



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA ELECTRÓNICA**

**MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA,  
OPCIÓN EN AUTOMATIZACIÓN**

**“MODELADO Y CONTROL DE UN ROBOT  
CARTESIANO DE DOS GRADOS DE LIBERTAD  
REALIZANDO ESTUDIOS DE JERK”**

## **T E S I S**

Presentada para obtener el título de:  
**Maestro en Ciencias de la Electrónica, Opción en Automatización**

Presenta:

**Lic. Luis Arturo Alcantar Vergara\*.**

Directores:

Dra. María Aurora Diozcora Vargas Treviño (FCE-BUAP)

Dra. Amparo Dora Palomino Merino (FCE-BUAP)

Dr. Jesús López Gómez (DAIA-UJAT)

Puebla, México

Octubre 2023

\*Becario CONAHCYT 1073243

**BUAP**®

## **Agradecimientos**

A mis Asesores la Dra. María Aurora Diozcora Vargas Treviño la directora de tesis, Dra. Amparo Dora Palomino Merino asesora de tesis y el Dr. Jesús López Gómez asesor externo que me dieron todo para que este trabajo se pudiera realizar.

A mi Jurado compuesto por la Dra. Olga Guadalupe Félix Beltrán, Dr. Félix González Canales, Dra. Alina Santillán Guzmán, que me hicieron las correcciones que necesitaba para poder llegar a ser un trabajo de calidad.

A la Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica, que ha sido mi casa de estudios permitiéndome formarme y dándome la oportunidad de seguirme desarrollando.

A mi familia que son lo que me sostiene y da fuerzas para avanzar.

A CONAHCYT que otorga el apoyo económico para poder realizar investigación y estudios que de otra manera no sería posibles ayudando no solo al desarrollo de estudiantes como yo sino el de las ciencias en el país dando la oportunidad de un mejor futuro.

### **Dedicatorias**

A mi familia que me dio la oportunidad y el apoyo para que este trabajo sea posible, a mis amigos que me dieron el impulso que necesitaba para realizarla y todos los que me ayudaron a lo largo de estos años.

## Resumen

En la actualidad la robótica es un elemento clave para el desarrollo industrial de un país, casi todos los procesos automatizados utilizan algún tipo de robot. La Ingeniería de Control es una disciplina que se enfoca en modelar matemáticamente una gran cantidad de sistemas dinámicos y el diseño de controladores que harán que estos sistemas se comporten de la manera deseada. En esta tesis se trabajó con un robot cartesiano de 2 grados de libertad. El modelo matemático consiste en la representación del sistema como ecuaciones que describen su comportamiento. Se propone un modelado usando la metodología parámetros agrupados, con el objetivo de tener un sistema más detallado. En esta propuesta se utilizará la ecuación de Lagrange basado en el análisis de las energías cinética y potencial de un sistema rígido para obtener el modelo dinámico del sistema mecánico. Con el modelo dinámico de la transmisión mecánica y el modelo eléctrico del motor se obtendrá el modelo matemático general para evaluar mediante simulaciones la respuesta dinámica del sistema. El sistema de posicionamiento consiste en aplicar un voltaje o torque modulado por ancho de pulso (PWM) controlado en la entrada del motor de DC en el que se agrupa a un reductor, un acoplamiento, un eje de tornillo de bola, una tuerca y el bloque deslizante. El Jerk también conocido como sobre aceleración es la 3ra derivada de la posición, representa el cambio en la aceleración, al no tratarse de las medidas principales de la mecánica de Newton tampoco tiene dispositivos para medirse, aun así, se ha desarrollado trabajos donde el Jerk forma parte fundamental funcionando el control a partir de reducir al mínimo el valor calculado de él. Se tiene una plataforma experimental que consta de un sistema embebido utilizando una tarjeta FPGA DE0-CV, se emplea el lenguaje de descripción de hardware (AHDL) para el diseño del firmware, así como el “lenguaje D”. Se realizó una trayectoria senoidal que oscila alrededor de los 90mm con una amplitud de 45mm y un periodo de 100s, se emplean encoders de cuadratura acoplados a los motores para recopilar la información, que será procesada por el FPGA y enviada a la interfaz de Labview, además se empleó el método MT para estimar el valor de la velocidad y aceleración en el FPGA, para todas las lecturas del sistema se usó un filtro tipo Bessel para cada variable.

# Índice

Índice de Figuras .....	3
Índice de Tablas .....	5
Introducción .....	6
<b>CAPÍTULO 1: Robot cartesiano de dos grados de libertad</b> .....	13
1.1 Descripción del robot .....	13
1.2 Diagrama a bloques general del sistema .....	14
1.3 Sistema embebido .....	15
<b>1.3.1 Hardware</b> .....	17
<b>1.3.2 Lenguaje D</b> .....	17
1.4 Conclusiones .....	19
<b>CAPÍTULO 2: Modelado dinámico del robot</b> .....	20
2.1 Metodología de parámetros agrupados .....	20
2.2 Modelo dinámico de un grado de libertad .....	23
2.3 Modelo dinámico de dos grados de libertad .....	26
2.4 Controlador .....	30
2.5 Conclusiones .....	31
<b>CAPÍTULO 3: Desarrollo de la electrónica para el control del robot.</b> .....	32
3.1 Sistema de adquisición de datos .....	32
3.2 Diseño de la etapa de potencia .....	39
3.3 Comunicación entre el robot y el sistema embebido .....	42
3.4 Diseño del firmware .....	47
3.5 Diseño del software .....	50
3.6 Caracterización de motores .....	58
3.7 Conclusiones .....	62
<b>CAPÍTULO 4: Metodología para la obtención del Jerk</b> .....	63
4.1 Efecto de muestreo .....	63
4.2 Filtro .....	66
4.3 Error de cuantificación .....	70
4.4 Método MT .....	71
4.5 Obtención de la posición y velocidad .....	72
4.6 Obtención de la aceleración y Jerk .....	76
4.7 Métrica .....	79

4.8 Conclusiones.....	79
<b>CAPÍTULO 5: Resultados</b> .....	<b>80</b>
5.1 Resultados en el eje X .....	80
5.2 Resultados en el eje Z.....	87
5.3 Conclusiones.....	92
<b>Conclusiones Generales.</b> .....	<b>93</b>
<b>Bibliografía.</b> .....	<b>94</b>
<b>Apéndice 1: Artículo publicado</b> .....	<b>97</b>
<b>Apéndice 2: Constancia de participación de congreso</b> .....	<b>104</b>

# Índice de Figuras

Figura 1.I: Elementos de un sistema feed-drive [5] .....	7
Figura 2.I: Utilización de dispositivos en publicaciones de la IEEE .....	9
Figura 1: Robot cartesiano con 2 grados de libertad .....	13
Figura 1.2.1: Diagrama a bloques general del sistema de control para el Robot cartesiano de 2 GDL .....	15
Figura 1.3.1: Base de FPGA y tarjetas acopladas, tarjeta de comunicación con modulo Wifly (verde), Puentes H (azul) y Tarjeta De Desarrollo FPGA Cyclone V Intel DE0-CV (amarillo), fuente de alimentación (naranja) .....	17
Figura 2.1.1: Diagrama de pasos para aplicación de la metodología de parámetros agrupados.....	20
Figura 2.1.2: Punto de masa.....	20
Figura 2.1.3: Punto de Inercia.....	21
Figura 2.1.4: Inductor.....	21
Figura 2.1.5: Resorte.....	21
Figura 2.1.6: Resorte torsional.....	22
Figura 2.1.7: Amortiguador.....	22
Figura 2.1.8: Amortiguador rotacional.....	22
Figura 2.2.1: Esquema del eje Z.....	23
Figura 2.3.1: Diagrama a bloques.....	27
Figura 2.4.1: Lazo cerrado para el control de posición de los ejes.....	31
Figura 3.1.1: Diagrama del PWM.....	32
Figura 3.1.2: Diagrama de tiempos del PWM.....	33
Figura 3.1.3: Puente H en funcionamiento, a) sentido horario, b) sentido antihorario.	33
Figura 3.1.4: Diagrama de flujo del puente H.....	34
Figura 3.1.5: Señales A y B del encoder.....	35
Figura 3.1.6: Diagrama de tiempos, donde se muestra la recuperación de la señal de reloj a partir de las señales A y B de un encoder.....	35
Figura 3.1.7: Circuito para recuperación del Reloj.....	36
Figura 3.1.8: Diagrama de estados del decoder.....	37
Figura 3.1.9: Circuito de switch.....	37
Figura 3.1.10: Diagrama del FPGA.....	38
Figura 3.2.1: Diagrama general del puente H.....	39
Figura 3.2.2: Diagrama eléctrico del puente H.....	41
Figura 3.2.3: PCB del puente H.....	41
Figura 3.2.4: Tarjeta implementada puente H.....	42
Figura 3.3.1: Capas del protocolo 802.11 y su equivalencia en el protocolo TCP/IP...	42
Figura 3.3.2: Medidas del módulo Wifly.....	44
Figura 3.3.3: Diagrama eléctrico de la tarjeta de comunicación Wifi.....	45
Figura 3.3.4: PCB terminado denominado tarjeta de comunicación Wifi.....	46
Figura 3.4.1: Firmware del sistema.....	47
Figura 3.4.2: Detector de posición.....	48
Figura 3.4.3: Cálculo de Delta T.....	48
Figura 3.4.4: ADC y fines de carrera.....	49

Figura 3.4.5 MUX y CPU.....	49
Figura 3.4.6: DEC PWM, generador de PWM y circuito de seguridad.....	50
Figura 3.5.1: Diagrama de flujo del software.....	51
Figura 3.5.2: Diagrama de flujo de regreso a casa.....	53
Figura 3.5.3: Diagrama de flujo de lectura y aplicación de comandos.....	54
Figura 3.5.4: Diagrama de flujo del Control X.....	54
Figura 3.5.5: Compilador de lenguaje de programación de alto nivel a lenguaje máquina.....	55
Figura 3.5.6: Interfaz de usuario pestaña 1.....	56
Figura 3.5.7: Interfaz de usuario pestaña 3, subpestaña 1.....	56
Figura 3.5.8: Interfaz de usuario pestaña 3, subpestaña 2.....	57
Figura 3.5.9: Intefaz de usario pestaña 3, subpestaña 3.....	57
Figura 3.6.1: Diagrama de caracterización del motor.....	59
Figura 3.6.2: Grafica de comparación entre ciclo de trabajo y torque del motor x y una función lineal.....	60
Figura 3.6.3 Grafica de comparación entre ciclo de trabajo y torque del motor y y una función lineal.....	62
Figura 4.1.1: Delta de Dirac.....	63
Figura 4.1.2: Tren de delta de Dirac.....	63
Figura 4.1.3: El muestreo de una señal $f(t)$ es igual a multiplicar la función $f(t)$ por el peine de Dirac. (a) Señal $f(t)$ , (b) Peine de Dirac y (c) Señal muestreada.....	64
Figura 4.1.4: Representación gráfica de la función $f(t)$ y $Comb(t)$ con sus respectivas transformadas de Fourier. (a) $f(t)$ , (b) $Comb(t)$ , (c) Transformada de Fourier de la función $f(t)$ y (d) Transformada de Fourier de la función $Comb(t)$ .....	65
Figura 4.1.5: Resultados de la convolución entre $F(\omega)$ con el peine de Dirac $Comb(\omega)$ .	65
Figura 4.2.1: Señal con recorte de filtro a), Señal filtrada b).....	66
Figura 4.2.2 Señal muestreada (azul), Bessel (verde), Tchebyshev rizo 1 hz(rojo), Butterworth (negro).....	67
Figura 4.2.3: Diagrama de flujo del filtro.....	70
Figura 4.3.1: Muestreo de posición por encoder y FPGA.....	71
Figura 4.4.1: Posición (negro), Posición encoder (azul), Posición Muestreada (verde), velocidad MT (rojo).....	72
Figura 4.5.1: Diagrama de flujo de contador entre flancos.....	73
Figura 4.5.2: Posición ideal muestreada a 200Hz con resolución de 0.0004222972973.	74
Figura 4.5.3: Diagrama de flujo de cálculo de velocidad.....	75
Figura 4.6.1: Torque vs velocidad.....	76
Figura 4.6.2: Simulación del método MT para la aceleración.....	78
Figura 4.6.3: Aceleración por método MT sin filtro (azul), aceleración después de filtro de .2Hz (negro).....	78
Figura 5.1.1: Posición del eje X: a) simulada con parámetros agrupados, b) resultados experimentales, c) comparación de error.....	81
Figura 5.1.2: Velocidad: a) simulada, b) resultados experimentales.....	82
Figura 5.1.3: Torque del eje X sin compensar del controlador.....	83
Figura 5.1.4: Torque sin compensar vs Velocidad del eje X.....	83
Figura 5.1.5: Torque del controlador del eje X: a) simulado por parámetros agrupados, b) resultado experimental .....	84
Figura 5.1.6: Torque compensado vs velocidad de eje X.....	84

Figura 5.1.7: Aceleración del eje X: a) simulada b) experimental .....	86
Figura 5.1.8: Jerk del eje X: a) simulado, b) experimental, c) obtenido previo a la investigación .....	87
Figura 5.2.1: Posición del eje Z: a) simulada por parámetros agrupados, b) experimental, c) comparación de error en Z .....	87
Figura 5.2.2: Velocidad en el eje Z: a) simulada en base a modelado por parámetros agrupados, b) resultado experimental .....	88
Figura 5.2.3: Torque enviado por el controlado en eje Z sin compensar.....	89
Figura 5.2.4: Torque vs velocidad en eje Z sin compensar. ....	89
Figura 5.2.5: Torque vs velocidad en eje Z sin compensar.....	90
Figura 5.2.6: Torque compensado vs Velocidad en eje Z.....	90
Figura 5.2.7: Aceleración en eje Z: a) simulada, b) experimental.....	91
Figura 5.2.8: Jerk en eje Z: a) simulado, b) experimental, c) obtenido previo a la investigación.....	92

## Índice de Tablas

Tabla 1.1.1: Características del robot cartesiano.....	14
Tabla 1.3.1 Distribución de puertos de entrada.....	16
Tabla 1.3.2 Distribución de puertos de salida.....	16
Tabla 1.3.3 Recursos disponibles y mapeo de memoria RAM.....	16
Tabla 1.3.4: Instrucciones de Lenguaje D.....	17
Tabla 2.2.1: Variables y constantes del sistema.....	24
Tabla 3.1.1: Casos del Decoder.....	34
Tabla 3.3.1: Variantes de IEEE 802.11[34].....	43
Tabla 3.3.2: Puertos de entrada.....	46
Tabla 3.3.3: Puertos de entrada y salida con bus de datos.....	47
Tabla 3.5.1: Tabla de lectura de Bits.....	52
Tabla 3.5.2: Lectura de parámetros y RAM asignadas.....	52
Tabla 3.6.1: Frecuencias de prueba en el motor X.....	59
Tabla 3.6.2: Tabla de ciclo de trabajo del motor X.....	59
Tabla 3.6.3: Frecuencias de prueba en motor Y.....	60
Tabla 3.6.4: Tabla de ciclo de trabajo del motor Y.....	61

# Introducción

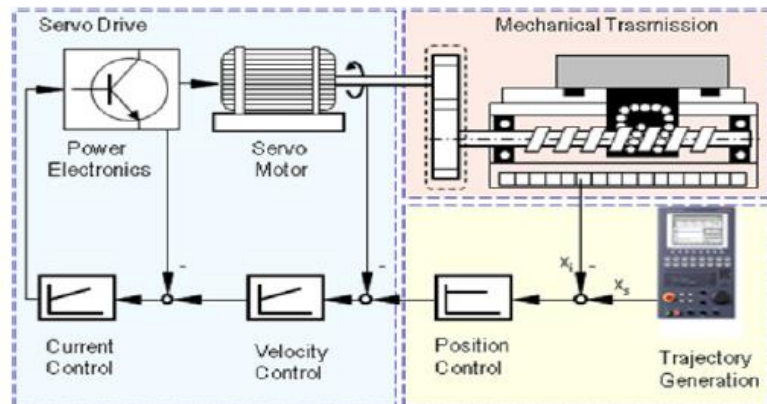
La robótica es esencial para el desarrollo industrial de un país, casi todos los procesos automatizados utilizan algún tipo de robot. Para lograr el acceso a un robot se requiere ser capaz de abrir la arquitectura de estos y esto representa trabajos de un alto grado de dificultad, utilizan varios sistemas, el electromecánico que representa la estructura mecánica y los motores, el electrónico que permite el manejo de las señales eléctricas que dan acceso a conocer sus variables físicas de posición, velocidad y par aplicado para cada motor, por último el software de control que se utiliza para programar con el fin de que realice una serie de instrucciones. Los robots industriales se usan para mejorar la productividad debido a que pueden realizar tareas monótonas y complejas sin errores de operación. Asimismo, puede trabajar en un ambiente imposible para operadores humanos.

Tras los primeros autómatas, los progenitores más directos de los robots fueron los tele-manipuladores. La evolución de los robots industriales desde sus comienzos ha sido vertiginosa. En poco más de 30 años las investigaciones y desarrollos acerca de robótica industrial han dado paso a que los robots tomen lugares en casi todas las áreas productivas y tipos de industria. En fábricas de diferentes tamaños, los robots pueden sustituir al hombre en aquellas áreas repetitivas y hostiles, adaptándose instantáneamente a los cambios de producción solicitados por los cambios en la demanda. La robótica tiene como objetivo a futuro aumentar su movilidad, destreza y autonomía de sus acciones. La mayoría de los robots actualmente poseen base estática, y se utilizan en aplicaciones industriales tales como ensamblado, soldadura, alimentación de máquinas herramientas, etc. El robot industrial es una máquina con características significativas de versatilidad y flexibilidad [1].

El robot de dos grados de libertad en configuración cartesiana se desarrolla por la necesidad de tener un dispositivo capaz de hacer un estudio espacial puntual con gran precisión. Se plantea de propósito general, ya que no solo se pretende usar en una sola aplicación, sino que puede ser modular. A la parte interna del robot de dos grados de libertad se le conoce por sus siglas en inglés como “*feed-drive*” consta de un motor eléctrico, un reductor, un controlador y los elementos de transmisión mecánica [1]; se muestra un ejemplo ilustrativo en la figura 1.

Con el objetivo de estudiar los sistemas de forma matemática se realiza el modelado para ello se puede hacer sus de diversas metodologías, una de ellas es la de parámetros agrupados. En esta se realiza una transformación de todos los cambios en el sistema a representaciones en energía cinética, potencial y disipativa, sin importar la fuente de la energía, es decir, si es eléctrica, mecánica o térmica, con la que se esté trabajando [2]. Es posible emplearla para tener un modelo general con varios tipos de energías, de forma similar a los parámetros conjuntos este método representa cada cambio por separado y después lo reúne, simplificando cada fenómeno, pero este último no es funcional para todos los tipos de energías y es más útil para analizar una estructura. Ya se ha comprobado que la metodología de parámetros agrupados es capaz de representar de forma fiable la mecánica de un robot cartesiano como fue reportado en el Laboratorio de Robótica y Control en la BUAP [3].

Los elementos de transmisión mecánica comprenden todas las partes de la máquina que se encuentran en el flujo de transmisión de par (potencia) entre el servomotor y la pieza de trabajo. En las diferentes versiones del diseño, los siguientes elementos de transmisión mecánica se usan con mayor frecuencia tales como: embragues, unidades de tuerca y husillo de bolas, unidades de cremallera y piñón, rodamientos, engranajes, cajas de engranajes (planetarios, cicloidales, armónicos), engranajes de correa dentada, guías, etc. El propósito principal del feed-drive es trasladar el efector de la máquina herramienta (mesa de trabajo, unidad de herramienta, unidad de husillo, etc.) a la posición deseada a través de los ejes de la máquina. Se requiere un feed-drive separado para cada eje de la máquina, aunque en general, los feed-drives tienen una estructura cinemática muy simple, el diseño óptimo es un problema que consiste en la selección de servomotores óptimos y elementos de transmisión mecánica eficientes, la cual deben cumplir algunos requerimientos para ser un sistema mecatrónico completo. Por lo tanto, su precisión y velocidad de posicionamiento definen la calidad y la productividad de los mecanismos de feed-drive. En la figura 1 se muestra una arquitectura general del hardware de la unidad de alimentación y su estructura de control por computadora de una mesa de máquina herramienta apoyada sobre la guía y movida linealmente por una tuerca-tornillo de bola (embalado) o por un sistema de motor lineal [4]. El tornillo de bola o embalado se puede conectar al servomotor rotativo directamente o mediante reducción de engranajes para máquinas grandes [4].



a. Physical components of a feed drive (WZL)

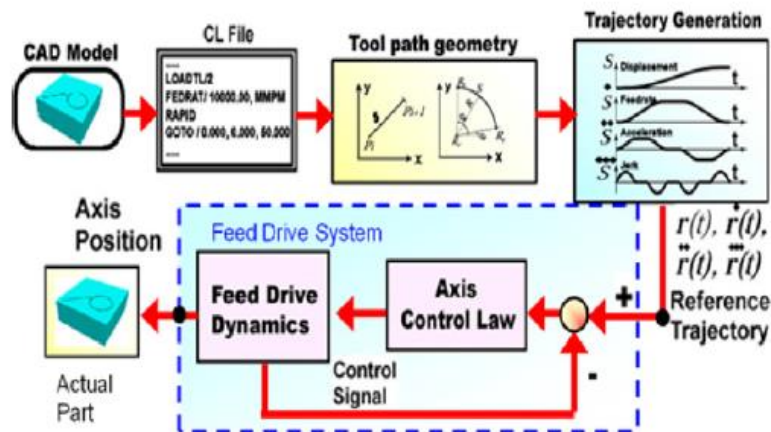


Figura 1.I: Elementos de un sistema feed-drive [5].

Los sistemas mecatrónicos de posicionamiento necesitan un algoritmo de control para realizar trayectorias o posiciones de alta precisión. Por otro lado, la adquisición de datos y el diseño del sistema embebido configurable es una parte fundamental para garantizar el procesamiento en hardware, alto rendimiento, recepción y envío de datos que generalmente está definido por las aplicaciones a las que se destina. Por lo tanto, las crecientes demandas sobre la mejora de los sistemas de precisión, basados en modernos dispositivos programables se están aplicando en las áreas científicas y de ingeniería. Hay aplicaciones en las que se requiere realizar trayectorias deseadas y trasladarse dentro de un espacio muy pequeño y limitado, como en sistemas ópticos y desarrollo de detectores ópticos de micro rayos X, etc [6-8]. Previo al diseño y el análisis se requiere identificar al modelo para conocer el comportamiento dinámico de cualquier sistema. El sistema con husillo de bolas es el mecanismo de accionamiento más empleado para largas trayectorias, posicionamiento y alta velocidad. Estudiar el mecanismo de un sistema con tuerca y husillo de bolas sigue siendo un desafío para los investigadores porque no es fácil obtener la resolución y precisión deseadas, lo que podría deberse a juego mecánico, vibraciones estructurales, adquisición, control electrónico y fricciones no lineales en el sistema [9-11]. El diseño de un controlador es necesario para asegurar el movimiento deseado, la velocidad y para reducir las señales de vibración detectadas, como en la literatura se desarrolla un modelo híbrido multidominio analítico y basado en datos para el sistema de alimentación de husillo de bolas para predecir el error de seguimiento [12]. Se realiza la comparación entre modelado y control, con el fin de mejorar la capacidad de seguimiento de trayectoria, para verificar la eficiencia de diferentes enfoques de control en la modernización del rendimiento de seguimiento en proyectos los cuales se enfocan en reunir los datos en tiempo real y otro en el análisis para comenzar anomalías [5, 13]. Los algoritmos en hardware y software tienen un amplio espectro de aplicaciones en robótica y mecatrónica que actualmente se implementan en sistemas embebidos en soluciones de arquitectura basadas en hardware [14-19].

Los sistemas embebidos se refieren a equipos electrónicos que poseen la capacidad de procesar datos y que posee un propósito específico, para ello puede contar con diversos tipos de instrumentos para procesar los datos, los más usados son los FPGA, como se puede ver en la gráfica que se encuentra en la figura [2], además de ellos se requieren sensores, para recoger la información y actuadores para interactuar con el entorno, actualmente vivimos en una época donde estamos rodeados por tecnología con usos específicos, así que muchos proyectos apuntan a aprovechar la estructura que ya ha sido diseñada para la comunicación, así como interactuar con otros dispositivos, este es el internet de las cosas [20].

El hardware que ha empleado los sistemas embebidos ha ido evolucionando conforme lo hace la micro tecnología, por ejemplo, según la fuente se ha logrado reproducir canales de hasta 8nm, lo que permite reducir el tamaño de los dispositivos, además este debe ser diseñado para ser compatible con un lenguaje de programación. Debido a que los sistemas embebidos son sistemas especializados no existe un estándar sobre el software con el que pueden ser programados, de hecho, esto causa que surjan los Frameworks que son herramientas de programación especializada que proveen de facilidades para el desarrollo de aplicaciones.

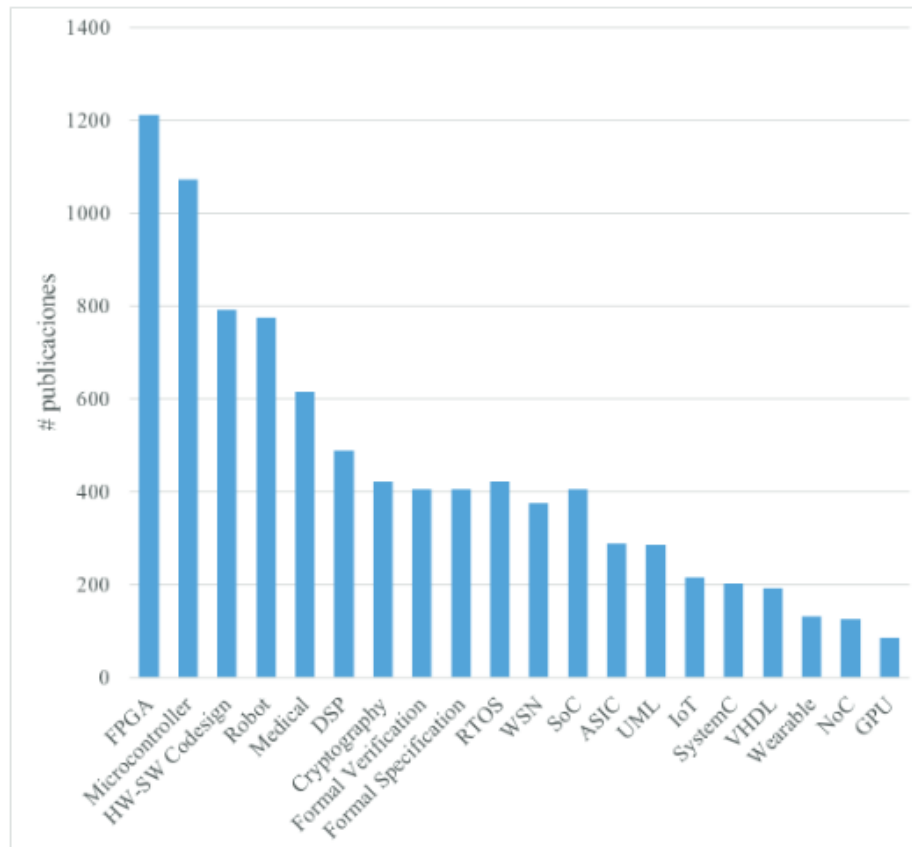


Figura 2.I: Utilización de dispositivos en publicaciones de la IEEE.

Los sistemas embebidos no trabajan necesariamente en tiempo real, pero gran parte de ellos se enfocan en este, debido a que de esta forma es posible realizar respuestas inmediatas a errores por medio de control, para hacerlo no solo se realizan desarrollos en el hardware para que se puedan emplear núcleos computacionales más potentes, sino que también se desarrollan nuevas técnicas de procesar la información recabada en tiempo real, así como generar aprendizaje autónomo.

Existen en la literatura algoritmos de control PID en hardware que fueron implementados proponiendo un diseño de controlador MIMO basado en Intel [21]; en otra literatura utilizaron la placa del microcontrolador Galileo 2 [22]; en otro caso se formuló la estrecha integración de una unidad de procesamiento de sensor (SPU) basada en hardware dentro de un procesador integrado [23]; en 2013 se desarrolló un controlador PI de orden fraccional para el control de velocidad del motor de CC en una placa FPGA (Field Programmable Gate Array); en Tailandia 2015 se presentó una solución que utiliza un FPGA de bajo costo de un solo chip para un sistema de accionamiento de motor PMSM de unidades múltiples [24]; y en 2018 que tiene como objetivo presentar tal realización de FO-PID implementable digitalmente para el control de motores de CC utilizando el procesador de señal digital (DSP) de punto flotante TMS320F28377S [25]; En todas las referencias previamente mencionadas, cada trabajo fue reportado con un modelo matemático general de un sistema de accionamiento de husillo de bolas con todas las variables dinámicas.

Con la finalidad de obtener señales lo más limpias posibles en los sistemas mecatrónicos, se utilizan diferentes tipos de acondicionamiento de dichas señales, uno de estos son el filtrado definiéndose como la eliminación de frecuencias que se presentan en los sistemas como ruido no deseable. Existen diferentes tipos de filtros dependiendo de la frecuencia que se desea eliminar. Los más conocidos son filtros pasa bajas y pasa altas en donde dependiendo de su naturaleza se tiene un valor conocido como frecuencia de corte. Si el filtro es pasa bajas entonces este valor representa el límite superior de los valores que podrán manejarse en el sistema. Los filtros digitales son los que se diseñan dentro de los microcontroladores, microprocesadores o sistemas embebidos y son empleados cada vez en más aplicaciones entre las cuales destaca el filtrado de imágenes y señales [26]. Otra aplicación de los filtros es el suavizado de las señales, este permite obtener mejora en la lectura de los datos, esto lleva a poder emplear de mejor forma los datos obtenidos pues se vuelven fiables permitiendo que se usen para sistemas de control.

El jerk también conocido como sobreaceleración es la 3ra derivada de la posición, representa el cambio en la aceleración, al no tratarse de las medidas principales de la mecánica de Newton tampoco tiene dispositivos para medirse, aun así, se ha desarrollado trabajos donde el jerk forma parte fundamental, funcionando el control a partir de reducir al mínimo el valor calculado de él [27]. Las vibraciones son parte de los problemas que pueden tener los sistemas mecánicos, por lo tanto, los robots con movimiento también pasan por esto, estas vibraciones, cuando se trabaja con precisión, pueden ser significativas a pesar de su rápida velocidad.

La tecnología avanza rápidamente y a cada momento nos sorprende con nuevas técnicas de diseño y medición. El desarrollo de instrumentos controlados por computadora está permitiendo automatizar procesos de caracterización y medición de diferentes sistemas usados en diversas áreas del conocimiento. En los dispositivos mecatrónicos de ultra alta precisión, la perturbación dominante es a menudo la fricción con sus características no lineales estocásticas, que dependen del tiempo y la posición. Esto influye significativamente en la precisión y la exactitud de las máquinas al inducir ciclos límite alrededor de la posición de referencia, errores de deslizamiento, estático y de seguimiento, así como grandes tiempos de asentamiento [28].

El presente proyecto de tesis incrementará la infraestructura de plataformas experimentales de control en la Facultad de Ciencias de la Electrónica, Este proyecto se puede considerar de gran impacto para la imagen de nuestra Institución ya que la infraestructura que se generaría basándose en el desarrollo propuesto junto con la infraestructura existente fortalecerá a la Maestría en ciencias de la electrónica, opción Automatización. También elevará la productividad, ya que se contaría con un prototipo institucional el que puede generar al menos una publicación en congreso internacional.

Los objetivos generales y específicos de este trabajo de tesis se enuncian a continuación.

## Objetivo general

*“Obtener el modelado y control de un robot de dos grados de libertad utilizando la metodología de parámetros agrupados para realizar los estudios de Jerk”.*

## Objetivos particulares

- Estudiar los fundamentos de la metodología parámetros agrupados.
- Obtener el modelado dinámico del robot utilizando la metodología parámetros agrupados.
- Estudiar el sistema embebido diseñado en la MCEA.
- Diseñar el hardware necesario para el controlador y los estudios de jerk.
- Diseñar el firmware necesario para el controlador y los estudios de jerk.
- Diseñar el software necesario para el controlador y los estudios de jerk.
- Seleccionar un controlador adecuado para el robot de dos grados de libertad.
- Realizar pruebas experimentales del sistema diseñado.

Para poder observar el jerk empleando únicamente como sensores los encoders del motor se debe eliminar el error de cuantificación, que se produce al interpretar de forma incorrecta la lectura de los datos y el efecto de muestreo que se presenta al pasar las cosas a un sistema digital, para el primero se cambiara la forma de estimar la velocidad, permitiendo tener datos más confiables, ahora empleando el firmware para realizar parte de los cálculos y terminarlos en el software, además para el segundo se implementará un filtro digital que elimine el efecto de muestreo y devuelva valores cercanos a los reales. A continuación, se implementó la compensación de fricción en el control.

En capítulo 1 se realiza una descripción de las características del sistema mecánico del robot, así como una descripción de cada elemento de la estructura por medio de diagrama de bloques, se abordará el sistema embebido utilizado, describiendo el hardware, así como el lenguaje de programación empleado.

En el capítulo 2 se hará una descripción de la metodología de parámetros agrupados, después se expondrá de forma detallada el desarrollo del modelo dinámico de un grado de libertad del sistema posteriormente, se realizará el desarrollo del segundo grado empleando lo obtenido con un solo grado de libertad y con ello se completará el modelo dinámico del robot, a continuación, se hará una explicación del control empleado y se expondrán conclusiones.

En el capítulo 3 se exponen el diseño del hardware empleado para el sistema, la tarjeta para la adquisición de datos y la etapa de potencia, se explicará la comunicación entre el sistema mecánico y el sistema embebido a continuación, se expondrá el diseño del firmware

desarrollado en Quartus seguido del software en Lenguaje D, se explicará la caracterización de los motores y se darán las conclusiones.

En el capítulo 4 se abordarán los problemas por los cuales no es posible recuperar el jerk, siendo el efecto de muestreo y el error de cuantificación, así como se propondrán formas de eliminarlos, para ello se propondrá una metodología la cual integra la aplicación de un filtro digital y el método de estimación por MT, con los cuales es posible obtener, la posición, velocidad, aceleración y jerk, para después dar las conclusiones.

En el último capítulo se presentan los resultados obtenidos al simular el modelo dinámico, así como los resultados prácticos al implementar lo expuesto en el capítulo 4 para la recuperación de la posición, velocidad, aceleración y jerk, para evaluar los resultados de la metodología empleada en la recuperación de datos en la plataforma experimental y se expondrán las conclusiones.

A continuación, se colocan las conclusiones generales de la tesis, después se muestra la bibliografía, en la cual fue basada esta investigación y, por último, se presenta el apéndice en el cual se muestra el reconocimiento obtenido por asistencia al congreso, el artículo publicado.

# CAPÍTULO 1:

## Robot cartesiano de dos grados de libertad

En este capítulo se incluirá la información del robot cartesiano con dos grados de libertad, que se encuentra en la Facultad de Ciencias de la Electrónica, el cual se desarrolló en la Maestría en Ciencias de la Electrónica, Opción en Automatización (MCEA) de la Benemérita Universidad Autónoma de Puebla (BUAP) que posee un sistema de desplazamiento por microposicionamiento, en el cual se realizarán los estudios de Jerk.

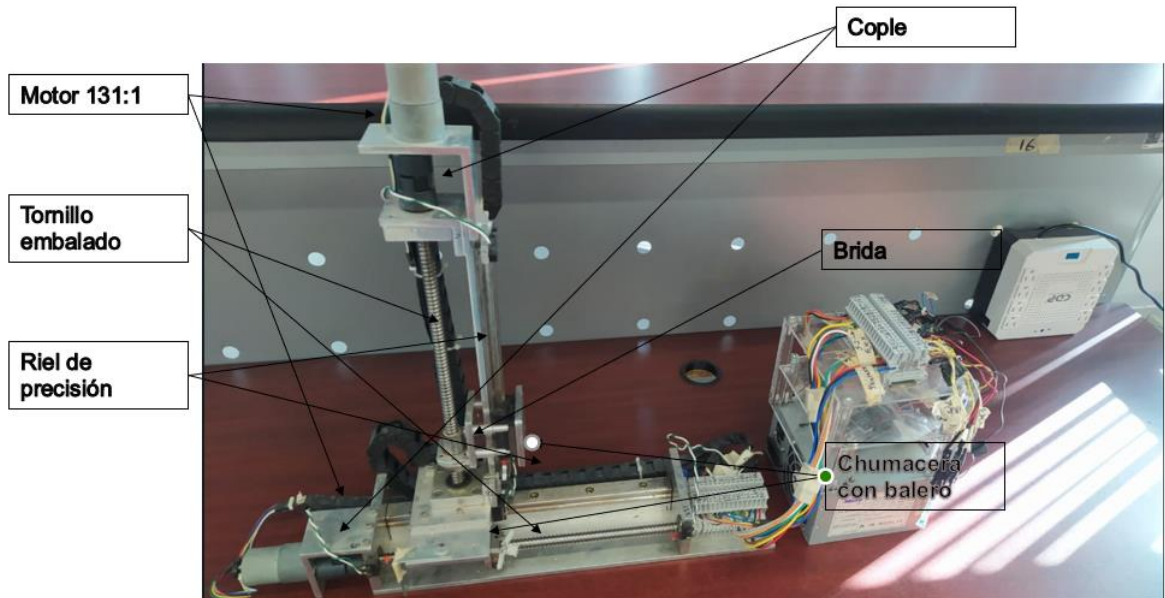


Figura 1: Robot cartesiano con 2 grados de libertad.

### 1.1 Descripción del robot

Cada eje del robot Robot cartesiano con 2 grados de libertad está compuesto por un motor eléctrico, una caja de engranajes, un cople, un tornillo embalado, sobre el cual se encuentra una chumacera con balero y la mesa de trabajo, para formar los dos grados de libertad, la estructura antes mencionada se coloca sobre la mesa de trabajo de otro eje, con la misma composición, de forma perpendicular, en la tabla 1.1.1 se presentan las características del robot de 2 grados de libertad.

<b>Estructura mecánica</b>	<b>Tipo Cartesiano</b>
Grados de libertad (GDL)	2 GDL
Rango de movimiento	Eje Z (vertical), eje Y (horizontal)
Retroalimentación	Encoder de cuadratura con una resolución de 64 pulsos por revolución del eje del motor.
Actuadores	-Motor de C.D. 12VCD con una caja de engranes de metal 131:1.
Capacidad de los motores	80 RPM y 300 mA funcionando libremente, 250 oz-in (18 kg-cm) y hasta 5 <sup>a</sup> .
Transmisión	Cople L-045
Precisión en el movimiento lineal para el eje Z	-Riel para guía lineal de 15mm -Bloque lineal para guía de 15mm. -Tornillo embalado C/Tuerca de 12 mm.
3Precisión en el movimiento lineal para el eje X	-Riel para guía lineal de 20mm -Bloque lineal para guía de 20mm. -Tornillo embalado C/Tuerca de 12 mm.
Peso del eje X	3.555kg
Peso del eje Z	2.005kg

Tabla 1.1.1: Características del robot cartesiano.

## 1.2 Diagrama a bloques general del sistema

A continuación, se muestra a detalle el diagrama de bloques general del sistema figura 1.2.1 el cual consta del sistema embebido y la estructura mecánica del robot cartesiano. Para este proyecto se ha propuesto la implementación del FPGA DE0-CV Board y un robot cartesiano 2 grados de libertad, además se agrega la etapa de potencia, para que con ella se pueda enviar la señal desde el FPGA y sea capaz de mover los motores. Seguido de eso está el encoder que se trata de una pieza incluida en los motores, es la parte que nos permitirá la retroalimentación no solo para corregir errores, sino también para poder enviarla haciendo uso del módulo de Wifly\*, después están los fines de carrera, estos funcionan como interruptores pues indican que se ha llegado al límite del área de trabajo y por lo tanto es necesario que se detenga el proceso.

Por parte de la programación es necesario diseñar un decoder para poder recolectar de manera correcta la información que nos arroja el encoder, después con ayuda de la retroalimentación, generar un control de lazo cerrado haciendo uso del microprocesador para ejecutar las instrucciones, el control propuesto es uno proporcional o de tangente hiperbólica. Además de esas funciones se cuenta con una conexión Wifi para poder enviar instrucciones y parámetros de inicio al FPGA para realizar el movimiento del Robot, así como la recolección de datos, esto es posible mediante un módulo externo al FPGA.

\*El módulo Wifly es el nombre del Hardware empleado para la comunicación por Wifi.

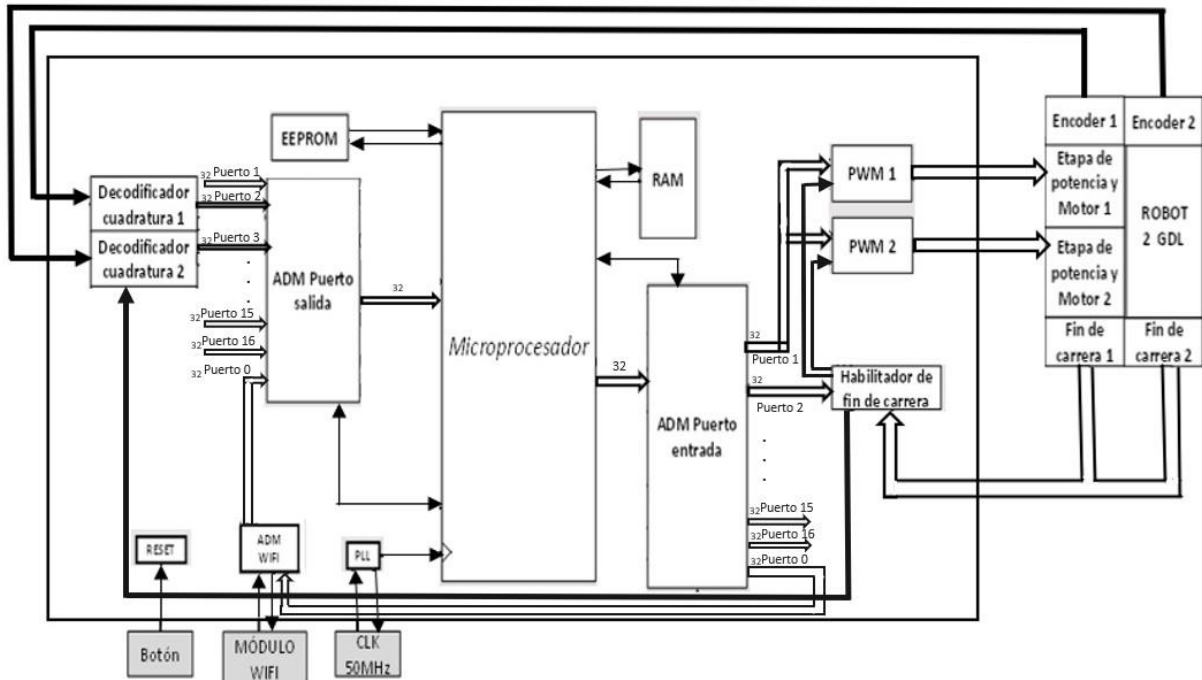


Figura 1.2.1: Diagrama a bloques general del sistema de control para el robot cartesiano de 2 GDL.

