

Marco Antonio Cabrera Rufino

2020



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Posicionamiento de un robot de 3  
grados de libertad con actuadores  
neumáticos

Que como parte de los requisitos para  
obtener el Grado de

Maestro en Ciencias en Control e  
instrumentación Automático

Presenta

Marco Antonio Cabrera Rufino

Dirigido por:

Dr. Juan Manuel Ramos Arregín

Querétaro, Qro. a 31/12/2020



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Instrumentación y

Control Automático

**POSICIONAMIENTO DE UN ROBOT DE 3 GRADOS DE LIBERTAD CON  
ACTUADORES NEUMÁTICOS**

TESIS

Que como parte de los requisitos para obtener el grado de  
Maestro en Ciencias en Instrumentación y Control Automático

Presenta:

**Marco Antonio Cabrera Rufino**

Dirigido por:

**Dr. Juan Manuel Ramos Arreguín**

SINODALES

Dr. Juan Manuel Ramos Arreguín

Presidente

Firma

Dr. Marco Antonio Aceves Fernández

Secretario

Firma

Dr. Juvenal Rodrigues Reséndiz

Vocal

Firma

Dr. Efren Gorrostieta Hurtado

Sinodal

Firma

Dr. Jesús Carlos Pedraza Ortega

Sinodal

Firma

Centro Universitario

Querétaro, QRO

México.

Diciembre 2020

Dirección General de Bibliotecas de la UAQ

© 2020 - Marco Antonio Cabrera Rufino

All rights reserved.

Dirección General de Bibliotecas de la UAQ

Dirección General de Bibliotecas de la UAQ

*Este trabajo esta dedicado principalmente a mis padres: **Adelina Rufino** y **Marcelino Cabrera**; por su amor y soporte siempre constante desde el momento en que nací, gracias a ellos por impulsar mi espíritu a cada día aprender algo nuevo. A mis hermanos, por su cariño y aliento, al Dr. Arreguín que desde que ingrese a esta gran institución me ha brindado un inmenso apoyo y a todos mis maestros y compañeros de licenciatura y maestría, por las risas, la alegría y la enseñanza que todos ellos me brindaron.*

Dirección General de Bibliotecas de la UAQ

# Abstract

Interest in the use of pneumatic actuators has become increasingly significant, especially because of the potential benefits of employing clean energies. There are previous works related to this project; some made abroad that innovates in control methods but only in specific applications, and others in Mexico that progressively increase the complexity of the system. In this paper the difficulty of a generic three-degrees-of-freedom (DoF) robot arm control is tested, searching for the most efficient method to reproduce the positioning of the final effector (a griper) accurately. Furthermore, the ambitious goal of the project is to adequately develop an algorithm capable of being properly adjusted to any multipurpose robot system with pneumatic actuators. It is carefully considered the various non-linearities of the system. This work relied mostly on intelligent control techniques and modern programmable devices, the intention is to achieve similar or better results than previous projects. The possible combination of intelligent and even classical control can reasonably achieve the estimated conclusion, the first being flexible for developing robust design for non-linear systems.

Dirección General de Bibliotecas de la UAQ

# Resumen

El interés por el uso de actuadores neumáticos se ha vuelto cada vez más importante, especialmente debido a los beneficios de emplear energías limpias. Existen relacionados con este artículo, algunos hechos fuera del país donde se innova en métodos de control, pero solo en aplicaciones específicas, y otros elaborados dentro del país donde se incrementa la complejidad del sistema y control. En este trabajo se prueba la dificultad de un brazo robótico genérico de tres grados de libertad (DoF), se busca el método más eficiente de reproducir el posicionamiento del efector final (una pinza) de una forma precisa. Además, la meta de este proyecto es de crear un algoritmo capaz de ser ajustado a cualquier sistema robótico multipropósito con actuadores neumáticos de tres grados de libertad. Se pretende obtener resultados similares o mejores que proyectos anteriores, los resultados preliminares muestran que la combinación de control neuronal, difuso, o incluso clásico puede obtener los resultados estimados, siendo los dos primeros muy flexibles para desarrollar un diseño original y efectivo de control.

Dirección General de Bibliotecas de la UAQ

# Contenido

<b>Abstract</b>	i
<b>Resumen</b>	iii
<b>Contents</b>	v
<b>List of Figures</b>	vii
<b>List of Tables</b>	ix
<b>1 Introducción</b>	<b>1</b>
1.1 Justificación	2
1.2 Descripción del Problema	2
1.3 Hipótesis	4
1.4 Objetivos	4
1.4.1 Objetivo General	4
1.4.2 Objetivos específicos	4
1.5 Estructura de la Tesis	4
<b>2 Investigación Literaria</b>	<b>7</b>
2.1 Marco Teórico	7
2.1.1 Cinemática Inversa para un robot de 3 gdl	7
2.1.2 Controlador PID digital	10
2.1.3 Control Difuso	12
2.1.4 Red Neuronal	14
2.1.5 Red ANFIS	16
2.2 Antecedentes	18
2.3 Estado del Arte	20
<b>3 Metodología</b>	<b>21</b>
3.1 Herramientas de hardware y software	21
3.1.1 Configuración del encoder	24
3.1.2 Configuración de la comunicación UAT	25
3.1.3 Configuración del controlador PID	27
3.2 Control del flujo de aire	27

3.2.1	Diseño del controlador del motor	28
3.3	Control inteligente	29
3.3.1	PI Difuso	29
3.3.2	<i>PID</i> con red neuronal para ajustes de ganancias	30
<b>4</b>	<b>Resultados y discusión</b>	<b>33</b>
4.1	Resultados del Motor DC	33
4.2	Resultados de simulaciones de controladores difusos	33
4.3	Resultados de un grado de libertad	34
4.4	Resultados con tres grados de libertad	35
4.5	Trabajos a futuro	35
<b>5</b>	<b>Conclusión</b>	<b>39</b>
	<b>Referencias Bibliográficas</b>	<b>43</b>
	<b>Appendix A Publicaciones</b>	<b>45</b>

# Lista de Figuras

1.1	Importación anual estimada (numero en unidades) de robots industriales multi-propósito Mexico (verde) y China (rojo) (IFR, 2018).	3
1.2	Robot utilizado en el proyecto.	3
2.1	Relación entre la cinemática directa e inversa (Delgado <i>et al.</i> , 2018).	7
2.2	Diagrama de un robot articulado con 3 grados de libertad (Ramos-Arreguin <i>et al.</i> , 2008; Sánchez Solar <i>et al.</i> , 2016).	8
2.3	Perspectiva en el plano XY (Ramos-Arreguin <i>et al.</i> , 2008; Sánchez Solar <i>et al.</i> , 2016).	9
2.4	Forma simplificada para el análisis Cinemático (Ramos-Arreguin <i>et al.</i> , 2008; Sánchez Solar <i>et al.</i> , 2016).	9
2.5	Forma modificada para el análisis trigonométrico (Ramos-Arreguin <i>et al.</i> , 2008; Sánchez Solar <i>et al.</i> , 2016).	10
2.6	Angulos complementarios (Ramos-Arreguin <i>et al.</i> , 2008; Sánchez Solar <i>et al.</i> , 2016).	10
2.7	Diagrama a bloques de un sistema de control.	11
2.8	Grafica del método rectangular directo (FRM) (Bobál <i>et al.</i> , 2006).	12
2.9	Diagrama a bloques de un controlador difuso (Passino <i>et al.</i> , 1998).	13
2.10	Particiones difusas con distinto número de términos negativo (N), cero (Z) y positivo (P). (Cruz, 2011).	13
2.11	Ejemplo de una red neuronal multicapa (Nguyen <i>et al.</i> , 2002; Cruz, 2011).	15
2.12	Funciones de activación (Cruz, 2011).	16
2.13	Modelo Sugeno de primer orden con dos reglas.	16
2.14	Diagrama a bloques de un sistema ANFIS (Cruz, 2011; Nguyen <i>et al.</i> , 2002).	17
3.1	Diagrama de bloques del proyecto.	21
3.2	Bosquejo del hardware completo para el proyecto.	22
3.3	Diagrama a bloques de la relación de software de la primera etapa.	23
3.4	Diagrama a bloques de la relación de software de la segunda etapa.	24
3.5	Fotografía de el hardware de control del robot.	24
3.6	Diagrama a bloques de los módulos VHDL del encoder.	25
3.7	Diagrama a bloques de los módulos VHDL para la transmisión serial.	26
3.8	Maquina de estados para el modulo de comunicación.	26
3.9	Diagrama a bloques del código VHDL para el controlador PID.	27
3.10	Diagrama de 1 gdl.	28
3.11	Base para el motor y la válvula reguladora de aire.	28

3.12 Sistema de control propuesto para los motores DC. . . . .	28
3.13 Sistema de control completo. . . . .	29
3.14 Sistema de control para un grado de libertad. . . . .	29
3.15 Diseño de los conjuntos de membresía de un controlador PI difuso. . . . .	30
3.16 PI difuso diseñado en Simulink. . . . .	31
3.17 Red neuronal para el ajuste de ganancias (Gorrostieta y Vargas Soto, 2008). . . . .	31
4.1 Simulaciones de la velocidad y posición de los motores DC. . . . .	33
4.2 Simulaciones de la velocidad y posición de los motores DC. . . . .	34
4.3 Simulaciones de un controlador PI difuso en: . . . . .	34
4.4 Resultado experimental 1 de un controlador <i>PID</i> en un gdl. . . . .	34
4.5 Resultado experimental 2 de un controlador <i>PID</i> en un gdl. . . . .	35
4.6 Resultado experimental 3 de un controlador <i>PID</i> en un gdl. . . . .	35
4.7 Graficas de los datos mas importantes del posicionamiento de un grado de libertad con ganancias ajustables. . . . .	36
4.8 Ubicación de los gdl en el robot. . . . .	37
4.9 Resultados del desplazamiento del robot con 3 gdl. . . . .	37
A.1 Publicación en <i>MECAMEX</i> . . . . .	45
A.2 Publicación en <i>CICOS</i> . . . . .	46

# Lista de Tablas

3.1 Matriz asociativa difusa para un controlador PI difuso, N = Negativo, Z = cero, P = Positivo Cruz (2011) . . . . .	30
--	----

Dirección General de Bibliotecas de la UAQ

Dirección General de Bibliotecas de la UAQ

---

# Introducción

A pesar de las abundantes marcas robots industriales y comerciales (ABB, FANUC, Mitsubishi, Kuka, etc.), el diseño de control para estos sigue siendo un área de alto interés por parte de las empresas y centros de investigación. Se puede argumentar sin presunción que los robots actuales son capaces de realizar correctamente una gran cantidad de tareas por lo que parecería inútil, en primera instancia, el desarrollo de investigaciones sobre el tema, sin embargo, su estudio no solo es importante, sino que ofrece grandes retos teóricos y prácticos. El estudio de este tema es indispensable en tareas específicas que no pueden ser llevadas a cabo por robots de la actualidad (Kelly y Santibáñez, 2003).

Algunos prospectos que la Federación Internacional de Robótica considera importantes en el 2018 y que son de nuestro interés se mencionan a continuación (IFR, 2018):

- La eficiencia energética y el uso de nuevos materiales requieren una actualización continua en la producción de robots.
- La entrega rápida de productos a precios competitivos son los principales incentivos para la automatización.
- La digitalización de la producción (Industria 4.0), que vincula la fábrica de la vida real con la realidad virtual, seguirá desempeñando un papel cada vez más importante en la fabricación mundial.
- Los robots trabajan las 24 horas con un estándar de calidad constante y realizan tareas peligrosas, de alto riesgo a la salud y de tareas aburridas, mejorando la seguridad, la salud y la satisfacción laboral de los trabajadores.
- Robótica en la nube: el almacenamiento de datos de múltiples robots que realizan el mismo proceso en la nube proporciona un almacenamiento de datos en el que aplicar "machine learning" para optimizar el rendimiento de los robots.

En el 2019, La reciente desaceleración de las ventas en un 12% refleja los tiempos difíciles que han experimentado debido a la pandemia, en especial en china donde se identifico primero el virus; pero puede tomar hasta 2022 o 2023 para alcanzar el nivel anterior a la crisis. En China, el 71% de los nuevos robots provienen de proveedores extranjeros. Los fabricantes chinos todavía atienden

principalmente al mercado nacional, donde obtienen cuotas de mercado cada vez mayores. Los proveedores extranjeros entregan alrededor del 29% de sus unidades a la industria automotriz, mientras que solo es alrededor del 12% para los proveedores chinos. Por lo tanto, los proveedores extranjeros se ven más afectados por la caída del negocio en la industria automotriz china que los proveedores nacionales (IFR, 2020).

## 1.1 Justificación

El interés en el uso de actuadores neumáticos ha cobrado cada vez mayor relevancia, especialmente por el interés del uso de energías limpias, así como de un mejor aprovechamiento en el uso de energía. Los actuadores eléctricos son fáciles de controlar, pero suelen ser caros, requieren de un mantenimiento costoso y se pueden dañar fácilmente. Los actuadores neumáticos, al tener como medio de generación de fuerza el aire, resulta en un modelo matemático altamente no lineal este problema es debido a la compresibilidad del aire, la fricción en las articulaciones, la no linealidad del flujo de aire en las válvulas y el retardo debido a la velocidad de propagación de la presión del aire. Este comportamiento tiene como consecuencia que el control de estos actuadores sea complejo, en comparación con los actuadores eléctricos. Los actuadores neumáticos son limpios y pueden ideales para aplicaciones en la industria alimentaria y en aplicaciones médicas (Sandler, 1999; Klafter *et al.*, 1989; Miramontes y Soto, 2003).

Este proyecto aporta al estado del arte la técnica de posicionamiento del robot manipulador, controlando actuadores neumáticos. El robot cuenta con 3 grados de libertad, lo que complica la manera de controlarlo, debido al comportamiento natural de los actuadores, por lo que se debe contemplar los resultados en el funcionamiento de cada actuador, con dependencia en el comportamiento del resto de actuadores.

## 1.2 Descripción del Problema

Desde la década de los 80 la robótica industrial despegó enormemente debido a la investigación por parte de las empresas fabricantes de robots industriales, ya en el año 2000 el parque industrial de robots rondaba más de 700,000 unidades (Kelly y Santibáñez, 2003).

Según la UNESCO las empresas extranjeras que fabrican robots industriales como de Dinamarca, Japón o Estados Unidos voltean a ver a nuestro país como un gran mercado atractivo para sus productos, sin embargo, México importa mucho del exterior, pero solo invierte menos del 1% del PIB en desarrollar su propia tecnología, nuestro país es el que más robots industriales importa en Latinoamérica, y el quinto en todo el mundo (González Islas, 2019).

