Engineering)". Springer. 1 Edición.

Le-Huy, H. 1994. "Microprocessors and Digital IC's for Motion Control", Proceedings of the IEEE. vol. 82. No. 8.

Kollmorgen Motion Technologies Group. 2011. AN0051- Starting Up and Tuning the MMC-SD. <http://www.kollmorgen.com>

Delta-Tau Motion Control. 2011. PMAC PCI. <http://www.deltatau.com/Common/products/pci.asp?connectionStr=release&node=id102>

PMD Corp.. 2011. MC73110 Motor Control IC. http://www.pmdcorp.com/downloads/MC73110_Motor_Control_Chip_Datasheet.pdf

Krishnan, R. 2001. "Electric Motor Drives- Modeling, analysis, and Control". Prentice Hall.

Parker Hannifin, "Fundamentals of Servo Motion Control", Parker Motion. Electromechanical Automation Division.

Sandoval, X. Y., 2011, "Prótesis de mano". Tesis de Maestría. Universidad Autónoma de Querétaro.

Altayef, J. A. y Qun-xiong, Z. 2009. "Real -Time DC Motor Position Control by (FPID) Controllers and design (FLC) Using Labview Software Simulation". International Journal of Engineering Studies, vol 1, Num. 4. pp. 247-256.

George W. y Younkin, P.E. 2011, "ELECTRIC SERVO MOTOR EQUATIONS AND TIME CONSTANTS", Industrial Controls Consulting, Div. Bulls Eye Marketing, Inc.

Malek, M., Makys, P. y Stulrajter, M. 2011. "Feedforward Control Of Electrical Drives- Rules and Limits". Power Engineering And Electrical Engineering, vol. 9. Num. 1, pp. 35-42.

Ohm, Y. D. 1994 "PID and PDF Compensators for Motion Control, Industry Applications Society Annual Meeting. vol. 3. pp. 1923-1929.

Dessaint, L. A., Hebert, B. J., Le-Huy, H, y Cavuoti, G., "A DSP-based adaptive controller for a smooth positioning system". IEEE Transactions On Industrial Electronics. vol. 37. no. 5. pp. 372-377, 1990.

Ogata, K. 1998. Ingeniería de Control Moderna. Prentice-Hall, 3a. Edición, México.

Cano, J. J. 2010. Identificación de servomotores en robot PUMA mediante redes neuronales implementadas en FPGA. Tesis Maestria, UAQ.

Fadali, M. S. 2009. "Digital Control Engineering – Analysis and Design". Elsevier, 1era Edición. pp. 176-182.

Levine, S. W. 2011. "Control System Fundamentals", CRC Press Taylor & Francis Group, 2da Edición.

Bishop, R. H. 2008. "Mechatronic System Control, Logic, and Data Acquisition", CRC Press Taylor & Francis Group. 2da Edición. Cap 15. pp. 12-18.

Wilamowski, B. M. y Irwin, J. D. 2011. "The Industrial Electronics Handbook - Control and Mechatronics", Editorial Taylor and Francis Group. 2da Edición. Cap. 14 pp. 7-8.

Visioli, A. 2006. "Practical PID Control", Springer-Verlag London Limited, 1era Edición. pp. 13-15.

González, S. A. 2006. "Nuevas técnicas en el accionamiento de máquinas de movimiento incremental". Tesis de doctorado, Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Buso, S. and Mattavelli, P. 2006. "Digital Control in Power Electronics". Morgan & Claypool, 1era Edición, pp. 49-50.

Adams, J. 2003. "AN-1052 Using the IR217x Linear Current Sensing ICs". International Rectifier. <http://www.irf.com/>

Yokote, H., and Watanabe, K. 1990. "A hybrid digital and analog controller for DC and brushless servomotors". IEEE Transactions on Instrumentation and Measurement, Vol. 39 No 1, pp. 259–263.

Wang, S., Li, T. and Zhou, Z., 2007, "FPGA-Based Single-Chip Servo System for Five Axes Actuators", IEEE International Conference on Automation and Logistics, pp. 1630-1634.