### 1.3 Sistema embebido

El sistema embebido se encuentra compuesto por la tarjeta DE0-CV, tiene un microprocesador Altera Cyclone V 5CEBA4F23C7N, que cuenta con una arquitectura Harvard, el diseño del firmware se hizo empleando el programa de Quartus II Web Edition, en este se realiza la conversión de ancho de pulsos A y B leída por medio de los Encoders de los motores al valor de posición. Como salida el Firmware se tiene una señal con ancho de pulso modulado (PWM), esta se produce con base en la frecuencia a la cual el motor trabaja de forma lineal, se indica en el software y el tiempo en alto será equivalente al porcentaje de velocidad requerida. Los datos son enviados por medio del Wifi al ordenador, pueden ser posición, una escala del torque y velocidad con intervalos limitados por el software. El torque puede ser obtenido por medio de una regla de 3, se leerá el torque deseado por el sistema el cual va de 1 a 1023 y se hace la regla de 3 con el torque máximo del motor que fue obtenido previamente en la caracterización. Cabe mencionar que el sistema embebido cuenta con memoria RAM y en ella se cuenta con miles de direcciones de 32 bits disponibles con el fin de almacenar diferentes variables y cálculos necesarios en el lazo de control.

N° PTO	Disposición de bits de puerto	
PT0	No usar PT0[32]	Reservado para la lectura datos WIFI PT0[31-1]
PT1	No usar PT1[32]	Bits de usuario disponible PT1[31-1]
PT2	No usar PT2[32]	Bits de usuario disponible PT2[31-1]
.	.	.
.	.	.
.	.	.
PT15	No usar PT5[32]	Bits de usuario disponible PT15[31-1]
PT16	No usar PT6[32]	Bits de usuario disponible PT16[31-1]

Tabla 1.3.1 Distribución de puertos de entrada.

N° PTO	Disposición de bits de puerto	
PT0	No usar PT0[32]	Reservado para la escritura datos WIFI PT0[31-1]
PT1	No usar PT1[32]	Bits de usuario disponible PT1[31-1]
PT2	No usar PT2[32]	Bits de usuario disponible PT2[31-1]
.	.	.
.	.	.
.	.	.
PT15	No usar PT5[32]	Bits de usuario disponible PT15[31-1]
PT16	No usar PT6[32]	Bits de usuario disponible PT16[31-1]

Tabla 1.3.2 Distribución de puertos de salida.

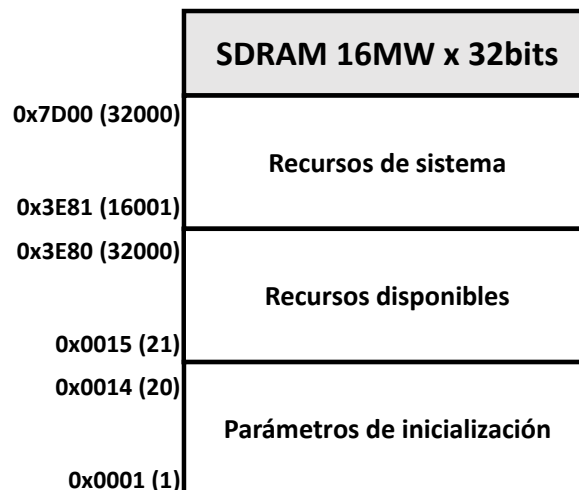


Tabla 1.3.3 Recursos disponibles y mapeo de memoria RAM.

### 1.3.1 Hardware

Los componentes empleados para el funcionamiento del sistema, es la tarjeta FPGA DE0-CV que cuenta con el microprocesador Altera Cyclone V 5CEBA4F23C7N, en la cual se conecta la tarjeta de comunicaciones, sobre la cual se coloca el módulo Wifly Rn 171, desde la tarjeta de comunicaciones empleando la extensión de los pines del FPGA se conecta el puente H, también los fines de carrera, los cuales en paralelo van a una resistencia y de ahí a tierra para su correcto funcionamiento, con los puentes H se encienden los motores en los cuales van los encoders, los cuales proporcionan una señal A y una desfasada 90° llamada B que se dirigen a la tarjeta de comunicaciones para que los datos ingresen al FPGA. En la imagen 5 se muestra en el recuadro azul el puente H, en el recuadro rojo el módulo Wifly y en recuadro amarillo el FPGA [7].

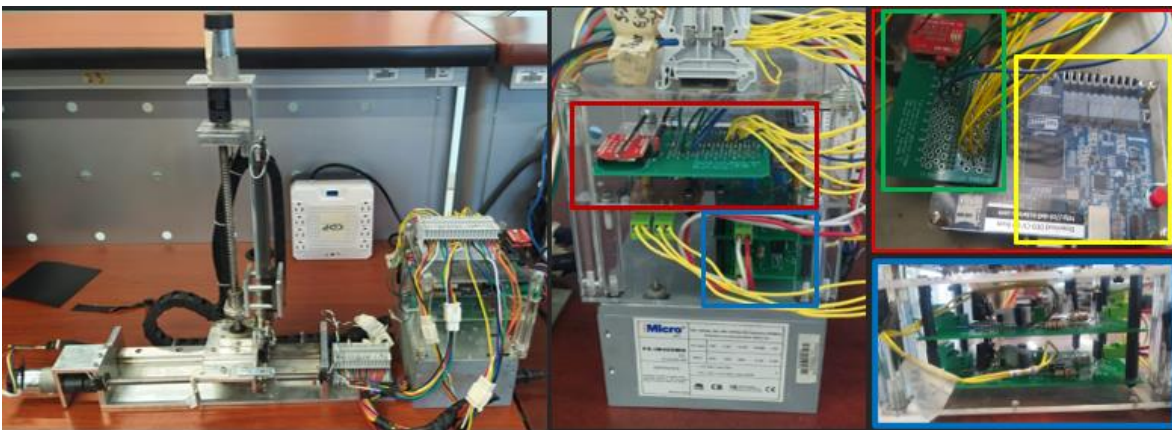


Figura 1.3.1: Base de FPGA y tarjetas acopladas, tarjeta de comunicación Wifly (verde), puentes h (azul) y tarjeta de desarrollo FPGA cyclone v intel DE0-CV (amarillo), fuente de alimentación (naranja).

### 1.3.2 Lenguaje D

Para poder emplear el FPGA necesitamos programarlo con un lenguaje que podamos modificar con facilidad y ser capaz de realizar los procesos que se necesiten, por lo que se ha seleccionado el Lenguaje D, el cual ha sido desarrollado en la Facultad de Ciencias de la Electrónica BUAP, este es posible de implementarlo gracias a que el FPGA incluyen una unidad de aritmética lógica, para realizar operaciones matemáticas además de eso debe contar operadores de salto de línea de instrucción, comparadores (IF, WHILE, FOR), operación bit a bit (instrucción AND, OR), entre otros. El lenguaje D es similar al lenguaje C con la diferencia de que el que usaremos está enfocada en registros, es decir está limitado a los apuntadores dentro de la RAM del microprocesador. La tabla 1.3.4 contiene las instrucciones de este lenguaje.

INSTRUCCIÓN Y SINTAXIS Lenguaje D				
IF	R	>,<,>=,<=	R o N	
CODIGO A REALIZAR SI LA CONDICION ES CIERTA				
ENDIF				
WHILE	R	>,<,>=,<=	R o N	
CODIGO A REALIZAR REPETIDAMENTE SI LA CONDICION ES CIERTA				
ENDWHILE				
FOR	N (VALOR DE INICIO)	N (VALOR DE AUMENTO)	N (VALOR FINAL)	
CODIGO A REALIZAR REPETIDAMENTE				
ENDFOR				
DELAY				
R	=	N		
R	=	R	+	R o N
R	=	R	-	R o N
R	=	R	*	R o N
R	=	R	/	R o N
R	=	R o N	^	R o N
R	=	LN	R o N	
R	=	EXP	R o N	
R	=	SQRT	R o N	
R	=	SIN	R o N	
R	=	COS	R o N	
R	=	TAN	R o N	
R	=	CSC	R o N	
R	=	SEC	R o N	
R	=	COT	R o N	
R	=	SINH	R o N	
R	=	COSH	R o N	
R	=	TANH	R o N	
R	=	CSCH	R o N	
R	=	SECH	R o N	
R	=	COTH	R o N	
R	=	ARCSIN	R o N	
R	=	ARCCOS	R o N	
R	=	ARCTAN	R o N	
LRAM	DIRECCION	R		
ERAM	R	DIRECCION		
LPOR	P	R		
EPOR	R	P		

Tabla 1.3.4: Instrucciones del lenguaje D.

Una característica peculiar de este lenguaje es que trabaja con datos con punto flotante, aunque estos se pueden convertir en enteros complemento 2.

## **1.4 Conclusiones**

Con todo lo presentado en el capítulo 1 ya se tiene una noción del funcionamiento del robot cartesiano, tanto de las características mecánicas de sus actuadores y de los elementos requeridos para su funcionamiento eléctrico, además se conoce el proceso por el cual se van a recopilar los datos dentro del FPGA para después analizarlos y poner en acción los actuadores, por lo que en el capítulo 2 se abordara el modelado dinámico del robot basándose en lo que conocemos de la parte mecánica por medio de la metodología de parámetros agrupados, mientras que en el capítulo 3 se expondrá el diseño de las tarjetas que se requieren acoplar al sistema para su funcionamiento, tales como la tarjeta de comunicación donde se debe colocar el módulo Wifly y sirve como extensión de las terminales del FPGA y la etapa de potencia en la cual emplearemos puentes H para el funcionamiento de los motores.

## CAPÍTULO 2: Modelado dinámico del robot

El modelo dinámico se trata del método con el cual se representa matemáticamente a un sistema físico, con el fin de simular sus respuestas y estudiar su comportamiento. En esta tesis se modela el sistema utilizando la metodología de parámetros agrupados. Así, el sistema de micro-posicionamiento pasará por el proceso en el cual se buscan todos los cambios de energía y fuerzas que suceden mientras se encuentran en movimiento, para estar en acción se aplica una tensión controlada al motor DC acoplado a un reductor, cople, tornillo embalado, tuerca y mesa de trabajo, después se utiliza la ecuación generalizada de Lagrange para el análisis de las energías con el fin de obtener el modelo dinámico del sistema mecánico [2].

### 2.1 Metodología de parámetros agrupados

La metodología de parámetros agrupados permite considerar múltiples variables físicas sobre un sistema, por lo que sirve para describir sistemas donde los diversos componentes presentan un juego mecánico. Los pasos para realizarlo se observan en la figura 2.1.1.

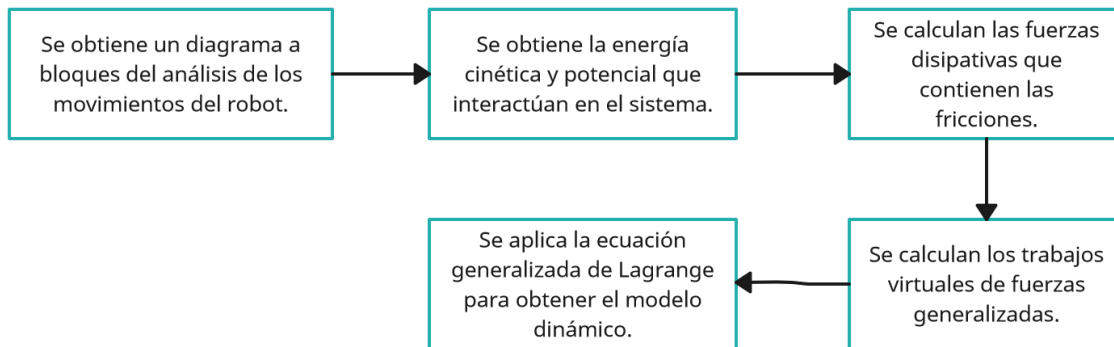


Figura 2.1.1: Diagrama de pasos para aplicación de la metodología de parámetros agrupados.

Para la construcción del diagrama a bloques se pueden emplear diferentes variables según la disciplina y el proceso que está realizando, los bloques que se emplearan para el modelar el robot de 2 GDL son los siguientes.

- Inductores.

Para sistemas mecánicos de traslación se simboliza con un punto de masa.



Figura 2.1.2: Punto de masa.

Se relaciona con la ecuación (2.1.1)

$$\rho = m\vec{v}, \quad (2.1.1)$$

donde:  $\rho$  es el momentum lineal,  $m$  es la masa y  $v$  es la velocidad [2].

Para sistemas mecánicos de rotacionales se sigue representado con la inercia de un punto de masa, pero ahora en rotacional.

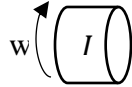


Figura 2.1.3: Punto de Inercia

Se relaciona con la ecuación (1.1.2) que se muestra a continuación:

$$H = I\vec{w}, \quad (2.1.3)$$

donde:  $H$  es el momentum angular,  $I$  es la inercia de la masa del motor y  $w$  es la velocidad angular [2].

En caso de los sistemas eléctricos se representa como un inductor.

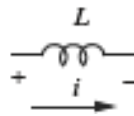


Figura 2.1.4: Inductor.

Y que describe la siguiente ecuación (2.1.4):

$$\lambda = L\vec{i}, \quad (2.1.4)$$

donde:  $\lambda$  es el flujo enlazado,  $L$  es la inductancia e  $i$  es corriente [2].

- Capacitores

Para la representación en la mecánica de traslación, ocupamos un resorte.

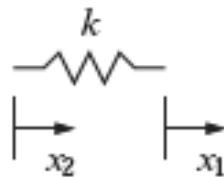


Figura 2.1.5 Resorte.

La ecuación (2.1.5) que se usa para describirlo es la Ley de Hooke:

$$f = k(x_2 - x_1), \quad (2.1.5)$$

donde:  $f$  es la fuerza deformante,  $k$  es la constante del resorte, y la resta representa el cambio de longitud que sufre [2].

En la representación torsional se trata de un resorte torsional.

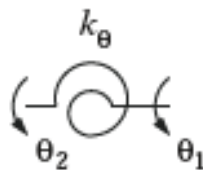


Figura 2.1.6: Resorte torsional.

Es este caso la ecuación (2.1.6) pasa a estar en función de los ángulos:

$$\tau = k_{\theta}(\theta_2 - \theta_1), \quad (2.1.6)$$

donde:  $\tau$  el torque,  $k_{\theta}$  es la constante de torsión angular,  $(\theta_2 - \theta_1)$  es la diferencia angular [2].

- Resistores

Para la representación en mecánica de traslación se usan los amortiguadores.

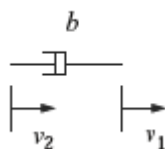


Figura 2.1.7: Amortiguador.

La ecuación (2.1.7) que lo presenta es:

$$F = b(\vec{v}_2 - \vec{v}_1), \quad (2.1.7)$$

donde:  $F$  es la fuerza aplicada al amortiguador,  $b$  es el coeficiente de amortiguamiento y  $v_2 - v_1$  es la diferencia de velocidad [2].

La representación en mecánica rotacional de los resistores es un amortiguador rotacional:

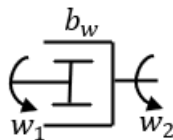


Figura 2.1.8: Amortiguador rotacional.

La ecuación (2.1.8) que lo representa es:

$$\tau = b_w(\vec{w}_2 - \vec{w}_1), \quad (2.1.8)$$

donde:  $\tau$  es el torque aplicado al amortiguador,  $b_w$  es el coeficiente de amortiguamiento rotacional y  $w$  es la velocidad angular [2].

Una vez que tenemos esta representación de cada elemento del sistema debemos definir un arreglo que esté compuesto por todas las variables del sistema, el cual llamaremos  $q$ , después es necesario separar cada una de las ecuaciones de nuestro sistema según se clasifique entre las siguientes, energía cinética está relacionada a las inercias, energía potencial que se relaciona a los capacitores, después de eso la energía de disipación relacionada a las resistencias y el trabajo imaginario que implica la transformación de energías.

$$\mathcal{L} = E_c - E_p \quad (2.1.9)$$

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}}{\partial q} + \frac{\partial E_d}{\partial \dot{q}} = \frac{\partial W}{\partial q} \quad (2.1.10)$$

En las ecuaciones (2.1.9) y (2.1.10) podemos distinguir la  $E_c$  siendo la energía cinética,  $E_p$  la energía potencial,  $E_d$  la energía disipativa y  $W$  es el trabajo, se debe realizar este procedimiento con cada variable del sistema modelado. Esta es la ecuación de Lagrange en su forma generalizada, que nos permite obtener el comportamiento físico de un sistema por medio de las energías implicadas en su movimiento las cuales pueden ser obtenidas por medio de la física newtoniana.

## 2.2 Modelo dinámico de un grado de libertad

La representación de este sistema como diagrama a bloques se ve en la figura 2.2.1.

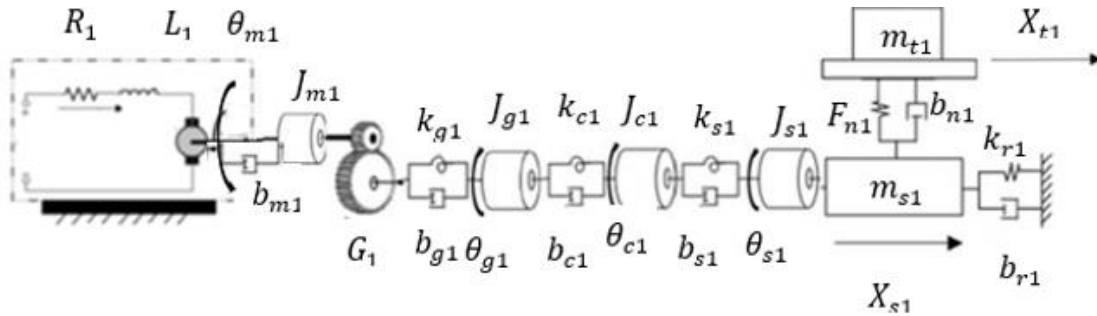


Figura 2.2.1: Esquema del eje Z.

Debido a la distribución inercial y de masas, la energía cinética  $E_c$  se expresa como:

$$E_c = \frac{1}{2} L \dot{q}_m^2 + \frac{1}{2} j_m \dot{\theta}_m^2 + \frac{1}{2} j_g G^2 \dot{\theta}_g^2 + \frac{1}{2} j_c \dot{\theta}_c^2 + \frac{1}{2} j_s \dot{\theta}_s^2 + \frac{1}{2} m_s \dot{X}_s^2 + \frac{1}{2} m_t \dot{X}_t^2 \quad (2.2.1)$$

La energía potencial  $E_p$  es almacenada en las partes con deformación elástica del sistema como incluye el husillo (tornillo embalado), el acoplamiento y la interfaz tornillo-tuerca. La ecuación (2.2.2) se expresan como:

$$E_p = \frac{1}{2} k_c (\theta_c - \theta_g)^2 + \frac{1}{2} k_s (\theta_s - \theta_c)^2 + \frac{1}{2} k_r (X_s)^2 \quad (2.2.2)$$

La fuerza de reacción en la caja de engranajes del actuador, el husillo (tornillo embalado) y la tuerca disminuye cuando se usan con frecuencia o después de un tiempo debido a la fricción, la rigidez y los modos estructurales, de modo que se reduce la vida útil restante del husillo (tornillo embalado). Se utiliza la función de disipación de Rayleigh para superar las fuerzas de fricción en el sistema de microposicionamiento. Por lo tanto, la fuerza de fricción es proporcional a la velocidad y la función de disipación de Rayleigh [12], que es igual a la mitad de la energía del sistema necesaria para superar las fuerzas de fricción de la caja de cambios, el husillo y la tuerca, se expresa de la siguiente manera, la fuerza de fricción se expresa de la siguiente manera:

$$Dt = \frac{1}{2}R\dot{q} + \frac{1}{2}b_m\dot{\theta}_m + \frac{1}{2}b_g(G\dot{\theta}_g - \dot{\theta}_m)^2 + \frac{1}{2}b_c(\dot{\theta}_g - \dot{\theta}_c)^2 + \frac{1}{2}b_s(\dot{\theta}_c - \dot{\theta}_s)^2 + \frac{1}{2}b_r(\dot{X}_s)^2 + \frac{1}{2}b_n(\dot{X}_t - [\dot{X}_s + Rn\dot{\theta}_s])^2 \quad (2.2.3)$$

$$\delta w = (E - Vm)\delta q + \tau_m\delta\theta_m + \tau_{g1}(G\delta\theta_g - \delta\theta_m) - F_f(\delta X_t) + F_{n1}(-\delta X_t + (\delta X_s + Rn\delta\theta_s)) \quad (2.2.4)$$

En la Tabla 2.2.1 se pueden observar las variables y constantes junto con su significado. Una vez tenemos estas ecuaciones (2.2.1-2.2.4) se puede sacar un vector con las variables que buscamos.

$$q = [q, \theta_m, \theta_g, \theta_c, \theta_s, X_s, X_t]^T \quad (2.2.5)$$

Haciendo uso de la ecuación generalizada de Lagrange para obtener el modelo dinámico de un sistema, sustituimos cada una en la ecuación (2.1.9) y (2.1.10) de forma que se obtienen las ecuaciones (2.2.6):

$$\begin{aligned} \frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}}{\partial q} + \frac{\partial E_d}{\partial \dot{q}} &= \frac{\partial W}{\partial q} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{\theta}_m} \right] - \frac{\partial \mathcal{L}}{\partial \theta_m} + \frac{\partial E_d}{\partial \dot{\theta}_m} &= \frac{\partial W}{\partial \theta_m} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{\theta}_g} \right] - \frac{\partial \mathcal{L}}{\partial \theta_g} + \frac{\partial E_d}{\partial \dot{\theta}_g} &= \frac{\partial W}{\partial \theta_g} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{\theta}_c} \right] - \frac{\partial \mathcal{L}}{\partial \theta_c} + \frac{\partial E_d}{\partial \dot{\theta}_c} &= \frac{\partial W}{\partial \theta_c} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{\theta}_s} \right] - \frac{\partial \mathcal{L}}{\partial \theta_s} + \frac{\partial E_d}{\partial \dot{\theta}_s} &= \frac{\partial W}{\partial \theta_s} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{X}_s} \right] - \frac{\partial \mathcal{L}}{\partial X_s} + \frac{\partial E_d}{\partial \dot{X}_s} &= \frac{\partial W}{\partial X_s} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{X}_t} \right] - \frac{\partial \mathcal{L}}{\partial X_t} + \frac{\partial E_d}{\partial \dot{X}_t} &= \frac{\partial W}{\partial X_t} \end{aligned} \quad (2.2.6)$$

S	Significado	S	Significado
$I_i$	Corriente eléctrica del motor [A]	$E_i$	Voltaje aplicado al motor [V]
$J_{mi}$	Inercia del motor [kgm <sup>2</sup> ]	$b_{gi}$	Coefficiente de Fricción caja engranajes [Nm s/rad]
$J_{gi}$	Inercia de la caja de engranajes [kgm <sup>2</sup> ]	$b_{ci}$	Coefficiente de Fricción cople [Nms/rad]
$J_{ci}$	Inercia de cople [kgm <sup>2</sup> ]	$b_{si}$	Coefficiente de Fricción t. tornillo embalado [Nms/rad]
$J_{si}$	Inercia del tornillo embalado [kgm <sup>2</sup> ]	$b_{ri}$	Coefficiente de Fricción a. tornillo embalado [Ns/m]
$m_{si}$	Masa del tornillo embalado [m]	$b_{ni}$	Coefficiente de Fricción mesa trabajo [Nm/s]
$m_{ti}$	Masa de la mesa trabajo [m]	$G_i$	Caja engranajes
$R_i$	Resistencia motor [Ohm]	$\theta_{mi}$	Posición angular eje del motor [rad]
$L_i$	Inductancia del motor [H]	$\theta_{gi}$	Posición angular Eje de la caja del motor [rad]
$k_{gi}$	Rigidez de caja de engranajes [Nm/rad]	$\theta_{ci}$	Posición angular eje del cople [rad]
$k_{ci}$	Rigidez de cople [Nm/rad]	$\theta_{si}$	Posición angular del tornillo embalado [rad]
$k_{si}$	Rigidez de t. tornillo embalado [Nm/rad]	$X_{si}$	Posición axial del tornillo embalado [m]
$F_{ni}$	Rigidez mesa de trabajo [N]	$X_{ti}$	Posición axial de la mesa de trabajo [m]
$k_{ri}$	Rigidez de a. tornillo embalado [N/m]	$q_i$	Carga en el motor [C]

Tabla 2.2.1: Variables y constantes del sistema.

Por lo tanto, se tienen las ecuaciones dinámicas (2.2.7-2.2.13):

$$L\ddot{q}_{m2} + R\dot{q} + k_e\dot{\theta}_m = E \quad (2.2.7)$$

$$J_m\ddot{\theta}_m + b_m\dot{\theta}_m + b_g[G\dot{\theta}_g - \dot{\theta}_m] - \tau_g = \tau_m \quad (2.2.8)$$

$$G^2J_{g2}\ddot{\theta}_{g2} - k_{c2}[\theta_{c2} - \theta_{g2}] + b_{g2}G(\dot{\theta}_g - \dot{\theta}_m) + b_c(\dot{\theta}_g - \dot{\theta}_c) = G\tau_g \quad (2.2.9)$$

$$J_c\ddot{\theta}_c + k_c(\theta_c - \theta_g) - k_s(\theta_s - \theta_c) + b_c(\dot{\theta}_g - \dot{\theta}_c) - b_s(\dot{\theta}_c - \dot{\theta}_s) = 0 \quad (2.2.10)$$

$$J_s\ddot{\theta}_s + K_s(\theta_s - \theta_c) - b_s(\dot{\theta}_c - \dot{\theta}_s) - Rnb_n(\dot{X}_t - (\dot{X}_s + Rn\dot{\theta}_s)) = RnFn \quad (2.2.11)$$

$$m_s\ddot{X}_s + (k_rX_s) + b_r\dot{X}_s - b_n[\dot{X}_t(\dot{X}_s + Rn\dot{\theta}_s)] = Fn \quad (2.2.12)$$

$$(m_t)\ddot{X}_t + b_n[\dot{X}_t(\dot{X}_s + Rn\dot{\theta}_s)] = -Fn - Ff \quad (2.2.13)$$

Las ecuaciones (2.2.7-2.2.13) representan la dinámica del sistema de un eje que consiste en un modelo matemático de 1 GDL debido con 7 variables físicas obtenidas con una sola entrada de voltaje  $E$  con el fin de estudiar el comportamiento dinámico de todas las variables internas del sistema, esto es posible usando el método de parámetros agrupados, es decir, en primera, es el motor la cual se aplica un voltaje controlado o torque en la entrada del actuador  $E$ ; luego el motor está agrupado o conectado a un reductor (caja de cambios); posteriormente a un acoplamiento o cople flexible; las siguientes dos ecuaciones representan un eje de husillo (tuerca embalada) de precisión que consiste en un resorte que analiza la deformación elástica axial y torsional del tornillo; y finalmente, el bloque deslizante que consiste en la deformación axial de la mesa deslizante debido a la interacción de la fuerza de contacto entre el mecanismo de transmisión del husillo (tornillo embalado) y la tuerca; respectivamente.

Las ecuaciones (2.2.14-2.2.18) que conectan directamente la parte eléctrica del motor a la parte mecánica de su eje, así como algunas relaciones mecánicas:

$$Vm = k_e \dot{\theta}_m \quad (2.2.14)$$

$$\tau_m = K_m \dot{q} \quad (2.2.15)$$

$$\tau_g = k_g (\theta_m - nG\delta\theta_g) \quad (2.2.16)$$

$$Rn = \frac{X_{step}}{2\pi\phi} \quad (2.2.17)$$

$$Fn = k_n (X_{t1} - (X_{s1} + Rn\theta_{s1})) \quad (2.2.18)$$

Donde  $\tau_m$  es el par electromagnético del rotor generado por una corriente de excitación y la corriente inducida del motor;  $K_m$  es una constante mecánica generada por el flujo de campo magnético del actuador;  $Vm$  es una fuerza eléctrica inducida en presencia del campo magnético,  $k_e$  es una constante eléctrica generada por el flujo magnético del inducido del actuador.  $G$  es la relación de transmisión de la caja de cambios,  $n$  es la eficiencia mecánica de la caja de cambios,  $\tau_g$  es el par o torque torsional de la caja de cambios que consiste en la deformación torsional del rotor del motor y el engranaje,  $Rn$  que es el paso del tornillo embalado  $X_{step}$ , sobre  $2\pi$  por la eficiencia del tornillo  $\phi$ ,  $Fn$  es la fricción en la mesa de trabajo, que está en función de las posiciones por la rigidez.

### 2.3 Modelo dinámico de dos grados de libertad

Debido a que el sistema trabaja en 2 ejes actuando el X como base del Z, el movimiento de estos se puede describir como en la ecuación (2.3.1),

$$\begin{matrix} EJE X \\ \begin{bmatrix} x_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} x_{s1} + x_{t1} \\ 0 \end{bmatrix} \end{matrix} \quad \begin{matrix} EJE Z \\ \begin{bmatrix} x_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_{s1} + x_{t1} \\ x_{s2} + x_{t2} \end{bmatrix} \end{matrix} \quad (2.3.1)$$

donde la posición del eje X es únicamente afectada por sus propios elementos, mientras que en el eje Z se puede observar como el eje X determina una de sus coordenadas. En la figura 2.3.1 se muestra el diagrama a boques del sistema mecatrónico en el cual se deben introducir

todas las variables que interactúan de tal manera que se realice el movimiento de cada uno de los ejes del robot. En el recuadro 1 está el motor, en el diagrama a bloques se encuentra su modelo eléctrico, el modelo mecánico y su caja de engranajes donde se da un movimiento rotacional, en el recuadro 2 se encuentra el cople que a su salida genera otra variable rotacional, mientras en el recuadro 3 se localiza el tornillo embalado que provoca otro cambio de variable rotacional, sobre el mismo está la brida que transforma el movimiento rotacional a lineal y encima de este el recuadro 5 que representa la mesa de trabajo donde hay otro cambio en la variable lineal, análogamente se representa de igual manera el eje Z, sin embargo, se agrega un efecto debido a la gravedad.

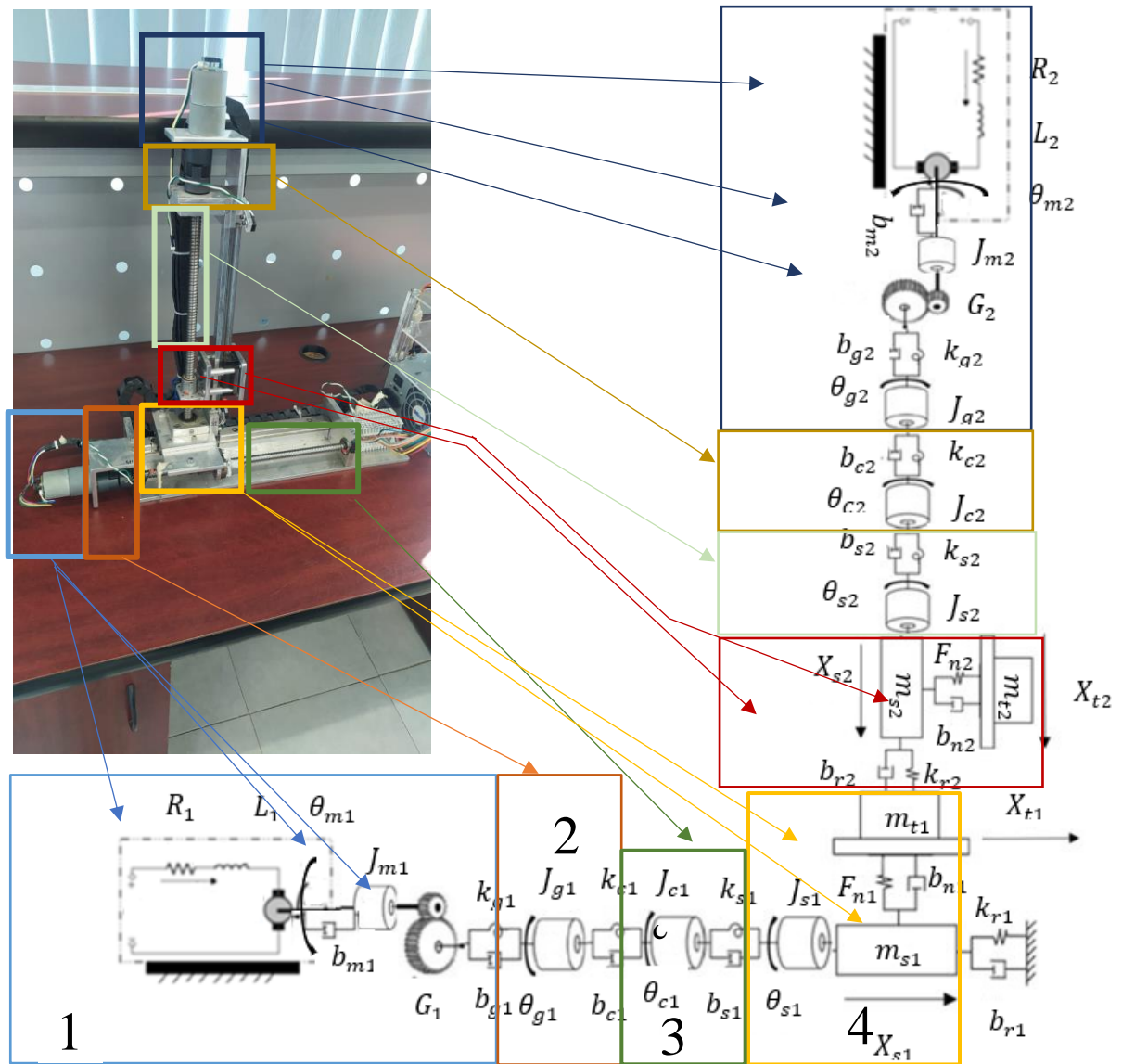


Figura 2.3.1: Diagrama a bloques.

Debido a la distribución inercial y de masas, la energía cinética  $E_{c_n}$ , donde n corresponde al eje, se expresa como:

$$Ec_1 = \frac{1}{2}L_1\dot{q}_1^2 + \frac{1}{2}j_{m1}\dot{\theta}_{m1}^2 + \frac{1}{2}j_{g1}G^2\dot{\theta}_{g1}^2 + \frac{1}{2}j_{c1}\dot{\theta}_{c1}^2 + \frac{1}{2}j_{s1}\dot{\theta}_{s1}^2 + \frac{1}{2}m_{s1}\dot{x}_{s1}^2 + \frac{1}{2}m_{t1}\dot{x}_{t1}^2 \quad (2.3.2)$$

$$Ec_2 = \frac{1}{2}L_2\dot{q}_2^2 + \frac{1}{2}j_{m2}\dot{\theta}_{m2}^2 + \frac{1}{2}Gj_{g2}\dot{\theta}_g^2 + \frac{1}{2}j_{c2}\dot{\theta}_{c2}^2 + \frac{1}{2}j_{s2}\dot{\theta}_{s2}^2 + \frac{1}{2}m_{s2}(\dot{x}_{s1}^2 + \dot{x}_{s2}^2) + \frac{1}{2}m_{t2}(\dot{x}_{t1}^2 + \dot{x}_{t2}^2) \quad (2.3.3)$$

La energía potencial  $Ep_n$ , donde n corresponde al eje, es almacenada en las partes de la deformación elástica del sistema la cual incluye el husillo (tornillo embalado), el acoplamiento y la interfaz tornillo-tuerca. Las ecuaciones se expresan como (2.3.4) y (2.3.5):

$$Ep1 = \frac{1}{2}k_{c1}(\theta_{c1} - \theta_{g1})^2 + \frac{1}{2}k_{s1}(\theta_{s1} - \theta_{c1})^2 + \frac{1}{2}k_{r1}(X_{s1})^2 \quad (2.3.4)$$

$$Ep2 = \frac{1}{2}k_{c2}(\theta_{c2} - \theta_{g2})^2 + \frac{1}{2}k_{s2}(\theta_{s2} - \theta_{c2})^2 + \frac{1}{2}k_{r2}(X_{s2})^2 \quad (2.3.5)$$

La fuerza de reacción en la caja de engranajes del actuador, el husillo (tornillo embalado) y la tuerca disminuye cuando se usan con frecuencia o después de un tiempo debido a la fricción, la rigidez y los modos estructurales, de modo que se reduce la vida útil restante del husillo (tornillo embalado). Se utiliza la función de disipación de Rayleigh para superar las fuerzas de fricción en el sistema de microposicionamiento. Por lo tanto, la fuerza de fricción es proporcional a la velocidad y la función de disipación de Rayleigh que es igual a la mitad de la energía del sistema necesaria para superar las fuerzas de fricción de la caja de cambios, el husillo y la tuerca, se representa como  $Dt_n$ , donde n corresponde al eje, se expresa de la siguiente manera.

$$Dt1 = \frac{1}{2}Rq\dot{q}_1 + \frac{1}{2}b_{m1}\dot{\theta}_{m1} + \frac{1}{2}b_{g1}(G\dot{\theta}_{g1} - \dot{\theta}_{m1})^2 + \frac{1}{2}b_{c1}(\dot{\theta}_{g1} - \dot{\theta}_{c1})^2 + \frac{1}{2}b_{s1}(\dot{\theta}_{c1} - \dot{\theta}_{s1})^2 + \frac{1}{2}b_{r1}(\dot{X}_{s1})^2 + \frac{1}{2}b_{n1}(\dot{X}_{t1} - [\dot{X}_{s1} + Rn\dot{\theta}_{s1}])^2 \quad (2.3.6)$$

$$Dt2 = \frac{1}{2}Rq\dot{q}_2 + \frac{1}{2}b_{m2}\dot{\theta}_{m2} + \frac{1}{2}b_{g2}(G\dot{\theta}_{g2} - \dot{\theta}_{m2})^2 + \frac{1}{2}b_{c2}(\dot{\theta}_{g2} - \dot{\theta}_{c2})^2 + \frac{1}{2}b_{s2}(\dot{\theta}_{c2} - \dot{\theta}_{s2})^2 + \frac{1}{2}b_{r2}(\dot{X}_{s2})^2 + \frac{1}{2}b_{n2}(\dot{X}_{t2} - [\dot{X}_{s2} + Rn\dot{\theta}_{s2}])^2 \quad (2.3.7)$$

$$\delta w1 = (E_1 - Vm_1)\delta q1 + \tau_{m1}\delta\theta_{m1} + \tau_{g1}(G\delta\theta_{g1} - \delta\theta_{m1}) + F_{n1}(-\delta X_{t1} + (\delta X_{s1} + Rn\delta\theta_{s1})) - F_f(\delta X_{t1}) \quad (2.3.8)$$

$$\delta w2 = (E_2 - Vm_2)\delta q2 + \tau_{m2}\delta\theta_{m2} + \tau_{g2}(\delta\theta_{m2} - G\delta\theta_{g2}) + F_{n2}(\delta X_{t2} - (\delta X_{s2} + Rn\delta\theta_{s2})) - F_f(\delta X_{t2}) + mg(\delta X_{t2}) \quad (2.3.9)$$

Las ecuaciones (2.3.10-2.3.14) que conectan directamente la parte eléctrica del motor a la parte mecánica de su eje, así como otras que forman parte del trabajo virtual se muestran a en las ecuaciones (2.3.10-2.3.14).

$$Vm_i = k_{ei}\dot{\theta}_{mi} \quad (2.3.10)$$

$$\tau_{mi} = K_{mi}\dot{q}_i \quad (2.3.11)$$

$$\tau_{gi} = k_{gi}(\delta\theta_{mi} - G\delta\theta_{gi}) \quad (2.3.12)$$

$$Rn_i = \frac{X_{step}}{2\pi\phi_i} \quad (2.3.13)$$

$$Fn_i = k_{ni}(X_{ti} - (X_{si} + Rn_i\theta_{si})) \quad (2.3.14)$$

Donde  $\tau_m$  es el par electromagnético del rotor generado por una corriente de excitación y la corriente inducida del motor;  $K_m$  es una constante mecánica generada por el flujo de campo magnético del actuador;  $Vm$  es una fuerza eléctrica inducida en presencia del campo magnético,  $k_e$  es una constante eléctrica generada por el flujo magnético del inducido del actuador.  $G$  es la relación de transmisión de la caja de cambios,  $\eta$  es la eficiencia mecánica de la caja de cambios,  $\tau_g$  es el par o torque torsional de la caja de cambios que consiste en la deformación torsional del rotor del motor y el engranaje,  $Rn$  que es el paso del tornillo embalado  $X_{step}$ , sobre  $2\pi$  por la eficiencia del tornillo  $\phi$ ,  $F_n$  es la fricción en la mesa de trabajo, que está en función de las posiciones por la rigidez. Una vez tenemos estas ecuaciones se puede sacar un vector con las variables que buscamos.

$$q = [q, \theta_m, \theta_g, \theta_c, \theta_s, X_s, X_t]^T \quad (2.3.15)$$

El método de Lagrange para obtener el modelo dinámico de un sistema requiere sustituir cada una en la ecuación (2.1.9) y (2.1.10) de forma que se obtienen las ecuaciones (2.3.16).

$$\begin{aligned} \frac{d}{dt} \left[ \frac{\partial \mathcal{L}_i}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}_i}{\partial q_i} + \frac{\partial E_{di}}{\partial \dot{q}_i} &= \frac{\partial W_i}{\partial q_i} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}_i}{\partial \dot{\theta}_{mi}} \right] - \frac{\partial \mathcal{L}_i}{\partial \theta_{mi}} + \frac{\partial E_{di}}{\partial \dot{\theta}_{mi}} &= \frac{\partial W_i}{\partial \theta_{mi}} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}_i}{\partial \dot{\theta}_{gi}} \right] - \frac{\partial \mathcal{L}_i}{\partial \theta_{gi}} + \frac{\partial E_{di}}{\partial \dot{\theta}_{gi}} &= \frac{\partial W_i}{\partial \theta_{gi}} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}_i}{\partial \dot{\theta}_{ci}} \right] - \frac{\partial \mathcal{L}_i}{\partial \theta_{ci}} + \frac{\partial E_{di}}{\partial \dot{\theta}_{ci}} &= \frac{\partial W_i}{\partial \theta_{ci}} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}_i}{\partial \dot{\theta}_{si}} \right] - \frac{\partial \mathcal{L}_i}{\partial \theta_{si}} + \frac{\partial E_{di}}{\partial \dot{\theta}_{si}} &= \frac{\partial W_i}{\partial \theta_{si}} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}_i}{\partial \dot{X}_{si}} \right] - \frac{\partial \mathcal{L}_i}{\partial X_{si}} + \frac{\partial E_{di}}{\partial \dot{X}_{si}} &= \frac{\partial W_i}{\partial X_{si}} \\ \frac{d}{dt} \left[ \frac{\partial \mathcal{L}_i}{\partial \dot{X}_{ti}} \right] - \frac{\partial \mathcal{L}_i}{\partial X_{ti}} + \frac{\partial E_{di}}{\partial \dot{X}_{ti}} &= \frac{\partial W_i}{\partial X_{ti}} \end{aligned} \quad (2.3.16)$$

Al resolverlas nos dan las ecuaciones (2.3.17-2.3.23) que describen cada parte del sistema.