En el 2017, las importaciones alcanzaron un valor de 209 millones de dólares. Se puede decir que México es un país que consume e importa su totalidad en robots industriales, pero no parece estar interesado en crear su propia tecnología. Para contrastar toda esta información se compara a México con China, el país número uno en mercado mundial (recordemos que México ocupa el quinto), el cual representa el 30% del mercado global (González Islas, 2019; IFR, 2018).

Esta información se puede contrastar con la figura 1.1 donde se aprecia el incremento de unidades de robots transportados en los últimos cuatro años, donde se prevé un incremento considerable en la exportación de robots para el 2021 (IFR, 2018).

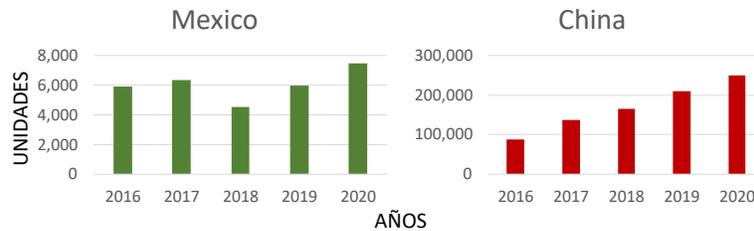


Figura 1.1: Importación anual estimada (numero en unidades) de robots industriales multipropósito Mexico (verde) y China (rojo) (IFR, 2018).

Por otro lado, China presenta en un artículo una agenda ambiciosa con el nombre “Made In China 2025” donde pretende en el año 2025 tener 70% de los robots industriales hechos nacionalmente, además, en el 2020 pretende tener la mitad del porcentaje; lo cual representa 1.8 millones de unidades para el 2025. También, el gobierno chino ofreció 498 billones de yuanes (alrededor de 132 billones de dólares) en subsidios entre 2015 y 2018 para ayudar a automatizar los manufactureros locales (Hui Feng y Chen, 2018).

En México aún no se ha desarrollado mucho sobre el tema de actuadores neumáticos, siendo difícil encontrar trabajos recientes dentro del país. En la Universidad Autónoma de Querétaro se ha estado trabajando en el desarrollo de robots neumáticos (Ramos-Arreguin *et al.*, 2008). Actualmente se cuenta con dos robots neumáticos en el Laboratorio de Mecatrónica. En estos robots, aún es necesario trabajar en el control de posición del efector final, considerando 3 grados de libertad, en la figura 1.2 se muestra el robot con el que se trabajó.



Figura 1.2: Robot utilizado en el proyecto.

No contamos con un parámetro previo, para establecer el nivel de precisión, pero se espera que la precisión alcanzada con el robot pueda estar en el orden de milímetros. No se cuenta con un

modelo matemático exacto, por lo que se va a plantear la manera de poder realizar simulaciones, y se está considerando el uso de técnicas de control inteligente para el control.

### 1.3 Hipótesis

Es posible obtener un buen desempeño al implementar un algoritmo de control inteligente para el comportamiento específico del movimiento de actuadores neumáticos y poner el efector final en la posición deseada, haciendo el ajuste preciso y replicable para un robot de 3 grados de libertad, embebido en FPGA.

### 1.4 Objetivos

#### 1.4.1 Objetivo General

Implementar un algoritmo para controlar el comportamiento de actuadores neumáticos y generar movimiento en un robot, y posicionar el efector final en una posición deseada.

#### 1.4.2 Objetivos específicos

Los objetivos específicos son los siguientes:

- Desarrollar e implementar un sistema de control para el flujo de aire en el robot.
- Definir la herramientas de software y hardware que se utilizaran en el proyecto.
- Definir el controlador inteligente (Neuronal, difuso, clásico o una combinación de los anteriores).
- Desarrollar scripts que faciliten la generación automática de código VHDL.
- Validar herramientas y controlador implementados en el proyecto.
- Implementar algoritmo para los 3 grados de libertad.

### 1.5 Estructura de la Tesis

La tesis esta organizada de la siguiente manera:

- En el capítulo 2 se encuentra el marco teórico, dando un desarrollo amplio en la cinemática inversa de manipuladores y en varias técnicas de control inteligente; además se incluye antecedentes y marco teórico, el primero comprende todos los trabajos relacionados al tema desde el año 1984 al 2019; como estado del arte se tienen los trabajos publicados en el 2020.
- En el capítulo 3 se desarrolla a detalle la metodología usada en este proyecto; este se desarrolla primero explicando las herramientas de hardware y software utilizados y como se implementaron las técnicas de control propuestas.

- En el capítulo 4 se muestran los resultados que se obtuvieron en cada técnica de control implementada. Como primer lugar los resultados del posicionamiento y velocidad del robot, los resultados obtenidos al ajustar los parámetros del PID, y los resultados finales de 3 grados de libertad.

Dirección General de Bibliotecas de la UAQ

Dirección General de Bibliotecas de la UAQ

# Investigación Literaria

## 2.1 Marco Teórico

Como se enseña a lo largo de la maestría, en esta parte se establecen las "armas" con las que se va a "atacar" en este proyecto, esto quiere decir que, en el marco teórico se establece las teorías matemáticas que sirve como herramientas para el desarrollo del proyecto.

### 2.1.1 Cinemática Inversa para un robot de 3 gdl

La cinemática directa e inversa ofrece un análisis matemático para robots con manipuladores, el primero es muy útil para simulación (el problema más sencillo), en el caso de tres grados de libertad representa el cálculo de la posición del efector final da través de los ángulos conocidos de las articulaciones en este trabajo no se enfoca en realizar simulaciones, pero se utilizan las realizadas en proyectos previos. La cinemática inversa es útil para el diseño del controlador, calcula los ángulos de las articulaciones a través de la posición del efector final, la solución depende enormemente del tipo de manipuladores a utilizar (en este caso neumáticos). Frecuentemente resulta en varias soluciones para los ángulos de las articulaciones para la posición final, la relación entre la cinemática directa e inversa se encuentra en la figura 2.1 (Kelly y Santibáñez, 2003).

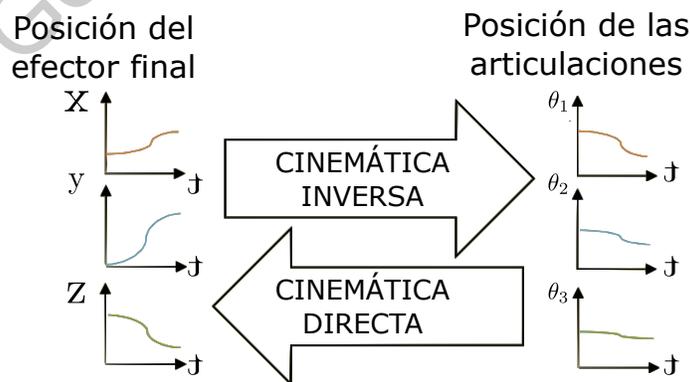


Figura 2.1: Relación entre la cinemática directa e inversa (Delgado *et al.*, 2018).

Un robot de tres grados de libertad se muestra en la figura 2.2 y su cinemática inversa esta dada en las ecuaciones, donde (Sánchez Solar *et al.*, 2016):

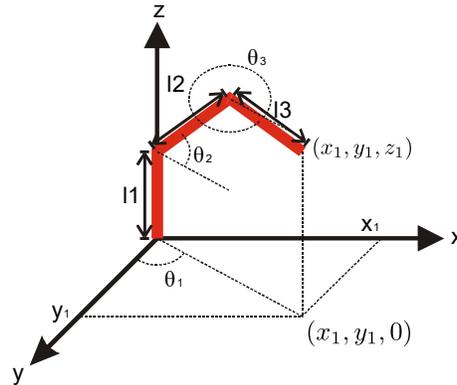


Figura 2.2: Diagrama de un robot articulado con 3 grados de libertad (Ramos-Arreguin *et al.*, 2008; Sánchez Solar *et al.*, 2016).

- $l_1, l_2, l_3 \rightarrow$  son las longitudes de las articulaciones del robot.
- $\theta_1 \rightarrow$  es el ángulo de la articulación  $l_1$  con respecto al eje Y.
- $\theta_2 \rightarrow$  es el ángulo de la articulación  $l_2$  con respecto al plano XY.
- $\theta_3 \rightarrow$  es el ángulo de la articulación  $l_3$  con respecto al plano XY.

La posición final del extremo del robot está definida por la ecuación (2.1)

$$\begin{aligned} x_1 &= [L_2 \cos(\theta_2) + L_3 \cos(\theta_3)] \operatorname{sen}(\theta_1) \\ y_1 &= [L_2 \cos(\theta_2) + L_3 \cos(\theta_3)] \operatorname{cos}(\theta_1) \\ z_1 &= L_1 + L_2 \operatorname{sen}(\theta_2) + L_3 \operatorname{sen}(\theta_3) \end{aligned} \quad (2.1)$$

De acuerdo a la figura 2.2, se inicia el análisis en el plano XY, obteniendo el valor de  $\theta_1$ , como se muestra en la ecuación (2.2).

$$\theta_1 = \tan^{-1} \left( \frac{y_1}{x_1} \right) \quad (2.2)$$

Ahora, el análisis se limita exclusivamente al plano XY, utilizando la figura 2.3.

Para simplificar el análisis, se traslada el origen  $l_1$  unidades hacia arriba, anulando el primer eslabón, por lo que el análisis se simplifica. El resultado de trasladar el origen, se muestra en la figura 2.4.

Para realizar el análisis, se toma en cuenta la figura 2.5, y se muestran las ecuaciones resultantes.

$$h = \sqrt{x^2 + z'^2} \quad (2.3)$$

$$\alpha = \operatorname{sen}^{-1} \left( \frac{z'}{\sqrt{x^2 + z'^2}} \right) \quad (2.4)$$

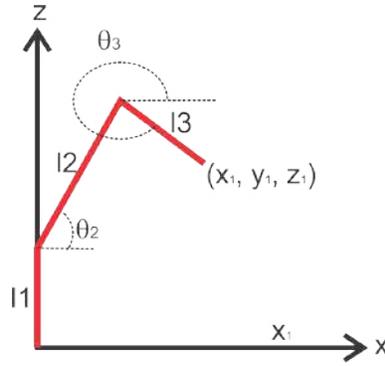


Figura 2.3: Perspectiva en el plano XY (Ramos-Arreguin *et al.*, 2008; Sánchez Solar *et al.*, 2016).

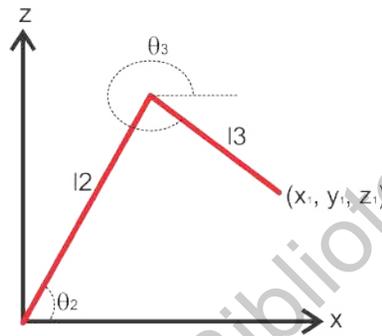


Figura 2.4: Forma simplificada para el análisis Cinemático (Ramos-Arreguin *et al.*, 2008; Sánchez Solar *et al.*, 2016).

$$l_3^2 = l_2^2 + h^2 - 2l_3h\cos(\beta) \quad (2.5)$$

$$\beta = \theta_2 - \alpha \quad (2.6)$$

$$\cos(\theta_2 - \alpha) = \frac{l_2^2 + h^2 - l_3^2}{2l_2h} \quad (2.7)$$

$$\theta_2 = \cos^{-1} \left( \frac{l_2^2 + h^2 - l_3^2}{2l_2h} \right) + \alpha \quad (2.8)$$

Aplicando la ley de cosenos, se obtiene  $h^2 = l_3^2 + l_2^2 - 2l_3l_2\cos(\gamma)$ , despejando  $\gamma$  se obtiene  $\gamma = \cos^{-1} \left( \frac{l_3^2 + l_2^2 - h^2}{2l_3l_2} \right)$ , De aquí,  $\gamma + \alpha' = 180^\circ$ ; y despejando  $\gamma'$ , obtenemos  $\alpha' = 180^\circ - \gamma$ . Finalmente, encontramos el valor  $\theta_3$ , que resulta en  $\theta_3 = \theta_2 - \alpha'$ . Sustituyendo el valor de  $\alpha'$  se obtiene  $\theta_3 = \theta_2 - 180^\circ + \gamma'$ . Dado que ya conoces el valor de  $\gamma$ , se obtiene, finalmente la ecuación (2.9).

$$\theta_3 = \theta_2 - 180^\circ + \cos^{-1} \left( \frac{l_3^2 - h^2 + l_2^2}{2l_3l_2} \right) \quad (2.9)$$

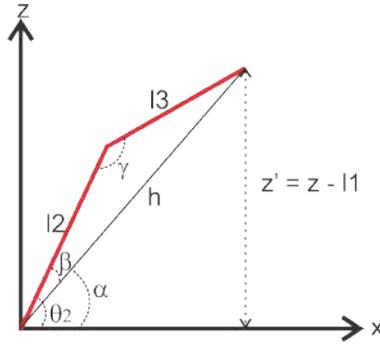


Figura 2.5: Forma modificada para el análisis trigonométrico (Ramos-Arreguin *et al.*, 2008; Sánchez Solar *et al.*, 2016).

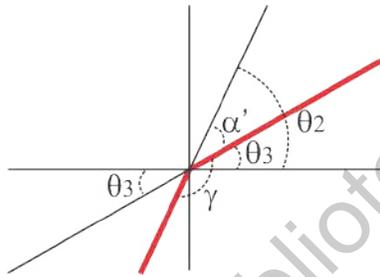


Figura 2.6: Ángulos complementarios (Ramos-Arreguin *et al.*, 2008; Sánchez Solar *et al.*, 2016).

En conclusión, las 3 ecuaciones características del movimiento cinemático son:

$$\begin{aligned}\theta_1 &= \tan^{-1} \left( \frac{y_1}{x_1} \right) \\ \theta_2 &= \cos^{-1} \left( \frac{x^2 + (z - l_1)^2 + l_2^2 - l_3^2}{2l_2 \sqrt{x^2 + (z - l_1)^2}} \right) + \text{sen}^{-1} \frac{z - l_1}{\sqrt{x^2 + (z - l_1)^2}} \\ \theta_3 &= \theta_2 + \cos^{-1} \left( \frac{l_3^2 + l_2^2 - x^2 - (z - l_1)^2}{2l_3 l_2} \right) - 180^\circ\end{aligned}\quad (2.10)$$

### 2.1.2 Controlador PID digital

La forma continua de un controlador *PID* en el dominio del tiempo está dada en la ecuación (2.11), su desarrollo y análisis está descrito en una gran cantidad de libros de control (Dorf y Bishop, 2005; Ogata, 2005; Bobál *et al.*, 2006).

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (2.11)$$

La variable  $u(t)$  expresa la salida del controlador, la variable de control  $e(t)$  representa el error de seguimiento, además  $e(t) = y(t) - w(t)$ , donde  $y(t)$  representa la salida del sistema y  $w(t)$  es la señal

de referencia. Un sistema de control con los parámetros mencionados se encuentra en la figura 2.7. Los parámetros del controlador  $PID$  son los siguientes: la ganancia proporcional  $K_p$ , la constante del tiempo integral  $T_i$ , la constante de tiempo derivativa  $T_d$ .

