

**UNIVERSIDAD PRIVADA ANTENOR  
ORREGO**

**FACULTAD DE INGENIERÍA**

**ESCUELA PROFESIONAL DE INGENIERÍA  
ELECTRÓNICA**



---

**“DESARROLLO DE UN CONTROLADOR PARA  
REDUCIR EL ERROR EN LAS TRAYECTORIAS  
LINEALES DE UN ROBOT CARTESIANO TIPO  
GANTRY”**

---

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO**

**LÍNEA DE INVESTIGACIÓN: APLICACIONES ROBÓTICAS.**

**AUTOR:**

Br. Carlos Alberto Jara Ayala.

Br. Renato Del Castillo Huaccha.

**ASESOR:**

Ing. Saúl Linares Vértiz

**TRUJILLO – PERÚ**

**2019**

## ACREDITACIONES

### TÍTULO

“DESARROLLO DE UN CONTROLADOR PARA REDUCIR EL ERROR EN LAS TRAYECTORIAS LINALES DE UN ROBOT CARTESIANO TIPO GANTRY”

Elaborado Por:

---

Br. Carlos Alberto Jara Ayala

---

Br. Renato Del Castillo Huaccha

Aprobado Por:

---

Ing. Jorge Luis Alva Alarcón  
PRESIDENTE  
N° CIP 214251

---

Ing. Luis Alberto Vargas Díaz  
SECRETARIO  
N° CIP 104175

---

Ing. Ovidio Hildebrando Ramos Rojas  
VOCAL  
N° CIP 92622

---

Ing. Saul Noé Linares Vértiz  
ASESOR  
N° CIP 142213

## PRESENTACIÓN

Señores miembros del Jurado:

De conformidad y en cumplimiento de los requisitos estipulados en el Reglamento de Grados y Títulos de la Universidad Privada Antenor Orrego y el Reglamento Interno de la Carrera Profesional de Ingeniería Electrónica para obtener el Título Profesional de Ingeniero Electrónico, ponemos a vuestra disposición el presente Trabajo de Investigación Titulado: **“DESARROLLO DE UN CONTROLADOR PARA REDUCIR EL ERROR EN LAS TRAYECTORIAS LINEALES DE UN ROBOT CARTESIANO TIPO GANTRY”**. Este trabajo, es el resultado de la aplicación de los conocimientos adquiridos en la formación profesional en la Universidad, excusándonos anticipadamente de los posibles errores involuntarios cometidos en su desarrollo.

Trujillo, 5 de junio del 2019

Br. Carlos Alberto Jara Ayala.

Br. Renato Del Castillo Huaccha.

## **DEDICATORIAS**

### **A Dios:**

Por guiarme en el buen camino durante todo este tiempo, por darme las fuerzas de seguir cada día en esta vida y darme la salud para disfrutar de todos los que me han rodeado y acompañado en este tiempo.

### **A mis padres y hermanas:**

Agradezco de manera infinita el apoyo brindado por todos mis familiares, sobre todo a mis padres que me motivaron el seguir día a día, confiaron plenamente en mis conocimientos y me cuidaron durante toda esta etapa de mi vida y sobretodo, la paciencia que han tenido para conmigo.

### **A ella:**

Por preocuparse siempre en mí, querer lo mejor para mi porvenir y entenderme en todo, gracias a ella que fue un apoyo incondicional y el ingrediente perfecto para lograr alcanzar esta dicha.

### **A mi asesor:**

Agradecerle por cada momento dedicado para aclarar cualquier duda que me surgiera en el desarrollo del presente trabajo y todo el apoyo brindado.

Carlos A. Jara Ayala

**A Dios:**

Por estar presente no solo en esta etapa tan importante de mi vida, sino en todo momento ofreciéndome lo mejor y buscando lo mejor para mi persona.

**A mis padres y hermano:**

Les agradezco la dedicación y paciencia con la que se preocupaban por mi avance y desarrollo de esta tesis, por ser los principales promotores de mis sueños, gracias por cada día confiar y creer en mí.

**A mi asesor:**

Gracias por haber sido mi asesor y haberme enseñado tan bien y por haberme permitido el desarrollo de esta tesis.

Renato del Castillo Huaccha

## **AGRADECIMIENTOS**

A Dios por este trabajo de tesis que ha sido una gran bendición en todo sentido, y no cesan nuestras ganas de decir que es gracias a él que esta meta está cumplida. Cada momento vivido durante estos años han sido simplemente únicos.

A nuestra Universidad Privada Antenor Orrego de Trujillo, por brindarnos todos los medios para formarnos en lo que hoy somos y a nuestros maestros de la facultad de Ingeniería Electrónica por sus conocimientos impartidos durante nuestra vida universitaria, que se ve culminada con la presentación de nuestra tesis.

Nuestro mayor agradecimiento a nuestro asesor Ing. Saúl Linares Vértiz, maestros que se permitieron darle una revisada a nuestro trabajo, y a todas aquellas personas que directa o indirectamente colaboraron para que se haga realidad este trabajo.

Infinitamente a nuestras familias, por contar con su apoyo en el desarrollo de este trabajo. El camino recorrido hasta este momento no ha sido fácil, pero gracias a su aporte, a su amor, a su apoyo, lo complicado de llegar a este punto y cumplir esta meta se ha notado menos

## **RESUMEN**

Este trabajo de investigación presenta el desarrollo de un controlador de seguimiento de trayectorias lineales para una articulación con la finalidad de reducir el error en las trayectorias lineales de un robot cartesiano tipo gantry, mediante una simulación para el movimiento coordinado de dos articulaciones lineales y pruebas de seguimiento de trayectorias que se realicen en un sistema de control de pruebas que comprende una articulación lineal conformado por un motor de corriente continua y un husillo. Para llevar a cabo el desarrollo del controlador, se considera abordar la elección del dispositivo programable a utilizar de acuerdo a los requerimientos de un robot cartesiano tipo gantry, una pesquisa de técnicas aplicables para la coordinación de articulaciones, y el modelamiento matemático de la articulación lineal del sistema de control. Se realizan diversas pruebas de seguimiento de trayectorias con un control en lazo abierto, así como con el controlador desarrollado, para finalmente validar la reducción del error en las trayectorias lineales de un robot cartesiano tipo gantry mediante gráficos comparativos.

## **ABSTRACT**

This research work presents the development of a linear trajectory tracking controller for an articulation in order to reduce the error in linear trajectories of a gantry cartesian robot by means of a simulation for the coordinated movement of two linear articulations and trajectory tracking tests that are performed on a test control system that includes a linear articulation composed of a direct current motor and a spindle. To carry out the development of the controller, it's considered to approach the choice of the programmable device to be used according to the requirements of a gantry cartesian robot, a search of applicable techniques for the coordination of articulations, and the mathematical modeling of the linear articulation of the control system. Several trajectory tracking tests are carried out with an open loop control as well as with the developed controller, to finally validate the reduction of the error in the linear trajectories of a gantry cartesian robot by means of comparative graphs.

## CONTENIDO

1. INTRODUCCIÓN .....	2
1.1. REALIDAD PROBLEMÁTICA.....	2
1.2. DELIMITACIÓN DEL PROBLEMA .....	4
1.3. CARACTERÍSTICAS Y ANÁLISIS DEL PROBLEMA .....	5
1.4. FORMULACIÓN DEL PROBLEMA .....	8
1.5. FORMULACIÓN DE LA HIPÓTESIS .....	8
1.6. OBJETIVOS DE LA INVESTIGACIÓN .....	9
1.6.1. General .....	9
1.6.2. Específicos .....	9
1.7. JUSTIFICACIÓN DE LA INVESTIGACIÓN: .....	9
1.7.1. Importancia de la investigación:.....	9
1.7.2. Viabilidad de la investigación: .....	10
1.8. LIMITACIONES DEL ESTUDIO:.....	10
2. MARCO TEÓRICO.....	12
2.1. ANTECEDENTES DE LA INVESTIGACIÓN .....	12
2.2. BASES TEÓRICAS.....	16
2.2.1. El robot industrial .....	16
2.2.1.1. Estructura mecánica del robot industrial .....	16
2.2.1.2. Tipos de articulaciones .....	17
2.2.1.3. Robot cartesiano tipo Gantry .....	17
2.2.2. Cinemática directa e inversa de robots .....	18
2.2.3. Generación de trayectorias .....	19
2.2.3.1. Trayectorias según el movimiento de las articulaciones .....	19
2.2.3.2. Perfil de velocidad trapezoidal.....	19
2.2.4. Control cinemático.....	21
2.2.5. Control en tiempo continuo.....	22
2.2.5.1. Sistema de primer orden.....	22
2.2.5.2. Cancelación de polos y ceros.....	23
2.2.6. Control en tiempo discreto .....	23
2.2.6.1. Control realimentado.....	24
2.2.6.2. Control PID discreto.....	25
2.2.6.3. Algoritmo de un controlador digital.....	25
2.2.7. Motor de corriente directa (DC) .....	26
2.2.8. Sensor de posición angular.....	28
2.2.8.1. Encoder incremental de cuadratura .....	28
2.2.8.2. Decodificación de un encoder.....	29
2.2.8.3. Decodificación en modo 4X .....	29
2.3. DEFINICIÓN DE TÉRMINOS .....	31
3. PROCEDIMIENTOS.....	33
3.1. ELECCIÓN DEL DISPOSITIVO PROGRAMABLE.....	33
3.1.1. Keil MDK v5 y Programación .....	35

3.1.2.	Timers .....	36
3.1.2.1.	Modo PWM .....	37
3.1.2.2.	Modo encoder .....	37
3.2.	SISTEMA DE CONTROL PARA SEGUIMIENTO DE TRAYECTORIAS LINEALES .....	39
3.2.1.	Características.....	40
3.2.2.	Velocidad del motor DC .....	43
3.2.3.	Lectura de la posición y velocidad lineal.....	43
3.2.4.	Temporizador con Systick.....	43
3.2.5.	Comportamiento de la velocidad lineal .....	44
3.2.6.	Manejo de órdenes .....	44
3.2.7.	Descripción de la toma de datos.....	45
3.3.	CONTROL EN LAZO ABIERTO.....	46
3.4.	TÉCNICAS PARA LA COORDINACIÓN DE ARTICULACIONES LINEALES.....	48
3.4.1.	Rasterización de trayectorias .....	48
3.4.2.	Cálculo cinemático .....	50
3.5.	DESARROLLO DEL CONTROLADOR DE SEGUIMIENTO DE TRAYECTORIAS LINEALES.....	51
3.5.1.	Requerimientos.....	53
3.5.2.	Identificación del modelo de la articulación lineal.....	53
3.5.3.	Control PID discreto .....	54
3.5.4.	Algoritmo Trapezoide – PID.....	60
3.5.5.	Seguimiento de trayectorias lineales.....	65
3.6.	SEGUIMIENTO DE TRAYECTORIAS LINEALES ISÓCRONAS.....	68
3.6.1.	Algoritmo .....	69
3.6.2.	Simulación.....	69
4.	RESULTADOS .....	73
5.	DISCUSIÓN DE RESULTADOS.....	80
6.	CONCLUSIONES .....	83
7.	RECOMENDACIONES .....	85
8.	REFERENCIAS BIBLIOGRÁFICAS.....	87
9.	ANEXOS.....	90

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Algoritmo de normalización de pasos en cada articulación del robot cartesiano.	8
<b>Figura 2.</b> Seis posibles articulaciones en un robot.	17
<b>Figura 3.</b> Estructura mecánica de un robot cartesiano.	18
<b>Figura 4.</b> Relación entre la cinemática directa e inversa.	18
<b>Figura 5.</b> Movimiento de contorno.	19
<b>Figura 6.</b> Simulación en Matlab de un perfil de velocidad trapezoidal.	20
<b>Figura 7.</b> Funcionamiento del control cinemático.	22
<b>Figura 8.</b> Respuesta en el tiempo de un sistema en primer orden.	23
<b>Figura 9.</b> Sistema de control en lazo cerrado y en tiempo discreto.	24
<b>Figura 10.</b> Motor de corriente directa.	26
<b>Figura 11.</b> Motor DC sin escobillas.	28
<b>Figura 12.</b> Encoders de cuadratura.	28
<b>Figura 13.</b> Resolución en la decodificación de encoders de cuadratura.	29
<b>Figura 14.</b> Decodificador de un encoder de cuadratura.	30
<b>Figura 15.</b> Decodificador de un encoder de cuadratura.	30
<b>Figura 16.</b> Tarjeta de evaluación STM32_F4XX v3.0.	34
<b>Figura 17.</b> ST-Link V2.	34
<b>Figura 18.</b> IDE $\mu$ Vision de Keil MDK v5.	35
<b>Figura 19.</b> Depurador de Keil MDK.	36
<b>Figura 20.</b> Operación del registro contador CNT de un Timer en modo encoder.	38
<b>Figura 21.</b> Diagrama bloques del sistema de control para pruebas de seguimiento de trayectorias lineales.	39
<b>Figura 22.</b> Motor DC NF5475E.	41
<b>Figura 23.</b> Caja reductora.	41
<b>Figura 24.</b> Módulo VNH2SP30.	42
<b>Figura 25.</b> Vernier digital sobre articulación lineal.	42
<b>Figura 26.</b> Velocidad máxima en articulación lineal.	44
<b>Figura 27.</b> Diagrama de flujo de la función de manejo de órdenes enviadas por computadora.	45
<b>Figura 28.</b> Ciclo de trabajo para una velocidad lineal aproximada de 1 mm/s.	46
<b>Figura 29.</b> Diagrama de flujo del manejo de la orden "sp=[coor1],[coor2]..", para el control en lazo abierto.	47
<b>Figura 30.</b> Diagrama de flujo del bucle principal del control en lazo abierto.	48
<b>Figura 31.</b> Rasterización de trayectorias lineales.	49
<b>Figura 32.</b> Representación del algoritmo de Bresenham.	49
<b>Figura 33.</b> Diagrama de flujo del manejo de la orden "sp=[coor1],[coor2]..", para el control PID discreto y el algoritmo Trapezoide - PID.	51
<b>Figura 34.</b> Diagrama de flujo del bucle principal del control PID discreto y algoritmo Trapezoide - PID.	52
<b>Figura 35.</b> Identificación de la articulación lineal.	54
<b>Figura 36.</b> Lugares geométricos del bloque PID en cascada con la articulación lineal en lazo realimentado.	55

<b>Figura 37.</b> Simulación del seguimiento de una trayectoria de 1 $\mu$ m con control PID en tiempo continuo. ....	56
<b>Figura 38.</b> Diagrama de flujo de la interrupción por Timer 4 del control PID discreto. ....	57
<b>Figura 39.</b> Respuesta en el tiempo de la posición en el seguimiento de una trayectoria de 1 mm con el control PID discreto implementado. ....	58
<b>Figura 40.</b> Respuesta en el tiempo de la velocidad en el seguimiento de una trayectoria de 1 mm con el control PID discreto implementado. ....	59
<b>Figura 41.</b> Resultado del depurador en el seguimiento de una trayectoria de 1 mm con el control PID discreto implementado. ....	59
<b>Figura 42.</b> Respuesta en el tiempo de la posición en el seguimiento de trayectorias consecutivas según las coordenadas: 3 mm, 6 mm y 1 mm con el control PID discreto implementado. ....	60
<b>Figura 43.</b> Diagrama de flujo de interrupción por Timer 4 para el algoritmo Trapezoide - PID. ....	62
<b>Figura 44.</b> Respuesta en el tiempo de la posición en el seguimiento de una trayectoria de 10 mm con el algoritmo Trapezoide – PID implementado. ....	63
<b>Figura 45.</b> Respuesta en el tiempo de la velocidad en el seguimiento de una trayectoria de 10 mm con el algoritmo Trapezoide – PID implementado. ....	64
<b>Figura 46.</b> Resultado del depurador en el seguimiento de una trayectoria de 10 mm con el algoritmo Trapezoide – PID implementado. ....	64
<b>Figura 47.</b> Respuesta en el tiempo de la posición en el seguimiento de trayectorias consecutivas según las coordenadas: 3 mm, 6 mm y 1 mm con el algoritmo Trapezoide – PID implementado. ....	65
<b>Figura 48.</b> Respuesta en el tiempo de la velocidad para el seguimiento simulado de una trayectoria de 10 mm en el eje Y, y 5 mm en el eje X. ....	70
<b>Figura 49.</b> Resultados promedio del seguimiento de trayectorias lineales en lazo abierto y con el controlador desarrollado. ....	77
<b>Figura 50.</b> Resultados del E.R. Porcentual del seguimiento de trayectorias lineales en lazo abierto y con el controlador desarrollado. ....	77
<b>Figura 51.</b> Resultados resumidos de Fiestas (2017). ....	78
<b>Figura 52.</b> Respuesta en el tiempo de la posición en el seguimiento de una trayectoria de 10 mm en el eje Y, y 5 mm en el eje X. ....	78

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Pruebas de figuras geométricas dibujadas. ....	6
<b>Tabla 2.</b> Análisis del error en 10 líneas dibujadas de 2, 6 y 20 mm. ....	6
<b>Tabla 3.</b> Error porcentual de los períodos generados por el FPGA. ....	7
<b>Tabla 4.</b> Parámetros para el controlador con el algoritmo Trapezoide - PID.....	66
<b>Tabla 5.</b> Trayectorias lineales de 0.1 mm con el controlador desarrollado PID y Trapezoide - PID.....	66
<b>Tabla 6.</b> Trayectorias lineales de 1 mm con el controlador desarrollado PID y Trapezoide - PID. ....	67
<b>Tabla 7.</b> Trayectorias lineales de 10 mm con el controlador desarrollado PID y Trapezoide - PID.....	67
<b>Tabla 8.</b> Trayectorias lineales de 70 mm con el controlador desarrollado PID y Trapezoide - PID.....	68
<b>Tabla 9.</b> Trayectorias Lineales de 0.1 mm en lazo abierto.....	74
<b>Tabla 10.</b> Trayectorias lineales de 1 mm en lazo abierto. ....	74
<b>Tabla 11.</b> Trayectorias lineales de 10 mm en lazo abierto. ....	75
<b>Tabla 12.</b> Trayectorias lineales de 70 mm en lazo abierto. ....	75
<b>Tabla 13.</b> Resultados del seguimiento de trayectorias lineales con el controlador desarrollado Trapezoide - PID.....	76

# **CAPITULO I**

# 1. INTRODUCCIÓN

## 1.1. Realidad problemática

Dentro del área industrial, la robótica debe su posicionamiento al crecimiento estimado del 13% anual de unidades robóticas en el mundo desde el 2017 al 2019 según la IFR (Federación Internacional de la Robótica) en el 2017, y según lo que nos menciona Carreño (2014), los tipos de robots mayormente usados en la industria son los robots manipuladores por ser útiles para tareas repetitivas, de precisión y en actividades peligrosas para los seres humanos; ofreciendo ventajas como: reducción de los costos de producción, aumento de la productividad y calidad, y una mayor flexibilidad comparada con las máquinas especializadas.

Dentro del ámbito universitario la importancia de la robótica se incrementó por el hecho de contar con profesionales capaces de poder manipular esta tecnología actual, más aún si es una tecnología que está cambiando e innovando cada año y requiere de conocimientos tales como: electrónica, informática, mecánica, inteligencia artificial, entre otros; dándonos una idea de cuan multidisciplinaria puede ser la aplicación de la robótica. Esto junto al tema industrial, ha hecho que una de las áreas prioritarias de investigación que ha tomado fuerza en los últimos años sea el campo de la robótica; tal como se puede observar, por ejemplo, en la feria industrial de Hannover.

Al observar con atención el desarrollo de la robótica a nivel mundial, se puede comprender como ha evolucionado y se ha convertido en lo que se conoce ahora. Esto ha hecho que las mejores universidades en el mundo se adentren en impartir cursos o carreras de esta fascinante tecnología, pero a la vez se ha convertido en todo un reto para ellas, ya que no se puede competir en equipamiento e infraestructura con las principales potencias en esta área. Esto sumado a que se han abocado a la investigación robótica, siendo toda una competición entre ellas para generar nuevos conocimientos y ser reconocidas por las mismas, permite que la investigación sobre robótica abarque diferentes contextos no explorados, quizás en su totalidad, por los centros de investigación más grandes en donde su principal contexto de desarrollo es el industrial.

Por lo mencionado con anterioridad acerca de investigación robótica, así mismo resaltan ramas de estudio dentro de ella tales como: técnicas de control de

manipuladores, generación de trayectorias, planificación y seguimiento de trayectorias, sistemas robóticos autónomos, por nombrar algunas. En el área de control se tienen en cuenta varios aspectos para asegurar que el sistema cumpla con ciertas especificaciones de rendimiento, dentro de lo que sobresale el algoritmo y/o técnica de control a implementar según menciona Craig (2006).

- **Contexto**

Mirando la realidad del Perú y concretamente de nuestra región La Libertad, en el sentido de generar conocimiento a partir de la producción de artículos científicos e investigaciones, es donde se puede apreciar que no tenemos una producción literaria basta y extensa en investigación robótica de las áreas que sobresalen de ella, por lo que, al realizar una búsqueda a nivel local y nacional de investigaciones robóticas de pregrado, se hallan relativamente pocas.

Considerando lo dicho anteriormente, las investigaciones realizadas llevan consigo errores que se pueden constatar en sus resultados obtenidos debido a problemas diversos dentro del desarrollo de las mismas; resalta entonces una investigación relacionada a la robótica realizada por *Fiestas* (2017), el cual desarrolla una metodología para controlar los movimientos articulares de un robot cartesiano de tres grados de libertad utilizando la estructura mecánica de una máquina de control numérico por computadora: CNC JOYO 2020, un módulo FPGA De0\_NANO\_SoC de Altera, una PC y la creación de un controlador en lazo abierto.

Dicho trabajo de pregrado evitó fabricar una estructura robótica, puesto que se necesitan conocimientos y criterios de mecánica fina para diseñar las piezas de dicha estructura, y más bien optó por determinar si dicha estructura de la máquina CNC cumplía con las condiciones mecánicas más importantes para el trabajo que iba a desarrollar, por lo tanto, la estructura mecánica se semejó a la de un robot cartesiano tipo gantry. Otra consideración es que los torques de los motores paso a paso eran suficientes para poder mover la estructura robótica correspondiente, esto con el fin de no realizar un estudio más profundo sobre los actuadores y alargar el tiempo de ejecución de la investigación.

Al observar sus resultados y el desarrollo de la misma, se encuentran ciertos problemas tales como:

- Daños ocasionados en los drivers de potencia para los MPP (motores paso a paso). El trabajo en conjunto de las articulaciones para lograr un movimiento coordinado conlleva a sincronizar los tiempos de trabajo de los 3 motores paso a paso, pero si el algoritmo implementado no asegura esto, se corre el riesgo de sufrir ciertos problemas como este.
- Períodos no menores de 500  $\mu$ s generados en un módulo FPGA de Altera para la velocidad en los MPP y además velocidad limitada en la traslación del efector final para la realización de la tarea encomendada, teniendo en cuenta que los motores paso a paso ofrecen altas velocidades y que el módulo FPGA de Altera fue configurado para ofrecer velocidades de hasta 62.8 rad/s.
- Errores de posición altos en trayectorias lineales y figuras geométricas de la estructura robótica cartesiana tipo gantry.

## **1.2.Delimitación del problema**

El problema del presente proyecto de investigación se delimita a los errores existentes de funcionalidad en un robot cartesiano compuesto por una estructura CNC JOYO 2020, un módulo FPGA De0\_NANO\_SoC de Altera y una PC.

### **1.3. Características y análisis del problema**

1. Error en las trayectorias lineales realizadas por las articulaciones de un robot cartesiano tipo gantry.
2. Períodos no menores de 500 us generados en un módulo FPGA de Altera y Velocidad limitada en la traslación del efector final de un robot cartesiano tipo gantry.
3. Daños ocasionados en los drivers de los MPP de un robot cartesiano tipo gantry por discontinuidad en las trayectorias.

### **Análisis de las características problemáticas**

1. Evidenciando los resultados obtenidos por Fiestas (2017) con respecto a los trabajos realizados por el robot cartesiano tipo gantry, se tienen:
  - a) Errores del 12.57%, 18.56% y 11.42% en las áreas de las siguientes figuras geométricas dibujadas por el robot cartesiano: Rectángulo, triángulo y círculo respectivamente.
  - b) Errores absolutos de 0.50, 0.85 y 1.0 mm en las longitudes de líneas dibujadas.

Haciendo un análisis del punto **a)**, si bien cabe la posibilidad de que los errores presentados en la tabla 1 fueron generados por la existencia de error de posición en las articulaciones lineales controlados en lazo abierto, puede también deberse al algoritmo realizado para una de las funciones en la metodología del control desarrollado, que consiste en calcular las coordenadas cartesianas provenientes de una imagen cargada en la aplicación de Visual Basic para posteriormente transformarlas en variables articulares. Así mismo para el punto **b)**, la aplicación es directa por medio del ingreso de las coordenadas cartesianas a realizar por el robot, con resultados como se puede apreciar en la tabla 2.

Por lo tanto, se prescinde de una acción correctiva para reducir el error de posición existente en las articulaciones lineales de un robot cartesiano tipo gantry, que en este caso sería esencial en la metodología desarrollada por Fiestas (2017), para corroborar la eficiencia de los trabajos realizados por el

robot que se ordenen por la computadora, que constituirá también un aporte a la automatización basada en la robótica industrial.