Para el eje X:

$$L\ddot{q}_{m1} + R\dot{q}_1 + k_{e1}\dot{\theta}_{m1} = E \quad (2.3.18)$$

$$J_{m1}\ddot{\theta}_{m1} + b_{m1}\dot{\theta}_{m1} + b_{g1}[G\dot{\theta}_{g1} - \dot{\theta}_{m1}] - \tau_{g1} = \tau_{m1}$$

$$G^2J_{g1}\ddot{\theta}_{g1} - k_{c1}[\theta_{c1} - \theta_{g1}] + b_{g1}G(\dot{\theta}_{g1} - \dot{\theta}_{m1}) + b_{c1}(\dot{\theta}_{g1} - \dot{\theta}_{c1}) = G\tau_{g1} \quad (2.3.19)$$

$$J_{c1}\ddot{\theta}_{c1} + k_{c1}(\theta_{c1} - \theta_{g1}) \quad (2.3.20)$$

$$-k_{s1}(\theta_{s1} - \theta_{c1}) + b_{c1}(\dot{\theta}_{g1} - \dot{\theta}_{c1}) - b_{s1}(\dot{\theta}_{c1} - \dot{\theta}_{s1}) = 0$$

$$J_{s1}\ddot{\theta}_{s1} \quad (2.3.21)$$

$$K_{s1}(\theta_{s1} - \theta_{c1}) - b_{s1}(\dot{\theta}_{c1} - \dot{\theta}_{s1}) - Rnb_{n1}(\dot{X}_{t1} - (\dot{X}_{s1} + Rn\dot{\theta}_{s1})) = RnFn$$

$$(m_{s1} + m_{s2})\ddot{X}_{s1} + (k_{r1}X_{s1}) + b_{r1}\dot{X}_{s1} - b_{n1}[\dot{X}_{t1}(\dot{X}_{s1} + Rn\dot{\theta}_{s1})] = Fn \quad (2.3.22)$$

$$((m_{t1} + m_{t2})\ddot{X}_{t1} + b_{n1}(\dot{X}_{t1}(\dot{X}_{s1} + Rn\dot{\theta}_{s1}))) + Fn = -Ff \quad (2.3.23)$$

Para el eje Z lo describen las ecuaciones (2.3.24-2.3.30):

$$L\ddot{q}_{m2} + R\dot{q}_2 + k_{e2}\dot{\theta}_{m2} = E \quad (2.3.24)$$

$$J_{m2}\ddot{\theta}_{m2} + b_{m2}\dot{\theta}_{m2} + b_{g2}[G\dot{\theta}_{g2} - \dot{\theta}_{m2}] - \tau_{g2} = \tau_{m2} \quad (2.3.25)$$

$$G^2J_{g2}\ddot{\theta}_{g2} - k_{c2}[\theta_{c2} - \theta_{g2}] + b_{g2}G(\dot{\theta}_{g2} - \dot{\theta}_{m2}) + b_{c1}(\dot{\theta}_{g2} - \dot{\theta}_{c2}) = G\tau_{g2} \quad (2.3.26)$$

$$J_{c2}\ddot{\theta}_{c2} + k_{c2}(\theta_{c2} - \theta_{g2}) - k_{s2}(\theta_{s2} - \theta_{c2}) + b_{c2}(\dot{\theta}_{g2} - \dot{\theta}_{c2}) - b_{s2}(\dot{\theta}_{c2} - \dot{\theta}_{s2}) = 0 \quad (2.3.27)$$

$$J_{s2}\ddot{\theta}_{s2} + K_{s2}(\theta_{s2} - \theta_{c2}) - b_{s2}(\dot{\theta}_{c2} - \dot{\theta}_{s2}) - Rnb_{n2}(\dot{X}_{t2} - (\dot{X}_{s2} + Rn\dot{\theta}_{s2})) = RnFn \quad (2.3.28)$$

$$m_{s2}\ddot{X}_{s2} + (k_{r2}X_{s2}) + b_{r2}\dot{X}_{s2} - b_{n2}[\dot{X}_{t2}(\dot{X}_{s2} + Rn\dot{\theta}_{s2})] = Fn \quad (2.3.29)$$

$$(m_{t2})\ddot{X}_{t2} + b_{n2}(\dot{X}_{t2}(\dot{X}_{s2} + Rn\dot{\theta}_{s2})) + g + Fn = -Ff \quad (2.3.30)$$

## 2.4 Controlador

Para realizar el control de posición de los 2 ejes del robot, se implementó un control con función tangente hiperbólica para mantener la acción de control acotada, ya que además de tener sustento científico, una de las cualidades de este esquema de control es que permite

operar al actuador en la parte lineal lejos de los límites de saturación [29]. En la figura 2.4.1 se muestra el diagrama a bloques del control de posición que se desea implementar en los ejes.

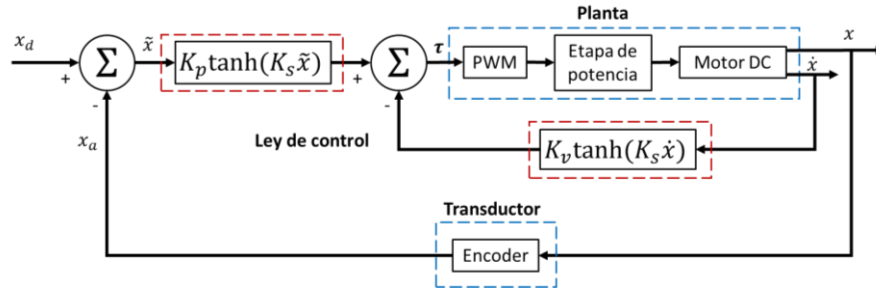


Figura 2.4.1: Lazo cerrado para el control de posición de los ejes.

El esquema de control para los ejes está dado por [29] y [30] mediante la ecuación (2.4.1):

$$\tau = K_p \tanh(K_s \tilde{x}) - K_v \tanh(K_s \dot{x}) \quad (2.4.1)$$

donde,

- $K_p$  es la ganancia proporcional y es definida positiva.
- $K_v$  es la ganancia derivativa y es definida positiva.
- $K_s$  es la pendiente del control tangente hiperbólica  $\tanh(K_s \tilde{x})$  y de la acción derivativa  $\tanh(K_s \dot{x})$  y es definida positiva.
- $\tilde{x}$  es el error de posición, definido como la diferencia entre la posición deseada y la posición actual, es decir,  $\tilde{x} = x_d - x_a$ .
- $\dot{x}$  es la velocidad de movimiento que se emplea para inyección de amortiguamiento.
- $K_p \tanh(K_s \tilde{x}) = 0 \Leftrightarrow \tilde{x} = 0$ .
- $K_v \tanh(K_s \dot{x}) = 0 \Leftrightarrow \dot{x} = 0$ .

La velocidad se puede estimar mediante la ecuación (2.4.2):

$$\dot{x}(k) = \frac{x(k) - x(k-1)}{h} \quad (2.4.2)$$

## 2.5 Conclusiones

Se expusieron los elementos necesarios para modelar el sistema con la metodología de parámetros agrupados, pasando de la representación de un grado a dos grados del robot propuesto en esta tesis, aprovechando que la estructura de ambos ejes es igual, pero respetando el cambio en las energías al estar de forma horizontal y vertical, con esto ya se cuenta con un modelo confiable que describe el sistema, además se ha preparado una teoría de control, por lo que se tiene todo lo necesario para simular el sistema, para hacerlo se empleó la herramienta de Simulink en la aplicación de Matlab lo cual se presentara en el último capítulo, en el siguiente capítulo se expondrá el diseño de las tarjetas y la recopilación de datos requeridos para poner el sistema real en funcionamiento

# CAPÍTULO 3:

## Desarrollo de la electrónica para el control del robot.

Para que el robot planteado en esta tesis se ponga en funcionamiento es necesario acoplarle la parte electrónica que se planteó en el capítulo 1, además de desarrollar un programa adecuado para el sistema, este programa debe permitir compartir los datos que están siendo procesados en su interior para que puedan ser recopilados y analizados posteriormente.

### 3.1 Sistema de adquisición de datos

El sistema completo funciona gracias a los motores, por lo tanto, es importante controlar la energía que se le proporciona para que funcione correctamente. Aunque estos motores no funcionan de manera simple, se han estudiado para entender cómo trabajan, más detalles sobre este proceso en el subcapítulo 3.6 de la tesis, si se le envía una señal rectangular con ancho de pulso modulado (PWM) a cada motor, con una frecuencia específica para cada uno y ajustada según el voltaje que reciban, la velocidad del motor cambia de forma lineal con respecto al ancho de pulso en la señal. Para crearla, se utilizará una tarjeta llamada DE0-CV, que tiene un componente llamado FPGA Altera Cyclone V integrado.

Para producir la señal se hizo uso de los tiempos en el generador de PWM, estos son determinados por medio de contadores que miden el tiempo en términos de ciclos de reloj del FPGA con una frecuencia de trabajo de 100MHz. Para la construcción del bloque de PWM hay que determinar la entrada y salida que usaremos en este caso tenemos como entradas: el periodo de operación, ya que, aunque una vez seleccionado el periodo ya no va a cambiar, si queremos que sea un parámetro que podamos ajustar de acuerdo a nuestras necesidades, este periodo se obtiene por medio de contar  $N_{clk}$  ciclos de reloj respecto al periodo de la señal de reloj del FPGA, en nuestro caso es de  $10ns$ , debido a que la frecuencia del reloj del FPGA es de 100MHz, ver ecuación 3.1.1. El ancho del pulso también será un grupo de bits de entrada, estará determinado por  $N_{AP}$ , ver ecuación 3.1.2 y otro bit de entrada es dirección que indica la dirección de giro del motor. Como salidas tendremos PWM+ y PWM-, cada salida está en función de la dirección de giro requerida. Para ello se propone el bloque en la figura 3.1.1.

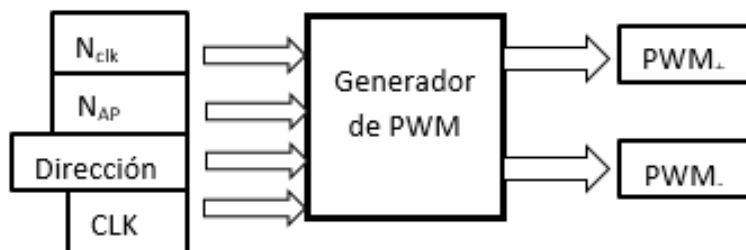


Figura 3.1.1: Diagrama del PWM.

Para obtener las variables se emplean las ecuaciones (3.1.1) y (3.1.2).

$$N_{clk} = \frac{T_{pwm}}{T_{clk}}, \quad (3.1.1)$$

donde:  $N_{clk}$  es el número de ciclos que corresponde a un periodo del PWM,  $T_{pwm}$  representa el periodo del PWM,  $T_{clk}$  es el periodo del reloj del FPGA.

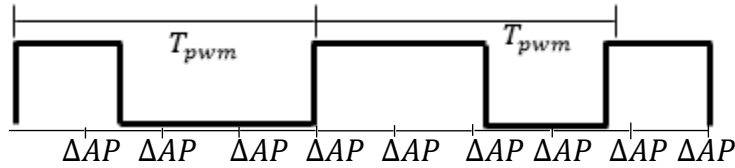


Figura 3.1.2: Diagrama de tiempos del PWM.

La fórmula para obtener el paso es la ecuación (3.1.2):

$$\Delta AP = \frac{T_{pwm}}{N_{APmax}}, \quad (3.1.2)$$

donde:  $\Delta AP$  es un número que representa el paso mínimo para establecer el ancho de pulso, se obtiene dividiendo el periodo del PWM ( $T_{pwm}$ ) entre el número máximo de particiones deseadas ( $N_{APmax}$ ), en nuestro caso será de 1000.

El número que representa el tiempo del ancho de pulso es:

$$N_{AP} = \frac{AP}{\Delta AP}, \quad (3.1.3)$$

donde:  $N_{AP}$  es el número que representa el ancho de pulso solicitado,  $AP$  es el tiempo que el pulso estará en alto y es el número que corresponde a la partición mínima del periodo del PWM. Esta señal va a dar a la etapa de potencia, la cual corresponde a un puente H diseñado para poder emplear los motores propuesto, el proceso está detallado en el subcapítulo 3.2 de la tesis.

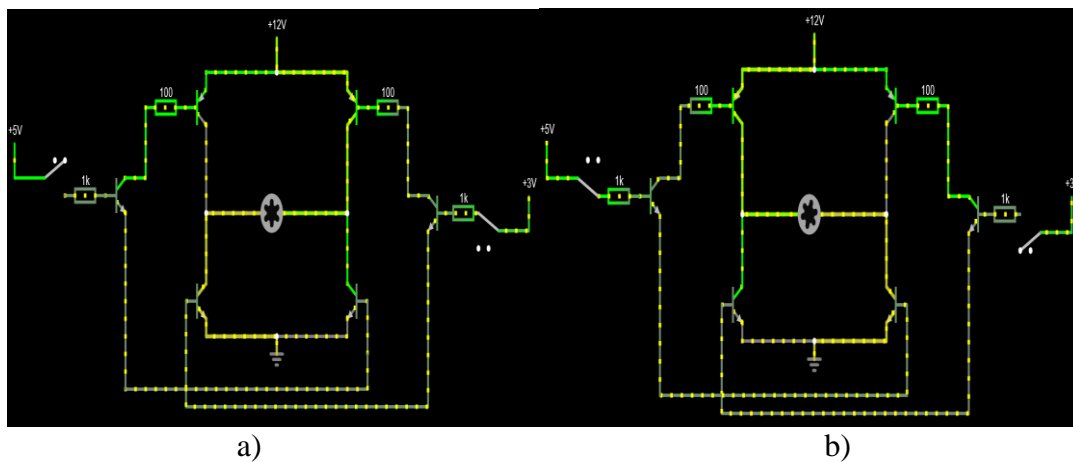


Figura 3.1.3: Puente H en funcionamiento a) sentido Horario, b) sentido Antihorario.

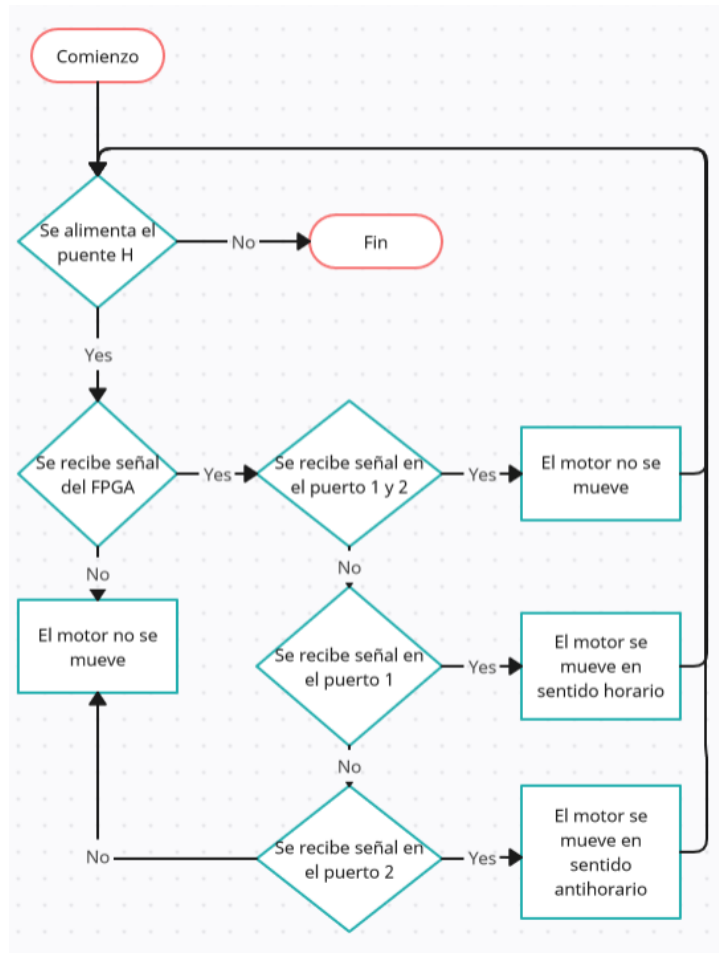


Figura 3.1.4: Diagrama de flujo del puente H.

El puente H se alimenta a través de tres fuentes distintas. Estas fuentes incluyen las 2 señales de modulación de ancho de pulso (PWM) generadas por el FPGA y la fuente de alimentación principal establecida a 12V. La dirección de rotación del motor se determina mediante el estado activo del puerto de salida de la señal PWM generada por el FPGA. Esto posibilita suministrar al motor un voltaje ajustado de 9V, con el cual fue calculado la señal PWM con la cual tiene un comportamiento lineal.

El motor está equipado con un encoder de cuadratura que suministra al FPGA dos señales. Estas señales son de forma rectangular y tienen el mismo periodo, pero están desfasadas en 90 grados. A partir de estas señales, es posible determinar cuando el sistema completa una vuelta, permitiendo calcular el desplazamiento del sistema y su dirección. Esto se logra mediante la interpretación lógica de ambas señales, que se dividen en cuatro casos según la relación entregada por el decoder, ver Tabla 3.1.1.

	Caso 1	Caso 2	Caso 3	Caso 4
B	0	0	1	1
A	0	1	1	0

Tabla 3.1.1: Casos del Decoder.

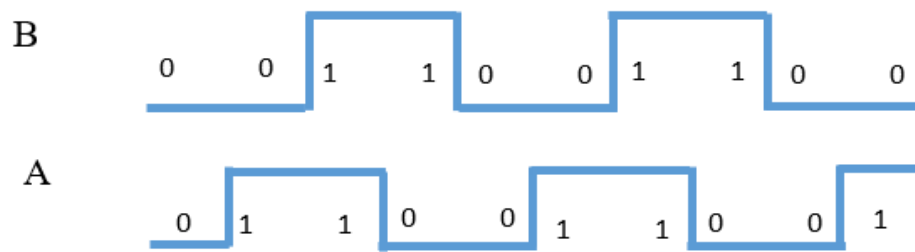


Figura 3.1.5: Señales A y B del encoder.

Para llevar a cabo el procesamiento de la información, el primer paso consiste en establecer un ciclo de muestreo que garantice la recuperación precisa de los datos en cualquier momento, esto es crucial, ya que la pérdida de información en alguno de los dos canales resultaría en una interpretación incorrecta, pues no se cuentan con otros sensores para corregir la información, cualquier error en la implementación sería crítico.

Para asegurar que el ciclo de muestreo evite la pérdida de información, se deben cumplir dos requisitos esenciales, en primer lugar, el ciclo debe ser lo suficientemente rápido para que incluso a la máxima velocidad del motor, la recolección de datos sea continua e ininterrumpida, además, el ciclo de muestreo debe permitir la obtención de al menos cuatro mediciones por ciclo, garantizando así una captura adecuada de la información.

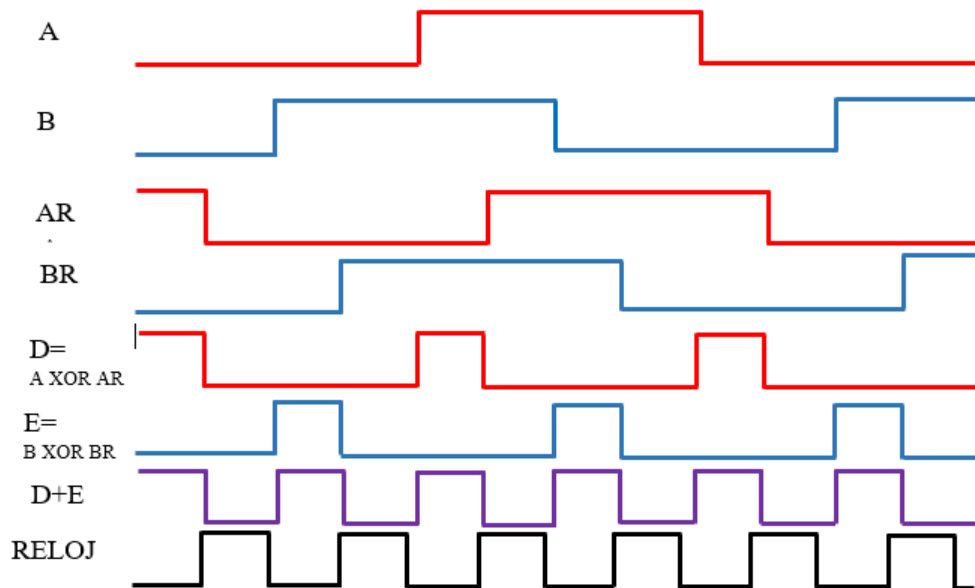


Figura 3.1.6: Diagrama de tiempos, donde se muestra la recuperación de la señal de reloj a partir de las señales A y B de un encoder.

La figura 3.1.6 se observan los pasos requeridos para obtener el reloj. En primer lugar, se capturan simultáneamente los datos de entrada A y B, luego, se introduce un retraso mediante la utilización de tres flip-flop, y a partir de esta instancia, se denominan estas

señales como AR y BR, posteriormente, se realiza una operación XOR con las señales originales y sus respectivos retrasos: A con AR y B con BR, estos resultados se etiquetan como D y E, respectivamente, después, se suman los resultados D y E, y se aplica un último retraso multiplicado por Delta t, el valor de Delta t es equivalente a un octavo del tiempo mínimo, este tiempo mínimo corresponde al inverso de la frecuencia máxima de las señales A y B cuando el motor opera a su velocidad máxima,

$$\text{FrecuenciamaxAB} = NV * V_{\text{max}} \quad (3.1.4)$$

donde NV corresponde a la cantidad de eventos necesarios de A y B para que se realice un periodo, y Vmax como la velocidad máxima del motor.

$$T_{\text{min}} = \frac{1}{\text{FrecuenciamaxAB}} \quad (3.1.5)$$

Ya con el tiempo mínimo se obtiene Delta T a partir de un octavo de este.

$$\Delta T = \frac{1}{8} T_{\text{min}} \quad (3.1.5)$$

En la figura 3.1.7 se pueden observar el circuito para obtener el reloj para poder recuperar la información.

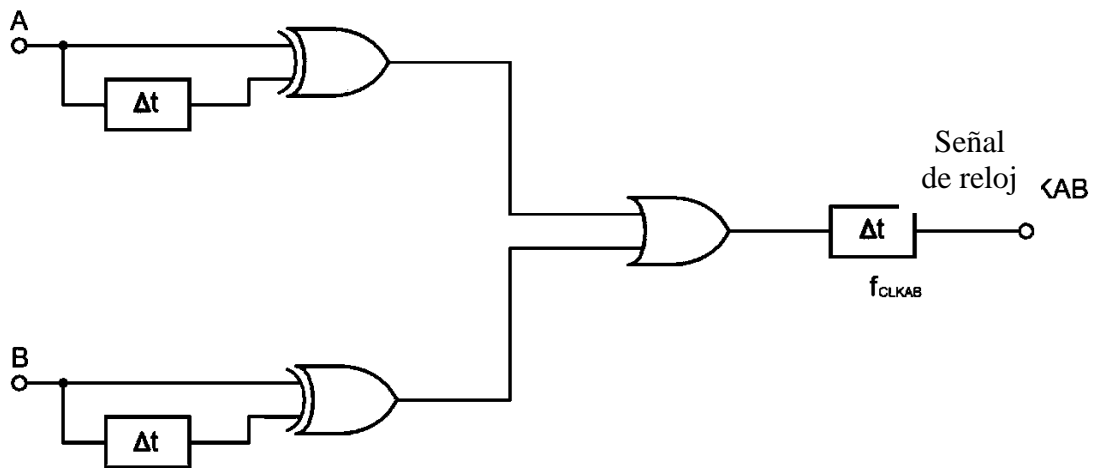


Figura 3.1.7: Circuito para recuperación del reloj.

La inclusión de los 3 flip-flop es para que nunca haya un momento en el cual la señal pase sin atraso pues ha habido casos en el que se logran sintonizarse 2 flip-flop, pero con 3 es imposible. Por otro lado, el que  $\Delta T$  sea un octavo de la mínima es para que tenga el tamaño máximo para suceder en cada una de las medidas de A y B a pesar del retraso. Si se trata de un motor con caja de engranajes es necesario tomar en cuenta la transformación para los cálculos siendo la salida tras esta la contada para los cálculos. Una vez culminado este proceso y con toda la información a nuestra disposición, es crucial interpretarla de manera adecuada. Esto se logra mediante el desarrollo de un programa que funcione en concordancia con los pasos que el motor lleva a cabo, acumulando los valores pertinentes. Al finalizar este

procedimiento, se obtiene una evaluación precisa del número de pasos realizados por el motor, lo que a su vez nos brinda la capacidad de conocer la distancia recorrida.

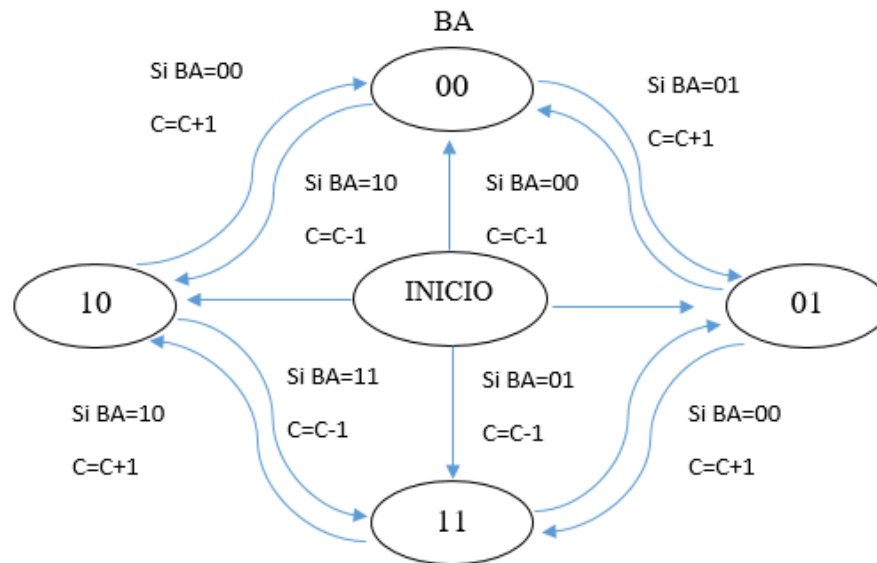


Figura 3.1.8: Diagrama de estados del decoder.

Además del encoder existe otra entrada al FPGA proveniente del sistema mecánico, los fines de carrera. Estos fines de carrera tienen la función de enviar una señal al FPGA cuando el sistema alcanza el límite del área de trabajo. Para lograr esto, se instala un interruptor en los extremos junto con una palanca. Al ser presionado por la mesa de trabajo, esta palanca cierra el circuito, generando así una señal lógica de valor 1. Para asegurar el funcionamiento correcto, se emplea un circuito, ver figura 3.1.9.

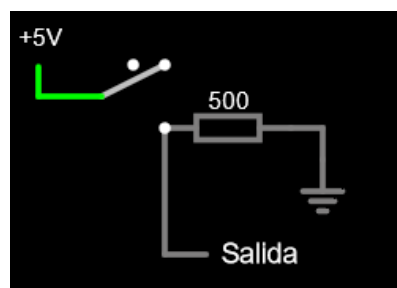


Figura 3.1.9: Circuito de switch.

En esta representación, se implementa una resistencia a tierra con el propósito de evitar interpretaciones incorrectas de los datos, esto garantiza que la señal se transmita exclusivamente cuando el interruptor está cerrado. Tanto la señal de fin de carrera como la de los encoders se conectan a la tarjeta de comunicaciones, la cual dispone de extensiones para los pines del FPGA. La información fluye hacia el FPGA, el cual convierte las señales analógicas entrantes en señales digitales, muestreándolas de acuerdo con el reloj del FPGA o el tiempo especificado para su lectura. En este caso, el reloj del FPGA opera a 100 MHz.

El microprocesador que funciona como la zona de operaciones del sistema en él se encuentra uno CPU (unidades centrales de procesamiento) que permiten realizar cálculos y operaciones lógicas, las 2 variantes más comunes son CISC (Complex Instruction Set Computer), los cuales son compiladores sencillos, código más compacto, facilita la depuración de errores y RISC (Reduced Instruction Set Computer), cada instrucción puede ser ejecutada en un solo ciclo del CPU, menos acceso a memoria, accesos a registros del micro, CPU más rápida.

La RAM (memoria de acceso aleatorio) se trata de la memoria volátil del sistema en ella se puede acceder a información temporal a una velocidad mayor de la que a datos que se encuentre en la memoria ROM, entre ellas existe la SDRAM se trata de una memoria sincrónica lo que permite el cambio del estado usando una señal de reloj, sincronizándolo con el bus del sistema.

Para el funcionamiento del sistema se cuenta con un Altera Cyclone V 5CEBA4F23C7N integrada a la tarjeta DE0-CV, el cual tiene equipado un procesador ARM con arquitectura RISC, la cual trabajara como nuestra CPU, y una SDRAM, pertenece a la familia Cyclone V E, cuenta con 49000 elementos lógicos, 484 pines y trabaja en temperaturas comerciales, cuenta con 3080 Kbits de memoria embebida y con una 64MB SDRAM x16 data bus original, el cual fue mapeado en proyectos anteriores en la MCEA para poderse emplear a 16MW x 32 [31]. Para el procesamiento de la información se tienen 2 bloques ADM que permite el acoplado de los datos recibidos en los puertos y su conversión para entrada digital como un arreglo de 32 bits que ingresa en el procesador, para que posteriormente al salir del procesador como un arreglo de 32 bits se pueda separar nuevamente en 16 puertos y convertido en analógico, en este caso la salida ira a un bloque que entregara la señal de PWM, los bloques en el Firmware se verán a continuación.

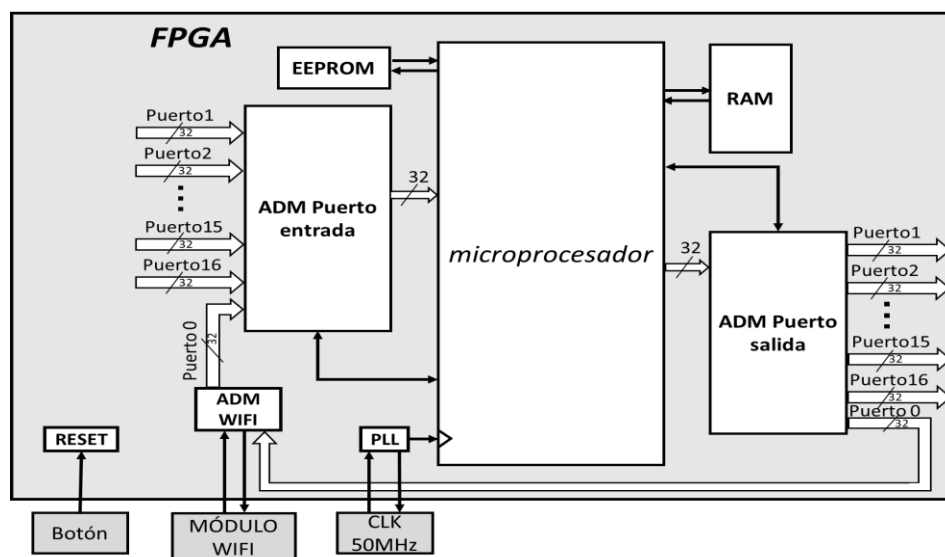


Figura 3.1.10: Diagrama del FPGA.

Para implementar lo expuesto previamente, se requiere la programación tanto del firmware como del software. El firmware se encargará de diversos aspectos, como traducir los pulsos del encoder de cuadratura en valores de posición. Además, utilizando el instante de cambio de la posición se obtiene información para el cálculo de la velocidad. Adicionalmente, el firmware incorporará el protocolo de comunicación destinado a trabajar con el módulo Wifly. También será responsable de generar la señal PWM para los motores, asegurándose de que la señal no sea enviada hasta verificar que el fin de carrera no esté activo. Este proceso de desarrollo se llevará a cabo utilizando el programa Quartus.

Por su parte, el software asumirá la tarea de leer los datos transmitidos desde la computadora al momento de la inicialización a través del Wifi. Posteriormente, llevará a cabo un proceso de "retorno a casa", en el cual los motores se moverán hasta que los dos fines de carrera designados como posición de inicio sean activados. Al concluir este procedimiento, se activará el control necesario para llevar a cabo la trayectoria predefinida en el software. Dependiendo de la programación, se enviarán datos relativos a posición, velocidad, torque, aceleración y jerk de los ejes que se encuentren en funcionamiento.

### 3.2 Diseño de la etapa de potencia

Para hacer funcionar los motores se requiere agregar una etapa de potencia que reciba las señales de dirección en forma digital y alimente con la energía necesaria al motor para su funcionamiento, además de aislar la parte digital del sistema, esa es la función del puente H, en esta ocasión se harán los preparativos para poder armar un puente H que trabaje con 10 A, por lo que se empezó por desarrollar los cálculos.

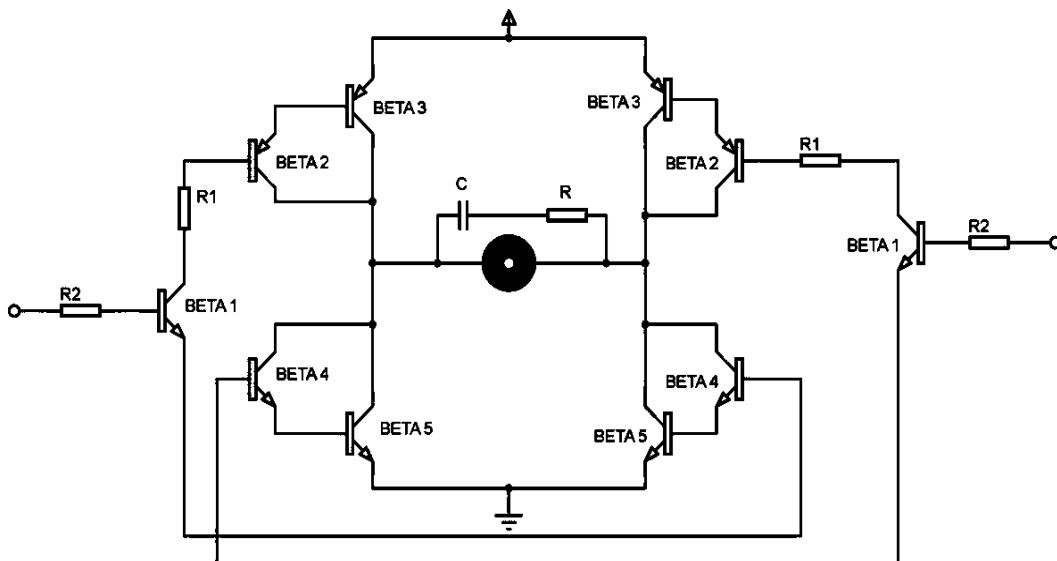


Figura 3.2.1: Diagrama general del puente H.

$$I_{motormax} = \frac{\beta_2 * \beta_3}{10} * I_{c1} \quad (3.2.1)$$

En la ecuación (3.2.1)  $I_m$  representa la corriente del motor,  $\beta_{2,3}$  las betas de los transistores en cascada y  $I_{c2}$  la corriente de colector en el transistor 1.

También conocemos el voltaje que puede darnos la tarjeta que es de 3.3v,

$$V1 = I_{b1}R2 + 0.7v + 0.7v + 0.7v \quad (3.2.2)$$

donde  $V1$  es el voltaje que sale del FPGA,  $I_{b1}$  la corriente de base de transistor 1 y  $R2$  la resistencia a la salida del FPGA.

Y por último sabemos cómo se relacionan ambas formulas por medio de la ganancia del transistor 1.

$$I_{c1} = \frac{\beta_1}{10} * I_{b1} \quad (3.2.3)$$

Con esto fue posible dar con las betas necesarias para el funcionamiento del puente H, únicamente se propone el valor de  $R1$  como 1Kohms por simplicidad y se despejan, el resto de los valores, la  $I_b=1.2mA$ . Para el transistor 1 se usará al típico 2n2222 con una Beta típica de 150 y como el transistor 3 al Tip35c con una beta típica de 40 por lo que se selecciona el 2SA1008 para ser el último transistor que permita los 10 Amp, sin embargo, por cuestión de existencias mejor se optó por el 2SB1143S con una Beta típica de 140, de esta forma el valor de la corriente del motor fue 10.08 Amp como se deseaba. Para los valores de la resistencia faltante se pueden despejar de la ecuación (3.2.4).

$$V_{CC} = 0.7v + 0.7v + I_{c1}R1 + V_{Cesat2} + 0.7v + 0.7v \quad (3.2.4)$$

Además, podemos hacer otra malla relacionada al voltaje de alimentación.

$$V_{CC} = V_{CESat} + V_{motor} + V_{ECsat} \quad (3.2.5)$$

Sustituimos  $V_{cc}$  de 3.2.5 en 3.2.4 y se despeja el valor de  $R1$ .

$$R1 = \frac{(0.7v + 0.7v + V_{Cesat2} + 0.7v + 0.7v - (V_{CESat} + V_{motor} + V_{ECsat}))}{I_{c1}} \quad (3.2.6)$$

Dando un valor de 117 ohms, para los transistores faltantes se usan los espejos o con beta similar siendo el Tip35c y el 2SD1498. El diagrama eléctrico que se obtiene se observa en la figura 3.2.2.

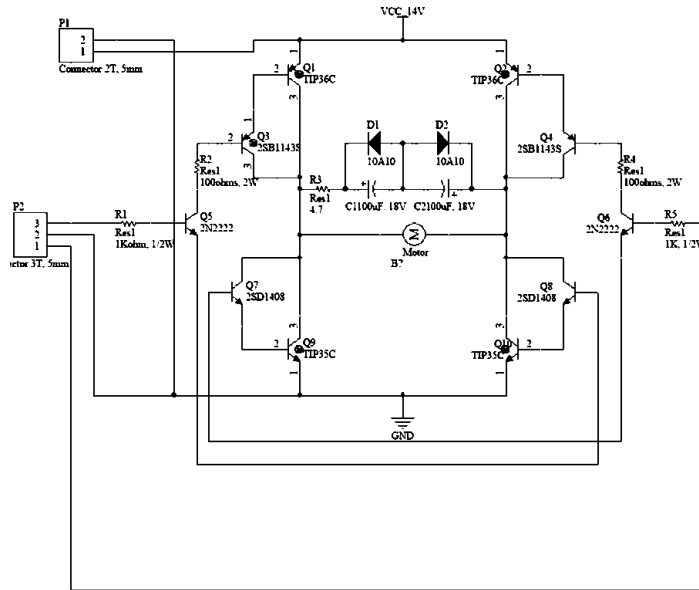


Figura 3.2.2: Diagrama eléctrico del puente H.

Se incorporó un circuito adicional en paralelo al motor para brindar protección contra picos de corriente y permitir la disipación de energía. En comparación con la versión anterior presentada en la figura 3.1.1, la cual tenía una representación simplificada del mismo disipador, esta nueva versión ofrece mejoras significativas. Este nuevo diseño es capaz de soportar picos de corriente más elevados y proporciona una mayor seguridad en su funcionamiento.

Utilizando el software Altium Designer, se procedió a diseñar la tarjeta de circuito impreso (PCB) del puente H, así como a crear los modelos de componentes que integran el circuito eléctrico. Entre estos componentes se incluyen diodos, capacitores y resistencias de potencia. Las dimensiones de estos componentes se validaron con las especificaciones proporcionadas en las hojas de datos de cada elemento. El diseño finalizado de la tarjeta se presenta en la figura 3.2.3.

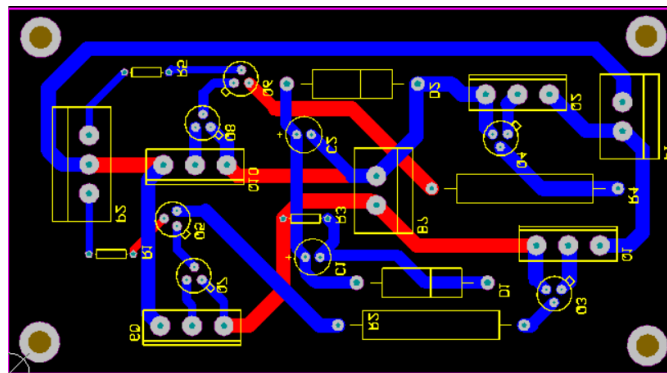


Figura 3.2.3: PCB del puente H.

Después de mandar a fabricar la tarjeta se colocaron cada uno de los componentes del circuito quedando como se muestra en la figura 3.2.4. Como no se requieren los 10A límite de corriente que a la que fue diseñada la tarjeta, no se ocupó el transistor de potencia.



Figura 3.2.4: Tarjeta implementada puente H.

### 3.3 Comunicación entre el robot y el sistema embebido

El sistema de adquisición de datos que se emplea para el control de este robot emplea una comunicación Wifi, por tal motivo se diseñó una tarjeta que nos permita tener este tipo de interfaz, esto con el objetivo que se acople directamente al FPGA. Se utiliza un módulo Wifly RN-171, además se agregaron extensiones de las terminales en el FPGA lo que permitirá tener una mejor conexión y garantizar la comunicación entre los dispositivos. El Wifi o estándar IEEE 802.11x, en donde la x denota la generación del estándar en cuestión es un tipo de comunicación inalámbrica de alta velocidad que nos permite enviar y recibir grandes cantidades de información. La IEEE (Institute of Electrical and Electronics Engineers) y ETSI (European Telecommunications Standards Institute) diseño el estándar 802.11 el cual fue aceptado en 1997, después para coordinar el desarrollo [32]. WiFi Alliance es una organización sin fines de lucro fundada en 1999 que certifica la interoperabilidad de dispositivos inalámbricos construidos considerando el estándar 802.11 y que también pretende promover y mejorar el estándar [33]. El estándar 802.11 provoco una revolución en las redes inalámbricas que está destinada a continuar. Aparte de los edificios y hogares, se ha empezado a instalar en plazas, trenes, aviones, barcos y automóviles de modo que las personas puedan navegar por Internet en cualquier parte a donde vayan [33]. El sistema de capas del protocolo 802.11 en donde se aprecia la equivalencia con el protocolo TCP/IP, ver figura 3.1.1.

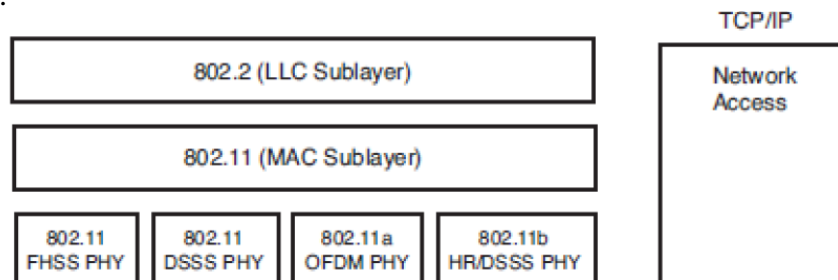


Figura 3.3.1: Capas del protocolo 802.11 y su equivalencia en el protocolo TCP/IP.

La capa inferior cuenta con diferentes opciones de formatos de transmisión inalámbrica:

- (FHSS) Frequency Hopping Spread Spectrum /Espectro Ensanchado por Salto de Frecuencia.
- (DSSS) Direct Sequence Spread Spectrum/Espectro ensanchado por secuencia directa.
- (OFDM) Orthogonal Frequency Division Multiplexing/ Multiplexación por División de Frecuencias Ortogonales.
- (HR/DSSS) High Rate Direct Sequence Multiplexing.