Usando la transformada de Laplace es posible transformar la ecuación (2.11) a la forma:

$$U(s) = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right] E(s) \quad (2.12)$$

Donde  $s$  representa el operador de la transformada de Laplace. De la ecuación (2.12) se puede determinar la función de transferencia del controlador  $PID$ .

$$G(s) = \frac{U(s)}{E(s)} = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right] \quad (2.13)$$

Para obtener la versión digital de un controlador  $PID$  continuo en el tiempo, se debe discretizar los componentes derivativo e integral de la ecuación (2.11). Cuando el periodo de muestreo  $T_0$  es demasiado pequeño y el ruido de la señal de salida del proceso es filtrada efectivamente, el algoritmo más simple es remplazando la derivada con una diferencia de primer orden.

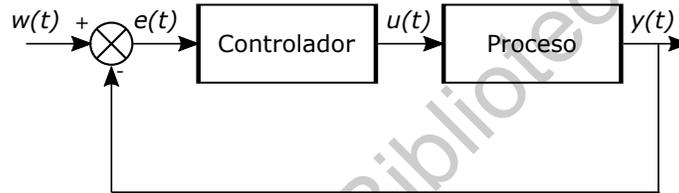


Figura 2.7: Diagrama a bloques de un sistema de control.

$$\frac{de}{dt} \approx \frac{e(k) - e(k-1)}{T_0} = \frac{\Delta e(k)}{T_0} \quad (2.14)$$

Donde  $e(k)$  es el error a la  $k$ -ésimo momento del muestreo, por ejemplo en el tiempo  $t = kT_0$ . La forma más fácil de aproximar la integral es sumando, por lo tanto, se aproxima la función de tiempo continuo haciendo muestras de los periodos  $T_0$  de la función constante (función escalón, rectangular). Usando el llamado método rectangular directo (ver la figura 2.8) se obtiene:

$$\int_0^t e(\tau) d\tau \approx T_0 \sum_{i=1}^k e(i-1) \quad (2.15)$$

Así que la ecuación del controlador  $PID$  digital esta dado por (Bobál *et al.*, 2006; Åström y Hägglund, 2009):

$$u(k) = K_p \left\{ e(k) + \frac{T_0}{T_i} \sum_{i=1}^k e(i-1) + \frac{T_d}{T_0} [e(k) - e(k-1)] \right\} \quad (2.16)$$

Para facilitar la programación en un dispositivo programable se hace una simplificación de la ecuación (2.16) descrita en (Bobál *et al.*, 2006), el resultado es el siguiente:

$$u(k) = q_1 e(k) + q_2 e(k-1) + q_3 e(k-2) + u(k-1) \quad (2.17)$$

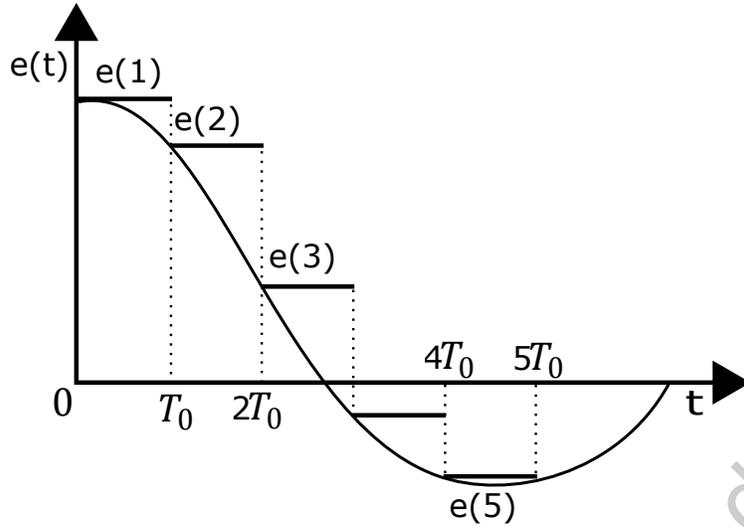


Figura 2.8: Grafica del método rectangular directo (FRM) (Bobál *et al.*, 2006).

donde:

$$\begin{aligned}
 q_1 &= K_p \left( 1 + \frac{T_d}{T_0} \right) \\
 q_2 &= -K_p \left( 1 - \frac{T_d}{T_i} + \frac{2T_d}{T_0} \right) \\
 q_3 &= K_p \frac{T_d}{T_0}
 \end{aligned} \tag{2.18}$$

### 2.1.3 Control Difuso

El diseño de un PID se basa en conocer el modelo matemático del sistema de proceso, cuando no se cuenta con este si existe el suficiente conocimiento de como el sistema debe ser controlado se puede diseñar un sistema difuso para el bloque de control. De hecho este tipo es excelente con controladores de lazo cerrado no lineales (Lilly, 2011).

Un controlador difuso está compuesto de cuatro partes principales: interfaz de **difusificación**, **base de conocimientos**, **mecanismo de inferencia** e **interfaz de desdifusificación**, en la figura esta descrito de forma mas clara; basándose en la figura 2.8, las cuatro partes del controlador se detallan a continuación (Cruz, 2011).

- **Difusificación:** Mide los valores de las variables de entrada para realizar un mapeo a escala que transfiere el rango de valores de las variables a un universo de discurso difuso. La difusificación convierte los datos de entrada en valores lingüísticos que son las etiquetas de las funciones de pertenencia o conjuntos difusos (Cruz, 2011; Ross, 2005).

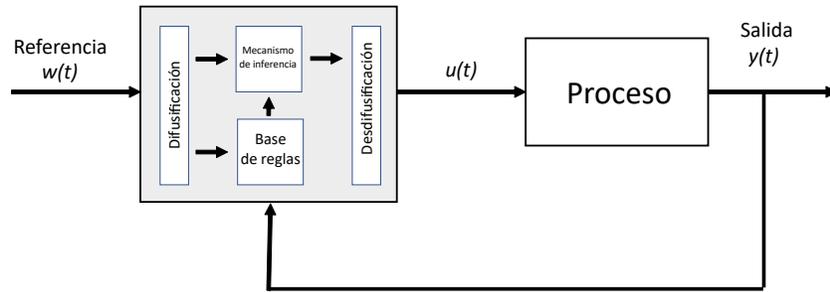


Figura 2.9: Diagrama a bloques de un controlador difuso (Passino *et al.*, 1998).

La representación de información (velocidad, aceleración, error, etc) a través de conjuntos difusos puede realizarse en forma discreta, es posible definir un conjunto difuso asignando un grado de pertenencia a cada elemento del nuevo universo discreto. Para la “discretización” es necesario realizar un mapeo a escala para transformar valores medidos en las variables a valores del universo discreto. Todo esto se puede aplicar para la elaboración de sistemas de control (Cruz, 2011).

Una variable lingüística  $\mu(x)$  se asocia a un conjunto de términos, definido en la misma variable del universo de discurso. Para encontrar cuántos términos son necesarios en un conjunto se emplean particiones difusas. El número de conjuntos difusos determina la complejidad del controlador, y éstos tienen un significado lingüístico como “negativo”, “cero”, “positivo”. La figura 2.10 muestra un ejemplo de una partición difusa en el mismo universo (Cruz, 2011).

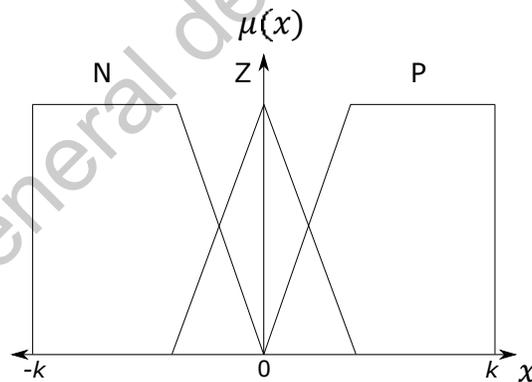


Figura 2.10: Particiones difusas con distinto número de términos negativo (N), cero (Z) y positivo (P). (Cruz, 2011).

- **Base de conocimientos:** Contiene el conocimiento, en la forma de un conjunto de reglas, de como es mejor el control del sistema (Passino *et al.*, 1998). Una regla difusa tiene generalmente la siguiente forma:

$$\text{IF } A_i \text{ AND } B_i \text{ THEN } C_i$$

Donde  $A_i$  y  $B_i$  son las variables representativas del proceso a medir y  $C_i$  es una cantidad difusa representativa de la señal de control (De Silva, 1995).

- **Mecanismo de inferencia:** Evalúa que reglas de control son importantes en el momento actual y después decide cual sera la entrada de la planta. Esta inferencia puede ser representada tomando en cuenta el máximo, mínimo o el producto de las reglas importantes (Ross, 2005; Passino *et al.*, 1998).
- **Interfaz de desdifusificación:** Convierte las conclusiones obtenidas por el mecanismo de inferencia en una solo entrada de la planta. Existen diferentes tipos de desdifusificación, en este proyecto abordaremos el tipo Sugeno para tener un recurso computacional mínimo.

En resumen se tiene como activación de reglas  $i = 1, 2$  la ecuación (2.19).

$$\begin{aligned}\alpha_1 &= A_1(x_0) \wedge B_1(y_0) \\ \alpha_2 &= A_2(x_0) \wedge B_2(y_0)\end{aligned}\tag{2.19}$$

Las salidas individuales de las reglas están descritas en la ecuación (2.20).

$$\begin{aligned}Z_1^* &= a_1x_0 + b_1y_0 \\ Z_2^* &= a_2x_0 + b_2y_0\end{aligned}\tag{2.20}$$

Como nivel de activación de la salida se tiene la ecuación (2.21).

$$Z_0 = \frac{\sum_{i=1}^n a_i Z_i^*}{\sum_{i=1}^n a_i}\tag{2.21}$$

#### 2.1.4 Red Neuronal

Las redes neuronales se basan en generalizar información extraída de datos experimentales o bases de datos, los cuales se determinan por expertos en el tema. Dichas redes neuronales toman en cuenta las entradas (corriente, voltaje) y como salidas las señales del sistema (velocidad, torque, posición) (Cruz, 2011).

En general, una red artificial neuronal se puede dividir en tres partes, las cuales son (Da Silva *et al.*, 2017):

1. **Capa de entrada:** Esta capa es la responsable de recibir señales o datos medidos de un ambiente externo. Estas entradas son usualmente normalizadas dentro de los valores limite producidos por las funciones de activación .
2. **Capa escondida o intermedia:** Estas capas están compuestas de neuronas que son responsables de extraer patrones que son asociados con el proceso o planta.
3. **Capa de salida:** Esta capa también esta compuesta de neuronas, es responsable de predecir y presentar la salida de la red.

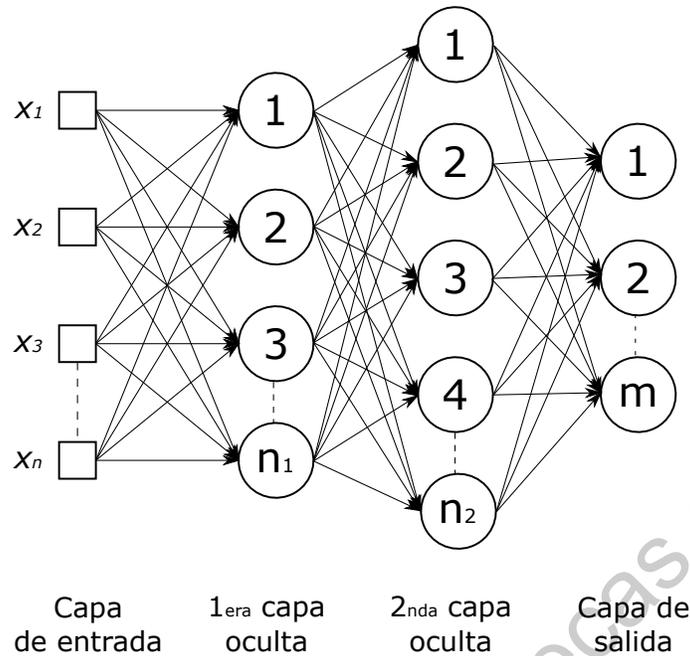


Figura 2.11: Ejemplo de una red neuronal multicapa (Nguyen *et al.*, 2002; Cruz, 2011).

La figura 2.11 muestra una red neuronal con múltiples capas compuesta de una capa de entrada con  $n$  muestras de señal, dos capas ocultas y finalmente capa de salida compuesta de  $m$  neuronas. En una neurona artificial el cálculo de la entrada neta se puede representar con la ecuación (2.22), y en forma vectorial se representa con la ecuación (2.23) (Caicedo Bravo y López Sotelo, 2009).

$$Net_j = \sum_{i=1}^N x_i w_{ji} + \theta_j \quad (2.22)$$

$$\begin{aligned} Net_j &= w_1 x_{j1} + w_2 x_{j2} + \dots + w_i x_{ji} + \dots + w_N x_{jN} + \theta_j \\ Net_j &= W^T X_j + \theta_j \end{aligned} \quad (2.23)$$

donde:

- $N$  es número de neuronas en la capa de entrada.
- $X_i$  es el componente  $i$ -ésima de la entrada correspondiente al  $p$ -ésimo patrón.
- $w_{ji}$  Valor del peso de la conexión entre la  $j$ -ésima neurona de la capa de procesamiento y la  $i$ -ésima neurona de la capa de entrada.
- $Net_j$  Valor de salida deseado para la  $j$ -ésima neurona de la capa de salida.

La función de activación generalmente es del tipo escalón, lineal o sigmoideal según se presenta en la figura 2.10. pendiente.

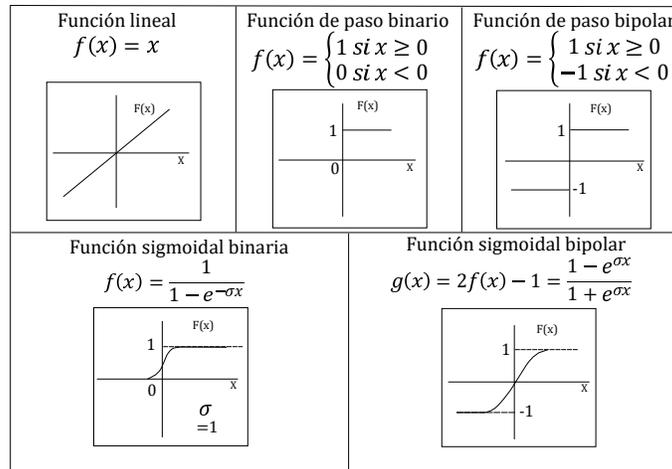


Figura 2.12: Funciones de activación (Cruz, 2011).