Dessaint, L. A., Hebert, B. J., Le-Huy, H., & Cavuoti, G. 1990. "A DSP-based adaptive controller for a smooth positioning system". IEEE Transactions On Industrial Electronics, Vol. 37 No 5, pp. 372–377.

Ellis, G. 2004, Control System Design Guide, Elsevier Academic Press, 3era Edición, pp. 377-384.

Ashenden, P. J. 1990. The VHDL Cookbook. University of Adelaide South Australia, 1era Edición.

Wilson, P. 2007. Design Recipes for FPGAs. Newnes – Elsevier, 1era edición.

Baese, U. M. 2001. Digital Signal Processing with Field Programmable Gate Arrays. Springer-Verlag, 1era edición.

Smith, G. R., 2010. FPGAs 101. Newnes – Elsevier. 1era Edición.

Onwubolu, G., 2005, Mechatronics Principles and Applications, Elsevier's Science & Technology, 1st Edition, pp. 282-288.

Ashenden, P. J., 2008, Digital Design: An Embedded Systems Approach Using VHDL, Morgan Kaufmann Publishers & Elsevier, 1st Edition, pp. 201-203.

W. Kafir, 2011, VHDL 101 Everything you need to know to get started, Elsevier, 1st Edition, pp. 65.

Shetty, D. y Kolk, R. A., 2010, Mechatronics System Design, CENGAGE Learning, 2nd Edition, pp. 162-164.

Sul, S. K., 2011, Control of Electric Machine Drive Systems, John Wiley & Sons, Inc & IEEE Press series on power engineering, , 1st Edition, pp. 180-188.

Kubo, K., et al., 1990, Disturbance torque compensated speed observer for digital servo drives, Proceedings in IEEE-IECON, pp. 1182–1187.

Sul, S. K., 2011, Control of Electric Machine Drive Systems, John Wiley & Sons, Inc & IEEE Press series on power engineering, , 1st Edition, pp. 180-188.

Petrella, R., Tursini, M., Peretti, L. Y Zigliotto, M., 2007, Speed Measurement Algorithms for Low-Resolution Incremental Encoder Equipped Drives: a Comparative Analysis, International Aegean Conference on Electrical Machines and Power Electronics, ACEMP '07, pp.780 – 787.

Kavanagh, R. C., 2001, Performance analysis and compensation of M/T-type digital tachometers, IEEE Transactions on Instrumentation and Measurement, VOL. 50, NO. 4, pp. 965–970.

Faccio, M., Grande, P., Parasiliti, F., Petrella, R., y Tursini, M., 2004, An embedded system for position and speed measurement adopting incremental encoders. Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting., pp. 1192–1199.

M. Gahbler, K. Karwowski, The Position And Speed Measurement Module Implemented In FPGA Device, Proceedings Of Electrotechnical Institute, Issue 240, 2009 pp. 222- 230.

Boggarpu, N. K., y Kavanagh, R. C. 2010. New learning algorithm for high-quality velocity measurement and control when using low-cost optical encoders. IEEE Transactions on Instrumentation and Measurement, Vol. 59 No 3, pp. 565–574.

Tsai, M., y Chen, C. P. 2002. Design of a quadrature decoder/counter interface IC for motor control using CPLD. IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02, pp. 1936–1941.

Maxim -IC, 2001, Application Note 578- New RS-485 IC Increases System Reliability and Fault Detection in Motor-Control Circuits.

Paul Acarnley, 2007, Stepping Motor a Guide to Theory and Practice, The Institution of Engineering and Technology (IET), 4th Edition, pp. 114-115.