El módulo que se va a emplear es Wifly RN-XV con la capacidad de contarse con 802.11 b/g, es decir que es compatible con la versión b y g. El estándar 802.11b funciona en la banda de 2,4 GHz. Debido al espacio ocupado por la codificación del protocolo CSMA/CA, en la práctica, la velocidad máxima de transmisión con este estándar es de aproximadamente 5,9 Mbit/s sobre TCP. Este utiliza la banda de 2,4 Ghz (al igual que 802.11b), pero opera a una velocidad teórica máxima de 54 Mbit/s, que en promedio es de 22,0 Mbit/s de velocidad real de transferencia, similar a la del estándar 802.11a. Es compatible con el estándar b y utiliza las mismas frecuencias. Sin embargo, en redes bajo el estándar g la presencia de nodos bajo el estándar b reduce significativamente la velocidad de transmisión.

Estándar	Alcances	Ancho de Banda [mbps]	Técnica de difusión
<b>802.11</b>	Define estándar para WLAN para capa PHY y MAC	2	FHSS, DSSS, IR
<b>802.11a</b>	Define una capa física de alta velocidad en la banda de 515-5.825 GHz.	6,12,24 opcional 54	OFDM con 24 subportadoras
<b>802.11b</b>	Define una capa física de alta velocidad en la banda de 2.4 GHz.	11	HR/DS, HR/DSSS
<b>802.11e</b>	Mejora del estándar original para implementar QoS (aplica a 802.11a/b/g)		
<b>802.11g</b>	Define una tasa de datos más alta en la capa física de 2.4 GHz.	22-54	ERP
<b>802.11h</b>	Define funciones MAC para permitir a equipos 802.11a cumplir con los requerimientos europeos.	6,12,24 opcional 54	OFDM
<b>802.11i</b>	Mejora de la capa MAC para proveer seguridad en 802.11a/b/g.		

Tabla 3.3.1 Variantes de IEEE 802.11[34].

Con el fin de construir la tarjeta primero se necesitó tomar las medidas lo más exactas posibles con un vernier, esto debido a que el objetivo es poderla colocar soportándose con los pies y con un tornillo en la parte inferior de forma que sea estable pequeña para que entre en la base original para la electrónica del robot la cual está compuesta de acrílico. Las medidas que se obtuvieron fueron.

- Medida entre los 2 segmentos de terminales del FPGA el cual es de 15.86mm
- Anchura de los 2 bloques de terminales de la tarjeta con el FPGA el cual vale 4.93mm
- Largo de fila de terminales de la tarjeta con el del FPGA el cual es de 50.8mm
- Característica de medida del módulo Wifly
- Distancia entre la hilera de terminales al tornillo la cual es de 15mm

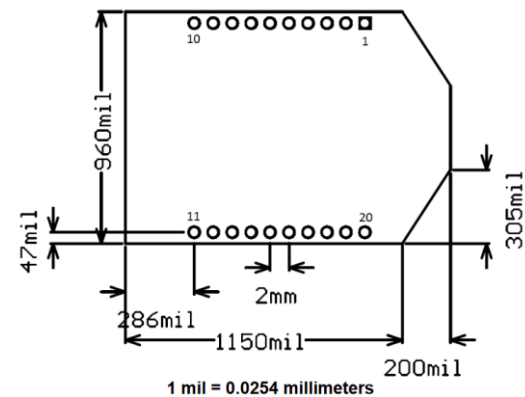


Figura 3.3.2: Medidas del módulo Wifly.

A partir de estos datos se hace la preparación de las 2 tiras de terminales y se usa un módulo con los mismos terminales para tener la conexión ya preparada al Wifly, debido a eso se tiene que el diagrama en la figura 3.3.2. Para el módulo Wifly es necesario una librería para poder colocarse en el diagrama eléctrico donde contenga los diversos terminales que tiene el Wifly. Una vez está terminado se puede crear un diagrama eléctrico, en el cual por medio de etiquetas se colocaran los nombres de las entradas internas del FPGA en cada uno de las hileras de terminales (GPIO-0 y GPIO-1), estas se pueden encontrar en el manual de la tarjeta que contiene el FPGA, cada uno de los GPIO cuenta con una salida de 5V y una de 3.3V con su tierra respectivamente, en total el FPGA da la opción de usar un total de 80 terminales 40 por cada GPIO, en el diagrama se puede observar como el módulo Wifly se alimenta con el voltaje de 3.3V de GPIO 0 y está conectado a sus últimos 2 terminales 39 y 40, los cuales T17 (T39) conectado al Uart\_Tx del módulo y T15 (T40) al Uart\_Rx para fines de comunicación.

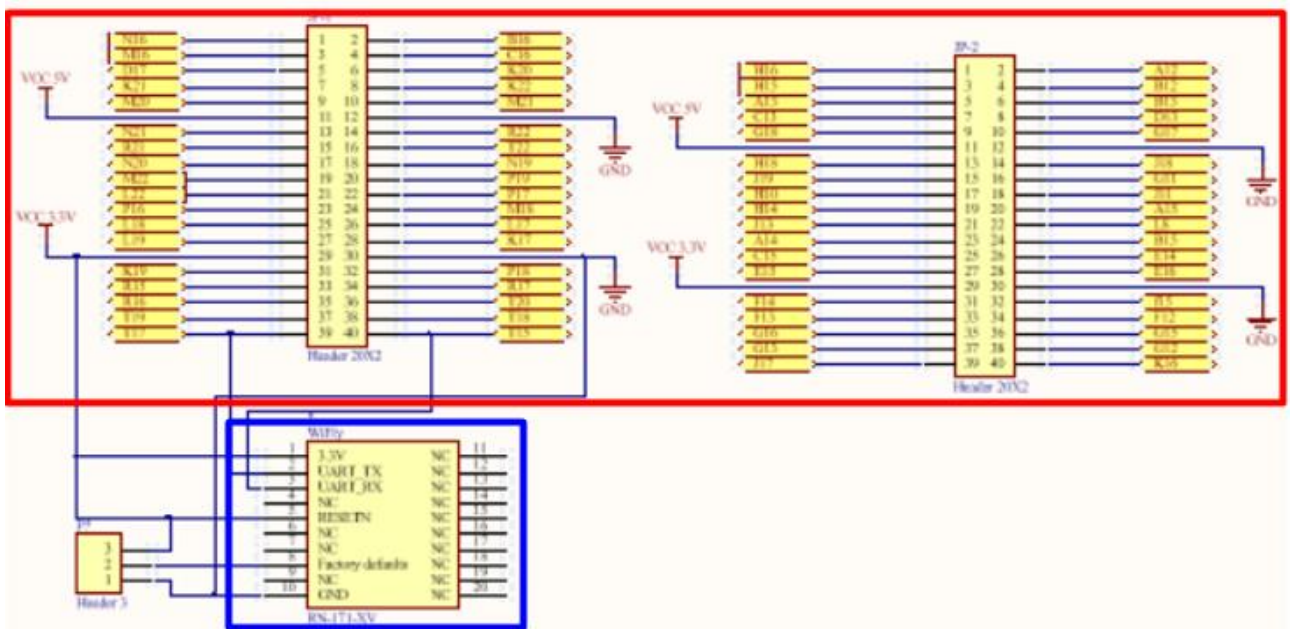


Figura 3.3.3: Diagrama eléctrico de la tarjeta de comunicación Wifi.

En la figura 3.3.3 el recuadro rojo contiene un Header asociado a las terminales que se conectan al FPGA, siendo el de la izquierda el GPIO-0 (JP-1) y el de la derecha GPIO-1 (JP-2), el recuadro azul contiene al módulo Wifly.

Se relaciona una la librería del Wifly a un footprint que igualmente es diseñado con sus dimensiones. El impreso se ve en la figura 3.3.4.

En los recuadros 1 están los headers que se unen al FPGA por medio de conectores macho hembra manteniendo el orden del diagrama eléctrico, el izquierdo es el GPIO-0 y el derecho el GPIO-1, ambos con 40 terminales cada uno, están directamente conectados a múltiples entradas que se encuentran en los recuadros 2, cada uno a la que se encuentra a su derecha, en esta las entradas tienen una dimensión de 2.034mm, estos se encuentran numerados y en el mismo orden de izquierda a derecha que las hileras, el recuadro 3 contiene al Wifly que ya se encuentra listo para que al montarse pueda emplearse, hay un header con 3 entradas este se encuentra para restaurar el Wifly en caso de ser requerido sin desmontar.

Para la conexión se requieren 8 señales de entrada, las cuales corresponden a la lectura de los encoder que van de T7 a T10, para los fines de carrera que van de T13 a T16 en él JP-1 y 4 de salida siendo los PWM, van del T31 al T34, también se conectarán 3 GND, en la misma tarjeta se conectan las tierras y alimentación al módulo de conexión y se conectan en T39 para escritura y T40 para lectura.

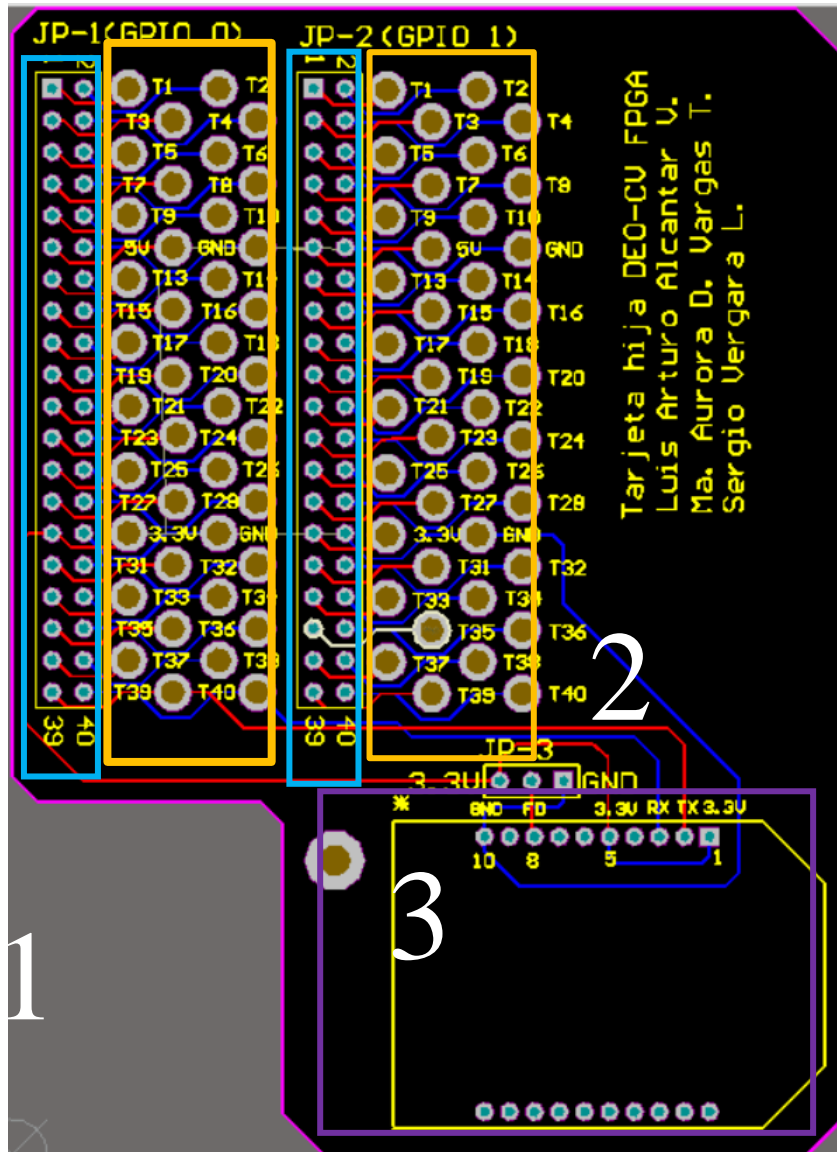


Figura 3.3.4: PCB terminado denominado tarjeta de comunicación Wifi.

PT1	PT1[32] No Usar	PT1[31...1]P. Encoder	A	B
FPGA Pins			K_21	K_22
Headers Pins			GPIO_7	GPIO_8
PT2	PT2[32] No Usar	PT2[31...1]P. Encoder	A	B
FPGA Pins			M_20	M_21
Headers Pins			GPIO_9	GPIO_10

Tabla 3.3.2 Puertos de entrada

PT1	PT3[32] No Usar	PT3[31...1]PWM	PWMI	PWMD
FPGA Pins			F_14	F_15
Headers Pins			GPIO_31	GPIO_32
PT2	PT4[32] No Usar	PT4[31...1]PWM	PWMI	PWMD
FPGA Pins			F_13	F_12
Headers Pins			GPIO_33	GPIO_34

Tabla 3.3.3: Puertos de entrada y Salida con Bus de datos.

La señal de los fines de carrera no contiene bus de datos solo se manejan como interrupciones. Una vez terminado se conecta a la tarjeta y se monta en el acrílico.

### 3.4 Diseño del firmware

A continuación, se coloca una imagen del firmware completo y se hará una breve explicación de la tarea que realiza cada sección.

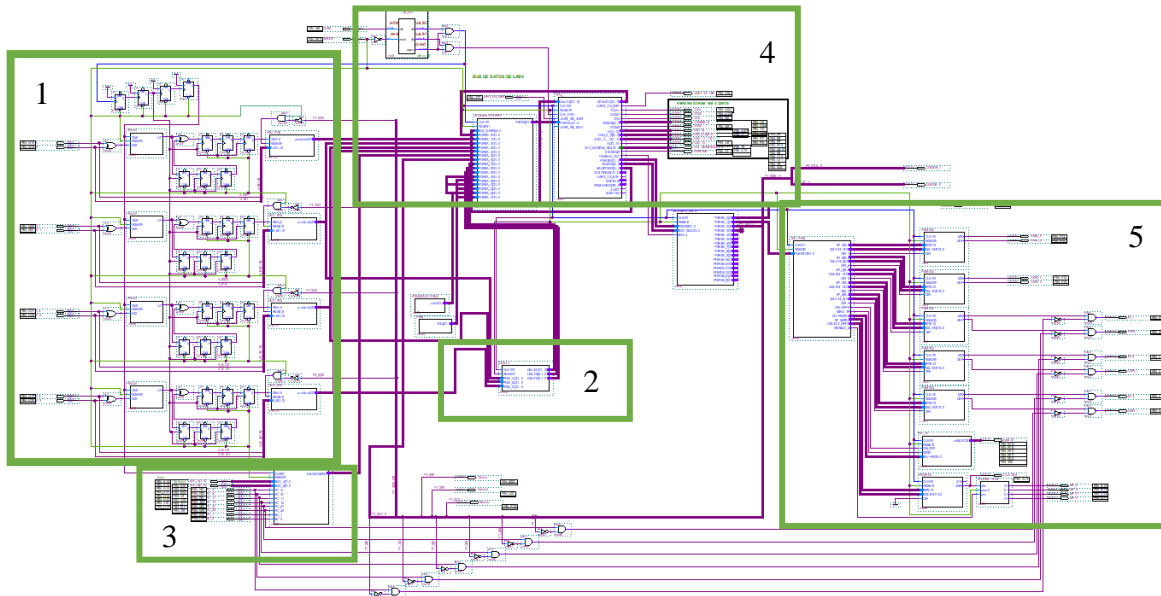


Figura 3.4.1: Firmware del sistema.

El firmware se estructura en 5 secciones distintas. En la sección 1 los decodificadores de cuadratura, los cuales posibilitan la recuperación de la posición a partir de las señales A y B mediante un contador, en esta sección se integran cuatro Decodificadores. En la 2 se dedica al cálculo del Delta t de la velocidad, utilizando el registro del tiempo transcurrido entre los flancos de posición. La 3 se centra en un convertidor Analógico-Digital (ADC) y un lector

de fin de carrera. Esta combinación permite capturar la señal y almacenarla en un bus de datos. En la 4 se encuentran el Multiplexor (MUX) y la Unidad de Procesamiento Central (CPU), los cuales se encargan de organizar los datos recibidos y llevar a cabo las operaciones necesarias. La 5 es el generador de modulación por ancho de pulso (PWM), construye la señal destinada al puente H para el control del motor. Cada una de estas secciones se detallará a continuación.

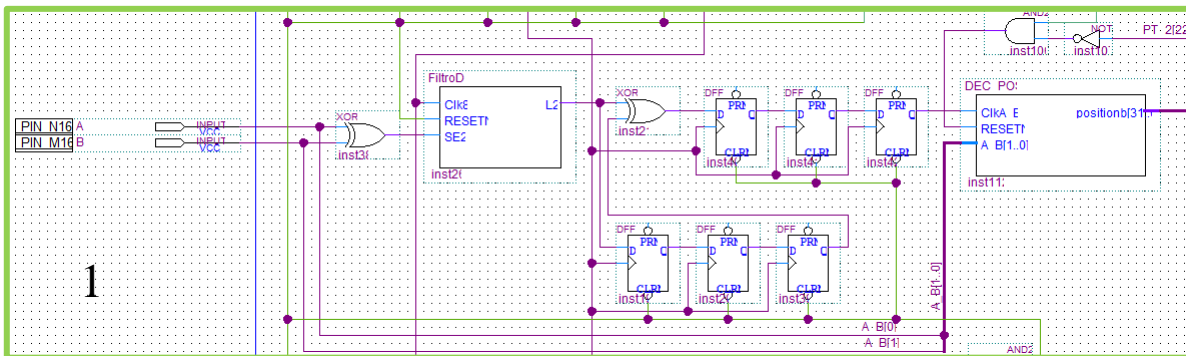


Figura 3.4.2: Detector de posición.

Los detectores de posición son 4 cada uno es como se ve en la figura 3.4.2 y cuenta con entradas diferentes, cada par de entradas corresponde a las señales A y B del respectivo motor. Estas dos señales se utilizan conjuntamente para el Decodificador. En la figura 3.4.2 se presenta un ejemplo de este tipo de Decodificador. La función XOR se emplea para cotejar las dos señales de entrada, y esta señal de comparación ingresa a un bloque que funciona como filtro. Este bloque de filtro se utiliza para evitar que el ruido sea interpretado como datos válidos. Luego de esta etapa, se efectúa un nuevo XOR utilizando la misma señal, pero con un desfase. El resultado es una señal asincrónica que señala los momentos en los que es posible detectar el cambio en la posición. El bloque final en esta sección recibe la señal derivada, las dos señales entregadas por el motor y el reset del sistema. Este bloque se encarga de procesar los datos y realizar el conteo que se traducirá en el Valor de Posición (VP) el cual ira a modulo Multiplexor que permite el acceso al CPU para el uso de los datos recuperados.

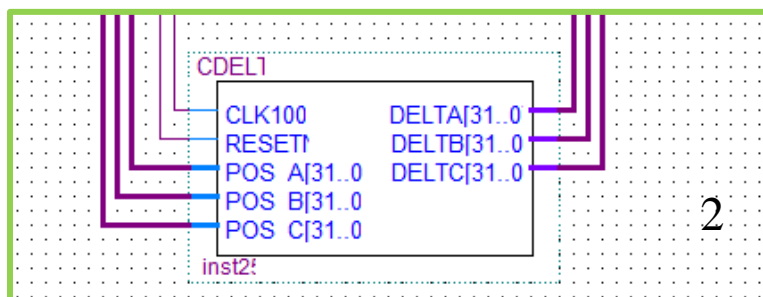


Figura 3.4.3: Calculo de delta T.

El cálculo de la Delta T para la velocidad se da en este bloque, este recibe como entrada la posición y realiza un conteo hasta vuelva a cambiar la posición en ese momento se reinicia y repite el ciclo hasta que el conteo sea tan grande que se interprete como que se

ha detenido el sistema, este bloque fue realizado para funcionar con 3 posiciones, aunque únicamente se requieren 2 para este proyecto.

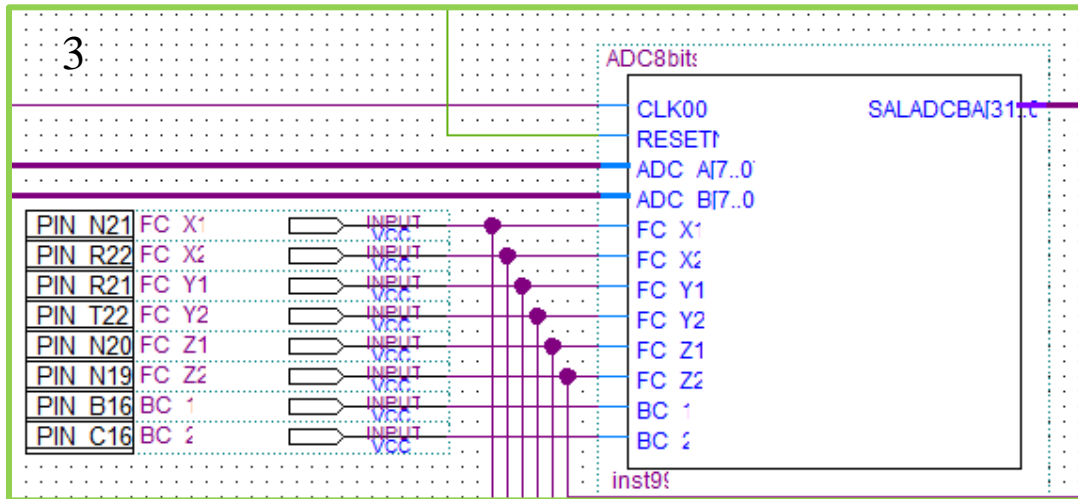


Figura 3.4.4: ADC y fines de carrera.

El bloque de ADC sirve para recibir señales analógicas y convertirlas en digital, entre estas señales se encuentran los fines de carrera en total se cuentan con 2 entradas ADC de 8 bits y 6 fines de carrera, estos se encuentran ordenados en pares para cada eje, además se cuentan con otras 2 entradas llamadas BC, todos estos valores se ordenan dentro de un vector de 32 bits y se envían a un multiplexor.

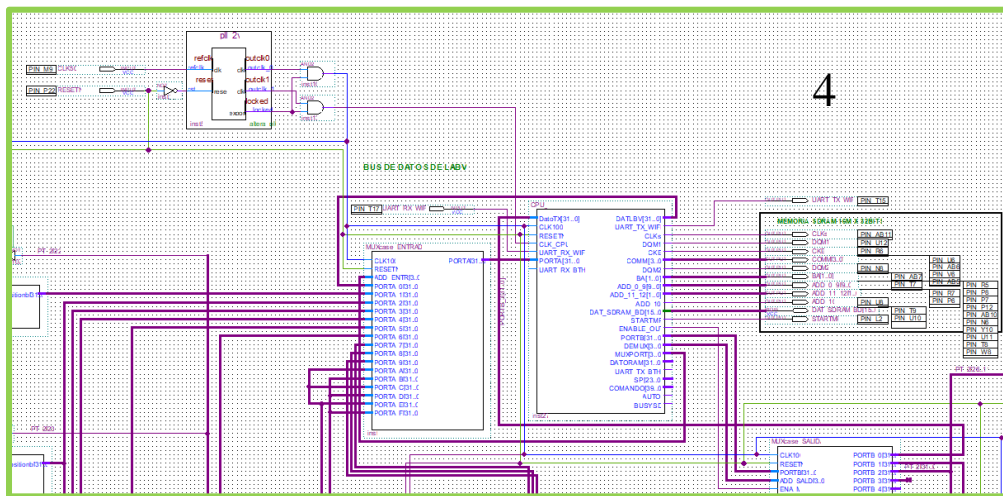


Figura 3.4.5: MUX y CPU.

Además del MUX y CPU en la figura 3.4.5 se observa la memoria SDRAM, además un PLL que divide el reloj del sistema en 3, 1 de 50 Mhz y 2 de 100Mhz, estos pasan por una compuerta AND para evitar desfases, una de las salidas va a dar el bloque multiplexor, este se encarga de que a la CPU ingresen los datos de cada entrada en secuencia para poder ser procesados, como entrada ingresan la salida del bloque para calcular la posición y las deltas

que requieren para calcular la velocidad, los fines de carrera y algunas constantes, la CPU trabaja en conjunto con la Memoria SDRAM y van a dar a otro multiplexor.

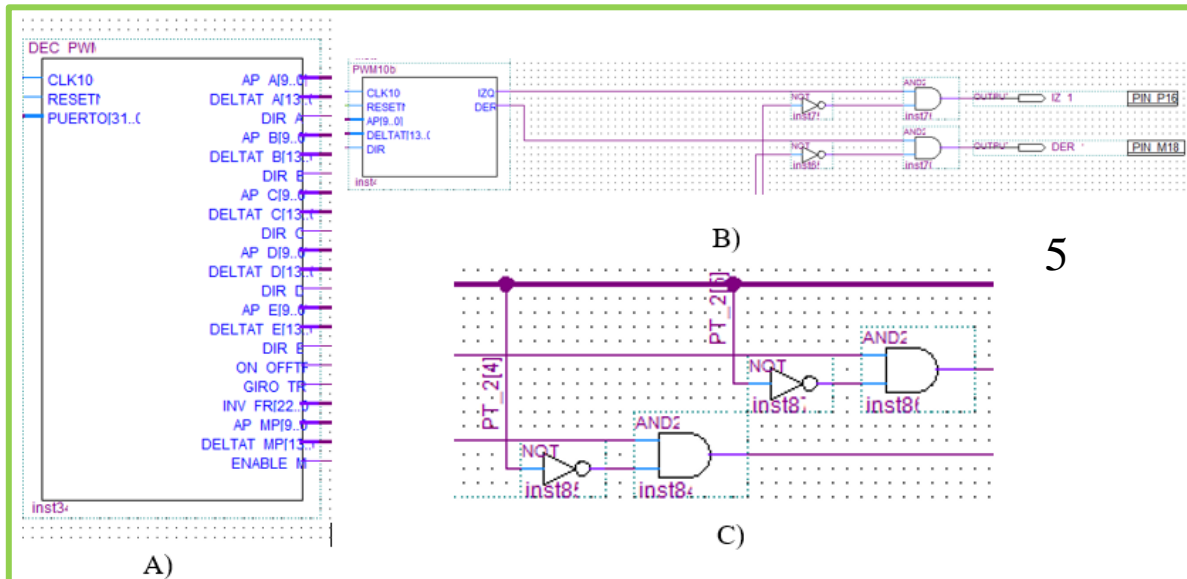


Figura 3.4.6: DEC PWM, generador de PWM y circuito de seguridad.

Se tienen 4 PWM se puede observar uno por separado en la figura 3.4.6 junto un bloque DEC PWM y un circuito de seguridad, para su funcionamiento entra primero al bloque DEC PWM que funciona como multiplexor de salida que separa el arreglo de 32 bits en diferentes señales, las salidas Ap, DeltaT y Dir son las que van a dar a bloques individuales de PWM que tienen 2 salidas, se tienen 4 bloques individuales uno para cada motor, la salida de estos bloques individuales presentase el comportamiento de una señal PWM y dependiendo la salida que esté activa será la dirección de giro que se indicara en el puente H. Para que los fines de carrera sea funcionales aun cuando haya problemas con el software se emplearon compuertas lógicas para que se compruebe de forma constante si el fin de carrera está activo y en caso de estarlo no permita que la señal de PWM salga del FPGA.

### 3.5 Diseño del software

El software se trata de un lenguaje de programación de alto nivel. Fue concebido para simplificar el proceso de diseño mediante el uso de comandos que permiten ejecutar múltiples instrucciones a nivel ensamblador. Sin embargo, es importante señalar que no puede ser directamente integrado en un dispositivo, debe ser procesado por un compilador que lo transforma en un lenguaje de nivel inferior, para su descarga. Para esta tesis, se optó por emplear el código D desarrollado en la Maestría de Ciencias en Electrónica, opción en Automatización (MCEA) [35]. Esto permite trabajar directamente con las direcciones de la memoria RAM y genera un código que ofrece una comprensión clara de los procesos realizados.

Los pasos programados en el sistema se presentan en la figura 3.5.1. En él se pueden observar la inicialización de las direcciones de la RAM, la asignación de registros, la función

de retorno a la posición de inicio utilizando los fines de carrera, la generación de una señal PWM con un ancho de pulso fijo, la descarga de constantes recibidas a través de Wifi y la señal para reiniciar el valor de la posición enviada al firmware. Además, se detallan los pasos para la aplicación de un sistema de control que permite solicitar una trayectoria o posición específica. Durante la implementación del control, se estima la velocidad, la aceleración y el jerk.

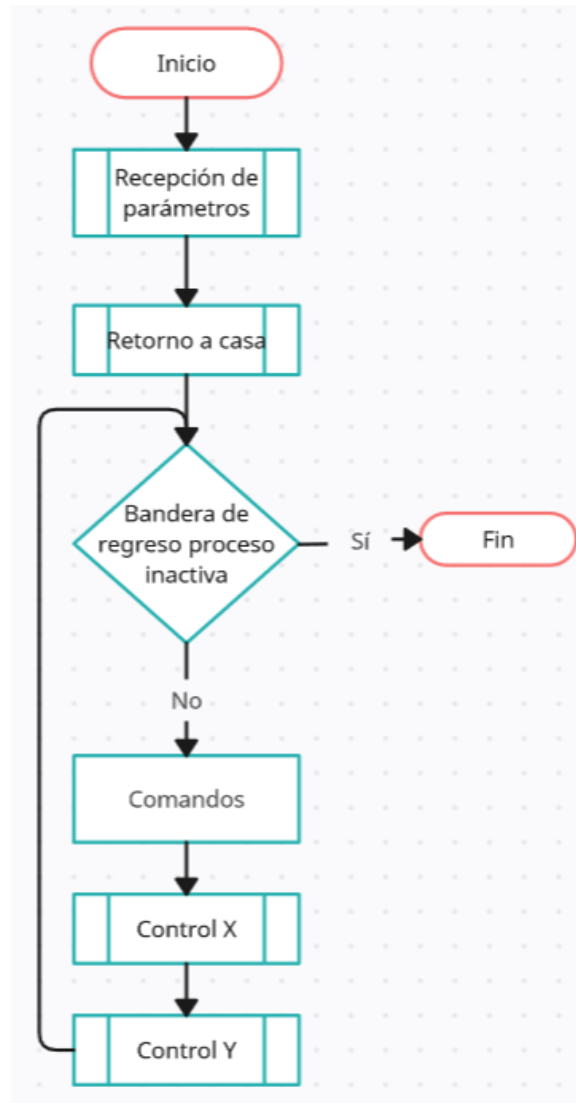


Figura 3.5.1: Diagrama de flujo del software.

La subrutina de recepción de parámetros de inicio tiene 2 partes en la primera se lee el dato recibido por el wifi y se extrae el número que corresponde al proceso, yendo del 1 al 16 como se muestra en la tabla 3.5.1, en la segunda se extraen los bits más significativos los cuales están en los impares y los menos significativos que se encuentran en los pares.

<i>Proceso</i>	<i>Respuesta</i>
<b>Lectura Bits [32..1]</b>	XXXX XXXXH
<b>Bits [32..1] AND 3FFF 0000H</b>	Bits[31...16] 0000H
<b>Bits[31...16] 0000H DIV 0001 0000H</b>	0000 Bits[31...16]H
<b>Valor de Etiqueta</b>	0000 Bits[31...16]H

Tabla 3.5.1: Tabla de Lectura de Bits.

Etiqueta	Bits [16..1]	Parámetro	Significado
1	Parametro1[32..17]	Para1[32..1]->RAM 1	Posición Deseada X
	Parametro1[16..1]		
2	Parametro2[32..17]	Para2[32..1]->RAM 2	Kp para X
	Parametro2[16..1]		
3	Parametro3[32..17]	Para3[32..1]->RAM 3	Ks para X
	Parametro3[16..1]		
4	Parametro4[32..17]	Para4[32..1]->RAM 4	Posición Deseada Y
	Parametro4[16..1]		
5	Parametro5[32..17]	Para5[32..1]->RAM 5	Kp para Y
	Parametro5[16..1]		
6	Parametro6[32..17]	Para6[32..1]->RAM 6	Ks para U
	Parametro6[16..1]		
7	Parametro7[32..17]	Para7[32..1]->RAM 7	Posición deseada Z
	Parametro7[16..1]		
8	Parametro8[32..17]	Para8[32..1]->RAM 8	Kp para Z
	Parametro8[16..1]		
9	Parametro9[32..17]	Para9[32..1]->RAM 9	Ks para Z
	Parametro9[16..1]		
10	Parametro10[32..17]	Para10[32..1]->RAM 10	Base
	Parametro10[16..1]		
11	Parametro11[32..17]	Para11[32..1]->RAM 11	Base
	Parametro11[16..1]		
12	Parametro12[32..17]	Para12[32..1]->RAM 12	Base
	Parametro12[16..1]		
13	Parametro13[32..17]	Para13[32..1]->RAM 13	Herramienta
	Parametro13[16..1]		
14	Parametro14[32..17]	Para14[32..1]->RAM 14	Herramienta
	Parametro14[16..1]		
15	Parametro15[32..17]	Para15[32..1]->RAM 15	Herramienta
	Parametro15[16..1]		
16	Parametro16[32..17]	Para16[32..1]->RAM 16	Free MP
	Parametro16[16..1]		

Tabla 3.5.2: Mapa de etiquetas.

Al terminar la recepción de parámetros se empieza la subrutina de regreso a casa en la cual el sistema empieza el movimiento usando las señales de PWM que ingresan al puente H y permiten el accionar de los motores en dirección negativa hasta que se detecte que ambos ejes activen la señal del fin de carrera que detectara el FPGA se saldrá del ciclo, para el código D se emplearán dos registros bandera en 0 y que solo cambiaran a 1 en el instante que se active el regreso a casa, además de un tercer registro que sirva para detectar la llegada de ambos y sirva para realizar un reset para algunos módulos del firmware.

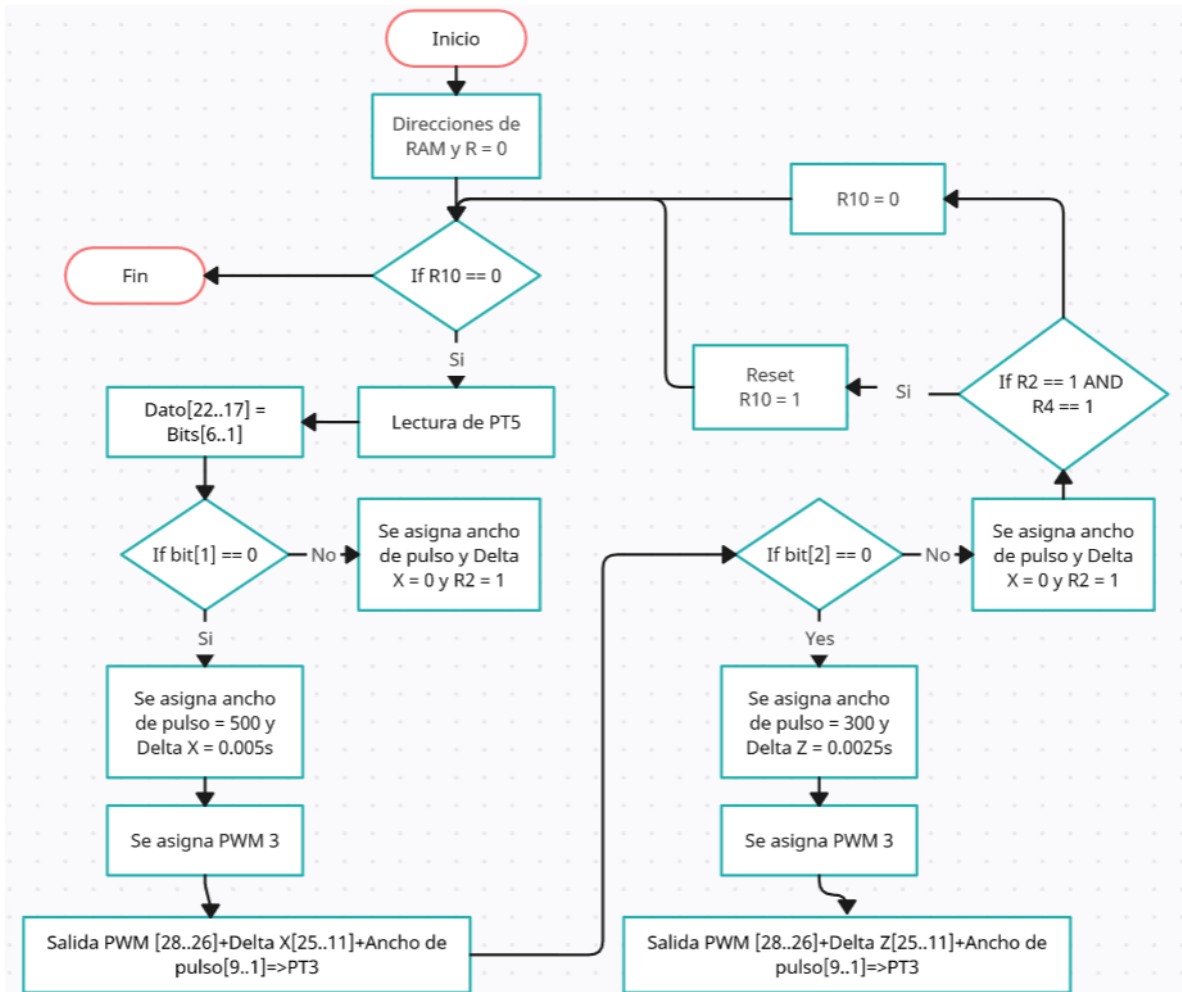


Figura 3.5.2: Diagrama de flujo de regreso a casa.

A continuación, se realiza una lectura de comandos, para hacerlo se reciben por medio del Wifi 32 bits y se extraen los bits[31..27] en Dato[5..1], mientras el Dato sea menor a 16 se enviara el parámetro en la RAM con el mismo número del Dato por medio del wifi, si Dato es 17 se acaba la lectura de comandos, si es mayor a 17 y menor a 27 los bits[25..1] se guardan en una dirección según en Dato recuperado, al ser mayor a 27 se acaba la lectura de comandos.

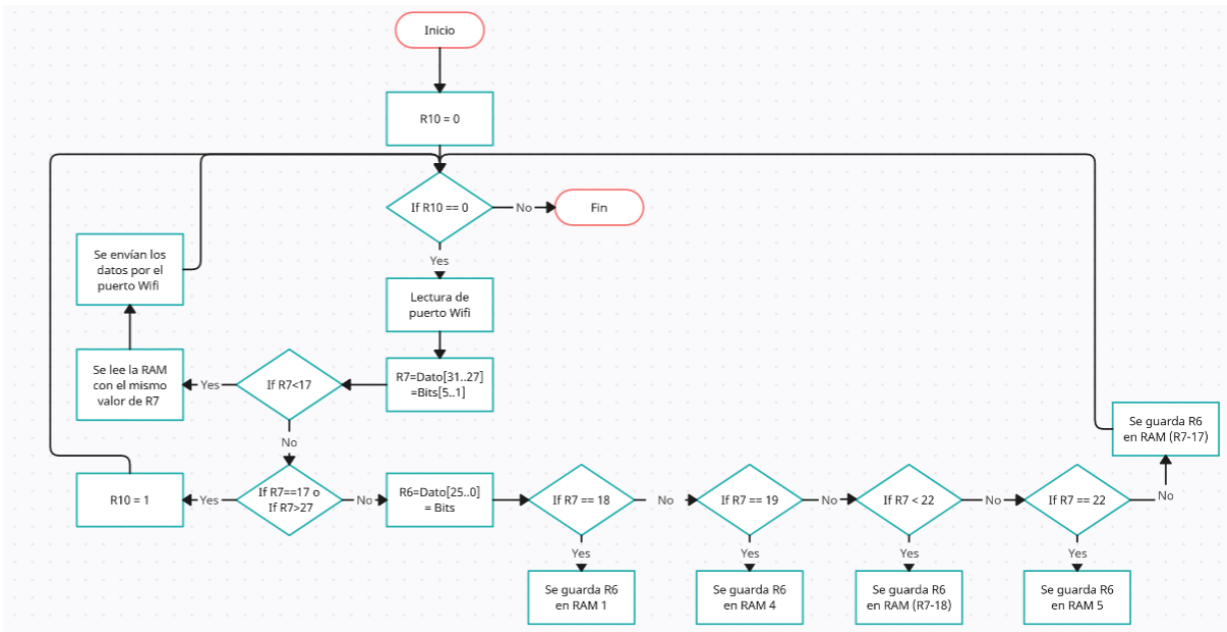


Figura 3.5.3: Diagrama de flujo de lectura y aplicación de comandos.

Por último se ingresa al módulo de control, en este se calcula la posición deseada y se usa el conteo que es proporcionado por el firmware para calcular la posición actual del sistema, a continuación, se calcula el error, después empleando la  $k_p$  y  $k_s$  que se recibieron por medio del wifi en el proceso anterior se compensa, luego con el contador del firmware se realiza una subrutina que será explicada en el siguiente capítulo para estimar la velocidad, aceleración y jerk, para después emplear otra subrutina que calcular la acción derivativa, esta se suma la compensación que se calculó con el error y se hace la conversión a PWM, se asigna el puerto de salida y verifica el signo del error, para asignarlo.

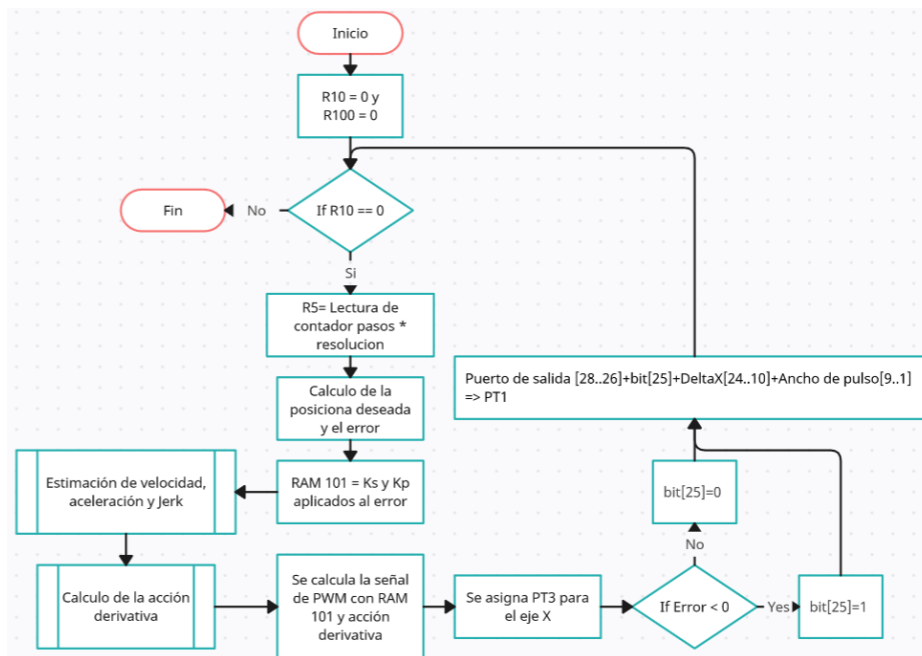


Figura 3.5.4: Diagrama de flujo del control X.