### 2.1.5 Red ANFIS

Para ilustrar el uso de redes neuronales para inferencia difusa, se presentan algunos sistemas exitosos de inferencia difusa de redes neuronales adaptativas. Estas estructuras, también conocidas como sistemas de inferencia neuro-difusos adaptativos o sistemas de inferencia difusos de red adaptativa (ANFIS) (Nguyen *et al.*, 2002).

Para corregir las ideas, se considera el problema de representar gráficamente la forma en que se logra el control difuso en el modelo Sugeno de primer orden con dos reglas *IF ELSE*.

$$R_1: \text{IF } x_1 \text{ IS } A_1 \text{ AND } x_2 \text{ IS } B_1 \text{ THEN } f_1 = p_1x + q_1y + r_1$$

$$R_2: \text{IF } x_1 \text{ IS } A_2 \text{ AND } x_2 \text{ IS } B_2 \text{ THEN } f_2 = p_2x + q_2y + r_2$$

Estas reglas pueden ser vistas de forma más gráfica, en la figura 2.13 se muestra esta relación.

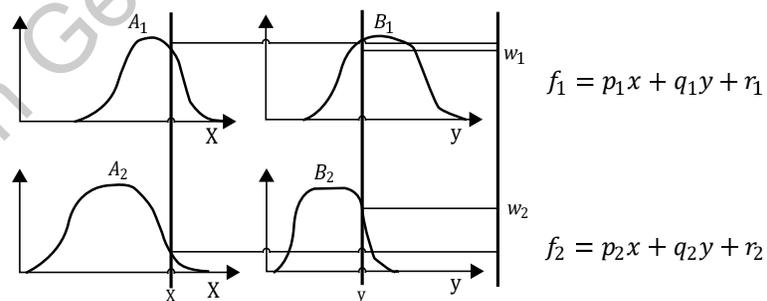


Figura 2.13: Modelo Sugeno de primer orden con dos reglas.

Un sistema ANFIS se representa en la figura 2.14, como se puede observar el sistema ANFIS se divide en cinco capas las cuales son descritas a continuación (Cruz, 2011; Nguyen *et al.*, 2002).

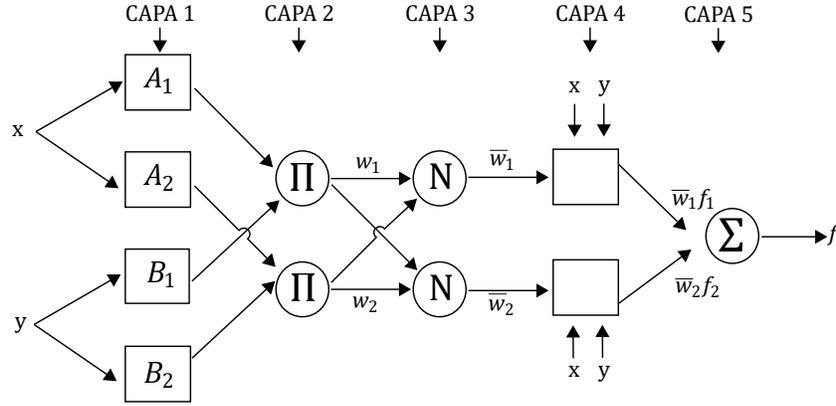


Figura 2.14: Diagrama a bloques de un sistema ANFIS (Cruz, 2011; Nguyen *et al.*, 2002).

- **CAPA 1** cada nodo  $i$  en esta capa es un nodo adaptativo con una función nodo:

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x), \text{ for } i = 1, 2 \\ O_{1,i} &= \mu_{B_{i-2}}(y), \text{ for } i = 3, 4 \end{aligned} \quad (2.24)$$

En otras palabras  $O_{1,i}$  es el grado de membresía del conjunto difuso  $A_i$  (o  $B_i$ ). Aquí la función de membresía puede ser cualquier función parametrizada, tal como la función campana de la ecuación.

$$\mu_{A_i} = \frac{1}{1 + \left[ \frac{x-c_i}{a_i} \right]^{2b}} \quad (2.25)$$

Donde  $a, b$  y  $c$  son parámetros del conjunto, estos pueden ser llamados parámetros de premisa.

- **CAPA 2:** Cada nodo en esta capa se ajusta con la etiqueta  $\Pi$ , estas salidas son el producto de todas las señales de entrada:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1, 2 \quad (2.26)$$

Cada salida del nodo representa la activación de una regla.

- **CAPA 3:** Cada nodo en esta capa se nombra con la etiqueta **N**. El  $i$ -ésimo nodo calcula el radio de la  $i$ -ésima regla de activación a la suma de todas las reglas de activación.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (2.27)$$

Por conveniencia, las salidas de esta capa son llamadas salidas de activación normalizadas.

- **CAPA 4:** Cada nodo  $i$  en esta capa es un nodo adaptativo con un nodo función, y tiene como salida la ecuación (2.28)

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.28)$$

Donde  $w_i$  es una función normalizada de activación de la capa 3 y  $(p_i, q_i, r_i)$  es un conjunto de parámetros de este nodo, estos se conocen como parámetros subsecuentes.

- **CAPA 5:** El único nodo en esta capa se ajusta con aa etiqueta  $\sum$ , la cual calcula la salida promedio de la suma de las señales de entrada

$$overall\_output = O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.29)$$

## 2.2 Antecedentes

- En enero de 1994, Hesselroth y su equipo emplea un algoritmo neuronal para controlar un brazo robótico de 5 grados de libertad y un gripper (pinza) a través de retroalimentación desde dos video cámaras, para controlar la posición del brazo, 200 neuronas formaron la red representando el espacio de trabajo de 3 dimensiones desde las dos cámaras (Hesselroth *et al.*, 1994).
- En el año 2006, Kiyama y Vargas desarrolla un modelo matemático para aproximar la posición de un manipulador de 1 grado de libertad, el manipulador está compuesto de un cilindro neumático equipado con almohadillas de frenado. Para el modelado neumático se basa en principios termodinámicos de la conservación de la energía y la materia, además el modelado mecánico es construido mediante la formulación de Newton-Euler (Kiyama y Vargas, 2004).
- En 2013 Gayo realiza la fabricación de un brazo robótico o robot industrial. se utilizarán únicamente actuadores neumáticos lineales, se pretende conseguir un coste de producción reducido para la fabricación de pocas unidades y se realizará un diseño sencillo tanto para su instalación y puesta en marcha como para su mantenimiento. Para desarrollar el diseño se utilizarán herramientas de software CAD-CAE de última generación (Gayo Martínez, 2013).
- En el año 2016, Sánchez y asesores desarrollaron una herramienta que permite simular el comportamiento de un manipulador antes de su implementación física, lo que permite al usuario tener una idea acerca de la respuesta del manipulador, en el documento se utiliza un manipulador neumático de dos grados de libertad (Sánchez Solar *et al.*, 2016).
- En enero del 2016, Ramos Arreguín y su equipo de trabajo realizan la implementación de un control PD para un actuador neumático embebido en un FPGA, la cual es aplicada para mover un robot manipulador de un grado de libertad (Ramos-Arreguín *et al.*, 2016).
- En mayo del 2017 Gaspar y su equipo centran su trabajo en el diseño de un robot neumático con 4 grados de libertad, incluida la interfaz de hardware para controlar los actuadores neumáticos, y algunas pruebas se desarrollan con un Texas DSP Delfino C2000. Los resultados muestran un diseño de robot neumático que incluye hardware electrónico y un algoritmo de control (Gaspar-Badillo *et al.*, 2017).
- En mayo del 2017, Tuvayanond y Parnichkun propone un novedoso desarrollo de control de un robot neumático quirúrgico de dos grados de libertad. Se consideró la respuesta dinámica en lazo cerrado, estabilidad y mínima entrada de control en los criterios de diseño para controlar la posición del robot quirúrgico neumático de dos grados de libertad. A diferencia de otros controladores convencionales, el controlador propuesto ofrece robustez, alto rendimiento, pero estructura simple rentable (Tuvayanond y Parnichkun, 2017).

Los siguientes trabajos fueron presentados el año pasado.

- Li y su equipo presentan un gripper basado en actuadores ligeros precargados neumáticamente. Estos actuadores son precargados con aire para un preset de presión a través de una válvula de un solo sentido. El ángulo y velocidad de inclinación son controlados por servomotores a través de tendones basados en datos de retroalimentación desde sensores de proximidad y de fuerza (Li *et al.*, 2019).
- En febrero, Lee y compañeros presentan el desarrollo y control de un actuador neumático de tres grados de libertad con visión por computadora. El sistema de visión robotizada (que incluye una cámara web montada en el actuador) recolecta imágenes de objetos en el transportador y las transfiere a través de la interfaz de programación de la aplicación LabVIEW para el procesamiento de imágenes (Lee *et al.*, 2019).
- Mu y su equipo proponen un nuevo método de control difuso predictivo combinado con una red neuronal para un sistema servo neumático. Con este método propuesto, se predice las futuras salidas de la planta, usando una planta imaginaria en el esquema predictivo difuso. El error entre el valor deseado y la salida de la planta imaginaria y el cambio en este error se aplica como señal de entrada al controlador difuso predictivo para generar una señal de entrada al control directo. Se introduce la red neuronal para construir el modelo imaginario de la planta del sistema (Mu *et al.*, 2019).
- Kaneishi y compañeros presentan en su artículo actuadores neumáticos para tecnologías tales como exoesqueletos. Este artículo presenta un nuevo método de control para hacer que un cilindro de aire funcione como un resorte pasivo no lineal con rigidez controlable. El controlador calcula las presiones deseadas en función de una rigidez dada. El método propuesto satisface la respuesta de rigidez deseada de menos de 20 milisegundos y dentro de  $\pm 5\%$  de las presiones deseadas, (Kaneishi *et al.*, 2019).
- En abril, Zhong y Zhao diseñaron una configuración única accionada por un actuador muscular neumático, y se construyeron modelos polinómicos cuánticos empíricos del aparato mediante el ajuste de datos de desplazamiento de presión muestreados. Los nuevos modelos son compactos y fácilmente computables. Los experimentos se realizaron para evaluar el controlador propuesto para mejorar la precisión de la posición del aparato (Zhong y Zhao, 2019).
- Zhang y su equipo presentan un novedoso diseño de un manipulador de cinco grados de libertad que se maneja por motores neumáticos. Este diseño es simple pero efectivo, los experimentos fueron desarrollados bajo diferentes presiones de aire para validar la exactitud de obtener el ángulo deseado. Se utiliza un método de control con retroalimentación, en el cual el ángulo requerido se obtiene por retroalimentación de sentido (Zhang *et al.*, 2019).
- Katzschmann y compañeros presentan en su artículo un método eficiente de modelar y simular robots ligeros. Para mostrar el rendimiento de este método de control, se realizan experimentos de pose a pose y seguimiento de trayectoria en un brazo ligero accionado neumáticamente ((Katzschmann *et al.*, 2019).
- Wang construye un brazo antropomórfico de dos enlaces, se construye con músculos artificiales neumáticos para entrenamientos de rehabilitación. El brazo debe realizar movimientos flexibles que puedan hacer los humanos. Debido a las fricciones e incertezas del modelo del

brazo se propone un control adaptativo de paso atrás difuso para garantizar la estabilidad y la adaptabilidad durante el movimiento, se demuestra que el método del control propuesto es eficiente y robusto (Wang *et al.*, 2019).

- Yusof y equipo desarrollan un sistema robótico neumático de selección y colocación con control de supervisión compartida donde es altamente deseable en la actual industria y está al día con la revolución industrial 4.0 que se está desarrollando en todo el mundo. Se utiliza una cámara no estática Pixy CMU cam5 como sensor de visión que proporciona señales para el movimiento del sistema de recogida y colocación en un modo automático (Yusof *et al.*, 2019).
- En el año 2019, Pérez alumno de la Universidad Autónoma de Querétaro desarrolla el modelado y simulación de un robot impulsado por actuadores neumáticos, a través de técnicas avanzadas de robótica, tales como el método Denavit Hartenberg y métodos geométricos, Se obtiene el modelo cinemático directo e inverso correspondiente al sistema robótico (Delgado *et al.*, 2018).

## 2.3 Estado del Arte

Los siguientes trabajos se presentan en el año en curso.

- Zoller y equipo presentan un sensor acústico activo que convierte los actuadores neumáticos ligeros en sensores de contacto. Toda la superficie del actuador se convierte en un sensor, lo que hace innecesaria la cuestión de dónde colocar mejor un sensor de contacto. Al mismo tiempo, el cumplimiento del actuador ligero no se ve afectado. Un pequeño altavoz incorporado emite un barrido de frecuencia que viaja a través del actuador antes de grabarlo con un micrófono incorporado (Zöller *et al.*, 2020).
- Chia y su equipo proporcionan un dispositivo de equilibrio de carga para un brazo de robot que incluye un cilindro neumático y un vástago de pistón. El cilindro neumático, usado para almacenar un gas, incluye una primera cámara, una segunda cámara y un paso de comunicación, en donde el paso de comunicación conecta la primera cámara y la segunda cámara. El vástago del pistón tiene un extremo conectado al brazo del robot y el otro extremo dispuesto de forma deslizante en el cilindro neumático. El vástago ajusta el volumen y la presión del gas en la primera cámara y la segunda cámara de acuerdo con una carga, en donde la primera cámara y la segunda cámara están dispuestas coaxialmente en la dirección axial del cilindro neumático (Chia-Chung *et al.*, 2020).
- Trumić y equipo presentan y comparan el rendimiento de dos técnicas para controlar de forma independiente y simultánea la posición y la rigidez de la articulación de un robot neumático ligero. Se considerará el sistema dinámico de un brazo robótico con articulaciones flexibles accionadas por músculos artificiales de McKibben, mientras que sus parámetros dinámicos se considerarán imprecisos. Se proporcionan resultados de simulación para verificar el rendimiento de los algoritmos (Trumić *et al.*, 2020).

# Metodología

Para el desarrollo de este proyecto, se va a utilizar una metodología donde el proceso se divide en 8 secciones representados en bloques, acorde con la figura 3.1. En el bloque (1) y (8) se trabaja la documentación del proyecto. En el bloque (2) se analizan las herramientas de hardware y software que se utilizan en el proyecto. En los bloques (3), (4) y (5) se profundiza en la parte teórica del controlador además de la simulación del controlador. En el bloque (6) se implementa el controlador en el hardware elegido ya implementado en el robot neumático.