ANEXO 1

Todos los códigos descritos son muy extensos y cuentan con diferentes arquitecturas para una misma entidad, aquí se anexan algunos de ellos, se tiene contemplado publicar los cores de manera gratuita y libre a través de Opencores.org.

```
--By Ing. Fortino Mendoza Mondragón           Date:16/03/2011
--E-Mail: fmendoza711@gmail.com               Rev 1.0
--Description:
--A generic PI controller, applying the ZOH - Z Transform

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity PI is                                     --Entity
  generic (
    Fen: integer := 32;                          --Number of bits of the error input
    Fef: integer := 12;                          --Fractional part
    Fkn: integer := 18;                          -- Format of the Kp y Ki gains
    Fkf: integer := 10;
    Fun: integer := 16;
    Fuf: integer := 0
  );
  Port (
    CLK:   in std_logic:= 'U';                  --Clock
    ARST:  in std_logic:= 'U';                  --Asynchronous reset='0'
    Ts:    in std_logic:= 'U';                  --Sampling time pulse
    Error: in signed (Fen-1 downto 0) :=(others=>'U'); --Error input
    Kp:    in signed (Fkn-1 downto 0):=(others=>'U'); --Proportional Gain Kp
    Ki:    in signed (Fkn-1 downto 0):=(others=>'U'); --Integral Gain Ki
    Max:   in signed(Fen+Fkn-1 downto 0);       -- Format is the output of Error*Gains
    Min:   in signed(Fen+Fkn-1 downto 0);
    Finish: out std_logic:= '0';                --Finish
    Output: out signed (Fun-1 downto 0):=(others=>'0') --Control Variable (CV)
  );
end PI;

architecture RTL_PI_V1 of PI is
  signal T1s,T2s: std_logic;
  signal R1s,R2s,R3s,R4s,R6s: signed(Fen+Fkn-1 downto 0);
  signal R5s:      std_logic_vector(Fen+Fkn-1 downto 0);
  signal R7s,R8s: std_logic_vector(Fun-1 downto 0);
begin

  U0: entity work.FF_D_V1          port map(CLK,ARST,T1s,T2s);
  U1: entity work.Multiplier_Signed_V3 generic map(Fen,Fkn) port map(Error,Kp,R1s);
  U2: entity work.Multiplier_Signed_V3 generic map(Fen,Fkn) port map(Error,Ki,R2s);
  U3: entity work.Adder_Signed      generic map(Fen+Fkn) port map(R2s,signed(R5s),R3s);
  U4: entity work.Saturation         generic map(Fen+Fkn,Fef+Fkf,Fen+Fkn,Fef+Fkf)
                                     port map(CLK,ARST,Ts,Max,Min,R3s,T1s,R4s);
  U5: entity work.RegisterLD         generic map(Fen+Fkn) port map(CLK,ARST,T2s,std_logic_vector(R4s),R5s);
  U6: entity work.Adder_Signed      generic map(Fen+Fkn) port map(R1s,R4s,R6s);
  R7s<= std_logic_vector(R6s(R6s'LENGTH-1)&R6s((Fun-Fuf)+Fef+Fkf-2 downto (Fef+Fkf-Fuf)));
  U7: entity work.RegisterLD         generic map(Fun) port map(CLK,ARST,T1s,R7s,R8s);
  Output<=signed(R8s);
  Finish<=T2s;

end RTL_PI_V1
```

ANEXO 2

```

-----
--By Ing. Fortino Mendoza Mondragón      Date: 06/09/2012
--E-Mail: fmendoza711@gmail.com          Rev 1.0
--Description: Trapezoidal Motion Profile
--
-----

```

```

library ieee ;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

```

```

entity Trapezoidal_Profile is          --Entity Trapezoidal_Profile
  port (
    CLK:      in std_logic:= 'U';      --Clock
    ARST:     in std_logic:= 'U';      --Asynchronous reset='1'
    Start:    in std_logic:= 'U';      --Start Motion
    Tacc:     in signed(41 downto 0):=(others=>'U');      -- Time in Micro seconds
    Tvel:     in signed(41 downto 0):=(others=>'U');      -- Time in Micro seconds
    Acc:      in signed (22 downto 0):=(others=>'U');      -- Acceleration value
    Post:     in signed (31 downto 0):=(others=>'U');      -- Final Position value
    Acceleration: out signed(22 downto 0):=(others=>'0'); --Acceleration in counts/sec^2
    Velocity:  out signed(36 downto 0):=(others=>'0');    --Velocity in counts/sec
    Position:  out signed(31 downto 0) :=(others=>'0')    --Number of counts/rev
  );
end Trapezoidal_Profile;