**Tabla 1.** Pruebas de figuras geométricas dibujadas.

<b>Figura</b>	<b>Área Calculada (mm<sup>2</sup>)</b>	<b>Área Experimental (mm<sup>2</sup>)</b>	<b>error %</b>
Rectángulo	1539	1350	12.57234085
Triángulo	388.795	320	18.56251476
Círculo	849.3793	754.7676	11.41920787

*Fuente: Fiestas (2017)*

**Tabla 2.** Análisis del error en 10 líneas dibujadas de 2, 6 y 20 mm.

<b>i</b>	<b>xi = 2 mm</b>	<b>xi = 6 mm</b>	<b>xi = 20 mm</b>
<b>1</b>	2.5	5	18.5
<b>2</b>	3	5.5	19
<b>3</b>	3	5.5	19
<b>4</b>	3	6	19
<b>5</b>	3	5.5	19
<b>6</b>	3	5.5	19
<b>7</b>	3	5.5	19
<b>8</b>	3	5.5	19.5
<b>9</b>	2.5	5.5	19
<b>10</b>	2.5	5.5	19
<b>promedio</b>	2.85	5.5	19
<b>Error Relativo (%)</b>	42.5	8.33	5
<b>Error Absoluto (mm)</b>	0.85	0.5	1

*Fuente: Fiestas (2017).*

2. Otra de las limitaciones presentes en la metodología desarrollada por Fiestas (2017), son los pulsos de periodo no menor a 500 uS, generados por el diseño del circuito lógico en el módulo FPGA DE0\_NANO\_SOC de Altera como se puede observar en la tabla 3, la cual, según comunicación directa con el investigador, tiende a producirse una onda triangular a partir de ese periodo. Estos pulsos tienen un ciclo de trabajo del 50% y son necesarios para la generación de pasos, en modo full-stepping, de los MPP. Este periodo límite de 500 uS conlleva a generar una velocidad angular máxima para los MPP de 62.8 rad/s y así poder realizar los trabajos de dibujo con mayor rapidez. Sin embargo, de acuerdo al código en el procesador HPS (Hard

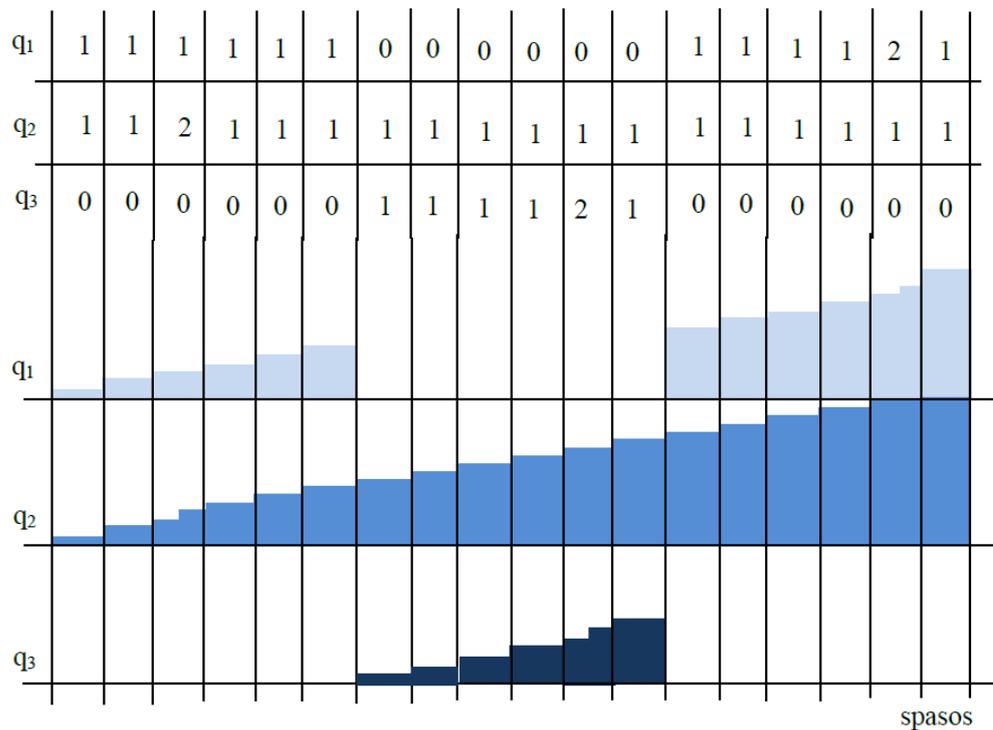
Processor System) del módulo FPGA, se limitó el funcionamiento a solo 16 rad/s como velocidad máxima.

**Tabla 3.** Error porcentual de los períodos generados por el FPGA.

Muestra	Período deseado (s)	Período medio (s)	e%
1	1	0.998	0.2
2	0.5	0.499	0.2
3	0.1	0.09956	0.44
4	0.01	0.01012	1.2
5	0.001	0.00101	1
6	0.0005	0.00052	4
7	0.0002	0.00024	20
8	0.00014	0.00016	14.29
9	0.00006	0.00014	133.3

*Fuente: Fiestas (2017).*

3. Según la metodología desarrollada por Fiestas (2017), se pretende una generación continua de las trayectorias en las articulaciones lineales de tornillo, sin embargo, esto no se evidencia en su algoritmo realizado y ejecutado haciendo uso del procesador HPS del módulo FPGA DE0\_NANO\_SOC de Altera, y más concretamente en la figura 1, donde para mantener la sincronidad de los pasos realizados se procede a generar pasos nulos en los MPP. Cabe resaltar que la generación de trayectorias está sujeto a seguir un perfil de velocidad trapezoidal, es decir con una rampa de subida y una de bajada, según el algoritmo de Austin (2004) para la generación de velocidades en los MPP. Por tanto, si la evolución de la trayectoria presenta pasos nulos (cuando el MPP se detiene) y sigue inicialmente una rampa de subida de velocidades (aceleración positiva), el MPP estará generando corrientes pico cada vez más altas para que venza el par de arranque a una velocidad cada vez mayor, por lo que cabe la posibilidad de quemar los drivers que suministran corriente a los MPP.



**Figura 1.** Algoritmo de normalización de pasos en cada articulación del robot cartesiano.

*Fuente: Fiestas (2017).*

#### 1.4. Formulación del problema

¿Cómo reducir el error en las trayectorias lineales realizadas por un robot cartesiano tipo gantry?

#### 1.5. Formulación de la hipótesis

El desarrollo de un controlador de seguimiento de trayectorias en una articulación lineal conformada por un motor DC y un husillo, reduce el error en las trayectorias lineales realizadas por un robot cartesiano tipo gantry.

## **1.6.Objetivos de la investigación**

### **1.6.1. General**

Reducir el error en las trayectorias lineales realizadas por un robot cartesiano tipo gantry.

### **1.6.2. Específicos**

- Describir brevemente las técnicas aplicables al movimiento coordinado de 2 articulaciones lineales conformados por un motor DC y un husillo.
- Desarrollar un controlador de seguimiento de trayectorias para una articulación lineal de acuerdo a las técnicas antes descritas.
- Realizar seguimientos de trayectorias lineales con un control en lazo abierto y con el controlador desarrollado.
- Validar la reducción del error en las trayectorias lineales realizadas.
- Simular una de las técnicas aplicables para el movimiento coordinado de 2 articulaciones lineales.

## **1.7.Justificación de la investigación:**

### **1.7.1. Importancia de la investigación:**

El estudiante universitario tendrá a la mano un trabajo con el cual podrá seguir una línea de investigación y/o realizar mejoras con respecto al área de sistemas de control o la robótica en general, esto permitirá que las nuevas generaciones de estudiantes se adentren en una tecnología que por el momento es poco aprovechable en el país, con miras a un desarrollo tecnológico en el área de mención, investigaciones de las áreas que abarca la misma o poder posicionarse en un puesto laboral mejor remunerado.

En el Perú, las PYME utilizan CNC o robots de bajo coste para su producción en el sector de calzado, productos metálicos y no metálicos; para lo cual adquieren máquinas con controles manuales, conllevando a errores de trazado o corte, por el error manual que presenta el manipular en los movimientos de la máquina y/o por las propias deficiencias del sistema. Con la presente investigación se trata de dar solución al control manual, control en lazo abierto, y poder brindar una alternativa

para generar movimientos precisos en trayectorias lineales de las articulaciones de un robot cartesiano.

### **1.7.2. Viabilidad de la investigación:**

- **Viabilidad técnica**

Se cuenta con suficiente acceso de información en la web, libros digitales y el apoyo de un asesor en el desarrollo de la investigación.

- **Viabilidad económica**

Se cuenta con los materiales necesarios al alcance de los investigadores provistos por los mismos y la universidad, de manera que, el proyecto no requiere financiamiento mayor o ser auspiciado por alguna entidad.

### **1.8.Limitaciones del estudio:**

Algunas de las limitaciones que se presentan en la investigación son las siguientes:

- ✓ Para el diseño del controlador se hará uso del modelo matemático, obtenido experimentalmente, de la articulación lineal conformada por el acople de un motor DC y husillo, el cual se tratará de simplificar sin considerar la fricción o torques necesarios del motor y considerando unas mediciones con el uso del software MATLAB para determinar el modelo final. Por lo mencionado anteriormente, se hará uso además, en lo posible, del modelo cinemático de la articulación lineal más no del modelo dinámico ya que conllevaría un estudio más profundo y detallado de la toda estructura.
- ✓ La estructura de la articulación lineal fue realizada por los bachilleres de la presente tesis, por lo que problemas mecánicos tales como: alineamiento y rozamientos, pueden incurrir en datos ligeramente alterados para obtener el modelo matemático y posteriormente en errores de posición en las pruebas de comprobación del controlador a desarrollar.

# **CAPITULO II**

## 2. MARCO TEÓRICO

### 2.1. Antecedentes de la investigación

Habiendo realizado una pesquisa bibliográfica en repositorios digitales, se destaca lo siguiente:

**Título:** “Análisis Comparativo del Desempeño de Técnicas de Control Conmutado Implementadas en Dispositivos Programables”

**Autor(es):** Fabio Andrés Moreno Castellanos, Diego Armando Centeno Ariza

**Institución:** Universidad Industrial de Santander

**Año:** 2013

El presente trabajo de investigación tuvo como objetivo realizar un estudio comparativo del desempeño de técnicas de control conmutado implementadas en dispositivos programables. Se utilizó una metodología de diseño de sistemas embebidos para implementar algoritmos de control usando técnicas de control conmutadas, sobre un dispositivo de arquitectura configurable, que en este caso fue un FPGA Xilinx Spartan 6 de referencia NEXYS-3, para controlar la velocidad de un motor de corriente continua. Así mismo se implementaron las mismas técnicas de control conmutadas para dicho proceso en un dispositivo de arquitectura configurada como en un Arduino MEGA 2560. Finalmente se comparó el desempeño de las técnicas control a partir de las diferentes alternativas en las que se implementaron.

**Aportes a la investigación:** Brinda información de un procedimiento para la identificación experimental del modelo matemático de un sistema físico, en ese caso el modelo de un motor de corriente continua NF5475E; además de un enfoque del desempeño de técnicas de control en dispositivos programables.

**Título: “Diseño de un Controlador PID, utilizando una Tarjeta FPGA cyclone V GX starter, Programada en VHDL”**

**Autor(es):** José Raúl Mejía Nuila, Guillermo Alfonso Maximiliano Narváez  
Henríquez

**Institución:** Universidad de el Salvador

**Año:** 2016

El presente trabajo de investigación tuvo como objetivo diseñar e implementar un sistema digital con un controlador proporcional, integral y derivativo (PID) digital en una tarjeta FPGA cyclone V GX Starter KIT de Altera mediante el lenguaje de programación VHDL, y aplicarlo al control de posición y velocidad de un motor de corriente continua. Desarrolla una metodología para diseñar un controlador PID para la posición y velocidad de un motor DC, en lazos de control independientes mediante el uso de un encoder de cuadratura de efecto Hall, y la programación del módulo FPGA en lenguaje VHDL para el diseño de bloques lógicos que hagan la tarea del controlador PID, la lectura de pulsos del encoder y otros bloques necesarios para alcanzar dicho propósito. Se utilizó métodos heurísticos para calcular las constantes del controlador PID y además programación con máquina de estados finito en VHDL para dividir el controlador PID digital en componentes de tal manera de mantener íntegro el componente PID para ser usado tanto en el control de posición y control de la velocidad.

Se obtuvieron resultados satisfactorios con respecto al control de posición y velocidad del motor de corriente continua y se logró cambiar la respuesta del sistema en el tiempo cambiando las constantes proporcional, integral y derivativa.

**Aportes a la investigación:** Brinda información en el diseño de una interfaz con compuertas lógicas haciendo uso de un FPGA de Altera, para leer los pulsos de un encoder de cuadratura para controlar la velocidad y posición angular de un motor de corriente continua. Dichos pulsos se leen con una resolución 4 veces por ciclo, haciendo además que para un sentido de giro del encoder un contador previamente diseñado se incrementa y para el otro sentido se decrementa.

**Título: “Control de los Movimiento de un Robot Industrial Utilizando un Módulo FPGA DE0\_NANO\_SOC de Altera, una PC y una Máquina CNC de Bajo Coste”**

**Autor:** Erick M. Fiestas Sorogastúa

**Institución:** Universidad Privada Antenor Orrego

**Año:** 2017

El presente trabajo de investigación tuvo como objetivo controlar los movimientos de un robot industrial. Desarrolla una metodología para controlar los movimientos articulares de un robot industrial de tres grados de libertad utilizando la estructura mecánica de una máquina CNC JOYO 2020, un módulo FPGA De0\_NANO\_SoC de Altera, una PC y la creación de un algoritmo de control en lazo abierto. El presente trabajo logró implementar una interfaz gráfica a través de una aplicación en visual studio, la cual al ser ejecutado en la PC da la posibilidad de enviar las coordenadas codificadas a seguir por el robot cartesiano; seguido de un procesador HPS perteneciente al módulo FPGA De0\_NANO\_SoC que se encarga de decodificar dichas coordenadas para luego generar variables articulares con un interpolador cúbico, seguido de la generación de un vector de de pulsos necesarios para los motores de paso; y finalmente un diseño de bloques de control en base a compuertas lógicas del FPGA en el mismo módulo para procesar dicho vector y generar los pulsos necesarios para el driver de dichos motores.

Se logró controlar el robot cartesiano sin embargo con errores en la precisión y algunas limitaciones en el tema de la velocidad del movimiento de las articulaciones.

**Aportes a la investigación:** Brinda la información más relevante a la hora de realizar esta investigación, ya que aporta los resultados de trayectorias lineales realizadas en un robot cartesiano tipo gantry construido a partir de una máquina CNC JOYO 2020, un módulo FPGA DE1\_SOC de Altera, una PC y la creación de un controlador un lazo abierto.

**Título: “Implementación de un Controlador Embebido para Controlar una Mesa XY”**

**Autor:** Diego Bustamante Varillas

**Institución:** Pontificia Universidad Católica del Perú

**Año:** 2013

El presente trabajo de investigación tuvo como objetivo principal desarrollar un controlador embebido basado en un microcontrolador ATmega128, que permita controlar la posición de una mesa XY.

Se realizaron pruebas en lazo abierto para determinar la relación desplazamiento - revolución haciendo distintas pruebas de distancias hechas con pulsos generados por un encoder para cada articulación de tornillo. En lazo cerrado, se logró optimizar un controlador PID obteniendo errores milimétricos de hasta 1.12% para el eje X y 1.35% para el eje Y. Además, se realizaron trayectorias lineales consecutivas con buenos resultados.

**Aportes a la investigación:** Brinda información acerca para llevar a cabo el desarrollo de los controladores para una articulación lineal conformada por un motor DC y un husillo, con el diseño de un perfil de velocidad trapezoidal para los motores y con la finalidad de generar trayectorias lineales consecutivas; además aporta una técnica para la coordinación de ejes.

**Título: “Control de una Fresadora CNC de uso Didáctico”**

**Autor:** Alejandro, Hernández Pérez

**Institución:** Instituto Politécnico Nacional

**Año:** 2014

El presente trabajo de investigación tuvo como objetivo proponer el diseño de control, para la rehabilitación de una fresadora de control numérico computarizado de uso didáctico. Realizó un análisis cinemático y dinámico de cada articulación de tornillo accionado por un motor DC de una fresadora CNC de 3 DOF para posicionar una herramienta, y posteriormente una interfaz gráfica en una PC para enviar datos de posición y velocidad como referencias a la fresadora.

**Aportes a la investigación:** Brinda información acerca de rasterización de trayectorias lineales y circulares, y el modelamiento matemático de una articulación

lineal conformado por un motor DC con caja reductora y un husillo. Aplica un control desacoplado a las articulaciones de una fresadora CNC haciendo uso de algoritmos de rasterización tales como el algoritmo de Bresenham y el algoritmo de punto medio, para sincronizar los movimientos de las tres articulaciones determinando el momento adecuado en el que se debe de mover cada una de ellas, para generar movimientos lineales y circulares respectivamente, y poder así seguir de manera autónoma la interpolación deseada que es plasmada a través códigos ISO G y M en una interfaz gráfica.

## **2.2. Bases teóricas**

### **2.2.1. El robot industrial**

Según la International Federation of Robotics (RIA) citado por Fiestas (2017), un robot industrial es definido por el estándar ISO 8373, el cual estipula lo siguiente: “Manipulador multifuncional reprogramable y controlado automáticamente que cuenta con tres o más ejes, los cuales pueden ser fijos o móviles y son orientados a aplicaciones de automatización industrial”.

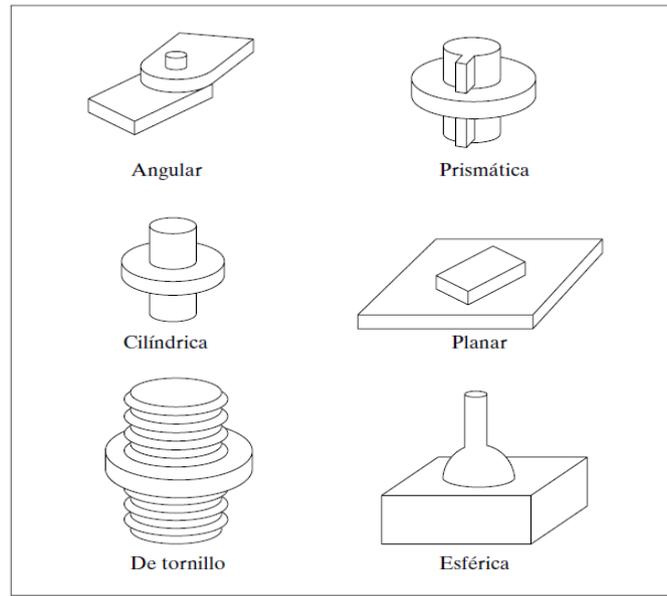
Así mismo Barrientos et. al., (2007) expone que la Asociación Francesa de Normalización (AFNOR) lo define de la siguiente manera: “Manipulador automático servocontrolado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectorias variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.”

#### **2.2.1.1. Estructura mecánica del robot industrial**

De acuerdo a Barrientos et. al. (2007), mecánicamente un robot está formado por una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre dos eslabones consecutivos.

### 2.2.1.2. Tipos de articulaciones

Los robots pueden tener varios movimientos ya sea de desplazamiento (Articulación prismática o lineal), de giro (Articulación rotacional) o una combinación de estas dos; sin embargo, Craig (2006), establece que existen seis posibles articulaciones como se muestran en la figura 2.



**Figura 2.** Seis posibles articulaciones en un robot.

*Fuente: Craig (2006).*

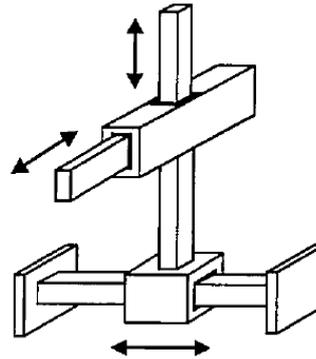
Por tanto, es posible inferir que las articulaciones de tipo tornillo, cilíndrica y prismática se encuentran dentro de las articulaciones lineales, y las articulaciones de tipo planar, angular y esférica, dentro de las articulaciones rotacionales.

### 2.2.1.3. Robot cartesiano tipo Gantry

De acuerdo Garibaldi (2009), el robot cartesiano tipo gantry es un robot con muchas aplicaciones en la industria cuyos ejes principales son prismáticos o lineales y forman entre ellos ángulos rectos. Además, el o los ejes horizontales se encuentran apoyados en ambos extremos.

Según Fiestas (2017), el movimiento del robot cartesiano en las direcciones de las coordenadas X, Y y Z del sistema de coordenadas cartesianas

rectangulares dextrógiras, se da con los movimientos: desplazamiento X, elevación Y, y alcance Z; y su espacio de trabajo tiene forma de una caja. Una de sus desventajas es que necesita un gran volumen para su operación, sin embargo, este robot cuenta con una estructura rígida y ofrece una posición precisa para el efector final.

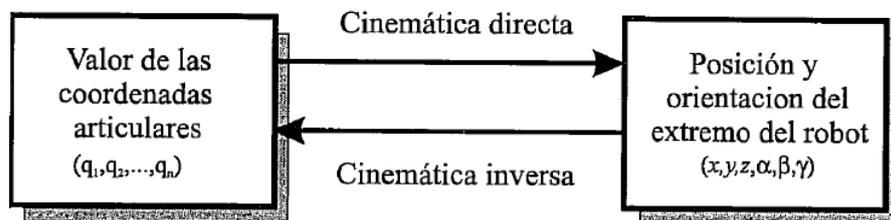


**Figura 3.** Estructura mecánica de un robot cartesiano.

*Fuente: Barrientos et al. (2007).*

### 2.2.2. Cinemática directa e inversa de robots

“La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia” (Barrientos et al., 2007, p. 93). Existen dos tipos de problemas cinemáticos: Directo e Inverso. A continuación, se muestran la relación que tienen ambos.



**Figura 4.** Relación entre la cinemática directa e inversa.

*Fuente: Barrientos et al. (2007).*

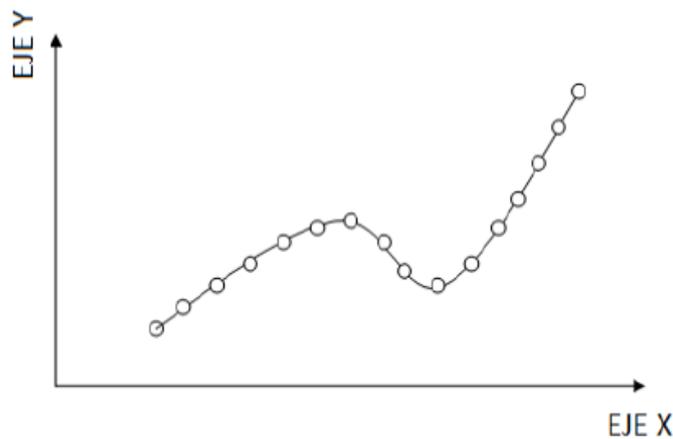
### 2.2.3. Generación de trayectorias

#### 2.2.3.1. Trayectorias según el movimiento de las articulaciones

De acuerdo a Fiestas (2017, p. 38) que cita a Barrientos et. al. (2007, p. 282):

Existen diversos tipos de trayectorias a generarse: Trayectorias punto a punto, que es cuando se establecen el punto de inicio y el final, y no interesa cómo es que evolucionan las variables articulares, y están a la vez conformadas por los movimientos eje a eje y el movimiento simultáneo en ejes; y trayectorias isócronas realizadas por movimientos coordinados en los ejes, que es cuando las articulaciones lineales empiezan y culminan al mismo tiempo. Así mismo están las trayectorias continuas, que es donde se tiene interés en los puntos que se recorra en una trayectoria preestablecida.

Según Bustamante (2013), las trayectorias continuas se llevan a cabo con un movimiento de contorno, donde se proporciona un historial de posiciones y se crea un camino suave o trazado a través de ellas tal como sucede con una interpolación de trayectorias.



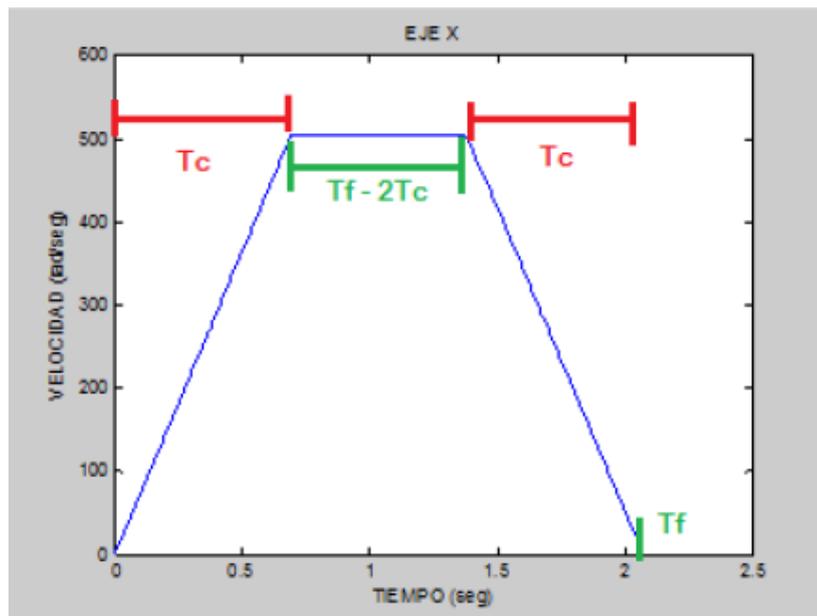
**Figura 5.** Movimiento de contorno.