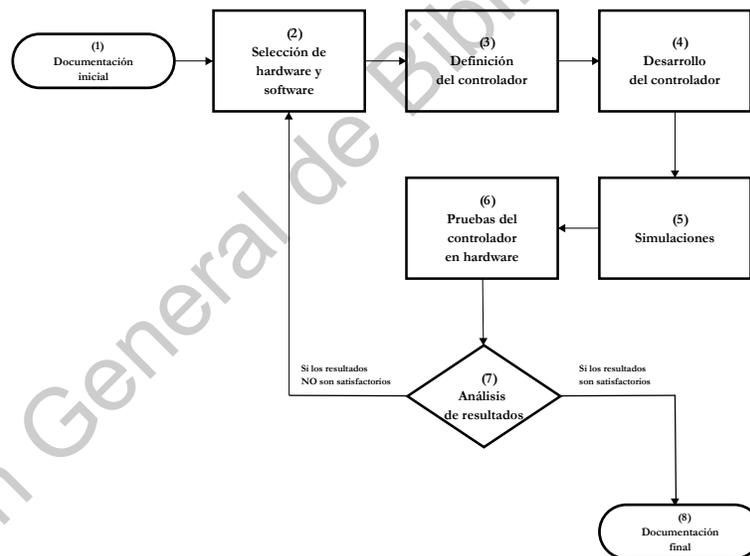


Figura 3.1: Diagrama de bloques del proyecto

## 3.1 Herramientas de hardware y software

En la figura 3.2 hay un bosquejo de todo el hardware necesario para este proyecto, se comienza con una fuente industrial ajustada a 20 volts seguido de un *buck-reductor* a 12 volts que alimenta

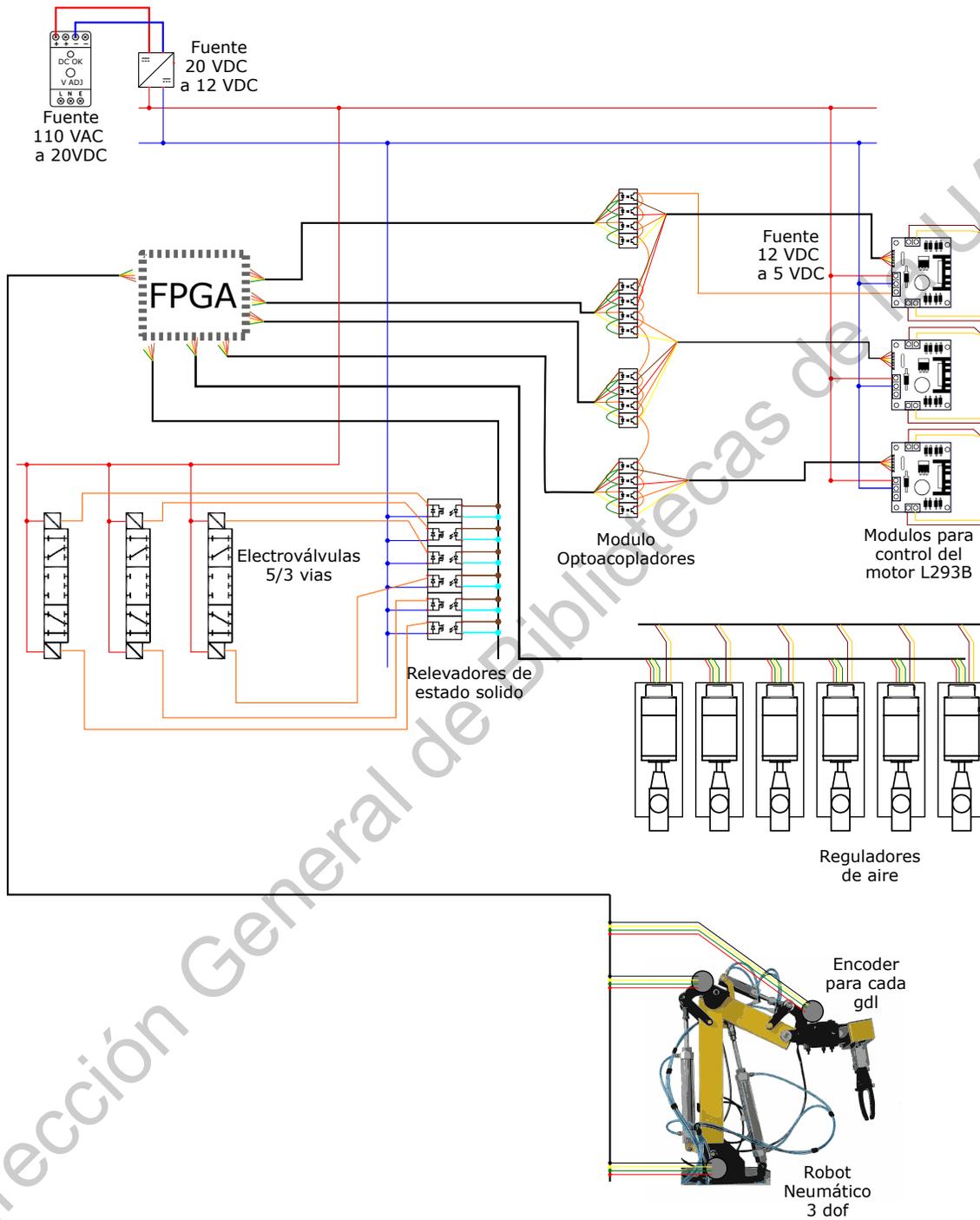


Figura 3.2: Bosquejo del hardware completo para el proyecto.

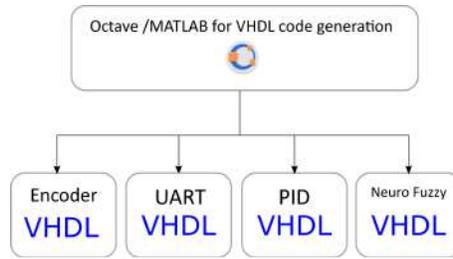


Figura 3.3: Diagrama a bloques de la relación de software de la primera etapa.

a la mayoría de los componentes del proyecto, por otro lado la FPGA, como componente central controla las señales de control de los motores la de los pistones y las señales que vienen de los encoders, adicionalmente tres encoders son conectados en el robot para medir la posición de las articulaciones  $q = (\theta_1, \theta_2, \theta_3)$ , las señales de control de los motores y la activación de los relevadores de las electro-válvulas son aisladas con opto-acopladores y relevadores de estado solido respectivamente.

En este proyecto se contemplan gran cantidad de herramientas, en su mayoría gratuitas o con licencia de estudiante; entre los numerosos lenguajes de programación (como *Python*, *C++*, *Java*, *C#*, etc.) se elige el lenguaje  $m$  que es especial para calculo numérico lo cual hace más fácil la programación de algoritmos, este lenguaje es *Matlab* pero a su vez existe un software libre que maneja el lenguaje  $m$  que es *Octave*.

El componente central que se encargara de todas las partes de control es un *FPGA* (*Field Programming Gate Array*), este tipo de dispositivos programables se programan con lenguajes especiales de descripción de hardware (*VHDL*, *verilog*, etc.) en este caso se elige trabajar con *VHDL*. En la figura 3.3 se observa la relación de software necesarias para la primera parte, donde con el código  $m$  se realiza de forma automática la generación de archivos *VHDL*; mucho del código esta basado en el trabajo del Dr. Romero Troncoso (Troncoso, 2007) así como varios algoritmos para la binarización por igual. Como se observa en la figura 3.3 existen códigos que generan varios bloques, entre los mas importantes son los de la comunicación serial, controlador *PID*, otro bloque para el difuso y uno para bloque sobre la configuración del encoder del motor.

Para la segunda parte se hace una interfaz en LabView para la adquisición de datos, estos son guardados en formato *.txt* y finalmente son graficados en un script de octave estos serán mostrados en la sección de resultados. En la figura 3.4 se muestra la relación de software de esta parte donde se puede observar que los parámetros más importantes son el tiempo, velocidad y posición de los motores, y valores de las articulaciones y posición del efector final.

Se utiliza para este proyecto la placa DE0-Nano, esta presenta una plataforma de desarrollo FPGA de tamaño compacto adecuada para una amplia variedad de proyectos de diseño portátiles, como robots y proyectos móviles; cuenta con una poderosa FPGA Altera Cyclone IV EP4CE22F17C6N (con 22,320 elementos lógicos y 132 multiplicadores de 9 bits), 32 MB de SDRAM, 2 Kb EEPROM y una serie de 64 Mb de memoria de configuración para conectarse a sensores del mundo real.

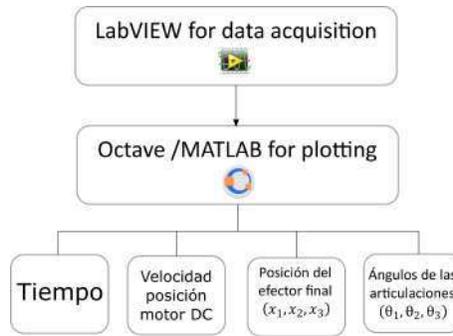


Figura 3.4: Diagrama a bloques de la relación de software de la segunda etapa.

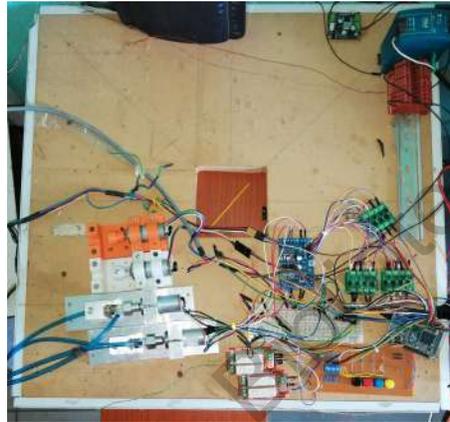


Figura 3.5: Fotografía de el hardware de control del robot.

En la figura 3.5 se muestra una fotografía del hardware del robot incluyendo la FPGA, las bases y los motores, optoacopladores y relevadores de estado sólido; se elaboró una pequeña botonera para el control manual del brazo.

### 3.1.1 Configuración del encoder

Para la configuración del encoder se elabora un script en lenguaje  $m$  el cual puede verse gráficamente en la figura 3.6, donde se tiene como entradas los dos canales  $A$  y  $B$  del encoder así como de una señal de muestreo  $T_s$  para el calculo de la velocidad del motor. Como se observa también en la figura 3.6, se tiene 5 salidas  $dir$  que indica la dirección del motor, las señales  $vel_{nom}$  y  $pos_{nom}$  que son los valores de la velocidad y posición respectivamente de las señales en 16 bits solo en pulsos. Para la salida de velocidad normalizada se tiene la ecuación (3.1) en 18 bits con formato 3.15. para este calculo se tiene la velocidad máxima del motor en este caso 108  $RPM$  con  $N_v = 3432$   $ppr$ , considerando la reducción  $T_s$  es el tiempo de muestreo que en este proyecto se trabaja a 10ms.

$$vel_{norm} = \frac{60}{(V_{MAX} N_v) T_s} \quad (3.1)$$

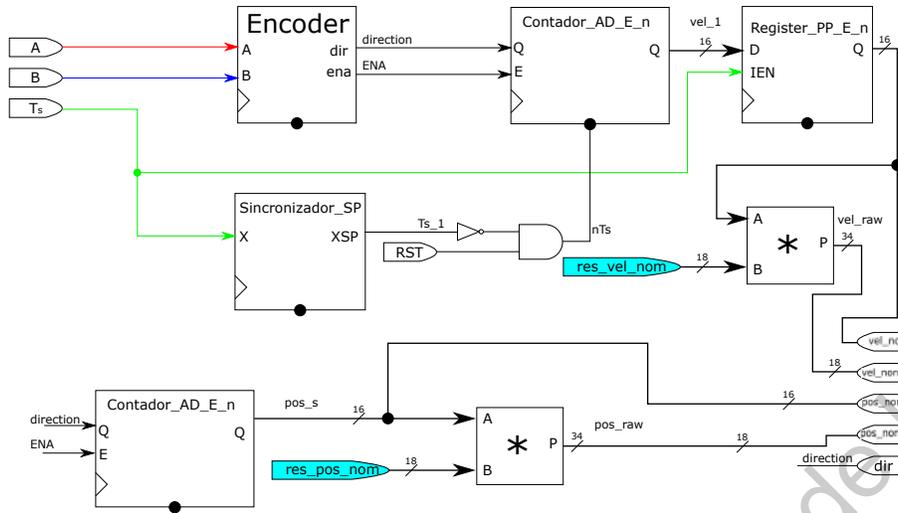


Figura 3.6: Diagrama a bloques de los módulos VHDL del encoder.

Para la normalización de la posición en 18 bits con formato 3.15 se toma en cuenta que el máximo de 10 vueltas de la válvula reguladora de flujo se tiene para su normalización la ecuación (3.2).

$$pos\_norm = \frac{I}{(P_{MAX} N_v)} \quad (3.2)$$

### 3.1.2 Configuración de la comunicación UAT

Para la comunicación serial se decide trabajar con una *FIFO* (*first in first out*) que permite una automatización más rápida de datos al ser transmitidos por el puerto serial estos al ser primero guardados en la *FIFO* antes de ser transmitidos deben ser múltiplos de 8 los datos a guardar, además desde el script configura los parámetros importantes de la transmisión como lo es la velocidad que se configura a 115200 baudios, en la figura esta descrito mediante bloques VHDL la estructura de este modulo.

La mayoría de los módulos están descritos en el libro del Dr. Romero Troncoso (Troncoso, 2007) la maquina de estados FSM\_myUART se muestra en la figura 3.8. En el estado  $S_0$  el bloque espera una señal de reloj hasta que esta se hace presente  $STP = "1"$  entonces se cambia al estado  $S_1$ , en el estado  $S_1$  la FIFO comienza a guardar los valores hasta que la bandera  $FULL = "1"$ , además el contador multifuncional sirve de indice y cambia el valor del dato cada ciclo del reloj; en el estado  $S_2$  se verifica que la FIFO este vacía si es así se vuelve al estado  $S_0$ , si la bandera  $EMPTY = "0"$  se procede al estado  $S_3$  y se pone la  $STT$  a "1", en el estado  $S_3$  se vacía un dato de la FIFO y se procede inmediatamente al estado  $S_4$ , por ultimo en el estado  $S_4$  se espera el fin de la transmisión cuando esta ocurre vuelve al estado  $S_2$  donde se vuelve a verificar que la FIFO este vacía, de no ser así se seguirá mandando un nuevo dato.

