



Universidad Veracruzana



Universidad Veracruzana

CENTRO DE INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL

**Implementación de Control Inteligente
para la Manipulación de Objetos con
una Pinza de Movimiento Angular en
Robots Móviles**

T E S I S

Que para obtener el Grado de:

Doctor en Inteligencia Artificial

Presenta:

Sergio Hernández Méndez

Maestro en I.A.

Director de tesis:

Dr. Antonio Marín Hernández

Co-Director de tesis:

Dra. Elvia Ruth Palacios Hernández

Xalapa-Enríquez, Veracruz, México, 2018

Abstract

In this paper, the design, implementation and analysis of three types of controllers, applied to a robotic gripper with two independent fingers with angular movement is described. Force and position sensors are mounted on both tips in the robotic gripper. The values of these sensors are the inputs for the controllers. Each finger of the gripper is controlled independently, in order to maintain the desired position and force to manipulate determined objects.

The PID controller is the first controller proposed in this work, this is one widely used due to its simplicity. This controller has to be properly tuned in its proportional, integral and derivative gains in order to get a correct operation. The position-force control is carried out by reducing the error between the value of the desired reference and the output of the system.

Intelligent controllers are the second and third controller proposed in this work, which are based on fuzzy logic type I and type II respectively. These controllers regulate the position and strength of the fingers by means of a series of rules and an inference machine. When considering the control by means of diffuse logic, the decisions that would normally be applied by an expert in the area are executed. For the input set of these controllers trapezoidal fuzzy sets are used, and for the output the singleton sets are used. In addition to the fuzzy Type-II controller, the Karnik-Mendel (KM) type reduction algorithm is implemented.

The contributions of this work are the following: a) the design and implementation of the controllers in the robotic gripper, b) the development of the kinematic equations and the configuration space of the gripper, c) the development of a control package and visualization in the environment of ROS and finally d) the comparison of these three controllers to determine among them, the best to execute the task of manipulation of objects.

Resumen

En este trabajo se presenta el diseño, implementación y análisis de tres tipos de controladores, aplicados a una pinza robótica de dos dedos independientes con movimiento angular. La pinza robótica utilizada cuenta tanto con sensores de fuerza, como de posición; los valores de dichos sensores son las entradas para los controladores. Se controla cada dedo de la pinza de manera independiente, manteniendo la posición y fuerza deseada para manipular ciertos objetos determinados.

El primer controlador que se propone es uno ampliamente usado debido a su simplicidad; éste es el controlador PID. Para su correcto funcionamiento, este controlador tiene que ser sintonizado adecuadamente en sus ganancias proporcional, integral y derivativa. El control de posición-fuerza se realiza reduciendo el error entre el valor de la referencia deseada y la salida del sistema.

El segundo y tercero, son controladores inteligentes, que están basados en lógica difusa tipo I y tipo II respectivamente. Estos controladores regulan la posición y fuerza de los dedos por medio de una serie de reglas y una máquina de inferencia. Al considerar el control por medio de lógica difusa se ejecutan las decisiones que normalmente aplicaría un experto en el área. Para el conjunto de entrada de estos controladores se utilizan conjuntos difusos trapezoidales, y para la salida conjuntos singleton. Además para el controlador difuso Tipo-II se implementó el algoritmo de reducción de tipo Karnik-Mendel (KM).

Las contribuciones de este trabajo son las siguientes: a) el diseño y la implementación de los controladores en la pinza robótica, b) el desarrollo de las ecuaciones cinemáticas y el espacio de configuraciones de la pinza, c) el desarrollo de un paquete de control y visualización en el entorno de ROS y por último d) la comparación de estos tres controladores para determinar de entre ellos, el mejor que ejecutara la tarea de manipulación de objetos.

Agradecimientos

Quiero dedicar este trabajo a mi familia, con su apoyo y motivación me impulsaron a terminar una etapa más en mi formación profesional. Agradecer a Carolina Maldonado Méndez, por todo el tiempo que compartimos juntos en el doctorado, sus comentarios, ideas que fueron siempre muy valiosas para mí.

A mis revisores de tesis, la Dra. Ericka Rechy Ramírez, Dr. Homero Vladimir Ríos Figueroa, Dr. Raúl Hernández Montoya, y el Dr. Jesus Savage Carmona gracias por sus comentarios sobre el trabajo de mi tesis.

La Dra. Elvia Ruth Palacios, gracias por permitirme trabajar a su lado durante la estancia de investigación en la Universidad Autónoma de San Luis Potosí.

Y por último agradecer especialmnte al Dr. Antonio Marín Hernández por su apoyo y consejos durante todo el doctorado, la confianza que ha tenido en mi me sirvió para crecer como investigador y alcanzar mis objetivos.

Índice general

Resumen	II
Indice de figuras	VII
Acrónimos	XII
Lista de variables	XIII
1. Introducción	1
1.1. Planteamiento del Problema	2
1.2. Hipótesis	3
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos particulares	3
1.4. Consideraciones	3
1.5. Contribución	4
1.6. Publicaciones	4
1.7. Organización de la tesis	5
2. Trabajos Relacionados	7
2.1. Pinzas robóticas	7
2.1.1. Tipos de Pinzas	8
2.1.2. Tipos de Agarre	9
2.1.3. Sensores para Pinzas	10
2.2. Control en pinzas robóticas	12
3. Conceptos Fundamentales	22
3.0.1. Control Clásico	23
3.0.1.1. Control PID	23
3.1. Control inteligente	24
3.1.1. Lógica difusa tipo-I	25
3.1.1.1. Conjuntos difusos	25
3.1.1.2. Funciones de Membresía	26
3.1.1.3. Controlador difuso tipo I	27
3.1.1.4. Fusificación	28
3.1.1.5. Base de conocimiento	28
3.1.1.6. Lógica de decisiones (máquina de inferencia)	28

3.1.1.7.	Defusificación tipo I	29
3.1.2.	Lógica difusa tipo II	29
3.1.3.	Controlador difuso tipo II	30
3.1.3.1.	Reductor de tipo	31
3.1.3.2.	Algoritmo de Karnik-Mendel.	32
3.1.3.3.	Defusificación tipo II	33
4.	Descripción de los componentes de la Pinza Robótica	34
4.1.	Componentes Mecánicos	35
4.1.1.	USB2 Dynamixel	36
4.2.	Componentes Electrónicos	37
4.2.1.	Sensor de fuerza	37
4.2.1.1.	Caracterización del sensor de fuerza	38
4.2.2.	Tarjeta de adquisición de datos	39
4.2.2.1.	Convertidor Analógico Digital externo	41
4.3.	ROS	42
4.3.1.	Paquete <i>Rosserial</i>	43
4.3.2.	Paquete <i>Dynamixel_motor</i>	44
4.3.3.	Estructura de comunicación de la pinza con ROS	45
4.4.	Análisis Cinemático de la Pinza	46
4.4.1.	Espacio de Configuraciones	49
5.	Diseño de los controladores	54
5.1.	Control Híbrido PID	54
5.1.1.	Control PID de Posición	55
5.1.2.	Control PID de Fuerza	55
5.2.	Control Difuso tipo I	56
5.3.	Control difuso Tipo-II	64
6.	Experimentos y Resultados	74
6.1.	Experimento 1	74
6.1.1.	Controlador Híbrido	75
6.1.2.	Controladores Difusos Tipo-I y Tipo-II	76
6.2.	Experimento 2.	80
7.	Conclusiones y Trabajo Futuro	87
7.0.1.	Trabajo futuro	89
A.	Apéndice A	90
A.1.	B	90
B.	Apéndice B	92
C.	Implementación en ROS	97
D.	Moveit!	104
D.0.1.	Planificadores Moveit	105
D.0.2.	Pasos para configurar el asistente de Moveit! con el brazo	106

Índice de figuras

1.1.	Proceso de manipulación tomado de [26].	2
2.1.	Diferentes tipos de pinzas robóticas a) Mano shadown (5 dedos)[20], b) Meka Robotics H2 (4 dedos) [29], c) Mano barret (3 dedos) [5], d) Pinza paralela (2 dedos)[79], e) Pinza angular de dos dedos, f) Pinza Universal [11].	9
2.2.	a) Ejemplo de agarre envolvente con la mano robótica de la DLR [9]. b) Agarre de precisión, hecho con una mano robot de tres dedos [59]	10
2.3.	a) Esquema de control para la mano robótica empleado en [78]. Los bloques amarillos representan el control PID para cada dedo, los bloques naranjas los sensores de posición. b) Prototipo de mano robótica, cada dedo cuenta con su propio servomotor.	13
2.4.	a) Prototipo de pinza de tres dedos de [70]. Los dedos fueron creados en una impresora 3D. El motor de esta pinza cuenta con un encoder que le permite utilizarlo como sensor de posición. b) Diagrama de bloques del sistema de control PID empleado en la pinza robótica.	14
2.5.	Arquitectura de control propuesta en [10] para un solo dedo. FLC1 es el encargado de aplicar la fuerza suficiente y FLC2 se encarga de controlar el deslizamiento del objeto.	14
2.6.	a) Prototipo de pinza de tres dedos empleado en [25]. Esta pinza fue creada en una impresora 3D y controlado por un solo motor. b) Diagrama de bloques de todo el sistema de control	15
2.7.	Esquema de control difuso propuesto en [18]. En su esquema dos niveles de retroalimentación son considerados <i>finger tactile feedback</i> que leer información del sensor táctil en cada dedo y <i>hand tactile action</i> lee la posición de cada uno de los dedos.	16
2.8.	a) La estructura del dedo controlado por un servomotor utilizado en el trabajo de [74]. b) El desplazamiento en la posición del dedo, es un cambio de PWM para el servomotor.	16
2.9.	Esquema de control propuesto en [47]. Su estrategia propuesta consta de un modelo de referencia de Impedancia, un controlador difuso y sensores de posición y fuerza.	17
2.10.	a) Prototipo de pinza de dos dedos utilizado en [76]. b) Esquema de control implementado en Matlab, el bloque <i>SystemDynamics</i> contiene las ecuaciones dinámicas de la pinza mostradas en [76]. El bloque <i>NonlinearSprings</i> representa las características de deformación de fuerza en los resortes de cada uno de los dedos.	18
2.11.	a) Estructura del sistema físico de la pinza, así como el acoplamiento de la señal enviada a la computadora empleado en [58]. b) Diagrama de bloques del sistema de control PD empleado en la pinza.	19

2.12. Prototipo de pinza paralela controlada por un solo motor empleado en [82].	19
2.13. a) Estructura de la pinza robotica de dos dedos empleada en [52]. Un motor de DC acciona los dos dedos mediante un sistema de engranajes b) . Esquema de control empleado en la pinza robótica.	20
3.1. Esquema control de un sistema.	22
3.2. PID respuesta	24
3.3. Ejemplo de un conjunto de temperatura que cuenta con tres subconjuntos $\{Media, Alta, Baja\}$	26
3.4. Esquema de un controlador basado en lógica difusa tipo-I.	27
3.5. Bloque de entrada de fusificación.	28
3.6. Función de membresia alta (UMF) y función de membresia baja (LMF) de \tilde{A}	31
3.7. Controlador difuso tipo-II[56].	31
3.8. a) Computo de yl , cambia el nivel de disparo superior al nivel de disparo inferior. b) Computo de yr , cambia el nivel de disparo inferior al nivel de disparo superior.	33
4.1. Pinza angular	34
4.2. Motor Dynamixel AX-12.	36
4.3. Conector USB2 Dynamixel.	37
4.4. sensor fuerza	37
4.5. Cilindro de foamy sobre el sensor (a), pesas para la caracterización del sensor (b), báscula digital (c).	40
4.6. Gráfica de caracterización de la señal de salida del sensor de fuerza FSS-SMT	40
4.7. Tarjeta Arduino Mega.	41
4.8. a) Tarjeta del convertidor analógico digital ADS1015. b) diagrama a bloques de la estructura interna del ADC. Las entradas AIN0-AIN-3, corresponden a las entradas Analógicas 0-3 respectivamente.	42
4.9. Componentes de electrónicos, mecánicos y de software empleados en la pinza robótica.	43
4.10. Estructura de comunicación entre los nodos de ROS.	45
4.11. Pinza angular Dynamixel, a) medidas de los dedos de la pinza, b)trayectoria circular que describe cada dedo en su movimiento, se establece como origen del marco de referencia el motor derecho.	46
4.12. Propuestos los valores x_i y y_i , y una distancia de separación cero $D = 0$, se hallan los valores x_d y y_d . La línea verde representa el dedo izquierdo y la roja el dedo derecho. La posición de los dedos apunta al lado derecho.	50
4.13. EL punto de intersección se presenta cuando los dos dedos se juntan en la punta, esto es cuando no hay objeto entre ellos.	50
4.14. Dados unos valores x_i y y_i , y una distancia de separación cero $D = 0$, en este caso los dedos apuntan al lado izquierdo.	51
4.15. Todas las posibles posiciones que pueden tener los puntos x, y de cada uno de los dedos.	51
4.16. Se grafican los ángulos que van de $0 - 180^\circ$ para ambos motores el punto de intersección se genera en 100° motor_derecho, 79° motor_izquierdo.	52

4.17. a) Cuando el objeto se encuentra en medio de la pinza, b) el objeto se encuentra desplazado a la derecha c) La posición cuando el objeto se encuentra desplazado a la izquierda	52
4.18. Ángulos obtenidos con distintas separaciones entre los ejes de los motores, dg, a) 1cm. b) 5cm. c) 7cm. d) 9cm. e) 10cm.	53
5.1. Esquema de control que cambia de Posición a Fuerza. El control de posición centra el objeto en medio de la pinza. Una vez que el objeto es centrado, el control de posición es apagado, y enciende el control de fuerza, entonces el objeto puede ser manipulado.	57
5.2. control difuso de la pinza.	57
5.3. Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Derecho.	58
5.4. Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Izquierdo.	59
5.5. Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Izquierdo.	60
5.6. Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Izquierdo.	61
5.7. a) Ejemplo de agarre envolvente con la mano robótica de la DLR [9]. b) Agarre de precisión, hecho con una mano robot de tres dedos [59]	62
5.8. Salida del controlador difuso en función de las variables de entrada de motor derecho vs sensor derecho.	63
5.9. Salida del controlador difuso en función de las variables de entrada de los sensores derecho e izquierdo.	63
5.10. Funciones de pertenencia altas y bajas, para sensor y motor derecho. . . .	64
5.11. Funciones de pertenencia altas y bajas, para sensor y motor derecho. . . .	66
5.12. Funciones de pertenencia altas y bajas, para sensor y motor derecho. . . .	68
5.13. Funciones de pertenencia altas y bajas, para sensor y motor derecho. . . .	69
5.14. Operacion meet usando el operador mínimo t-norm en nuestro controlador difuso tipo II.	71
6.1. En la primera fila se muestra cómo el objeto está centrado en el medio de la pinza. En la segunda fila se muestra cómo se levanta el objeto de la mesa.	76
6.2. En la gráfica de arriba se muestran las lecturas del sensor de posición que se usan para ajustar el PID de posición. En la gráfica de abajo se muestran las lecturas del sensor de fuerza que se utilizan para ajustar el PID de fuerza.	77
6.3. En la gráfica de arriba se muestra el error de la posición y fuerza del PID. En el siguiente gráfica se muestra la salida de la posición PID y la fuerza. . . .	77
6.4. Posiciones de los motores durante el experimento. En la gráfica de arriba se muestran las lecturas de posición del motor izquierdo. En la parte de abajo se muestran las posiciones del motor derecho.	78
6.5. Salidas de los controladores difusos Tipo-I y Tipo-II. Estas salidas son comandos de velocidad que se envían a los motores. En la parte de arriba se muestran las salidas de los controladores para el motor derecho. En la parte de abajo se observan las salidas de los controladores para el motor izquierdo	79

6.6.	Lecturas de los sensores, en la parte de arriba se muestran las lecturas del sensor derecho, en la parte de abajo las lecturas del sensor izquierdo.	80
6.7.	Brazo de tres grados de libertad, colocado en un robot móvil (iRobot Create). Sobre el la parte posterior del robot se coloca una placa de adquisición de datos, así como una batería de 12V para alimentar el sistema.	81
6.8.	Secuencia del experimento de tomar un objeto. En la fila de arriba se aprecian las etapas 1 (a-c) y parte de la etapa 2 (d-h), en la fila de abajo se aprecia al final la etapa 3 (i-j).	82
6.9.	Posición de cada motor en las diferentes etapas del experimento.	83
6.10.	Desempeño de cada controlador por cada etapa. En la gráfica de arriba se muestra la respuesta del sensor de fuerza derecho. En la gráfica de abajo, se muestra la respuesta del sensor de fuerza izquierdo.	84
6.11.	Salida de los controladores en comandos de velocidad. En la parte de arriba se muestra la gráfica de los comandos de velocidad del motor derecho para cada uno de los controladores. En la parte de abajo se muestra la gráfica velocidad del motor izquierdo para cada uno de los controladores	85
A.1.	Motores Dynamixel.	90
A.2.	Carcaterísticas Motores Dynamixel.	91
B.1.	Base	92
B.2.	Unión-Hombro	93
B.3.	Brazo	93
B.4.	Antebrazo	93
B.5.	Base codo-muñeca	93
B.6.	Bracket de la muñeca	94
B.7.	Brazo con todas las piezas	94
B.8.	Impresión de la pieza llamada antebrazo.	95
B.9.	a) Motor dynamixel conectado a la pieza antebrazo, b) pinza montada sobre la pieza antebrazo.	95
B.10.	Ensamble de los dos motores dynamixel MX-106T	96
B.11.	Prototipo de brazo finalmente armado	96
C.1.	Estructura del diseño del brazo y comunicación con ROS.	97
C.2.	Joint.	98
C.3.	Cuando RViz es ejecutado los nodos llamados <i>joint_state_publisher</i> , <i>robot_state_publisher</i> y <i>RViz</i> , y los topics llamados <i>joint_states</i> y <i>tf</i> , son transportados siguiendo la dirección de las flechas.	98
C.4.	a) Modelo 3D del brazo en FreeCad, b) Medidas y nombres de cada una de las partes que se asignan en ROS.	100
C.5.	Secuencia de armado de los joints y links del brazo robótico.	101
C.6.	Esquema de todas las transformaciones del brazo desde el <i>base.link</i> hasta <i>grasping.frame</i>	102
C.7.	Visualización en RViz del prototipo de brazo de 3 grados de libertad.	103
D.1.	Diversas plataformas robóticas que emplean movit!.	105
D.2.	Diagrama de la descripción del <i>planning scene</i> en Moveit!.	105
D.3.	Asistente de configuración de movit!.	106

D.4. Carga con éxito el modelo de robot en la herramienta Asistente de configuración.	107
D.5. Regenerando la matriz de auto colisión.	108
D.6. Agregando grupo de planificación al brazo.	108
D.7. Agregando grupo de planificación al brazo y el gripper.	109
D.8. Agregando pose inicial del brazo.	109
D.9. Agregando end effector.	110
D.10. Generando el paquete de configuración de Moveit!.	110
D.11. Interfaz de planificación de movimiento de Moveit! en RViz.	111
D.12. Arquitectura del hardware y software con los paquetes de ROS.	111

Acrónimos

DOF	D egree O f F redoom
PID	P roporcional I ntegral D erivate
ROS	R obot O perating S ystem
SMT	S urface M ount T echnology
ADC	A nolog D igital C onverter
UMF	U pper M embership F unction
LMF	L ower M embership F unction
FOU	F ootprint of U ncertainty
SPS	S amples P er S econd
PGA	P rogrammable G ain A mplifier
PWM	P ulse W idth M odulation
URDF	U nified R obot D escription F ormat
IAE	I ntegral A bsolute E rror
ITAE	I ntegral T ime A bsolute E rror
SDA	S ensor D erecho A pagado
SDTS	S ensor D erecho T ocando S uave
SDTF	S ensor D erecho T ocando F uerte
SDAP	S ensor D erecho A pretando
SIA	S ensor I zquierdo A pagado
SITS	S ensor I zquierdo T ocando S uave
SITF	S ensor I zquierdo T ocando F uerte
SIAP	S ensor I zquierdo A pretando
MDMA	M otor D erecho M uy A bierto
MDA	M otor D erecho A bierto
MDC	M otor D erecho C entrado
MDCR	M otor D erecho C errado
MIMA	M otor I zquierdo M uy A bierto
MIA	M otor I zquierdo A bierto
MIC	M otor I zquierdo C entrado
MICR	M otor I zquierdo C errado

Lista de variables

x_d	Posición x dedo derecho
y_d	Posición y dedo derecho
x_i	Posición x dedo izquierdo
y_i	Posición y dedo izquierdo
r	radio
D	Distancia entre las puntas de los dedos
dg	Distancia entre los ejes de los motores
K_p	Ganancia Proporcional
K_i	Ganancia Integral
K_d	Ganancia Derivativa
U	Universo de discurso
mV	milivolts
$\mu\bar{A}(x)$	Función de membresía superior
$\mu\underline{A}(x)$	Función de membresía inferior

Capítulo 1

Introducción

Los robots de servicio son cada día más comunes en ambientes humanos. Para que estos robots ofrezcan sus servicios de manera eficiente y robusta estos deberán desarrollar diversos tipos de tareas [48],[75]. Una de las principales tareas para estos robots es la de manipular diversas clases de objetos cotidianos, como por ejemplo: vasos, lápices, cucharas, libros, perillas de puertas, entre otros. Todo esto con la finalidad de poder interactuar, ya sea con humanos, con otros robots o con el ambiente.