```

```

-----
architecture RTL_Trapezoidal_Profile_V1 of Trapezoidal_Profile is
  -- Build an enumerated type for the state machine
  type Count_State is (Zero, One, Two, Three, Four);

```

```

  -- Registers to hold the current state and the next state
  signal Present_State, Next_State      : Count_State;

```

```

  -- Attribute to declare a specific encoding for the states
  attribute Syn_Encoding: string;
  attribute Syn_Encoding of Count_State : type is "000 001 010 011 100";
  signal Count: signed(Tacc'length-1 downto 0);
  signal Accs1,Aux3: signed(Acceleration'length-1 downto 0);
  signal

```

```

LD1,LD2,LD3,LD4,LD5,LD6,ENA1,SRST1,SRST2,SRST3,SRST4,SRST5,SRST6,Ts1,S1,S2,Enable,Ts,Aux1,Dly1: std_logic;
  constant Aux: std_logic_vector(Acceleration'length-1 downto 0):=(others=>'1');
  signal Reg1,Reg2,Vels1: signed(36 downto 0):= (others=>'0');
  signal Reg3,Poss1: signed(51 downto 0):= (others=>'0');
  signal Reg4,Aux2,Reg5: signed(31 downto 0):= (others=>'0');
begin

```

```

  TB_1: entity work.TB_Ascending generic map(10,999) port map(CLK, ARST, Enable, Ts);

```

```

  Acceleration<=Accs1;
  Velocity<= Reg2;
  Position<=Reg4;

```

```

  FF_D1:process(ARST,CLK)
  begin
    if ARST='1' then
      Ts1<='0';
    elsif rising_edge(CLK) then
      Ts1<=aux1;
    end if;
  end process;

```

```

        end if;
    end process FF_D1;

FF_D2:process(ARST,CLK)
begin
    if ARST='1' then
        Dly1<='0';
    elsif rising_edge(CLK) then
        Dly1<=S2;
    end if;
end process FF_D2;

process(Vels1,Ts)
begin
    if Vels1=0 then
        Aux1<='0';
    else
        Aux1<=Ts;
    end if;
end process;

LD2<=Ts1;
LD3<=Ts;
LD4<=Ts1;
LD5<=Ts or S2;
LD6<=Dly1;

process (Present_State, Start, Count, Tvel, Tacc)
begin
    case Present_State is
        when Zero => --Wait for start
            if (Start = '1') then
                Next_State <= One;
            else
                Next_State <= Zero;
            end if;
        when One => -- Load acceleration and wait Tacc
            if Tvel=0 then
                if (Count = Tacc-1) then
                    if(Tvel=0) then
                        Next_State <= Three;
                    else
                        Next_State <= Two;
                    end if;
                else
                    Next_State <= One;
                end if;
            else
                if (Count = Tacc) then
                    if(Tvel=0) then
                        Next_State <= Three;
                    else
                        Next_State <= Two;
                    end if;
                else
                    Next_State <= One;
                end if;
            end if;
        when Two=>
            if (Count = (Tacc+Tvel+1)) then -- Load acceleration and wait Tacc+Tv
                Next_State <= Three;
            end if;
    end case;
end process;