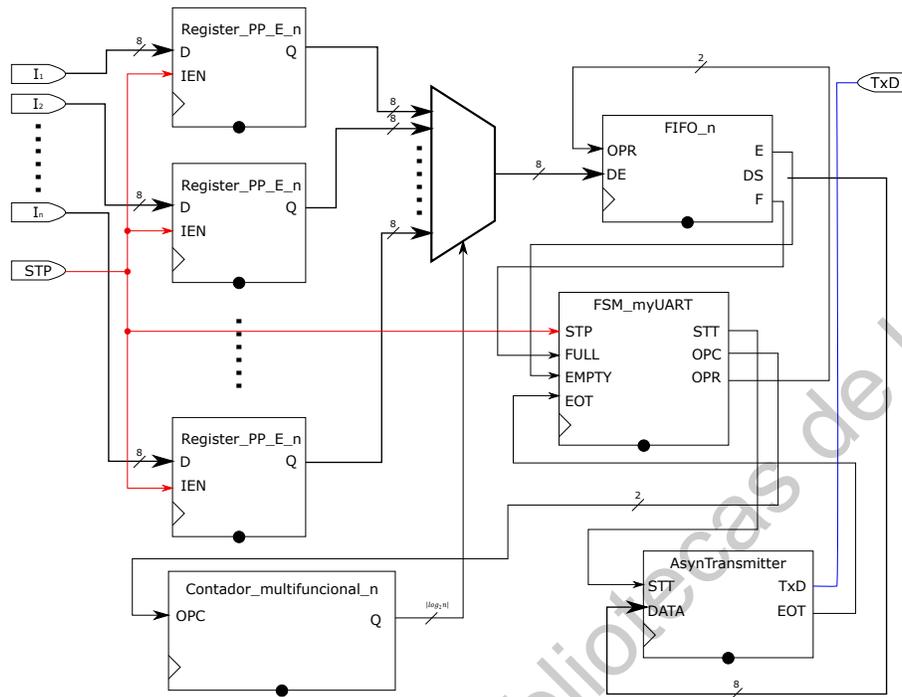


Figura 3.7: Diagrama a bloques de los módulos VHDL para la transmisión serial.

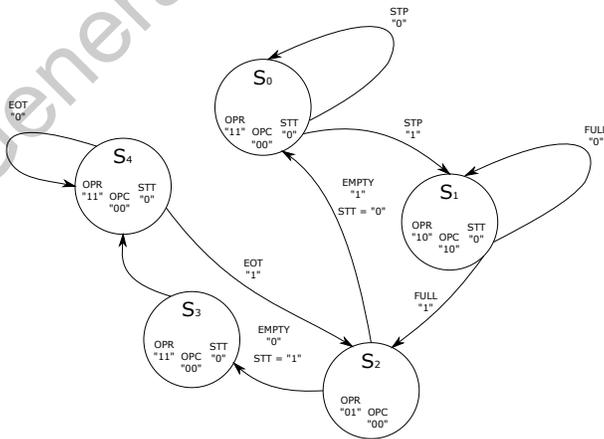


Figura 3.8: Máquina de estados para el módulo de comunicación.

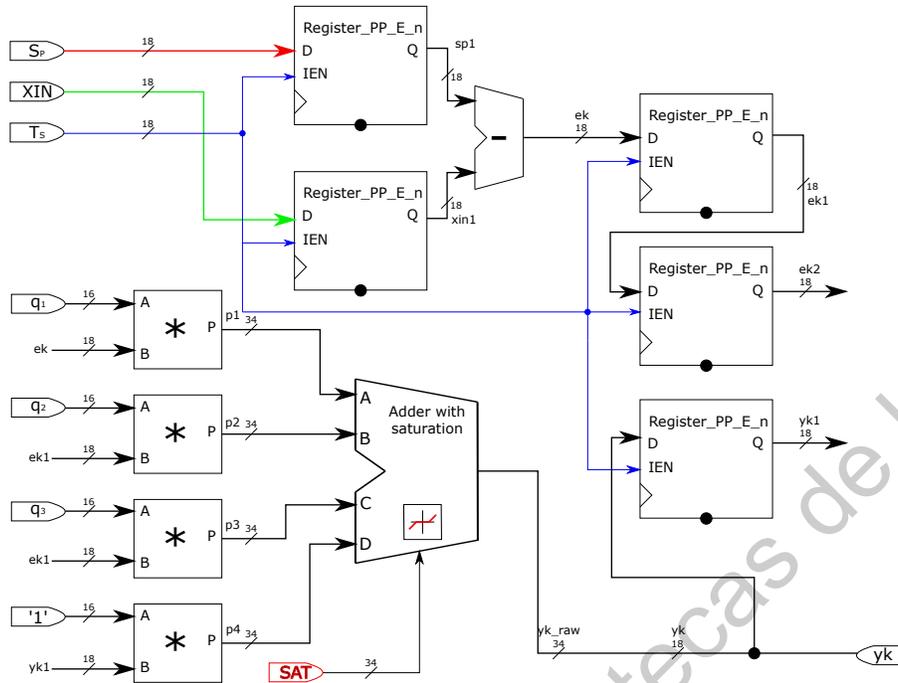


Figura 3.9: Diagrama a bloques del código VHDL para el controlador PID.

### 3.1.3 Configuración del controlador PID

Basándose en la ecuación (2.17) se diseña un script que genere el código VHDL para un controlador PID el cual es mostrado en la figura 3.9.

Se puede observar las principales señales de entrada  $Sp$  (*setpoint*),  $XIN$  (dato de entrada), y  $T_s$  (señal de muestreo) estas son guardadas cada vez que se active la señal de muestreo después se aplica una resta  $ek = sp - xin$ ;  $q_1$ ,  $q_2$  y  $q_3$  son las constantes del controlador, se tiene un bloque especial de nombre *Adder with saturation* donde realiza la sumatoria  $y_{k\_raw} = q_1 \times ek + q_2 \times ek_2 + q_3 \times ek_3 + y_{k1}$  además de una señal de saturación que elimina el efecto *windup*. La señal de salida del controlador  $y_k$  es ajustada a 18 bits.

## 3.2 Control del flujo de aire

Se propone la siguiente configuración el control del flujo de aire aplicado a grado de libertad, se constituye por 1 electroválvula  $\frac{5}{3}$  vías, 2 válvulas reguladoras de aire, y 1 encoder que indica la posición del grado correspondiente. El diagrama de conexión para 1 gdl se muestra en la figura 3.10.

Para la conexión entre los motores y la válvula reguladora de flujo se utiliza la diseñada por Pedro Camacho (Camacho Angulo, 2019), este es una base donde se conectan los dos componentes, además, de una pieza especial que conecta el vástago del motor con la forma hexagonal del regulador de la válvula; en la figura 3.11 se muestra la base utilizada.

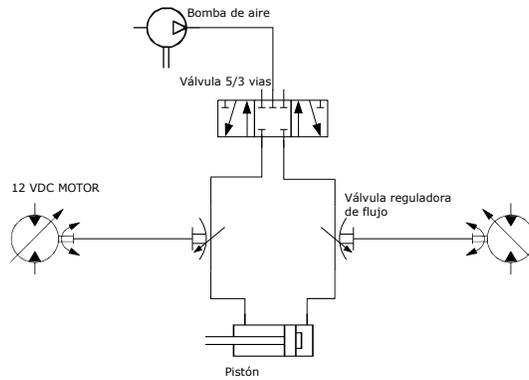


Figura 3.10: Diagrama de 1 gdl.

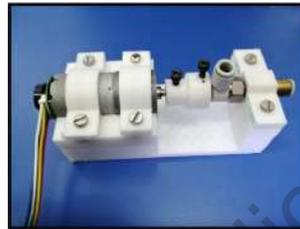


Figura 3.11: Base para el motor y la válvula reguladora de aire.

### 3.2.1 Diseño del controlador del motor

Como el motor DC representa un sistema lineal, un controlador clásico es el indicado. Para poder comenzar con la posterior desarrollo del control inteligente es importante que esta etapa sea optima, tomando en cuenta los movimientos abruptos del robot y que los cambios del flujo de aire no sean grandes se decide el diseño de controlador que pueda manejar tanto velocidad como posición, el diseño del controlador esta dado en la figura 3.12 un  $PD$  se utiliza para el control de velocidad y un  $PID$  para el de posición.

Para el diseño del controlador se toma en cuenta el diagrama a bloques de la figura 3.9 y la ecuación (2.17). En la sección de resultados se encuentran las simulaciones y experimentaciones realizadas.

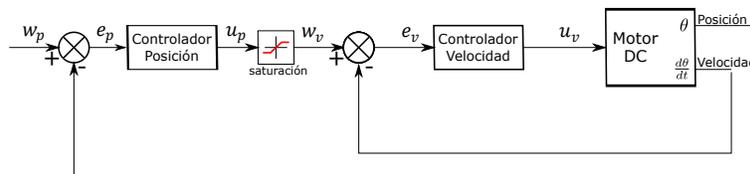


Figura 3.12: Sistema de control propuesto para los motores DC.

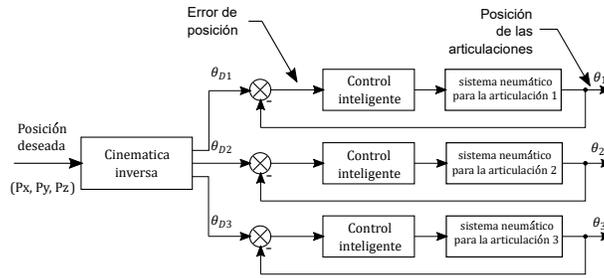


Figura 3.13: Sistema de control completo.

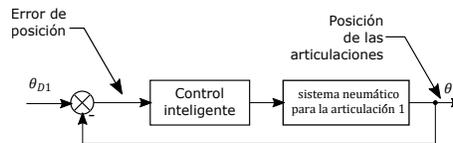


Figura 3.14: Sistema de control para un grado de libertad.

### 3.3 Control inteligente

Para este trabajo se plantean técnicas de control inteligente, en combinación con el clásico ya que sigue siendo eficaz con este tipo de problemas, las técnicas a utilizar son las siguientes:

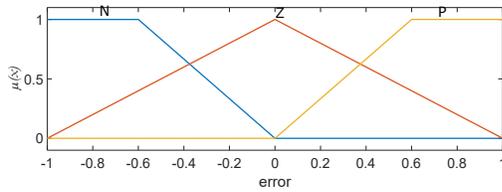
- *PI* difuso (solo para aprendizaje). Cuando se inicio el análisis de controladores difusos en dispositivos programables se eligió este tipo ya que puede ser implementado en motores DC, dando buenos resultados, pero se descarto como controlador principal del brazo neumático ya que al cambia los ajustes de *setpoint* este se comportaba de manera no adecuada, pero aun así, fue un gran paso al conocimiento de este tipo de controladores en FPGA.
- *PID* con red neuronal para ajustes de ganancias.
- Sistema *ANFIS*, se trata de una red neuronal con inferencia difusa permite un ajuste en tiempo real del controlador.

En la figura 3.13 se muestra la propuesta de control para este proyecto, cada grado de libertad sera posicionado por un controlador mencionado en esta sección, en el bloque de cinemática inversa se lleva a cabo lo descrito en la ecuación (2.10), el bloque de sistema neumático se refiere al dibujo de la figura 3.10.

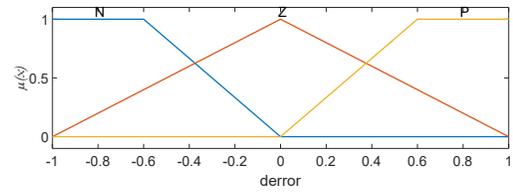
Para simplificar el problema y abordar la resolución del posicionamiento final, se toma primero en cuenta un grado de libertad como se muestra en la figura. Su diseño es evaluado en las primeras etapas mostrado en la sección de resultados.

#### 3.3.1 PI Difuso

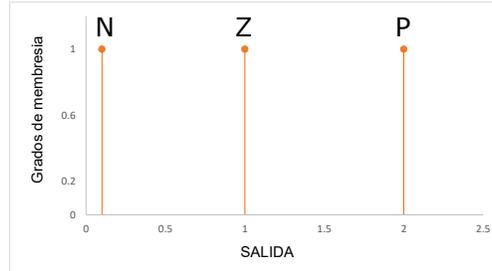
Para diseñar los controladores tipo PI difuso, se utilizó las funciones de pertenencia de la figura 3.15 los nombres de las entradas son error y derivada del error (derror) cada entrada tiene 3 conjuntos



(a) Entrada membresía error.



(b) Entrada derivada del error (derror).



(c) Salida del controlador.

Figura 3.15: Diseño de los conjuntos de membresía de un controlador PI difuso.

Tabla 3.1: Matriz asociativa difusa para un controlador PI difuso, N = Negativo, Z = cero, P = Positivo [Cruz \(2011\)](#)

	ERROR			
	N	Z	P	
DERIVADA DEL ERROR	N	N	Z	P
	Z	N	Z	P
	P	N	Z	P

de membresía (negativo, cero, positivo) que va entre los valores  $-1$  a  $1$ ; la relación *IF-ELSE* esta en la tabla [3.1](#). En la salida de los controladores se colocó bloques de integración para convertirlos en *PI*, como se muestra en la figura [3.16](#), este diseño esta basado en el desarrollado por Ponce Cruz [\(Cruz, 2011\)](#).

### 3.3.2 PID con red neuronal para ajustes de ganancias

Este controlador esta basado en el realizado por el Dr. Efrén Gorrostieta [\(Gorrostieta y Vargas Soto, 2008\)](#), que con el propósito de dar seguimiento a las trayectorias generadas automáticamente, y aplicarlas como parte de la estrategia de los algoritmos de locomoción libre, se utilizó, diseño y aplicó un sistema de control neuronal, mismo que se representa en la figura [3.17](#).

La ecuación [\(3.3\)](#) muestra el cálculo del error. Este error se obtiene de la posición deseada para cada una de las articulaciones del robot  $q = (\theta_1, \theta_2, \theta_3)$  y la posición real que se esta midiendo del sistema  $r = (x_1, x_2, x_3)$ .

$$e(t) = y(t) - w(t) \quad (3.3)$$

Los ajustes de los parámetros  $K_p$ ,  $T_D$  y  $T_i$  se realizan mediante la ayuda de una red neuronal

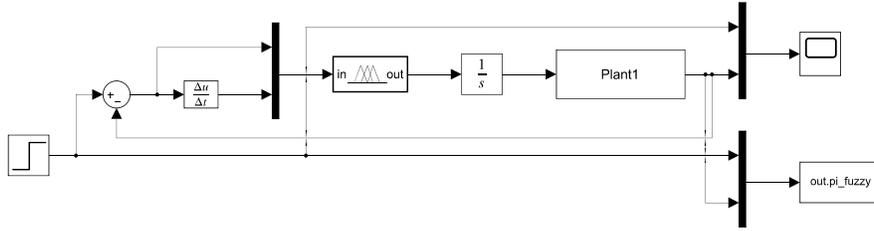


Figura 3.16: PI difuso diseñado en Simulink.