De acuerdo con [26], la manipulación robótica de objetos puede ser dividida en cuatro fases (Fig. 1.1):

- *Pre-agarre*. En esta fase el brazo es dirigido desde su posición inicial a una posición cercana al objeto. Para lograr este objetivo es común usar un planificador de trayectorias.
- *Adquisición de agarre*. La pinza hace contacto con el objeto y puede desplazarlo incluso romperlo. Para evitar este tipo de problemas se tiene que utilizar algún controlador de fuerza-posición.
- *Post-agarre transporte*. En esta fase el objeto es manipulado; el objeto es cambiado desde una posición inicial y en dirección a alguna otra posición deseada.
- *Meta*. Finalmente en esta fase, el objeto es depositado correctamente en el lugar objetivo.

En la tarea de manipulación robótica, la selección de las pinzas robóticas juega un papel muy importante, ya que es mediante esta pinza que el robot podrá interactuar con el ambiente. Para poder realizar eficientemente esta tarea la pinza robótica debe estar dotada de diferentes tipos de sensores, los cuáles pueden ser de fuerza, posición,

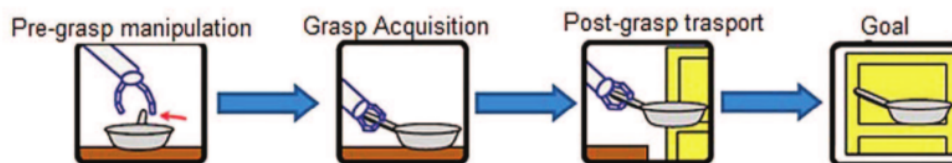


FIGURA 1.1: Proceso de manipulación tomado de [26].

temperatura, textura o inercial. Estos sensores le permiten al robot obtener información del objeto y del medio ambiente; y aunque ya se tenga información *a priori* sobre el tipo de objeto (forma, textura, puntos de contacto), para realizar la tarea de manera eficiente, se requiere de un controlador de fuerza que le permita al robot tomar dicho objeto.

Para el caso en el cual se tenga una pinza con un motor para cada dedo, como es el caso que se presenta aquí, es importante calibrar estos sensores para que actúen de manera simultánea y acoplada, y así lograr un agarre eficaz. Esto es un problema complejo que no siempre puede ser representado en un modelo matemático de forma precisa, ya que influyen diversos factores como pueden ser, sincronización de frecuencias de comandos, desgaste diferente en cada motor, respuestas diferenciadas de cada motor, entre otras.

En este trabajo se propone diseñar, implementar y evaluar tres tipos de controladores: un controlador PID, un controlador Difuso Tipo-I y un controlador Difuso tipo-II. Todos estos para una pinza robótica de dos dedos con movimiento angular independiente.

Por un lado, dentro de los controladores clásicos se encuentra el controlador de tipo Proporcional Integral y Derivativo o PID, que realiza un buen desempeño para reducir el error de un sistema dado. Desafortunadamente estos controladores muestran una deficiencia cuando al sistema se le presentan perturbaciones externas como ruido, cambios de parámetros, interferencias, etc.

Por otro lado, con el control inteligente, como lo es un controlador difuso; este puede tomar el conocimiento de un experto, el cual mediante una serie de reglas y una máquina de inferencia, toma las decisiones que normalmente tomaría una persona.

1.1. Planteamiento del Problema

Se desea que un robot móvil pueda manipular objetos de manera robusta. Para esto se considera una pinza robótica de dos dedos independientes. Este tipo de pinza requiere de un controlador capaz de realizar una sincronización de cada uno de los dedos y así poder manipular un objeto y transportarlo a una ubicación deseada.

1.2. Hipótesis

El diseño de controladores inteligentes mejorará el desempeño de un control clásico para manipular objetos con una pinza robótica de dos dedos con movimiento independiente.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar el control con retroalimentación de posición-fuerza de una pinza robótica de dos dedos para la manipulación de objetos.

1.3.2. Objetivos particulares

- Evaluar y caracterizar el hardware que se necesitara para la pinza robótica.
- Desarrollar el modelo cinemático de la pinza para obtener el espacio de configuraciones de esta.
- Diseñar un brazo en el cual se sujetará la pinza robótica.
- Desarrollar e implementar un controlador Híbrido PID.
- Desarrollar e implementar uno o más controladores inteligentes.
- Implementar y evaluar los controladores de la pinza.
- Analizar los resultados y obtener las conclusiones.

1.4. Consideraciones

Para este trabajo de investigación se tomaran las siguientes consideraciones:

- Los objetos a utilizar son considerados objetos rígidos no deformables.
- El peso y el tamaño de los objetos es conocido.
- Cada dedo es no redundante.
- El ambiente de trabajo es semi-controlado.
- Todos los puntos de contacto son conocidos.

1.5. Contribución

En esta tesis se realizaron diversos procedimientos para lograr los objetivos propuestos. A continuación se presentan las contribuciones mas relevantes de este trabajo.

- Diseño e implementación de un controlador Híbrido PID de posición-fuerza, un controlador Difuso Tipo-I y un controlador Difuso Tipo-II; para poder manipular objetos con una pinza robótica de dos dedos independiente.
- Desarrollo de las ecuaciones cinemáticas y el espacio de configuraciones de la pinza.
- Desarrollo de un paquete de control y visualización en el entorno de ROS.
- Comparación de estos tres controladores para determinar de entre ellos, el mejor que ejecutara la tarea de manipulación de objetos, en términos de minimizar la señal de error en los controladores de fuerza y posición al trasladar un objeto de una posición inicial a una posición meta.

1.6. Publicaciones

Durante el desarrollo de esta investigación se obtuvo la producción de manera directa de las siguientes publicaciones:

- **S. Hernandez-Mendez**, A. Marin-Hernandez, E. R. Palacios-Hernandez, and H. Vazquez-Leal, "Characterization of two force sensors to be used in a robotic hand", in Electronics, Communications and Computers (CONIELECOMP), 2015 International Conference on, Feb 2015, pp.155-160.
- **S. Hernandez-Mendez**, A. Marin-Hernandez, E. R. Palacios-Hernandez, and K. L. Luna-Gallegos, "A switching position/force controller for two independent finger gripper over ros", in 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP), Feb 2017, pp. 1-6.
- **S. Hernandez-Mendez**, C. Maldonado-Mendez, A. Marin-Hernandez, and H. V.Rios-Figueroa, "Detecting falling people by autonomous service robots: A ros module integration approach". In 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP) (Feb 2017), pp. 1-7.

-
- **S. Hernandez-Mendez**, C. Maldonado-Mendez, A. Marin-Hernandez, H. V. Rios-Figueroa, H. Vazquez-Leal, and E. R. Palacios-Hernandez. "*Design and Implementation of a Robotic Arm using ROS and MoveIt!*". In 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (Nov 2017), ACEPTADO.

Otras Publicaciones

Además de las publicaciones anteriores, se colaboro de manera indirecta en las siguientes publicaciones durante mis estudios de Ph.D.

- K. L. Luna-Gallegos, E. R. Palacios-Hernandez, **S. Hernandez-Mendez**, and A. Marin-Hernandez, "*A proposed software architecture for controlling a service robot*". In 2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (Nov 2015), pp. 1-6.
- C. Maldonado-Mendez, **S. Hernandez-Mendez**, Ana Luisa Solis, H. V. Rios-Figueroa and A. Marin-Hernandez, "*The effects of using a noise filter and feature selection in action recognition: an empirical study*". International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE) (Nov 2017), ACEPTADO.

1.7. Organización de la tesis

Esta tesis consta de siete capítulos, los cuales están organizados de la siguiente manera:

- El capítulo 2, trata sobre el estado del arte de trabajos relacionados a diferentes técnicas de control empleadas en el uso de pinzas para la manipulación.
- En el capítulo 3, se establecen las bases necesarias para el entendimiento de las técnicas utilizadas en este trabajo.
- La estructura de la pinza robótica es definida en el capítulo 4; sección en la cual se incluye un análisis cinemático de la pinza, el espacio de configuraciones y el hardware correspondiente a este (sensores y actuadores).
- En el capítulo 5 se detalla el diseño de los controladores propuestos y en el capítulo 6 se muestran los desempeño y resultados de los controladores durante los experimentos propuestos de manipulación.

- Finalmente en el capítulo 7 se comentan las conclusiones obtenidas de los experimentos y se mencionan posibles trabajos futuros

Capítulo 2

Trabajos Relacionados

La manipulación de objetos es un área muy activa de investigación. Particularmente se desea que los robots de servicio puedan interactuar en un ambiente no controlado y ejecutar tareas exitosamente [41]. Para cumplir este objetivo se han desarrollado muchas plataformas robóticas, desde humanoides [43], hasta robots más personalizados [51] [73], cada uno de ellos con diversos enfoques y propuestas para resolver específicamente el problema de manipulación robótica. Estas plataformas deben contar con algún mecanismo que les permita llevar a cabo el agarre de objetos diversos. En [30] se presenta, un análisis interesante de diferentes manos robóticas realizadas en los últimos 30 años. En dicho trabajo, se muestra una tabla comparativa de las características particulares que pueden presentar las manos robóticas, i.e. el número de dedos, los tipos de actuadores, los grados de libertad en cada dedo o los sensores. En las siguientes secciones se explican las pinzas comúnmente empleadas en robótica así como el tipo de control empleado en estas.

2.1. Pinzas robóticas

Las pinzas robóticas están compuestas por muchas partes, tanto a nivel de hardware así como software. En este capítulo se abordaran las cuatro características principales que las componen, las cuales son consideradas a nivel de diseño, y éstas son: los actuadores, el número de dedos, sensado y control.

De acuerdo con [57], una pinza (o gripper) puede definirse como: *Subsistemas de mecanismos de manipulación los cuales proveen contacto temporal con el objeto a ser tomado. Estos aseguran la posición y orientación cuando llevan y toman el objeto con el equipo de manejo. La presión se consigue por la fuerza producida y los elementos*

de forma que coinciden. El término pinza o gripper, es también usado en los casos en donde no hay un agarre real, sino que solo se mantiene el objeto por algún otro medio, por ejemplo en la succión, en donde la fuerza de retención puede actuar sobre un punto, línea o superficie.

2.1.1. Tipos de Pinzas

La forma de una pinza puede variar desde manos antropomórficas complejas hasta simples pinzas paralelas o pinzas sin dedos. Las manos antropomórficas son pinzas con cinco o cuatro dedos poliarticulados, cuya configuración es similar a la mano humana. Durante las últimas dos décadas se han construido diversos tipos de ellas y se han clasificado en función del grado de antropomorfismo [46]. La mayoría de las manos emplea un mecanismo de control que es accionado por cables (tendones) a través de actuadores que son colocados fuera del cuerpo principal de la mano (*Link-hosted actuation*)[55][20][36][63]. Sin embargo este mecanismo es complicado y todo el sistema es grande y pesado. Las manos antropomórficas Shadow Hand [20] y la mano Meka Robotics H2 [29] las cuales se pueden apreciar en la Fig. 2.1 a) y b) respectivamente. Pinzas con tres dedos [5] es el que se muestra en la Fig. 2.1 c). La pinza de dos dedos, de movimiento paralelo es la más común, generalmente los dedos son rígidos para sujetar el objeto, y tienen la particularidad de que sus dedos siempre se mantienen paralelos entre sí. La pinza paralela del robot PR2 [79], el cual se puede observar en la Fig. 2.1 d). La pinza de la Fig. 2.1 e) es una pinza de movimiento angular (o movimiento circular) con dos dedos. Cada dedo también es rígido, pero cada uno es independiente y se puede ajustar a los lados de una pieza desde varios ángulos, por ejemplo 30°, 40° o 80°, de acuerdo al espacio de trabajo de la pieza. En cuanto a las pinzas mencionadas de tres y dos dedos otra característica importante es que su mecanismo que lo acciona esta lo más cerca al dedo (*Direct-drive actuation*). Esto los hace más ligeros y prácticos, aunque cabe mencionar que también se han diseñado pinzas antropomórficas donde los actuadores se colocan lo más cerca posible al dedo [40] [12].

Por último la pinza (o gripper) universal[11] o de succión de aire, se muestra en la Fig. 2.1 f). Esta está compuesta por un pequeño saco flexible relleno de material granulado (café, por ejemplo) y cuando es presionado sobre el objeto a sujetar, fluye alrededor de este conforme a su forma.

Las manos complejas ofrecen mayor generalidades, pero a menudo las manos simples o pinzas simples, son más prácticos[53].

Muchas de estas manos también son empleadas para uso de prótesis, en [6] se presenta un trabajo de las características de diversas manos protéticas.

Durante las últimas cinco décadas se ha generado bastante literatura relacionada con investigaciones sobre técnicas de manipulación y tipos de pinzas (simples y antropomórficas) [54][8][72].



FIGURA 2.1: Diferentes tipos de pinzas robóticas a) Mano shadow (5 dedos)[20], b) Meka Robotics H2 (4 dedos) [29], c) Mano barret (3 dedos) [5], d) Pinza paralela (2 dedos)[79], e) Pinza angular de dos dedos, f) Pinza Universal [11].

2.1.2. Tipos de Agarre

En la actualidad las pinzas robóticas incluyen desde solo dos dedos (pinza), hasta manos robóticas muy elaboradas con cinco dedos. En función de esto se pueden presentar distintos tipos de agarre. En [21] se presentan los diferentes tipos de agarre, los cuales están básicamente divididos en dos grandes grupos: agarre envolvente (*enveloping grasp*, *power grasps*) y agarre de precisión (*precision grasps*). En el agarre envolvente tanto la palma como los dedos son usados. Este tipo de agarre tiene un alto nivel de contacto de superficie entre el objeto y la pinza, cuando se desea alta estabilidad, este agarre

puede ser usado. Basado en este enfoque en el trabajo de [34], se presenta un análisis teórico para la manipulación de objetos. Por otro lado el agarre de precisión usa solo las puntas de los dedos y es particularmente muy útil cuando se necesita una alta precisión en la manipulación del objeto. En [19] se muestra un análisis de agarre simulado para la mano robótica de la DLR empleando la técnica de *precision grasps*. Cabe mencionar, que aún dentro de este último tipo de agarre, otros autores aun lo dividen en otro llamado *soft-fingers*, el cual es más suave al momento de manipular objetos [61][38].

También puede darse la combinación de ambas técnicas como es el caso del trabajo presentado en [65] donde inicia con un agarre de precisión y posteriormente se convierte en un agarre envolvente.

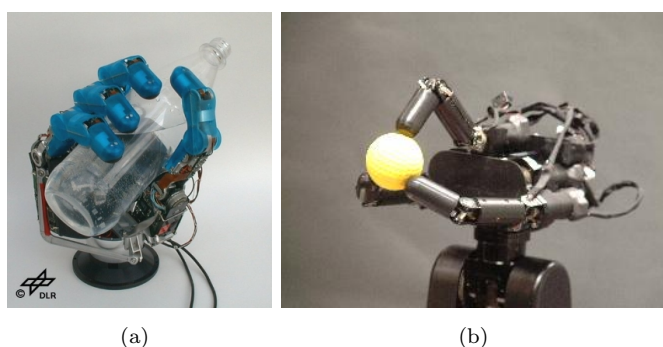


FIGURA 2.2: a) Ejemplo de agarre envolvente con la mano robótica de la DLR [9]. b) Agarre de precisión, hecho con una mano robot de tres dedos [59]

Ambos tipos de agarre envolvente y de precisión son divididos en varias subcategorías dependiendo del dedo y posición de la palma.

En [8] se expone otra clasificación más general para la manipulación de objetos la cual es dividida en dos categorías: "*dexterous grasping*" que significa manipulación hábil con los dedos, cuya metodología de agarre consiste en imitar el agarre humano para tomar distintos tipos de objetos. Y por otro lado esta la "*fixturing*" que se relaciona con la acción de contener o sujetar diversos objetos en tareas de ensamblado o acciones de traslado de un lugar a otro sin considerar la habilidad con que se toma el objeto.

2.1.3. Sensores para Pinzas

Los sensores permiten al robot conocer su estado y percibir el entorno en el cual se desarrollan. Los sensores en pinzas robóticas pueden ser clasificados en dos grandes categorías: *propioceptivos* y *exteroceptivos*. Los primeros miden información física relacionada con el estado del dispositivo mismo (posición-velocidad), los segundos miden los datos relacionados a la interacción ambiente/objeto (fuerza aplicada, torque, fricción y forma).

En manipulación robótica, la pinza más los sensores *propioceptivos* y *exteroceptivos* nos permiten realizar dos acciones muy importantes:

- *presión*: la habilidad de tomar y sostener con la suficiente fuerza sin dañar el objeto de diferentes tamaños y formas.
- *aprensión*: la capacidad de comprender o captar información a través del tacto (textura, temperatura).

En cuanto a sensores que permiten detectar diversas características del objeto manipulado se encuentran por ejemplo el [50] para detectar textura, [15] forma, fuerza [71], temperatura [81] infrarrojo [64]. Algunos de los sensores más utilizados en pinzas son: capacitivos, resistivos, piezoresistivos, ópticos, ultrasónicos, magnéticos y piezoelectrónicos. Un aspecto a considerar en la selección de los sensores es el espacio limitado con el que se cuenta. La siguiente tabla tomada de [22] muestra algunas características importantes de estos sensores.

El trabajo de [3] hacen una revisión sobre sensores de fuerza aplicados en manipulación y presentan una tabla con algunos trabajos de los últimos 20 años de sensores de presión utilizados en dedos y palmas robóticas.

Tipo	Características a favor	Características en contra
Resistivo	sensitivo, bajo costo	Generalmente detectan un solo punto de contacto, alto consumo de energía
Piezoresistivo	bajo costo, electrónica simple, buena sensibilidad, bajo ruido	Sensible a temperatura, respuesta no lineal, histéresis Frágil
Capacitivo	Bajo costo, sensitivo, disponibilidad comercial	Histéresis electrónica compleja
Óptico	Sin interconexiones, rápido sensitivo, flexible, inmune a la interferencia electromagnética	Voluminoso, computo complejo, alto consumo de energía
Ultrasónico	Respuesta dinámica rápida, buena resolución de fuerza	Electrónica compleja, sensible a la temperatura
Magnético	Alta sensibilidad, buen rango dinámico, robustez física	Sufre de interferencia magnética, cómputo complejo, voluminoso
Piezoelectrico	Respuesta dinámica, alto ancho de banda	Sensible a la temperatura
Goma conductora	Físicamente flexible	Respuesta no lineal, histéresis mecánica

CUADRO 2.1: Sensores en pinzas robóticas.

En cuanto a revisión sobre el estado de arte de sensores ya se han hecho varios trabajos desde los 80's [23] [62]. También en [13] además de hacer una revisión de tipos sensores empleados en pinzas antropomórficas, muestran la caracterización de un sensor táctil para ser usado en la mano MAC-HAND.

Y más recientemente un excelente compendio sobre diversas tecnologías de sensores empleados en diversas plataformas de manos robóticas [69]

2.2. Control en pinzas robóticas

A continuación se presentan algunos trabajos en los cuales se implementan distintos tipos de controladores en diferentes pinzas robóticas, para la manipulación de objetos. Trabajos teóricos y experimentales en control para la manipulación se han enfocado en lograr los movimientos deseados de los dedos sobre el objeto. Por ejemplo en el control de fuerza y sus variantes están aun en desarrollo, éstos incluyen control cartesiano [68], de impedancia [7], control híbrido de posición/fuerza [80] y combinación de estos [45]. Por otra parte, actualmente se ha tenido interés en nuevas estrategias de control, las

cuales emplean alguna técnica de inteligencia artificial, por ejemplo lógica difusa. La lógica difusa se está empleando en diversos problemas de robótica, sobre todo en el área de manipulación autónoma de robots [60] [24] [27].

En [77], desarrollan un controlador de una pinza (gripper) de dos dedos para la manipulación de objetos deformables. Los autores proponen controladores Proporcional-Integral (PI) paralelos de fuerza con modelos de predicción con el objetivo de regular la fuerza que ejerce el robot sobre el objeto; y así, reducir el riesgo de daño a los objetos durante el agarre.

En el trabajo de [78] simulan y construyen un prototipo de una mano robot con cinco dedos, Fig.2.3 b). Para controlar los movimientos cinemáticos y dinámicos en la trayectoria de cada dedo, se aplica un control Proporcional, Integral y Derivativo (PID) para cada uno de los dedos, Fig. 2.3 a). En sus experimentos muestran la posición angular en la trayectoria de cada dedo que se mueve rápidamente, hacia el ángulo objetivo con una señal de error y sobretiro pequeños.