*Fuente: Bustamante (2013).*

#### 2.2.3.2. Perfil de velocidad trapezoidal

De acuerdo a Bustamante (2013), en este perfil de velocidad el motor parte del reposo y acelera constantemente hasta alcanzar una velocidad máxima

establecida. El movimiento continúa a la velocidad constante deseada por un determinado periodo hasta que el controlador decide que es tiempo de comenzar una desaceleración constante disminuyendo la velocidad del movimiento para detenerse exactamente en la posición deseada. Así mismo es un perfil de velocidad adecuado para aplicaciones robóticas porque minimiza el desgaste mecánico de los componentes.



**Figura 6.** Simulación en Matlab de un perfil de velocidad trapezoidal.

*Fuente: Bustamante (2013).*

Como se puede ver en la figura 6, Bustamante (2013) presenta una simulación en Matlab del perfil de velocidad trapezoidal con la velocidad en radianes por segundo. La velocidad máxima es definida por el usuario y se establece en 500 rad/s, el tiempo crítico ( $T_c$ ) es el tiempo que le toma tanto a la aceleración como a la desaceleración y puede ser calculado de la siguiente manera:

$$T_c = \frac{v_{max}}{a}$$

El tiempo final ( $T_f$ ) es el tiempo total que debe seguir la trayectoria ( $x_f - x_i$ ) y es calculado en función de  $T_c$ .

$$x_f - x_i = v_{max}(T_f - 2T_c) + 2\left(\frac{T_c v_{max}}{2}\right)$$

$$T_f = T_c + \frac{\Delta x}{v_{max}}$$

Para aplicar este perfil de velocidad es necesario la siguiente condición principal:

$$T_f > 2 T_c$$

Lo que lleva a tener una aceleración en el siguiente intervalo:

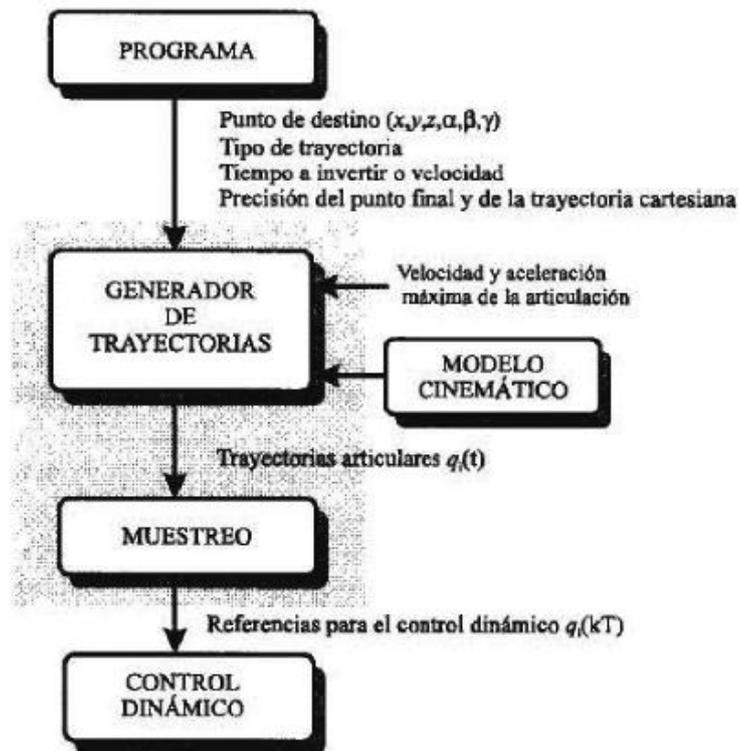
$$a > \frac{v_{max}^2}{\Delta x}$$

Entonces variando cualquiera de los parámetros en el intervalo anterior permite cumplir con la condición principal.

#### **2.2.4. Control cinemático**

Según Hernández (2014) que cita a Barrientos et al. (2007), el control cinemático establece cuales son las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo para lograr los objetivos fijados por el usuario (punto de destino, trayectoria cartesiana del efector final del robot, tiempo invertido por el usuario, etc.). Estas trayectorias se seleccionarán atendiendo las restricciones físicas propias de los accionamientos y a ciertos criterios de calidad de trayectoria, como suavidad o precisión de la misma.

La figura 7 muestra el funcionamiento del control cinemático. Se recibe como entradas los datos de movimiento procedentes del programa del robot escrito por el usuario (punto de destino, precisión, tipo de trayectoria deseada, velocidad o tiempo invertido, etc.) y, apoyándose en el modelo cinemático del robot, se establecen las trayectorias para cada articulación como funciones del tiempo. Estas trayectorias deben ser muestreadas con un período T a decidir, generando en cada instante kT un vector de referencias articulares para los algoritmos de control dinámico.



**Figura 7.** Funcionamiento del control cinemático.

*Fuente: Barrientos et al. (2007).*

## 2.2.5. Control en tiempo continuo

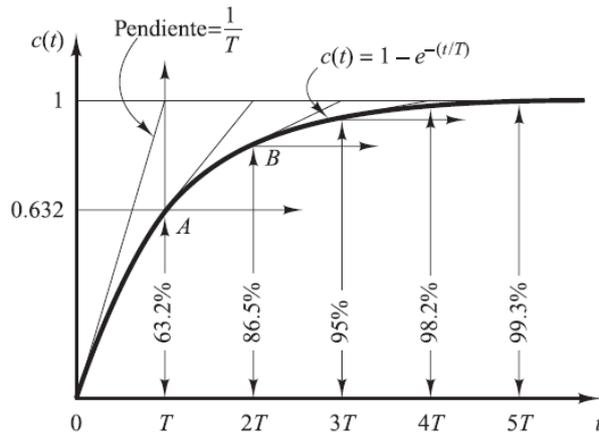
Para analizar un sistema de control en tiempo continuo es necesario primero obtener un modelo matemático del mismo, el cual se encuentra compuesto de un polinomio en el numerador, el cual sus raíces se denominan ceros, y un polinomio en el denominador, el cual sus raíces se denominan polos. Estos polinomios en sistemas de control en tiempo continuo están en función de la variable compleja  $s$  de Laplace.

### 2.2.5.1. Sistema de primer orden

De acuerdo a Ogata (2010, p. 161): “Un sistema de primer orden queda representado por la siguiente función de transferencia en el plano imaginario de Laplace”.

$$\frac{C(s)}{R(s)} = \frac{1}{Ts + 1}$$

El cual tiene un comportamiento como se muestra a continuación.



**Figura 8.** Respuesta en el tiempo de un sistema en primer orden.

*Fuente: Ogata (1996).*

### 2.2.5.2. Cancelación de polos y ceros

De acuerdo a Kuo (1996), en teoría de control automático, un sistema de control rige su comportamiento a la existencia de polos y ceros. Por cada polo existe un cero al que este sigue al variar los coeficientes de la función de transferencia. Cuando a la función de transferencia le falta un cero en el numerador, se entiende que este se encuentra en el infinito y el polo que le sigue puede variar su valor a lo largo de ese trayecto variando el comportamiento del sistema. Sin embargo, cuando el valor del cero es la misma que la del polo, el efecto de ambos sobre el sistema es nulo.

Para sistemas de orden superior es posible aplicar esto con la concatenación de otro bloque que contenga los ceros necesarios para la cancelación de polos, y de esta manera obtener un sistema de primer orden.

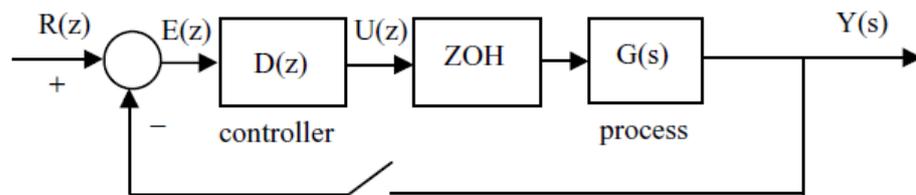
### 2.2.6. Control en tiempo discreto

Según Ogata (1996), hoy en día se ha incrementado el uso de controladores digitales en sistemas de control. Los controladores se utilizan para alcanzar el desempeño óptimo. El diseño e implementación de estos controladores está orientada a dispositivos programables y trabajan con señales digitales o señales de datos muestreados que en si vienen a formar parte de las señales en tiempo discreto.

Una de sus características más importantes es la elección de un periodo de muestreo para discretizar el comportamiento de un sistema en tiempo continuo. Esta elección está basada en el teorema de muestreo de Nyquist-Shannon, que menciona que la frecuencia de muestreo tiene que ser mayor igual a dos veces la frecuencia máxima del sistema.

### 2.2.6.1. Control realimentado

Según Bustamante (2013), con el control realimentado en un sistema de control, se obtiene una mejor respuesta en estado estable y transitorio, menor sensibilidad al ruido y disturbios o variaciones en la planta, y más exactitud en comparación con control en lazo abierto.



**Figura 9.** Sistema de control en lazo cerrado y en tiempo discreto.

*Fuente: Ibrahim (2006).*

Sin embargo, una de sus desventajas es que su diseño es más complejo y su desarrollo más costoso.

En la figura 9,  $D(z)$  es la función de transferencia de un controlador en función de la variable  $z$  que representa una secuencia de números o eventos en su dominio complejo.  $G(s)$  es la función de transferencia del proceso en función de la variable compleja  $s$  de Laplace. Y ZOH es un retenedor de orden cero en función de  $s$ , que según Ibrahim (2006), representa la aproximación de un conversor D/A (digital - analógico) en un sistema de control discreto realimentado, y es igual a:

$$ZOH = \frac{1 - e^{-Ts}}{s}$$

### 2.2.6.2. Control PID discreto

Ibrahim (2006), establece la función de transferencia de un controlador PID convencional, que relaciona una salida continua y la entrada continua del error de acuerdo a la figura 9, es:

$$\frac{U(s)}{E(s)} = K_p + \frac{K_p}{T_i s} + K_p T_d s$$

La forma discreta del controlador PID se puede obtener encontrando la transformada Z de la función continua del controlador en cascada con el retenedor de orden cero según la figura 9. El resultado de acuerdo a Ibrahim (2006) es la siguiente:

$$PID(z) = (1 - z^{-1})Z\left[\frac{U(s)}{s}\right]$$

$$PID(z) = PID(z)z^{-1} + b_0 * E(z) + b_1 * E(z)z^{-1} + b_2 * E(z)z^{-2}$$

Donde:

$$b_0 = k_p + k_d$$

$$b_1 = -(k_p + 2 * k_d - k_i * T_s)$$

$$b_2 = k_d$$

De las constantes anteriores  $T_s$  es el tiempo de muestreo y la función  $PID(z)$  puede ser implementada en una unidad computacional obteniendo su ecuación en diferencias.

### 2.2.6.3. Algoritmo de un controlador digital

Según Ibrahim (2006), la operación del algoritmo de un controlador digital, generalmente es resumida como una secuencia de simples pasos:

**Repetir por siempre lo siguiente:**

Cuando es momento de tomar la siguiente muestra

- ✓ Leer un valor deseado, R
- ✓ Leer la salida actual de la planta, Y
- ✓ Calcular la señal de error,  $E = R - Y$
- ✓ Calcular la salida del controlador, U
- ✓ Enviar la salida del controlador a un convertidor D/A

- ✓ Esperar por la siguiente muestra

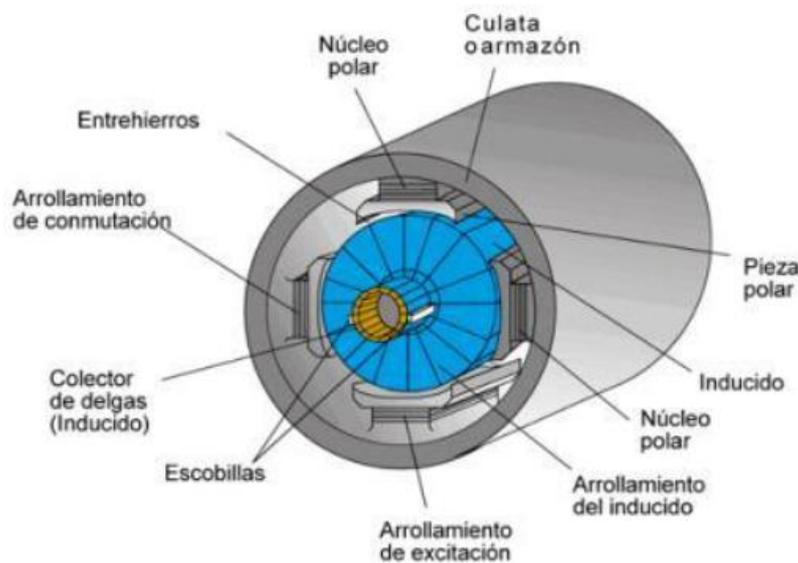
### Fin

Sin embargo, una de los detalles más importantes de los algoritmos, es que una vez que ellos han empezado ellos se ejecutan continuamente hasta que algunos eventos ocurran para detenerlos o hasta que ellos sean detenidos manualmente por un operador. Es importante asegurarse de que se ejecuten continuamente y exactamente en los instantes de muestreo. Esto es llamado sincronización y hay muchas maneras de alcanzarlo en práctica:

- ✓ Esperando en un ciclo finito el siguiente muestreo.
- ✓ Usando interrupciones externas.
- ✓ Usando interrupciones por temporizador.
- ✓ Usando un RTC (real-time clock).

### 2.2.7. Motor de corriente directa (DC)

De acuerdo a Hernández (2014), en la actualidad los motores de corriente continua directa son los más usados debido a su facilidad de control. Los motores DC están contruidos por dos devanados internos: Un inductor y un inducido.



**Figura 10.** Motor de corriente directa.

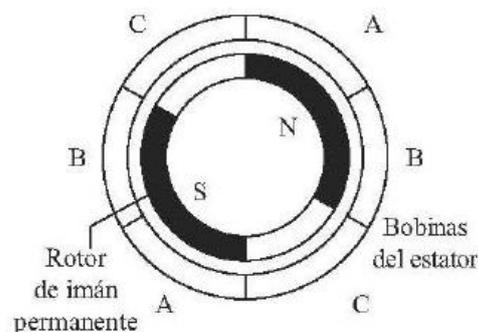
*Fuente: Hernández (2014)*

El inductor, que está situado en el estator (compuesto mayormente por el armazón y el arrollamiento de excitación como se puede observar en la figura 10, crea un campo magnético de excitación.

El inducido, situado en el rotor, hace girar al mismo gracias a la fuerza de Lorentz que surge por la combinación de la corriente circulante por él y el campo magnético de excitación antes mencionado. Recibe la corriente del exterior a través del colector de delgas, en el que se apoyan escobillas de grafito que sirven para la conmutación automática del sentido de la corriente en el arrollamiento del inducido al momento de girar el rotor.

La tensión en el inducido es proporcional a la velocidad angular del rotor. En el caso de controlar la velocidad por el inducido, la intensidad del inductor se mantiene constante. De acuerdo a Hernández (2014) que cita a Barrientos et al. (2007) este tipo de control por el inducido en los motores DC es el que se usa para el accionamiento de robots.

Para mejorar el comportamiento de los motores DC, el campo magnético de excitación se genera mediante imanes permanentes en el estator en vez de bobinas de campo. Hernández (2014) que cita a Barrientos et al. (2007), menciona que los motores DC presentan el inconveniente del constante mantenimiento a las escobillas, además que las conmutaciones de éstas originan chispas eléctricas que pueden ser perjudiciales en ambientes con riesgo de explosión. Así mismo existe una variación en base a los antes mencionado, donde se diseñaron motores sin escobillas (llamados comúnmente Brushless) que consisten básicamente en una secuencia de la bobina del estator y un rotor de imanes permanentes. En el motor DC de imanes permanentes sin escobillas los conductores de corriente están fijos y el imán en el rotor se mueve.



**Figura 11.** Motor DC sin escobillas.

*Fuente: Hernández (2014).*

### **2.2.8. Sensor de posición angular**

La opción más viable y adecuada para medir la posición angular de un motor DC es hacer uso de un encoder incremental de cuadratura que por lo general son ópticos.

El encoder óptico básicamente consiste en una fuente de luz y un disco unido a un eje rotatorio con secciones opacas y transparentes. Un sensor de luz al otro lado de la rueda detecta luz y por ende un pulso es producido cuando en la sección transparente del disco incide el haz de luz de la fuente. El controlador del encoder cuenta los pulsos en un tiempo dado, y esto es proporcional a la velocidad del eje (Ibrahim, 2006); y por tanto las cuentas, a la posición angular del eje.



**Figura 12.** Encoders de cuadratura.

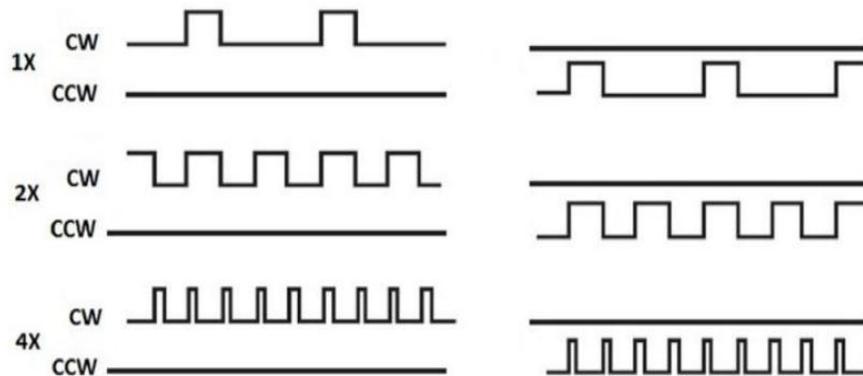
*Fuente: Ibrahim (2006).*

#### **2.2.8.1. Encoder incremental de cuadratura**

Según Mejía y Narváez (2016), se caracterizan por tener 2 canales de salida, llámese “canal A” y “canal B” desfasados entre sí 90 grados; por tanto, de esta manera los encoders serían útiles para determinar el sentido de giro de un motor ya sea por el adelanto o atraso del canal A al canal B.

### 2.2.8.2. Decodificación de un encoder

Como se había mencionado anteriormente, los encoders serían útiles para medir desplazamiento angular y la dirección de rotación y así mismo diferenciando el desplazamiento angular se puede obtener la velocidad angular, por lo que es necesario decodificarlo. De acuerdo a Mejía y Narváez (2016), esto se puede llevar a cabo usando circuitos lógicos secuenciales y menciona que se pueden obtener tres diferentes resoluciones: 1X, 2X y 4X. La resolución 1X proporciona un pulso de salida en cada flanco ya sea negativo o positivo de la señal A o B, lo que resulta en un solo pulso para cada ciclo o periodo de la señal de cuadratura. La resolución 2X ofrece un pulso de salida en cada flanco positivo y negativo bien de la señal A o la señal B, lo cual resulta en el doble de pulsos de salida. La resolución 4X ofrece un pulso de salida en cada flanco positivo y negativo de las señales A y B, dando lugar a 4 veces el número de pulsos de salida.

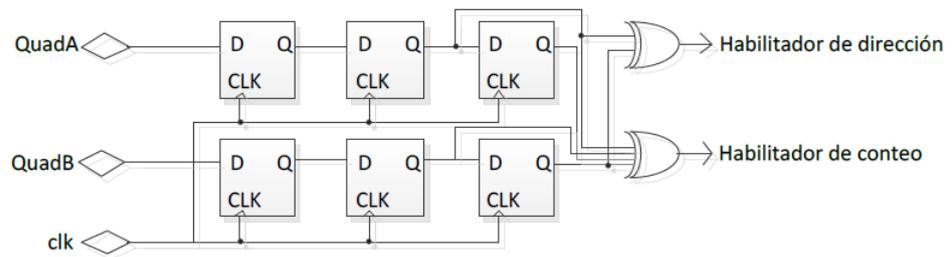


**Figura 13.** Resolución en la decodificación de encoders de cuadratura.

*Fuente:* Mejía y Narváez (2016).

### 2.2.8.3. Decodificación en modo 4X

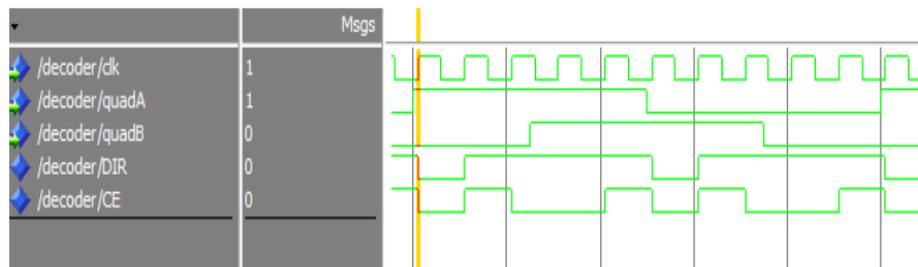
Según Mejía y Narváez (2016), Para realizar este tipo de decodificación se debe tener un reloj que sea mucho más rápido que las señales que se reciban en el encoder de cuadratura con el fin de obtener un muestreo eficiente.



**Figura 14.** Decodificador de un encoder de cuadratura.

*Fuente:* Mejía y Narváez (2016).

Con un enfoque en los circuitos digitales, se utiliza el circuito lógico mostrado en la figura 14, conformado por tres flip flops tipo D por canal y dos compuertas XOR. En sí un diseño inicial más práctico y simplificado consiste en un flip flop por canal, pero en la práctica las señales de cuadratura no son sincronicas con la señal de reloj “clk”, por lo que agregar dos flip flops adicionales, permite darle una mayor estabilidad al circuito.



**Figura 15.** Decodificador de un encoder de cuadratura.

*Fuente:* Mejía y Narváez (2016).

Como se puede observar en la figura 15, el funcionamiento del circuito de la figura 14 consiste en poner en uno lógico el Habilitador de Conteo (señal “CE”) 4 veces por cada ciclo ya sea del canal A o canal B. Por consiguiente, se puede establecer como señal de reloj la señal “CE” para un contador que se incremente en cada flanco de subida y de bajada de cada canal, que a la vez es proporcional al desplazamiento angular del encoder.

### 2.3. Definición de términos

**Interpolación:** Obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos.

**Robot:** manipulador automático servocontrolado, reprogramable, capaz de ejecutar tareas variadas como posicionar y orientar piezas siguiendo trayectorias variables. Normalmente tiene la forma de uno o varios brazos terminados en muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.

**Grados de Libertad (GDL):** son cada uno de los movimientos independientes que puede realizar cada articulación robótica con respecto a la anterior.

**Efecto final:** Es el actuador o también conocido como la herramienta que ejercerá el trabajo sobre un material.

**Algoritmo:** Es una secuencia de pasos lógicos necesarios para llevar a cabo una tarea específica, como la solución de un problema.

**Cinemática:** estudio de los movimientos de los cuerpos con posición y orientación en un espacio cartesiano prescindiendo de las fuerzas ejercidas.

# **CAPITULO III**

### **3. PROCEDIMIENTOS**

Para poder realizar una experimentación que nos permita reducir el error en las trayectorias lineales de un robot cartesiano tipo gantry es necesario:

1. Determinar un dispositivo programable adecuado para el desarrollo del controlador de seguimiento de trayectorias lineales.
2. Describir el sistema de control para el seguimiento de trayectorias lineales el cual gobernará dicho dispositivo programable para el desarrollo del controlador.
3. Desarrollar un control en lazo abierto para validar la reducción del error en las trayectorias lineales.
4. Describir las técnicas aplicables para la coordinación de articulaciones lineales compuestas de un motor DC y un husillo.
5. Desarrollar el controlador sobre el sistema de control de seguimiento de trayectorias lineales.
6. Simular el seguimiento de trayectorias lineales isócronas con una de las técnicas aplicables.

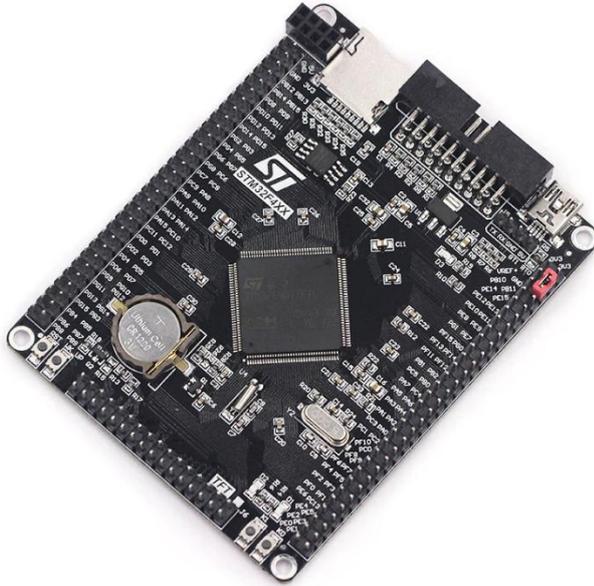
#### **3.1. Elección del dispositivo programable**

Se escogió el dispositivo programable STM32F407ZG, el cual según el sitio web de la empresa STMicroelectronics (<https://www.st.com>), es un microcontrolador basado en arquitectura ARM (Advanced RISC Machine) Cortex M4, que funciona a una frecuencia de reloj de hasta 168 MHz con FPU (Floating Point Unit), cuenta con 1 MByte de memoria Flash, 192 KBytes de memoria SRAM, y una diversidad de periféricos y/o controladores. En resumen, presenta los recursos que se muestran en el anexo 1.

La elección de este dispositivo programable es principalmente de acuerdo a lo siguiente:

- Según el manual de referencia de STMicroelectronics (2017), este microcontrolador tiene asignado a dos Timers de 32 bit (Timer 2 y Timer 5), el modo encoder de cuadratura con sus dos canales respectivos cada uno, por lo que sería conveniente utilizarlo para controlar el seguimiento de trayectorias lineales en un robot cartesiano tipo gantry.