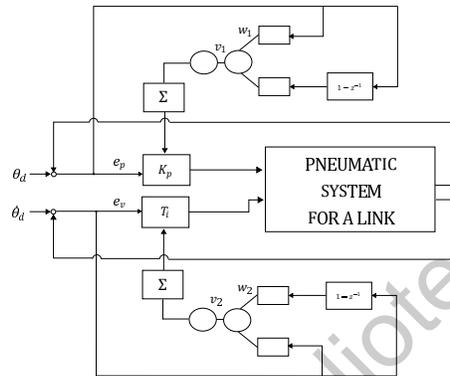


Figura 3.17: Red neuronal para el ajuste de ganancias (Gorrostieta y Vargas Soto, 2008).

para cada una de las constantes, en la figura 3.17 se muestra el sistema de control donde se puede ver como interactúa la red neuronal en el auto ajuste de cada uno de estos parámetros. La entrada en cada uno de las dos redes es el error y la primera diferencia de error, queda descrito en la ecuación (3.4).

$$X(t) = [e(t) \quad \Delta e(t)] \quad (3.4)$$

Las redes son entrenadas por el método conocido como: algoritmo de retro-propagación. La razón de minimización esta dado por la función cuadrática del error de la ecuación (3.5).

$$E(t) = \frac{1}{2} \sum e^2(t) \quad (3.5)$$

La ganancia proporcional esta denotada por las ecuaciones (3.6) y (3.7), donde  $v$  representa los pesos de la última neurona y  $h$  la función de activación, mientras que en la ecuación (3.8) se representa la suma de los pesos de la neurona en la primera capa.

$$K_p = K_p + \Delta K_p \quad (3.6)$$

$$\Delta K_p = vh \quad (3.7)$$

$$s = w_1 e(t) + w_2 \Delta e(t) \quad (3.8)$$

La función de activación se muestra en la ecuación (3.9) de las neuronas intermedias es:

$$h = \frac{1}{1 + e^{-s}} \quad (3.9)$$

Las ecuaciones adaptables (3.10) y (3.11) permiten ajustar los valores de los coeficientes proporcionales  $K_p$ . Un desarrollo similar, se puede utilizar para encontrar las ecuaciones de adaptación para  $T_d$  y  $T_i$ . El valor  $\eta$  es el coeficiente de aprendizaje de la red.

$$v(t + 1) = v(t) + \eta e(t)^2 h \quad (3.10)$$

$$w_j(t + 1) = w_j(t) + \eta e(t)^2 v h (1 - h) x_j \quad (3.11)$$

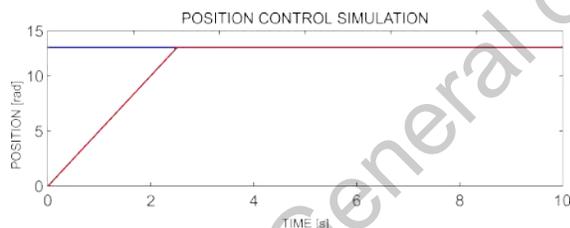
Dirección General de Bibliotecas de la UAQ

# Resultados y discusión

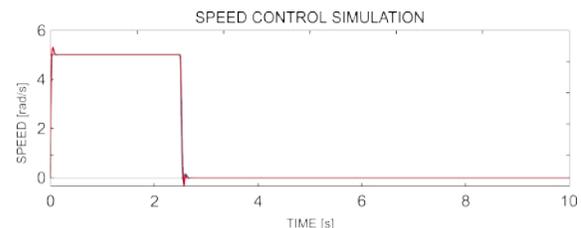
Los resultados aquí presentados se componen de lo siguiente, simulaciones y resultados experimentales del comportamiento de los motores DC, simulaciones de controladores difusos, resultados experimentales para un gdl, resultados del posicionamiento de 3 gdl.

## 4.1 Resultados del Motor DC

En la figura 4.1 se muestra las simulaciones. se configura una posición de setpoint a  $2 \text{ rad}$  y la velocidad a  $5 \text{ rad/s}$ , también se trabaja con un periodo de muestreo de  $10 \text{ ms}$ ; todas las señales son normalizadas. Como se muestra en la figura 4.1a el error de posición equivale a menos del 1%, en la figura 4.1b se muestra la señal de velocidad, esta alcanza su máxima antes de llegar al setpoint de posición, cuando el motor esta alcanzando la posición deseada la velocidad tiende a ser cero.



(a) Grafica de la posición de la simulación.



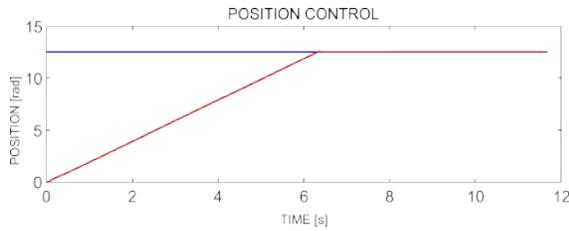
(b) Grafica de la velocidad de la simulación.

Figura 4.1: Simulaciones de la velocidad y posición de los motores DC.

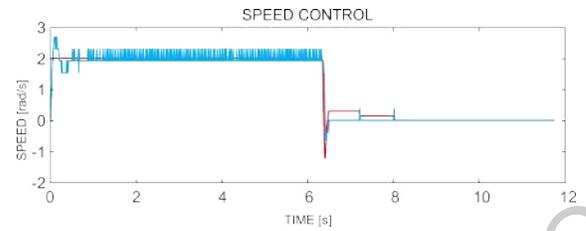
En la figura 4.2 se muestran las graficas de los resultados experimentales, donde la figura 4.2a muestra el control de posición y 4.2b la de velocidad. Se observó una vibración en la grafica de la velocidad por lo que se implementa un filtro pasa-bajas de  $10 \text{ Hertz}$ , los resultados son similares a los de las simulaciones.

## 4.2 Resultados de simulaciones de controladores difusos

Para el aprendizaje en la creación de controladores difusos se realiza unas simulaciones donde se pueda implementar el lenguaje *VHDL* por lo que se realiza una co-simulación entre *ModelSim* y



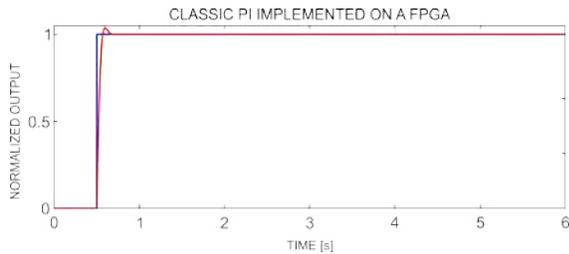
(a) Grafica de la posición de la simulación.



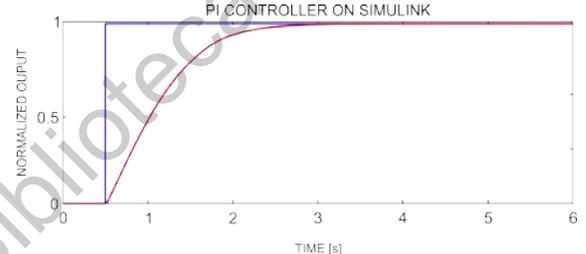
(b) Grafica de la velocidad de la simulación.

Figura 4.2: Simulaciones de la velocidad y posición de los motores DC.

*Simulink* el primero se utiliza para probar códigos *VHDL* y el segundo se emplea en este caso simular sistemas reales; combinados hacen una poderosa herramienta que puede tomarse como la simulación de una *FPGA*, en la figura 4.3, se muestra el resultado de las simulaciones siendo la figura 4.3.4.3a la respuesta entre la co-simulación y la 4.3.4.3b como una simulación hecha enteramente en *Simulink*, estas simulaciones están basadas en la figura 3.16.



(a) Co-simulación entre *ModelSim* y *Simulink*.



(b) Simulación en *Simulink*.

Figura 4.3: Simulaciones de un controlador PI difuso en:

### 4.3 Resultados de un grado de libertad

En base a la ecuación (2.17), y haciendo énfasis en las figuras 2.7 y 3.10. Los resultados y ajuste de las ganancias se muestra en las figuras 4.4, 4.5 y 4.6. En esta parte solo se ajusta la constante  $K_p$  disminuyéndola adquiriendo un mejor resultado.

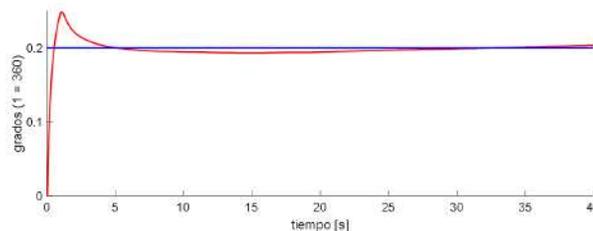


Figura 4.4: Resultado experimental 1 de un controlador *PID* en un gdl.

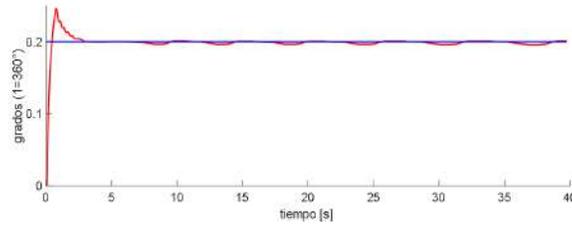


Figura 4.5: Resultado experimental 2 de un controlador *PID* en un gdl.

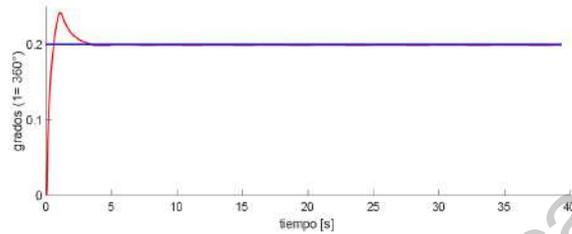


Figura 4.6: Resultado experimental 3 de un controlador *PID* en un gdl.

En la figura 4.7 se encuentran graficas de parámetros importantes para el diseño del control inteligente, la figura 4.7a se muestra el posicionamiento de un gdl con ganancias ajustables, como se observa se muestra un resultado sumamente aceptable; las figura 4.7b y 4.7c representan el error y la derivada del error respectivamente, estos son de importancia para el diseño de controladores, por ultimo la figura 4.7d muestra el setpoint de posición que deben seguir los motores para el posicionamiento de un gdl.

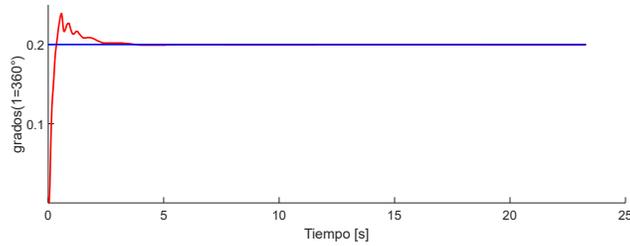
#### 4.4 Resultados con tres grados de libertad

En la figura 4.9 se presentan los resultados del posicionamiento de los tres grados de libertad, en la figura 4.8 se indica donde estan ubicados los gdl en el robot siendo  $\theta_1$  rotativo y ubicado en la base del robot; para esta parte se utilizó un motor neumático, el segundo gdl  $\theta_2$  esta ubicado frente al robot y se encarga del desplazamiento vertical del robot, por ultimo  $\theta_3$  se encarga del desplazamiento vertical del link superior del robot.

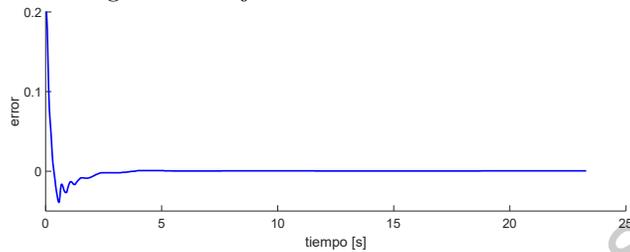
En la figura 4.9a se encuentra el resultado del primer grado de libertad, este fue el más exacto en el posicionamiento del robot, en las figuras 4.9b y 4.9c se encuentran los resultados de los ángulos  $\theta_2$  y  $\theta_3$  su sincronización fue más difícil de llevar a cabo.

#### 4.5 Trabajos a futuro

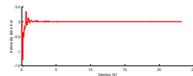
Concluido este trabajo, se pretende realizar un análisis de trayectorias de todos los ejes, además se pretende comenzar a trabajar con 4 gdl y el control de la pinza. Otro aspecto importante a ser tomado en cuenta es el guardado de parámetros de las posiciones de los encoders con el fin de no



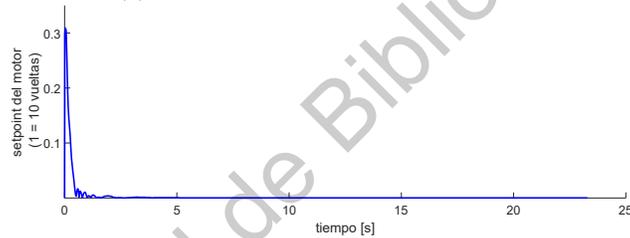
(a) Grafica del posicionamiento de un grado de libertad con ganancias ajustables.



(b) Grafica del error.



(c) Grafica de la derivada del error.



(d) Setpoint de posición para los motores de 1 gdl.

Figura 4.7: Graficas de los datos mas importantes del posicionamiento de un grado de libertad con ganancias ajustables.

manipular manualmente los elementos del sistema ya que esto a la larga daña tanto los motores como las bases que los sostienen. Aunque se considera varios tipos de control inteligente solo se pudo probar el de las ganancias ajustables con redes neuronales, es por esto, que se contempla con el desarrollo de nuevos sistemas de control inteligente mas sofisticados como el ANFIS y desarrollados en hardware configurable.

Por ultimo otro hardware contemplado y que se espera se incluya en un futuro es el uso de tarjetas FPGA con una mini-computadora base linux, como lo es la DE10-nano tarjeta con la que también se cuenta. Esto podria ser demasiado útil para poder desarrollar mas rápido y de forma eficiente los controladores y ser graficadas en tiempo real, ya que se pueden utilizar lenguajes como C, C++ y Python. La comunicación entre la FPGA y el procesador fue algo que se esperaba pudiese ser aplicado a este proyecto, pero debido a que se requiere más tiempo para su aprendizaje quedo pospuesto.