Además del Lenguaje D se desarrolló en el Posgrado un compilador para pasar los procesos a lenguaje máquina y ser descargados en el FPGA, este compilador fue realizado en el programa de Labview en el Laboratorio de Robótica y Control en la BUAP, en este cargamos las instrucciones y el programa los transforma a ensamblador y lenguaje máquina, esta transformación es mostrada al usuario y se pone a disposición de este la dirección donde se guardan los datos del lenguaje máquina.

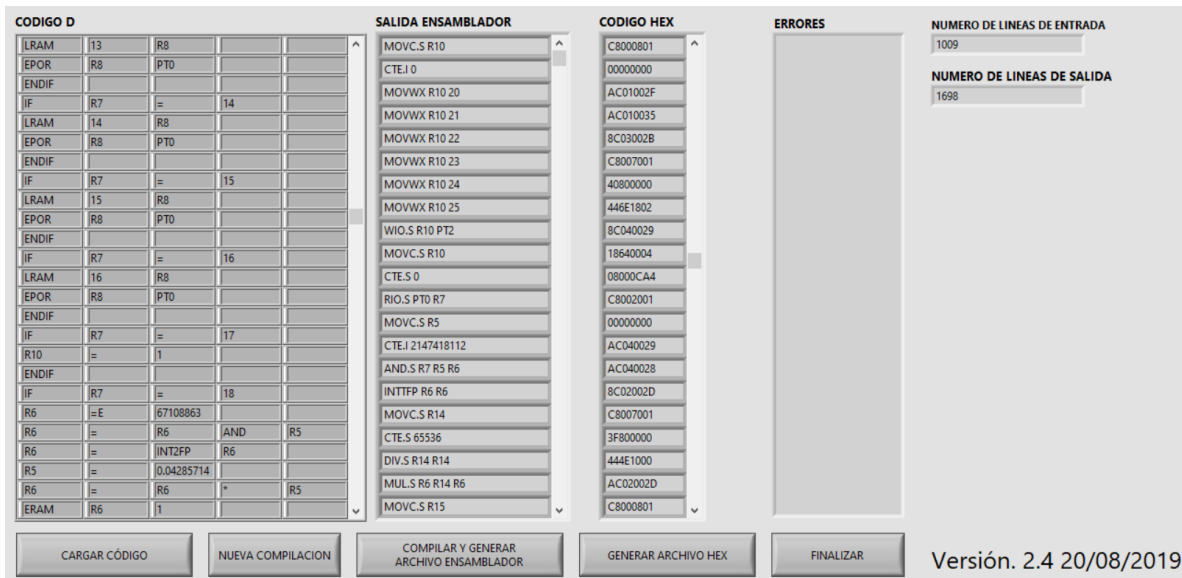


Figura 3.5.5: Compilador de lenguaje de programación de alto nivel a lenguaje máquina.

Una vez se cargan el programa debemos poder monitorearlo y enviar datos, esto se puede por medio de la comunicación Wifi que es posible gracias al módulo Wifly en el FPGA y una Interfaz de usuario desarrollada en LABVIEW en el posgrado, sin embargo, es importante recordar que, aunque está diseñado para trabajar en 3 dimensiones únicamente podemos trabajar con 2 grados de libertad. Esta cuenta con 3 pestañas para diferentes sus cada uno, durante todas se mantiene la parte derecha en la cual podemos ver si la comunicación se encuentra activa, representada por la luz verde o si no hay conexión representada por una luz roja, además de eso está la opción de terminar que desconecta del dispositivo. La primera pestaña es la que carga el código ya en lenguaje máquina, que nos fue proporcionado por el programa anterior, en el FPGA además se puede ver una barra que nos informa del proceso del envío.

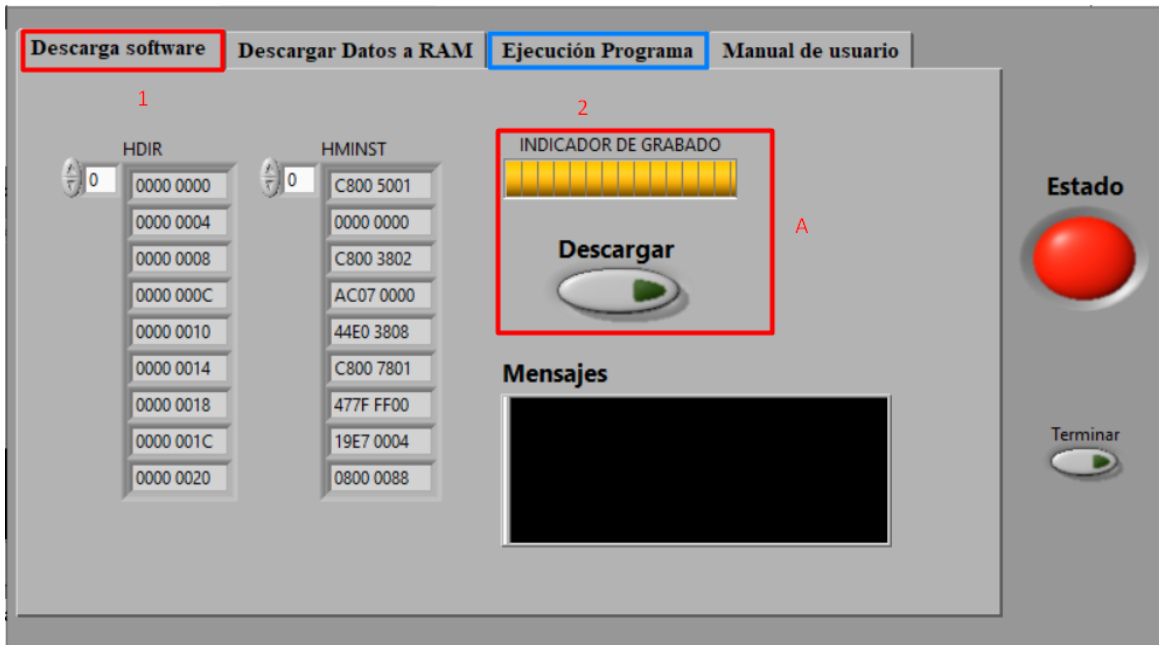


Figura 3.5.6: Interfaz de usuario pestaña 1.

En la 3ra pestaña nos es útil cuando el sistema ya está conectado y con los datos necesarios para iniciar el programa, en la subpestaña 1 podemos mandar la información inicial y arrancar el sistema, usando el botón de Inicio podemos enviar la información que se encuentra en la sección B o detenerlo, en la sección B se rellenan las casillas con los datos de posición que deseamos llegar durante las pruebas.

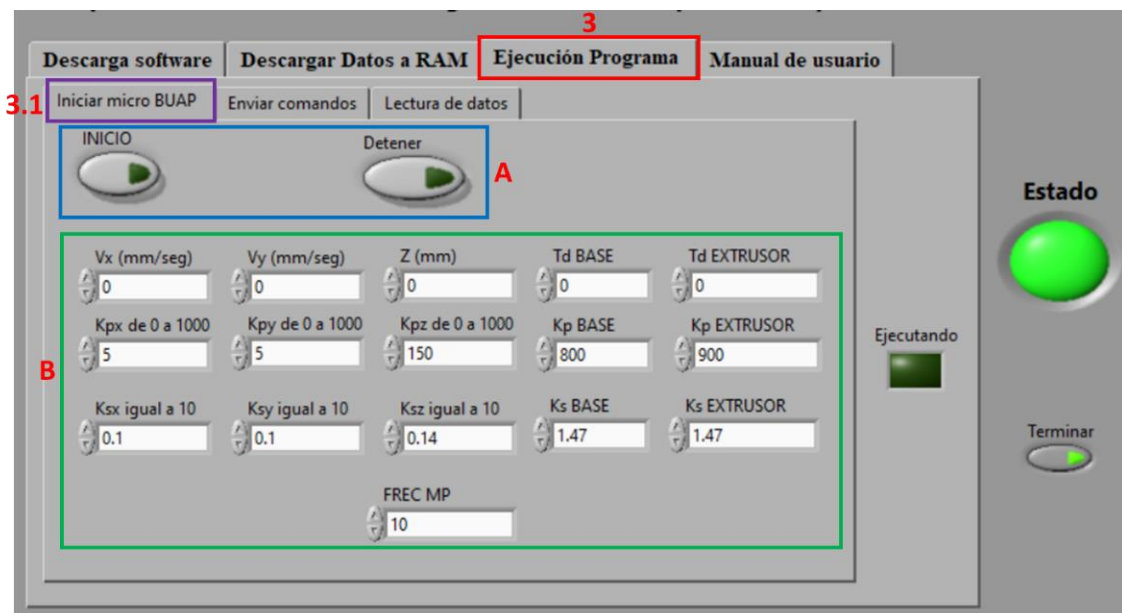


Figura 3.5.7: Interfaz de usuario pestaña 3, subpestaña 1.

En la segunda subpestaña podemos mandar instrucciones directamente a la RAM estos pueden ser independientes de la labor del sistema, pues entre las opciones esta leer cierta dirección, también es posible enviar nuevas posiciones.

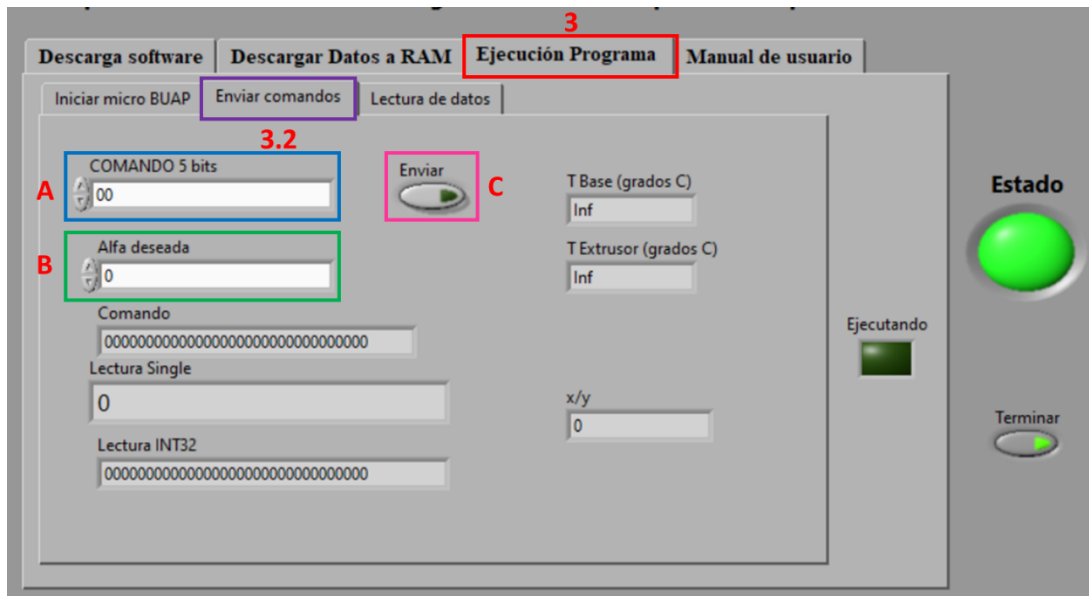


Figura 3.5.8: Interfaz de usuario pestaña 3, subpestaña 2.

En la figura 3.5.9 se presenta la última pestaña en esta podemos mandar una instrucción para que seamos capaces de guardar en nuestra computadora el registro de datos recolectados en este caso la posición, es importante agregar que esta pestaña nos permite obtener la información desde el momento en el que se indique hasta que se detenga, no obtenemos información previa o desde el encendido del dispositivo.

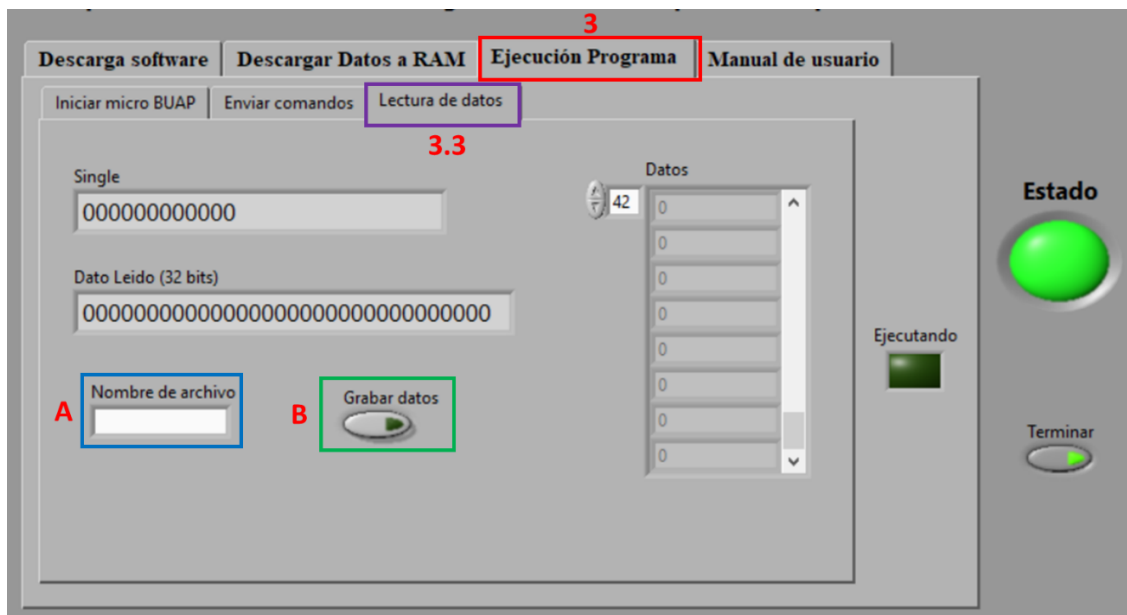


Figura 3.5.9: Interfaz de usuario pestaña 3, subpestaña 3.

### 3.6 Caracterización de motores

La caracterización de motores es necesaria para que al momento de programar el sistema se pueda tener una relación clara entre la señal enviada a los motores y su funcionamiento, para ello se busca obtener una relación lineal, esto es posible gracias que se ha observado que al exponer a los motores a una frecuencia determinada hay una relación directa entre el ancho de pulso y el torque del motor, para realizar el proceso se requiere una estructura donde montar el motor que tenga una base a paralela a un poste superior y se deben seguir los pasos que se encuentran en la figura 3.6.1.

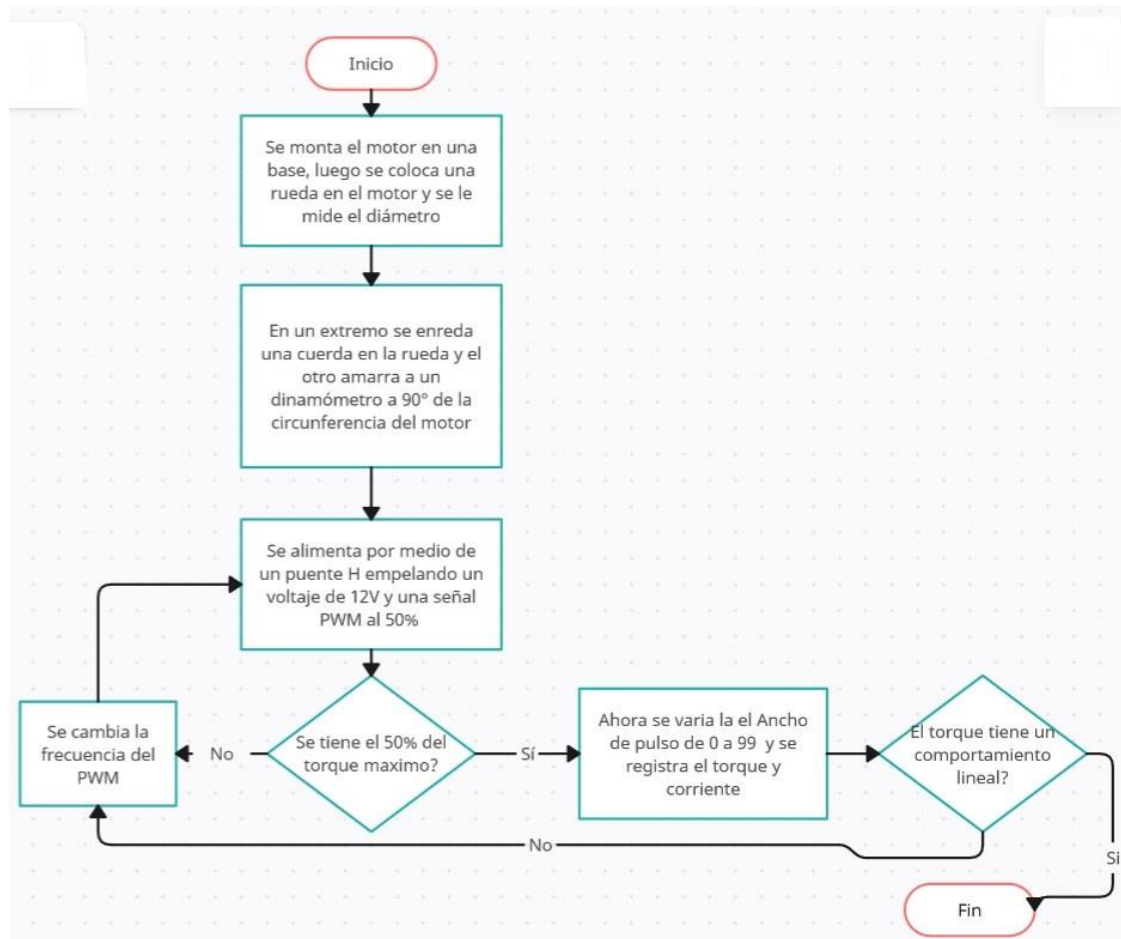


Figura 3.6.1: Diagrama de caracterización del motor.

Se requiere colocar una rueda al motor y enredar en ella un hilo, este hilo debe estar en 90° con la base del motor y llegar a un dinamómetro, para la alimentación se tiene que usar un puente H el cual tendrá alimentación de voltaje de 12V y en el lugar de la señal del FPGA se empleara un generador de señales, por la cual se envía un PWM al 50% con frecuencia variable, esto con el fin de buscar una frecuencia en la cual el dinamómetro nos dé la mitad del torque máximo del sistema, la frecuencia encontrada para el motor que se empleara en el eje X es de 300Hz. Una vez se tiene la frecuencia ahora variaremos el ancho de pulso mientras que la frecuencia esta fija tabla 3.6.2.

Frecuencia (Hz)	Corriente (A)	Carga (kg)
50	2.6	15
100	2.36	13.8
150	2.04	11.8
200	1.8	11.56
250	1.63	11.16
300	1.5	11
350	1.45	10.7

Tabla 3.6.1: Frecuencias de prueba en el motor X.

CT (%)	I (A)	M (kg)	Torque (Nm)
0	0	0	0
3	0.06	0.69	0.084525
7	0.14	1.61	0.197225
10	0.2	2.3	0.28175
13	0.2665	2.73	0.334425
17	0.3485	3.57	0.437325
20	0.41	4.2	0.5145
23	0.52133333	4.21666667	0.51654167
27	0.612	4.95	0.606375
30	0.68	5.5	0.67375
33	0.78375	6.27	0.768075
37	0.87875	7.03	0.861175
40	0.95	7.6	0.931
43	1.075	8.17	1.000825
47	1.175	8.93	1.093925
50	1.25	9.5	1.16375
53	1.36916667	9.89333333	1.21193333
57	1.4725	10.64	1.3034
60	1.55	11.2	1.372
63	1.665	11.34	1.38915
67	1.77071429	12.06	1.47735
70	1.85	12.6	1.5435
73	2.09875	13.23125	1.62082813
77	2.21375	13.95625	1.70964063
80	2.3	14.5	1.77625
83	2.62833333	15.1244444	1.85274444
87	2.755	15.8533333	1.94203333
90	2.85	16.4	2.009
93	2.83894737	17.6210526	2.15857895
97	3.0846	18.915	2.3170875
100	3.18	19.5	2.38875

Tabla 3.6.2: Tabla de ciclo de trabajo del motor X.

Para comprobar su funcionamiento se grafica junto una línea recta esperando que sean similares.

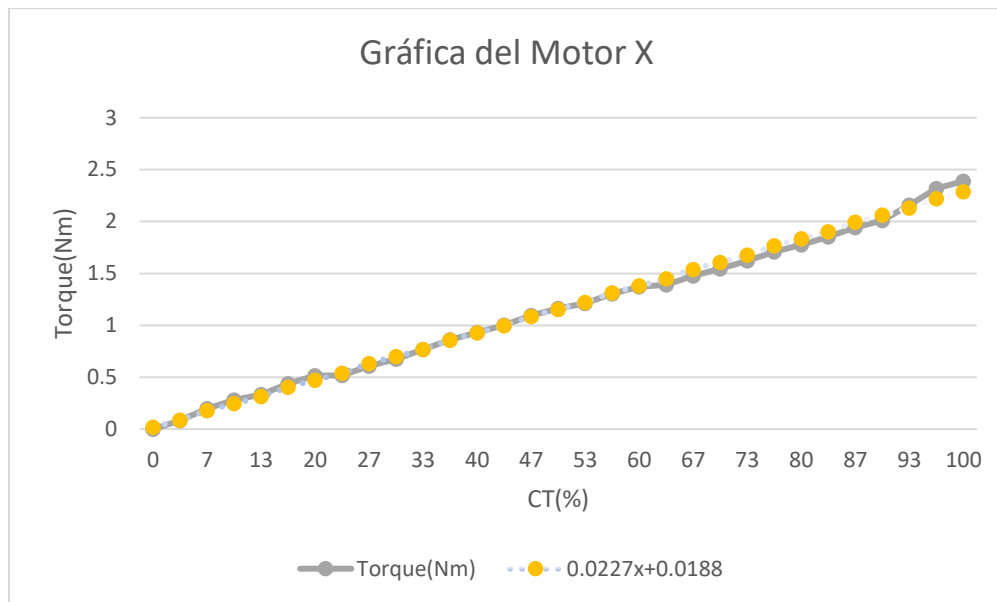


Figura 3.6.2: Grafica de comparación entre ciclo de trabajo y torque del motor X y una función lineal.

De esta manera podemos afirmar que el motor X posee un comportamiento aproximadamente lineal aun con el puente H, ahora se pasara a evaluar el Motor Z.

Para el eje Z se tiene la tabla 3.6.3, en donde se puede observar que la frecuencia en la cual se puede evaluar su comportamiento lineal es de 200.

Frecuencia (Hz)	Corriente (A)	Carga (kg)
50	2.3	13.5
100	2.1	13
150	1.9	12.5
200	1.79	12.2
250	1.72	11

Tabla 3.6.3: Frecuencias de prueba en motor Z.

Al encontrar la frecuencia de 200 se puede pasar a obtener la respuesta a distintos ciclos de trabajo.

En la figura 3.6.3 se comparan los datos de la tabla 3.6.4 con una función lineal para comprobar la respuesta. El comportamiento es diferente al de una función lineal alrededor del 50% sin embargo como hay similitud en los extremos y no se aleja demasiado se puede emplear esta frecuencia.

CT (%)	I (A)	M (kg)	Torque (Nm)
0	0	0	0
3	0.06	0.42	0.05145
7	0.14	0.98	0.12005
10	0.2	1.4	0.1715
13	0.312	0.975	0.1194375
17	0.408	1.275	0.1561875
20	0.48	1.5	0.18375
23	0.59033333	2.60666667	0.31931667
27	0.693	3.06	0.37485
30	0.77	3.4	0.4165
33	0.89925	4.125	0.5053125
37	1.00825	4.625	0.5665625
40	1.09	5	0.6125
43	1.2212	5.805	0.7111125
47	1.3348	6.345	0.7772625
50	1.42	6.75	0.826875
53	1.58116667	8.83333333	1.08208333
57	1.7005	9.5	1.16375
60	1.79	10	1.225
63	1.863	10.44	1.2789
67	1.98128571	11.1028571	1.3601
70	2.07	11.6	1.421
73	2.24475	12.31875	1.50904688
77	2.36775	12.99375	1.59173438
80	2.46	13.5	1.65375
83	2.26866667	12.45	1.525125
87	2.378	13.05	1.598625
90	2.46	13.5	1.65375
93	3.069	12.555	1.5379875
97	3.201	13.095	1.6041375
100	3.3	13.5	1.65375

Tabla 3.6.4: Tabla de ciclo de trabajo del motor Z.

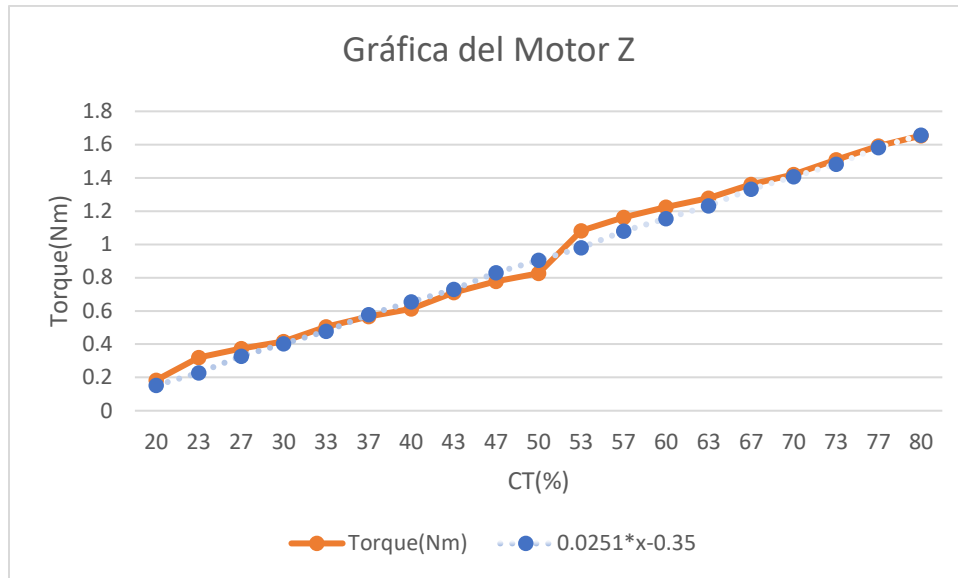


Figura 3.6.3 Grafica de comparación entre ciclo de trabajo y torque del motor Z y una función lineal.

### 3.7 Conclusiones

En este capítulo se ha presentado todos los elementos que se diseñaron para el funcionamiento del sistema, tanto el hardware con los puentes H y tarjeta de comunicación, el firmware realizado por medio del programa de Quartus II y la lógica del software que requiere emplear para que el sistema pueda comunicarse con la plataforma de Labview, a continuación, se presentan el filtro y sistema de estimación para la velocidad, Aceleración y Jerk.

# CAPÍTULO 4:

## Metodología para la obtención del Jerk

### 4.1 Efecto de muestreo

El muestreo de señales es una etapa esencial en el proceso de convertir una señal continua en el tiempo en una forma digital. Esta fase posibilita la recolección de información y su posterior manipulación en sistemas electrónicos. Es importante tener en cuenta que, durante este proceso, es inevitable la pérdida de ciertos datos.

Para poder recuperar esta información perdida, es necesario seguir el Teorema muestreo de Nyquist. Este teorema establece que si muestreamos la señal a una frecuencia que sea al menos el doble de la frecuencia de la señal original, será posible recuperar la información original de manera efectiva. De esta forma se pueden recuperar la función original por medio del muestreo, es conocida como frecuencia de Nyquist.

El proceso de digitalización se puede representar matemáticamente, para esto se emplea la función delta de Dirac [ $\delta(t)$ ], ver Fig. 4.1.1 que tiene varias propiedades, pues se trata de una función que solo existe un instante de tiempo, pero su amplitud tiende al infinito, por lo que su integral es 1, ver Ec. 4.1.1.

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (4.1.1)$$

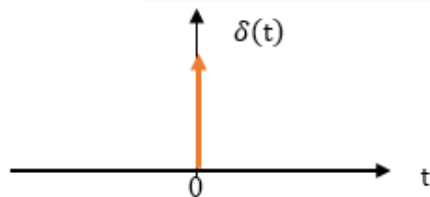


Figura 4.1.1: Delta de Dirac.

Otra función importante del muestreo es el tren o peine de Dirac, ver Fig. 4.1.2, también es conocido por su representación en inglés como:

$$Comb(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (4.1.2)$$

donde:  $T$  es el periodo del tren de Dirac.

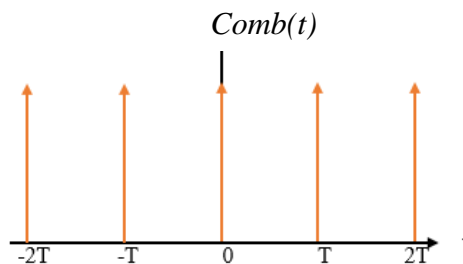


Figura 4.1.2: Tren de delta de Dirac.

El muestreo de la función  $f(t)$  matemáticamente se puede expresar con la ecuación (4.1.3):

$$f_m(t) = f(t) \text{Comb}(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (4.1.3)$$

La ecuación (4.1.3) la podemos representar gráficamente como se observa en la figura 4.1.3.

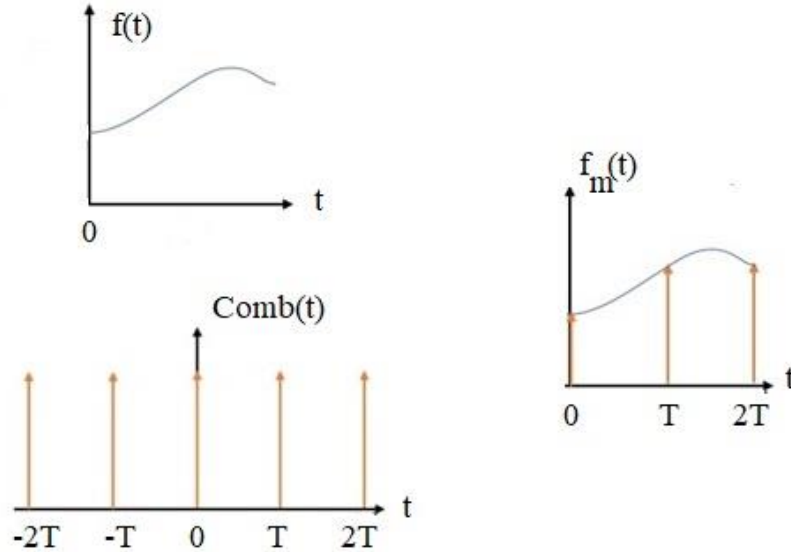


Figura 4.1.3: El muestreo de una señal  $f(t)$  es igual a multiplicar la función  $f(t)$  por el peine de Dirac. (a) Señal  $f(t)$ , (b) Peine de Dirac y (c) Señal muestreada.

De acuerdo con las propiedades de la transformada de Fourier la multiplicación de dos funciones en el tiempo es igual a la convolución de sus transformadas de Fourier en frecuencia [36], ver ecuación 4.1.4,

$$F(\omega) * \omega_0 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0) = \frac{\omega_0}{2\pi} \int_{-\infty}^{\infty} F(\tau) \text{Comb}(\omega - \tau) d\tau \quad (4.1.4)$$

$$\mathcal{F}[\text{Comb}(t)] = \omega_0 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0) \quad (4.1.5)$$

$$F[f(t)] = F(\omega) \quad (4.1.6)$$

La representación gráfica de las transformadas de Fourier tanto para  $f(t)$  como para el peine de Dirac  $\text{comb}(t)$ , se muestran en la figura 4.1.4.

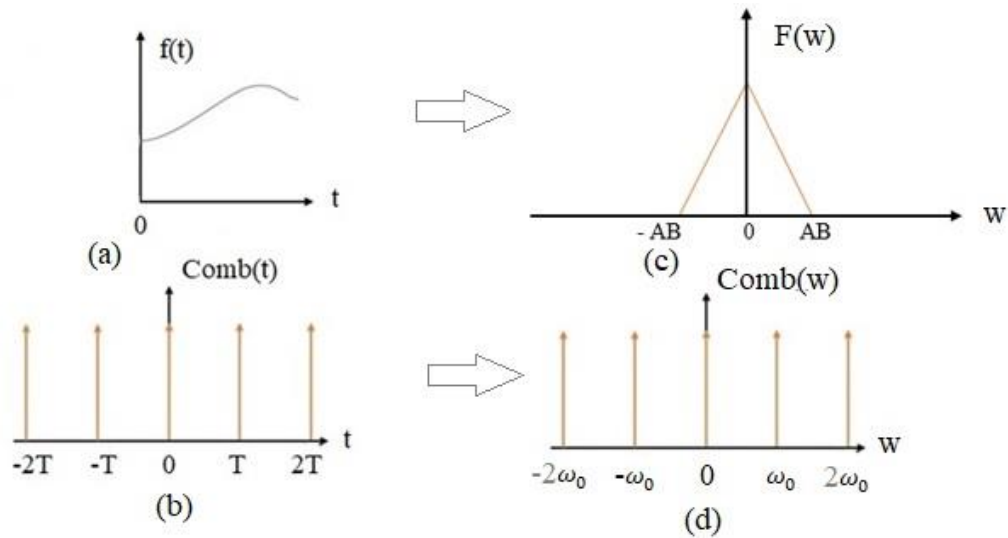


Figura 4.1.4: Representación gráfica de la función  $f(t)$  y  $Comb(t)$  con sus respectivas transformadas de Fourier. (a)  $f(t)$ , (b)  $Comb(t)$ , (c) Transformada de Fourier de la función  $f(t)$  y (d) Transformada de Fourier de la función  $Comb(t)$ .

La convolución puede presentar tres diferentes resultados según el valor del Ancho de Banda (AB) de la señal  $f(t)$  respecto de la frecuencia de muestreo ( $\omega_0$ ), los tres posibles resultados muestran en la figura 4.1.5.

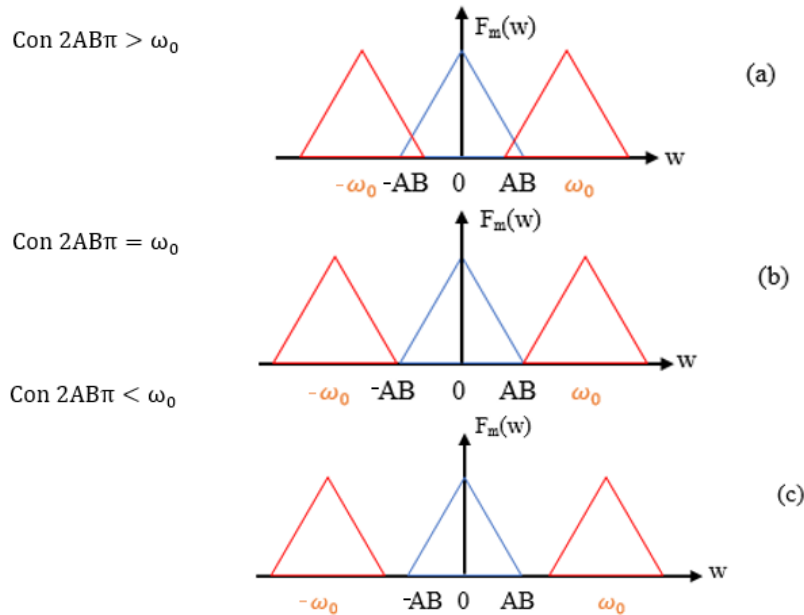


Figura 4.1.5: Resultados de la convolución entre  $F(\omega)$  con el peine de Dirac  $Comb(\omega)$ . a) Cuando la frecuencia angular de muestreo  $\omega_0$  es menor a 2 veces el ancho de banda (AB) multiplicado por  $2\pi$ , b) cuando la frecuencia angular de muestreo  $\omega_0$  es igual a 2 veces el ancho de banda (AB) multiplicado por  $2\pi$  y c) cuando la frecuencia angular de muestreo  $\omega_0$  es mayor a 2 veces el ancho de banda (AB) multiplicado por  $2\pi$ .

En el caso (a) de la figura 4.1.5 la información se pierde al encontrarse traslapadas los espectros  $F(\omega)$  y no hay forma de recuperar la señal  $f(t)$  este fenómeno se conoce como Aliasing. En el caso (b) de la figura 4.1.5, la información se puede recuperar únicamente empleando un filtro pasa bajas ideal que recupere únicamente el espectro  $F(\omega)$  centrado en  $\omega = 0$  pues cumple con el valor exacto de 2 veces frecuencia de la señal dicho en el teorema de muestreo. En el caso (c) de la figura 4.1.5, se puede recuperar la información con un filtro pasa bajas normal, pues cumple el teorema de muestreo. Siempre y cuando solo se conserve el espectro  $F(\omega)$  central y elimine los demás espectros no deseados. Por lo anterior, podemos decir que el muestreo genera espectros no deseados en el dominio de la frecuencia, sin embargo, siempre que use la frecuencia de Nyquist pueden ser eliminados por medio de un filtro pasa bajas.

## 4.2 Filtro

Los filtros desempeñan un papel esencial al eliminar componentes no deseadas de una señal en el dominio de la frecuencia. Su funcionalidad está vinculada a factores cruciales como la Frecuencia de Corte y la Banda de Paso. Estas características permiten comprender el comportamiento y la operación de un filtro en particular. La Banda de Paso ofrece información que permite identificar el tipo de filtro utilizado. Por ejemplo, un filtro pasa bajos se emplea para suprimir frecuencias que exceden la Frecuencia de Corte, lo cual resulta relevante en aplicaciones de comunicación.

Los filtros pueden adoptar formas analógicas o digitales. En el contexto de este trabajo, se opta por emplear un filtro digital implementado en el software. Este filtro digital está diseñado para eliminar frecuencias generadas por el proceso de muestreo, conservando únicamente la frecuencia esencial necesaria para recuperar la información de manera precisa esto solo es posible mientras se siga el teorema de muestreo.

Para el filtro se deben considerar 2 cosas que debe realizar el menor retraso posible para que se pueda aplicar un control óptimo con el fin de tener mejor definidas las señales y que debe ser posible implementarlo en el software de nuestro sistema.

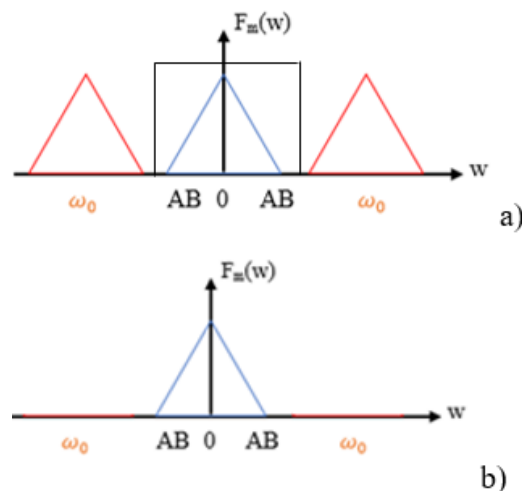


Figura 4.2.1: Señal con recorte de filtro: a), señal filtrada b).

El tipo de filtro elegido es un filtro Bessel, primero se descartaron todos los filtros que tengan ondulaciones en su respuesta, pues requerimos que se mantenga lo más cercano a la señal original (analógica) que se muestreó eso descarta algunos como el elíptico y Chebyshev pues presentan ondulación que se hace presente al filtrar la señal muestreada, a continuación, se presenta una gráfica donde se puede observar el comportamiento de los filtros Tchebyshev y Bessel empleando tablas encontradas en la bibliografía en respuesta a la señal muestreada.

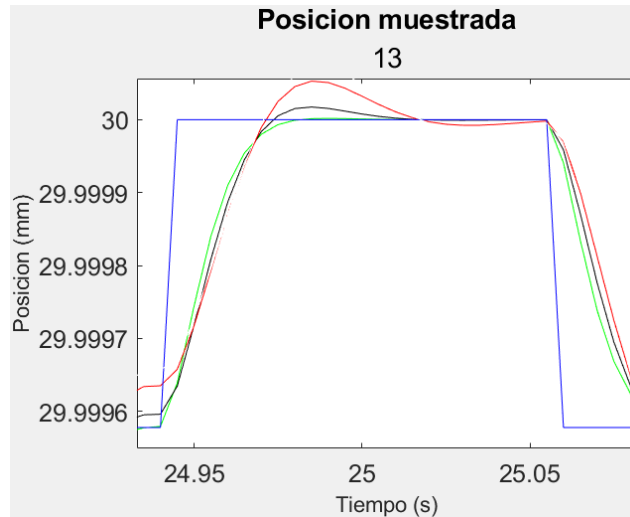


Figura 4.2.2 Señal muestreada (azul), Bessel (Verde), Tchebyshev rizo 1 Hz (Rojo), Butterworth (Negro).

Como se puede ver el Bessel no presenta un rizo lo cual lo deja como el filtro más útil de los revisados para recuperar la señal, a cambio se requerirá una frecuencia de corte alejada de la función pues su retardo de fase será mayor, pero debido a que trabajamos a altas frecuencias de lectura en comparación a la frecuencia a muestrear no es un problema. Es importante agregar que el método usado para la conversión a digital puede quitar propiedades a los filtros sin embargo el Bessel sigue comportándose como si solo tuviera polos.

Los filtros Bessel reciben su nombre del matemático alemán Friedrich Bessel (1784-1846), quien desarrolló la teoría matemática en la que se basa el filtro, estos filtros son usados según su linealidad, aunque a costa del retardo de fase. Estos filtros son construidos por medio de una función inversa basada en los polinomios de Bessel, que se trata de polinomios ortogonales con secuencia polinomial, por lo tanto, su función de transferencia puede ser escrita como [37],

$$H(s) = \frac{1}{B_n(s)}, B_n(s) = \sum_{k=0}^N a_k s^k \quad (4.2.1)$$

donde  $B_N(s)$  es el polinomio de Bessel cuyos coeficientes son:

$$a_k = \frac{(2N - k)!}{2^{N-k} * k! (N - k)!} \quad (4.2.2)$$

Una vez elegido el filtro se usarán tablas de las constantes asociadas a este para la construcción de la función de transferencia, esto para poder hacer un código en Matlab para los cálculos según la frecuencia de corte y según las tablas mencionadas empleando la ecuación (4.2.3) [38],

$$\frac{1}{(1 + a_{(1,1)}s + a_{(1,2)}s^2) * (1 + a_{(2,1)}s + a_{(2,2)}s^2) * ... * (1 + a_{(n,1)}s + a_{(n,2)}s^2)} \quad (4.2.3)$$

donde  $2n$  es el valor del filtro de valor par, para filtros impares es  $2n-1$ ;  $a_{(n,1)}$  es el valor en la tabla entre la frecuencia de corte y  $a_{(n,2)}$  es el valor en la tabla [38] entre el cuadrado de la frecuencia de corte.

El filtro empleado es un Pasa-bajas, pues la frecuencia de muestreo que se busca eliminar es mayor a la señal de posición deseada, para trabajarla se propone su representación como una

función de transferencia  $H(s)$  tal que,

$$H(s) = \frac{Y(s)}{X(s)} \rightarrow h(t) = \mathcal{L}^{-1}(H(s)) \quad (4.2.4)$$

donde la señal de entrada es  $X(s)$  y la señal de salida  $Y(s)$  es en el dominio de la frecuencia.

Los filtros pueden representarse en digital y analógico para cambiar de una representación a otra es necesario contar con la representación en variables de estado del filtro.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (4.2.5)$$

Esta representación permite emplear la solución general para el caso donde el sistema no empieza en 0.