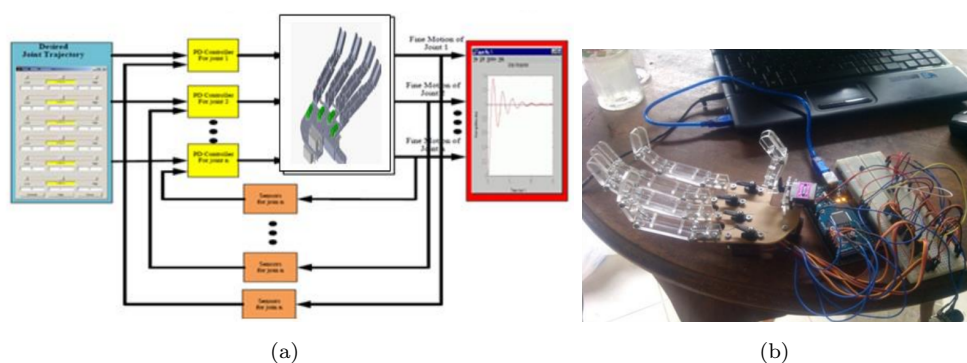


FIGURA 2.3: a) Esquema de control para la mano robótica empleado en [78]. Los bloques amarillos representan el control PID para cada dedo, los bloques naranjas los sensores de posición. b) Prototipo de mano robótica, cada dedo cuenta con su propio servomotor.

En [32], realizan un análisis matemático de las fuerzas de contacto para un dedo robótico de 3 grados de libertad. En este trabajo los autores simulan dos controladores PID, uno para posición y otro para fuerza. Las ganancias de estos controladores son sintonizadas mediante lógica difusa.

Para realizar tareas de manipulación con objetos sólidos (redondos y rectangulares), en [70] se presentan una pinza de tres dedos, Fig.2.4 a). Para controlar esta pinza los autores emplean solamente un control PID en posición para los tres dedos, Fig.2.4 b). Dos dedos se encuentran alineados y uno enfrente de ellos. Un motor actúa el movimiento de los tres dedos al cerrar, y un segundo motor mueve el dedo opuesto para poder alinear los tres dedos. Debido a la complejidad dinámica del sistema la sintonización de este controlador fue hecha mediante prueba y error. El objetivo del controlador PID es

poder minimizar el sobretiro, tiempo de establecimiento, y el error en estado estacionario durante la manipulación de objetos.

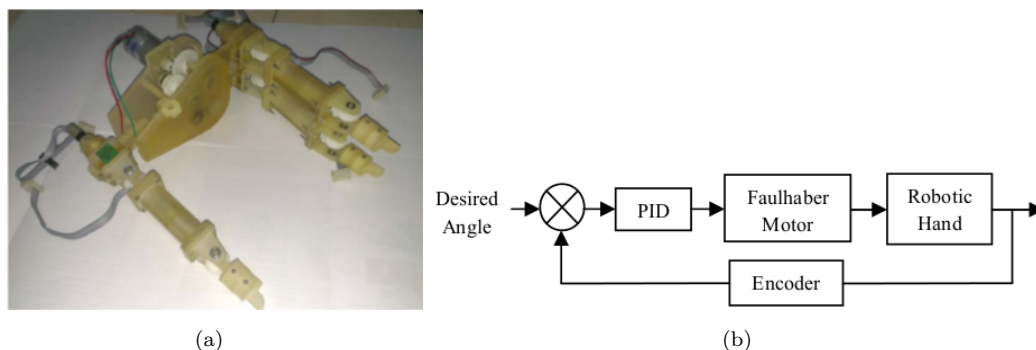


FIGURA 2.4: a) Prototipo de pinza de tres dedos de [70]. Los dedos fueron creados en una impresora 3D. El motor de esta pinza cuenta con un encoder que le permite utilizarlo como sensor de posición. b) Diagrama de bloques del sistema de control PID empleado en la pinza robótica.

En [1], se presenta un análisis del efecto de tres diferentes funciones de pertenencia (triangular, trapezoidal y gaussiano) en un esquema de control difuso para un sistema de tres dedos.

Un control difuso es presentado en [10], con el fin de manipular objetos. Este controlador consta de una combinación de dos controladores difusos. El primer controlador es usado para asegurar un agarre estable del objeto y el segundo controlador para evitar el deslizamiento del objeto. Las pruebas de este controlador solamente son simuladas en Matlab con un solo dedo, y los autores suponen que el controlador tendrá el mismo comportamiento con los otros dedos. En la Fig. 2.5 se muestra la arquitectura de control para un dedo.

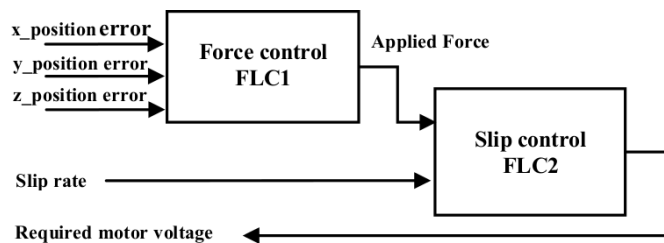
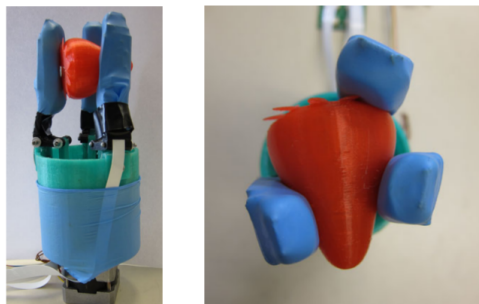


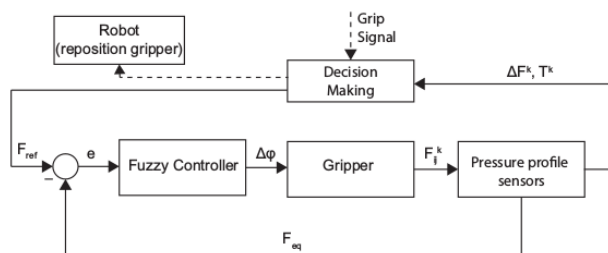
FIGURA 2.5: Arquitectura de control propuesta en [10] para un solo dedo. FLC1 es el encargado de aplicar la fuerza suficiente y FLC2 se encarga de controlar el deslizamiento del objeto.

En [25], desarrollan una pinza de tres dedos con el objetivo de manipular fresas sin dañarlas, Fig 2.6 a). Ellos emplean un controlador difuso, en el cual se determina la fuerza que se debe ejercer sobre la fresa a partir de las lecturas de un sensor capacitivo

colocado en cada dedo Fig. 2.6 b). La salida del controlador es la posición de un motor el cual los tres dedos de manera simultánea.



(a)



(b)

FIGURA 2.6: a) Prototipo de pinza de tres dedos empleado en [25]. Esta pinza fue creada en una impresora 3D y controlado por un solo motor. b) Diagrama de bloques de todo el sistema de control

En el trabajo de [18] simulan un controlador difuso con información táctil para manipular objetos con una pinza de dos dedos, Fig.2.7. El objetivo de emplear un controlador difuso es para garantizar un movimiento suave de los dedos al tocar el objeto, así como tratar con las incertidumbres y no linealidades del sensor. Con esto logran reducir la necesidad de conocer las dimensiones exactas de los objetos así como la posición relativa de la pinza respecto al objeto. La principal función de su controlador es enviar comandos de posición a cada uno de los dedos con el objetivo de alcanzar una cierta posición en la tarea de manipulación. Las entradas del controlador son las lecturas del sensor táctil, así como las lecturas de las posiciones de cada uno de los dedos. El sensor táctil es capaz además de detectar microvibraciones las cuales les permiten establecer si hay deslizamiento del objeto. La simulación de su trabajo fue hecha en Matlab y muestra excelentes resultados pero desafortunadamente no lo implementaron en una pinza real.

Para manipular objetos desconocidos los autores en [74] proponen un controlador difuso con una pinza accionada por un solo servomotor, Fig. 2.8. La entrada del controlador difuso es la señal que provee un sensor de fuerza resistivo, y la salida de este controlador genera un ángulo al cual debe moverse el dedo. Para llegar al ángulo deseado se envía un señal PWM al servomotor y este la ejecuta. Muestran la efectividad del controlador con tres diferentes tipos de objetos suave, moderado y duro.

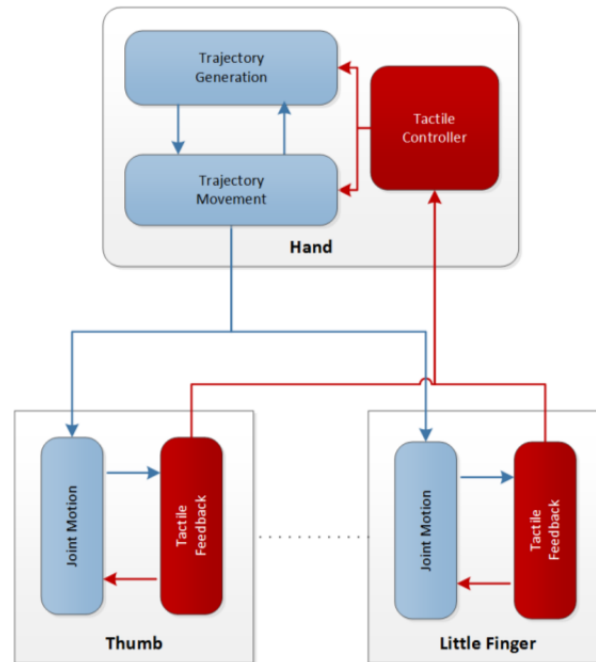
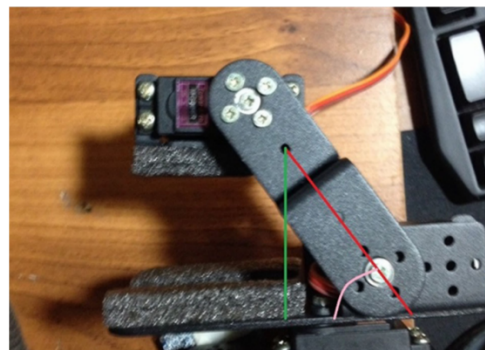
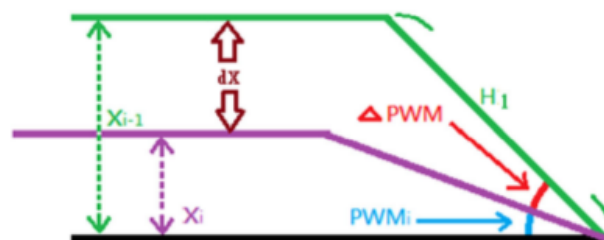


FIGURA 2.7: Esquema de control difuso propuesto en [18]. En su esquema dos niveles de retroalimentación son considerados *finger tactile feedback* que leer información del sensor táctil en cada dedo y *hand tactile action* lee la posición de cada uno de los dedos.



(a)



(b)

FIGURA 2.8: a) La estructura del dedo controlado por un servomotor utilizado en el trabajo de [74]. b) El desplazamiento en la posición del dedo, es un cambio de PWM para el servomotor.

En [47], presentan un controlador difuso para reducir el impacto de las fuerzas durante la manipulación de objetos o interacción con humanos. Los autores desarrollan un

control difuso de conformidad, en el cual un sistema masa-resorte-amortiguador les permite determinar un nivel de conformidad deseado. Dos sensores se usan principalmente para la medición de la retroalimentación de los controladores de posición y fuerza. Para probar el desempeño del controlador de conformidad difuso lo simulan en una mano robótica, la cual cuenta con sensores de fuerza y sensores de posición. En la Fig.2.9 se muestra el esquema de control empleado en su simulación.

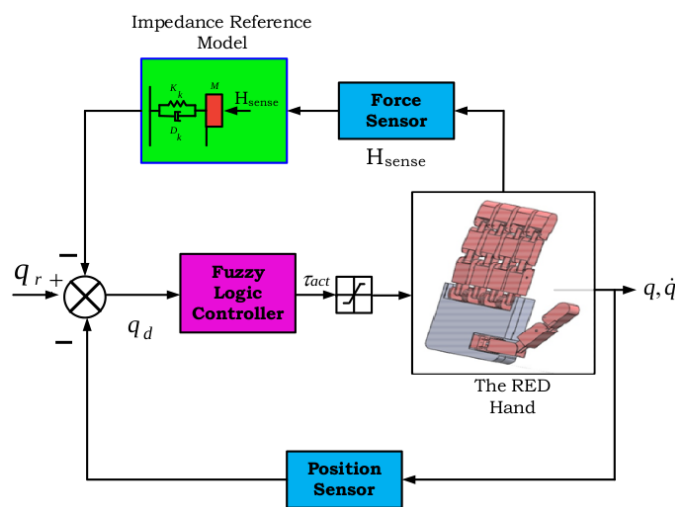


FIGURA 2.9: Esquema de control propuesto en [47]. Su estrategia propuesta consta de un modelo de referencia de Impedancia, un controlador difuso y sensores de posición y fuerza.

En [31] presentan un método de control difuso para manipular objetos. Para probar el desempeño de su control, realizan pruebas simuladas y lo comparan con un controlador PID. En su experimento simulan una pinza paralela al cual le han colocado sensores para detectar el deslizamiento del objeto. Durante la prueba el PID es sintonizado con el método de Ziegler-Nichols [33] pero su resultado no fue satisfactorio por que tenían que recalcular las ganancias cuando cambiaban de objeto. Así que tenían modificarlas mediante prueba y error hasta que lograban los resultados satisfactorios. Debido a este problema de recalcular las ganancias, el control difuso era mejor, ya que no requería hacer reajustes, y por otro lado el controlador difuso tuvo un mejor tiempo de respuesta que el PID.

El trabajo de [49] diseñó una pinza de dos dedos equipada con sensores táctiles y presentaron un algoritmo para mover un objeto a lo largo de una pista deseada. Para cada movimiento elemental, se calcula la ubicación y orientación del objeto, mientras que el sensor táctil detecta el punto de contacto. La ubicación conjunta de los dedos está controlada basado en el análisis cinemático, para que el objeto no cambie la orientación moviéndose a lo largo de la pista deseada.

En [37] presentan un controlador PD con un sensor de fuerza para manipular objetos de manera diestra. La señal de retroalimentación para el controlador es la señal del sensor de fuerza. La principal desventaja de este trabajo es que un modelo matemático del sensor el cual debe incluir la masa del sensor y la rigidez del sensor.

En [76] para manipular objetos con una pinza de dos dedos , Fig. 2.10 a), emplean un controlador PID de fuerza Fig. 2.10 b). Los experimentos se basan primero en pruebas simuladas y posteriormente lo ejecutan en una pinza física, la cual es accionada por un solo motor. Los dedos de la pinza no son rígidos, cada uno tiene dos grados de libertad con lo cual le permite adaptarse a la forma del objeto.

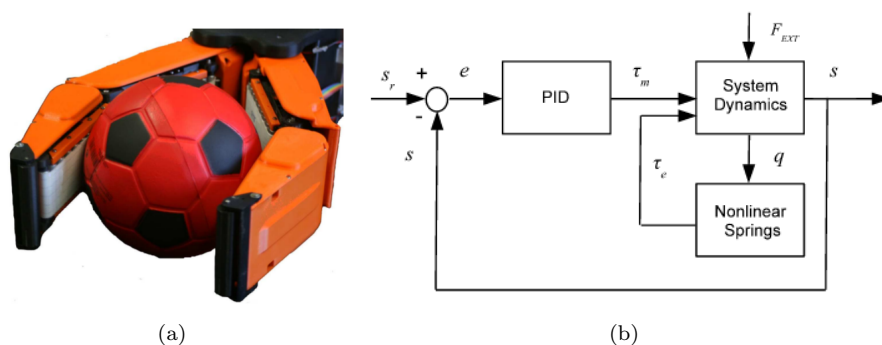


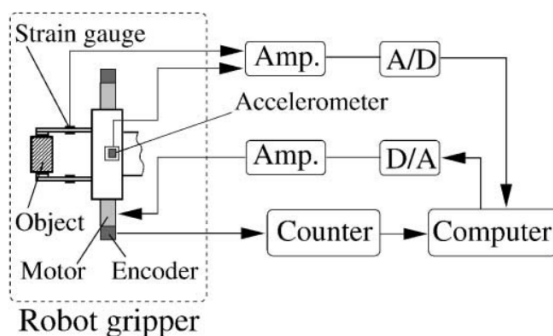
FIGURA 2.10: a) Prototipo de pinza de dos dedos utilizado en [76]. b) Esquema de control implementado en Matlab, el bloque *SystemDynamics* contiene las ecuaciones dinámicas de la pinza mostradas en [76]. El bloque *NonlinearSprings* representa las características de deformación de fuerza en los resortes de cada uno de los dedos.

En [35] se propone un experimento con sensores de fuerza en una pinza paralela de dos dedos. El objetivo es identificar las fases de manipulación (pre-agarre, agarre, post-agarre) y el comportamiento del sensor, para determinar que estrategia de control a aplicar (PID, Impedancia, y torque).

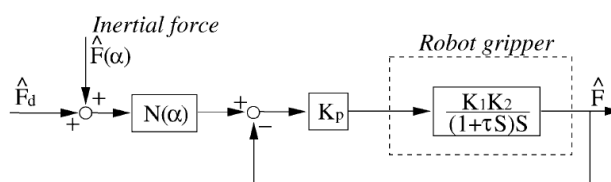
El trabajo de [58] presenta un control de fuerza para tomar objetos. Este control se aplica a una pinza paralela de dos dedos, cada dedo es controlado por un motor Fig. 2.11 a). El control a diferencia de los presentados anteriormente solo emplea un PD (Fig. 2.11 b)), para controlar la fuerza sobre le objeto, muestran resultados tanto simulados como implementados en la pinza.

Un controlador de fuerza para una pinza paralela es presentado en [82]. La pinza es controlada por un solo motor (Fig. 2.12), y el controlador está basado en lógica difusa. Las reglas están hechas para que la pinza aplique la fuerza suficiente sin dañarlo, la entrada de este controlador difuso es un sensor de fuerza resistivo. Realizan preliminarmente pruebas simuladas y después lo implementan en la pinza paralela.

Otro trabajo en el cual emplean un controlador con un sensor de fuerza resistivo [14]. La pinza cuenta con dos dedos paralelos controlados por un servo motor. El controlador



(a)



(b)

FIGURA 2.11: a) Estructura del sistema físico de la pinza, así como el acoplamiento de la señal enviada a la computadora empleado en [58]. b) Diagrama de bloques del sistema de control PD empleado en la pinza.

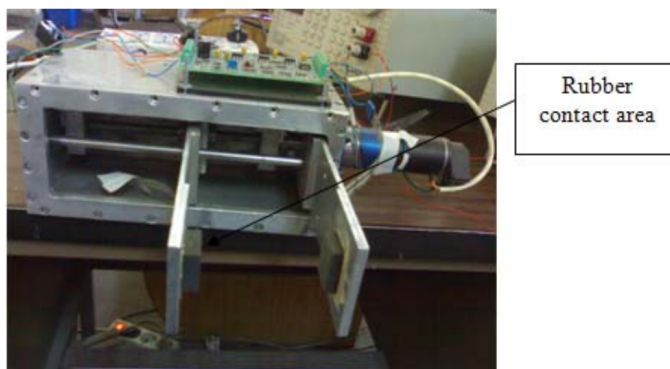


FIGURA 2.12: Prototipo de pinza paralela controlada por un solo motor empleado en [82].

que ocupan es un PID el cual modifican su parte de integración usando un método trapezoidal y otro rectangular. Las ganancias son calculadas a prueba y error.

Un controlador de fuerza basado en lógica difusa es presentado en [28]. En este trabajo diseñan un conjunto de reglas que le permitan al controlador aplicar la suficiente fuerza a un objeto con características desconocidas (masa). Realizan primero pruebas simuladas en Matlab y luego las implementan en el hardware. La entrada del controlador es la lectura de dos sensores uno de fuerza resistivo y otro fotoelástico, y son colocados en la punta de un dedo. La salida del controlador es un comando de velocidad que controla únicamente este dedo accionado por un motor DC.

Algunos de los trabajos mencionados anteriormente solo trabajan en simulación o sólo cuentan con un motor para accionar los dedos. En este trabajo se propone controlar de forma experimental una pinza de dos dedos, en el cual cada dedo tiene su propio motor que se puede controlar de manera independiente. Para esto se propone evaluar 3 controladores, PID, difuso Tipo-I y difuso Tipo-II con el objetivo de controlar posición y fuerza de cada motor de la pinza. Haciendo hincapié que el controlador difuso tipo-II, a la fecha de este trabajo no ha sido reportado en este tipo de sistemas. En el siguiente capítulo se explicará los conceptos elementales en control clásico así como los controladores difusos.

Capítulo 3

Conceptos Fundamentales

En este capítulo se explicaran los conceptos básicos relacionados con control clásico y control inteligente. Para el caso de control inteligente en especial los controladores empleados en esta tesis, es decir, controlador difuso Tipo-I y controlador difuso Tipo-II. A continuación se explican los términos básicos que se utilizan en sistemas de control general:

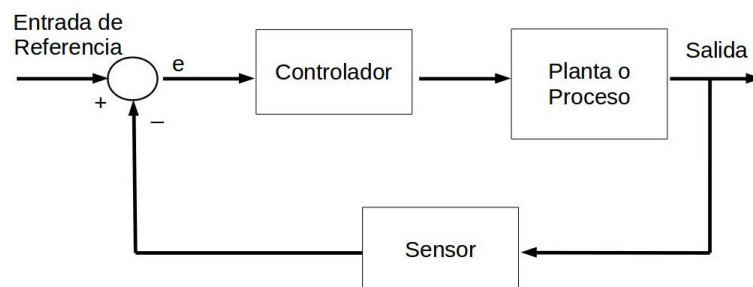


FIGURA 3.1: Esquema control de un sistema.