```

```

                else
                    Next_State <= Two;
                end if;
            when Three =>
                if Tvel=0 then
                    if (Count = ((Tacc sll 1)+Tvel-1)) then -- Load acceleration and wait
                        2*Tacc+Tv
                            Next_State <= Four;
                            else
                                Next_State <= Three;
                            end if;
                    elsif (Count = ((Tacc sll 1)+Tvel+2)) then -- Load acceleration and wait
                        2*Tacc+Tv
                            Next_State <= Four;
                            else
                                Next_State <= Three;
                            end if;
                when Four =>
                    Next_State <= Zero;
                end case;
            end process;

        process (Present_State)
            begin
                case Present_State is
                    when Zero=> --Wait for start signal
                        LD1 <= '0';
                        SRST1<='1'; --Reset counter
                        SRST2<='1'; --Reset register acceleration
                        SRST3<='1'; --Reset counter velocity
                        SRST4<='0'; --Reset counter position
                        SRST5<='1';
                        SRST6<='1';
                        ENA1<='0'; --Enable timer
                        Enable<='0';
                        S1<='0';
                    S2<='0';
                    when One => -- Load Acceleration and enable timer and wait for Tacc
                        LD1 <= '1';
                        SRST1<='0'; --Reset counter
                        SRST2<='0'; --Reset register acceleration
                        SRST3<='0'; --Reset counter velocity
                        SRST4<='0'; --Reset counter position
                        SRST5<='0';
                        SRST6<='0';
                        ENA1<='1';
                        Enable<='1';
                        S1<='0';
                    S2<='0';
                    when Two => -- Wait for Tvel
                        LD1 <= '1';
                        SRST1<='0'; --Reset counter
                        SRST2<='1'; --Reset register acceleration
                        SRST3<='0'; --Reset counter velocity
                        SRST4<='0'; --Reset counter position
                        SRST5<='0';
                        SRST6<='0';
                        ENA1<='1';
                        Enable<='1';
                        S1<='0';
                    S2<='0';
                end case;
            end process;
        end process;
    end architecture;

```

```

when Three =>    -- Wait for 2*Tacc+Tvel
    LD1 <= '1';
    SRST1<='0';           --Reset counter
    SRST2<='0';           --Reset register acceleration
    SRST3<='0';           --Reset counter velocity
    SRST4<='0';           --Reset counter position
    SRST5<='0';
    SRST6<='0';
    ENA1<='1';
    Enable<='1';
    S1<='1';
S2<='0';
when Four =>     -- Load Final Position
    LD1 <= '0';
    SRST1<='1';           --Reset counter
    SRST2<='1';           --Reset register acceleration
    SRST3<='1';           --Reset counter velocity
    SRST4<='0';           --Reset counter position
    SRST5<='1';
    SRST6<='1';
    ENA1<='0';
    Enable<='0';
    S1<='0';
S2<='0';
end case;
end process;

process(ARST,CLK)
begin
    if ARST='1' then
        Present_State <=Zero;
    elsif rising_edge(CLK) then
        Present_State <= Next_State;
    end if;
end process;

Timer:process (ARST,CLK)           --Microseconds counter
    variable cnt1    : signed(Tacc'length-1 downto 0):= (others=>'0');
begin
    if ARST = '1' then
        cnt1 := (others=>'0');           -- Reset the counter to 0
    elsif rising_edge(CLK) then
        if(SRST1 = '1') then
            cnt1 := (others=>'0');           -- Reset the counter to 0
        elsif ENA1 = '1' then
            cnt1 := cnt1 + 1;           -- Increment the counter if counting is enabled
        end if;
    end if;
    Count <= cnt1;           -- Output the current count
end process Timer;

FF1: process (ARST,CLK) begin --Acceleration register
    if ARST = '1' then
        Accs1<= (others=>'0');
    elsif rising_edge(CLK) then
        if(SRST2 = '1') then
            Accs1<= (others=>'0');
        elsif (LD1 = '1') then
            Accs1 <= Aux3; --Two complement
        end if;
    end if;
end process;