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (4.2.6)$$

De esta forma podemos observar la salida del filtro y usamos la ecuación (4.2.7) para representar la entrada en forma discreta.

$$u(t) = u(kT), \text{ para } kT \leq t < kT + T \quad (4.2.7)$$

Así se puede escribir la solución general para caso discreto de la siguiente manera.

$$x((k+1)T) = e^{A(k+1)T}x(0) + e^{A(k+1)T} \int_0^{(k+1)T} e^{-A\tau} Bu(\tau) d\tau \quad (4.2.8)$$

Sin embargo, como se requiere pasar a variables de estado para su aplicación, por lo que se multiplica por  $e^{AT}$  y se le resta la ecuación (4.2.9).

$$x(kT) = e^{AkT}x(0) + e^{AkT} \int_0^{kT} e^{-A\tau} Bu(\tau) d\tau \quad (4.2.9)$$

Como resultado se obtiene la ecuación (4.2.10).

$$x((k+1)T) = e^{AT}x(kT) + e^{A(k+1)T} \int_{kT}^{(k+1)T} e^{-A\tau} Bu(\tau) d\tau \quad (4.2.10)$$

Como  $u(\tau)$  es igual a  $u(kT)$ , entonces podemos considerarla una constante y sacarla de la integral además reacomodamos los extremos de la integral para que empiece en 0.

$$x((k+1)T) = e^{AT}x(kT) + \left( \int_0^T e^{A\tau} d\tau \right) Bu(kT) \quad (4.2.11)$$

De esta forma podemos distinguir 2 matrices constantes que multiplican a  $x(kT)$  y  $u(kT)$ , estas son  $G$  y  $H$  estas serán calculadas por medio de Matlab.

$$G = e^{AT}, \quad H = \left( \int_0^T e^{A\tau} d\tau \right) B \quad (4.2.12)$$

Con esto podemos representar nuestro sistema discreto en forma de 2 ecuaciones lineales donde sabemos que  $x_{(1,1)}(kT)$  corresponde a la salida de nuestro filtro.

Para su implementación  $u$  será el valor de la entrada que es la velocidad, posición, aceleración o jerk actual mientras que el de  $x$  será la misma variable, pero calculado en el ciclo anterior este será llamado de una dirección RAM, estas se operan con las matrices  $G$  y  $H$ , se obtendrá de salida el valor filtrado y este se guardará para usarse para el siguiente filtro.

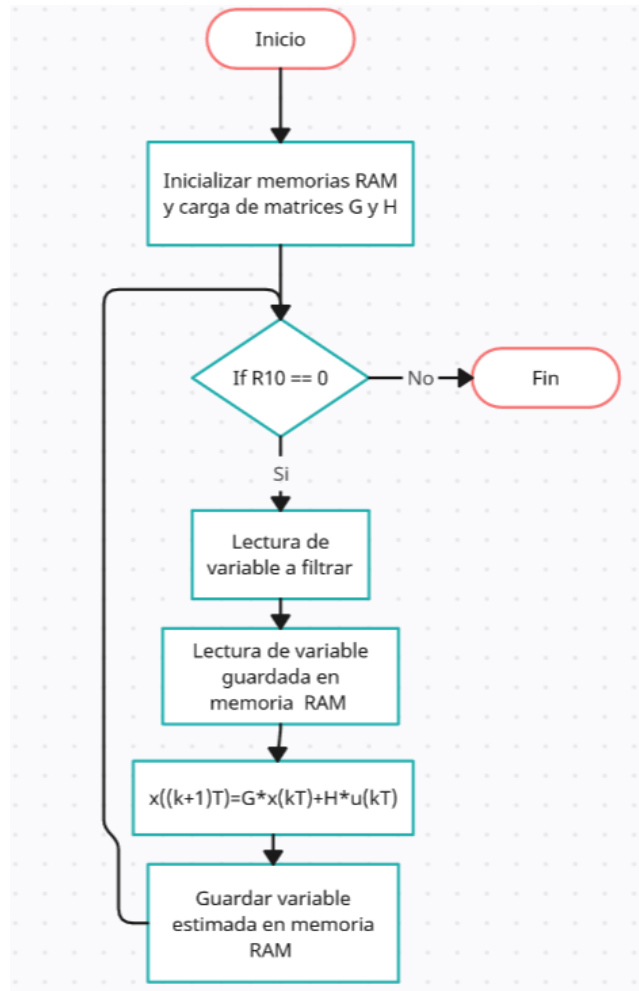


Figura 4.2.3: Diagrama de flujo del filtro.

### 4.3 Error de cuantificación

Como mencionamos previamente, para trabajar con la información es necesario llevar a cabo un proceso de muestreo. Sin embargo, no solo se produce una pérdida de información en términos de la cantidad de datos, sino que también existe la posibilidad de perder el instante exacto en el cual se produce un cambio en la señal en tiempo real. Este aspecto se vuelve especialmente relevante cuando se necesita conocer con precisión el valor del tiempo en el cual ocurrió el cambio. Un ejemplo práctico es cuando se requiere utilizar el valor de posición interpretado por un sistema digital. En situaciones donde el sensor no opera con el mismo reloj que el sistema digital, se puede observar el desafío ilustrado en la figura 4.3.1.

En la figura 4.3.1 se puede observar que si tomamos los datos de la posición y el muestreo para calcular la velocidad se obtendrán errores, esto debido a que cuando se guarda el valor de la posición puede recién haber cambiado o estar punto de volver a cambiar, ese desfase afectara la estimación de la velocidad, también en casos donde la posición no cambia entre lecturas no por haberse detenido sino porque el movimiento es muy lento, por lo que se generan ceros indeseados, para poder corregir esto se propone emplear el método MT.

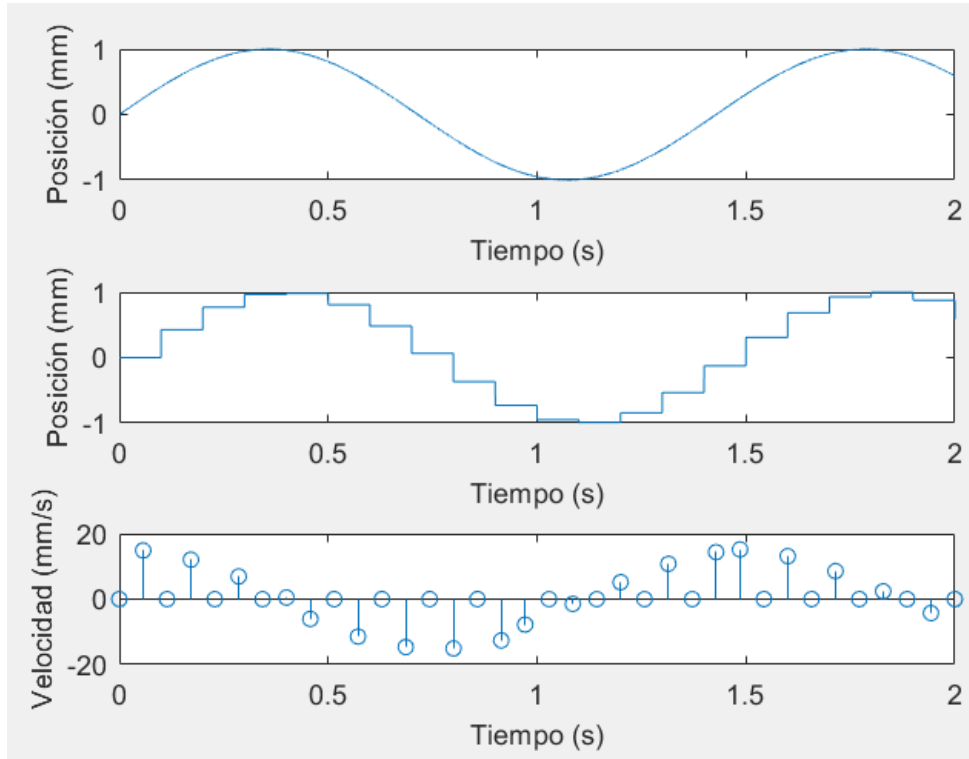


Figura 4.3.1: Muestreo de posición por encoder y FPGA.

#### 4.4 Método MT

La velocidad se estima comúnmente mediante el método M, el cual involucra calcular la diferencia entre los valores medidos en el encoder durante intervalos de tiempo de muestreo. El nombre "M" proviene de "medición". Además, existe otro método conocido como método T, en el cual se mide el valor deseado utilizando el inverso del reloj auxiliar. Su nombre proviene de "tiempo". La combinación de estos dos métodos se conoce como el método MT, que empleamos con el propósito de reducir el error de cuantificación en la velocidad. La ecuación 4.4.1 permite estimar la posición al considerar los desfases que resultan de la diferencia de tiempo entre el cambio registrado en el encoder y el momento de la lectura de la posición según el método MT [39].

$$v_m^{MT} = \frac{x_k - x_{k-1}}{T_m + \delta t_{k-1} - \delta t_k} \quad (4.4.1)$$

donde  $x_k$  es la posición actual,  $x_{k-1}$  es la posición pasada y  $T_m$  es el intervalo de muestreo,  $\delta t_{k-1}$  es la diferencia entre el cambio de la posición previo al anterior ciclo de muestreo y  $\delta t_k$  es la diferencia de tiempo antes muestreo de la posición actual.

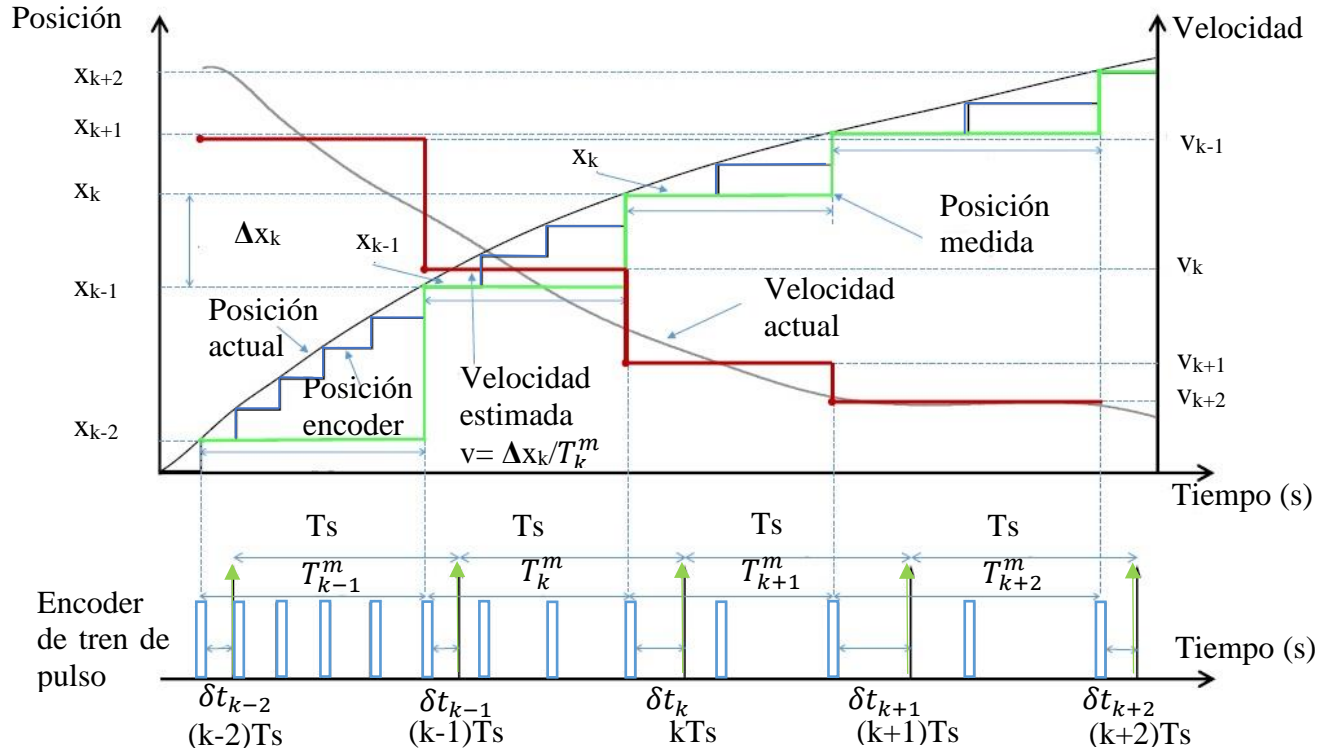


Figura 4.4.1: Posición (negro), posición encoder (azul), posición muestreada (verde), velocidad MT (rojo).

#### 4.5 Obtención de la posición y velocidad

Para lograr la obtención de la posición, se requiere emplear el decodificador de cuadratura descrito en el subcapítulo 3.1. Su funcionamiento se detalla en la figura 3.1.8, y su implementación en el firmware puede observarse en la figura 3.4.2. En esta etapa, el valor del contador obtenido se transmite al software, donde es multiplicado por el valor de resolución del tornillo embalado, que es 0.000422297297. Así, se obtiene el valor de la posición. Posteriormente, este valor de posición se somete a un filtro de segundo orden que elimina el efecto del muestreo. El filtro está configurado con una frecuencia de corte de 16 Hz, conforme se abordó en el subcapítulo 4.2.

En cuanto a la velocidad, será estimada utilizando el método MT, empleando la ecuación (4.5.1),

$$\frac{x(k) - x(k - 1)}{Ts - \delta(k) + \delta(k - 1)} \quad (4.5.1)$$

donde  $x(k)$  es la posición muestreada,  $x(k - 1)$  es la posición muestreada del ciclo anterior,  $T_s$  es el tiempo de muestreo que en este trabajo corresponde a 0.005s,  $\delta(k)$  es la diferencia entre el ultimo cambio en la posición y el instante en que fue muestreado,  $\delta(k - 1)$  es la diferencia de tiempo del ciclo pasado entre el cambio de la posición y el instante de muestreo.

Para obtener el valor de  $\delta(k)$  se colocó un contador en el firmware que se reinicia cada que el valor de la posición como se ve en el diagrama.

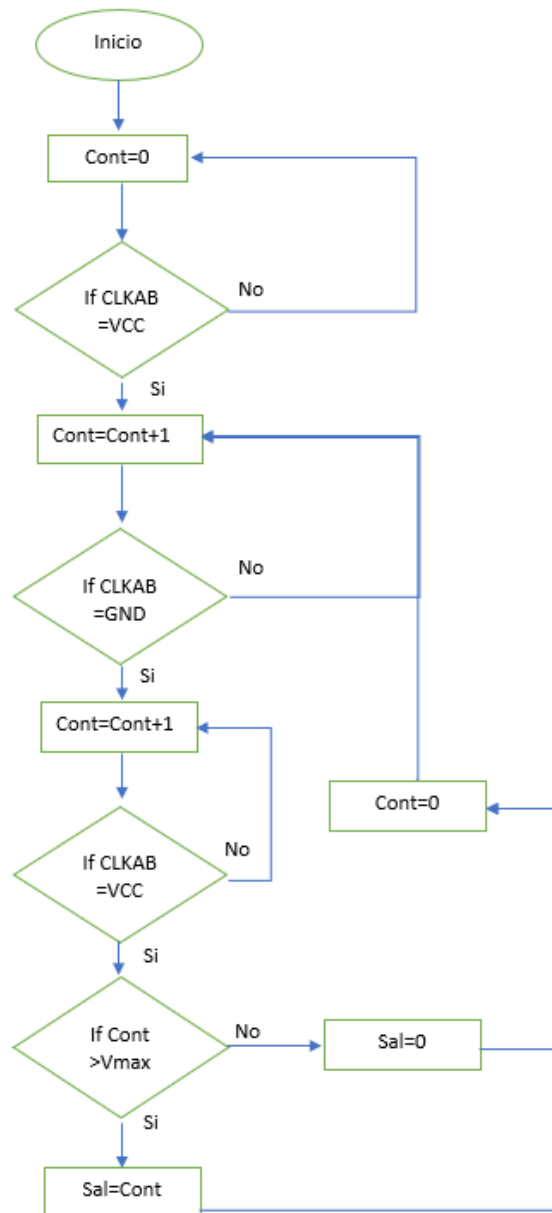


Figura 4.5.1: Diagrama de flujo de contador entre flancos.

Este fue implementado en el firmware como se puede observar en la figura 3.4.2.

El valor del contador será llamado por el software y multiplicado por  $10^{-8}$ s para convertirlo a segundos, pues el reloj del firmware está a 100Mhz, con esto tenemos todos los datos para llevar a cabo la operación sin embargo es necesario considerar los casos cuando la ecuación (4.5.1) no puede estimar la velocidad estos serian.

- Cuando el denominador es 0, un caso donde la división se indefina, para evitarlo en vez de realizar la operación se guarda el valor de la velocidad del ciclo anterior.
- Cuando el numerador es 0, esto se da cuando la posición no ha cambiado desde el ciclo anterior, se envía la velocidad anterior, si esto se repite más de 9 veces seguidas se envía un 0.

El valor de la velocidad que recuperemos será pasado por un filtro de segundo orden con frecuencia de corte de 16Hz generado con lo visto en la subcapítulo 4.2, tendrá las mismas constantes que el de la posición, la frecuencia de corte fue elegida debido a que se encontró que la resolución del encoder no es capaz de notar cambios muy lentos produciéndose, para esta trayectoria hasta 20 veces se repite el mismo valor, por lo que idealmente la velocidad es de 0 en medio, así que dividiendo los 200Hz/10 obtenemos 20Hz y dado que no se tiene un filtro perfecto, se hicieron pruebas para elegir una frecuencia de corte lo más cercana posible.

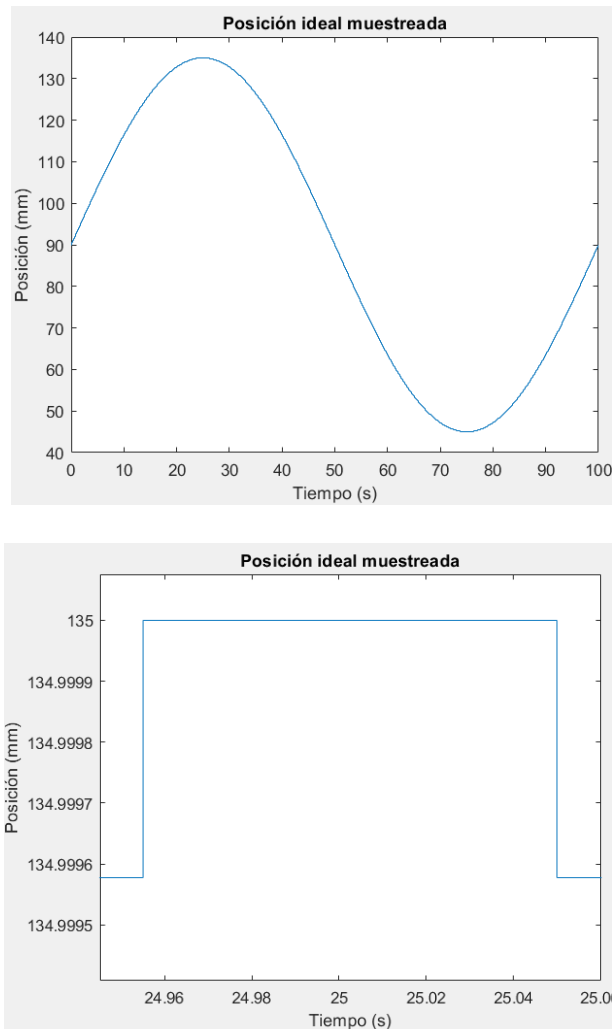


Figura 4.5.2: Posición ideal muestreada a 200Hz con resolución de 0.0004222972973.

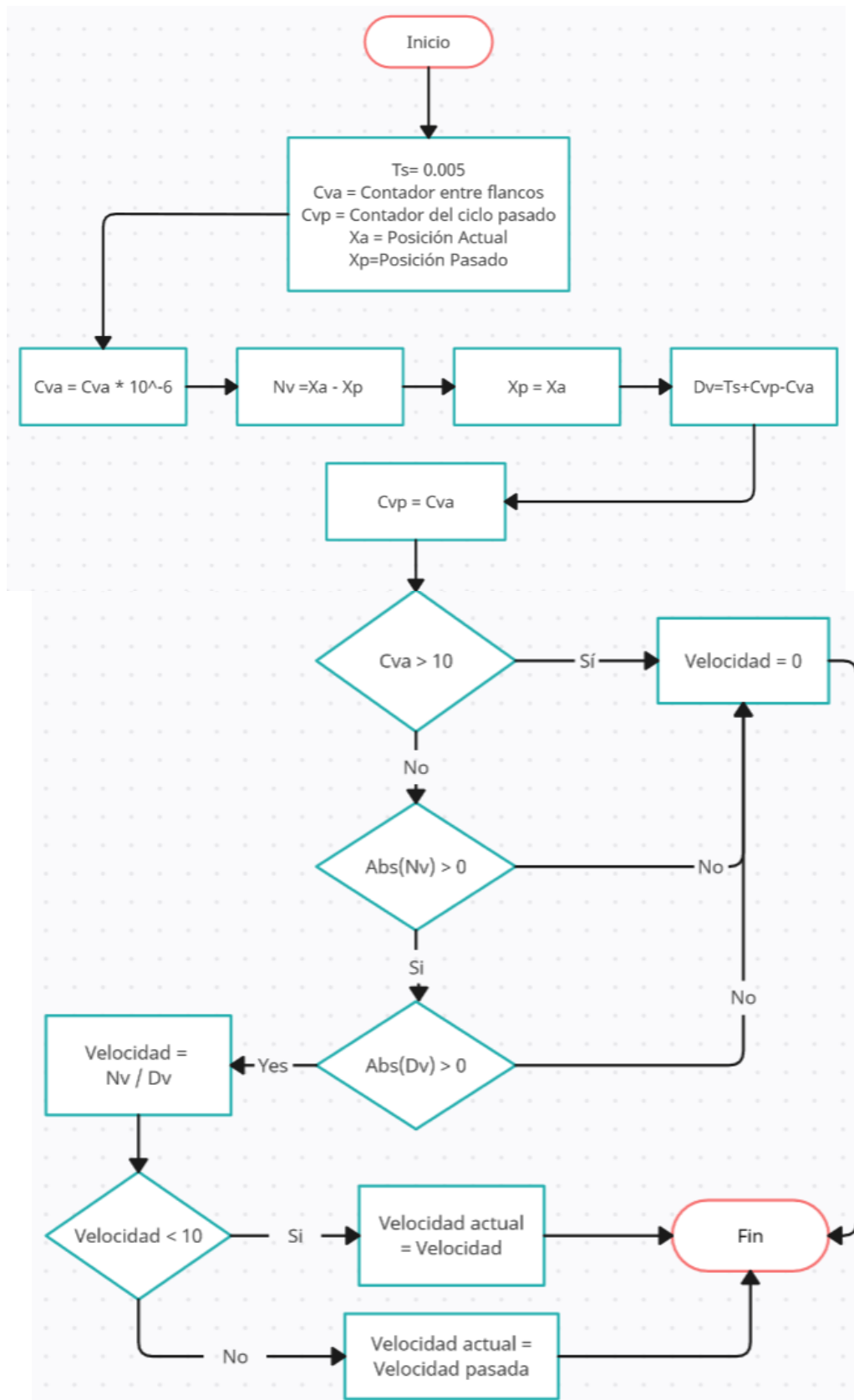


Figura 4.5.3: Diagrama de flujo de cálculo de velocidad.

#### 4.6 Obtención de la aceleración y Jerk.

Para mejorar la relación entre la velocidad y el torque, con el propósito de obtener una aceleración más acorde a lo esperado, es fundamental aplicar una mejora al control que permita compensar las fuerzas de fricción presentes en el sistema. Al representar la velocidad en función del torque, se observan tres tipos de fricciones predominantes: fricción de Coulomb, fricción viscosa y el efecto de Stribeck.

La fricción de Coulomb, que es constante, representa la fuerza necesaria para vencer la resistencia al movimiento en el sistema. Por otro lado, el efecto de Stribeck corresponde a una fricción variable que surge al arrancar el motor. Esta fricción presenta un pico poco después del inicio, independientemente de la dirección de giro, y va disminuyendo hasta desaparecer. La fricción viscosa, en cambio, es proporcional a la velocidad. Una vez que el efecto de Stribeck ha desaparecido, se observa una relación entre el torque y la velocidad que sigue un patrón en forma de ángulo. Mejorar esta relación entre velocidad y torque se logra implementando ajustes en el control, con el objetivo de compensar y mitigar estas diferentes formas de fricción

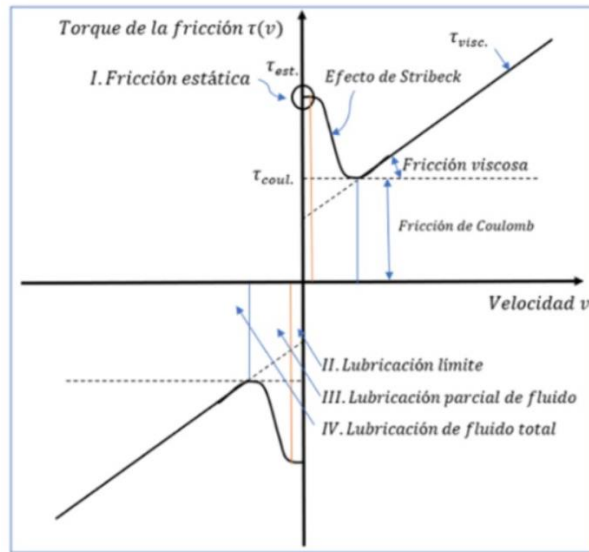


Figura 4.6.1: Torque vs velocidad.

En el contexto de esta tesis, el efecto de Stribeck no tiene un impacto significativo, por lo que es necesario centrarse en la fricción de Coulomb y la fricción viscosa cuando se realiza una trayectoria sinusoidal, para compensar la de Coulomb se busca el valor del torque alrededor del 0 de velocidad, tanto en ascendente como descendente, para la viscosa se requiere obtener la pendiente de la trayectoria ascendente que va desde el momento en que el torque va incrementando en función de la velocidad hasta que cambia de ascendente a descendente y los mismos en la trayectoria descendente, estos valores se convertirán en las unidades usadas dentro del sistema para el ancho de pulso empleando la ecuación (5.2.1).

$$A_m = \frac{c_{1,2}}{\tau_{max}} * kp_{max}, \quad (5.2.1)$$

donde colocaremos en el lugar de  $c$  el valor de compensación que se quiera transformar, siendo  $c_1$  la que requerimos para la fricción de coulomb y  $c_2$  la requerida para la fricción viscosa dividido entre el torque máximo, el cual depende el motor, multiplicado como el valor máximo de ganancia posible, en nuestro caso 1000, dando como resultado el valor que usaremos en la ecuación (5.2.2) para compensar el error, siendo  $k$  para la fricción de coulomb y  $b$  para la fricción viscosa. Una vez la sumamos a nuestra función la compensación de fricción al ancho de pulso deseado.

$$A\tau_a = A\tau + A\tau_{cf}, \quad (5.2.2)$$

donde  $A\tau_a$  que es el ancho de pulso en función del torque aplicado en el sistema  $A\tau$  es el ancho de pulso deseado en función del torque que obtenemos del control y  $A\tau_{cf}$  es el ancho de pulso en función del torque para compensar la fricción.

$$\tau_{cfa} = \frac{1}{2} \left( (1 + \text{signo}(\dot{q}_d))(k_{pa} \tanh(\lambda \dot{q}_d) + b_{pa} \dot{q}_d) - (\text{signo}(\dot{q}_d) - 1)(k_{na} \tanh(\lambda \dot{q}_d) + b_{na} \dot{q}_d) \right) \quad (5.2.3)$$

$$\tau_{cfd} = \frac{1}{2} \left( (1 + \text{signo}(\dot{q}_d))(k_{pd} \tanh(\lambda \dot{q}_d) + b_{pd} \dot{q}_d) - (\text{signo}(\dot{q}_d) - 1)(k_{nd} \tanh(\lambda \dot{q}_d) + b_{nd} \dot{q}_d) \right) \quad (5.2.4)$$

Considerando la función  $\text{signo}(\dot{q}_d)$  como sigue:

$$\text{signo}(\dot{q}_d) = \begin{cases} 1 & 0 < \dot{q}_d < \infty \\ 0 & -\infty < \dot{q}_d \leq 0 \end{cases}$$

En la función (5.1.3) está la compensación para la ascendente y en la (5.1.4) está la compensación para la descendente. Estamos empleando una lambda grande para aproximar una función signo por medio de la  $\tanh$ ,  $k$  corresponde al valor para compensar la fricción estática y de coulomb y  $b$  para la fricción viscosa [40].

Para calcular la aceleración, se ha propuesto un enfoque similar al utilizado para la velocidad. Se emplea el método MT, pero en lugar de la posición en el numerador, se coloca la velocidad. Las diferencias, en este caso, serían calculadas utilizando el contador junto con el tiempo en que ocurrió el cambio en la velocidad.

$$\frac{v(k) - v(k - 1)}{Ts - \delta v(k) + \delta v(k - 1)} \quad (4.6.1)$$

Para probarlo se replicó el proceso por el cual se realiza el método MT pero empleando Matlab, se realiza una trayectoria senoidal, con una amplitud de 30cm, una frecuencia de 0.01s y se estima la aceleración.

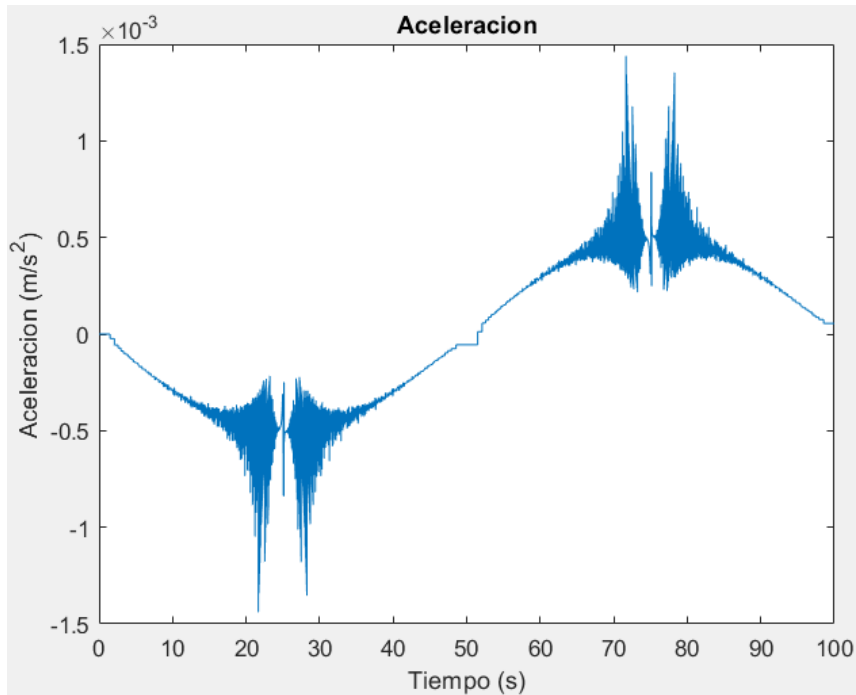


Figura 4.6.2: Simulación del método MT para la aceleración.

Como se puede ver esta observan deltas producidas por el problema al estimar la velocidad como se va acercando a 0, se propuso que estas pueden ser eliminadas mediante un filtro, esto aprovechando que se emplea para eliminar el efecto de muestreo.

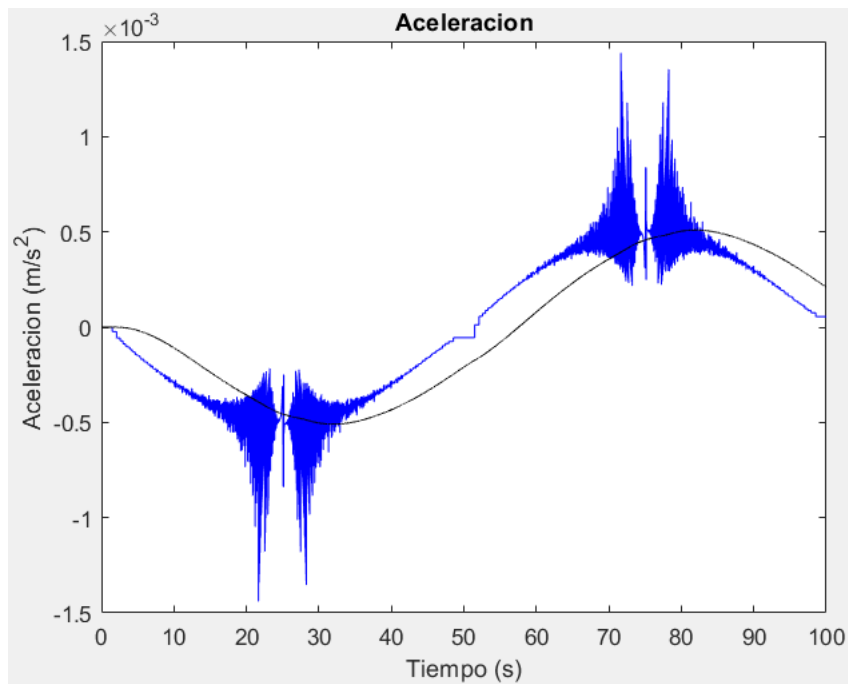


Figura 4.6.3: Aceleración por método MT sin filtro (azul), aceleración después de filtro de 0.2Hz (negro).

Como podemos observar se genera un desfase muy grande de aproximadamente 10 segundos, por lo que debe descartarse este método para la obtención de aceleración en tiempo real y por lo tanto tampoco es funcional para el cálculo de jerk, por lo que se empleara un método más común.

Como ya tenemos la velocidad se puede usar la aproximación de Euler, para estimar la aceleración, para ello empleamos la ecuación (4.6.2).

$$A = \frac{V_A - V_P}{0.005s} \quad (4.6.2)$$

Se toma la velocidad actual ( $V_A$ ) y la velocidad pasada ( $V_P$ ) y se divide entre 0.005s que es el intervalo de lectura del software. Así mismo el cálculo de la aceleración sucede en el software se puede emplear el mismo filtro que se usó para la velocidad, de esta manera se puede estimar la aceleración sin efecto de muestreo.

Para la estimación del jerk se puede realizar el mismo proceso se emplea el mismo procedimiento usando la aceleración estimada en el instante de cálculo  $A_A$  y la aceleración del ciclo pasado  $A_P$ .

$$J = \frac{A_A - A_P}{0.005s} \quad (4.6.3)$$

Con el jerk calculado se emplea una vez más filtro que fue diseñado para la velocidad permitiendo que se elimine el efecto de muestreo.

#### **4.7 Métrica**

La métrica para evaluar la similitud entre la simulación y los datos experimentales es nombrada “metr2ev” [42], este será ejecutado por medio del programa de Matlab, esta métrica sirve para sacar el porcentaje de similitud entre 2 arreglos, contemplando la forma en tiempo y la frecuencia, para hacerlo emplea la transformada discreta del coseno que pasa los arreglos al dominio de la frecuencia y el resultado es normalizado, se saca la diferencia entre los valores y empleando la sumatoria se reducen a un valor para poder extraer un promedio esto se hace tanto con la representación en tiempo como en frecuencia, con los valores obtenidos de ambas comparaciones se calcula un porcentaje de similitud general que varía entre 0-100%.

#### **4.8 Conclusiones**

En este capítulo se completó la teoría necesaria para la estimación de la velocidad, aceleración y jerk en tiempo real, empleando método MT para el error de cuantificación y filtros para el efecto de muestreo, lo cual se implementó en el sistema a nivel de firmware y software, en el siguiente capítulo se muestran los resultados.

## CAPÍTULO 5:

### Resultados

A continuación, se presentan los resultados de la simulación de los parámetros agrupados y los resultados experimentales relacionados con la posición, velocidad, torque, aceleración y jerk. También se revisa el método utilizado para la estimación de los datos, para las simulaciones se emplean las ecuaciones (2.3.17-2.3.30) obtenidas mediante el modelado por parámetros agrupados con las cuales se hace una representación a bloques mediante Simulink, de ahí se obtiene la posición, a partir de esta por medio de derivadas se obtiene la velocidad, aceleración y jerk, el torque también se obtiene de la simulación. Para los datos experimentales se emplean los datos recibidos por medio de la comunicación Wifi, al no contar con una comunicación síncrona se requirió ajustar el tiempo para que corresponda al intervalo y emplear filtros digitales para eliminar el ruido.

- **Posición, velocidad y aceleración:** Se explica el método empleado para su obtención, se emplea el método de la Transformada Discreta del Coseno (DCT) para comparar las simulaciones entre los datos simulados y experimentales. Además, se compara el error en la posición con los resultados obtenidos en la tesis de Ricardo Torres [41], que utilizó la misma mecánica.
- **Torque:** Se presenta el torque proporcionado por el controlador antes de la compensación de la fricción. También se muestra una gráfica que ilustra cómo la fricción afecta la relación entre el torque y la velocidad.
- **Torque compensado:** Se muestra el torque proporcionado por el controlador después de aplicar la compensación de la fricción y se compara con el simulado para después mostrar la gráfica ahora de torque vs velocidad.
- **Jerk:** Se presentan los resultados del jerk en las simulaciones, así como el experimental para demostrar el avance con respecto a lo que se obtenía con la tercera derivada de la posición antes de la investigación de esta tesis.

En resumen, estos resultados y comparaciones proporcionan una evaluación completa de la efectividad de la estimación de datos en el sistema y la mejora con respecto a su comportamiento antes de esta investigación. Las conclusiones basadas en estos resultados se discuten en el capítulo correspondiente.

#### 5.1 Resultados en el eje X

En la gráfica 5.1.1 a) se observa el resultado de la simulación por parámetros agrupados, en el cual podemos observar que es casi idéntica a la 5.1.1 b), que tiene los resultados experimentales. Al compararlas empleando métrica “metr2ev” arroja un 99.1% de similitud. En la gráfica 5.1.1 c) se puede observar una comparación entre el error encontrado en la tesis previa [41], donde se trabajó previamente con la mecánica del sistema y se reportó un error de 2600  $\mu\text{m}$  mientras que el error actual es de 23  $\mu\text{m}$ , estos datos son tomados después de pasar el intervalo de traslación.

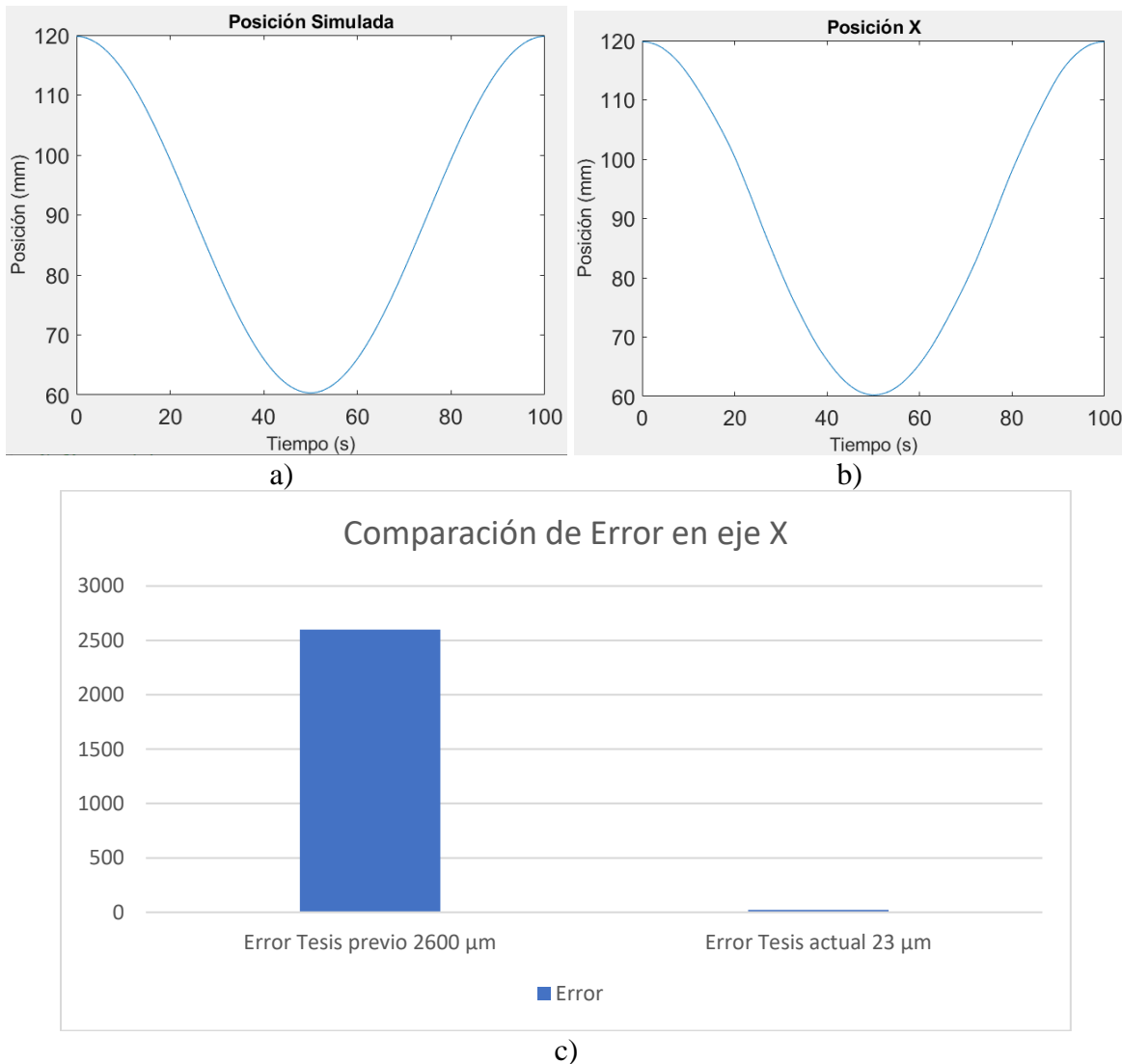


Figura 5.1.1: Posición del eje X: a) simulada con parámetros agrupados, b) resultados experimentales, c) comparación de error.

Para el cálculo de la posición de forma experimental se emplean como sensores los encoders que se encuentran en los motores, los cuales nos entregan señales cuadradas A y B de la cual es posible extraer cuando da un paso y en qué dirección lo da, en el firmware se lleva a cabo un conteo de los pasos empleando un decodificador y se envían al software donde se multiplica por el paso del tornillo embalado.

En la figura 5.1.2 a) se observa la velocidad simulada la cual es muy similar a la b) que contiene los resultados experimentales finales de la velocidad, si las comparamos por medio de la métrica “metr2ev” obtenemos una similitud del 98.96%, la cual es bastante significativa debido a que previamente se complicaba el hecho de observar la velocidad sin errores debido al efecto de muestreo que generaba ceros y el error de cuantificación que afectaba la estimación de la velocidad.