Planta o sistema a controlar: entidad física que se desea controlar. Consiste en un conjunto de elementos que actúan coordinadamente para realizar un objetivo determinado. Ejemplo de planta o sistema puede ser un motor, un horno, un sistema de disparo, un sistema de navegación, un tanque de combustible, etc.

Proceso: Un proceso se define como la sucesión de un conjunto de fases sucesivas de un fenómeno natural o de una operación.

Señal de salida: es la variable que se desea controlar (posición, velocidad, presión, temperatura, etc.). También se denomina variable controlada.

Señal de referencia: es el valor que se desea que alcance la señal de salida.

Señal de control: es la señal que produce el controlador para modificar la variable controlada de tal forma que se disminuya, o elimine, el error.

3.0.1. Control Clásico

3.0.1.1. Control PID

El control PID (Proporcional, Integral, Derivativo) combina tres acciones de control, que tienen las siguientes características :

- Un controlador proporcional (K_p) tendrá el efecto de reducir el tiempo de elevación y reducirá, sin jamás eliminar, el error de estado estacionario. Su ecuación que describe este controlador es:

$$u(t) = K_p e(t) \quad (3.1)$$

- Un control integral (K_i) tendrá el efecto de eliminar el error de estado estacionario, pero puede empeorar la respuesta transitoria.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (3.2)$$

- Un control derivativo (K_d) tendrá el efecto de incrementar la estabilidad del sistema, reduciendo el sobrepico, y mejorando la respuesta transitoria y sirve como predictor, pero amplifica el ruido de alta frecuencia no deseado.

$$u(t) = K_p e(t) + k_p T_d \frac{de(t)}{dt} \quad (3.3)$$

Al unir los tres controladores anteriores se genera el controlador PID cuya descripción matemática se encuentra en la mayoría de textos, y se muestra en la siguiente ecuación:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (3.4)$$

Donde T_i y T_d son el tiempo de integración y tiempo derivativo respectivamente. La expresión anterior puede igualmente expresarse como la siguiente función de transferencia del controlador PID, aplicando la transformada de Laplace:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (3.5)$$

El controlador PID es una estructura de control en la que la señal de control del proceso se expresa en función del error, $e(t) = y_{ref}(t) - y(t)$, según la expresión estándar.

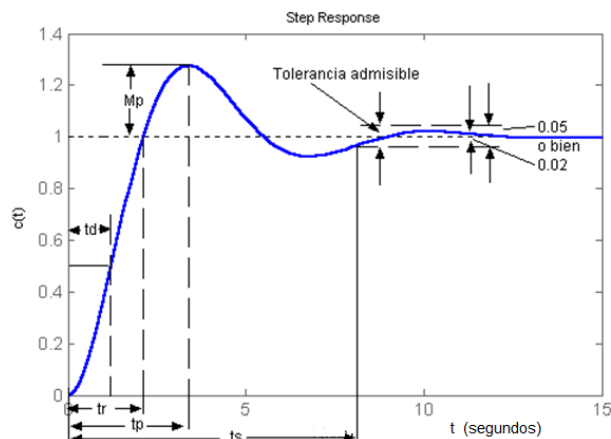


FIGURA 3.2: PID respuesta

Existen ciertas características que ayudan a calificar el desempeño de un sistema de control. Estas involucran ciertos parámetros asociados con la respuesta en el tiempo ante un escalón unitario, estas cantidades son representadas en la Fig. 3.2.

- tr : Tiempo de retardo. Es el tiempo requerido para que la respuesta alcance el 50% del valor total.
- tp : Tiempo de pico. Es el tiempo necesario que le lleva a la respuesta alcanzar el primer pico de sobreeleongación.
- Mp . Porcentaje de sobreeleongación máxima. La máxima sobreeleongación, es el pico más alto de la respuesta, medido a partir de la unidad.
- ta : Tiempo de asentamiento. Es el tiempo que le toma alcanzar a la respuesta el valor cercano al valor deseado, alrededor del 2% a 5%.

3.1. Control inteligente

El control inteligente, trata con el desarrollo de métodos de control para emular características importantes de la inteligencia humana:

- adaptación.
- aprendizaje.
- tratamiento de grandes cantidades de datos.
- tratamiento de incertidumbre.

En [4] se define el control inteligente a través de varias propiedades propias de los sistemas inteligentes:

- Adaptación y Aprendizaje: capacidad para adaptarse a condiciones cambiantes.
- Autonomía e Inteligencia: habilidad para actuar adecuadamente en un entorno con incertidumbre.
- Estructura y Jerarquías: Arquitectura funcional apropiada para afrontar problemas complejos.

3.1.1. Lógica difusa tipo-I

En el desarrollo de este controlador, no se requiere del modelo matemático del sistema. Por lo tanto este es un controlador heurístico basado en conocimiento, utilizado para controlar un sistema indefinido y complejo. En esta sección se explicará de manera breve la teoría necesaria para el diseño de un controlador difuso, conceptos y definiciones fundamentales que son necesarios para comprender los temas aquí mencionados.

3.1.1.1. Conjuntos difusos

Un conjunto difuso A de un *universo de discurso* U es caracterizado por una función de membresía $\mu:U \rightarrow (0, 1)$, que asocia a cada elemento u de U , un número $\mu(u)$ que puede tomar todos los valores reales comprendidos dentro del intervalo $(0,1)$, que representa el grado de membresía de u en A . El conjunto difuso de A de $U = u_1, u_2, \dots, u_n$ es expresado por:

$$A = \sum_{i=1}^n \mu(u_i)/u_i = \sum_i (u_i) \quad (3.6)$$

donde \sum representa unión.

A este conjunto se le asocia un determinado *valor lingüístico* definido por una palabra o *etiqueta lingüística*, donde está el nombre del conjunto. Para cada conjunto se define una función de membresía denominada $\mu_{A(x)}$, ésta indica el grado de pertenencia que tiene la variable x en el conjunto representado por la etiqueta A , si la función $\mu_{A(x)}$ vale 0, indica que tal valor de x no pertenece al conjunto A , por el contrario si vale 1, el correspondiente valor de x se encuentra totalmente en A , es decir $(0 \leq \mu_{A(x)} \leq 1)$. En la Fig.3.3 se observa un ejemplo del conjunto de temperatura con variable x , el cual está subdividido en 3 subconjuntos $\{Media, Alta, Baja\}$.

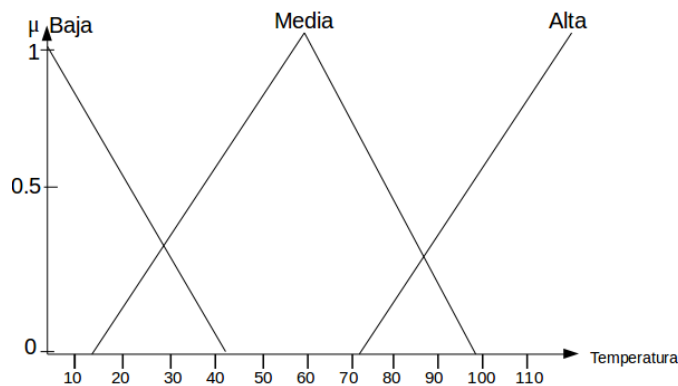


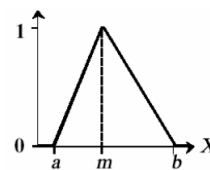
FIGURA 3.3: Ejemplo de un conjunto de temperatura que cuenta con tres subconjuntos $\{Media, Alta, Baja\}$.

3.1.1.2. Funciones de Membresía

El grado de membresía en los conjuntos difusos, puede ser representado por una función continua, la cual es denominada como función de membresía. En otras palabras, si F es un conjunto difuso, entonces la función de membresía $\mu_F(x)$ mide el grado con el cual el valor x pertenece al conjunto F , de la cual puede tomar diferentes formas, dependiendo de la aplicación en particular. Algunas de las funciones más conocidas son: función singleton, cuyo valor de membresía es igual a 1 para un solo punto y el resto es cero, función trapezoidal, triangular, sigmoideal, entre otras.

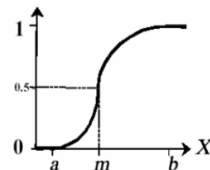
Forma triangular

$$A(x) = \begin{cases} 0 & \text{si } x < a \\ (x-a)/(m-a) & \text{si } a \leq x \leq m \\ (b-x)/(b-m) & \text{si } m \leq x \leq b \\ 0 & \text{si } x > b \end{cases}$$



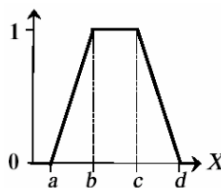
Forma Sigmoide

$$A(x) = \begin{cases} 0 & \text{si } x < a \\ 2\{(x-a)/(b-a)\}^2 & \text{si } a < x < m \\ 1 - 2\{(x-b)/(b-a)\}^2 & \text{si } m < x < b \\ 1 & \text{si } x > b \end{cases}$$



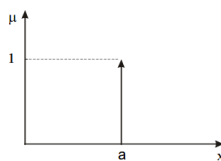
Forma Trapezoidal

$$A(x) = \begin{cases} 0 & \text{si } (x < a) \text{ ó } (x > d) \\ (x-a)/(b-a) & \text{si } a \leq x \leq b \\ 1 & \text{si } b \leq x \leq c \\ (d-x)/(d-c) & \text{si } c \leq x \leq d \end{cases}$$



Forma Singleton

$$A(x) = \begin{cases} 1 & \text{si } x = a \\ 0 & \text{si } x \neq a \end{cases}$$



3.1.1.3. Controlador difuso tipo I

El controlador difuso básicamente consta de tres operaciones: *fusificación*, *máquina de inferencia*, *defusificación*. La Figura 3.4 muestra la estructura general de un controlador difuso.

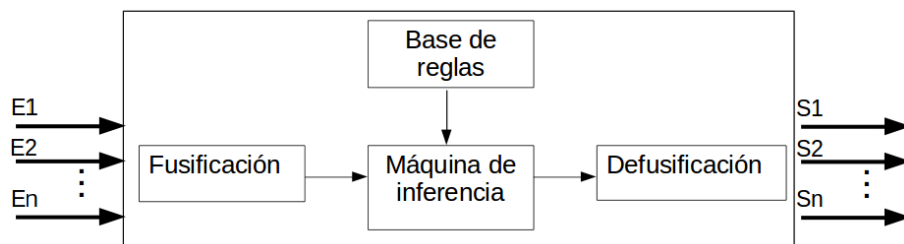


FIGURA 3.4: Esquema de un controlador basado en lógica difusa tipo-I.

3.1.1.4. Fusificación

La fusificación es la etapa que consiste en la cuantificación difusa de los valores reales de una variable.

Es el proceso de asignar valores de membresía o pertenencia a un valor numérico de entrada para cada una de las etiquetas difusas que forman la variable lingüística; por ejemplo, la variable lingüística “Temperatura de la sala” puede tomar los valores “baja”, “semi-baja”, “media” y “alta”; para este caso la entrada al fusificador es un valor de temperatura preciso (crisp), y la salida estará formada por los valores de verdad de cada una de las etiquetas “baja”, “semi-baja”, “media” y “alta”. A este bloque pueden llegar múltiples entradas, y producirse un conjunto difuso para cada una de ellas, esto se puede observar en la Fig. 3.5. El conjunto que se produce en este bloque se encuentra definido sobre el universo de discurso de la variable lingüística respectiva. Puede tener una función de membresía cuya forma puede ser distinta para cada variable de entrada.

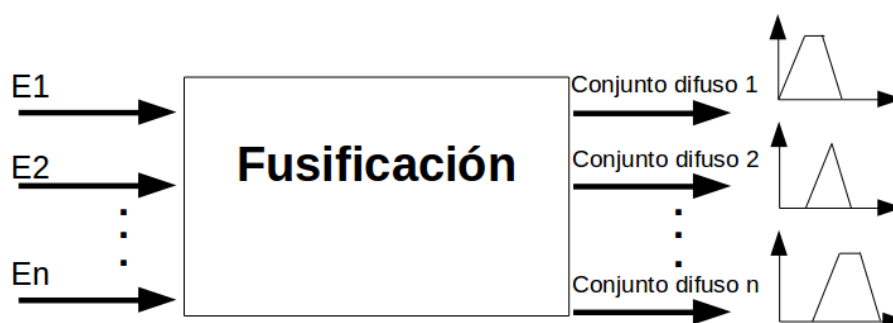


FIGURA 3.5: Bloque de entrada de fusificación.

3.1.1.5. Base de conocimiento

Se obtiene de la experiencia desarrollada por un operador, o algún experto en el área de control, y depende del proceso a controlar y los requerimientos de diseño. Está compuesta de dos partes, una base de datos y una base de reglas de control difuso (utiliza variables lingüística). Las bases de reglas también se puede obtener utilizando métodos de optimización.

3.1.1.6. Lógica de decisiones (máquina de inferencia)

Se encarga de aplicar el mecanismo de inferencia seleccionado a las reglas, almacenadas en la base de reglas. Realiza la tarea de calcular las variables de salida a partir de las variables de entrada, mediante las reglas del controlador y la inferencia difusa, entregando conjuntos difusos de salida. Es decir, la máquina de inferencia recibe los p

conjuntos difusos producidos por la interfaz de fuzificación, y los aplica a cada una de las m reglas de la Base de reglas, para producir $m * q$ conjuntos difusos (un conjunto difuso por cada variable de salida en cada una de las reglas) definidos sobre los universos.

3.1.1.7. Defusificación tipo I

Este tercer paso consiste en convertir un conjunto difuso en un número real, es decir se realiza un mapeo de un espacio de acciones de control definido sobre un universo de discurso de salida, a un espacio de acciones de control no-difuso (valores precisos). A continuación se mencionan algunas técnicas empleadas para la defusificación.

Método del centro de áreas.

Es una técnica muy utilizada en lógica difusa, en la cual se calcula el centro de gravedad de la distribución de las salidas de control difuso, y está se aplica al sistema que se controla. Para un Universo de discurso discreto, la salida defusificada se puede obtener con la siguiente expresión:

$$Salida = \frac{\sum_{j=1}^n P_j * W_j}{\sum_{j=1}^n W_j} \quad (3.7)$$

donde P_j es el valor pico de la i^{th} función de membresía y W_j es el peso asociado con las i^{th} regla.

Criterio de máximo (MC) . La salida es aquella para la cuál la función de membresía alcanza su máximo valor.

El método de la media de máximo (MOM, middle of maximum). La salida es el valor medio de los valores cuyas funciones de membresía alcanzan el valor máximo. La salida puede definirse por la siguiente ecuación:

$$\sum_{j=1}^I \frac{w_j}{I} \quad (3.8)$$

Donde w_j es el valor central de la etiqueta lingüística "j" para la cual la función de membresía alcanza el máximo valor $\mu(w_j)$ y "I" es el número de etiquetas que alcanzan el valor máximo.

3.1.2. Lógica difusa tipo II

La lógica difusa tipo II es caracterizada por una función de membresía alta (UMF) y una función de membresía baja (LMF) (Fig.3.6). La unión de todas esas funciones de

membresía es llamada *huella de incertidumbre*, (*footprint of uncertainty*, (*FOU*)) de \tilde{A} , donde

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (3.9)$$

para $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. J_x representa la función de membresía alta x y $\mu_{\tilde{A}}(x, u)$ es conocido como conjunto secundario, que es un conjunto de tipo difuso I. La Fig. 3.6 muestra un ejemplo de un conjunto difuso tipo II de \tilde{A} , que tiene un *limite superior* llamado A_{sup} y un *limite inferior* de $FOU\tilde{A}$ llamado, A_{inf} . \tilde{A} es representado por los siguientes parametros en el eje x . A_{sup} y A_{inf} son definidos en la ecuaciones siguientes respectivamente.

$$A_{sup}(x) = \begin{cases} 0, x < izq_{sup} \\ (x - izq_{sup}) / (m_{izq_{sup}} - izq_{sup}), izq_{sup} \leq x \leq m_{izq_{sup}} \\ 1, izq_{sup} \leq x < m_{der_{sup}} \\ (der_{sup} - x) / (der_{sup} - m_{der_{sup}}), m_{der_{sup}} \leq x \leq m_{der_{sup}} \\ 0, x > der_{sup} \end{cases} \quad (3.10)$$

$$A_{inf}(x) = \begin{cases} 0, x < izq_{inf} \\ \beta(x - izq_{inf}) / (m_{izq_{inf}} - izq_{inf}), izq_{inf} \leq x \leq m_{izq_{inf}} \\ \beta, izq_{inf} \leq x < m_{der_{inf}} \\ \beta(der_{inf} - x) / (der_{inf} - m_{der_{inf}}), m_{der_{inf}} \leq x \leq m_{der_{inf}} \\ 0, x > der_{inf} \end{cases} \quad (3.11)$$

donde izq_{sup} , $m_{izq_{sup}}$, $m_{der_{sup}}$, der_{sup} , $m_{der_{sup}}$, izq_{inf} , $m_{izq_{inf}}$, $m_{der_{inf}}$, der_{inf} , se muestran en la Fig. 3.6.

3.1.3. Controlador difuso tipo II

Para el difuso tipo II, el proceso de inferencia es muy similar, ya que el motor de inferencia combina reglas y proporciona un mapeo de los conjuntos difusos de entrada tipo-II a los conjuntos difusos de salida tipo-II. Este tipo de sistema es llamado sistema generalizado difuso tipo-II [39] el cual es complicado de implementar. Por lo tanto en [44] proponen un caso especial de un sistema generalizado tipo-II en el cual solamente intervalos difusos tipo-II son usados. La arquitectura de un sistema de intervalo difuso tipo-II es mostrado en la Fig. 3.7.

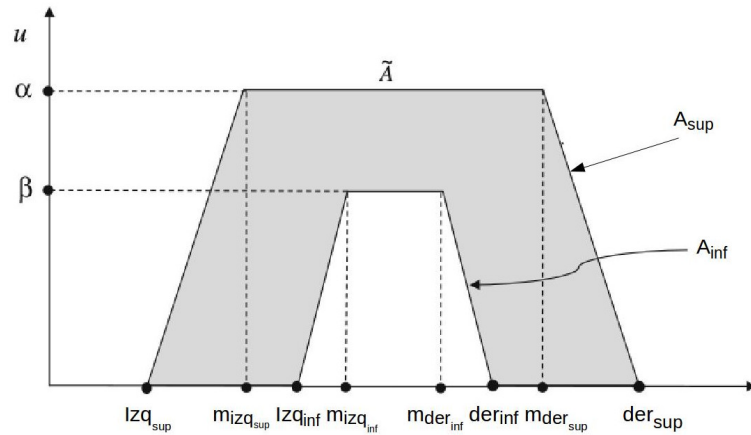


FIGURA 3.6: Función de membresía alta (UMF) y función de membresía baja (LMF) de \tilde{A} .

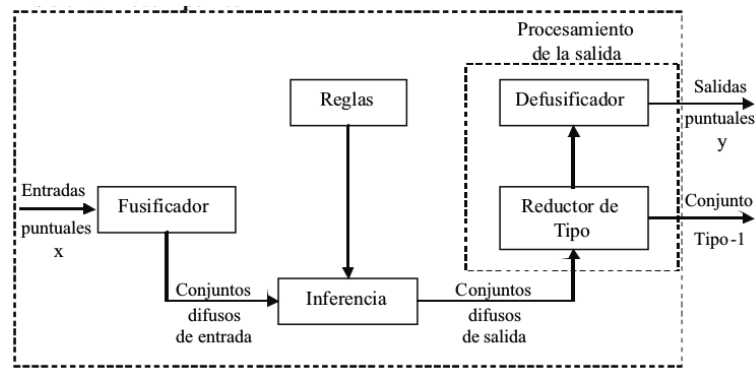


FIGURA 3.7: Controlador difuso tipo-II[56].

Las salidas del motor de inferencia de éste, son conjuntos difusos de tipo II y se necesita un reductor de tipo para convertirlos en un conjunto difuso tipo I antes de que se pueda llevar a cabo la defusificación. La estructura de reglas de un sistema fuzzy tipo I y el intervalo del sistema fuzzy tipo II es dado por:

Si x_1 es F_1^i Y x_n es F_n^i , entonces y es Y^i donde $i = 1, \dots, M$

3.1.3.1. Reductor de tipo

En un controlador difuso tipo-II la salida Y^i es representada por dos intervalos $[y^i, \bar{y}^i]$ y para calcularlo, se debe aplicar primero un procesamiento llamado reductor de tipo. El reductor de tipo es un proceso que representa un mapeo de un conjunto difuso tipo-II a un conjunto difuso tipo-I. Ya que el procedimiento para obtener el centroide de un conjunto difuso tipo-II es una extensión del cálculo para un difuso tipo-I, existen también varios métodos para obtenerlo. Un algoritmo de reducción de tipo se puede realizar mediante un algoritmo iterativo llamado Karnik-Mendel [39]. En la siguiente sección se explica este algoritmo el cual es implementado en este trabajo.