```

```

end process FF1;

process(S1,Acc)
begin
    if S1='0' then
        Aux3<=Acc;
    else
        Aux3<=signed(std_logic_vector(Acc) XOR Aux) + 1;
    end if;
end process;

FF2: process (ARST,CLK) begin -- Velocity accumulator
    if ARST = '1' then
        Reg1<= (others=>'0');
    elsif rising_edge(CLK) then
        if SRST6='1' then
            Reg1<= (others=>'0');
        elsif (LD2 = '1') then
            Reg1 <= Vels1;
        end if;
    end if;
end process FF2;

Intgr1:process (ARST,CLK) --Velocity Integer
    variable Int1 : signed(36 downto 0):= (others=>'0');
begin
    if ARST = '1' then
        Int1 := (others=>'0'); -- Reset the counter to 0
    elsif rising_edge(CLK) then
        if(SRST3 = '1') then
            Int1 := (others=>'0'); -- Reset the counter to 0
        else
            Int1 := resize(Accs1,37) + Reg1 ; -- Increment the counter if counting is enabled
        end if;
    end if;
    Vels1 <= Int1; -- Output the current count
end process Intgr1;

FF3: process (ARST,CLK) begin -- Velocity accumulator
    if ARST = '1' then
        Reg2<= (others=>'0');
    elsif rising_edge(CLK) then
        if SRST5='1' then
            Reg2<= (others=>'0');
        elsif (LD3 = '1') then
            Reg2 <= Vels1;
        end if;
    end if;
end process FF3;

FF4: process (ARST,CLK) begin -- Position accumulator
    if ARST = '1' then
        Reg3<= (others=>'0');
    elsif rising_edge(CLK) then
        if (LD4 = '1') then
            Reg3 <= Poss1;
        end if;
    end if;
end process FF4;

Intgr2:process (ARST,CLK) --Velocity Integer

```

```

variable Int2      : signed(51 downto 0):= (others=>'0');
begin
  if ARST = '1' then
    Int2 := (others=>'0');           -- Reset the counter to 0
  elsif rising_edge(CLK) then
    if(SRST4 = '1') then
      Int2 := (others=>'0');       -- Reset the counter to 0
    else
      Int2 := resize(Vels1,51) + Reg3 ;   -- Increment the counter if counting is enabled
    end if;
  end if;
  Poss1 <= Int2;                   -- Output the current count
end process Intgr2;

FF5: process (ARST,CLK)  begin      -- Position accumulator
  if ARST = '1' then
    Reg4<= (others=>'0');
  elsif rising_edge(CLK) then
    if (LD5 = '1') then
      Reg4 <= Aux2;
    end if;
  end if;
end process FF5;

process(S2,Poss1,Post,Reg5)
begin
  if S2='0' then
    Aux2<=Poss1(51 downto 20);
  else
    Aux2<=Post+Reg5;
  end if;
end process;

FF6: process (ARST,CLK)  begin      -- Position accumulator
  if ARST = '1' then
    Reg5<= (others=>'0');
  elsif rising_edge(CLK) then
    if (LD6 = '1') then
      Reg5 <= Reg4;
    end if;
  end if;
end process FF6;

end RTL_Trapezoidal_Profile_V1;

```

ANEXO 3

```

-----
--By Ing. Fortino Mendoza Mondragón
--E-Mail: fmendoza711@gmail.com
--Description: Enhanced QEI
--

```

```

Date:06/09/2012
Rev 1.0

```

```

-----
library ieee ;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

```

```

entity QEI is      --Entity QEI
  generic (
    N :integer:= 32;          -- Number of bits of the Encoder Unit
    Fdn :integer:= 32; -- Fixed Point Format of the Input Signal - D(k)
    Fdf :integer:= 0;
    Fk1n :integer:= 18;      -- Fixed Point Format of the Constant K1
    Fk1f :integer:= 5;      -- Recommended to be multiples of 9 for full use of multipliers
    Fk2n :integer:= 18;      -- Fixed Point Format of the Constant K2
    Fk2f :integer:= 15; -- Recommended to be multiples of 9 for full use of multipliers
    Fwn :integer:= 32;      -- Fixed Point Format of the Output Signal - W(k)
    Fwf :integer:= 5;
    Fyn :integer:= 27; -- Fixed Point Format of the Output Signal - Y(k)
    Fyf :integer:= 0
  );
  port (
    CLK      :      in std_logic:='U';      --Clock
    ARST     :      in std_logic:='U';      --Asynchronous reset='1'
    CHA      :      in std_logic:='U';      --Encoder channel A
    CHB      :      in std_logic:='U';      --Encoder channel B
    CHI      :      in std_logic:='U';      --Encoder channel I
    DS       :      in std_logic:='U';      --Decoding Selection 1X or 4X
    Ts       :      in std_logic:='U';      --Sampling period
    FA,FB,FI,FD: in std_logic:='U'; --Failure detection
    Fault:    out std_logic:='U';      --Failure alarm
    Ready:    out std_logic:='U';      --Data Ready
    Status:   out std_logic_vector(3 downto 0):=(others=>'0'); --Status of the encoder signals
    Velocity: out signed(Fyn-1 downto 0):=(others=>'0'); --Velocity in counts/sec
    Position: out signed(N-1 downto 0) :=(others=>'0') --Number of counts/rev
  );
end QEI;