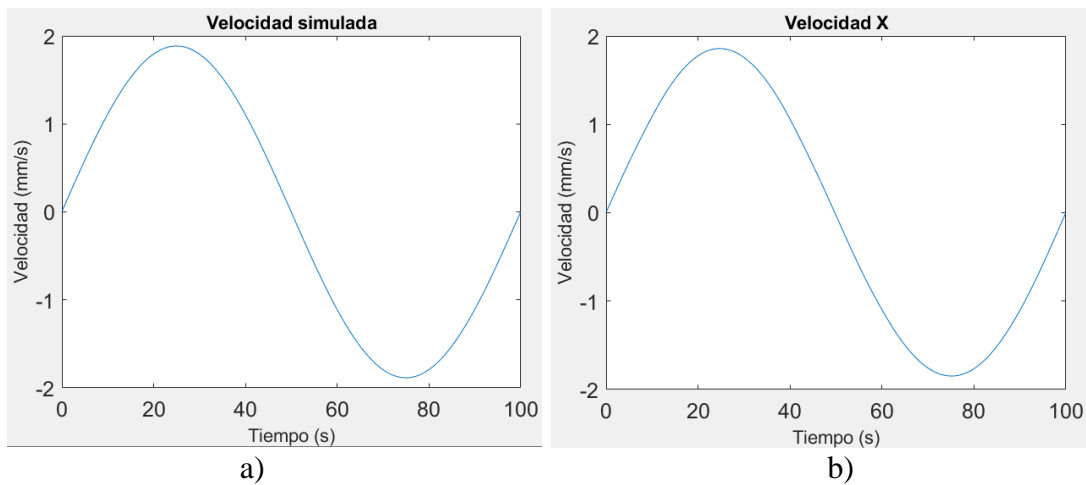


Figura 5.1.2: Velocidad: a) simulada, b) resultados experimentales.

Para lograr esta mejora en la representación de la velocidad, se implementó una metodología propuesta en esta tesis. En primer lugar, para calcular la velocidad, se emplea el método MT. Este método asegura que la estimación de la velocidad solo se realiza cuando hay un cambio en la posición y, además, se realiza un ajuste en el tiempo para corregir el error de sincronización entre el instante del cambio y la lectura del software, para conseguir ese ajuste se realiza un contador en el firmware, mientras que la condición que se emplea para detectar el cambio en la posición, así como el cálculo de la velocidad se ejecuta en el software. En el siguiente paso, se aplica un filtro de Bessel en su forma digital con una frecuencia de corte de 16 Hz. Esto tiene como objetivo corregir el efecto de muestreo, que se vuelve más evidente debido al aumento de momentos en los cuales el sistema interpreta que la velocidad es constante. Esta interpretación se ve influida por las limitaciones en la resolución del sistema.

A pesar de contar con esta gran mejora en la velocidad se requiere mejorar la relación entre el torque y velocidad, esto debido a que el objetivo de estas tesis es obtener el jerk y no será posible calcularlo hasta que se tenga una recuperación confiable de la aceleración.

En la figura 5.1.3 se puede observar el torque experimental entregado por el controlador, en él se pueden observar que está siendo afectado por la fricción de coulomb esto se puede observar en los intervalos en los cuales es cercano a 0 donde se pueden ver cambios muy rápidos para compensar la fricción, además se puede observar la fricción viscosa que se observa al contemplar la escala en que se encuentra trabajando el torque para compensar la fricción, el efecto de Stribeck no se puede observar a simple vista por lo que podemos considerar que es despreciable para esta investigación.

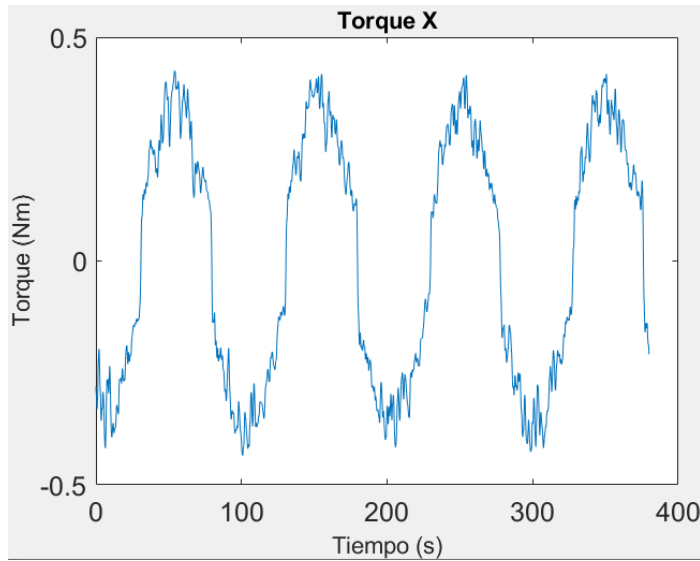


Figura 5.1.3: Torque del eje X sin compensar del controlador.

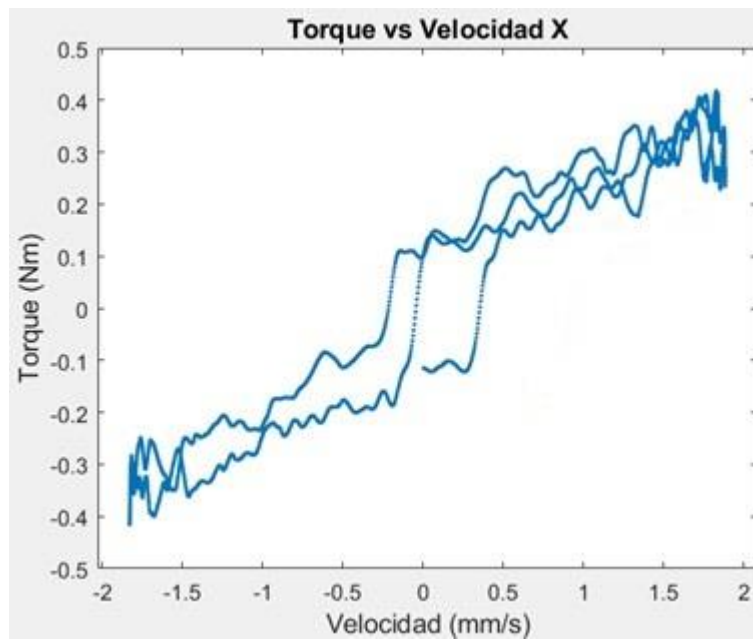


Figura 5.1.4: Torque sin compensar vs velocidad del eje X.

En la figura 5.1.4 se compara la velocidad y el torque en el eje X de esta manera es más fácil observar los efectos de fricción, y es posible obtener las constantes que se requieren para compensar la fricción, para la primera requerimos ver el valor del torque alrededor de la velocidad cuando vale 0, el cual ronda entre 0.14 para el positivo y 0.12 para el negativo, estas serán empleadas para compensar la fricción de coulomb, para la viscosa se saca la pendiente en los intervalos en los extremos, para la parte negativa es aproximadamente 0.3355 y para la parte positiva es aproximadamente 0.139.

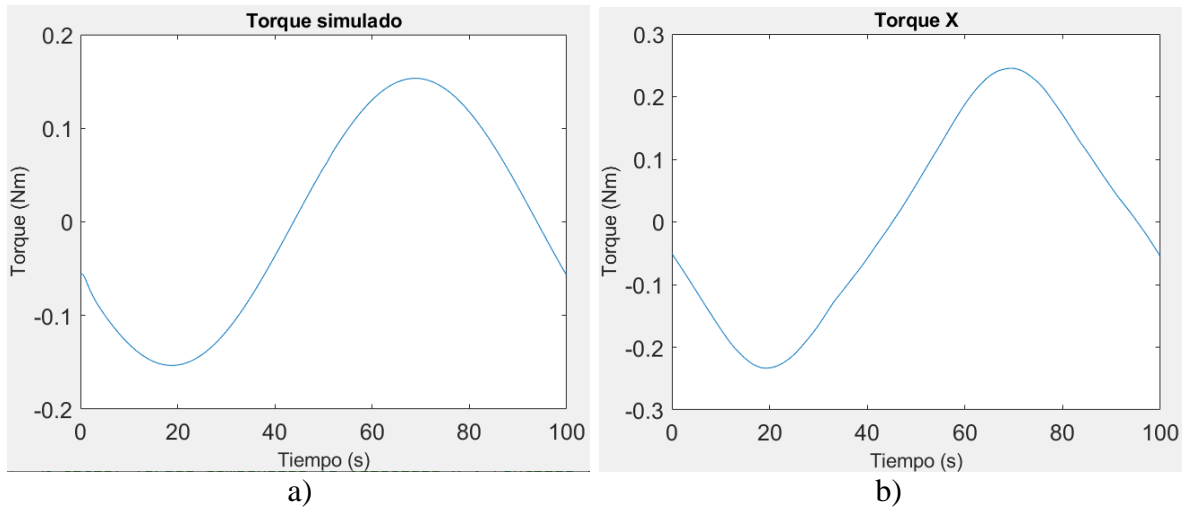


Figura 5.1.5: Torque del controlador del eje X: a) simulado por parámetros agrupados, b) resultado experimental.

En la figura 5.1.5 a) se observa el torque simulado que debería entregado por el controlador; b) se observa el torque experimental entregados por el controlador tras la compensación. Como se puede ver ya no requiere superar la fricción de coulomb y viscosa por lo que el valor disminuye, aunque se puede observar en su forma y valor máximo que aún hay una diferencia de 0.08 Nm esto puede deberse a efectos que restringen el movimiento que no se encuentran modelados, otra diferencia notable es que la experimental es menos suave teniendo una forma que recuerda a la triangular.

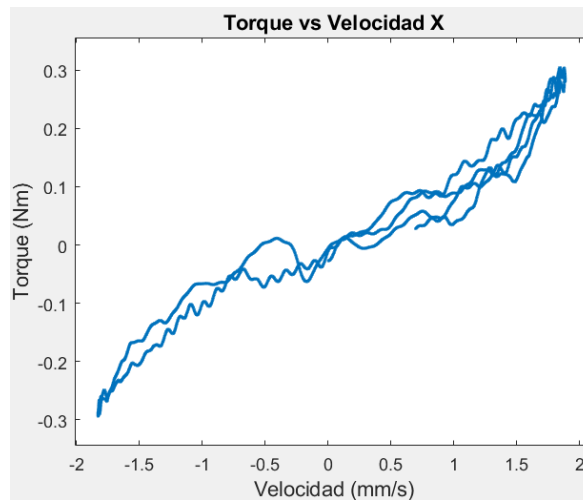


Figura 5.1.6: Torque compensado vs velocidad de eje X.

En la figura 5.1.6 se puede observar la relación que hay ahora entre la velocidad y el torque podemos ver una gran mejora en la relación entre ambas principalmente alrededor de la velocidad en 0 con respecto a la figura 5.1.4, además se han compactado en el resto la gráfica permitiendo que se pueda observar un comportamiento más cercano al de una función lineal, sin embargo, es afectada en gran medida por las oscilaciones y aún presenta 2 pendientes diferentes entre positivo y negativo.

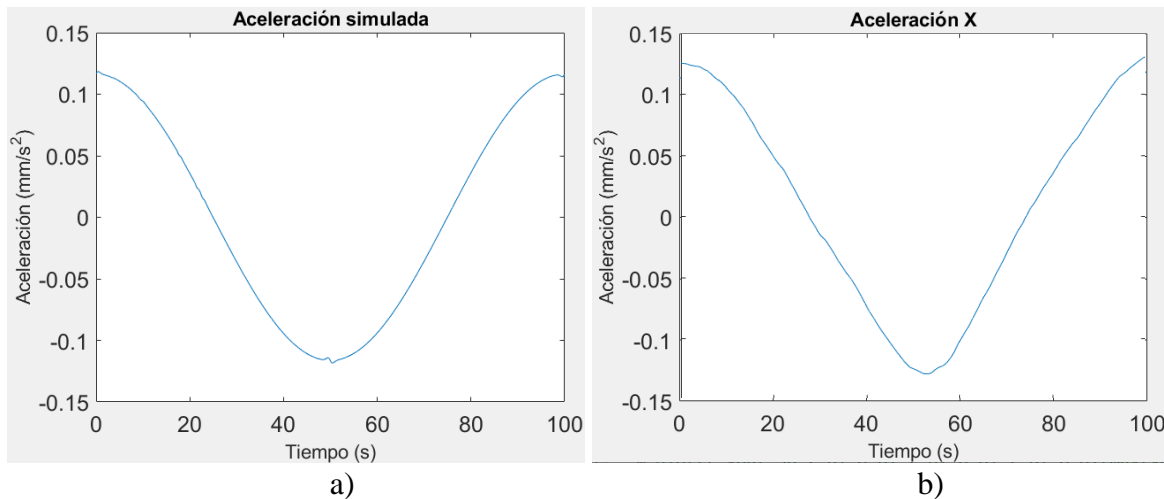


Figura 5.1.7: Aceleración del eje X: a) simulada b) experimental.

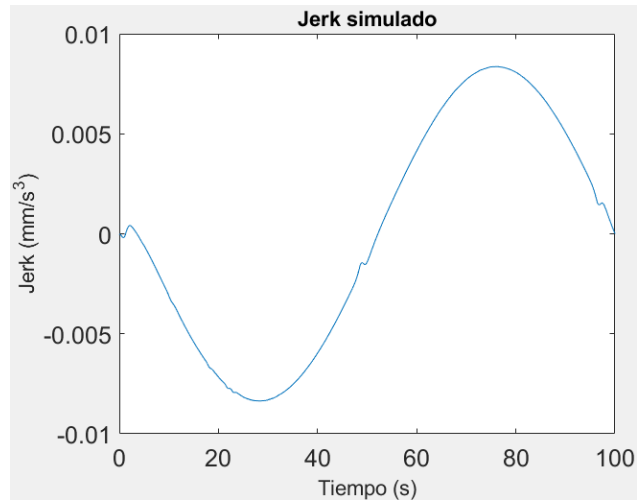
En la figura 5.1.7 a) se observa la aceleración simulada la cual es muy similar a una función coseno con leves deformaciones, mientras que en la b) se puede ver la aceleración experimental, la cual si bien se distingue el parecido no es tan suave y presenta perturbaciones leves. Comparando ambas por la métrica “metr2ev” obtenemos una similitud del 80.75%.

Tras la compensación se puede calcular el valor de la aceleración empleando la velocidad pasada y la velocidad actual que son tomadas después del filtrado, así como el intervalo de muestreo, sólo se le aplicará el mismo filtro que a la velocidad, un filtro Bessel de segundo orden a 16Hz, esto porque el muestreo es producido por el software es el mismo que el de la velocidad, sin embargo, esto sólo es funcional porque se ha observado que la velocidad se encuentra en cambio constante.

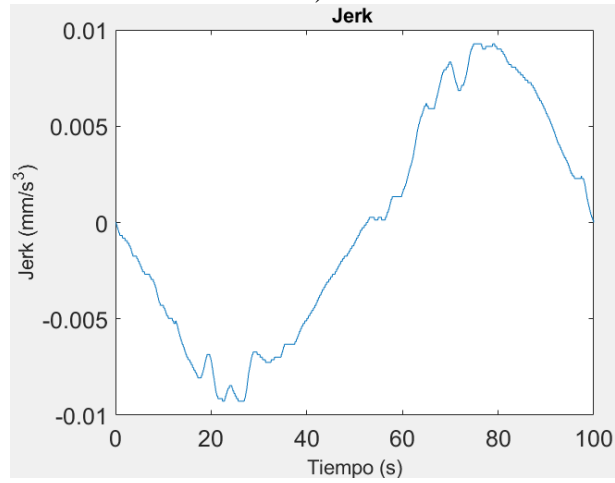
La aceleración es diferente a la simulada esto puede deberse a que no se consiguió una linealidad perfecta en los motores, además de efectos mecánicos que no han sido modelados. Se considera que podría ser mejorado empleando un mejor control, sin embargo, no se considera necesario trabajar en la mejora de su obtención.

En la figura 5.1.8 a) se observa el jerk simulado y en la figura 5.1.8 b) el obtenido de forma experimental aquí es donde vemos mayor diferencia, aunque es posible observar la similitud al seno inverso tiene múltiples perturbaciones en especial en al estar cerca de los valores máximos y mínimos, esto puede deberse a factores no contemplados en la simulación como deformaciones mecánicas, aunque es un resultado mucho mejor al que se presenta en la figura 5.1.8 c) que contiene un intento de obtener el jerk a partir de la posición previo a la investigación en esta tesis, donde se perdía completamente la información. La similitud obtenida por la métrica “metr2ev” es del 58.6%.

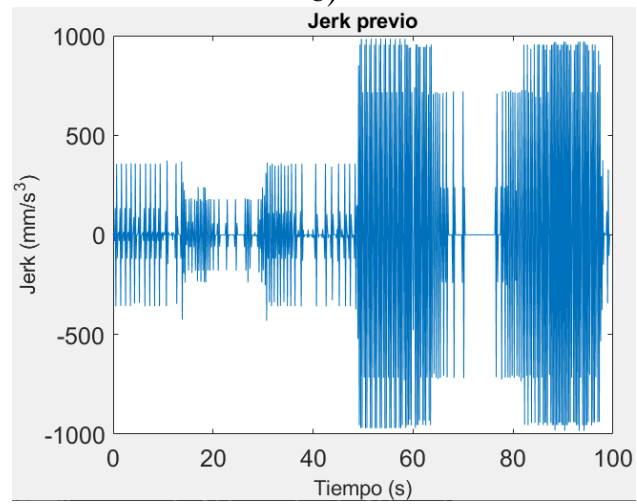
Para obtener el jerk, se realizó un cálculo utilizando la aceleración en el ciclo anterior y la aceleración en el ciclo actual, dividiendo entre el tiempo de muestreo. Nuevamente, se empleó el mismo filtro, un filtro Bessel de segundo orden con una frecuencia de corte de 16 Hz, para eliminar el efecto de muestreo que surge al calcularse a través del software.



a)



b)



c)

Figura 5.1.8: Jerk del eje X: a) simulado, b) experimental, c) obtenido previo a la investigación.

## 5.2 Resultados en el eje Z

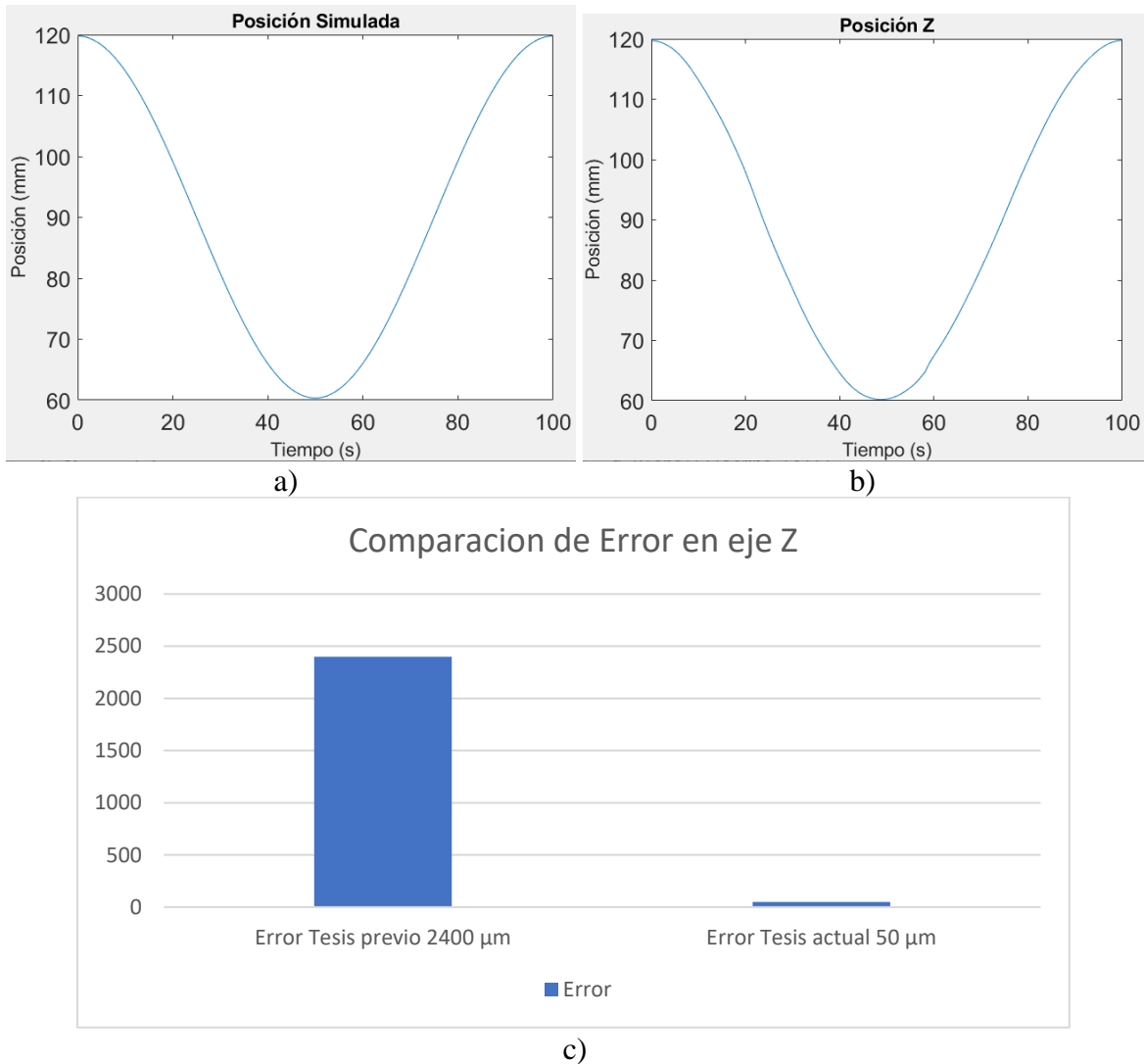


Figura 5.2.1: Posición del eje Z: a) simulada por parámetros agrupados, b) experimental, c) comparación de error en Z.

En la figura 5.2.1 a) se muestra la posición simulada gracias a la metodología de parámetros agrupados y en la b) el resultado experimental, aunque presenta menos perturbaciones que el eje X el error máximo en los extremos es de 0.05mm, por otro lado, al comparar ambas por la métrica “metr2ev” da 99.4% de similitud. La grafica en la figura 5.2.1 c) muestra el error previo [41], a esta investigación siendo del 2400  $\mu\text{m}$  y el actual de 50  $\mu\text{m}$ , estos datos son tomados después de pasar el intervalo de transición, gracias a esto podemos concluir que hubo una gran mejora en el comportamiento del sistema.

De la misma forma que en el eje X para el cálculo de la posición de forma experimental se emplean como sensores los encoders que se encuentran en los motores, los cuales nos entregan señales cuadradas A y B de la cual es posible extraer cuando da un paso

y en qué dirección lo da, en el firmware se lleva a cabo un conteo de los pasos empleando un decodificador y se envían al software donde se multiplica por el paso del tornillo embalado.

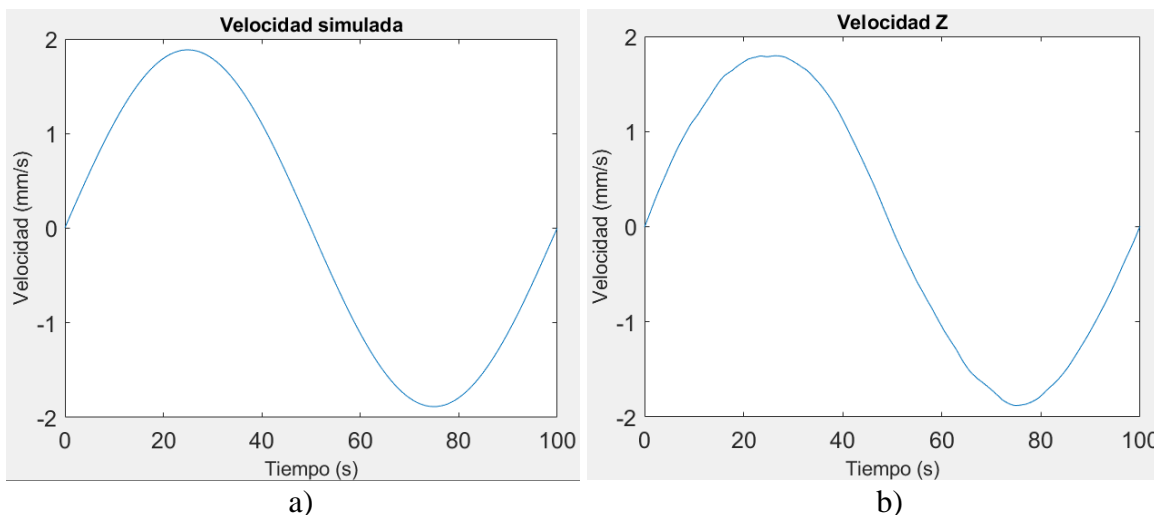


Figura 5.2.2: Velocidad en el eje Z: a) simulada en base a modelado por parámetros agrupados, b) resultado experimental.

En la figura 5.2.2 a) se observa la velocidad simulada usando el modelado por parámetros agrupados para la simulación se empleó Simulink, en la figura 5.2.1 b) se encuentran los resultados experimentales, la diferencia más notable se encuentra en el intervalo negativo sin embargo esto puede deberse a una desconexión a la hora de que la computadora recibe los datos, ya que parece tratarse de una línea, empleando la métrica “metr2ev” se estimó un 97.7% de similitud.

Para la velocidad en el eje Z se emplea la metodología propuesta en esta tesis. Para estimar la velocidad, se emplea el método MT para solo calcular la velocidad cuando cambia la posición, además, se realiza un ajuste en el tiempo para corregir el error de sincronización entre el instante del cambio y la lectura del software, para conseguir ese ajuste se realiza un contador en el firmware, mientras que la condición que se emplea para detectar el cambio en la posición, así como el cálculo de la velocidad se ejecuta en el software. Al igual que en el eje X se aplica un filtro de Bessel en su forma digital con una frecuencia de corte de 16 Hz. Este filtro a demostrador ser efectivo para eliminar el efecto de muestreo sin generar un cambio en la función. El mismo problema con respecto a la transición en el eje X se presenta los primeros ciclos son afectados por el intervalo de transición.

Se repetirá el proceso de comparación entre el torque y la velocidad para poder obtener una aceleración más cercana a la esperada.

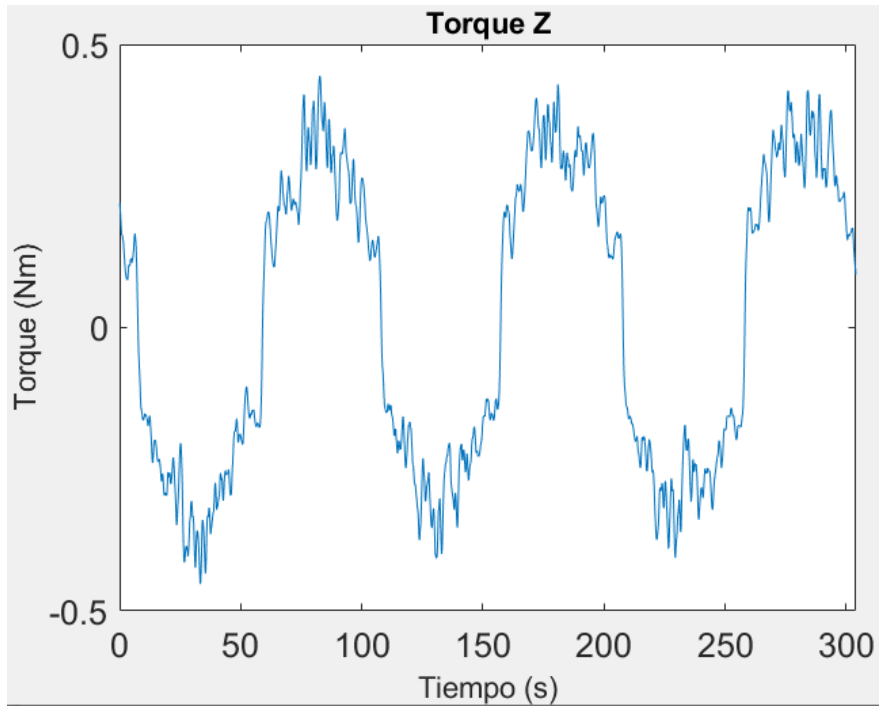


Figura 5.2.3: Torque enviado por el controlado en eje Z sin compensar.

En la figura 5.2.3 se observa el torque en el eje Z de forma similar al del eje X se puede observar la fricción de coulomb alrededor del 0 y la viscosa en la cantidad de torque requerido para su funcionamiento, el efecto de Stribeck puede observarse levemente a diferencia del eje X esto al verse picos tras alejarse del 0, sin embargo, este valor aún es despreciable así que no será compensado.

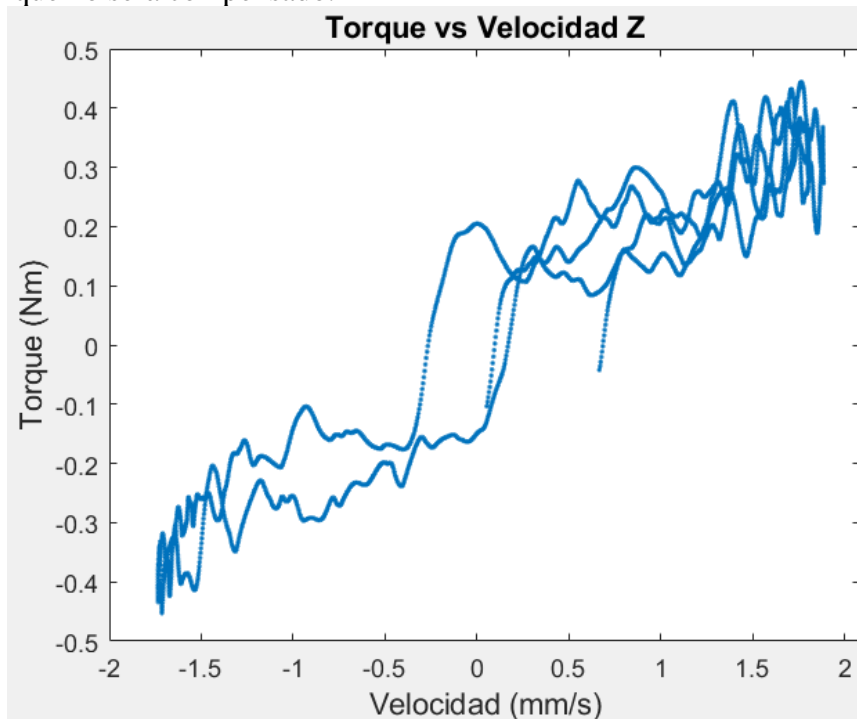


Figura 5.2.4: Torque vs velocidad en eje Z sin compensar.

En la figura 5.2.4 se observa el torque contra la velocidad en el eje Z, en este podemos ver que se presenta la fricción de coulomb alrededor del instante en que la velocidad es 0 y la fricción viscosa con las pendientes en el resto de la gráfica. Además, se presentan varias vibraciones al igual que en el eje X, para compensarlo se requiere el valor del torque alrededor de la velocidad en 0, para el positivo se tiene 0.2 Nm y para el negativo -0.15 Nm, la pendiente positiva es de 0.17 y para la negativa es 0.15.

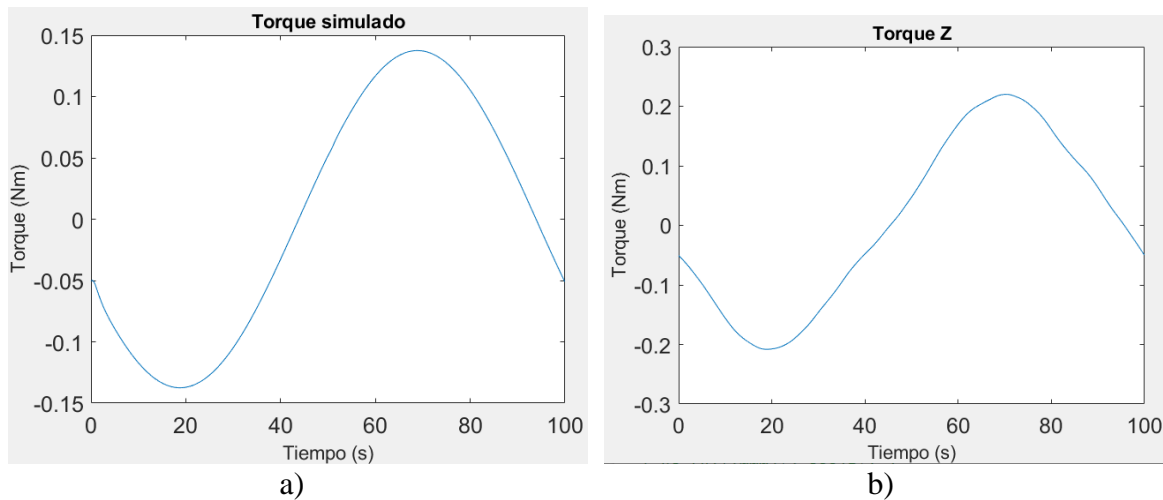


Figura 5.2.5: Torque vs velocidad en eje Z sin compensar.

En la figura 5.2.5 a) se observa el torque del controlador simulado y en la figura 5.2.5 b) se encuentra el torque experimental del controlador tras la compensación. Se puede observar cómo cambia alrededor del 0 además de como disminuyo el máximo y mínimo, esta vez se puede ver claramente que la forma de la función en vez de ser sinusoidal tiene un comportamiento similar a la una triangular además de que el torque requerido es superior al simulado.

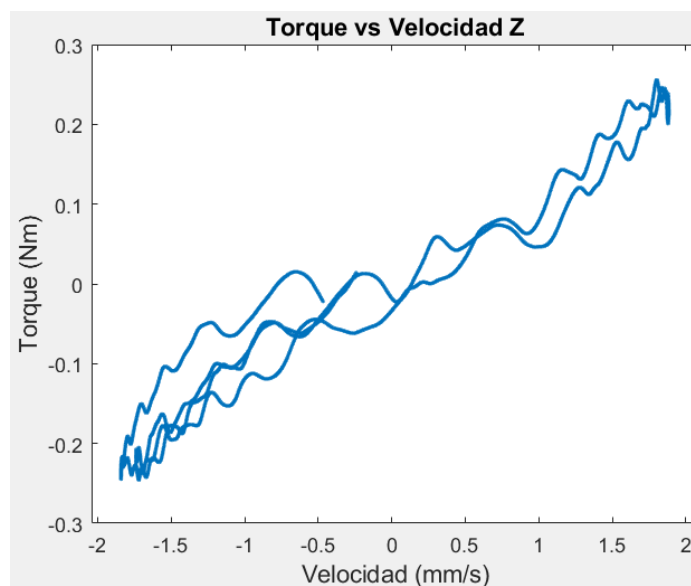


Figura 5.2.6: Torque compensado vs velocidad en eje Z.

La figura 5.2.6 se trata del torque entregador por el controlador tras la compensación de la fricción contra la velocidad del motor en el eje Z como se puede observar a pesar de la oscilación se distingue que se ha eliminado la fricción de coulomb y tiene una pendiente muy similar en la parte positiva y negativa por lo que tenemos una muy buena compensación.

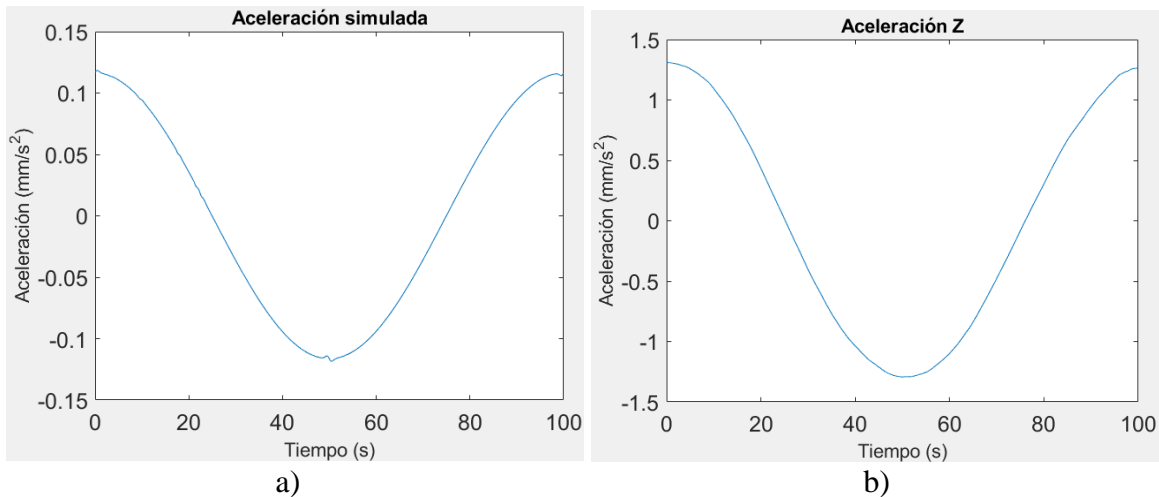


Figura 5.2.7: Aceleración en eje Z: a) simulada, b) experimental.

En la figura 5.2.7 a) se encuentra la aceleración simulada y en 5.2.7 b) se observa la experimental. Aquí podemos estar seguros de que el problema en la velocidad era por comunicación, pues se nota un gran parecido entre la aceleración simulada y experimental, al compararlas por medio de la métrica “metr2ev” da una similitud del 91.62%. Para el cálculo de la aceleración se tomó la velocidad actual y pasada después del filtro y se dividió entre el intervalo de muestreo.

En la figura 5.2.8 a) se observa el jerk simulado y en la 5.2.8 b) el experimental. Como se puede observar, la simulación no tiene un comportamiento ideal, por otro lado, el experimental presenta varias diferencias con el simulado, se nota especialmente en los máximos y mínimos. Además, se observa un comportamiento lineal conforme se acerca cero, esto puede ser por pérdida de datos, a pesar de esto se distingue la forma de seno negativo que se esperaba muy diferente a la figura 5.2.8 c), la cual contiene lo que se obtenía al calcular el jerk con base en la posición, previo a la investigación de esta tesis.

El jerk se pudo recuperar y con ello se pueden observar que aun cuenta con varias deformaciones en comparación con la aceleración, posiblemente si se controla la aceleración estas pueden disminuir. Aun así, se trata de una gran mejora en comparación con el jerk obtenido previo a esta tesis, pues se puede ver claramente su similitud a una función sinusoidal. La similitud entre ambas empleando la métrica “metr2ev” es del 65.96%.

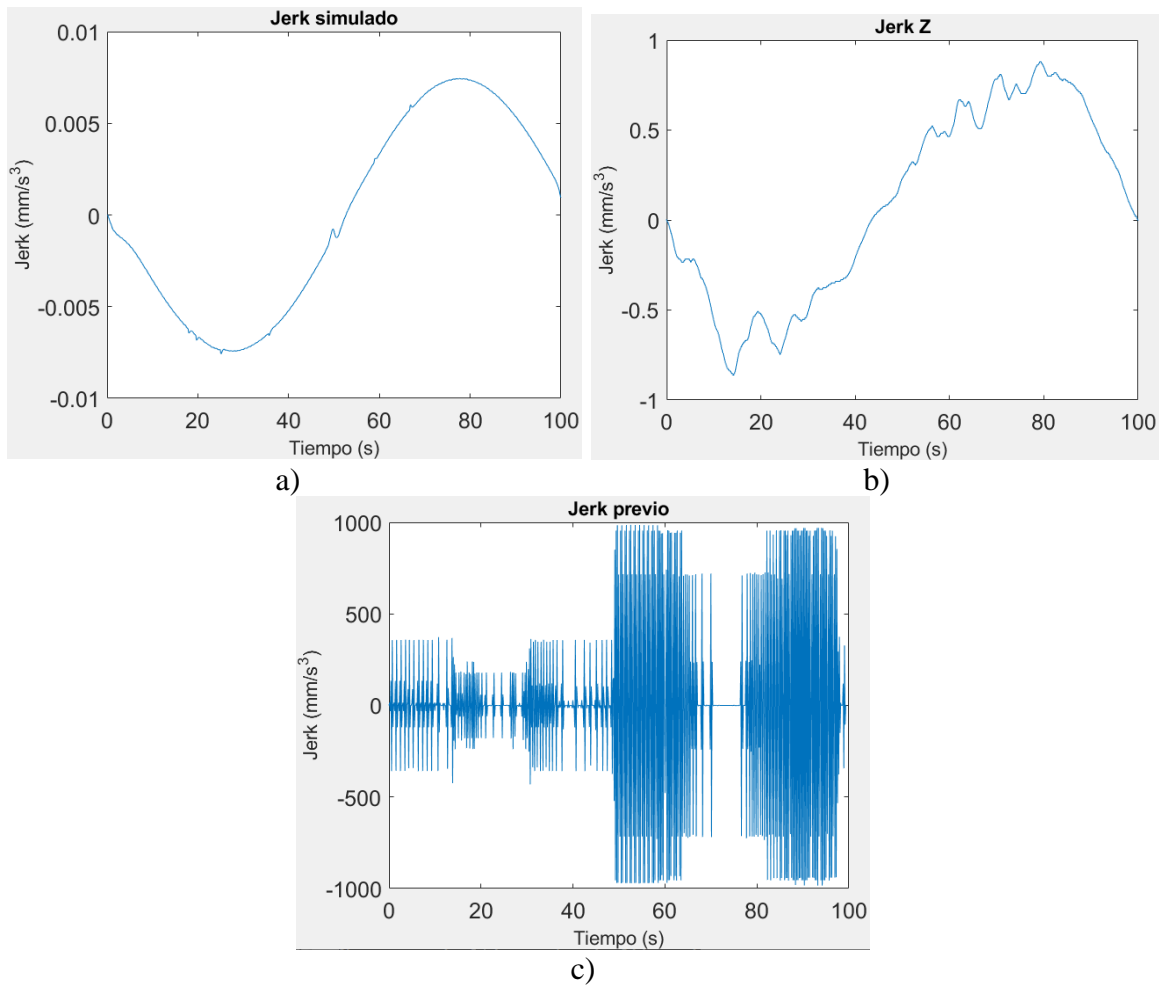


Figura 5.2.8: Jerk en eje Z: a) simulado, b) experimental, c) obtenido previo a la investigación.

### 5.3 Conclusiones

En este trabajo se han logrado mejoras significativas en comparación con el comportamiento del robot de 2 grados de libertad antes de esta investigación. Se ha observado una reducción considerable en el error mientras el robot antes mencionado sigue una trayectoria sinusoidal. Se ha comprobado el funcionamiento del método MT para la medición de la velocidad. Además, se ha demostrado que la compensación de la fricción mejora la relación entre el torque y la velocidad, llegando a ser casi lineal, lo que resulta en un mejor funcionamiento del sistema. Esta mejora permite obtener datos más precisos en tiempo real, como la aceleración y el jerk. Esto representa un avance importante, ya que previamente los datos aceleración y jerk no mostraban una señal legible.