- Disponibilidad de la tarjeta de evaluación STM32\_F4XX v3.0, el cual contiene al microcontrolador STM32F407ZG como se muestra en la figura 16.



**Figura 16.** Tarjeta de evaluación STM32\_F4XX v3.0.

*Fuente: Elaboración Propia.*

- Disponibilidad del ST-Link V2, dispositivo con funcionalidad SWD y JTAG necesario para usar el canal depurador ITM y grabar el código al microcontrolador.

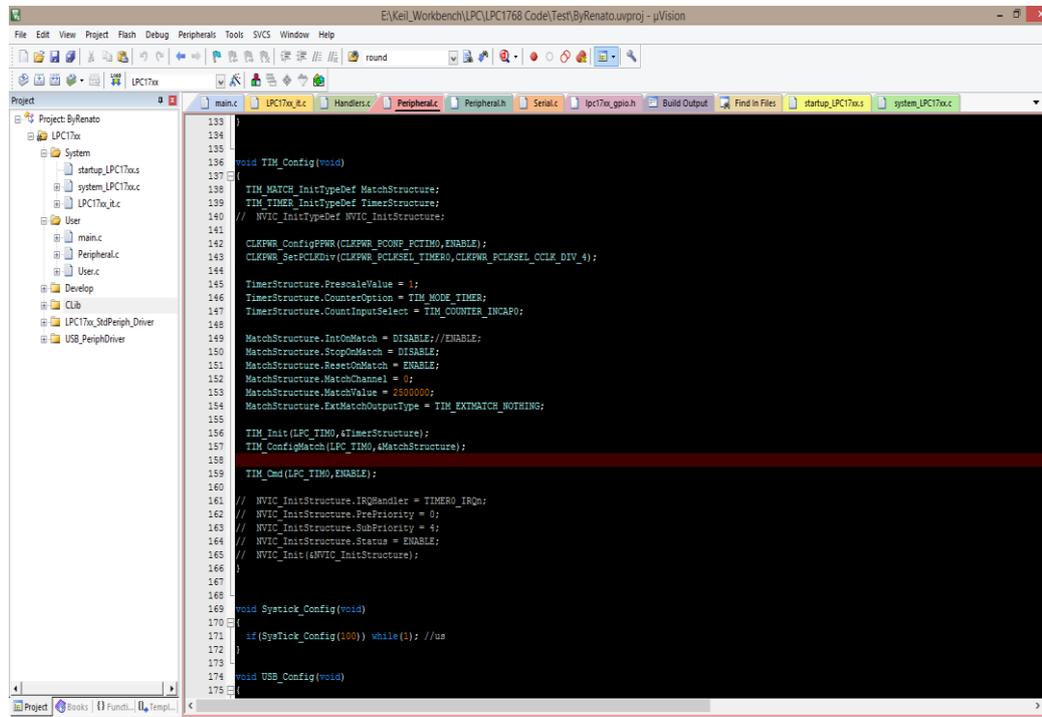


**Figura 17.** ST-Link V2.

*Fuente: Elaboración Propia.*

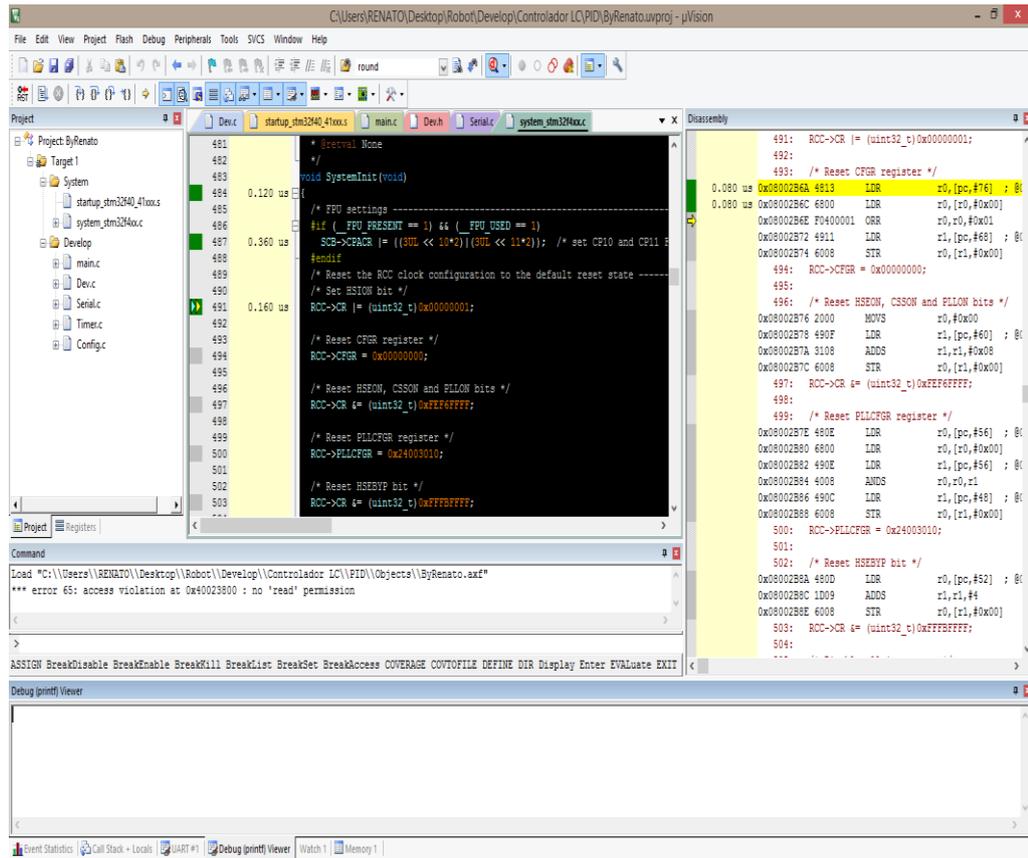
### 3.1.1. Keil MDK v5 y Programación

Según el sitio web de Keil (<http://www.keil.com>), herramientas de desarrollo integradas para ARM (Keil - MDK); es un entorno de desarrollo de software completo para una amplia gama de microcontroladores basados en ARM Cortex-M. Incluye el IDE  $\mu$ Vision (figura 18), un depurador (figura 19), compiladores ARM C / C ++ y componentes de middleware esenciales. Además provee de archivos usualmente usados en lenguaje C tales como math.h,stdio.h,etc; así como archivos de inicio necesarios para la ejecución de código en microcontroladores ST, hacer interface con el procesador, y la configuración de relojes del sistema para el funcionamiento de puertos y otros recursos.



**Figura 18.** IDE  $\mu$ Vision de Keil MDK v5.

*Fuente: Elaboración Propia.*



**Figura 19.** Depurador de Keil MDK.

*Fuente: Elaboración Propia.*

Como menciona Fiestas (2017), la programación con variables estructurales en lenguaje C, permite un fácil entendimiento del programa a realizar, así como un fácil seguimiento del algoritmo de control que se implementa. Las variables y tipos de variables estructurales a lo largo del desarrollo se pueden ver en los anexos 4, 5 y 6.

### 3.1.2. Timers

El microcontrolador STM32F407ZG cuenta con una serie de recursos asociados a los Timers, útiles para el desarrollo del controlador de seguimiento de trayectorias lineales. Naturalmente los Timers presentan registros que lo controlan y trabajan en función de una señal de reloj para realizar la temporización en un período T según la siguiente ecuación:

$$T = \frac{(ARR_y + 1)(PSC_y + 1)}{CK_{INT}} \dots \dots \dots (1)$$

Donde:

$CK_{INT}$ : Frecuencia de reloj en el timer.

$T$ : Período.

$ARR_y$ : Registro de autorecarga de timer  $y$ .

$PSC_y$ : Registro prescalador de timer  $y$ .

El registro contador CNT se incrementará hasta alcanzar el valor almacenado en el registro ARR. El reloj principal del sistema usualmente es configurado a la frecuencia máxima que es de 168 MHz, mientras que el reloj de los Timers generalmente es igual a la mitad de esta.

$$CK_{SYS} = 168 \text{ MHz} \qquad CK_{INT} = 84 \text{ MHz}$$

### 3.1.2.1. Modo PWM

En conjunto con la fórmula (1), este modo principalmente hace uso de uno de los registros comparador captura (CCR) del Timer, que en este caso es el CCR1, el cual hace variar el ciclo de trabajo de la señal al compararse con el registro CNT.

Es decir, la velocidad del motor puede ser controlada escribiendo al registro CCR1 según el siguiente intervalo:

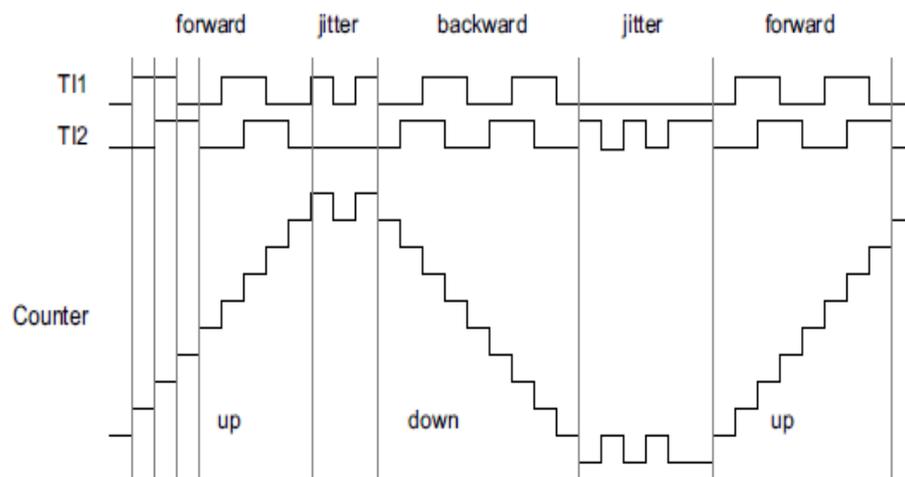
$$0 \leq CCR1_y \leq ARR_y$$

Donde:

$CCR1_y$ : Registro comparador captura 1 del timer  $y$ .

### 3.1.2.2. Modo encoder

En este modo es necesario configurar 2 pines necesarios del microcontrolador para poder leer un encoder de cuadratura, y consiste básicamente en un contador de pulsos UP/DOWN producido por un reloj externo o encoder, pudiéndose saber la posición y sentido de giro, así como también poder invertir la polaridad del contador en la configuración del Timer, y de esta manera poder contar pulsos de manera solo ascendente o descendente para ambos sentidos de giro.



**Figura 20.** Operación del registro contador CNT de un Timer en modo encoder.

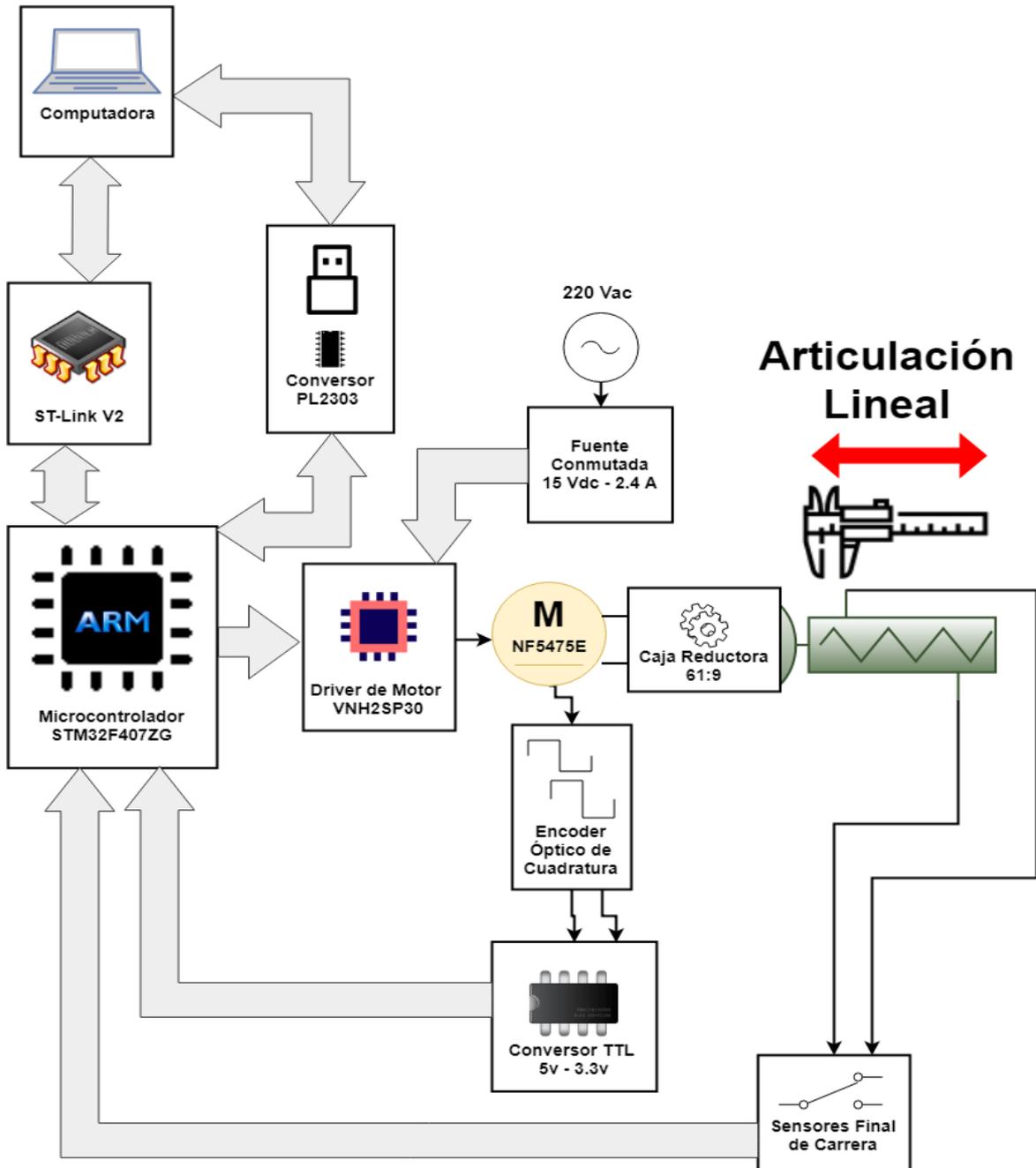
*Fuente: STMicroelectronics (2017).*

Según se muestra en la figura 20, es posible aumentar la resolución del encoder a una decodificación 4x puesto que es un encoder de cuadratura, así mismo es posible medir el período entre cuentas hechas con la adición de otro Timer según la disponibilidad. Por ejemplo, según el manual de referencia de STMicroelectronics (2017), es posible configurar el Timer 5 para dicho propósito haciendo que este sea reseteado y activado cada flanco de subida del pulso de un solo canal, y así obtener las cuentas hechas del Timer 5 en su mismo registro de captura CCR1. Es entonces que si se configura el Timer 5 con el registro  $ARR=0xFFFFFFFF$  y  $PSC = 0$ , es decir el registro CNT cuenta cada ciclo de reloj, y enlazado al Timer 2 en modo encoder, es posible determinar los pulsos por segundo (PPS) mediante la siguiente ecuación.

$$PPS = \frac{CK_{INT}}{CCR1_5}$$

### 3.2. Sistema de control para seguimiento de trayectorias lineales

En la figura 21 se muestra el diagrama de bloques del sistema de control como objeto para realizar pruebas de seguimiento de trayectorias lineales.



**Figura 21.** Diagrama bloques del sistema de control para pruebas de seguimiento de trayectorias lineales.

*Fuente: Elaboración Propia.*

### 3.2.1. Características

- ✓ La computadora se encuentra conectada con el microcontrolador mediante el puerto UART hacia un módulo conversor USB - TTL serial PL2303 para realizar adquisición de datos y alimentarlo, y también con el grabador ST-Link V2 que por medio del canal depurador ITM sirve para leer información de las pruebas que se realicen y enviar órdenes.
- ✓ La articulación lineal se encuentra conformado por el motor NF5475E (figura 22), una caja reductora (figura 23) de relación 61:9, un husillo de 13 anillos por pulgada y de 42 cm de largo aproximadamente, y un husillo guía que es el efector final (figura 25).

$$R = \frac{61}{9} \qquad D_{aa} = \frac{25400}{13} \text{ um}$$

Donde:

*R*: Relación de reducción de la caja reductora.

*D<sub>aa</sub>*: Distancia entre anillos de la articulación.

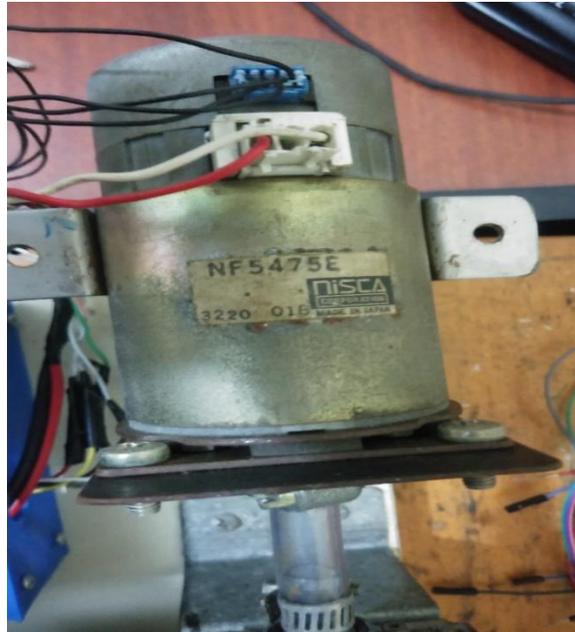
- ✓ El microcontrolador actúa sobre el módulo VNH2SP30 (figura 24), el cual es un driver adecuado para el funcionamiento del motor, con la posibilidad de controlar el sentido de giro (mediante INA e INB) y la velocidad de giro (PWM).
- ✓ Existe 2 sensores de final de carrera en ambos extremos de la articulación para limitar el desplazamiento del efector final.
- ✓ El driver para darle fuerza al motor está siendo alimentado por una fuente conmutada de 15 v.
- ✓ El microcontrolador se encuentra conectado al encoder óptico de 200 ppr (Pulsos por revolución), mediante un conversor de nivel lógico TTL de 5.0 v a 3.3v, para leer datos de posición y velocidad lineal en la articulación.

$$PPR_{1x} = 200 \text{ ppr}$$

Donde:

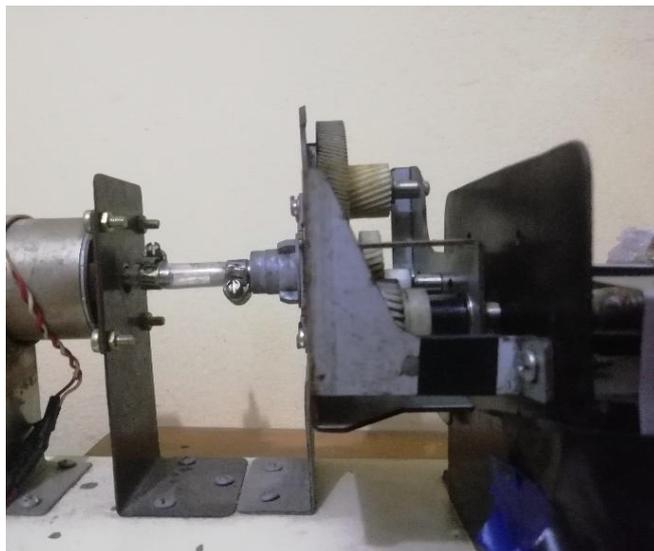
*PPR<sub>1x</sub>*: ppr del encoder óptico con decodificación 1x.

✓ Sobre la articulación lineal se encuentra acoplado un vernier digital (figura 25) de 10  $\mu\text{m}$  de incertidumbre y 15 cm de rango, de tal manera que sea movido por el efector final para corroborar las lecturas del encoder.



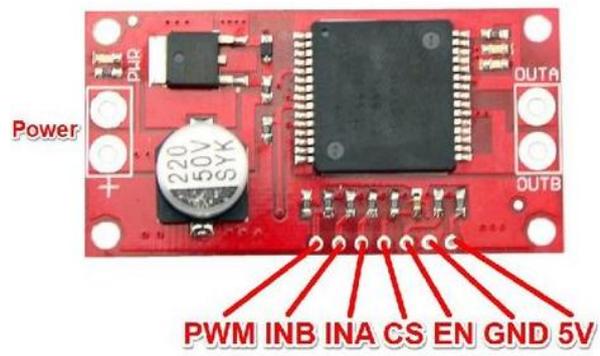
**Figura 22.** Motor DC NF5475E

*Fuente: Elaboración Propia.*



**Figura 23.** Caja reductora.

*Fuente: Elaboración Propia.*



**Figura 24.** Módulo VNH2SP30

*Fuente: Elaboración Propia.*



**Figura 25.** Vernier digital sobre articulación lineal.

*Fuente: Elaboración Propia.*

### 3.2.2. Velocidad del motor DC

Para variar la velocidad del motor se hace uso del Timer 13 del microcontrolador en modo PWM a una frecuencia de 20 KHz variando el registro CCR1. Se busca trabajar a una resolución máxima con la frecuencia antes mencionada, por lo que si  $PSC = 0$ :

$$T = \frac{1}{20 \text{ KHz}} = \frac{ARR_{13} + 1}{CK_{INT}}$$
$$ARR_{13} = 4199$$

### 3.2.3. Lectura de la posición y velocidad lineal

Para poder leer la posición en la articulación, es necesario entonces configurar el Timer 2 del microcontrolador en modo encoder ascendente para ambos sentidos de giro con decodificación 4x, enlazado al Timer 5 para determinar la velocidad lineal.

$$PPR_{4x} = 4 * PPR_{1x} = 800 \text{ ppr}$$

Donde:

$PPR_{4x}$ : ppr del encoder óptico con decodificación 4x.

Para determinar la posición, se puede aplicar la siguiente ecuación:

$$Posicion = CNT_2 * \frac{D_{aa}}{PPR_{4x} * R} [um]$$

$$Posicion = CNT_2 * 0.3603404792 [um]$$

Y para determinar la velocidad lineal, primero es necesario leer los PPS y luego calcular la velocidad usando la siguiente ecuación

$$Velocidad = PPS * \frac{D_{aa}}{PPR_{1x} * R} [um]$$

$$Velocidad = PPS * 1.441361917 \left[ \frac{um}{s} \right]$$

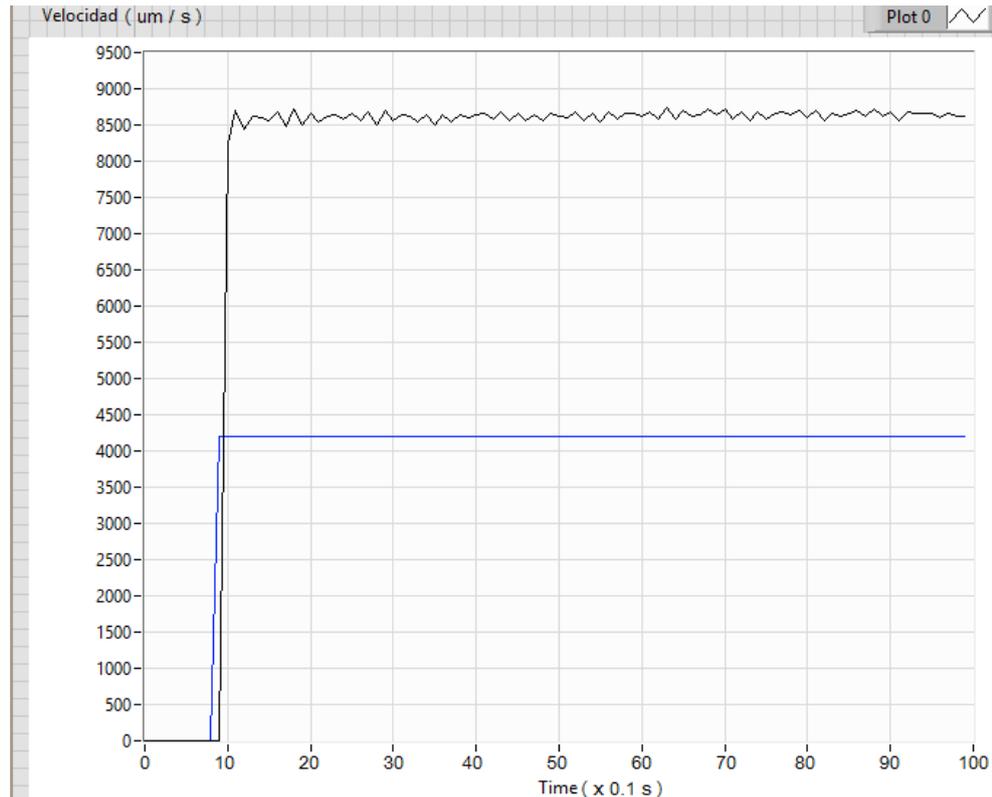
### 3.2.4. Temporizador con Systick

El Timer interno del Cortex M4, systick, es utilizado como base de tiempos con periodos de 1 ms, para una variable estructural llamada "Timer" y así poder temporizar ciertos eventos en el mismo bucle infinito que se suele usar para la

ejecución de un programa escrito en C, sin la necesidad de usar interrupciones u otros Timer en el microcontrolador.

### 3.2.5. Comportamiento de la velocidad lineal

Como se puede ver en la figura 26, la velocidad lineal máxima obtenida en la articulación lineal es de aproximadamente 8.6 mm/s.