```

```

-----
architecture RTL_QEI_V1 of QEI is
  constant K1: signed(Fk1n-1 downto 0):="000101000000010011"; --000101000000010011
  constant K2: signed(Fk2n-1 downto 0):="000101011100000000"; --000101011100000000
  signal CHAs,CHBs,CHIs: std_logic:='0';
  signal Faults,UpDown_1X,UpDown_4X,Count_1X,Count_4X,Counts,UpDowns:std_logic:='0';
  signal Count_1Xs,UpDown_1Xs:std_logic:='0';
  signal Counters:signed(N-1 downto 0):=(others=>'0');
  signal Positions: std_logic_vector(N-1 downto 0):=(others=>'0');
  signal FaultReg: std_logic_vector(Status'length-1 downto 0):=(others=>'0');

```

```

begin

```

```

  Faults<= FA or FB or FI or FD;          -- General Fault
  Fault<=Faults;
  FaultReg<=FD&FI&FB&FA;                -- Code for Faults inputs

```

```

  S1: entity work.RegisterLD      generic map(Status'length) port map(CLK,ARST,Faults,FaultReg,Status);
- Fault alarm's register

```



```

-- Digital Filters
F1: entity work.Digital_Filter(RTL_DF_V1) port map(CLK,ARST,CHA,CHAs);
F2: entity work.Digital_Filter(RTL_DF_V1) port map(CLK,ARST,CHB,CHBs);
F3: entity work.Digital_Filter(RTL_DF_V1) port map(CLK,ARST,CHI,CHIs);
-- Quadrature Decoder Unit for 1X y 4X
U1: entity work.QuadratureDecoder_1X(Behavior_QuadratureDecoder_1X_V1)
    port map(CLK,ARST,CHAs,CHBs,Count_1X,UpDown_1X);
U2: entity work.Rising_Edge_Detector    port map(CLK,ARST,Count_1X,Count_1Xs);
U3: entity work.Rising_Edge_Detector    port map(CLK,ARST,UpDown_1X,UpDown_1Xs);
U4: entity work.QuadratureDecoder_4X(RTL_QuadratureDecoder_4X_V3)
    port map(CLK,ARST,CHAs,CHBs,Count_4X,UpDown_4X);
U5: entity work.MUX2_1                  port map(UpDown_1Xs,not UpDown_4X,DS,UpDowns);
U6: entity work.MUX2_1                  port map(Count_1Xs,Count_4X,DS,Counts);
-- Counter
U7: entity work.UpDown_Counter generic map(N) port map(CLK,ARST,UpDowns,Counts,Counters);
U8: entity work.RegisterLD              generic map(N)
    Port map(CLK,ARST,Ts,std_logic_vector(Counters),Positions);

Position<=signed(Positions);

U9: entity work.Derivative_LPF generic map(Fdn,Fdf,Fk1n,Fk1f,Fk2n,Fk2f,Fwn,Fwf,Fyn,Fyf)
    port map(CLK,ARST,Ts,K1,K2,Counters,Ready,Velocity);

end RTL_QEI_V1;
-----
configuration QEI_CFG of QEI is
    for RTL_QEI_V1
        end for;
end QEI_CFG;

```