## Conclusiones Generales.

Se han cumplido satisfactoriamente todos los objetivos planteados en esta tesis:

- El sistema ha sido restaurado y ahora muestra un comportamiento que puede considerarse dentro del rango industrial que va de 23 micras en el eje X (1 milésima de pulgada) y de 50 micras en el eje Y (2 milésimas de pulgada).
- El modelado por parámetros agrupados ha demostrado ser un método eficaz para describir con precisión el comportamiento del sistema, ya que se ha logrado alcanzar un 99.1% en X y un 99.4% de coincidencia entre la respuesta real y el modelo dinámico.
- La metodología MT utilizada para la obtención de la velocidad ha resultado efectiva al eliminar los problemas de cuantificación que producían valores erróneos de 0 o infinitos. Además, el filtrado instrumentado en el lenguaje D y realizado en tiempo real, mostró mejoras en el desempeño del robot, ya que elimina vibraciones producidas por la digitalización. La instrumentación de la metodología MT y la cuantificación de la velocidad se hace por medio de un bloque de firmware contenido en el FPGA.
- Así como la velocidad se calcula en función de los datos de la posición, la aceleración se obtiene en función de los datos de la velocidad, cabe señalar que también estos datos se filtran en tiempo real por medio del lenguaje D. Todo lo anterior permite que los datos experimentales de aceleración sean de excelente calidad, ya que concuerdan en 91% con los resultados obtenidos en la simulación.
- El jerk medido experimentalmente no es exactamente el esperado de acuerdo con la simulación, sin embargo, muestra que la mayoría de las vibraciones antes presentes en el robot han desaparecido y ahora su forma corresponde en un 61% con los resultados obtenidos en la simulación del modelo dinámico, por lo que representa una gran mejora.
- La compensación de la fricción estática y viscosa produjo una relación entre el torque y la velocidad casi lineal y esto contribuyó a un mejor desempeño del controlador.

Como trabajo a futuro se puede mejorar el modelado por parámetros agrupados, considerando fenómenos como imperfecciones mecánicas y otros factores que puedan afectar el comportamiento del sistema. También se puede utilizar la inteligencia artificial para realizar la identificación paramétrica como intento de mejorar el desempeño del controlador de trayectoria que se está utilizando en este trabajo. La metodología propuesta para determinar la velocidad, la aceleración y el jerk se puede aplicar a otros robots. Utilizando las mejoras propuestas en este trabajo se puede continuar desarrollando robots con aplicaciones en la industria.

## Bibliografía.

- [1] Hess DP, Soom A. Friction at a lubricated line contact operating at oscillating sliding velocities. *J Tribol* 1990;112(1):147–52.
- [2] B. Fabien. *Analytical System Dynamics: Modeling and Simulation*. Springer US, 2008.
- [3] Vargas-Trevino, M.A., Lopez-Gomez, J., et al., “A mechatronic approach for ball screw drive system: modeling, control, and validation on an FPGA based architecture”, *Int J Adv Manuf Technol*, vol. 104, pp. 2329- 2346, Jul 2019.
- [4] Chien-Yu Lin, and Ching-Hung Lee, “Remote Servo Tuning System for Multi-Axis CNC Machine Tools Using a Virtual Machine To Approach,” *Applied Sciences*, vol. 7, pp. 776, July 2017.
- [5] Marton L, Lantos B. Modeling: identification, and compensation of stick-slip friction. *IEEE Trans Indl Electron* 2007;54(1):511–21.
- [6] K. Aljaseem, L. Froehly, A. Seifert, and H. Zappe, “Scanning and Tunable Micro-Optics for Endoscopic Optical Coherence Tomography,” *Journal of Microelectromechanical Systems*, vol. 20, pp. 1462-1472, December 2011.
- [7] D. Hampai, A. Liedl, G. Cappuccio, E. Capitolo, M. Iannarelli, M. Massussi, S. Tucci, R. Sardella, A. Sciancalepore, C. Polese, and S. B. Dabagov, “2D-3D XRF elemental mapping of archeological samples,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 402, pp. 274-277, July 2017.
- [8] D. Cocco, M. Idir, D. Morton, L. Raimondi, and M. Zangrando, “Advances in X-ray optics: From metrology characterization to wavefront sensing-based optimization of active optics,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 907, pp. 105–115, November 2018.
- [9] Yi-Cheng Huang, and Xiang-Yuan Chen, “Investigation of a Ball Screw Feed Drive System Based on Dynamic Modeling for Motion Control,” *Advances in Technology Innovation*, vol. 2, pp. 29–33, January 2017.
- [10] D. Lyu, Q. Lyu, H. Lyu et al., “Dynamic error of CNC machine tools: a state-of-the-art review,” *The International Journal of Advanced Manufacturing Technology*, vol. 106, pp. 1869–1891, January 2020.
- [11] H. Yang, Z. Wang, T. Zhang et al., “A review on vibration analysis and control of machine tool feed drive systems,” *Int J Adv Manuf Technol*, vol. 107, pp. 503–525, March 2020.
- [12] Z. Mei; J. Ding; L. Chen; T. Pi; Z. Mei, “Hybrid Multi-Domain Analytical and Data-Driven Modeling for Feed Systems in Machine Tools,” *Symmetry*, vol. 11, pp. 1156, September 2019.
- [13] M. Fatih Corapsiz and Koksal Erenturk, “Trajectory Tracking Control and Contouring Performance of Three-Dimensional CNC,” *IEE Transactions on Industrial Electronics*, vol. 63, pp. 2212–2220, April 2016.
- [14] M. Nandayapa, C. Mitsantisuk, and K. Ohishi, “Enhancing transparency on a ball screw teleoperation robot system using a novel velocity estimation method with FPGA,” *Advanced Robotics*, vol. 27, pp. 211–222, January 2013.

- [15] P. Ponce, A. Molina, H. Bastida, and B. MacCleery, “Real-time hardware ANN-QFT robust controller for reconfigurable micro-machine tool,” *Int J Adv Manuf Technol*, vol. 79, pp. 1–20, January 2015.
- [16] A. Alabdo, J. Perez, G. J. Garcia, J. Pomares, and F. Torres, “FPGA based architecture for direct visual control robotic systems,” *Mechatronics*, vol. 39, pp. 204–216, June 2016.
- [17] R. C. Sampaio, J. M. S. T. Motta, and C. H. Llanos, “An FPGA-based controller design for a five degrees of freedom robot for repairing hydraulic turbine blades,” *J Braz. Soc. Mech. Sci. Eng.*, vol. 39, pp. 3121–3136, May 2017.
- [18] A. Beghi, F. Marcuzzi, P. Martin, F. Tinazzi, and M. Zigliotto, “Virtual prototyping of embedded control software in mechatronic systems: A case study,” *Mechatronics*, vol. 43, pp. 99–111, March 2017.
- [19] I. De La Cruz-Malagon, A. Favela-Contreras, and A. Avila, “Performance-Improved Implementation of the SISO Adaptive Predictive Control Algorithm for Embedded Systems,” *IEEE Transactions on Industrial Electronics*, vol. 67, pp. 8054–8063, September 2020.
- [20] Micco, Luciana & Vargas, Fabian & Fierens, P. (2019). A Literature Review on Embedded Systems. *IEEE Latin America Transactions*. 18. 188-205. 10.1109/TLA.2019.9082229.
- [21] S. Vyas, A. Gupte et al., “Hardware Architectural Support for Control Systems and Sensor Processing,” *ACM Transactions on Embedded Computing Systems*, vol. 13, pp. 16:1–16:25, September 2013.
- [22] M. Martinez-Prado, A. Franco-Gasca, G. Herrera-Ruiz, and O. Soto-Dorantes, “Multi-axis motion controller for robotic applications implemented on an FPGA,” *Int J Adv Manuf Technol*, vol. 67, pp. 2367–2376, January 2013.
- [23] C. I. Muresan, S. Folea, G. Mois, and E. H. Dulf, “Development and implementation of an FPGA based fractional order controller for a DC motor,” *Mechatronics*, vol. 23, pp. 798–8046, May 2013.
- [24] S. Amornwongpeeti, M. Ekpanyapong, et al., Nattapon Chayopitak, “A single chip FPGA-based solution for controlling of multi-unit PMSM motor with time-division multiplexing scheme,” *Microprocessors and Microsystems*, vol. 39, pp. 621–633, September 2015.
- [25] S. Khubalkar, A. Junghare, M. Aware, A. S.Chopade, and S. Das, “Demonstrative fractional order – PID controller based DC motor drive on digital platform,” *ISA Transactions*, vol. 82, pp. 79–93, November 2018.
- [26] J. J. Medel, Pedro Guevara López y Alberto Flores Rueda, *Caracterización de Filtros Digitales en Tiempo Real para Computadoras Digitales*, Laboratorio de Electrónica, Centro de Investigación en Computación – IPN C. P. 07738 México D. F (2004)
- [27] Chang, Yih-Fang ; “Design and implementation of a linear jerk filter for a computerized numerical controller”; *Control Engineering Practice*, Volume 13, Issue 5, Pages 567-576. (2005)
- [28] Vargas-Trevino, M.A., Lopez-Gomez, J., et al., “A mechatronic approach for ball screw drive system: modeling, control, and validation on an FPGA based architecture”, *Int J Adv Manuf Technol*, vol. 104, pp. 2329- 2346, Jul 2019.
- [29] F. Reyes, *Robótica*. Control de robots manipuladores. México: Alfaomega, 2011.

- [30] R. Kelly, V. Santibáñez D. A. Loría (2005) PD Control with Gravity Compensation. In: Control of Robot Manipulators in Joint Space. Advanced Textbooks in Control and Signal Processing. Springer, London. [https://doi.org/10.1007/1-85233-999-3\\_9](https://doi.org/10.1007/1-85233-999-3_9).
- [31] Ing. Zepeda Arce, G., “DISEÑO E IMPLEMENTACIÓN DE UNA TARJETA CON FPGA RAM Y ROM EXTERNAS”, Faculty of Electronics Sciences of the Benemérita, Autonomous University of Puebla (FCE-BUAP), Puebla city, Mexico. (2018)
- [32] Galván Salazar, Karla Rocío, & Carlos Pérez, Héctor de Jesús (2006). Redes Inalámbricas 802.11n en el Nuevo Estándar. Conciencia Tecnológica, (32),0.[fecha de Consulta 25 de Mayo de 2022]. ISSN: 1405-5597. Disponible en: <https://www.redalyc.org/articulo.oa?id=94403212>
- [33] Eric Griffith (2002), WECA becomes Wi-Fi Alliance, 80211-Planet.
- [34] Intel(2021), Diferentes protocolos de Wi-Fi y velocidades de datos, Información y documentación sobre productos, 28/10/2021, Id: 000005725, <https://www.intel.la/content/www/xl/es/support/articles/000005725/wireless/legacyintelwireless-products.htm>
- [35] D. M. Gonzalez Arriaga, M.A.D. Vargas Treviño, S. Vergara Limon “Lenguaje D” Programas de computacion, Registro Publico del Derecho de Autor, numero de registro: 03-2021-120110454200-01, 2021.
- [36] [HSU, H. P. ANALISIS DE FOURIER (1a. 1ra ed.), WILMINGTON: ADDISON WESLEY LONGMAN de Mexico S.a de C.V., 1987, Pages (102-151)
- [37] Ashu Soni and Maneesha Gupta, , Analysis and Design of Optimized Fractional Order Low-Pass Bessel Filter, Journal of Circuits, Systems and Computers VOL. 30, NO. 02 2020.
- [38] Arthur Williams, Fred J Taylor, Electronic Filter Design Handbook, McGraw-Hill's, Julio 2006, ISBN 0071471715.
- [39] Hace, Ales & Čurkovič, Milan. Accurate FPGA-Based Velocity Measurement with an Incremental Encoder by a Fast Generalized. 2018.
- [40] Margarita Carmina García López, Control de seguimiento de trayectoria y contorneado basado en el modelo de fricción para un sistema CNC de feed-drive, Faculty of Electronics Sciences of the Benemérita, Autonomous University of Puebla (FCE-BUAP), Puebla city, Mexico. (2022)
- [41] Lic. Torres Reyes R., “Análisis, Diseño y Puesta en Marcha de un Sistema Cartografo Semi-Automatico Multi-Posicionador Modular de Resolución Controlada” Faculty of Electronics Sciences of the Benemérita, Autonomous University of Puebla (FCE-BUAP), Puebla city, Mexico. (2016)
- [42] C. Diaz D.L., Carlos L. & Vergara-Limon, Sergio & Vargas-Treviño, María & López Gómez, Jesús & Gonzalez Calleros, Juan & Arriaga, Daniel Marcelo. Parameter Identification of a Robot Arm Manipulator Based on a Convolutional Neural Network. IEEE Access. 10. 55002 - 55019. 10.1109/ACCESS.2022.3177209. (2022).
-

# Apéndice 1: Artículo publicado

Publicado en Artículos del Congreso Internacional de Investigación Academia Journals Puebla TecNM con ISSN online 1946-5351, Volumen 15, No. 1, 2023.

Ingeniería para la Innovación y  
Desarrollo Tecnológico Sostenible

© Academia Journals 2023

Puebla, Puebla, México  
22, 23, y 24 de febrero, 2023

## Obtención y Filtrado Digital de la Posición y Velocidad de un Robot Cartesiano de Dos Grados de Libertad por Medio de un Sistema Embebido en Plataforma FPGA

Lic. Luis Arturo Alcantar Vergara<sup>1</sup>, Dra. María Aurora Dioxcora Vargas Treviño<sup>2</sup>, Dr. Sergio Vergara Limón<sup>3</sup>, Dr. Jesús López Gómez<sup>4</sup>, Dra. Amparo Palomino Merino<sup>5</sup>, Dra. Olga Guadalupe Félix Beltrán<sup>6</sup>, M.C. Margarita Carmina García López<sup>7</sup>

**Resumen**—En el presente artículo se presenta el modelo dinámico de un robot cartesiano con 2 grados de libertad (2GDL) obtenido con la metodología de parámetros agrupados, se empieza por generar un diagrama del sistema, en el cual se representa cada elemento donde se tenga un cambio de energía, se generan sus ecuaciones; posteriormente se realiza una simulación del modelo empleando la herramienta Simulink en el programa de Matlab. Se realizan los experimentos con el robot real utilizando un sistema embebido en plataforma FPGA para poder realizar control de posición y velocidad, se da una explicación de la metodología propuesta para filtrar digitalmente las señales obtenidas utilizando filtros digitales de tipo Bessel y estimación MT mostrando finalmente los resultados, las conclusiones y el trabajo a futuro de esta investigación.

**Palabras clave**— Modelado matemático, parámetros agrupados, control, FPGA.

### Introducción

La Ingeniería de control es una disciplina que se focaliza en modelar matemáticamente una gama diversa de sistemas dinámicos y el diseño de controladores que harán que estos sistemas se comporten de la manera deseada. El modelo matemático consiste en un sistema mecánico modelado acoplado el modelo eléctrico de los motores. Se utiliza la metodología parámetros agrupados con el objetivo de tener un sistema más detallado. Se emplea la ecuación de Lagrange basado en el análisis de las energías cinética y potencial de un sistema rígido para obtener el modelo dinámico del sistema mecánico [1]. Con el modelo dinámico de la transmisión mecánica y el modelo eléctrico del motor se obtendrá el modelo matemático general para evaluar mediante simulaciones la respuesta dinámica del sistema. El sistema de posicionamiento consiste en aplicar un voltaje o torque modulado por ancho de pulso (PWM) controlado en la entrada del motor de DC en el que se agrupa a un reductor, un acoplamiento, un eje de tornillo de bola, una tuerca y el bloque deslizante. Los sistemas mecatrónicos de posicionamiento requieren un algoritmo de control para realizar trayectorias o posiciones de alta precisión. Por otro lado, la adquisición de datos y el diseño del sistema embebido configurable es una parte muy importante para garantizar el procesamiento en hardware, alto rendimiento, recepción y envío de datos que generalmente está definido por las aplicaciones a las que se destina. Con la finalidad de obtener señales lo más limpias posibles en los sistemas mecatrónicos, se utilizan diferentes tipos de acondicionamiento de dichas señales, uno de estos son el filtrado y así suavizar las señales, esto permite obtener mejora en la lectura de los datos, esto lleva a poder emplear de mejor forma los datos obtenidos pues se vuelven fiables permitiendo que se usen para sistemas de control. En el presente trabajo se explica la implementación de un filtro tipo Bessel dentro del sistema embebido que nos permite tener más limpias las señales que vienen del robot, así como la instrumentación del método MT para eliminar el error de cuantificación en la velocidad.

<sup>1</sup> Lic. Luis Arturo Alcantar Vergara estudiante de Maestría en Electrónica Opción en Automatización en la Benemérita Universidad Autónoma de Puebla, Puebla, luis.alcantar@alumno.buap.mx

<sup>2</sup> Dra. María Aurora Dioxcora Vargas Treviño Profesora investigadora en la Benemérita Universidad Autónoma de Puebla, Puebla, aurora.vargas@correo.buap.mx

<sup>3</sup> Dr. Sergio Vergara Limón Profesor investigador en la Benemérita Universidad Autónoma de Puebla, Puebla, sergio.vergara@correo.buap.mx

<sup>4</sup> Dr. Jesús López Gómez, Profesor investigador en la Universidad Juárez Autónoma de Tabasco, DAIA-UJAT, Tabasco, México, jesuslopez.asaf@gmail.com

<sup>5</sup> Dra. Amparo Palomino Merino Profesora investigadora en la Benemérita Universidad Autónoma de Puebla, Puebla, amparo.palomino@correo.buap.mx

<sup>6</sup> Dra. Olga Guadalupe Félix Beltrán, Profesor investigador en la Benemérita Universidad Autónoma de Puebla, Puebla, olga.felix@correo.buap.mx

<sup>7</sup> M.C. Margarita Carmina García López, estudiante de Doctorado en ingeniería del lenguaje y del conocimiento en la Benemérita Universidad Autónoma de Puebla, margaritacarmina.garcialopez@viep.com.mx

### Descripción del Método

Para el modelado del robot se empleó la metodología de parámetros agrupados en la cual representa cada cambio de la energía del sistema en cada elemento que lo conforma, se obtiene la energía cinética ( $E_c$ ), la energía potencial ( $E_p$ ), la energía disipativa ( $D_r$ ) y el trabajo virtual ( $w$ ), también se considera la dependencia del eje  $Y$  al  $X$  con relación a las masas y gravedad, después se aplica Euler Lagrange para obtener las ecuaciones características del sistema [1].

Las ecuaciones resultantes son las siguientes [2]:

$$L_i + R I_i + k_{ei} \dot{\theta}_{mi} = E_i \quad (1)$$

$$J_{m_i} \ddot{\theta}_{m_i} + b_{m_i} \dot{\theta}_{m_i} + b_{2i} [G \dot{\theta}_{2i} - \dot{\theta}_{m_i}] - \tau_{2i} = \tau_{m_i} \quad (2)$$

$$G^2 J_{2i} \ddot{\theta}_{2i} - k_{2i} [\theta_{2i} - \theta_{2i}] + b_{2i} G (\dot{\theta}_{2i} - \dot{\theta}_{m_i}) + b_{2i} (\theta_{2i} - \theta_{2i}) = G \tau_{2i} \quad (3)$$

$$J_{ci} \ddot{\theta}_{ci} + k_{ci} (\theta_{ci} - \theta_{2i}) - k_{2i} (\theta_{2i} - \theta_{2i}) + b_{2i} (\dot{\theta}_{2i} - \dot{\theta}_{ci}) - b_{2i} (\theta_{ci} - \theta_{2i}) = 0 \quad (4)$$

$$J_{ti} \ddot{\theta}_{ti} + K_{ti} (\theta_{ti} - \theta_{2i}) - b_{2i} (\dot{\theta}_{2i} - \dot{\theta}_{ti}) - R n b_{ni} (\dot{X}_{ti} - (\dot{X}_{ti} + R n \dot{\theta}_{2i})) = R n F n \quad (5)$$

$$(m_{2i}) \ddot{X}_{2i} + (k_{r_i} X_{2i}) + b_{r_i} \dot{X}_{2i} - b_{ni} [\dot{X}_{ti} (\dot{X}_{ti} + R n \dot{\theta}_{2i})] = F n \quad (6)$$

$$(m_{ti}) \ddot{X}_{ti} + b_{ni} (\dot{X}_{ti} (\dot{X}_{ti} + R n \dot{\theta}_{2i})) = -F n - F f \quad (7)$$

Donde  $i$  indica el grado de libertad del robot. Así mismo los términos  $m_{2i}$  y  $m_{ti}$  denotan las masas totales que carga cada eje del robot, en la Tabla 1 se describen las variables del sistema y su correspondiente definición.

S	Significado	S	Significado
$I_i$	Corriente eléctrica del motor [A]	$E_i$	Voltaje aplicado al motor [V]
$J_{m_i}$	Inercia del motor [kgm <sup>2</sup> ]	$b_{2i}$	Coefficiente de Fricción caja engranajes [Nm s/rad]
$J_{2i}$	Inercia de la caja de engranajes [kgm <sup>2</sup> ]	$b_{ci}$	Coefficiente de Fricción cople [Nm s/rad]
$J_{ci}$	Inercia de cople [kgm <sup>2</sup> ]	$b_{ti}$	Coefficiente de Fricción t. tornillo embalado [Nm s/rad]
$J_{ti}$	Inercia del tornillo embalado [kgm <sup>2</sup> ]	$b_{r_i}$	Coefficiente de Fricción a. tornillo embalado [Ns/m]
$m_{2i}$	Masa del tornillo embalado [m]	$b_{ni}$	Coefficiente de Fricción mesa trabajo [Nm/s]
$m_{ti}$	Masa de la mesa trabajo [m]	$G_i$	Caja engranajes
$R_i$	Resistencia motor [Ohm]	$\theta_{m_i}$	Posición angular eje del motor [rad]
$L_i$	Inductancia del motor [H]	$\theta_{2i}$	Posición angular Eje de la caja del motor [rad]
$k_{2i}$	Rigidez de caja de engranajes [Nm/rad]	$\theta_{ci}$	Posición angular eje del cople [rad]
$k_{ci}$	Rigidez de cople [Nm/rad]	$\theta_{ti}$	Posición angular del tornillo embalado [rad]
$k_{ti}$	Rigidez de t. tornillo embalado [Nm/rad]	$X_{ti}$	Posición axial del tornillo embalado [m]
$F_{ni}$	Rigidez mesa de trabajo [N]	$X_{2i}$	Posición axial de la mesa de trabajo [m]
$k_{r_i}$	Rigidez de a. tornillo embalado [N/m]	$q_i$	Carga en el motor [C]

Tabla 1.- Constantes y variables del sistema

### Diagrama general del sistema

La arquitectura del robot se instrumenta en un procesador desarrollado en el laboratorio de Automatización. Este microprocesador está interconectado con "núcleo de propiedad intelectual" (IP Core, por sus siglas en inglés) a través del puerto de entrada y salida (Figura 1), con tareas para leer y escribir datos de propósito específico, contiene un módulo de administración de datos WIFI, administrador de datos en memoria RAM y EEPROM, un módulo generador de señal PWM, decodificador de posición de motor, entre otros. Para el control del robot es necesario decodificar la señal de posición de cada uno de los motores, saber si alguno de las articulaciones ha llegado a su fin de carrera, además de mandar la acción de control a través de cada uno de los motores con la señal PWM. Cabe mencionar que el sistema embebido cuenta con memoria RAM y en ella se cuentan con miles de direcciones de 32 bits disponibles con el fin de almacenar diferentes variables y cálculos necesarios en el lazo de control [3]. El microprocesador implementado dentro de la tarjeta de desarrollo FPGA contiene una unidad aritmética lógica (ALU). Es posible ejecutar las diferentes funciones del microprocesador acorde a las necesidades de nuestro algoritmo a implementar, para lograrlo, se ha desarrollado un compilador que "traduce" el lenguaje de programación de alto nivel propio y llamado lenguaje D a lenguaje máquina [4]. El procesador trabaja de forma nativa en punto flotante, lo que significa que, para hacer operaciones matemáticas y comparaciones, estas se ejecutarán en punto flotante, no obstante, existe la posibilidad de convertir un dato de punto flotante a entero y viceversa.

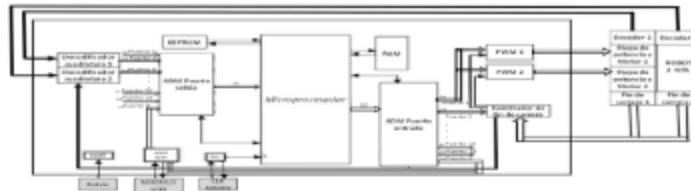


Figura 1. Diagrama general del sistema

**Sistema Embebido**

El sistema embebido consta de las siguientes partes: Tarjeta FPGA DE0-CV [5], dispositivo WiFi, codificador acondicionamiento de señal y circuito controlador de motor para el movimiento mando de dirección (Puente H). Se cuenta con un software de adquisición con comunicación WiFi que se utiliza como una interfaz para enviar parámetros de posiciones y ganancias para el controlador propuesto. La tarjeta FPGA DE0-CV utiliza un Cyclone V FPGA 5CEBA4F23C7N y posee un microprocesador integrado. El Firmware se desarrolla en la programación AHDL (Altera Hardware Description Language) y consta de bloques de instrucciones para fines específicos que se graban en una memoria ROM. Las memorias RAM y ROM son dispositivos externos al FPGA. La memoria ROM está destinada a almacenar la instrucción que interpreta y ejecuta el microprocesador. Sin embargo, los datos de almacenamiento de la memoria RAM están destinados a ser procesados. Además, el firmware de control comprende la implementación en la FPGA de un microprocesador RISC de 32 bits con una arquitectura Harvard, un conjunto de 32 bits con 15 instrucciones basadas en MIPS, 16 registros de propósito general, una unidad aritmética de precisión simple IEEE 754 de punto flotante que controla el sistema de accionamiento del husillo de bolas. El sistema embebido conectado con todas sus partes al robot se muestra en la Figura 2.

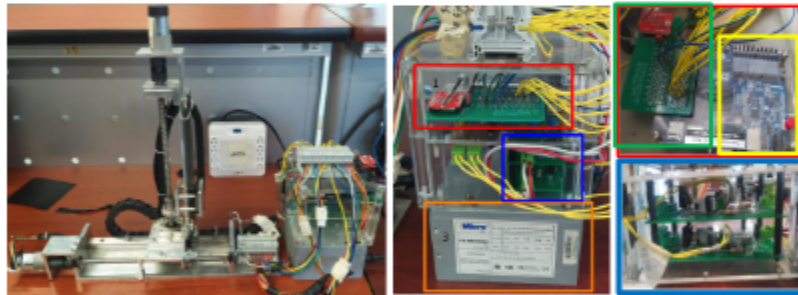


Figura 2.- Sistema embebido: tarjetas acopladas, tarjeta de comunicación con modulo Wify (roja), Puentes H (azul) y Tarjeta De Desarrollo FPGA Cyclone V Intel DE0-CV (amarillo)

El firmware posee diversos bloques que resultan necesarios para los procesos requeridos por el funcionamiento del mecanismo. Los principales son: Administrador de Wi-Fi que permite al FPGA comunicarse con protocolo serial al módulo WiFi, el cual conecta de forma inalámbrica con la computadora. De esta forma se tiene comunicación entre la computadora y la tarjeta del robot por medio del software de control. Administrador de SDRAM que consta de un bloque de control de memoria que permite enviar comandos para escribir y leer datos de 32 bits en forma paralela y se encarga de comunicarse con los datos que manda el administrador Wi-Fi enviados por la computadora. Decodificador de señales que procesa las señales de los encoder de los motores que controlan los ejes del motor. Los encoders incrementales de cuadratura determinan la posición actual en cada eje, para interpretar las señales A y B de los encoders, que están desfasadas 90°, al firmware se le agregaron bloques para decodificar dichas señales, donde se lleva a cabo el conteo de los pulsos de los encoders para poder determinar la posición actual de cada eje y mediante dichos conteos se puede determinar el sentido de giro de cada motor. En las señales A y B se encuentra codificada la información correspondiente al avance y su dirección, a partir de ese valor cualquier cambio que haga el encoder genera un cambio en las combinaciones lógicas de la señal y según el cambio que se genera se puede saber si el giro del motor es en sentido horario o en sentido antihorario. El firmware genera la señal de PWM y controla la velocidad de los motores. Se puede recalcar que el funcionamiento general del sistema estudiado se basa en la manipulación por

parte del usuario a través de una interfaz de LabView donde es posible visualizar y modificar los parámetros del sistema, como lo son las posiciones deseadas para los motores y las ganancias de ajuste de los controles usados en cada eje del mecanismo.

#### Metodología

Con la finalidad de obtener señales lo más limpias posibles en los sistemas mecatrónicos, se utilizan diferentes tipos de acondicionamiento de dichas señales, uno de estos son el filtrado definiéndose como la eliminación de frecuencias que se presentan en los sistemas como ruido no deseable. Existen diferentes tipos de filtros dependiendo de la frecuencia que se desea eliminar. Los filtros digitales se diseñan dentro del sistema embebido de tal manera que se obtienen las señales ya sin ruido permitiendo obtener mejora en la lectura de los datos, se ha comprobado que su aplicación ayuda a disminuir los errores en las respuestas de los sistemas controlados. Para lograr tener las señales limpias se corrige el efecto de muestreo en la posición, implementado un filtro tipo Bessel dentro del microprocesador, después se elimina el error de cuantificación en la velocidad por medio del método MT, implementado en el firmware del sistema embebido. Con esta metodología las señales que obtenemos del movimiento del robot son lo más limpias posibles.

#### Filtro digital

Para poder obtener los valores reales se debe aplicar un filtro que elimine la repetición de espectros que se genera al digitalizar una señal, este proceso se realiza con frecuencia de forma externa empleando un filtro analógico, pero si requerimos los datos reales en procesos internos del FPGA se requiere emplear un filtro digital. La frecuencia sobre la cual se debe desarrollar el filtro Bessel es con una frecuencia de corte de 20Hz. Este filtro está instrumentado dentro del microprocesador y tiene la siguiente estructura:

$$H(s) = \frac{1}{(1 + 0.07071s + 0.0025s^2)} \quad (8)$$

La función de transferencia de la ecuación 8 ha sido transformada a tiempo discreto tal como se muestra en [5]. Posteriormente se obtiene una ecuación en función del muestreo de la señal y después haciendo uso del lenguaje D se instrumenta directamente en el sistema embebido.

#### Corrección de error de cuantificación empleando el método MT

La estimación de la velocidad es realizada con diversos métodos como la resta de la posición actual menos la anterior dividida entre el tiempo de muestreo:  $\dot{x}_i \approx (x_i - x_{i-1})/h$ . Este método es conocido como M y se considera que el tiempo de muestreo  $h$  es fijo. Sin embargo, debido a que el encoder cuantifica la posición, este método no estima adecuadamente la velocidad. En la Figura 3 se observa una señal de posición que ha sido cuantificada. Utilizando el método M, se observa que la velocidad obtenida consiste un tren de impulsos de distinta amplitud, tal y como se muestra en la Figura 3. El método MT reportado en [6] considera el tiempo que transcurre entre cada cambio de valor en la señal de posición. De este modo, se obtiene una señal estimada de velocidad que no contiene los impulsos de la Figura 3. El módulo de firmware instrumentado en el FPGA para estimar la velocidad con este método consiste en, un detector de flancos de la señal de posición del encoder, un contador y un detector de signo para la velocidad: el contador determina el tiempo transcurrido entre cada cambio del encoder y el signo determina el sentido de la velocidad. La señal del contador es enviada al procesador embebido en el FPGA para implementar el método MT.

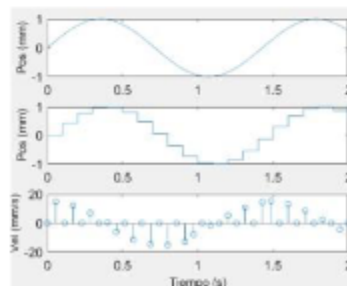


Figura 3. Estimación de velocidad con el método M.

**Control de posición y velocidad.**

Es necesario mencionar que entre los controles existentes más usados y que son capaces de llenar los requerimientos necesarios para su uso de manera industrial el más conocido es el algoritmo de control PD clásico y algunas de sus variaciones, el cual constituye una buena base para el diseño y mejora de un modelo de control adecuado. Una diferencia fundamental entre un control de posición y un control de trayectoria o de movimiento, es que este último incluye la dinámica completa del robot manipulador en la estructura matemática del controlador, es decir, se basa en el modelo dinámico del robot. Es por esto, que el diseño y realización de uno de estos controladores es mucho más complejo y es por esto también por lo que la nueva ley de control adquiere una exactitud, desempeño y robustez que dependen del grado de precisión con que se conozcan los parámetros dinámicos que se involucran en la descripción matemática del modelo dinámico del robot. Es importante notar, que mientras del control de posición depende de la posición del motor (encoder), la propuesta de los términos de fricción depende de la velocidad lineal de la mesa de trabajo, por lo que se controla la posición final del tornillo embalado (la cual se da en mm) que permitirá tener la posición y velocidad de la mesa de trabajo y basar el control de posición en dicha información, con lo que el torque inyectado en el sistema será una función que depende de la posición y velocidad de dicha variable. Por este motivo el control que se tiene implementado se basa en la posición deseada de la mesa de trabajo, dada en mm, para lo cual el programa y el firmware del robot realizan la conversión entre la posición del motor y la posición de la parte final, por lo que es posible adoptar dicha propuesta de control. Con esto, nuestra función de torque propuesto para el control de posición a implementar quedaría como:

$$\tau_i = K_{p_i}(q_{d_i} - q_i) + K_{v_i}(\dot{q}_{d_i} - \dot{q}_i) + J_i \ddot{q}_{d_i} + \tau_{c_f_i} \tag{9}$$

$$\tau_{c_f_i} = \frac{1}{2} [(1 + \text{signo}(\dot{q}_{d_i})) [k_{p_i} \tanh(\lambda_i \dot{q}_{d_i}) + b_{p_i} \dot{q}_{d_i}] - [\text{signo}(\dot{q}_{d_i}) - 1] [k_{n_i} \tanh(\lambda_i \dot{q}_{d_i}) + b_{n_i} \dot{q}_{d_i}]] \tag{10}$$

donde  $\tau_i$  es el torque de control,  $K_{p_i}$  y  $K_{v_i}$  son las ganancias proporcional y derivativa,  $q_{d_i}$  es la posición deseada,  $J_i$  es el momento total de inercia,  $k_{p_i}$  y  $k_{n_i}$  son los coeficientes de la fricción de Coulomb,  $b_{p_i}$  y  $b_{n_i}$  son los coeficientes de fricción viscosa,  $\lambda_i$  es el coeficiente de acondicionamiento de la fricción de Coulomb: el subíndice n de negativo y p de positivo de los coeficientes  $k_{p_i}$ ,  $k_{n_i}$ ,  $b_{p_i}$  y  $b_{n_i}$  indican el sentido de la velocidad donde son usados. Con esto, se tiene que el torque total que se provee por medio del funcionamiento del PWM está regulado por las compensaciones de fricción correspondientes a la función de fricción de coulomb, la función correspondiente a la fricción viscosa de la mesa de trabajo. La estabilidad de estas leyes de control se demuestra que son asintóticamente estables [7].

**Resultados**

A continuación en la Figura 4 se encuentran los resultados de la simulación del modelo dinámico obtenido con la metodología de parámetros agrupados correspondientes a el valor de la posición y la velocidad, realizando un movimiento senoidal de con amplitud de 45 mm con centro en 90mm: Se ha comparado las señales experimentales con las de simulación. Podemos observar que hay en la posición un 97% de coincidencia entre la simulación y los datos obtenidos experimentalmente y en la velocidad se obtuvo un 98%.

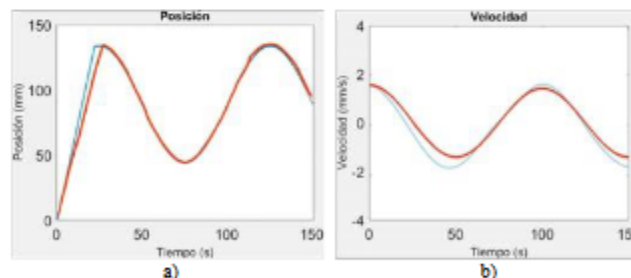


Figura 4.-a) Posición experimental (roja), posición simulación (azul) y b) Velocidad experimental (roja), Velocidad de simulación (Azul)

En la Figura 5 se compara la posición sin el proceso de filtrado y la posición con el filtro implementado, se puede observar en los recuadros deformaciones causadas por el efecto de muestreo, con la implementación en el

sistema embebido del filtro tipo Bessel dentro del microprocesador se limpia la señal como se observa en la Figura 5-b:

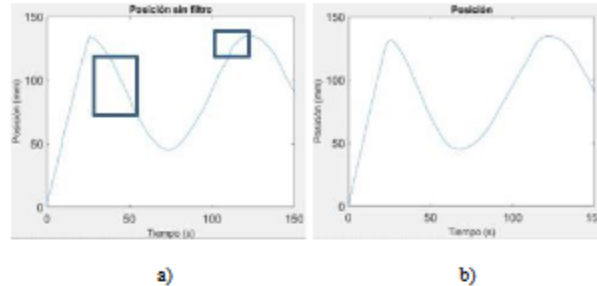


Figura 5.- Posición obtenida: a) Posición sin filtrado, b) Posición con proceso de filtrado.

En la figura 6 se compara la velocidad obtenida sin el método MT, en donde se observa el efecto de la información obtenida con el encoder del sistema con la generada aplicando la metodología propuesta, es decir con el método MT instrumentado en el microprocesador, se puede ver una clara diferencia entre el resultado de ambas gráficas, teniendo una señal al final limpia.

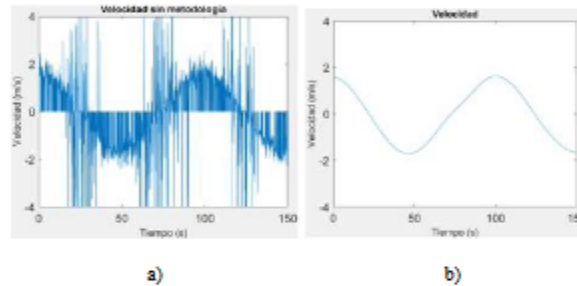


Figura 6.- Velocidad obtenida: a) velocidad obtenida sin método MT ni proceso de filtrado, b) velocidad obtenida tras aplicar método MT y filtrado.

### Conclusiones

Los resultados de las simulaciones del modelo dinámico obtenido con la metodología de parámetros agrupados realizando la simulación en Simulink de Matlab, y los resultados experimentales presentaron una correlación del 97% en la posición y un 98% en la velocidad, de acuerdo con la masa que tienen que mover estos dos ejes se considera una buena aproximación. Al instrumentar dentro del sistema embebido el filtro tipo Bessel para poder suavizar la señal que se obtiene de la posición tiene un buen desempeño ya que se elimina satisfactoriamente el efecto de muestreo obteniendo una señal mucho más limpia. La implementación en firmware del método MT para eliminar los errores de cuantificación cuando obtenemos la velocidad a partir de la posición del robot fue muy satisfactoria pues se logró tener una señal de la velocidad limpia y con esto tener una mejor respuesta.

### Referencias

1. B. Fabien. Analytical System Dynamics: Modeling and Simulation. Springer US, 2008.
2. Vargas-Treviño, M.A., López-Gómez, J., et al., "A mechatronic approach for ball screw drive system: modeling, control, and validation on an FPGA based architecture", *Int. J. Adv. Manuf. Technol.*, vol. 104, pp. 2329- 2346, Jul 2019.
3. G. Zapeda Arca, "Diseño de un microprocesador de arquitectura RISC de 32 bits", tesis de maestría, *Maestría en Ciencias de la Electrónica opción Automatización*, Benemérita Universidad Autónoma de Puebla, Puebla, 2017.
4. D. M. Gonzalez Arriaga, M.A.D. Vargas Treviño, S. Vargara Limón "Lenguaje D" *Programas de computación*, Registro Publico del Derecho de Autor, numero de registro: 03-2021-120110454200-01, 2021
5. Manual de usuario, "DEO-CV User Manual", [En línea]. Disponible en: [https://www2.pes.usp.br/~labdig/material/DEO\\_CV\\_User\\_Manual.pdf](https://www2.pes.usp.br/~labdig/material/DEO_CV_User_Manual.pdf). [Accedido: octubre 2021].

6. Se-Han Lee & Jae-Bok Song. (2001); "Acceleration Estimator for Low-Velocity and Low-Acceleration Regions Based on Encoder Position Data", *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, VOL. 6, NO. 1.
7. R. Kelly, V. Santibáñez, *Control de Movimiento de Robots Manipuladores*, Madrid: Pearson Education, 2003.

## Apéndice 2: Constancia de participación de congreso

Se realizó una ponencia en el Congreso Internacional de Investigación de Academia Journals Puebla TecNM 2023.

**PROYECTOS Y SERVICIOS ESTRATÉGICOS PARA INVESTIGACIÓN Y POSGRADO**  
**PYSEIP**  
CENTRO DE INVESTIGACIÓN

**GOBIERNO DE MÉXICO**

**EDUCACIÓN**

**TECNOLOGÍA NACIONAL DE MÉXICO**

**ACADEMIA JOURNALS**  
REVISTA DE INVESTIGACIÓN Y POSGRADO

Congreso Internacional de Investigación de Academia Journals Puebla TecNM 2023  
Ingeniería para la Innovación y Desarrollo Tecnológico Sostenible

# CERTIFICADO

otorgado a

**Lic. Luis Arturo Alcantar Vergara**  
**Dra. María Aurora Diozcora Vargas Treviño**  
**Dr. Sergio Vergara Limón**  
**Dr. Jesús López Gómez**  
**Dra. Amparo Palomino Merino**  
**Dra. Olga Guadalupe Félix Beltrán**  
**MC. Margarita Carmina García López**

por su artículo titulado

**Obtención y Filtrado Digital de la Posición y Velocidad de un Robot Cartesiano de Dos Grados de Libertad por Medio de un Sistema Embebido en Plataforma FPGA**

(Artículo No. ITP126)

La ponencia correspondiente fue presentada en el congreso desarrollado los días 22, 23, y 24 de febrero de 2023, teniendo como sede las magníficas instalaciones del Instituto Tecnológico de Puebla, institución líder en el sistema TECNM. El artículo está incluido en las siguientes publicaciones: (1) en el portal de Internet AcademiaJournals.com, con ISSN 1946-5351 online, Vol. 15, No. 01, 2023 e indexación en la base de datos [Fuente Académica Plus de EBSCOHOST](#), Massachusetts, Estados Unidos y (2) en libros ebook digitales compilados por área temática, con números ISBN online\*. Se tiene acceso a todas las publicaciones del congreso en el portal de internet de Academia Journals, con acceso libre.

Los organizadores del congreso reconocen la participación de los autores en el congreso, agradeciendo sus contribuciones.

**YEYETZIN SANDOVAL GONZÁLEZ**  
Directora  
Instituto Tecnológico de Puebla  
Tecnológico Nacional de México

**DR. RAFAEL MORAS, P.E.**  
Director General  
Academia Journals

**DR. PEDRO LÓPEZ EIROÁ**  
Director General  
PYSEIP- Productos y Servicios Estratégicos  
para Investigación y Posgrado

\* Las Ciencias Administrativas para la Innovación y Desarrollo Tecnológico Sostenible 978-1-939982-01-9  
Las Ciencias de la Salud para la Innovación y Desarrollo Tecnológico Sostenible 978-1-939982-02-6  
Matemáticas y Ciencias Exactas para la Innovación y Desarrollo Tecnológico Sostenible 978-1-939982-03-3  
Ciencias de la Educación para la Innovación y Desarrollo Tecnológico Sostenible 978-1-939982-04-0  
Humanidades, Ciencias Sociales y Bellas Artes para la Innovación y Desarrollo Tecnológico Sostenible 978-1-939982-06-4  
Ingeniería para la Innovación y Desarrollo Tecnológico Sostenible 978-1-939982-04-1

Congreso Academia Journals Puebla TecNM 2023 <https://www.academiajournals.com/pueblatecnm>