3.1.3.2. Algoritmo de Karnik-Mendel.

Karnik-Mandel diseñaron un algoritmo iterativo para el cálculo de centroides de los conjuntos difusos tipo-II . Este algoritmo permite hallar y_r y y_l . El algoritmo para determinar y_r es el siguiente:

Algoritmo 1 Algoritmo KM para y_r

- 1: Ordenar y_r^i de manera ascendente.
- 2: Calcular y_r como

$$y_r = \frac{\sum_{i=1}^M f_r^i y_r^i}{\sum_{i=1}^M f_r^i} \quad (3.12)$$

donde $f_r^i = \frac{\bar{f}^i + f^i}{2}$ y \bar{f}^i, f^i son los intervalos de disparo. Sea $y'_r = y_r$.

- 3: Encontrar R , tal que $y_r^R \leq y'_r \leq y_r^{R+1}$.
 - 4: Encontrar $y'_r = \frac{\sum_{i=1}^M f_r^i y_r^i}{\sum_{i=1}^M f_r^i}$ con $f_r^i = \underline{f}^i$ para $i \leq R$ y $f_r^i = \bar{f}^i$ para $i > R$. Sea $y''_r = y'_r$.
 - 5: Si $y''_r \neq y'_r$, ir al paso 6. Si $y''_r = y'_r$, asignamos $y_r = y''_r$, y paramos.
 - 6: Sea $y'_r = y''_r$, ir al paso 3.
-

El algoritmo para determinar y_l , se muestra a continuación

Algoritmo 2 Algoritmo KM para y_l

- 1: Ordenar y_l^i de manera ascendente.
- 2: Calcular y_l como

$$y_l = \frac{\sum_{i=1}^M f_l^i y_l^i}{\sum_{i=1}^M f_l^i} \quad (3.13)$$

donde $f_l^i = \frac{\bar{f}^i + f^i}{2}$ y \bar{f}^i, f^i son los intervalos de disparo. Sea $y'_l = y_l$.

- 3: Encontrar L , tal que $y_l^L \leq y'_l \leq y_l^{L+1}$.
 - 4: Encontrar $y'_l = \frac{\sum_{i=1}^M f_l^i y_l^i}{\sum_{i=1}^M f_l^i}$ con $f_l^i = \underline{f}^i$ para $i \leq L$ y $f_l^i = \bar{f}^i$ para $i > L$. Sea $y''_l = y'_l$.
 - 5: Si $y''_l \neq y'_l$, ir al paso 6. Si $y''_l = y'_l$, asignamos $y_l = y''_l$, y paramos.
 - 6: Sea $y'_l = y''_l$, ir al paso 3.
-

La idea principal del algoritmo KM es encontrar los puntos de conmutación para y_r y y_l . Es decir para el caso de encontrar el punto de conmutación L , para $n \leq L$ los grados de membresía de la parte alta son usados para calcular y_l , y para $n > L$ los grados de membresía de la parte baja son usados. Este procesamiento asegura que y_l sea el mínimo. Como ejemplo se muestra este proceso en la Fig. 3.8.

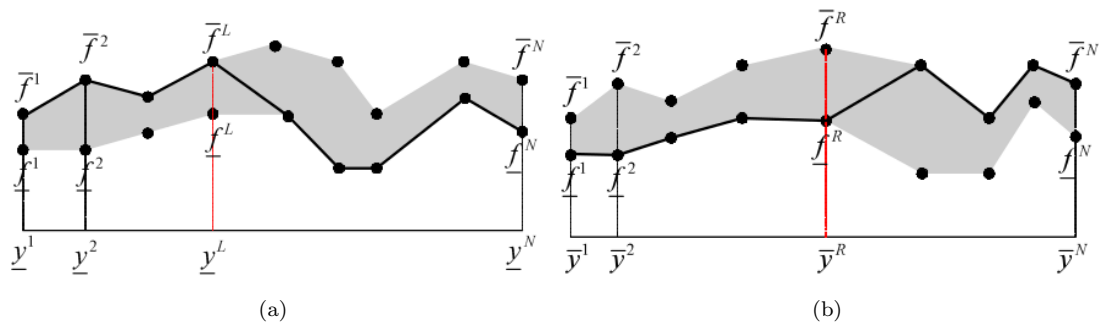


FIGURA 3.8: a) Compuo de y_l , cambia el nivel de disparo superior al nivel de disparo inferior. b) Compuo de y_r , cambia el nivel de disparo inferior al nivel de disparo superior.

3.1.3.3. Defusificación tipo II

Finalmente la salida defusificada es el promedio de y_r y y_l mediante la siguiente ecuación:

$$Y_{final} = \left(\frac{y_l + y_r}{2} \right). \quad (3.14)$$

En este capítulo se presento la teoría basica relacionada con control clásico, PID, y control inteligente, difuso tipo-I y difuso tipo-II. En el siguiente capítulo se explicará a detalle los componentes electrónicos y mecánicos que integran la pinza robótica. Además también se menciona el software para comunicarse con los motores de la pinza, es decir ROS.

Capítulo 4

Descripción de los componentes de la Pinza Robótica

En este capítulo se abordan los componentes mecánicos, electrónicos y de software de la pinza robótica. La pinza utilizada en este trabajo, es fabricada por CrustCrawler Robotics ¹, las partes de la pinza son hechas en aluminio. En la punta de cada dedo se colocó un sensor de fuerza piezoeléctrico la descripción de este sensor está en la sección 4.2.1.

A continuación se explican a detalle cada uno de los componentes principales.



FIGURA 4.1: Pinza angular

¹<http://www.crustcrawler.com/products/dualgripper/>

4.1. Componentes Mecánicos

La pinza angular, Fig.4.1, en su parte mecánica está compuesta por dos motores Dynamixel AX-12², Fig.4.2. Los motores Dynamixel de la empresa Robotics³ son dispositivos que integran motor, un engranaje de reducción, un microcontrolador y red de interconexión. Existen diversos modelos de motores Dynamixel, los cuales pueden ser consultados en el Apéndice A. Todos los motores Dynamixel, incluido por supuesto el AX-12, tienen un microcontrolador que puede ejecutar funciones específicas al recibir diferentes comandos, la mayoría de los cuales establecen o leen parámetros internos del Dynamixel. Estos motores ofrecen dos modos interesantes de operación en los que puede trabajar como rotación continua ó como un servo que se mueve de 0° a 320°. Por otra parte, cada motor tiene la capacidad de controlar su velocidad, y posición. Además se puede obtener las variables internas del motor las cuales pueden ser temperatura, posición, tensión y carga soportada. Es por esto que se seleccionó este tipo de pinza que incluye estos motores ya que pueden reaccionar al entorno utilizando la información leída de los sensores, para nuestro caso el sensor de posición es el que más nos interesa leer.

A continuación se presentan las principales características del motor Dynamixel AX-12:

- Dispone de sensor de posición con una resolución de $(300^\circ/1024 = 0.3^\circ)$.
- Peso: 54.6gr
- Dimensión: 32mmx50mmx40mm
- Ratio de reducción: 254 : 1
- Par motor: 1.52N.m (a 12.0V, 1.5A)
- Velocidad sin carga: 59rpm (a 12V)
- Grados de giro: 0° 300°
- Rotación continua
- Tensión de operación: 9 12V (tensión de operación recomendada 11.1V)
- Señal de comandos: paquete digital

²Ver Apéndice A

³<http://support.robotis.com/en/>

- Tipo de protocolo: Se controlan mediante un bus TTL (TTL level multidrop) en el que cada actuador tienen un identificador único. La interconexión se realiza en configuración *daisy chain*, lo que facilita el cableado.
- ID: 254 ID (0 253)
- Material: plástico

Mediante un controlador USB, se permite leer cada uno de los sensores de los motores, este controlador es detallado en la siguiente sección.



FIGURA 4.2: Motor Dynamixel AX-12.

4.1.1. USB2 Dynamixel

Para poder acceder a las lecturas de los sensores así como enviar los comandos a estos se necesita un conector usb especial. El USB2 Dynamixel, desarrollado también por Robotics, es una interfaz de comunicación computadora-Motor que permite comunicarse con los motores Dynamixel. Este permite tres tipos de conexión para los diversos motores fabricados por Dynamixel:

- TTL: permite la conexión de dispositivos usando cables con 3 pines como en los motores AX-12/12+, AX-S1, y AX-20.
- RS485: permite la comunicación con dispositivos usando cables con cuatro pines como DX-Series, RX-Series, etc.
- RS232: permite la comunicación con dispositivos mediante el cable serial.

Por lo tanto este controlador permite conectarse con todos tipos de motores de Dynamixel, lo cual es muy importante para este trabajo de investigación. Para el caso de la pinza los motores Dynamixel AX-12 se comunican únicamente con el protocolo TTL y también se desea controlar los motores del brazo descritos en el Apéndice B,



FIGURA 4.3: Conector USB2 Dynamixel.

los cuales algunos de estos motores soportan el protocolo RS485 . Además el USB2 dynamixel permite de manera fácil y rápida conectarse con ROS, mediante un nodo, el cual se describe en la sección 4.3.

4.2. Componentes Electrónicos

La pinza además de contar con partes mecánicas tiene componentes electrónicos externos que le permiten detectar la fuerza aplicada por cada dedo sobre el objeto y enviar esta señal a la computadora. Esta señal del sensor es muy importante procesarla correctamente pues servirá de entrada a los controladores los cuales se describen en la sección 5.

4.2.1. Sensor de fuerza



FIGURA 4.4: sensor fuerza

El sensor de fuerza piezoresistivo FSS-SMT (4.4) de la empresa Honeywell⁴, con tecnología de montaje superficial (SMT) permite el ensamblaje rápido en un circuito impreso, lo cual ayuda a reducir costos de montaje. Su elemento piezoresistivo es un silicio micromecanizado, este sensor es de bajo consumo, se alimenta solo con un voltaje de *5volts*. Su diseño de circuito de puente Wheatstone sin compensar, le provee una estable salida diferencial en *mV* sobre el rango de fuerza aplicado. Por otra parte este voltaje es radiométrico , con lo cual la salida es proporcional a la tensión con que se alimenta el sensor.

⁴<https://sensing.honeywell.com>

4.2.1.1. Caracterización del sensor de fuerza

La caracterización de diversos tipos de sensores de fuerza es presentado en varios trabajos de robótica, en especial en el área de manipulación [2], como en el trabajo de [67] donde muestran la caracterización de un arreglo de sensores piezoresistivos para ser usados en el interior de un dedo robótico. En el trabajo de [16] también se muestra el proceso de caracterización de un sensor capacitivo para una pinza paralela de uso industrial.

La caracterización de un sensor consiste en calcular la ecuación característica de su comportamiento. Esta ecuación ayuda a determinar la razón de cambio en la variable de salida, es decir, una variable eléctrica con respecto a una variable de entrada. Es muy importante realizar la caracterización para la interpolación de valores así como obtener mediciones con buena exactitud.

Para la caracterización del sensor de fuerza se hicieron los siguientes pasos:

1. El sensor de fuerza piezoresistivo se sujeto a la mesa de tal forma que no se moviera, utilizando foamy alrededor de el (Fig. 4.5.a). El sensor es conectado a un ADC (Convertidor analógico digital) con entradas diferenciales para enviar el valor del sensor a la computadora, el uso de este ADC se explica en la sección 4.2.2.
2. Sobre el sensor se colocaron masas de pesos constantes, desde $2gr$ con incrementos de $2gr$ y hasta $500gr$, en la Fig. 4.5.b,c se muestran algunos de ellos. Para cada peso se obtiene la lectura del sensor de fuerza el cual corresponde a una salida en milivoltios (Fig. 4.5.c). En la tabla 4.1, solo se muestran algunos pesos con sus respectivas lecturas del la salida del sensor de fuerza
3. Para cada peso se tomaron 60 muestras, siendo un total de 10312 muestras.
4. Posteriormente con todas las muestras de los valores que se obtuvieron de voltaje para cada masa, se obtiene la ecuación de regresión, la cual como se dijo anteriormente nos ayuda a determinar la razón de cambio en la variable de salida.

Como se observa en la tabla 6.7 en casi todas las lecturas la desviación estándar es cercana a cero lo cual nos indica que el sensor es poco ruidoso, por lo tanto no se requiere ni una etapa de filtrado y se puede conectar directo a la tarjeta de adquisición de datos para ser procesado por ROS.

Una vez que se realizaron los pasos 1. 2. y 3., se obtiene la gráfica que se muestra en la Fig. 4.6, colocando en el eje de la variable independiente (eje x) el rango de mediciones de los pesos y en el eje de la variable dependiente (eje y) el valor correspondiente al

Peso(g)	Voltaje (mV)			Peso (g)	Voltaje (mV)		
	Media	Desviación	Varianza		Media	Desviación	Varianza
0	-2.184	0.110	0.012	260	18.18846	0.12524	0.01568
10	-1.292	0.094	0.008	270	18.98846	0.05286	0.00279
20	-0.5	0	0	280	19.73076	0.06713	0.00450
30	0.25	0	0	290	20.5	0	0
40	0.96538	0.087017	0.00757	300	21.5	0	0
50	1.75	0	0	310	22.49609	0.07031	0.00494
60	2.50373	0.03054	0.00093	320	23.21538	0.08701	0.00757
70	3.25	0	0	330	24.00781	0.04384	0.00192
80	4	0	0	340	24.99632	0.03031	0.00091
90	4.98076	0.06713	0.00450	350	25.75	0	0
100	5.99615	0.03100	0.00096	360	26.74230	0.04350	0.00189
110	6.74615	0.03100	0.00096	370	27.77459	0.07506	0.00563
120	7.5	0	0	380	28.75384	0.03100	0.00096
130	8.25	0	0	390	29.75	0	0
140	9	0	0	400	30.51153	0.06890	0.00474
150	9.5	0	0	410	31.45454	0.15133	0.02290
160	10.5	0	0	420	32.22692	0.07292	0.00531
170	11.25	0	0	430	32.99615	0.03100	0.00096
180	12	0	0	440	33.75	0	0
190	12.75	0	0	450	34.46153	0.14810	0.02193
200	13.5	0	0	460	35.25	0.0625	0.00390
210	14.24615	0.03100	0.00096	470	36.01153	0.05286	0.00279
220	15	0	0	480	36.75384	0.05398	0.00291
230	15.75	0	0	490	37.74615	0.03100	0.00096
240	16.5	0	0	500	38.81923	0.12109	0.014663
250	17.25	0	0				

CUADRO 4.1: Mediciones de peso y voltaje para el sensor de fuerza.

voltaje. Como se puede observar el comportamiento es lineal, por lo tanto se utilizó una regresión polinomial, en este caso de primer orden, la cual nos proporciona la relación de voltaje a gramos fuerza. El polinomio resultante del ajuste de los datos es:

$$y = 0.0814x - 2.4907 \quad (4.1)$$

donde y es el voltaje y x es el peso de los objetos en gramos.

4.2.2. Tarjeta de adquisición de datos

Para enviar la señal de los sensores a la computadora se necesita de una tarjeta de adquisición de datos. Se optó por una tarjeta Arduino⁵, que es una variedad de placas con microcontroladores de bajo costo. Arduino es en esencia una plataforma electrónica

⁵ <http://www.arduino.cc>

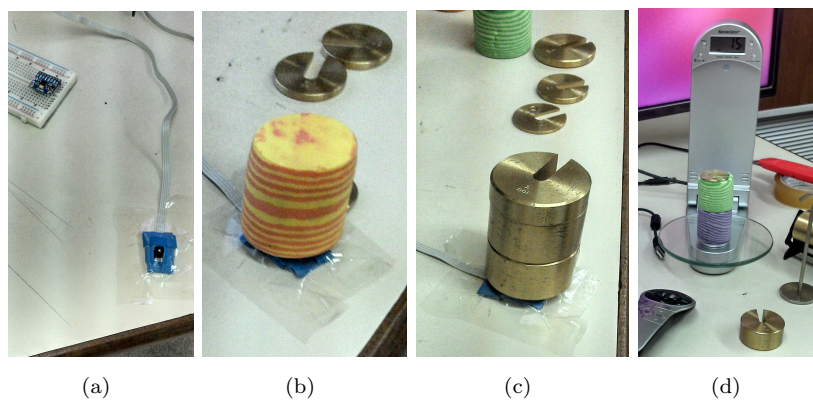


FIGURA 4.5: Cilindro de foamy sobre el sensor (a), pesas para la caracterización del sensor (b), báscula digital (c).

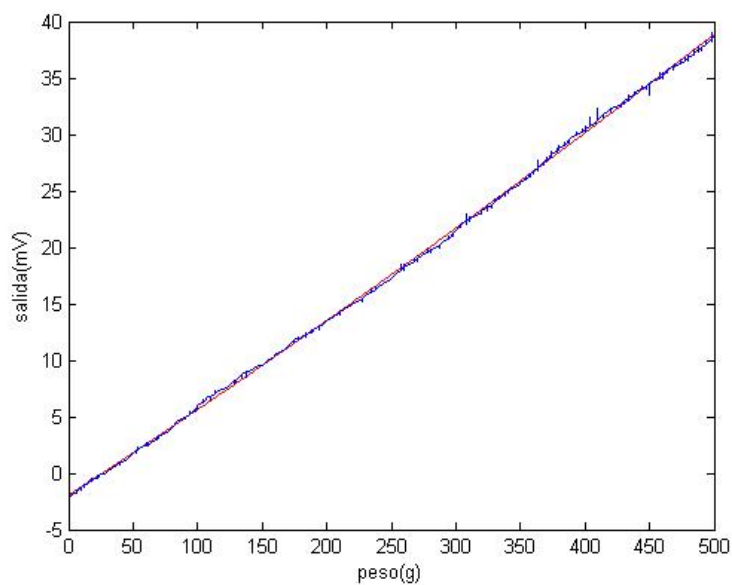


FIGURA 4.6: Gráfica de caracterización de la señal de salida del sensor de fuerza FSS-SMT

abierta para la creación prototipos basada en software y hardware flexibles y fáciles de usar. La tarjeta que se empleó en este trabajo es la Arduino Mega (Fig. 4.7).

Algunas de sus principales características son :

- Microcontrolador: ATmega2560 de 8 bits.
- Alimentación: 5V
- Entrada: 7 – 12V
- Pines digitales: 54 (14 con PWM)
- Pines analógicos: 16

- Protocolos: SPI, I2C.
- Corriente por pin: $40mA$
- Corriente sobre pin $3.3V$: $50mA$
- Memoria Flash (programa): $256KB$ (8 KB usados para el bootloader)
- SRAM: $8KB$
- EEPROM: $4KB$
- Reloj: $16MHz$



FIGURA 4.7: Tarjeta Arduino Mega.

Durante la caracterización del sensor de fuerza mostrado en la sección 4.2.1.1 se determinó que el Convertidor Analógico Digital (ADC) de la tarjeta Arduino no alcanzaba a detectar las señales de los sensores de fuerza, ya que son del orden de $0.1mV$ por cada dos gramos y la tarjeta Arduino Mega solo detecta un cambio de voltaje mayor a $4.9mV$. Por esta razón se optó por usar en ADC externo el cual se explica en la siguiente sección.

4.2.2.1. Convertidor Analógico Digital externo

EL ADC empleado fue el ADS1015⁶(Fig. 4.8) que es un circuito integrado de 12 bits. Las principales características de este ADC son las siguientes:

- Rango de voltaje 2.0 a 5.0 *volts*.
- Velocidad de muestreo de 128 a 3300 SPS (Samples Per Second).
- Ganancia programable (PGA). Gracias a que la ganancia se puede programar por software no es necesario agregar algún circuito operacional que amplifique la señal del sensor.

⁶<http://www.ti.com/product/ads1015>

- Protocolo de comunicación I2C. El cual permite ser conectado directamente a la tarjeta Arduino mega.
- Bajo consumo de corriente.
- Posee un multiplexor (MUX) de 4 entradas analógicas, las cuales pueden ser usadas también para voltaje diferencial que son muy importantes estas entradas para poder conectar el sensor de fuerza descrito en la sección anterior.

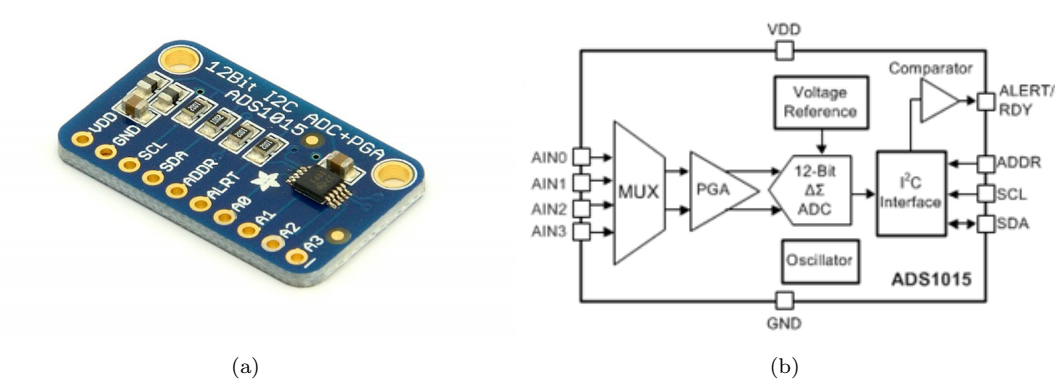


FIGURA 4.8: a) Tarjeta del convertidor analógico digital ADS1015. b) diagrama a bloques de la estructura interna del ADC. Las entradas AIN0-AIN-3, corresponden a las entradas Analógicas 0-3 respectivamente.