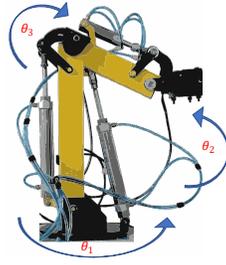
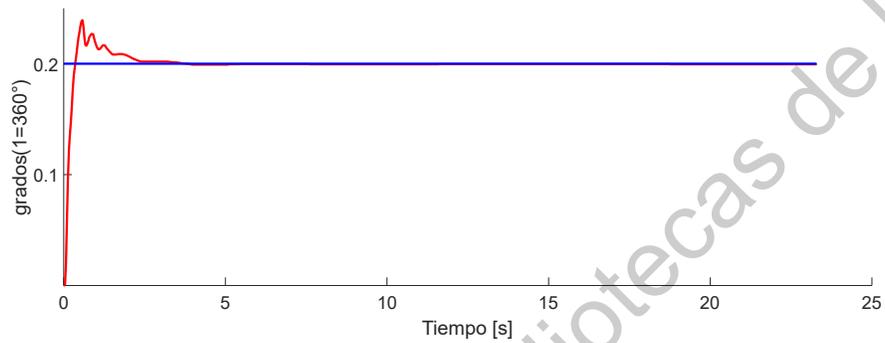
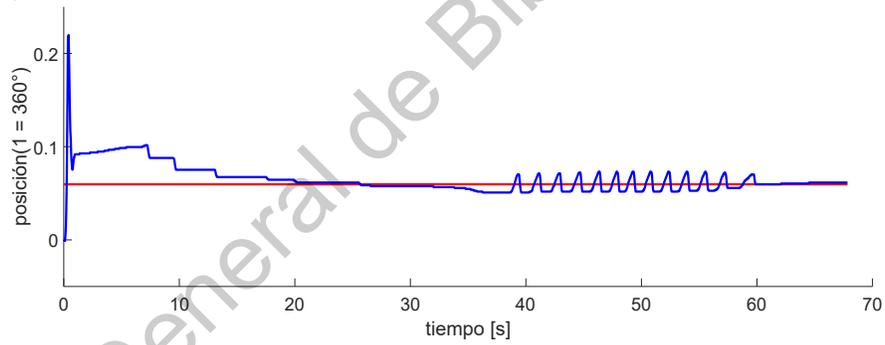


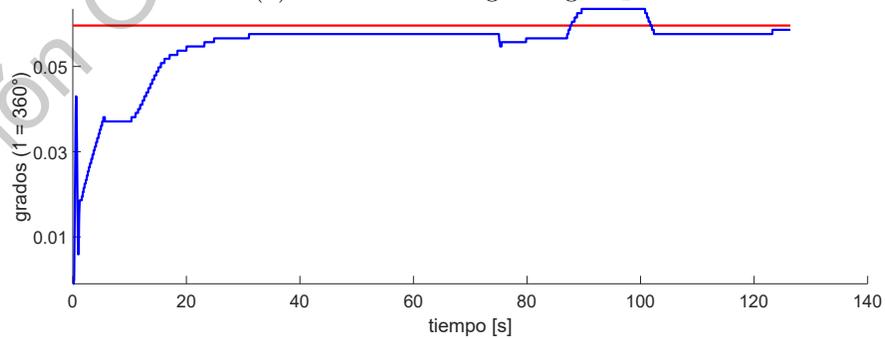
Figura 4.8: Ubicación de los gdl en el robot.



(a) Resultado del primer gdl, ubicado en la base del robot, este es rotativo y se representa con  $\theta_1$ .



(b) Resultado del segundo gdl  $\theta_2$ .



(c) Resultado del tercer grado de libertad  $\theta_3$ .

Figura 4.9: Resultados del desplazamiento del robot con 3 gdl.

Dirección General de Bibliotecas de la UAQ

---

## Conclusión

Uno de los puntos fuertes de este proyecto es el control de flujo de aire, el diseño del controlador fue eficiente para la aplicación de este proyecto. El aprendizaje del lenguaje VHDL fue de largo trabajo pero se alcanza el nivel requerido para el desarrollo de todos los códigos que se desarrollaron en este proyecto, se puede decir que se puede desarrollar de cualquier código que se implemente en un microcontrolador, pero con las ventajas del FPGA. Durante el curso del Dr. Romero Troncoso se aprendió a automatizar el código VHDL mediante scripts desarrollados en Matlab, creando diseños propios que permiten la configuración rápida de módulos de comunicación, lecturas de encoders, controladores, etc. Con solo llenar unos parámetros; se considera esto uno de los principales logros de aprendizaje de este proyecto.

La parte del control de los grados de libertad fue sumamente desafiante, teniendo resultados ambiguos pero concluyentes, la utilización de algoritmos inteligentes facilita el control de estos, sin la necesidad de profundizar demasiado en el desarrollo matemático, aun así, este diseño puede llegar a ser complicado si no se conoce el comportamiento del sistema, primero se indago en el resultado de un solo grado de libertad obteniendo buenos resultados con un controlador clásico, mejoro bastante su respuesta con la red neuronal.

Por ultimo se tienen los resultados de los 3 grados de libertad, hubo algunas complicaciones como falta de material y algunas piezas del robot faltaron, haciendo tiempo para solventar estos problemas, aun así el resultado no fue tan preciso con los últimos dos grados de libertad. Pero los resultados parciales muestran un comportamiento aceptable y de tener más tiempo se obtendria mejores resultados.

Dirección General de Bibliotecas de la UAQ

---

# Bibliografía

- Åström, K. J. y Hägglund, T. (2009). *Control PID avanzado*. Pearson, Madrid.
- Bobál, V., Böhm, J., Fessler, J., y Macháček, J. (2006). *Digital self-tuning controllers: algorithms, implementation and applications*. Springer Science & Business Media.
- Caicedo Bravo, E. F. y López Sotelo, J. A. (2009). *Una aproximación práctica a las redes neuronales artificiales*. Programa Editorial UNIVALLE.
- Camacho Angulo, Pedro Alonso; Ramos Arreguin, J. M. (2019). Design of a digital system for air flow control in pneumatic actuators.
- Chia-Chung, S., Shang-Te, C., Chu, C.-Y., Meng-He, W., y Chen, J.-H. (2020). Load balancing device for robot arm. US Patent App. 16/193,623.
- Cruz, P. P. (2011). *Inteligencia artificial con aplicaciones a la ingeniería*. Alfaomega.
- Da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., y dos Reis Alves, S. F. (2017). Artificial neural network architectures and training processes. En *Artificial neural networks*, pp. 21–28. Springer.
- De Silva, C. W. (1995). *Intelligent control: fuzzy logic applications*. CRC press.
- Delgado, J. A. P., Arreguín, J. M. R., Soto, G. I. P., y Hurtado, E. G. (2018). Diseño, modelo cinemático y simulación de un robot neumático de 4 dof (design, kinematic model and simulation of a 4 dof pneumatic robot). *Pistas Educativas*, 40(130).
- Dorf, R. y Bishop, R. H. (2005). *Sistemas de Control Moderno*. PRENTICE HALL, Ribera del Loira, 28 28042 Madrid(España), décima edición.
- Gaspar-Badillo, J., Ramos-Arreguin, J., Macias-Bobadilla, G., Talavera-Velazquez, D., Rivas-Araiza, E., y Víctor-Alexis, H. (2017). Four dof pneumatic robot design and hardware interface. En *2017 XIII International Engineering Congress (CONIIN)*, pp. 1–7.
- Gayo Martínez, J. M. (2013). Diseño de un brazo industrial neumático. Tesis de máster.
- González Islas, E. (2019). La era de los robots. url: <https://www.milenio.com/negocios/la-era-de-los-robots>.
- Gorrostieta, E. y Vargas Soto, E. (2008). Fuzzy algorithm of free locomotion for a six legged walking robot. *Computación y Sistemas*, 11(3):260–287.

- Hesselroth, T., Sarkar, K., Van Der Smagt, P. P., y Schulten, K. (1994). Neural network control of a pneumatic robot arm. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(1):28–38.
- Huifeng, H. y Chen, C. (2018). ‘Made In China 2025’: a peek at the robot revolution under way in the hub of the ‘world’s factory’. url: <https://www.scmp.com/economy/china-economy/article/2164103/made-china-2025-peek-robot-revolution-under-way-hub-worlds>.
- IFR (2018). Executive summary world robotics 2018 service robots.
- IFR (2020). Executive summary world robotics 2020 service robots.
- Kaneishi, D., Matthew, R. P., Estrada, M. K., y Tomizuka, M. (2019). A semi-active control method for pneumatic actuators with evolutionary algorithm. En *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 428–433. IEEE.
- Katzschmann, R. K., Thieffry, M., Goury, O., Kruszewski, A., Guerra, T.-M., Duriez, C., y Rus, D. (2019). Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer. En *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 717–724. IEEE.
- Kelly, R. y Santibáñez, V. (2003). *Control de movimiento de robots manipuladores*. Pearson educación.
- Kiyama, F. F. y Vargas, E. (2004). Modelling of a single degree of freedom flexible arm pneumatically operated. *International Symposium on Robotics and Automation*, pp. 25–27.
- Klafter, R. D., Chmielewski, T. A., y Negin, M. (1989). Robotic engineering: an integrated approach. *Robotic Engineering: An Integrated Approach, by Richard D. Klafter, Thomas A. Chmielewski, and Michael Negin, Englewood Cliffs, New Jersey: Prentice Hall, ISBN 0134687523*.
- Lee, Lian-Wang and Chiang, Hsin-Han and Li, I and others (2019). Development and control of a pneumatic-actuator 3-dof translational parallel manipulator with robot vision. *Sensors*, 19(6):1459.
- Li, Y., Chen, Y., y Li, Y. (2019). Pre-charged pneumatic soft gripper with closed-loop control. *IEEE Robotics and Automation Letters*, 4(2):1402–1408.
- Lilly, J. H. (2011). *Fuzzy control and identification*. John Wiley & Sons.
- Miramontes, F. F. K. y Soto, J. E. V. (2003). Modelación de la operación neumática de un brazo manipulador. *Tc*, 2:2.
- Mu, S., Goto, S., Shibata, S., y Yamamoto, T. (2019). Intelligent position control for pneumatic servo system based on predictive fuzzy control. *Computers & Electrical Engineering*, 75:112–122.
- Nguyen, H. T., Prasad, N. R., Walker, C. L., y Walker, E. A. (2002). *A first course in fuzzy and neural control*. CRC press.
- Ogata, K. (2005). *Ingeniería de control moderna*. PRENTICE HALL, Calle One Lake Nueva Jersey 07458, quinta edición.

- Passino, K. M., Yurkovich, S., y Reinfrank, M. (1998). *Fuzzy control*, volumen 42. Citeseer.
- Ramos-Arreguin, J. M., Pedraza-Ortega, J. C., Gorrostieta-Hurtado, E., Romero-Troncoso, R., Vargas-Soto, J. E., y Hernandez-Hernandez, F. (2008). Pneumatic fuzzy controller simulation vs practical results for flexible manipulator. *Automation and Robotics*, 5:191–200.
- Ramos-Arreguin, J.-M., Tovar-Arriaga, S., Vargas-Soto, J., y Aceves-Fernandez, M. (2016). Fpga embedded pd control of a 1 dof manipulator with a pneumatic actuator. *International Journal of Robotics and Automation*, 31.
- Ross, T. J. (2005). *Fuzzy logic with engineering applications*. John Wiley & Sons.
- Sandler, B. Z. (1999). *Robotics: designing the mechanisms for automated machinery*. Elsevier.
- Sánchez Solar, S., Pedraza Ortega, J. C., Gorrostieta-Hurtado, E., y Ramos-Arreguín, J.-M. (2016). *Path Tracking Simulation for a Two-Degree-of-Freedom Pneumatic Manipulator Robot*.
- Troncoso, R. d. F. R. (2007). *Electrónica digital y lógica programable*. Universidad de Guanajuato.
- Trumić, M., Jovanović, K., y Fagiolini, A. (2020). Comparison of model-based simultaneous position and stiffness control techniques for pneumatic soft robots. En *International Conference on Robotics in Alpe-Adria Danube Region*, pp. 218–226. Springer.
- Tuvayanond, W. y Parnichkun, M. (2017). Position control of a pneumatic surgical robot using pso based 2-dof h loop shaping structured controller. *Mechatronics*, 43:40–55.
- Wang, T., Chen, X., y Qin, W. (2019). A novel adaptive control for reaching movements of an anthropomorphic arm driven by pneumatic artificial muscles. *Applied Soft Computing*, 83:105623.
- Yusof, A. A., Asohkumaran, S. K., y Ramli, M. A. A. (2019). Non-static vision guided pneumatic pick and place robotic system with shared supervisory control. En *Symposium on Intelligent Manufacturing and Mechatronics*, pp. 580–598. Springer.
- Zhang, Y., Chen, X., Wang, H., y Zhang, L. (2019). A study on the control of a sensorless pneumatic joint using predictive control method. *IEEE Access*, 7:59923–59932.
- Zhong, J. y Zhao, C. (2019). A phenomenological model-based controller for position tracking of a pneumatic muscle actuator driven setup. *IEEE Access*, 7:45662–45669.
- Zöllner, G., Wall, V., y Brock, O. (2020). Active acoustic contact sensing for soft pneumatic actuators.

Dirección General de Bibliotecas de la UAQ

# Publicaciones

Se obtuvieron dos publicaciones con este trabajo una en la revista *MECAMEX* (figura [A.1](#)) y otra en el congreso *CICOS* (figura [A.2](#)).



Figura A.1: Publicación en *MECAMEX*.

### Positioning of a 3-degrees-of-freedom robot with pneumatic actuators

Marco-Antonio Cabrera-Rufino<sup>1</sup>, Juan-Manuel Ramos-Arreguin<sup>1</sup>\*, Saul Toval-Arriaga<sup>2</sup>, Efraim Gorrostieta-Hurtado<sup>4</sup>, Marco-Antonio Aceves-Fernandez<sup>2</sup>

Universidad Autónoma de Querétaro, Facultad de Ingeniería

\*Corresponding author's email: jsstidg@yahoo.com.mx

#### ABSTRACT

Interest in the use of pneumatic actuators has become increasingly significant, especially because of the potential benefits of employing clean energies. There are previous works related to this project, some made abroad that innovates in control methods but only in specific applications, and others in Mexico that progressively increase the complexity of the system. In this paper, the difficulty of a generic three-degrees-of-freedom (DoF) robot arm control is tested, searching for the most efficient method to reproduce the positioning of the final effector (a gripper) accurately. Furthermore, the ambitious goal of the project is to adequately develop an algorithm capable of being properly adjusted to any multipurpose robot system with pneumatic actuators. It is carefully considered the various non-linearities of the system. This work relied mostly on intelligent control techniques and modern programmable devices, the intention is to achieve similar or better results than previous projects. The possible combination of intelligent and even classical control can reasonably achieve the estimated conclusion, the first being flexible for developing robust design for non-linear systems.

#### RESUMEN

El interés por el uso de actuadores neumáticos se ha vuelto cada vez más importante, especialmente debido a los beneficios de emplear energías limpias. Existen relacionados con este artículo, algunos hechos fuera del país donde se innova en métodos de control, pero solo en aplicaciones específicas, y otros elaborados dentro del país donde se incrementa la complejidad del sistema y control. En este trabajo se prueba la dificultad de un brazo robótico genérico de tres grados de libertad (DoF), se busca el método más eficiente de reproducir el posicionamiento del efector final (una pinza) de una forma precisa. Además, la meta de este proyecto es de crear un algoritmo capaz de ser ajustado a cualquier sistema robótico multipropósito con actuadores neumáticos de tres grados de libertad. Se pretende obtener resultados similares o mejores que proyectos anteriores. Los resultados preliminares muestran

Figura A.2: Publicación en *CICOS*.