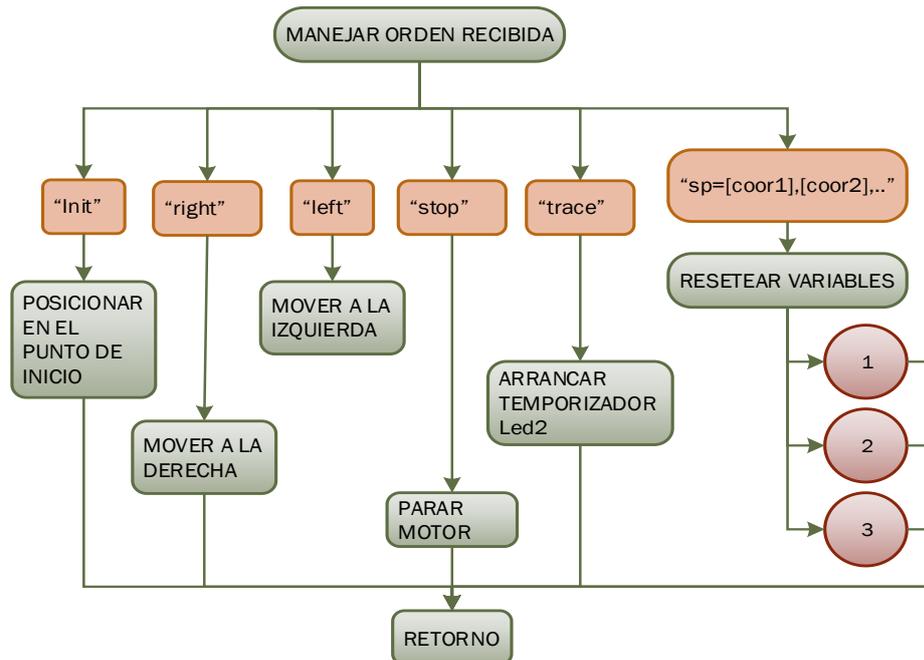


**Figura 26.** Velocidad máxima en articulación lineal.

*Fuente: Elaboración Propia.*

### 3.2.6. Manejo de órdenes

Es necesario hacer uso de la programación de una función encargada de manejar ordenes enviadas por computadora para ordenar el seguimiento de trayectorias y/o controlarlas manualmente. Esta función servirá para todos los programas que serán desarrollados de aquí en adelante. En la figura 27 se presenta su diagrama, y la cadena recibida “sp = [coor1],[coor2],...” es el encargado de ordenar el seguimiento de las trayectorias.



**Figura 27.** Diagrama de flujo de la función de manejo de órdenes enviadas por computadora.

*Fuente: Elaboración Propia.*

### 3.2.7. Descripción de la toma de datos

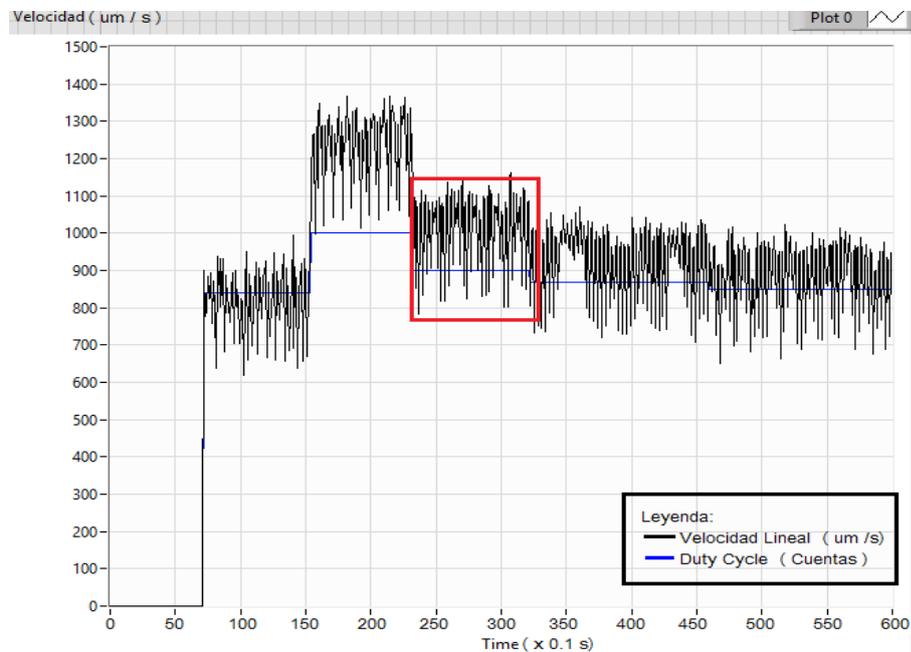
Es necesario realizar tomas de datos de seguimiento de trayectorias en el sistema de control antes descrito con un control en lazo abierto y con uno realimentado (denominado controlador de aquí en adelante) haciendo uso del siguiente procedimiento:

- Posicionar el efector final, que mueve el vernier, a la posición inicial con la orden “init” enviada desde la computadora.
- Esperar que el efector final active el sensor final de carrera que indica la posición inicial, e iniciar un leve desplazamiento en sentido contrario.
- Iniciar el envío de datos hacia la computadora de la posición y la velocidad lineal de la articulación sobre un instrumento virtual creado en LabView 2018, con la orden “trace”. Esta orden activa el temporizador “Led2” el cual cada 100 ms entrará a una rutina de envío de datos por UART.

- Ordenar los puntos cartesianos de las trayectorias a seguir con el comando “sp=[coor1],[coor2],...”.
- Realizar 10 pruebas: Consecutivas para cada punto cartesiano de 0.1, 1 y 10 mm; para 70 mm es necesario primero regresar a la posición inicial.
- Leer por el canal depurador ITM la posición final de la trayectoria seguida según el encoder y observar la posición final medida desde el vernier digital.
- Leer tiempos de establecimiento por el canal depurador.
- Calcular el error relativo porcentual (E.R) y el error absoluto (E.A) con 3 decimales de precisión, del promedio de las pruebas.

### 3.3. Control en lazo abierto

Para llevar a cabo un control no realimentado o en lazo abierto, se buscó primero determinar el ciclo de trabajo de la señal PWM para el driver de motor con la finalidad de obtener una velocidad lineal de aproximadamente 1 mm/s en la articulación, el cual como se aprecia dentro del marco rojo (figura 28), corresponde a un ciclo de trabajo equivalente a 900 cuentas en el registro CCR1 del Timer 13.



**Figura 28.** Ciclo de trabajo para una velocidad lineal aproximada de 1 mm/s.

*Fuente: Elaboración Propia.*

Es posible seguir una trayectoria durante un tiempo igual a:

$$Tiempo = \frac{x_f - x_i}{v (CCR1_{13})} [s]$$

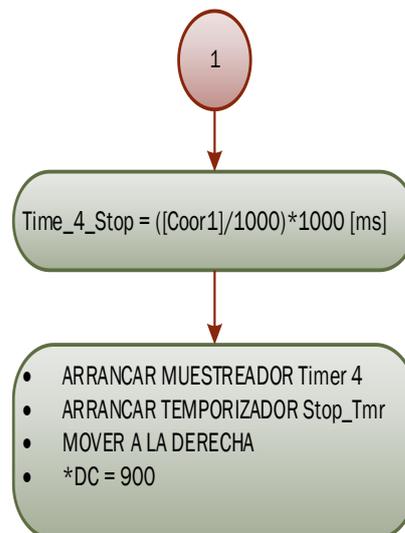
Donde:

$$CCR1_{13} = 900 \qquad v (CCR1_{13}) = 1mm/s$$

La variable “Tiempo” es establecido como parámetro de un temporizador creado con la variable estructural “Timer” anteriormente mencionada.

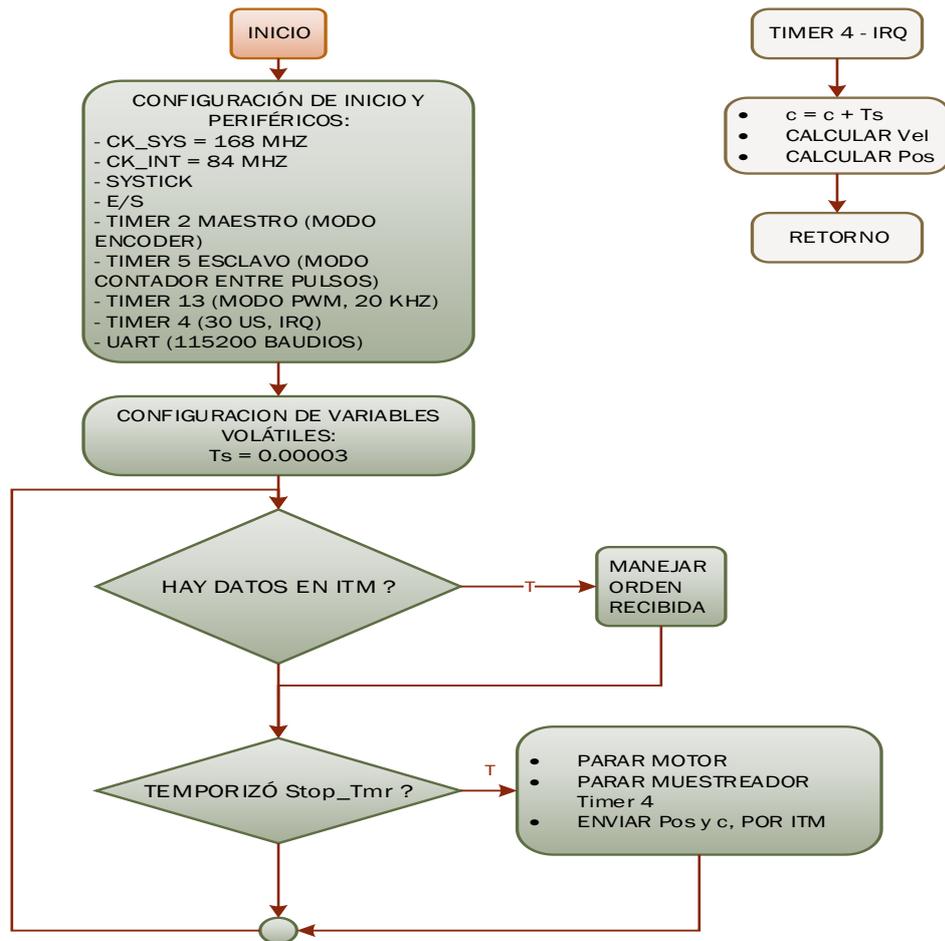
Se utilizó un tiempo de muestreo de 30 us realizado por el Timer 4, configurado para generar interrupciones, con la finalidad de medir el tiempo que le toma a cada trayectoria y verificarlo con la ecuación cinemática anterior.

Los diagramas de flujo del programa para este tipo de control se muestran a continuación, como parte de la continuación de la figura 27.



**Figura 29.** Diagrama de flujo del manejo de la orden “sp=[coor1],[coor2]..”, para el control en lazo abierto.

*Fuente: Elaboración Propia.*



**Figura 30.** Diagrama de flujo del bucle principal del control en lazo abierto.

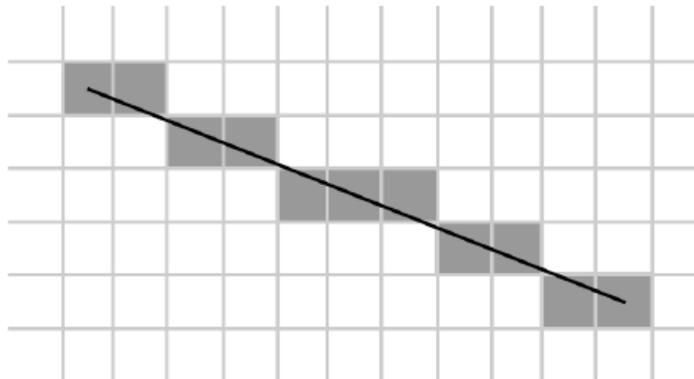
*Fuente: Elaboración Propia.*

### 3.4. Técnicas para la coordinación de articulaciones lineales

De acuerdo a una pesquisa, se encuentran las siguientes técnicas que pueden ser aplicadas a coordinación de 2 articulaciones lineales (conformadas por un motor DC y un husillo) para el seguimiento de trayectorias.

#### 3.4.1. Rasterización de trayectorias

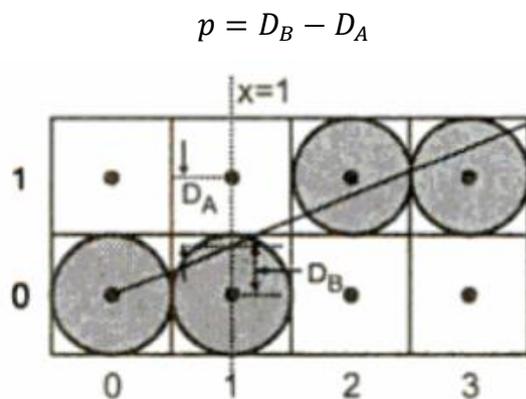
De acuerdo a Hernández (2014), la rasterización es el proceso por el cual datos vectoriales se transforman en un conjunto de píxeles, en este caso, pasos o sub desplazamientos que deben ser realizados en las articulaciones. Existe el algoritmo de Bresenham para la rasterización de líneas rectas y el algoritmo de punto medio para la rasterización de circunferencias.



**Figura 31.** Rasterización de trayectorias lineales.

*Fuente: Hernández (2014).*

En el algoritmo de Bresenham los pasos en las articulaciones son incrementados por 1 dependiendo de la pendiente que formen en la trayectoria a seguir. Básicamente los incrementos dependen de la variable de decisión “p” que se relaciona con las distancias que existen entre los puntos medios de los pixeles y la recta de la trayectoria.



**Figura 32.** Representación del algoritmo de Bresenham.

*Fuente: Hernández (2014).*

Cabe resaltar que mientras los pasos sean más pequeños, el seguimiento de la trayectoria es mejor.

### **3.4.2. Cálculo cinemático**

De acuerdo a Bustamante (2013), es posible aplicar esta técnica para la coordinación de 2 articulaciones lineales, a partir del conocimiento de las variables cinemáticas involucradas en el seguimiento de una trayectoria. Es entonces que de acuerdo a la aplicación de un perfil de velocidad y los puntos que ha de seguir una articulación, es posible calcular de los parámetros de la otra y de esta manera conseguir un movimiento coordinado.

Para aplicar esta técnica es necesario:

- Para cada articulación: El punto de inicio y de llegada; la evolución de la velocidad y la aceleración.
- Determinar la articulación que le tome más tiempo en realizar una trayectoria de tal manera que los parámetros de la segunda sean restablecidos de acuerdo a los parámetros de la primera.

### 3.5. Desarrollo del controlador de seguimiento de trayectorias lineales

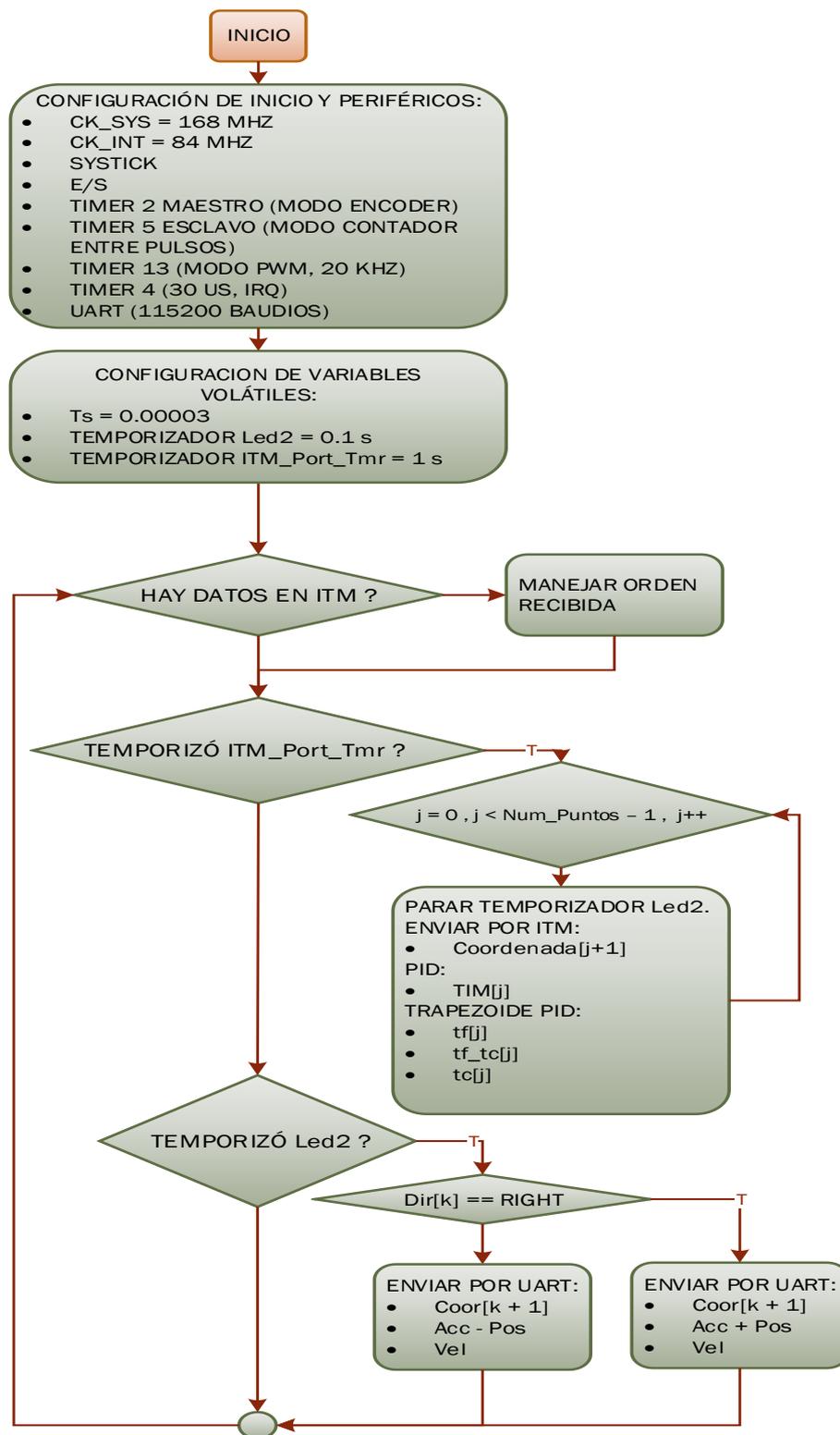
Se optó por desarrollar un control PID discreto que posteriormente sería útil para el desarrollo de un algoritmo de control que incluya un perfil trapezoidal de velocidades (algoritmo Trapezoide – PID), ya que es adecuado para aplicaciones robóticas (Bustamante, 2013) y el movimiento de un robot requiere de velocidades controladas (Fiestas, 2017).

Para llevar a cabo las pruebas de seguimiento con el control PID discreto y el algoritmo Trapezoide - PID, se tiene el siguiente diagrama de flujo, parte de la continuación del diagrama “MANEJAR ORDEN RECIBIDA” de la figura 27.



**Figura 33.** Diagrama de flujo del manejo de la orden “sp=[coor1],[coor2]..”, para el control PID discreto y el algoritmo Trapezoide - PID.

*Fuente: Elaboración Propia.*



**Figura 34.** Diagrama de flujo del bucle principal del control PID discreto y algoritmo Trapezoide - PID.

*Fuente: Elaboración Propia.*

Así mismo la figura 34 muestra el diagrama del bucle principal de ambos diseños de control, el cual se reduce a uno solo ya que solo varía la información que se envía por el canal depurador ITM.

Como se puede observar la temporización del canal depurador es para enviar información referente a la posición y tiempo de seguimiento de la trayectoria, e información de la posición y tiempos de recorrido que se siguen en el perfil trapezoidal de velocidades.

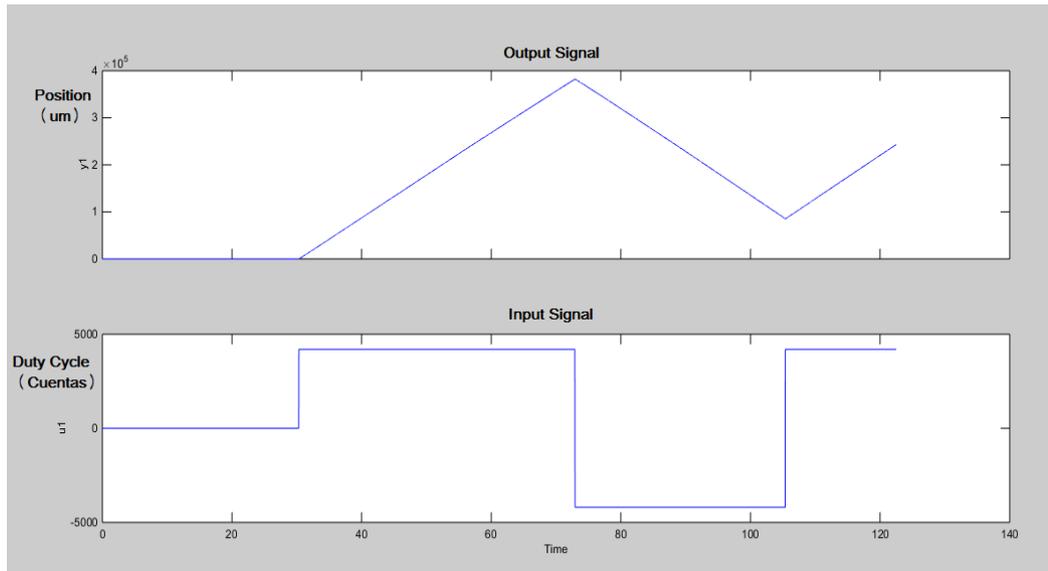
### **3.5.1. Requerimientos**

Para el desarrollo de un controlador de seguimiento de trayectorias lineales orientado a un robot cartesiano tipo gantry, se considera lo siguiente:

1. Sobre pico menor al 1% o nulo.
2. Tiempo de establecimiento menor a 200 ms.
3. Error en estado estable igual a cero.
4. Para un robot cartesiano tipo gantry es necesario poder seguir trayectorias lineales con un movimiento isócrono y continuo (Fiestas, 2017).

### **3.5.2. Identificación del modelo de la articulación lineal**

Se optó por realizar una adquisición de datos con un instrumento virtual elaborado en LabView 2018, al cual se le enviaron datos de posición en micrómetros y ciclo de trabajo de la señal PWM para posteriormente utilizar la herramienta System Identification Toolbox de Matlab para identificar el modelo matemático aproximado que describe el comportamiento de la articulación lineal.



**Figura 35.** Identificación de la articulación lineal.

*Fuente: Elaboración Propia.*

Se obtuvo el siguiente modelo matemático en el plano S de Laplace:

$$\frac{\text{Posicion (um)}}{\text{Ciclo de Trabajo}} = \frac{b}{s(s + a)}$$

Donde

$$b = 367.0544 \quad a = 177.283$$

Se puede ver entonces que el sistema tiene el primer polo en  $S = 0$ , y el segundo en  $S = -177.283$ .

### 3.5.3. Control PID discreto

Se procedió primero con el diseño de un control PID en tiempo continuo con el método de la cancelación de polos y la adición de un polo dominante que rija el comportamiento de la articulación lineal en el seguimiento de una trayectoria.