Para utilizar el ADS1015, se programó con una tasa de muestreo de 1600 SPS, en modo de entrada diferencial y una ganancia de 8, es decir por cada cambio de $0.15mV$ se incrementa en un bit el ADC.

Finalmente los motores Dynamixel AX-12 son conectados en serie y son controlados cada uno mediante el USB2 Dynamixel, el cual recibe los comandos desde la computadora. Los dos sensores de fuerza son conectados a las entradas diferenciales del ADS1015. El ADC externo es conectado a la tarjeta Arduino Mega mediante I2C. Finalmente Arduino envía a la computadora las señales de los sensores de fuerza. Esto se puede apreciar en la Fig. 4.9.

4.3. ROS

El software ocupado para controlar la pinza robótica es Robot Operating System (ROS) [66]. ROS está basado en *nodos*, *mensajes*, *topics* y *servicios*. Cada fragmento se denomina nodo y, por lo general, se ejecuta como un proceso separado. Los nodos se comunican entre sí al pasar mensajes (*messages*). Un mensaje es simplemente una estructura de datos, que contiene un tipo de datos, como por ejemplo: entero, flotante,

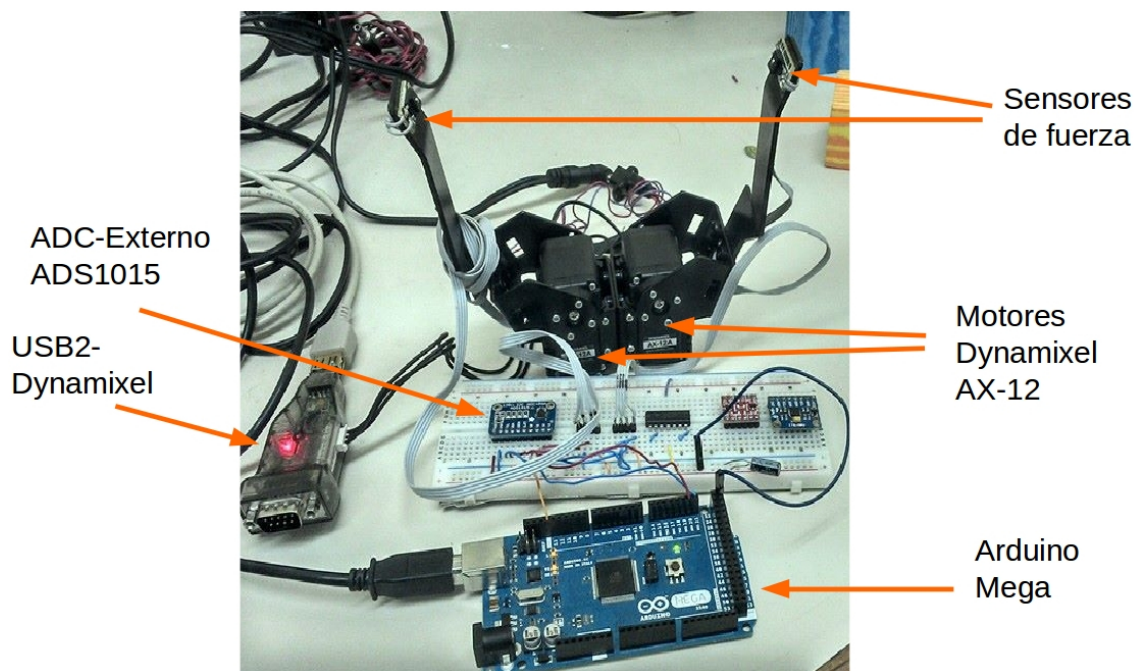


FIGURA 4.9: Componentes de electrónicos, mecánicos y de software empleados en la pinza robótica.

booleano, etc. Un nodo envía un mensaje para ser publicado en un determinado *Topic*. El *Topic* es un nombre que se usa para identificar el contenido del mensaje. Un nodo que está interesado en un cierto tipo de datos se suscribirá al *Topic* (tema) apropiado. Puede haber múltiples suscriptores y publicadores para un solo *Topic*, y un solo *nodo* puede publicar o suscribirse a múltiples *topics*. Para controlar la pinza se necesitan tres nodos llamados: *Serial_node*, *dynamixel_manager* y *Force_positon_controller*. Los nodos *Serial_node* y *dynamixel_manager* se encuentran en los paquetes que se explican en las secciones siguientes.

4.3.1. Paquete *Rosserial*

El paquete *rosserial* es un conjunto de protocolos de comunicación estandarizados implementados para comunicarse desde ROS con otros dispositivos como por ejemplo puertos seriales, sockets y viceversa. El protocolo *rosserial* puede convertir los mensajes estándar y servicios de ROS a datos equivalentes para dispositivos embebidos. Los datos seriales son enviados como un paquete de datos que incluye un *header* y una cola de bytes.

El paquete *rosserial* es muy útil para este trabajo de investigación, ya que como se mencionó en la sección 4.2.2 se emplea una tarjeta de adquisición de datos y este paquete incluye una librería para esta plataforma llamada:

- *rosserial_arduino*: funciona en plataformas Arduino como Arduino UNO y Leonardo, y también trabaja con la Arduino Mega y Arduino Due.

Por otra parte del lado del equipo de cómputo, también se necesita una librería que lea los datos enviados de la tarjeta Arduino a través del puerto USB:

- *rosserial_python*: Esta librería es la encargada de manejar los datos enviados desde un dispositivo embebido. Esta librería está completamente escrita en Python.

4.3.2. Paquete *Dynamixel_motor*

En ROS está disponible un paquete que permite comunicarse con los motores Dynamixel y el USB2 Dynamixel, este paquete se llama *dynamixel_motor* el cual consta de las siguientes sub paquetes:

- *dynamixel_driver*: Este paquete es el manejador (driver) de Dynamixel, que puede hacer una comunicación de entrada/salida de bajo nivel con cada motor Dynamixel desde la computadora. Como se mencionó en la sección 4.1 cuando se desea leer/escribir algún comando al motor, este paquete es el encargado de hacerlo. También es usado por paquetes de más alto nivel como es el caso del paquete *dynamixel_controllers*
- *dynamixel_controllers*: Este paquete es el de alto nivel usado por *dynamixel_motor*. Con este paquete podemos crear una articulación (Joint) en un robot. El paquete contiene un scrip configurable de nodos y servicios para iniciar, detener y reiniciar plugins de una articulación. Cada controlador Dynamixel se puede configurar utilizando parámetros de ROS o pueden ser configurados mediante un archivo YAML. El *dynamixel_controllers* los paquetes son compatibles con los siguientes tipos de controladores:
 - Joint Position controllers
 - Joint Torque controllers
 - Joint Trajectory Action controller
- *dynamixel_msgs*: Estas son las definiciones de mensaje que se usan dentro del paquete *dynamixel_motor*.

4.3.3. Estructura de comunicación de la pinza con ROS

Con los paquetes que se mencionaron anteriormente se debe instanciar los Topics y Nodos a nuestro caso, que es la pinza, el cual se puede observar en la Fig. 4.10 cómo está implementado. Para la pinza se necesitan diversos Topics en los cuales se publicarán los valores:

- Topic Gripper_Force (Publica los valores que envía la tarjeta Arduino a través del puerto USB, estos valores corresponden a las lecturas de los sensores izquierdo y derecho.)
- Topic gripper_right_controller state publica los valores del estado interno del motor derecho (posición,temperatura,voltaje)
- Topic gripper_left_controller state publica los valores del estado interno del motor derecho (posición,temperatura,voltaje)
- Topic gripper_right_controller command publica los valores de velocidad para que el motor los ejecute
- Topic gripper_left_controller command publica los valores de velocidad para que el motor los ejecute

El nodo Force_position_controller se suscribe el topic Gripper Force, para obtener los valores que servirán de entrada a los controladores que se explican en el capítulo 5 y posteriormente la salida del controlador será un comando de velocidad que se publicará en el topic gripper_right_controller command.

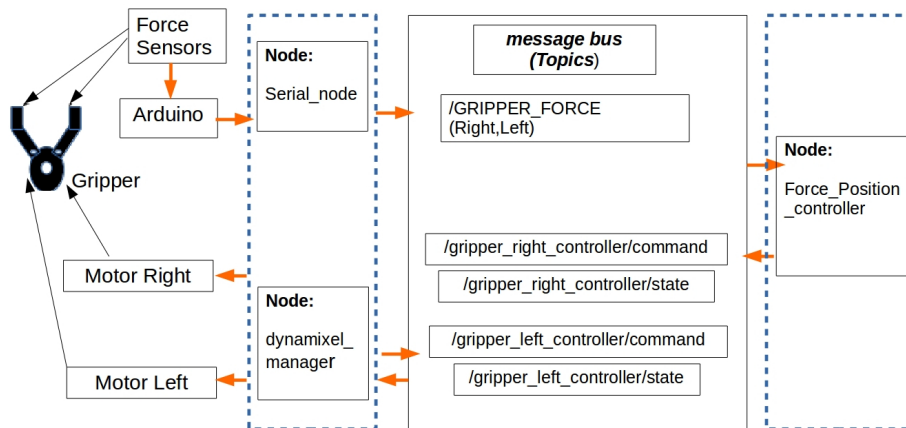


FIGURA 4.10: Estructura de comunicación entre los nodos de ROS.

4.4. Análisis Cinemático de la Pinza

En el sistema de nuestra pinza, se propone hallar una ecuación que permita obtener los valores x_d, y_d del dedo derecho de la pinza, para esto se hacen pruebas en las cuales se propone una distancia D , unos puntos x_i, y_i izquierdos, un radio r , y con eso hallar los valores x_d, y_d derechos. A continuación se muestran las ecuaciones las cuales al final solo se instancian con valores para obtener los resultados x_d, y_d derechos.

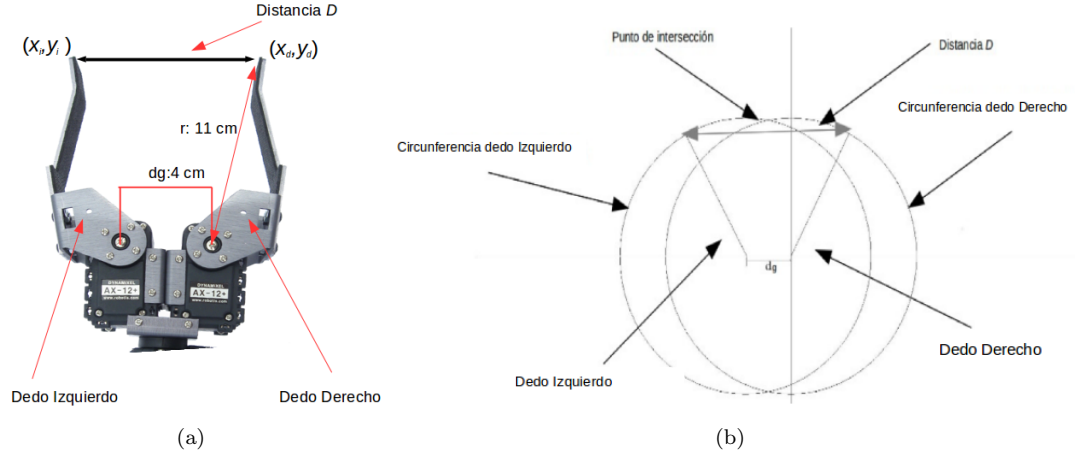


FIGURA 4.11: Pinza angular Dynamixel, a) medidas de los dedos de la pinza, b) trayectoria circular que describe cada dedo en su movimiento, se establece como origen del marco de referencia el motor derecho.

Partiendo de la Fig.4.11, donde se muestra la trayectoria que sigue cada dedo formando una circunferencia de radio r , establecemos como origen de nuestro sistema el motor del dedo derecho. La ecuación de la circunferencia para el dedo derecho de la pinza es:

$$r_d^2 = (x_d - a)^2 + (y_d - b)^2 \quad (4.2)$$

Donde a y b valen cero por estar en el origen, por lo tanto despejando la variable y_d de ecuación anterior tenemos:

$$y_d = \sqrt{r_d^2 - x_d^2} \quad (4.3)$$

Sea D la distancia entre dos puntos (el ancho del objeto que se encuentra entre los dos dedos), es decir la punta del dedo izquierdo (x_i, y_i) , y la punta del dedo derecho (x_d, y_d) :

$$D^2 = (x_i - x_d)^2 + (y_i - y_d)^2 \quad (4.4)$$

sustituyendo y_d de 4.3 en 4.4 y desarrollando tenemos:

$$D^2 = x_i^2 - 2x_dx_i + x_d^2 + y_i^2 - 2y_i\sqrt{r_d^2 - x_d^2} + r_d^2 - x_d^2 \quad (4.5)$$

elevando al cuadrado ambos términos para eliminar la raíz cuadrada:

$$2y_i(\sqrt{r_d^2 - x_d^2})^2 = (x_i^2 - 2x_dx_i + y_i^2 + r_d^2 - D^2)^2 \quad (4.6)$$

$$4y_i^2(r_d^2 - x_d^2) = (x_i^2 - 2x_dx_i + y_i^2 + r_d^2 - D^2)^2 \quad (4.7)$$

Desarrollando 4.7 agrupando y simplificando:

$$\begin{aligned} x_d^2(4x_i^2 + 4y_i^2) + x_d(4x_iD^2 - 4x_i^3 - 4x_iy_i^2 - 4x_ir_d^2) + x_i^4 + y_i^4 + r_i^4 + D^4 + 2x_i^2y_i^2 + 2x_i^2r_i^2 \\ - 2x_i^2D^2 + 2y_i^2r_i^2 - 2y_i^2D^2 - 2r_i^2D^2 - 4y_i^2r_i^2 = 0 \end{aligned} \quad (4.8)$$

Claramente se observa que nos queda una ecuación de segundo orden donde,

$$a = 4x_i^2 + 4y_i^2 \quad (4.9)$$

$$b = 4x_iD^2 - 4x_i^3 - 4x_iy_i^2 - 484x_i \quad (4.10)$$

$$\begin{aligned} c = x_i^4 + y_i^4 + r_i^4 + D^4 + 2x_i^2y_i^2 + 2x_i^2r_i^2 \\ - 2x_i^2D^2 + 2y_i^2r_i^2 - 2y_i^2D^2 - 2r_i^2D^2 - 4y_i^2r_i^2 \end{aligned} \quad (4.11)$$

Por lo tanto tendremos dos posibles soluciones:

$$x_{d1} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \quad (4.12)$$

$$x_{d2} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \quad (4.13)$$

Las cuales se tendrán en 4.3, por lo tanto tendremos cuatro posibles soluciones.

$$y_{d1} = \sqrt{r^2 - x_{d1}^2} \quad (4.14)$$

$$y_{d2} = \sqrt{r^2 - x_{d2}^2} \quad (4.15)$$

También se necesita saber el punto máximo que puede alcanzar un griper respecto al otro sin chocar contra éste. Es decir conocer la trayectoria que puede alcanzar cuando se mueve éste sin objeto de por medio. Por lo tanto se requiere el punto máximo de intersección entre una línea recta descrita, por ejemplo por el motor izquierdo, y la circunferencia del griper derecho. Esto es cuando el griper derecho se mueve en dirección hacia el griper izquierdo. Tenemos la ecuación de la línea recta que pasa por los puntos (a, b) y (x_i, y_i)

$$y_d - b = \frac{y_i - b}{x_i + a}(x_d + a) \quad (4.16)$$

Despejando la variable y_d de 4.16

$$y_d = \frac{y_i - b}{x_i + a}(x_d + a) + b \quad (4.17)$$

Por otra parte tenemos la ecuación de la circunferencia derecha, donde a y b valen cero por estar en el origen:

$$r_d^2 = x_d^2 + y_d^2 \quad (4.18)$$

Sustituyendo 4.17 en 4.18 tenemos

$$r^2 = x^2 + \left(\frac{y_i - b}{x_i + a}(x + a) + b\right)^2 \quad (4.19)$$

Desarrollando y agrupando 4.19 se obtiene

$$\begin{aligned} x_d^2 \left(\frac{y_i^2}{(x_i - a)^2} + \frac{b^2}{(x_i - a)^2} + 1 - \frac{2y_i b}{(x_i - a)^2} \right) + x \left(\frac{2y_i b}{x_i - a} + \frac{2b^2 a}{(x_i - a)^2} - \frac{2b^2}{x_i - a} - \right. \\ \left. \frac{4y_i b a}{(x_i - a)^2} + \frac{2y_i^2 a}{(x_i - a)^2} \right) - \frac{2y_i a^2 b}{(x_i - a)^2} + b^2 - r^2 - \frac{2b^2 a}{x_i - a} + \frac{2y_i a b}{x_i - a} + \frac{b^2 a^2}{(x_i - a)^2} + \frac{y_i^2 a^2}{(x_i - a)^2} = 0 \end{aligned} \quad (4.20)$$

donde:

$$a_coe = \frac{y_i^2}{(x_i - a)^2} + \frac{b^2}{(x_i - a)^2} + 1 - \frac{2y_i b}{(x_i - a)^2} \quad (4.21)$$

$$b_coe = \frac{2y_i b}{x_i - a} + \frac{2b^2 a}{(x_i - a)^2} - \frac{2b^2}{x_i - a} - \frac{4y_i b a}{(x_i - a)^2} + \frac{2y_i^2 a}{(x_i - a)^2} \quad (4.22)$$

$$c_{.coe} = \frac{2y_i a^2 b}{(x_i - a)^2} + b^2 - r^2 - \frac{2b^2 a}{x_i - a} + \frac{2y_i a b}{x_i - a} + \frac{b^2 a^2}{(x_i - a)^2} + \frac{y_i^2 a^2}{(x_i - a)^2} \quad (4.23)$$

Resolviendo la ecuación cuadrática se encuentra el punto x_d de intersección con dos posibles valores:

$$x_{d1} = \frac{-b_{.coe} + \sqrt{b_{.coe}^2 - 4a_{.coec_{.coe}}}}{2a_{.coe}}. \quad (4.24)$$

$$x_{d2} = \frac{-b_{.coe} - \sqrt{b_{.coe}^2 - 4a_{.coec_{.coe}}}}{2a_{.coe}}. \quad (4.25)$$

finalmente se sustituye 4.24 o 4.25 en 4.17 para hallar el punto y_d de intersección.

4.4.1. Espacio de Configuraciones

Las ecuaciones generales presentadas anteriormente permiten conocer las posiciones e intersecciones de un dedo respecto al otro. Para simular el análisis del griper respecto a las ecuaciones anteriores se hicieron pruebas en Matlab, de las ecuaciones 4.12,4.13,4.14,4.15 se programaron en Matlab para comprobar sus respectivas posiciones dados unos puntos (x_i, y_i) . Para el caso de:

$$x_i = 6.33$$

$$y_i = 3.76$$

$$r = 11cm$$

$$dg = 4cm$$

$$D = 0cm$$

tenemos los siguientes resultados en xy del dedo derecho:

$$x_d = 9.45,$$

$$y_d = 5.61$$

Y los ángulos para ambos motores:

$$motor_{izquierdo} = 20^\circ$$

$$motor_{derecho} = 30.69^\circ. \text{ Ver Fig.4.12}$$

El punto de intersección se muestra en la Fig. 4.13. En el cual tenemos que los puntos xy de las puntas de los dedos son:

$$x_i = -2$$

$$y_i = 10.79$$

$$x_d = -1.90$$

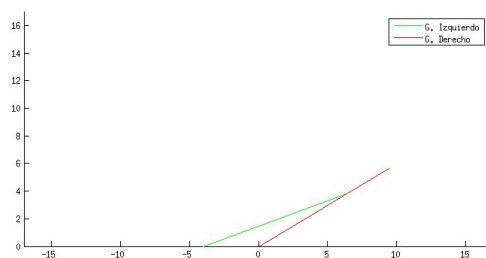


FIGURA 4.12: Propuestos los valores x_i y y_i , y una distancia de separación cero $D = 0$, se hallan los valores x_d y y_d . La línea verde representa el dedo izquierdo y la roja el dedo derecho. La posición de los dedos apunta al lado derecho.

$$y_d = 10.83$$

Y los ángulos para ambos motores:

$$motor_izquierdo = 79.89^\circ$$

$$motor_derecho = 99.98^\circ .$$

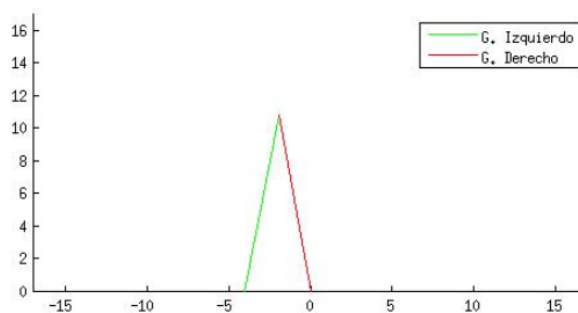


FIGURA 4.13: EL punto de intersección se presenta cuando los dos dedos se juntan en la punta, esto es cuando no hay objeto entre ellos.