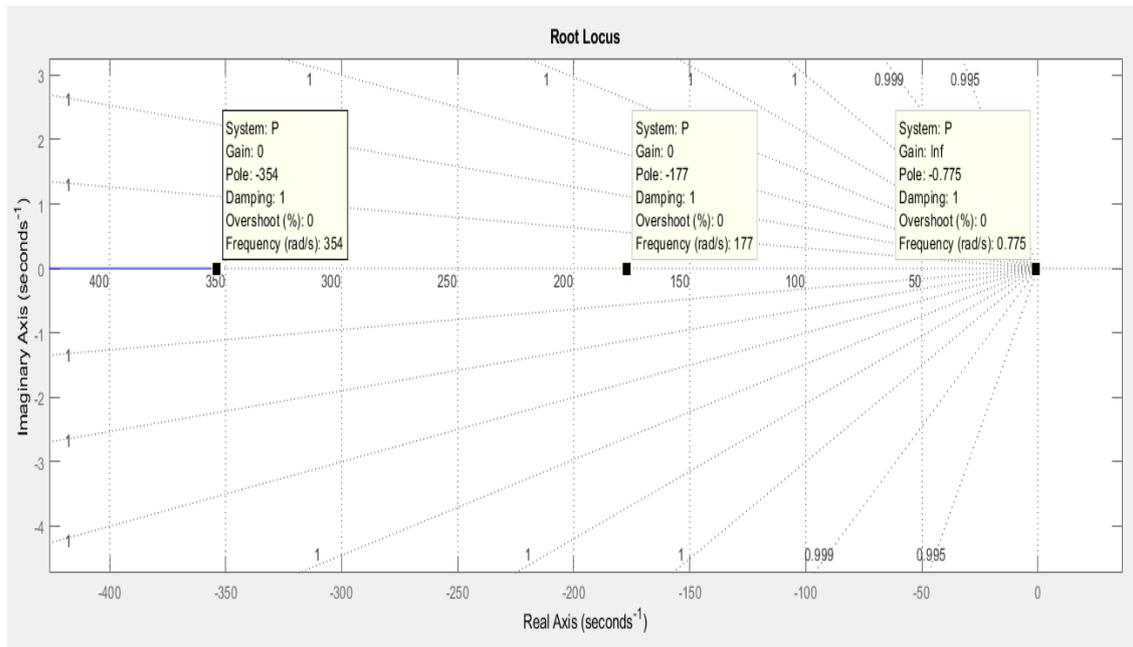
Se obtuvieron las siguientes constantes:

$$k_p = 172$$

$$k_i = 132.7409554$$

$$k_d = 0.965976705$$

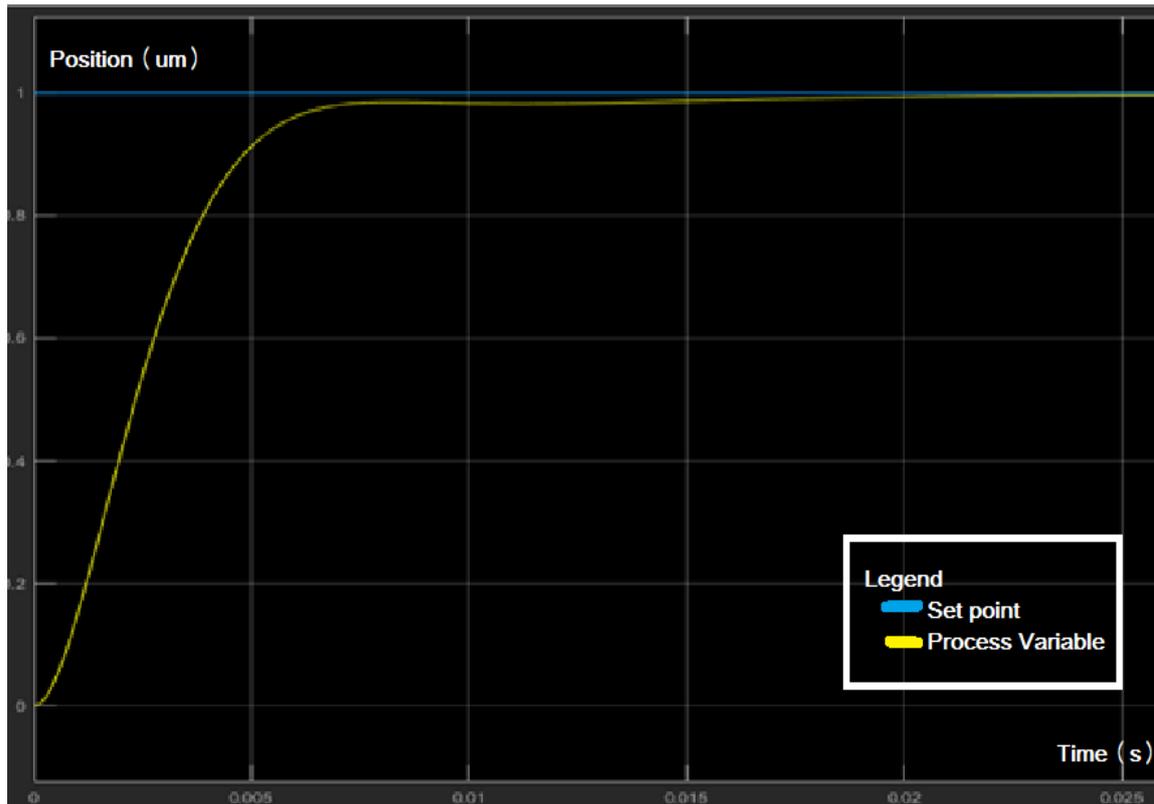
Como se observa en la figura 36, los lugares geométricos del bloque PID en cascada con el modelo de la articulación y en lazo realimentado, muestra que los dos polos existentes se cancelan con los dos ceros provenientes del PID y a la vez surge un nuevo polo en aproximadamente  $-354$  que es casi el doble del segundo ( $-177.283$ ).



**Figura 36.** Lugares geométricos del bloque PID en cascada con la articulación lineal en lazo realimentado.

*Fuente: Elaboración Propia.*

Como se observa en la figura 37, en la simulación del control PID para una trayectoria de  $1 \mu\text{m}$ , el tiempo de establecimiento es menor a  $30 \text{ ms}$  con un error en estado estable muy próximo a cero y un sobre impulso menor al  $1\%$ .



**Figura 37.** Simulación del seguimiento de una trayectoria de 1  $\mu\text{m}$  con control PID en tiempo continuo.

*Fuente: Elaboración Propia.*

Posteriormente se implementa dicho control PID en el microcontrolador con la siguiente ecuación en diferencias:

$$y(k) = y(k - 1) + b_0 * e(k) + b_1 * e(k - 1) + b_2 * e(k - 2)$$

Que fue obtenida en el apartado 2.2.6.2, y siguiendo con el uso del tiempo de muestreo de 30  $\mu\text{s}$ , ya que es mucho menor que la inversa del polo dominante.

$$b_0 = 172.9659767$$

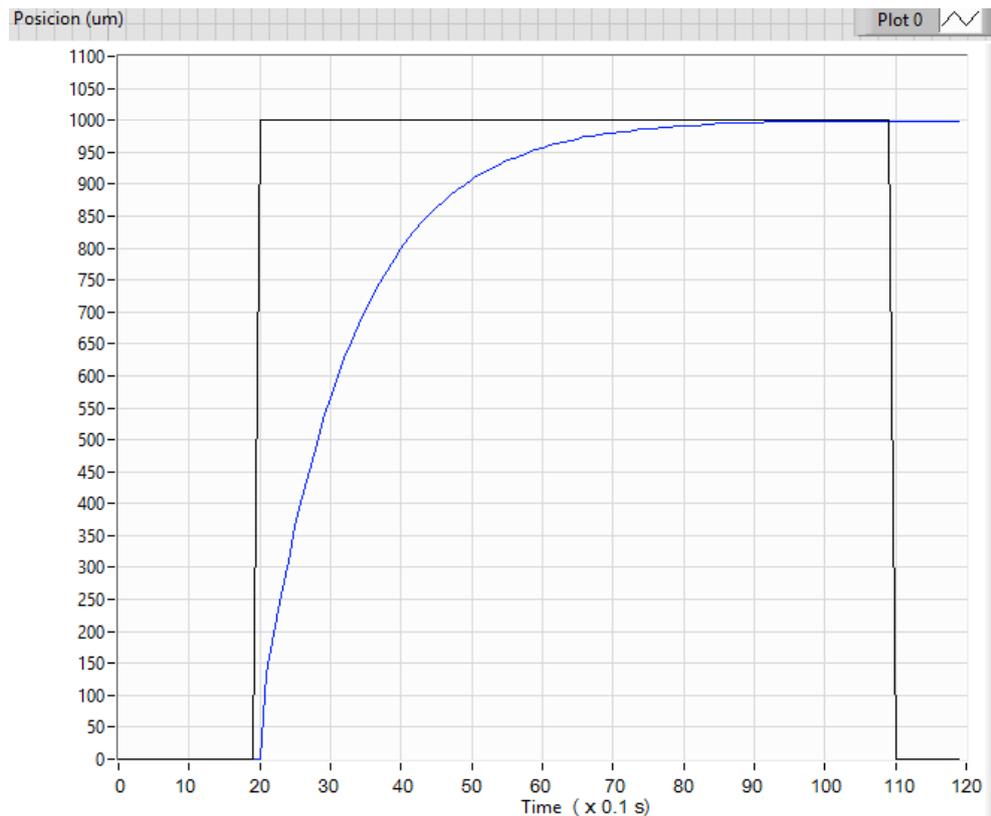
$$b_1 = -173.9279712$$

$$b_2 = 0.965976705$$

Ahora la rutina de la interrupción por Timer 4 se muestra a continuación en la figura 38.

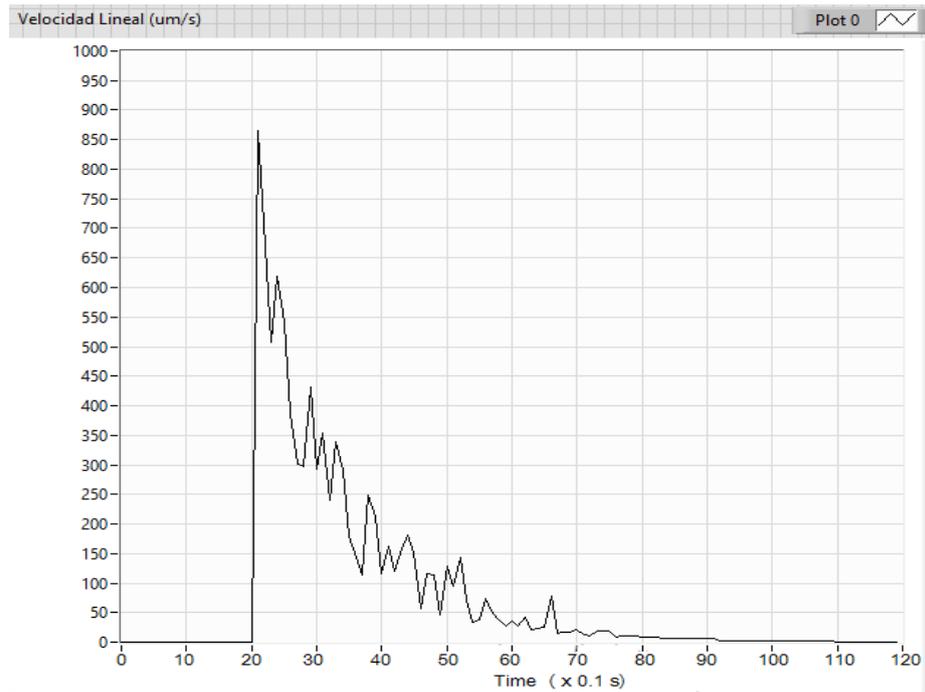


A continuación, se muestra la respuesta en el tiempo tanto de la posición (figura 39) como de la velocidad lineal (figura 40) para el seguimiento de una trayectoria de 1 mm con el control PID ya implementado en el microcontrolador y en operación con la articulación lineal, en un instrumento virtual realizado en LabView 2018 (anexo 7). La velocidad varía de acuerdo a la trayectoria a seguir y no es definida en el controlador. En la figura 41 se muestra el resultado del depurador para el seguimiento de dicha trayectoria.



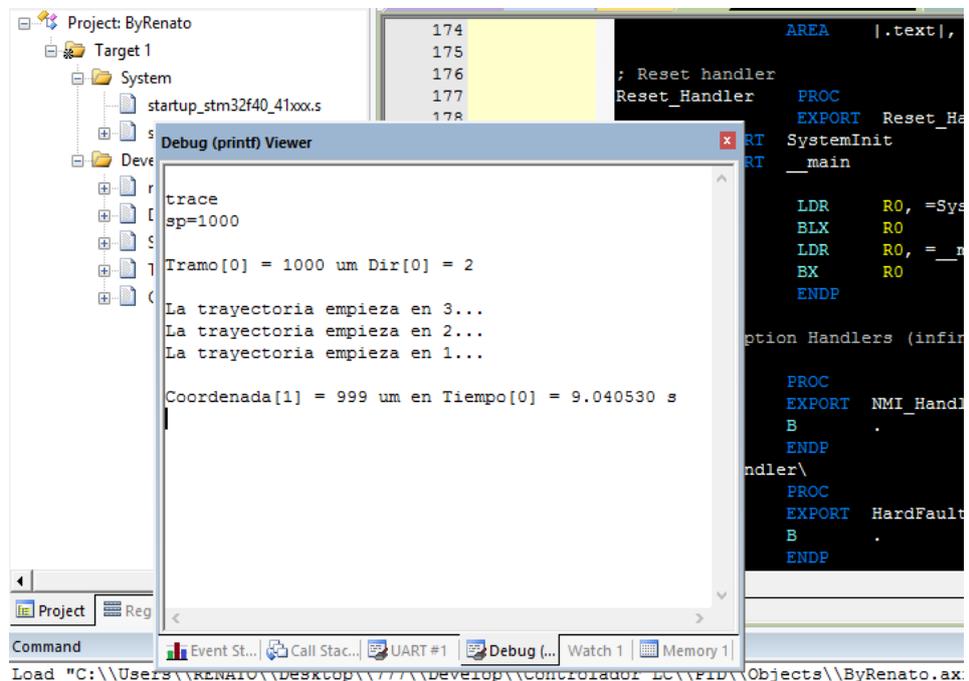
**Figura 39.** Respuesta en el tiempo de la posición en el seguimiento de una trayectoria de 1 mm con el control PID discreto implementado.

*Fuente: Elaboración Propia.*



**Figura 40.** Respuesta en el tiempo de la velocidad en el seguimiento de una trayectoria de 1 mm con el control PID discreto implementado.

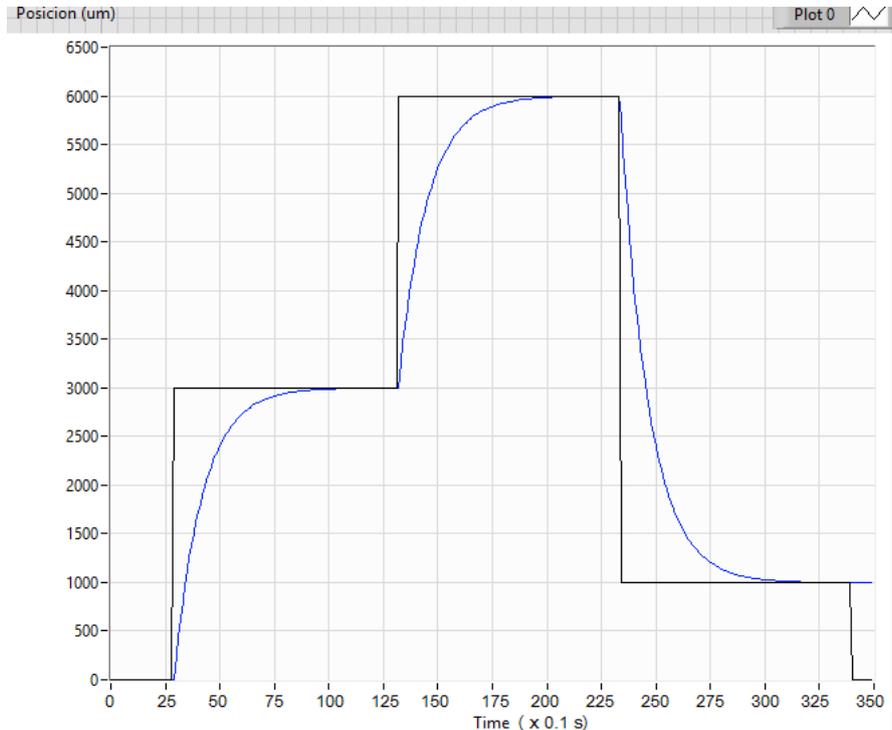
*Fuente: Elaboración Propia.*



**Figura 41.** Resultado del depurador en el seguimiento de una trayectoria de 1 mm con el control PID discreto implementado.

*Fuente: Elaboración Propia.*

Adicionalmente, se demuestra el seguimiento de trayectorias consecutivas y en ambos sentidos de la articulación para formar un movimiento continuo o de contorno, y de esta manera simular una generación de trayectorias que suele producir un robot cartesiano tipo gantry para posicionar el efecto final.



**Figura 42.** Respuesta en el tiempo de la posición en el seguimiento de trayectorias consecutivas según las coordenadas: 3 mm, 6 mm y 1 mm con el control PID discreto implementado.

*Fuente: Elaboración Propia.*

### 3.5.4. Algoritmo Trapezoide – PID

En base al control PID discreto anteriormente desarrollado, es posible aplicar un algoritmo que realice un perfil trapezoidal para la velocidad lineal en el seguimiento, permitiendo así un control de la posición en un tiempo específico controlando implícitamente la velocidad.

Este algoritmo debe establecer en tiempo real las posiciones que se han de seguir en la trayectoria partir del cálculo de velocidades dentro de un tiempo de llegada o final ( $t_f$ ) y un tiempo crítico ( $t_c$ ) según los parámetros establecidos: Punto de

inicio ( $x_i$ ) y de llegada ( $x_f$ ), velocidad máxima ( $V_{max}$ ), y aceleración ( $a$ ); como se demostró en el apartado 2.2.3.2.

El algoritmo requiere lo siguiente:

- Calcular  $t_f$  y  $t_c$  de acuerdo a las velocidades máximas y aceleraciones previamente definidas y el ingreso de los puntos de inicio y final de la trayectoria a seguir.
- Tener en cuenta que  $t_f > 2t_c$ . Si no se cumple dicha condición, variar uno de los parámetros, excepto la trayectoria ya definida, de acuerdo a la siguiente ecuación:

$$a > \frac{v_{max}^2}{\Delta x}$$

- Utilizar las siguientes ecuaciones cinemáticas para el cálculo de las velocidades y posiciones en tiempo real y discreto para cada tramo del perfil trapezoidal.

**Tramo de aceleración** ( $t_k < t_c$ )

$$v_k = a * t_k$$

**Tramo de velocidad constante** ( $t_c \leq t_k < t_f - t_c$ )

$$v_k = v_{max}$$

**Tramo de desaceleración** ( $t_f - t_c \leq t_k \leq t_f$ )

$$v_k = a * (t_f - t_k)$$

**Para todos los tramos**

$$x_k = x_{k-1} + v_k * T_s$$

Donde:

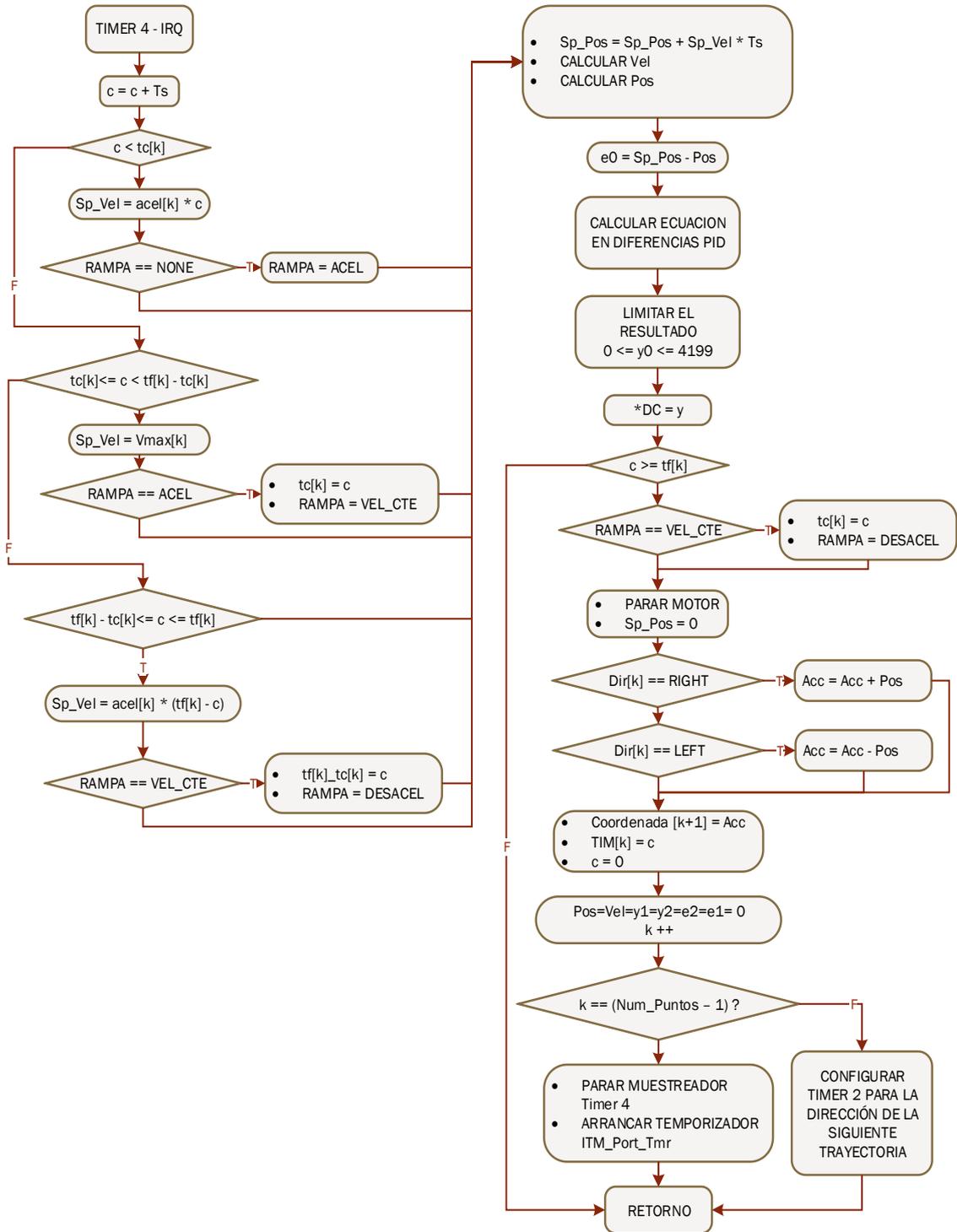
$v_k$ : Velocidad Lineal cada  $T_s$

$x_k$ : Posición Lineal cada  $T_s$

$x_{k-1}$ : Posición en un  $T_s$  anterior

$t_k$ : Tiempo muestreado cada  $T_s$

Se mantiene el tiempo de muestreo de 30 us y la rutina de la interrupción del Timer 4 se muestra en el siguiente diagrama.



**Figura 43.** Diagrama de flujo de interrupción por Timer 4 para el algoritmo Trapezoide - PID.

*Fuente: Elaboración Propia.*

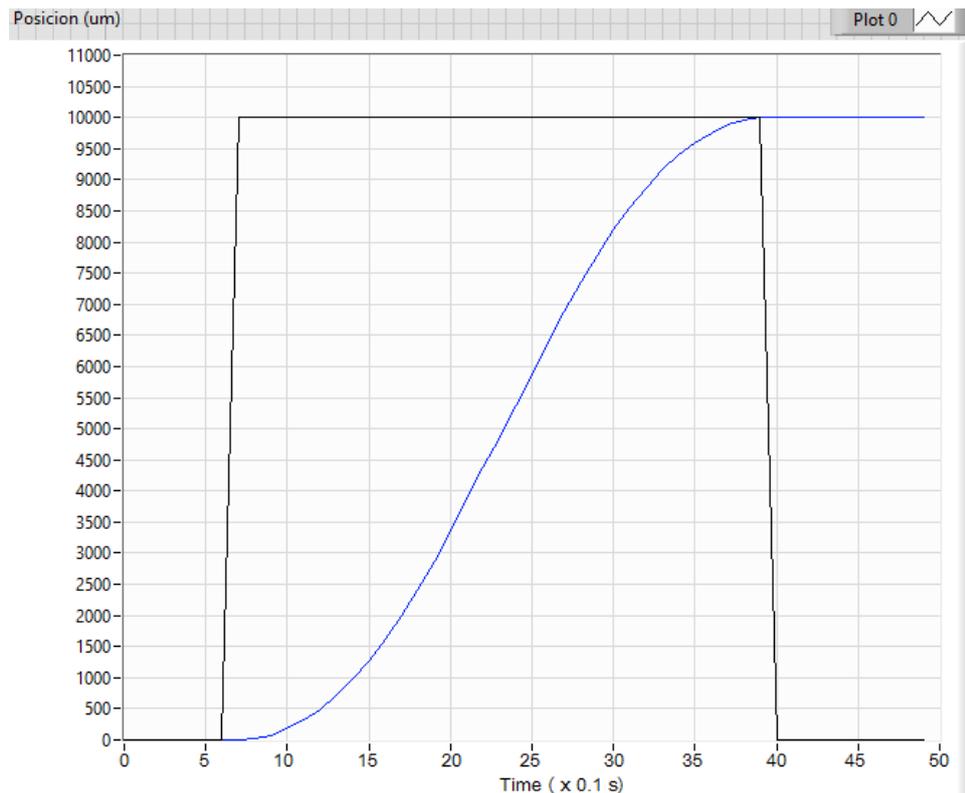
A continuación, se muestra la respuesta en el tiempo tanto de la posición (figura 44) como de la velocidad (figura 45) para el seguimiento de una trayectoria de 10 mm con el algoritmo Trapezoide - PID ya implementado en el microcontrolador y en operación con la articulación lineal, en un instrumento virtual realizado en LabView 2018 (anexo 7). Los parámetros fueron los siguientes:

$$x_0 = 0, \quad x_1 = 10000 \text{ } \mu\text{m}, \quad v_{max} = 5000 \frac{\mu\text{m}}{\text{s}}, \quad a = 4000 \frac{\mu\text{m}}{\text{s}}$$

Obteniendo:

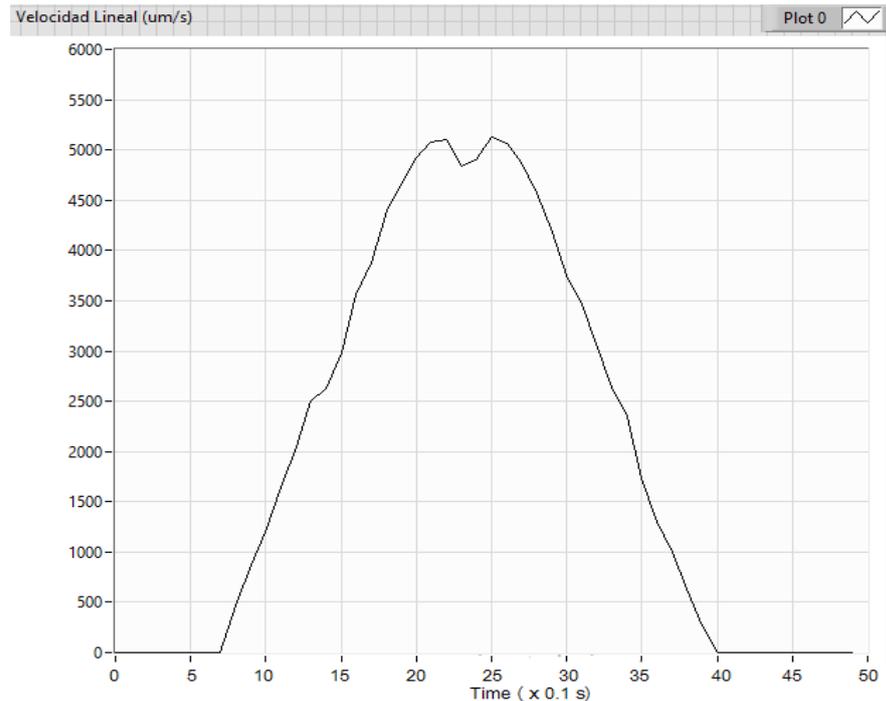
$$t_c = 1.25 \text{ s}, \quad t_f = 3.25 \text{ s}$$

En la figura 46 se muestra el resultado del depurador para el seguimiento de la trayectoria.



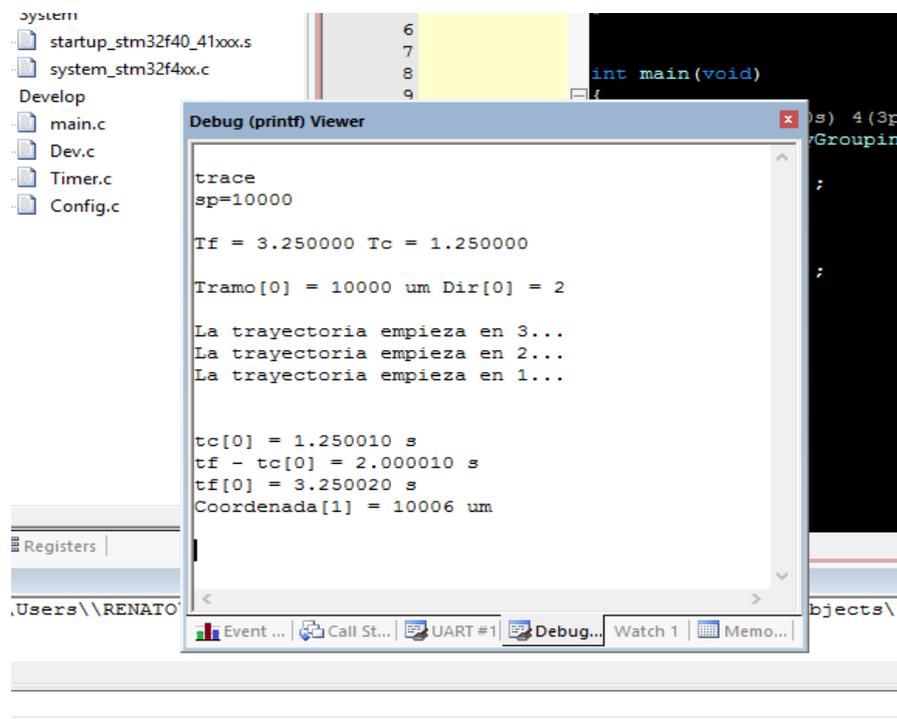
**Figura 44.** Respuesta en el tiempo de la posición en el seguimiento de una trayectoria de 10 mm con el algoritmo Trapezoide – PID implementado.

*Fuente: Elaboración Propia.*



**Figura 45.** Respuesta en el tiempo de la velocidad en el seguimiento de una trayectoria de 10 mm con el algoritmo Trapezoide – PID implementado.

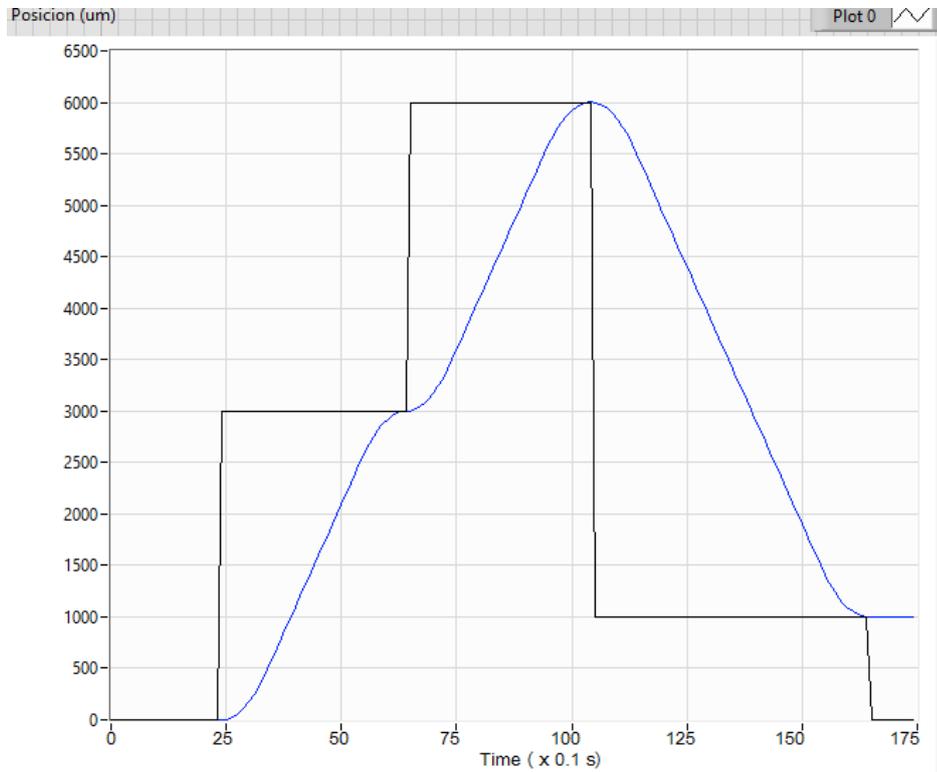
*Fuente: Elaboración Propia.*



**Figura 46.** Resultado del depurador en el seguimiento de una trayectoria de 10 mm con el algoritmo Trapezoide – PID implementado.

*Fuente: Elaboración Propia.*

Al igual que el control PID discreto, es necesario demostrar el seguimiento trayectorias consecutivas y en ambos sentidos de la articulación para formar un movimiento continuo o de contorno.



**Figura 47.** Respuesta en el tiempo de la posición en el seguimiento de trayectorias consecutivas según las coordenadas: 3 mm, 6 mm y 1 mm con el algoritmo Trapezoide – PID implementado.

*Fuente: Elaboración Propia.*

### 3.5.5. Seguimiento de trayectorias lineales

Previamente se realizan pruebas con ambos controladores para determinar el más conveniente a la hora de querer reducir el error en las trayectorias lineales. Con respecto a las pruebas del controlador con el algoritmo Trapezoide - PID, se utilizaron los parámetros de la tabla 4.

**Tabla 4.** Parámetros para el controlador con el algoritmo Trapezoide - PID.

Trayectoria (mm) \ Parámetros	0.1	1	10	70
<b>x0 (um)</b>	0	0	0	0
<b>x1 (um)</b>	100	1000	10000	70000
<b>vmax (um/s)</b>	1000	1000	5000	8000
<b>a &gt;</b>	10000	1000	2500	914.2857
<b>a (um/s^2)</b>	11000	4000	4000	1000
<b>tc (s)</b>	0.0909	0.25	1.25	8
<b>tf (s)</b>	0.1909	1.25	3.25	16.75

*Fuente: Elaboración Propia.*

**Tabla 5.** Trayectorias lineales de 0.1 mm con el controlador desarrollado PID y Trapezoide - PID.

0.1	PID			PID + T		
k	Encoder (mm)	Vernier (mm)	Tiempo (s)	Encoder (mm)	Vernier (mm)	Tiempo (s)
1	0.099	0.03	3.78663	0.099	0.09	0.190909
2	0.099	0.11	2.57361	0.098	0.12	0.190909
3	0.099	0.11	3.5733	0.098	0.11	0.190909
4	0.099	0.10	3.95082	0.099	0.12	0.190909
5	0.099	0.10	3.15795	0.098	0.13	0.190909
6	0.099	0.10	3.67794	0.098	0.08	0.190909
7	0.099	0.08	4.2534	0.099	0.10	0.190909
8	0.099	0.09	3.56145	0.099	0.10	0.190909
9	0.099	0.09	3.91989	0.098	0.09	0.190909
10	0.099	0.09	4.34928	0.099	0.09	0.190909
<b>Promedio</b>	0.099	0.09		0.0985	0.103	
<b>E.R %</b>	1.000	10.000		1.500	3.000	
<b>E.A (mm)</b>	0.001	0.010		0.002	0.003	

*Fuente: Elaboración Propia.*

**Tabla 6.** Trayectorias lineales de 1 mm con el controlador desarrollado PID y Trapezoide - PID.

1 k	PID			PID + T		
	Encoder (mm)	Vernier (mm)	Tiempo (s)	Encoder (mm)	Vernier (mm)	Tiempo (s)
1	0.999	0.99	8.43021	1.000	0.98	1.25001
2	0.999	1.04	8.91837	1.001	1.02	1.25001
3	0.999	0.96	8.93979	1.001	0.99	1.25001
4	0.999	1.09	9.05637	1.001	1.09	1.25001
5	0.999	0.93	8.86722	1.001	0.94	1.25001
6	0.999	1.12	8.93184	1.000	1.11	1.25001
7	0.999	0.86	8.74116	1.000	0.87	1.25001
8	0.999	1.1	8.89752	1.001	1.09	1.25001
9	0.999	0.87	8.72127	1.001	0.90	1.25001
10	0.999	1.16	8.90922	1.001	1.14	1.25001
<b>Promedio</b>	0.999	1.012		1.0007	1.013	
<b>E.R %</b>	0.100	1.200		0.070	1.300	
<b>E.A (mm)</b>	0.001	0.012		0.001	0.013	

*Fuente: Elaboración Propia.*

**Tabla 7.** Trayectorias lineales de 10 mm con el controlador desarrollado PID y Trapezoide - PID.

10 k	PID			PID + T		
	Encoder (mm)	Vernier (mm)	Tiempo (s)	Encoder (mm)	Vernier (mm)	Tiempo (s)
1	9.999	10.25	11.50866	10.007	10.13	3.25002
2	9.999	10.03	11.58657	10.006	10.11	3.25002
3	9.999	9.91	11.58522	10.005	10.04	3.25002
4	9.999	9.96	11.69877	10.007	9.91	3.25002
5	9.999	10	11.67885	10.007	9.95	3.25002
6	9.999	9.99	11.46834	10.006	9.97	3.25002
7	9.999	10.03	12.04488	10.006	10.04	3.25002
8	9.999	10.01	11.63817	10.007	10.04	3.25002
9	9.999	9.92	11.96667	10.005	9.95	3.25002
10	9.999	9.58	11.53689	10.006	9.57	3.25002
<b>Promedio</b>	9.999	9.968		10.0062	9.971	
<b>E.R %</b>	0.010	0.320		0.062	0.290	
<b>E.A (mm)</b>	0.001	0.032		0.006	0.29	

*Fuente: Elaboración Propia.*

**Tabla 8.** Trayectorias lineales de 70 mm con el controlador desarrollado PID y Trapezoide - PID.

70	PID			PID + T		
k	Encoder (mm)	Vernier (mm)	Tiempo (s)	Encoder (mm)	Vernier (mm)	Tiempo (s)
1	69.999	70.03	32.21358	70.002	70.43	16.75
2	69.999	70.11	32.49012	70.002	70.16	16.75
3	69.999	70.13	32.05659	70.001	70.2	16.75
4	69.999	70.09	32.42388	70.001	70.24	16.75
5	69.999	70.12	32.50956	70.002	70.47	16.75
6	69.999	70.18	31.85844	70.002	70.26	16.75
7	69.999	70.21	32.28252	70.001	70.07	16.75
8	69.999	70.14	32.17827	70.002	70.19	16.75
9	69.999	70.24	31.39269	70.001	70.16	16.75
10	69.999	70.27	31.88574	70.001	70.2	16.75
<b>Promedio</b>	69.999	70.152		70.0015	70.238	
<b>E.R %</b>	0.001	0.217		0.002	0.340	
<b>E.A (mm)</b>	0.001	0.152		0.002	0.238	

*Fuente: Elaboración Propia.*

De acuerdo a estos resultados, se opta por considerar el controlador con el algoritmo Trapezoide - PID para reducir el error en las trayectorias lineales, debido a sus parámetros cinemáticos programables convenientes para un robot cartesiano, sus resultados muy similares al control PID discreto, y debido a que, en comparación con este, resulta con tiempos de llegada constantes para una misma trayectoria.

### 3.6. Seguimiento de trayectorias lineales isócronas

En este apartado se busca demostrar el seguimiento de trayectorias lineales isócronas, ya que según las pruebas anteriores se abordó el seguimiento de trayectorias continuas. Es posible aplicar la técnica de “cálculo cinemático” para el movimiento coordinado de 2 articulaciones lineales, como también es posible aplicar la “rasterización de trayectorias”. Se opta por aplicar la primera ya que es más viable. Haciendo uso del microcontrolador STM32F407ZG es posible demostrar el seguimiento de trayectorias lineales isócronas mediante una simulación y la modificación del algoritmo Trapezoide – PID.

### 3.6.1. Algoritmo

De acuerdo al algoritmo Trapezoide – PID previamente diseñado, para este caso son necesarias las siguientes modificaciones:

- Establecer una articulación en el eje X y en el eje Y.
- Calcular para las 2 articulaciones, los tiempos de llegada ( $t_f$ ) y críticos ( $t_c$ ) a una misma velocidad máxima y aceleración.
- Al menos una de las articulaciones debe cumplir la condición  $t_f > 2t_c$ .
- Si solo una articulación cumple dicha condición, quiere decir que ese es la que le toma más tiempo en seguir su trayectoria.
- De lo contrario, si ambas cumplen con ello, se debe comparar y determinar cuál es la articulación que le toma más tiempo en seguir su trayectoria.
- Determinar la pendiente que existe en la trayectoria seguida por ambas articulaciones.

$$Pdte = \frac{\Delta S_{menor}}{\Delta S_{mayor}}$$

Donde:

$\Delta S_{menor}$ : diferencia de los puntos de la trayectoria de menor tiempo.

$\Delta S_{mayor}$ : diferencia de los puntos de la trayectoria de mayor tiempo.

- Con las ecuaciones cinemáticas antes descritas, calcular en tiempo real las posiciones de la trayectoria de mayor tiempo.
- Calcular en tiempo real las posiciones y velocidades de la trayectoria de menor tiempo con las siguientes ecuaciones:

$$x_{k\_menor} = Pdte * x_{k\_mayor}$$
$$v_{k\_menor} = \frac{x_{k\_menor} - x_{k-1\_menor}}{t_k - t_{k-1}}$$

### 3.6.2. Simulación

Se tuvieron que hacer algunas modificaciones al algoritmo Trapezoide - PID de tal manera de visualizar solo los puntos de consigna de las posiciones y

velocidades en la trayectoria. El seguimiento de las trayectorias en una articulación lineal es gobernado por dichos puntos de consigna como se ve en la figura 44. Se consideraron los siguientes parámetros para la simulación:

$$x_0 = 0, \quad x_1 = 5000 \text{ } \mu\text{m}, \quad v_{max} = 5000 \frac{\mu\text{m}}{\text{s}}, \quad a = 4000 \frac{\mu\text{m}}{\text{s}}$$

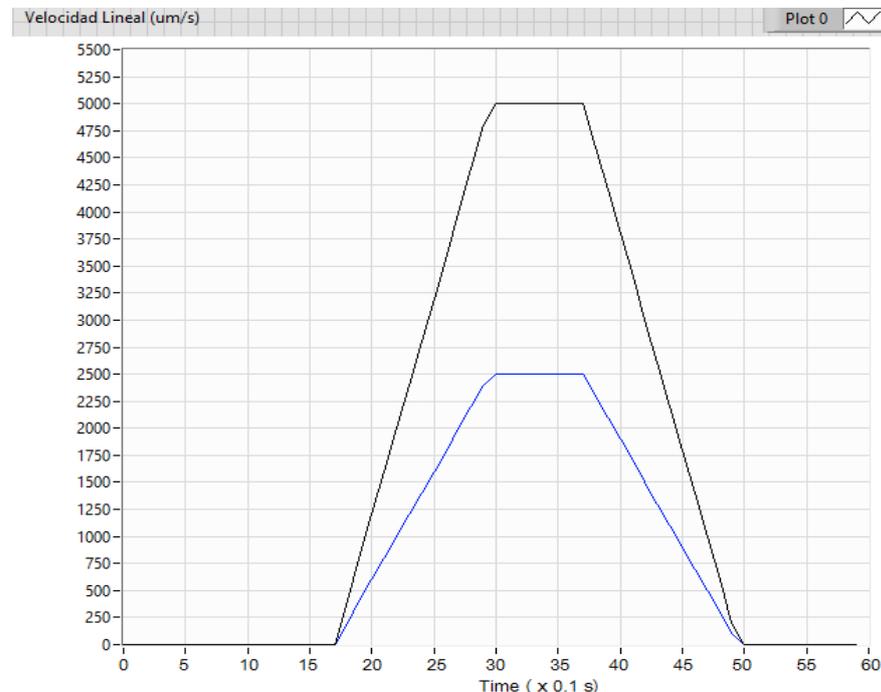
$$y_0 = 0, \quad y_1 = 10000 \text{ } \mu\text{m}$$

Obteniendo:

$$t_{xc} = 1.25 \text{ s}, \quad t_{xf} = 2.25 \text{ s} \qquad t_{yc} = 1.25 \text{ s}, \quad t_{yf} = 3.25 \text{ s}$$

Como la trayectoria en el eje Y tarda más en realizarse, entonces:

$$t_c = 1.25 \text{ s}, \quad t_f = 3.25 \text{ s} \quad Pdte = \frac{\Delta x}{\Delta y} = 0.5$$



**Figura 48.** Respuesta en el tiempo de la velocidad para el seguimiento simulado de una trayectoria de 10 mm en el eje Y, y 5 mm en el eje X.

*Fuente: Elaboración Propia.*

Según la figura 48, todo indica que ambos seguimientos de trayectoria empiezan y culminan al mismo tiempo, donde los parámetros cinemáticos de la articulación en el eje X dependen de los de la articulación en el eje Y.

# **CAPITULO IV**

#### 4. RESULTADOS

➤ Se tienen las siguientes técnicas aplicables al movimiento coordinado de 2 articulaciones lineales compuestas por el acople de un motor DC y un husillo:

1. Rasterización de Trayectorias. El Algoritmo de Bresenham es una técnica que aplica sub desplazamientos a modo de pasos, de acuerdo a una variable de decisión.
2. Cálculo Cinemático. A partir del conocimiento de las variables cinemáticas para el seguimiento de una trayectoria en una articulación, es posible calcular los parámetros del segundo y conseguir un movimiento isócrono.

➤ El controlador desarrollado está en función de un control PID discreto y contiene un algoritmo para la generación de un perfil de velocidad trapezoidal. Las constantes del PID resultaron:

$$k_p = 172 \quad k_i = 132.7409554 \quad k_d = 0.965976705$$

Para el controlador desarrollado se utilizó un tiempo de muestreo de 30 us y la siguiente ecuación en diferencias implementada en el microcontrolador.

$$y(k) = y(k - 1) + b_0 * e(k) + b_1 * e(k - 1) + b_2 * e(k - 2)$$

$$b_0 = k_p + k_d = 172.9659767$$

$$b_1 = -(k_p + 2 * k_d - k_i * T_s) = -173.9279712$$

$$b_2 = k_d = 0.965976705$$

➤ Los resultados del seguimiento de trayectorias lineales con una velocidad de 1 mm/s con el control en lazo abierto son los siguientes.

**Tabla 9.** Trayectorias Lineales de 0.1 mm en lazo abierto.

<b>0.1</b>			
<b>k</b>	<b>Encoder (mm)</b>	<b>Vernier (mm)</b>	<b>Tiempo (s)</b>
1	0.057	0.05	0.09963
2	0.064	0.06	0.09954
3	0.061	0.07	0.09936
4	0.052	0.05	0.09945
5	0.039	0.05	0.09942
6	0.045	0.04	0.09963
7	0.047	0.04	0.09933
8	0.053	0.05	0.09999
9	0.058	0.06	0.09993
10	0.058	0.06	0.09921
<b>Promedio</b>	0.053	0.052	
<b>E.R %</b>	47.111	47.778	
<b>E.A (mm)</b>	0.047	0.048	

*Fuente: Elaboración Propia.*

**Tabla 10.** Trayectorias lineales de 1 mm en lazo abierto.

<b>1</b>			
<b>k</b>	<b>Encoder (mm)</b>	<b>Vernier (mm)</b>	<b>Tiempo (s)</b>
1	0.678	0.70	0.99936
2	0.691	0.66	0.99903
3	0.69	0.77	0.99939
4	0.715	0.75	0.9993
5	0.742	0.7	0.99921
6	0.767	0.9	0.99978
7	0.777	0.69	0.99924
8	0.785	0.9	0.99978
9	0.705	0.65	0.99966
10	0.76	0.7	0.99912
<b>Promedio</b>	0.728	0.747	
<b>E.R %</b>	27.222	25.333	
<b>E.A (mm)</b>	0.272	0.253	

*Fuente: Elaboración Propia.*

**Tabla 11.** Trayectorias lineales de 10 mm en lazo abierto.

<b>10</b>			
<b>k</b>	<b>Encoder (mm)</b>	<b>Vernier (mm)</b>	<b>Tiempo (s)</b>
1	8.275	8.48	9.99921
2	8.782	8.84	9.9993
3	8.847	9.06	9.99921
4	8.642	8.74	9.99942
5	9.305	9.23	9.99999
6	9.051	8.96	9.99957
7	9.309	9.42	9.99921
8	9.362	9.28	9.99987
9	9.465	9.59	9.99936
10	8.984	8.92	9.99945
<b>Promedio</b>	9.004	9.067	
<b>E.R %</b>	9.958	9.333	
<b>E.A (mm)</b>	0.996	0.933	

*Fuente: Elaboración Propia.*

**Tabla 12.** Trayectorias lineales de 70 mm en lazo abierto.

<b>70</b>			
<b>k</b>	<b>Encoder (mm)</b>	<b>Vernier (mm)</b>	<b>Tiempo (s)</b>
1	69.233	69.43	69.99954
2	73.155	73.33	69.9996
3	74.599	74.7	69.99921
4	75.81	75.85	69.99927
5	77.402	77.46	69.99951
6	78.657	78.78	69.99975
7	79.739	79.85	69.99918
8	79.436	79.47	69.99951
9	79.778	79.91	69.99924
10	80.898	81.07	69.99981
<b>Promedio</b>	76.423	76.531	
<b>E.R %</b>	9.176	9.330	
<b>E.A (mm)</b>	6.423	6.531	

*Fuente: Elaboración Propia.*

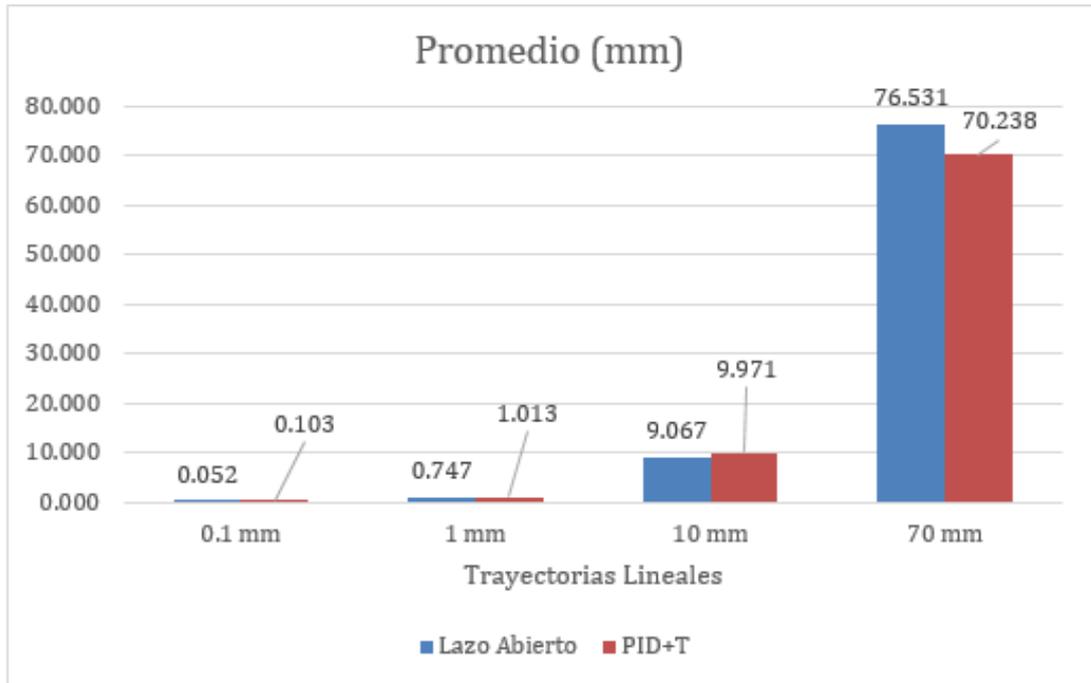
Los resultados del seguimiento de trayectorias lineales con el controlador desarrollado son los siguientes.