En la Fig. 4.14 se muestra una posición desplazada hacia la izquierda en la cual tenemos los puntos xy de la punta del dedo izquierdo:

$$x_i = -9.50$$

$$y_i = 9.52$$

Y obtenemos los puntos xy de la punta del dedo derecho:

$$x_d = -8.22$$

$$y_d = 7.30$$

Y por último los ángulos para ambos motores:

$$motor_izquierdo = 120^\circ$$

$$motor_derecho = 138.35^\circ .$$

Con el fin de obtener todas las posiciones x_d, y_d del dedo derecho con respecto a los puntos x_i, y_i del dedo izquierdo en la Fig.4.15 se muestran las correspondientes posiciones.

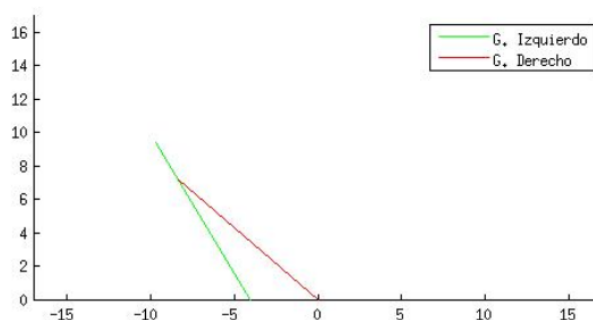


FIGURA 4.14: Dados unos valores x_i y y_i , y una distancia de separación cero $D = 0$, en este caso los dedos apuntan al lado izquierdo.

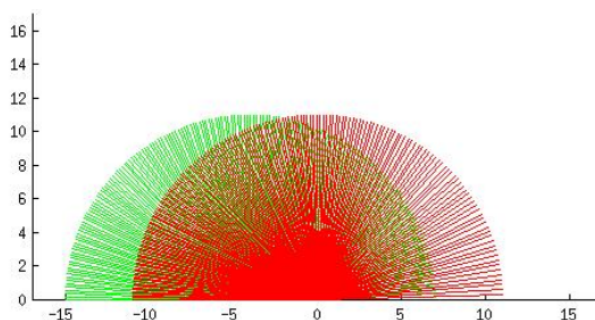


FIGURA 4.15: Todas las posibles posiciones que pueden tener los puntos x, y de cada uno de los dedos.

Por otro lado, se grafican los ángulos obtenidos del dedo derecho contra los ángulos del dedo izquierdo, esto para validar las posiciones posibles de ambos (Fig.4.16). Donde se ve una ligera curva en el punto (100° motor_derecho, 79° motor_izquierdo), en cual se genera el punto de intersección de ambos dedos justo cuando están en medio de la pinza y no hay ningún objeto entre ellos. Con estas ecuaciones también se graficaron los ángulos del dedo derecho contra los ángulos del dedo izquierdo con distintas separaciones entre los centros de los motores Fig.4.18. Se observa en las gráficas, como el punto de intersección entre los dedos se acentúa en una curvatura a medida que se alejan los centros de los motores.

Además se hicieron pruebas con la ecuación 4.5 simulando una separación de 6cm entre la punta de los dedos, es decir con un objeto entre ellos. En la Fig 4.17, se muestran algunas de sus configuraciones.

En este capítulo se mostró a detalle cada uno de los componentes de la pinza robótica. También se explicaron las ecuaciones que permiten obtener el espacio de configuraciones que puede alcanzar con los dedos derecho e izquierdo, para este y cualquier otra pinza de movimiento angular con dedos independientes. En el siguiente capítulo se explicará el diseño de los controladores para esta pinza.

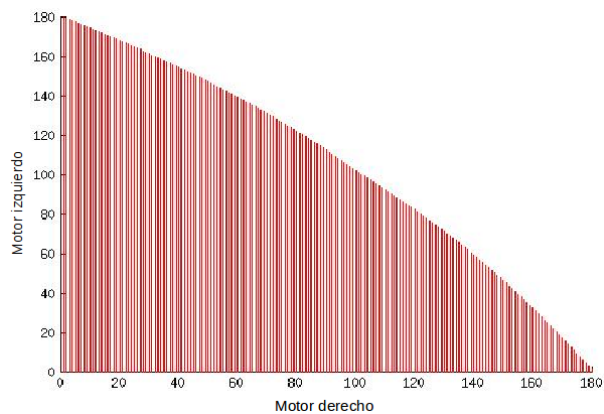


FIGURA 4.16: Se grafican los ángulos que van de $0 - 180^\circ$ para ambos motores el punto de intersección se genera en 100° motor_derecho, 79° motor_izquierdo.

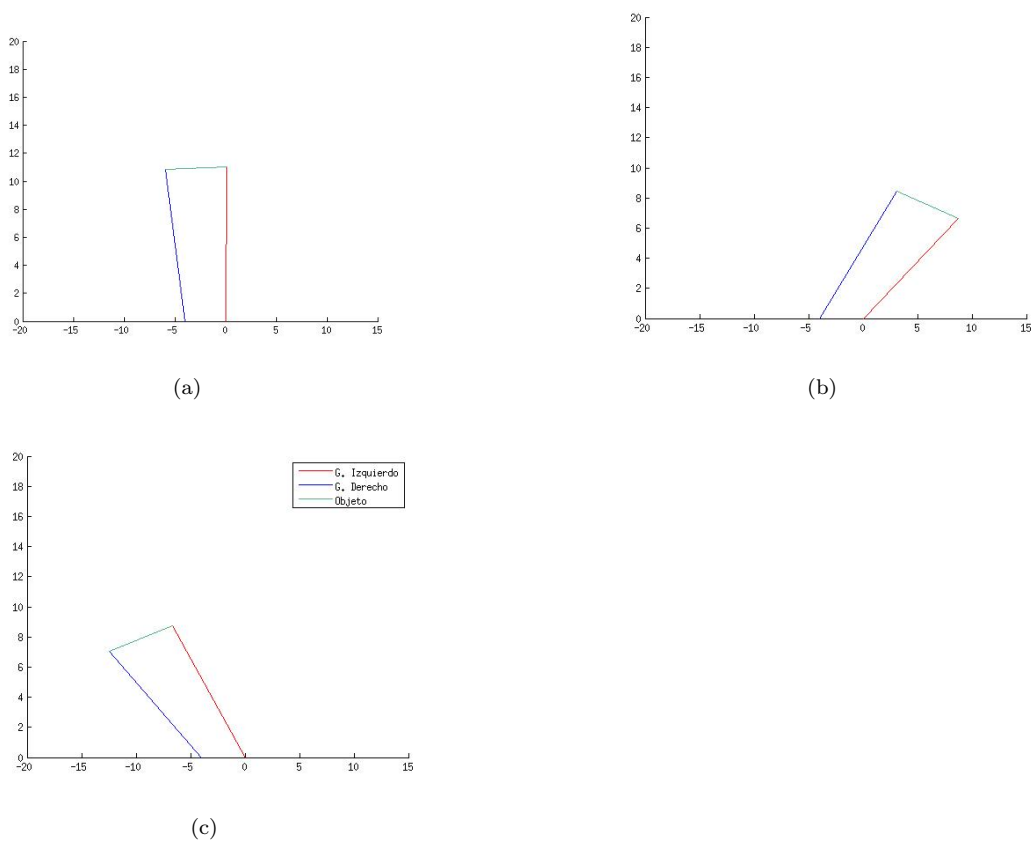


FIGURA 4.17: a) Cuando el objeto se encuentra en medio de la pinza, b) el objeto se encuentra desplazado a la derecha c) La posición cuando el objeto se encuentra desplazado a la izquierda

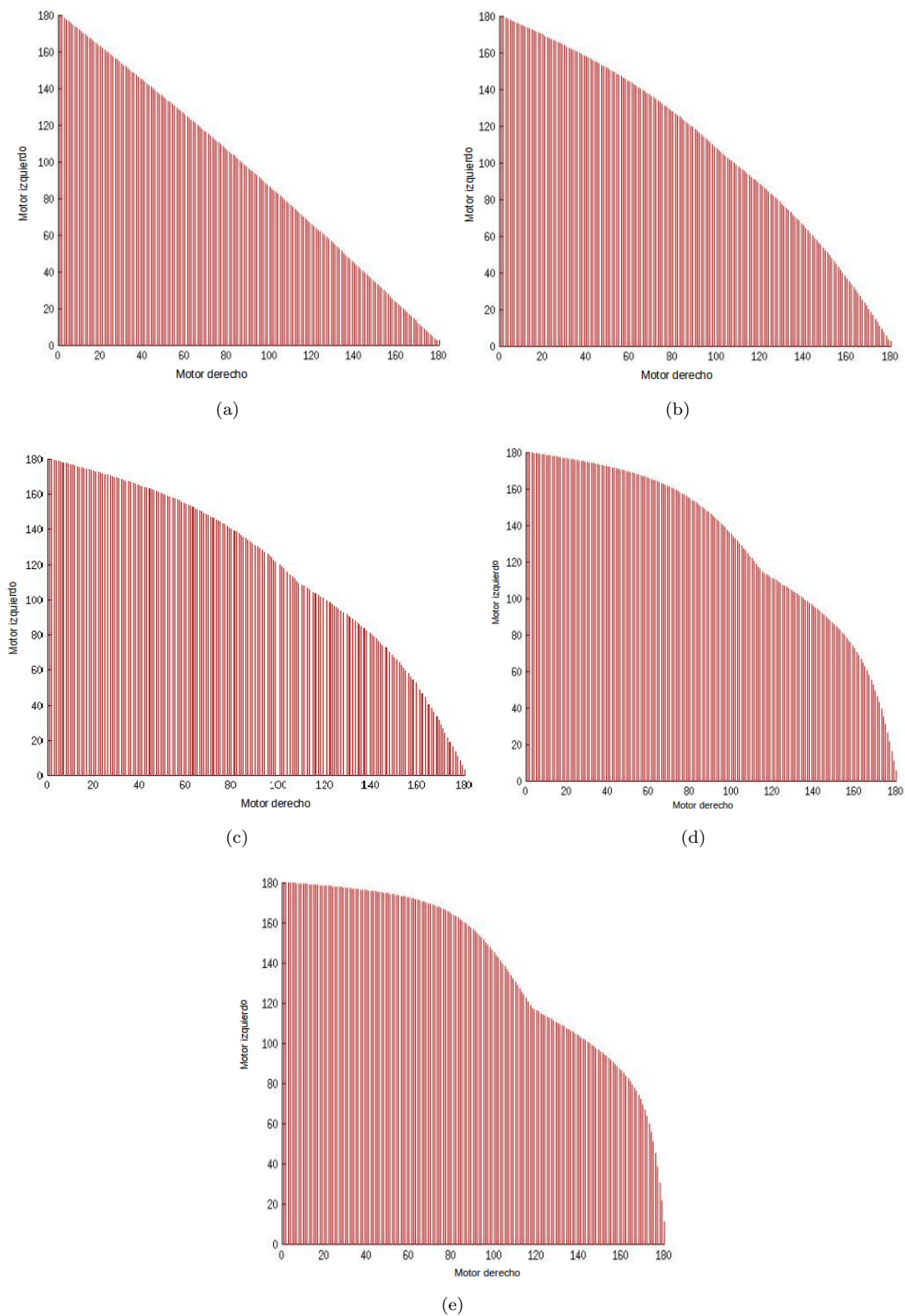


FIGURA 4.18: Ángulos obtenidos con distintas separaciones entre los ejes de los motores, dg, a) 1cm. b) 5cm. c) 7cm. d) 9cm. e) 10cm.

Capítulo 5

Diseño de los controladores

En esta sección se explican cada uno de los controladores realizados. Primero se empezará con el control PID, posteriormente el controlador Difuso Tipo-I, y por último el Difuso Tipo-II.

5.1. Control Híbrido PID

En la Fig.5.1 se muestra el esquema del controlador PID híbrido que se diseñó, el cual opera de la siguiente manera:

- **Posición:** Cuando un objeto es detectado, este controlador es activado y el objeto es colocado en el centro de la pinza. Cuando la salida del PID es menor a un umbral determinado, este controlador es apagado. El umbral se establece como el porcentaje absoluto del valor de referencia, en este caso es del 98 %.
- **Fuerza:** Una vez que el controlador PID de posición es apagado y que el objeto está centrado, el controlador PID de fuerza es activado. Este es usado para aplicar suficiente fuerza al objeto para manipularlo sin dañarlo.

Ambos controladores operan en lazo cerrado para mantener una referencia preestablecida, mismo que asegurara que los motores se mantengan con la velocidad y posición deseada durante la manipulación.

5.1.1. Control PID de Posición

Para el control PID de posición la señal de realimentación es el eje del motor. El algoritmo ideal del PID de posición (el cuál será denotado por P) se escribe como:

$$\frac{U_P(s)}{E_P(s)} = K_{Pp} \left[1 + \frac{1}{T_{Pi}s} + T_{Pd}s \right] \quad (5.1)$$

y aplicando el método de diferencia hacia atrás, se obtiene:

$$\frac{U_P(s)}{E_P(s)} = K_{Pp} \left[1 + \frac{1}{T_{Pi}s} + T_{Pd}s \right] \quad (5.2)$$

$$\xrightarrow{B.D.} u_P(t) = e_P(t) K_{Pp} \left[1 + \frac{T_{Ps}}{T_{Pi}(1-z^{-1})} + T_{Pd} \frac{(1-z^{-1})}{T_{Ps}} \right]$$

y simplificando:

$$\begin{aligned} u_P(t) = u_P(t-1) + K_{Pp} [e_P(t) - e_P(t-1)] & \quad (5.3) \\ & + \frac{K_{Pp} T_{Ps}}{T_{Pi}} e_P(t) \\ & + \frac{K_{Pp} T_{Pd}}{T_{Ps}} [e_P(t) - 2e_P(t-1) + e_P(t-2)] \end{aligned}$$

Donde $e_p(t) = Ref_{pos} - Pos_{\omega(t)}$ es el error de posición. De esta manera, el PID de posición colocará al objeto en el centro de la pinza. Cuando el error de posición sea suficientemente pequeño, es decir

$$e_p(t) \leq h, \text{ con } h = \pm 2\% Ref_{pos} \quad (5.4)$$

se realizará una conmutación al PID de fuerza.

5.1.2. Control PID de Fuerza

Como se mencionó anteriormente, cada dedo tiene su propio motor, y en cada motor se pueden consultar los parámetros velocidad, temperatura y posición. Además en la punta de cada dedo se colocó un sensor de fuerza, la señal del sensor es usada como realimentación para este controlador PID de fuerza. Se desea saber que valor del sensor de fuerza equivale al torque que el motor ejerce cuando se incrementa la velocidad de éste.

Para este propósito se realizó una caracterización del sensor de fuerza cuando la pinza sostiene un objeto y el torque que tiene el motor. El resultado es una ecuación de segundo orden, que se muestra a continuación:

$$v(f) = 0.069913f^2 + 0.266199f + 0.708132 \quad (5.5)$$

El algoritmo ideal del PID de fuerza (el cuál será denotado por F) se escribe como:

$$\frac{U_F(s)}{E_F(s)} = K_{Fp} \left[1 + \frac{1}{T_{Fi}s} + T_{Fd}s \right] \quad (5.6)$$

en donde, aplicando el método de diferencia hacia atrás, se obtiene:

$$\frac{U_F(s)}{E_F(s)} = K_{Fp} \left[1 + \frac{1}{T_{Fi}s} + T_{Fd}s \right] \quad (5.7)$$

$$\xrightarrow{B.D.} u_F(t) = e_F(t) K_{Fp} \left[1 + \frac{T_{Fs}}{T_{Fi}(1-z^{-1})} + T_{Fd} \frac{(1-z^{-1})}{T_{Fs}} \right]$$

y simplificando :

$$\begin{aligned} u_F(t) = u_F(t-1) + K_{Fp} [e_F(t) - e_F(t-1)] & \quad (5.8) \\ & + \frac{K_{Fp} T_{Fs}}{T_{Fi}} e_F(t) \\ & + \frac{K_{Fp} T_{Fd}}{T_{Fs}} [e_F(t) - 2e_F(t-1) + e_F(t-2)] \end{aligned}$$

Posteriormente al diseño de los controladores de posición y fuerza, se requirió de un algoritmo que permitiera la conmutación entre ellos. El algoritmo híbrido propuesto es el que se describe en el algoritmo 3.

5.2. Control Difuso tipo I

En el desarrollo de éste controlador, como se mencionó en la sección 3.1.1, no se requiere del modelo matemático del sistema. Por lo tanto este es un controlador heurístico basado en conocimiento, utilizado para controlar un sistema indefinido y complejo. El diagrama de bloques del sistema de la pinza junto con el controlador difuso se muestra en la Fig.5.2. El bloque del controlador difuso se compone de los siguientes elementos:

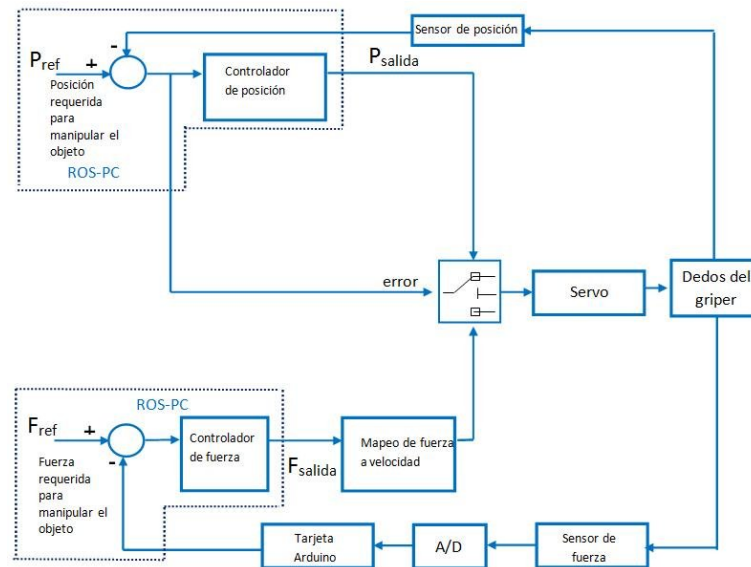


FIGURA 5.1: Esquema de control que cambia de Posición a Fuerza. El control de posición centra el objeto en medio de la pinza. Una vez que el objeto es centrado, el control de posición es apagado, y enciende el control de fuerza, entonces el objeto puede ser manipulado.

fusificación, mecanismo de inferencia, base de reglas y defusificación. Entonces, para es-

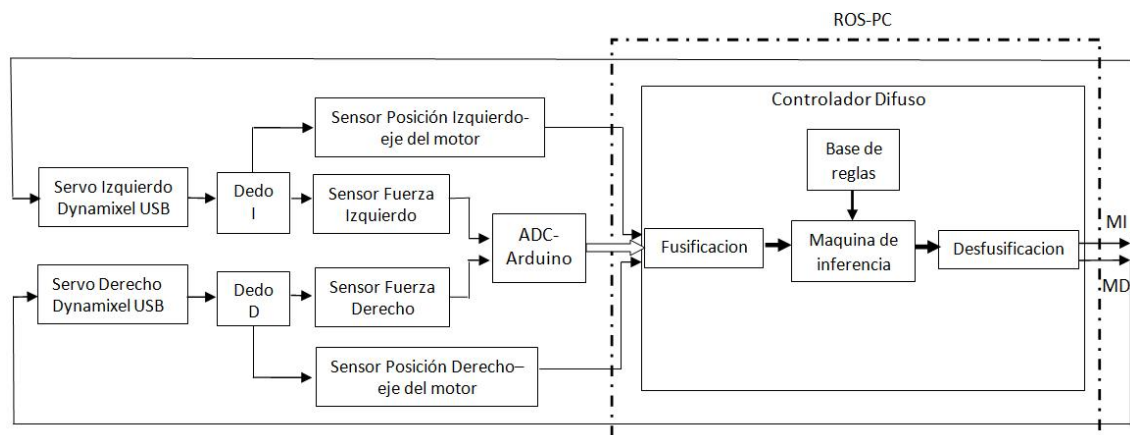


FIGURA 5.2: control difuso de la pinza.

te controlador las entradas serán: las lecturas de los dos sensores de fuerza (izquierdo y derecho) y las mediciones de los dos sensores de posición de cada motor (izquierdo y derecho) y por último la salida del controlador será un comando de velocidad para motor.