**Tabla 13.** Resultados del seguimiento de trayectorias lineales con el controlador desarrollado Trapezoide - PID.

	0.1		1		10		70	
<b>k</b>	<b>Encoder (mm)</b>	<b>Vernier (mm)</b>						
<b>1</b>	0.099	0.09	1.000	0.98	10.007	10.13	70.002	70.43
<b>2</b>	0.098	0.12	1.001	1.02	10.006	10.11	70.002	70.16
<b>3</b>	0.098	0.11	1.001	0.99	10.005	10.04	70.001	70.20
<b>4</b>	0.099	0.12	1.001	1.09	10.007	9.91	70.001	70.24
<b>5</b>	0.098	0.13	1.001	0.94	10.007	9.95	70.002	70.47
<b>6</b>	0.098	0.08	1.000	1.11	10.006	9.97	70.002	70.26
<b>7</b>	0.099	0.10	1.000	0.87	10.006	10.04	70.001	70.07
<b>8</b>	0.099	0.10	1.001	1.09	10.007	10.04	70.002	70.19
<b>9</b>	0.098	0.09	1.001	0.90	10.005	9.95	70.001	70.16
<b>10</b>	0.099	0.09	1.001	1.14	10.006	9.57	70.001	70.20
<b>Promedio</b>	0.099	0.103	1.001	1.013	10.006	9.971	70.002	70.238
<b>E.R %</b>	1.500	3.000	0.070	1.300	0.062	0.290	0.002	0.340
<b>E.A (mm)</b>	0.001	0.003	0.001	0.013	0.006	0.029	0.002	0.238

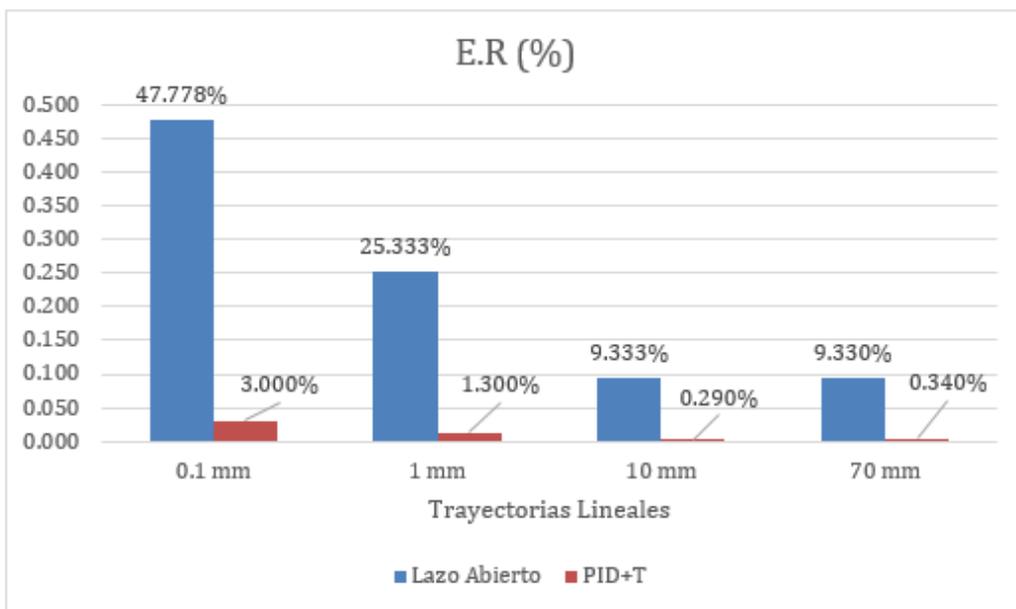
*Fuente: Elaboración Propia.*

➤ La validación de la reducción del error en las trayectorias lineales realizadas por un robot cartesiano tipo gantry, se puede simplificar tomando como referencia los resultados de seguimiento de trayectorias lineales en la articulación del sistema de control utilizado, y teniendo en cuenta los resultados del robot cartesiano tipo gantry desarrollado por Fiestas (2017). A continuación, en la figura 49 y 50 se muestran el promedio y el error relativo porcentual respectivamente, de las trayectorias realizadas tanto en lazo abierto como con el controlador desarrollado, y en la figura 51, los resultados resumidos de Fiestas (2017).



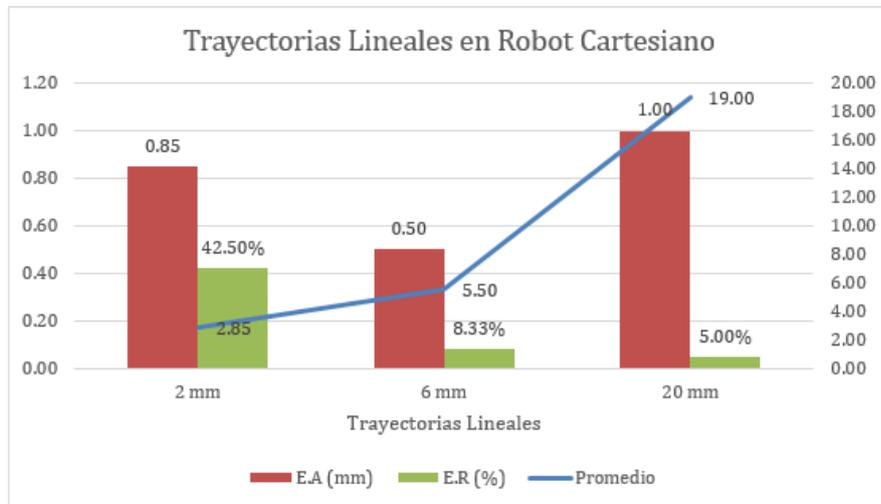
**Figura 49.** Resultados promedio del seguimiento de trayectorias lineales en lazo abierto y con el controlador desarrollado.

*Fuente: Elaboración Propia.*



**Figura 50.** Resultados del E.R. Porcentual del seguimiento de trayectorias lineales en lazo abierto y con el controlador desarrollado.

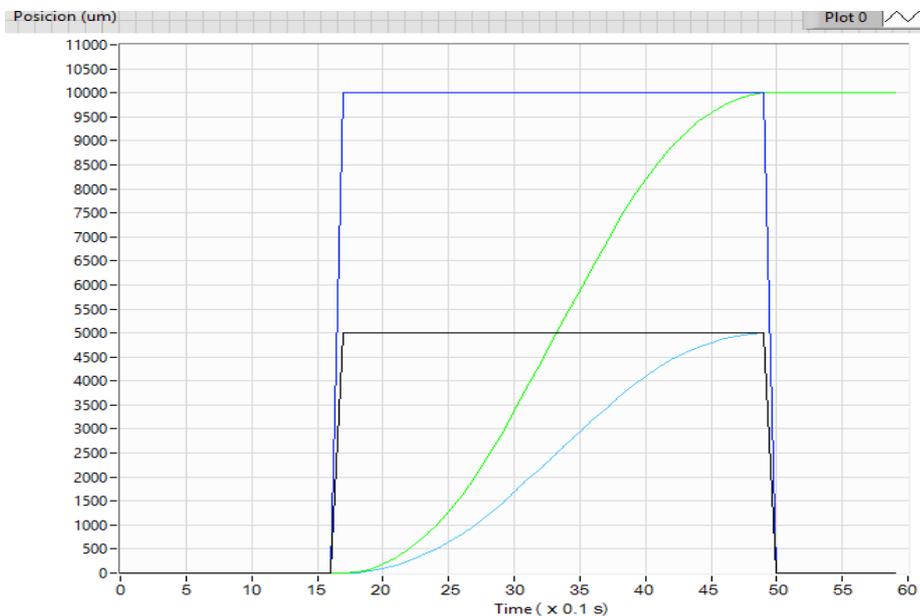
*Fuente: Elaboración Propia.*



**Figura 51.** Resultados resumidos de Fiestas (2017).

*Fuente: Elaboración Propia.*

- Con respecto a la simulación del movimiento coordinado de 2 articulaciones lineales con la técnica de “cálculo cinemático”, en ambos ejes se sigue una trayectoria con un movimiento isócrono.



**Figura 52.** Respuesta en el tiempo de la posición en el seguimiento de una trayectoria de 10 mm en el eje Y, y 5 mm en el eje X.

*Fuente: Elaboración Propia.*

# **CAPITULO V**

## 5. DISCUSIÓN DE RESULTADOS

➤ Según las técnicas descritas para la coordinación de 2 articulaciones lineales conformados por un motor DC y un husillo:

✓ La rasterización de trayectorias haciendo uso de el algoritmo de Bresenham, es una técnica adecuada para la coordinación de articulaciones en dos y tres ejes haciendo uso de desplazamientos mínimos en la articulación según Hernández (2014).

✓ El cálculo cinemático, viene a ser llamado así debido a que, Bustamante (2013) en su trabajo de investigación, establece un movimiento coordinado en un eje lineal de una mesa XY de acuerdo a los parámetros cinemáticos del otro eje.

➤ Se desarrolló un controlador con un algoritmo para la generación de un perfil de velocidad trapezoidal en función de un control PID discreto con una ecuación en diferencias proveniente de la discretización de un retenedor de orden cero en cascada con un bloque PID en tiempo continuo, ya que de esta manera se obtiene una mejor respuesta, para el seguimiento de trayectorias.

➤ Las mediciones del vernier digital en las pruebas realizadas, presentan variaciones mucho mayores que la incertidumbre de 10  $\mu\text{m}$  que este tiene, debido al juego que hace el efector final (husillo guía) al moverse en la estructura de la articulación lineal. Cabe resaltar que debido a que el vernier digital es un patrón de medidas certificado, fueron considerados solo las mediciones de este.

➤ El error en las trayectorias lineales es reducido considerablemente de acuerdo a las comparaciones que se pueden hacer en los resultados que se muestran en la figura 49, 50 y 51.

En la figura 51 el error absoluto máximo es de 1 mm que corresponde a trayectorias de 20 mm, y en la figura 49, para trayectorias de 70 mm, se tiene un error absoluto máximo de 6.531 mm en lazo abierto, mientras que, con el controlador desarrollado, es de 0.238 mm.

En la figura 51 el error relativo máximo es de 42.5% que corresponde a trayectorias de 2 mm, y en la figura 49, para trayectorias de 0.1 mm, se tiene un error relativo máximo de 47.778% en lazo abierto, mientras que, con el controlador desarrollado, es de 3%.

➤ No hubo inconvenientes con simular un movimiento coordinado de 2 articulaciones lineales, debido a que en pruebas anteriores con el controlador Trapezoide – PID, se puede observar que en la articulación se sigue una trayectoria con un perfil de velocidad

trapezoidal de acuerdo a puntos de consigna calculados en tiempo real, por lo que con el movimiento isócrono es posible asumir que se sigue una trayectoria igual a los puntos de consigna calculados en ambas articulaciones ya que el error tiende a ser nulo.

# **CAPITULO VI**

## 6. CONCLUSIONES

- Se logró describir 2 técnicas aplicables al movimiento coordinado de 2 articulaciones lineales conformados por un motor DC y un husillo: Rasterización de trayectorias (Algoritmo de Bresenham) y cálculo cinemático.
- Se consiguió desarrollar un controlador que integra un algoritmo para la generación de un perfil de velocidad trapezoidal haciendo uso del diseño de un control PID discreto y de acuerdo a las técnicas anteriormente descritas para el seguimiento de trayectorias lineales en una articulación.
- Se lograron realizar pruebas para el seguimiento de trayectorias lineales de 0.1, 1, 10 y 70 mm, con el controlador desarrollado.
- Se logró validar la reducción del error en las trayectorias lineales realizadas por un robot cartesiano tipo gantry, teniendo en consideración los resultados de Fiestas (2017), y los resultados del seguimiento de trayectorias en una articulación lineal con un control en lazo abierto y con el controlador desarrollado obteniendo un error absoluto máximo de 0.238 mm para trayectorias de 70 mm, y un error relativo máximo de 3% para trayectorias de 0.1 mm.
- Se logró simular el movimiento coordinado de 2 articulaciones lineales, como complemento para el desarrollo del controlador de seguimiento de trayectorias lineales en una articulación. Según la figura 52, la trayectoria en el eje X culmina al mismo tiempo que en el del eje Y, resultando en un movimiento isócrono y dando lugar al seguimiento de trayectorias lineales en todos los sentidos para un robot cartesiano tipo gantry.

# **CAPITULO VII**

## 7. RECOMENDACIONES

- Un controlador con el algoritmo Trapezoide – PID implementado presenta una baja acción correctiva, por lo que brinda mejores resultados en estructuras dinámicas de alta linealidad. Es posible usar ambas técnicas de coordinación de articulaciones con este controlador, sin embargo, se recomienda usar el cálculo cinemático para trayectorias netamente lineales.
- Para estructuras dinámicas de baja linealidad es recomendable optar por el uso del control PID discreto desarrollado, sin embargo, para la coordinación de articulaciones lineales solo sería posible utilizar la técnica de rasterización de trayectorias, debido a que los tiempos de llegada son variables. Sin embargo, debido a que el seguimiento de trayectorias con el control PID discreto desarrollado presenta tiempos de estabilización altos (inclusive para 100  $\mu\text{m}$ ), quedaría como próximo objetivo, optimizarlo para mejorar el tiempo de respuesta o en todo caso diseñar un controlador más robusto y apropiado para aplicaciones robóticas.
- En lo posible utilizar tiempos de muestreo bajos para el algoritmo Trapezoide – PID, ya que de esta manera se obtienen más puntos de consigna en el seguimiento de una trayectoria, es decir, mayor resolución para el control y establecimiento del tiempo de llegada.

# **CAPITULO VIII**

## 8. REFERENCIAS BIBLIOGRÁFICAS

Austin, D. (2004). *Generate stepper-motor speed profiles in real time*. Recuperado de: <https://www.embedded.com/design/mcus-processors-and-socs/4006438/Generate-stepper-motor-speed-profiles-in-real-time>.

Barrientos, A., Peñín, L. F., Balaguer, C., Aracil, R. (2007). *Fundamentos de robótica*. (2ª ed.) Madrid, España: McGraw-Hill/Interamericana de España, S.A.U.

Bustamante, D. (2013). *Implementación de un controlador embebido para controlar una mesa XY*. (Tesis para optar el título de Ingeniero Electrónico). Pontificia Universidad Católica del Perú, Lima.

Carreño, J. (2014). *Control robusto de la dinámica de un robot manipulador, considerando incertidumbre paramétrica*. (Tesis para optar el título de Magíster en Ingeniería Electrónica). Universidad Industrial de Santander, Bucaramanga.

Craig, J. (2006). *Robótica* (3ra ed.). México D.F.: Pearson Education.

Fiestas, E. (2017). *Control de los movimientos de un robot industrial utilizando un módulo FPGA DE0\_NANO\_SOC de Altera, una PC y una máquina CNC de bajo coste*. (Tesis para optar el título de Ingeniero Electrónico). Universidad Privada Antenor Orrego, Trujillo.

Garibaldi, J. (2009). *Modelado, análisis y diseño del sistema de control de un robot gantry usando un sistema operativo de tiempo real convencional*. (Tesis para optar el título de Ingeniero Informático). Universidad de Buenos Aires, Argentina.

Hernández, A. (2014). *Control de una fresadora CNC de uso didáctico*. (Tesis para optar el título de Ingeniero en Robótica Industrial). Instituto Politécnico Nacional, México D.F.

Ibrahim, D. (2006). *Microcontroller Based Applied Digital Control*. England: John Wiley & Sons, Ltd.

Kuo, B. (1996). *Sistemas de control automático* (7ma. ed.). México: Prentice Hall Inc.

Mejía, E & Narvárez G. (2016). *Diseño de un controlador PID, utilizando una Tarjeta FPGA Cyclone V GX starter, programada en VHDL*. (Tesis para optar el título de Ingeniero Electricista). Universidad de el Salvador, San Salvador.

Moreno, F. & Centeno, D (2013). *Análisis comparativo del desempeño de técnicas de control conmutado implementadas en dispositivos programables*. (Tesis para optar el título de Ingeniero Electrónico). Universidad Industrial de Santander, Bucaramanga.

Ogata, K. (2010). *Ingeniería de control moderna* (5da. ed.). Madrid: Pearson Educación.

Ogata, K. (1996). *Sistemas de control en tiempo discreto* (2da. ed.). México: Prentice Hall Inc.

STMicroelectronics (2017). *RM0090: STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs*. Recuperado de: <https://www.st.com>.

# **CAPITULO IX**

## 9. ANEXOS

### Anexo 1: Hoja de datos resumido del microcontrolador STM32F407ZG.



# STM32F405xx

# STM32F407xx

---

ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera

Datasheet - production data

#### Features

- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
  - Memories
    - Up to 1 Mbyte of Flash memory
    - Up to 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM
    - Flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
    - LCD parallel interface, 8080/6800 modes
    - Clock, reset and supply management
      - 1.8 V to 3.6 V application supply and I/Os
      - POR, PDR, PVD and BOR
      - 4-to-26 MHz crystal oscillator
      - Internal 16 MHz factory-trimmed RC (1% accuracy)
      - 32 kHz oscillator for RTC with calibration
      - Internal 32 kHz RC with calibration
    - Low-power operation
      - Sleep, Stop and Standby modes
      - V<sub>BAT</sub> supply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
    - 3×12-bit, 2.4 MSPS A/D converters: up to 24 channels and 7.2 MSPS in triple interleaved mode
    - 2×12-bit D/A converters
    - General-purpose DMA: 16-stream DMA controller with FIFOs and burst support
- 

LQFP64 (10 × 10 mm)  
LQFP100 (14 × 14 mm)  
LQFP144 (20 × 20 mm)  
LQFP176 (24 × 24 mm)



WLCSP90  
(4.223×3.969 mm)



UFBGA176  
(10 × 10 mm)
- Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 168 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
  - Debug mode
    - Serial wire debug (SWD) & JTAG interfaces
    - Cortex-M4 Embedded Trace Macrocell™
  - Up to 140 I/O ports with interrupt capability
    - Up to 136 fast I/Os up to 84 MHz
    - Up to 138 5 V-tolerant I/Os
  - Up to 15 communication interfaces
    - Up to 3 × I<sup>2</sup>C interfaces (SMBus/PMBus)
    - Up to 4 USARTs/2 UARTs (10.5 Mbit/s, ISO 7816 interface, LIN, IrDA, modem control)
    - Up to 3 SPIs (42 Mbits/s), 2 with muxed full-duplex I<sup>2</sup>S to achieve audio class accuracy via internal audio PLL or external clock
    - 2 × CAN interfaces (2.0B Active)
    - SDIO interface
  - Advanced connectivity
    - USB 2.0 full-speed device/host/OTG controller with on-chip PHY
    - USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
    - 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII

**Anexo 2:** Hoja de datos del motor NF5475E.

Model No.	at Maximum Efficiency					at Continuous Operation(at 50°C)	No Load		Starting	
	Torque [mN * m]	Speed [rpm]	Current [A]	Output [W]	Efficiency [%]	Possible Max Output [W]	Speed [rpm]	Current [A]	Torque [mN * m]	Current [A]
NA5475F	71.07	3234	1.494	24.07	67.1	33.7	3818	0.270	464.10	8.264
NF5475A	81.40	3199	1.617	27.27	70.3	38.9	3716	0.261	585.10	10.003
NF5475E	74.83	4162	1.256	32.62	68.3	38.9	4884	0.218	506.57	7.246

Encoder Characteristic						
Output Phase	Resolution [P/R]	Supply Voltage [V]	Current [mA]	Frequency Response [kHz]	Encoder Type	Signal Format
2	100	5	MAX 50	10	Optical	Incremental
2	200	5	MAX 50	20	Optical	Incremental

**Anexo 3:** Especificaciones técnicas del vernier digital.

● ESPECIFICACIONES TÉCNICAS	
Rango de medición:	0-150mm/0-6", 0-200mm/0-8", 0-300mm/0-12".
Resolución:	0.01mm/0.0005"
Precisión:	±0.02mm/0.001"(<100mm) ±0.03mm/0.001"(>100mm-200mm) ±0.04mm/0.0015"(>200mm-300mm)
Repeticidad:	0.01mm/0.0005"
Velocidad máxima de medición:	1.5m/sec, 60"/sec.
Sistema de medición:	Sistema de medición lineal capacitiva.
Pantalla:	Pantalla de Cristal Líquido.
Poder:	Una batería de óxido de plata SR44, 1.55v capacidad:165mAh
Temperatura de funcionamiento:	5°C-40°C/41 to 104 degree F.
Influencia de la humedad:	No importa si está bajo 80% de la humedad relativa.

**Anexo 4:** Descripción y lista de tipos de variables enumeradores utilizados en el programa.

Variable Enumeradora	Miembro	Descripción
RAMPA_e Contiene valores enumerados como indicadores para guardar los tiempos que sigue el algoritmo de control en el seguimiento de trayectorias.	NONE	Indicador de motor detenido.
	ACEL	Indicador de rampa de aceleración en el trapecio.
	VEL_CTE	Indicador de aceleración igual a 0 en el trapecio.
	DESACEL	Indicador de rampa de desaceleración en el trapecio.
DIR_e Contiene valores enumerados de referencia para el seguimiento de la trayectoria.	BRAKE	Indica que no hay trayectorias por seguir.
	LEFT	Indica que la trayectoria a seguir se encuentra a la izquierda.
	RIGHT	Indica que la trayectoria a seguir se encuentra a la derecha.

**Anexo 5:** Descripción y lista de tipos de variable estructural utilizados en el programa.

Tipo de Variable Estructural	Miembro	Contenido
<i>T_Trayectoria</i> Contiene variables correspondientes al seguimiento de una trayectoria en un eje lineal.	T_PID PID	Variables para ecuación PID.
	T_Trapezoide Tz	Variables para el Perfil de velocidad trapezoidal.
	DIR_e Dir[NUM_PUNTOS]	Direcciones de las trayectorias.
	Unsigned int Sp[NUM_PUNTOS-1]	Tramos o trayectorias a seguir.
	Unsigned int Coor[NUM_PUNTOS]	Coordenadas ingresadas para el seguimiento.
	Unsigned int Puntos	Número de Coordenadas.

	long double Pos	Variable de proceso: Posición lineal.
	long double Vel	Variable de proceso: Velocidad lineal.
	long double Acc	Acumulador de trayectorias realizadas.
<i>T_Trapezoide</i> Contiene variables para el perfil de velocidad trapezoidal.	RAMPA_e Rampa	Indicadores para guardar los tiempos del perfil trapezoidal.
	long double tf[NUM_PUNTOS]	Tiempos finales que deben seguir las trayectorias.
	long double tc[NUM_PUNTOS]	Tiempos críticos que deben seguir las trayectorias.
	long double Vmax[NUM_PUNTOS]	Velocidades lineales máximas que deben seguir las trayectorias.
	long double acel[NUM_PUNTOS]	Aceleraciones que deben seguir las trayectorias.
	long double Sp_Vel	Velocidad lineal de consigna en tiempo real
	long double Sp_Pos	Posición lineal de consigna en tiempo real
<i>T_PID</i> Contiene variables	long double kp	Término Proporcional
	long double kd	Término Derivativo
	long double ki	Término Integral
	long double b0	Coefficiente 0
	long double b1	Coefficiente 1
	long double b2	Coefficiente 2
	long double e0	Error de posición
	long double e1	Error de posición una muestra anterior

involucradas en la ecuación PID.	long double e2	Error de posición dos muestras anteriores
	long double y0	Resultado de la ecuación PID
	long double y1	Resultado de la ecuación PID una muestra anterior
<i>Timer</i> Contiene variables involucradas en la temporización de eventos con la base de tiempos SysTick	unsigned int startTime	Tiempo de inicio
	unsigned int interval	Intervalo de tiempo a temporizar
	bool en	Habilitador del temporizador

**Anexo 6:** Descripción y lista de variables principales en el desarrollo de los programas que fueron utilizados.

Tipo de Variable	Variable	Función
T_Trayectoria	Eje	Presentar las variables involucradas en una articulación del robot cartesiano tipo gantry.
Volatile unsigned int*	DC	Referenciar el registro CCR1 del Timer 13
volatile unsigned int*	CUENTAS	Referenciar el registro CNT del Timer 2
volatile unsigned int*	PERIODO	Referenciar el registro CCR1 del Timer 5
Timer	Led2	Temporizador para graficar las variables de la trayectoria por UART.
Timer	ITM_Port_Tmr	Temporizador para mostrar información de las variables de la trayectoria por el puerto ITM.
Timer	Stop_Tmr	Temporizador para detener el motor.

long double	Time_4_Stop	Tiempo que requiere Stop_Tmr.
long double	Ts	Referenciar el tiempo de muestreo usado en el controlador.
Unsigned int	k	Indexar las variables involucradas en el seguimiento de acuerdo al número de puntos.
long double	c	Acumular tiempos de muestreo
long double	Coordenada [32]	Almacenar cada punto final de las trayectorias seguidas en la articulación.
long double	Tim [32]	Almacenar cada tiempo transcurrido por cada trayectoria seguida en la articulación.
long double	tf[32]	Almacenar cada tiempo final transcurrido según el perfil de velocidad por cada trayectoria seguida en la articulación.
long double	tf_tc[32]	Almacenar cada tiempo transcurrido, hasta antes de la desaceleración según el perfil de velocidad, de las trayectorias seguidas en la articulación.
long double	tc[32]	Almacenar cada tiempo crítico inicial transcurrido según el perfil de velocidad, de las trayectorias seguidas en la articulación.

**Anexo 7:** Instrumento Virtual de Labview 2018 para leer datos de posición y velocidad lineal en la articulación lineal.