Para cada sensor de fuerza, es decir para el derecho y el izquierdo, se propusieron cuatro funciones de membresía tipo trapezoidal éstas son :

SXA= Sensor X Apagado

SXTS= Sensor X tocando suave

Algoritmo 3 Control Híbrido de posición-fuerza

Entrada: Sensor de posición, sensor de fuerza

Salida: Comando de velocidad de motor derecho, motor izquierdo.

```

1: mientras no exista objeto entre los dedos de la pinza hacer
2:   Mantener velocidad constante en ambos dedos, (derecho izquierdo).
3:   si hay objeto hacia la derecha entonces
4:     PID Posición derecho encendido.
5:     si objeto centrado entonces
6:       PID Posición derecho apagado, PID Fuerza encendido.
7:     fin si
8:   si no
9:     si Hay objeto hacia la izquierda entonces
10:      PID Posición izquierdo encendido.
11:      si Objeto centrado entonces
12:        PID Posición izquierdo apagado, PID Fuerza encendido.
13:      fin si
14:    fin si
15:  fin si
16:  si PID Fuerza encendido entonces
17:    Aplicar misma velocidad en ambos dedos.
18:  fin si
19: fin mientras
20: devolver cierto

```

SXTF= Sensor tocando fuerte y

SXAP= Sensor X apretando

donde X será D para sensor derecho y I para sensor izquierdo. La función de pertenencia trapezoidal para el sensor derecho se muestra en la Fig. 5.3.

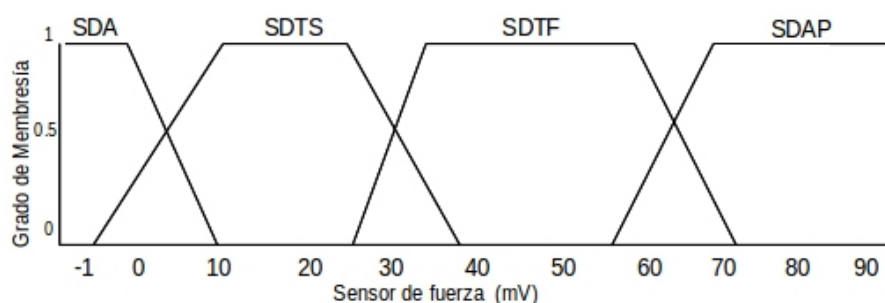


FIGURA 5.3: Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Derecho.

Debido a que la función trapezoidal es una función no lineal, como se observa en la Fig.5.3, es necesario realizar una evaluación por intervalos; como se había mencionado en la sección 3.1.1.2, la cual aplicada a nuestro caso queda de la siguiente manera:

$$\mu_{Apagado}(SD) = \begin{cases} 0 & SD > 10 \\ 1 & SD < 0 \\ (10 - SD)/(10) & 0 \leq SD \leq 10 \end{cases}$$

$$\mu_{T.Suave}(SD) = \begin{cases} 0 & (SD < -1) \text{ ó } (SD > 38) \\ (SD - (-1))/(8 - (-1)) & -1 \leq SD < 8 \\ 1 & 8 \leq SD \leq 30 \\ (38 - SD)/(38 - 30) & 30 < SD \leq 38 \end{cases}$$

$$\mu_{T.Fuerte}(SD) = \begin{cases} 0 & (SD < 22) \text{ ó } (SD > 72) \\ (SD - 22)/(32 - 22) & 22 \leq SD < 32 \\ 1 & 32 \leq SD \leq 62 \\ (72 - SD)/(72 - 62) & 62 < SD \leq 72 \end{cases}$$

$$\mu_{Apretando}(SD) = \begin{cases} 0 & SD < 58 \\ 1 & SD > 70 \\ (SD - 58)/(70 - 58) & 58 \leq SD \leq 70 \end{cases}$$

De la misma manera se presenta la función trapezoidal del sensor de fuerza izquierdo en la Fig 5.4, así como su respectiva evaluación por intervalos.

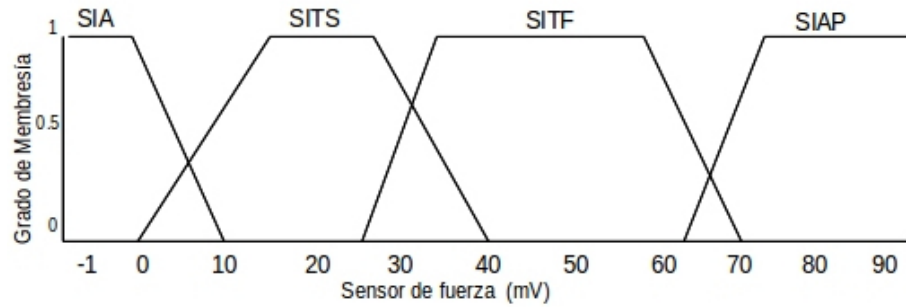


FIGURA 5.4: Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Izquierdo.

$$\mu_{Apagado}(SI) = \begin{cases} 1 & SI < 0 \\ 0 & SI > 10 \\ (10 - SI)/(10) & 0 \leq SI \leq 10 \end{cases}$$

$$\mu_{T.Suave}(SI) = \begin{cases} 0 & (SI < 0) \text{ ó } (SI > 40) \\ (SI - (0))/(10 - (0)) & 0 \leq SI < 10 \\ 1 & 10 \leq SI \leq 28 \\ (40 - SI)/(40 - 28) & 28 < SI \leq 40 \end{cases}$$

$$\mu_{T.Fuerte}(SI) = \begin{cases} 0 & (SI < 22) \text{ ó } (SI > 69) \\ (SI - 22)/(33 - 22) & 22 \leq SI < 33 \\ 1 & 32 \leq SI \leq 60 \\ (69 - SI)/(69 - 60) & 60 < SI \leq 69 \end{cases}$$

$$\mu_{Apretando}(SI) = \begin{cases} 0 & SI < 62 \\ 1 & SI > 70 \\ (SI - 62)/(70 - 62) & 62 \leq SI \leq 70 \end{cases}$$

Por otra parte, se definió una función de membresía trapezoidal para la posición angular del motor derecho y el motor izquierdo, éstas son :

MXMA = Motor X Muy Abierto

MXA = Motor X Abierto

MXC = Motor X Centrado

MXCR = Motor X Cerrado

en donde *X* será *D* para motor derecho y *I* para motor izquierdo. A continuación se muestra la función de pertenencia trapezoidal para el motor derecho en la Fig. 5.5.

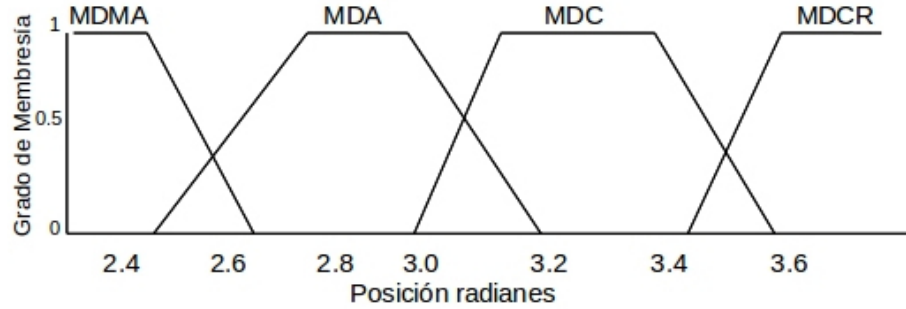


FIGURA 5.5: Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Izquierdo.

$$\mu_{Muy_abierto}(MD) = \begin{cases} 0 & \text{si } MD > 3.0 \\ 1 & \text{si } MD < 1 \\ (3.0 - MD)/(3.0 - 1.0) & \text{si } 1.0 \leq MD \leq 3.0 \end{cases}$$

$$\mu_{Abierto}(MD) = \begin{cases} 0 & (MD < 2.60) \text{ ó } (MD > 3.34) \\ (MD - 2.60)/(3.14 - 2.60) & 2.60 \leq MD < 3.14 \\ 1 & 3.14 \leq MD \leq 3.26 \\ (3.34 - MD)/(3.34 - 3.26) & 3.26 < MD \leq 3.34 \end{cases}$$

$$\mu_{centrado}(MD) = \begin{cases} 0 & (MD < 3.26) \text{ ó } (MD > 3.6) \\ (MD - 3.26)/(3.30 - 3.26) & 3.26 \leq MD < 3.30 \\ 1 & 3.30 \leq MD \leq 3.5 \\ (3.35 - MD)/(3.5 - 3.6) & 3.5 < MD \leq 3.6 \end{cases}$$

$$\mu_{cerrado}(MD) = \begin{cases} 0 & MD < 3.6 \\ 1 & MD > 3.66 \\ (MD - 3.6)/(3.66 - 3.60) & 3.4 \leq MD \leq 3.66 \end{cases}$$

De la misma manera se presenta la función trapezoidal del motor izquierdo en la Fig 5.6, así como su respectiva evaluación por intervalos.

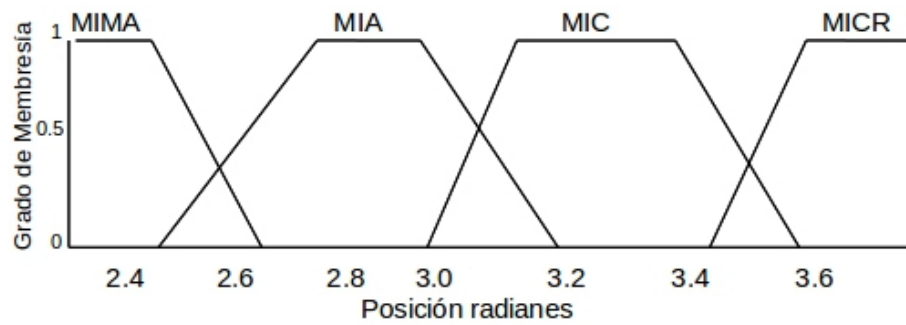


FIGURA 5.6: Función de membresía trapezoidal del difuso Tipo-I, aplicado el Sensor de fuerza Izquierdo.

$$\mu_{Muy_abierto}(MI) = \begin{cases} 0 & MI < 1.98 \\ 1 & MI > 2.2 \\ (1.98 - MI)/(1.98 - 2.2) & 1.98 \leq MI \leq 2.2 \end{cases}$$

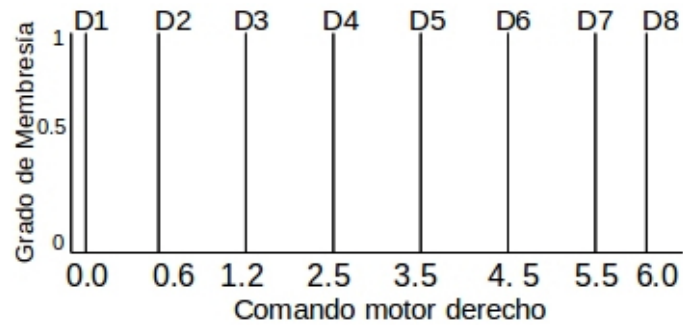
$$\mu_{Abierto}(MI) = \begin{cases} 0 & (MI > 2.46) \text{ ó } (MI < 1.91) \\ (MI - 2.46)/(1.99 - 2.46) & 1.99 \leq MI \leq 2.46 \\ 1 & 1.94 \leq MI \leq 1.99 \\ (1.91 - MI)/(1.91 - 1.94) & 1.91 \leq MI < 1.94 \end{cases}$$

$$\mu_{centrado}(MI) = \begin{cases} 0 & (MI > 1.95) \text{ ó } (MI < 1.70) \\ (MI - 1.95)/(1.93 - 1.95) & 1.93 < MI \leq 1.95 \\ 1 & 1.87 \leq MI \leq 1.93 \\ (1.70 - MI)/(1.70 - 1.87) & 1.70 \leq MI \leq 1.87 \end{cases}$$

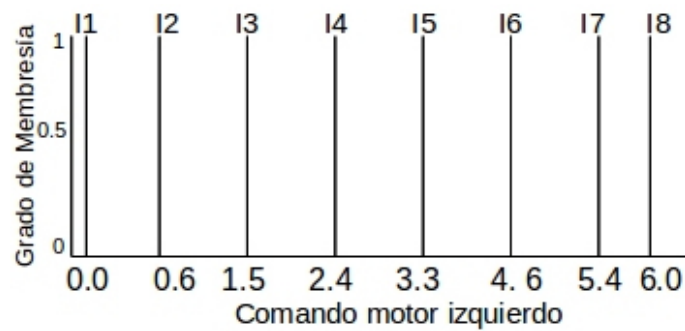
$$\mu_{cerrado}(MI) = \begin{cases} 0 & MI > 1.8 \\ 1 & MI < 1.7 \\ (MI - 1.8)/(1.7 - 1.8) & 1.7 \leq MI \leq 1.8 \end{cases}$$

Finalmente, las salidas son funciones de pertenencia tipo singleton y se declaran ocho para cada motor, las cuales son $X1, X2, X3, X4, X5, X6, X7, X8$ en donde X tomará el valor de D para la salida derecha e I para la salida izquierda. Estas salidas se ordenan

desde el valor de velocidad más bajo, al valor de velocidad más alto que puede ejecutar el motor.








(a)



(b)

FIGURA 5.7: a) Ejemplo de agarre envolvente con la mano robótica de la DLR [9]. b) Agarre de precisión, hecho con una mano robot de tres dedos [59]

Para mostrar la complejidad del controlador se muestran las superficies de control en la Fig. 5.8 considerando el sensor derecho *vs* el motor derecho y en la Fig. 5.9 teniendo en cuenta las variables de entrada del sensor izquierdo *vs* el sensor derecho. Se puede observar que en la Fig. 5.9 se aprecian algunas zonas booleanas pero en otras zonas se observan superficies no lineales y suaves.

con objeto entre los dedos			sin objeto entre los dedos	
case1 : objeto a la izquierda	case2 : objeto centrado	case3 : objeto a la derecha	case4 : pinza abierta	case5 : pinza cerrada
				
Reglas:18,26,27	Rules:24,25,28,29,23,33	Reglas:21,28,30	Reglas:1	Reglas:4

CUADRO 5.1: Casos para la pinza.

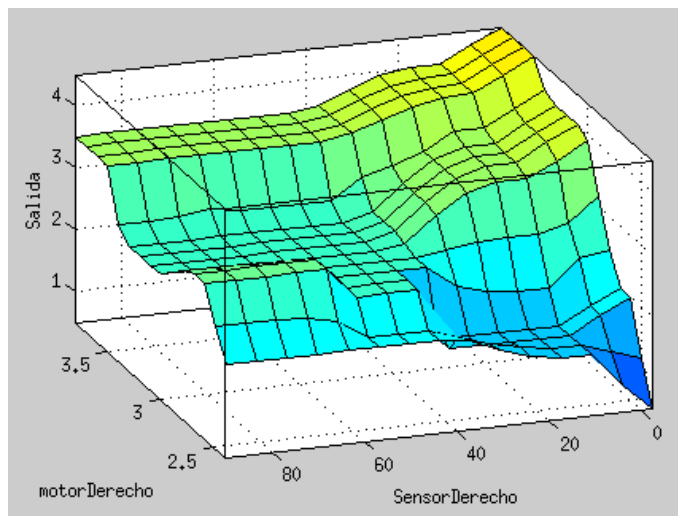


FIGURA 5.8: Salida del controlador difuso en función de las variables de entrada de motor derecho vs sensor derecho.

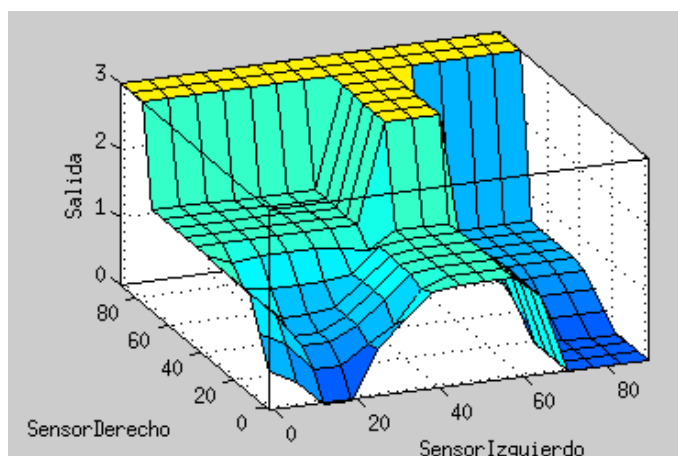


FIGURA 5.9: Salida del controlador difuso en función de las variables de entrada de los sensores derecho e izquierdo.

Después de definir las funciones de pertenencia, se diseñó el conjunto de reglas, el cual se basó básicamente en dos situaciones. Una es cuando la pinza toma un objeto entre los dedos y, cuando no hay ningún objeto en la pinza. Cuando hay un objeto en la pinza, se pueden presentar 3 casos posibles:

- El objeto está ubicado en el lado izquierdo con respecto al centro de la pinza.
- El objeto está centrado con respecto al centro de la pinza.
- El objeto está ubicado en el lado derecho con respecto al centro de la pinza.

A partir de estos casos y la combinación entre ellos de las variables de entrada, se obtuvieron 50 reglas para controlar la posición del objeto y la fuerza, las cuales se

muestran en la tabla 5.2. Cabe mencionar que el objetivo de estas reglas es centrar el objeto en medio de la pinza y aplicar la fuerza requerida de manera simultánea para manipular el objeto.

Por último, cabe mencionar que todas las evaluaciones de las funciones de pertenencia trapezoidal así como el conjunto de reglas, son programadas en un Nodo de ROS, el cual se había mencionado en la sección 4.3.

5.3. Control difuso Tipo-II

En los sistemas difusos Tipo-II las funciones de pertenencia ahora pueden devolver un rango de valores que varía dependiendo de la incertidumbre involucrada, no solo en la entrada sino también en la misma función de membresía. Para sistemas difusos tipo-I la función de membresía es bidimensional.

La Figura 5.10 muestran los límites superior e inferior de las funciones de pertenencia del controlador difuso Tipo-II.

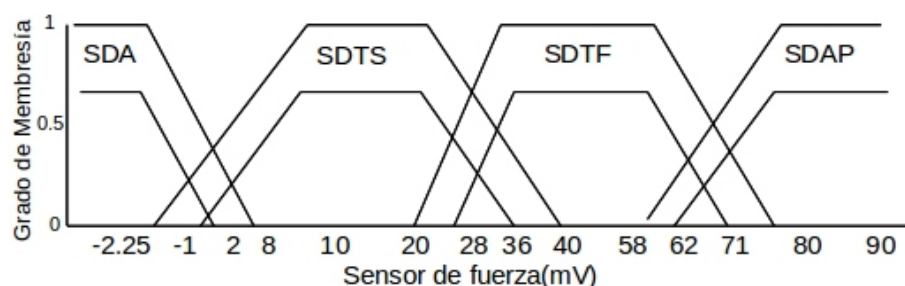


FIGURA 5.10: Funciones de pertenencia altas y bajas, para sensor y motor derecho.

La evaluación por intervalos para el sensor Derecho parte alta se muestra a continuación:

$$\mu_{Apagado}(\overline{SD}) = \begin{cases} 0 & SD > 10 \\ 1 & SD < 0 \\ (10 - SD)/(10) & 0 \leq SD \leq 10 \end{cases}$$

$$\mu_{T.Suave}(\overline{SD}) = \begin{cases} 0 & (SD < -1) \text{ ó } (SD > 38) \\ (SD - (-1))/(8 - (-1)) & -1 \leq SD < 8 \\ 1 & 8 \leq SD \leq 30 \\ (38 - SD)/(38 - 30) & 30 < SD \leq 38 \end{cases}$$

No	Reglas difusas	$M_I(t), (M_D(t))$
1	Si (SI, Apagado) y (SD Apagado) y (MI. Muy_abierto) y (MD. Muy_abierto)	→ I4, D2
2	Si(SI, Apagado) y (SD Apagado) y (MI. abierto) y (MD. Muy_abierto)	→ I4, D3
3	Si(SI, Apagado) y (SD Apagado) y (MI. centrado) y (MD. centrado)	→ I2, D2
4	Si(SI, Apagado) y (SD Apagado) y (MI. cerrado) y (MD. cerrado)	→ I4, D4
5	Si(SI, Apagado) y (SD T. Suave) y (MI. abierto) y (MD. abierto)	→ I4, D4
6	Si(SI, Apagado) y (SD Apagado) y (MI. centrado) y (MD. centrado)	→ I4, D4
7	Si(SI, Apagado) y (SD Apagado) y (MI. cerrado) y (MD. cerrado)	→ I2, D4
8	Si(SI, Apagado) y (SD Apagado) y (MI. abierto) y (MD. centrado)	→ I2, D4
9	Si(SI, Apagado) y (SD Apagado) y (MI. abierto) y (MD. cerrado)	→ I4, D4
10	Si(SI, Apagado) y (SD Apagado) y (MI. centrado) y (MD. abierto)	→ I2, D4
11	Si(SI, Apagado) y (SD Apagado) y (MI. centrado y (MD. cerrado)	→ I2, D4
12	Si(SI, Apagado) y (SD Apagado) y (MI. cerrado) y (MD. abierto)	→ I2, D2
13	Si(SI, Apagado) y (SD Apagado) y (MI. cerrado) y (MD. cerrado)	→ I1, D1
14	Si(SI, T.Suave) y (SD Apagado) y (MI. abierto) y (MD. abierto)	→ I2, D2
15	Si(SI, T.Suave) y (SD Apagado) y (MI. centrado) y (MD. centrado)	→ I2, D2
16	Si(SI, T.Suave) y (SD Apagado) y (MI. cerrado) y (MD. cerrado)	→ I2, D4
17	Si(SI, T.Suave) y (SD Apagado) y (MI. abierto) y (MD. centrado)	→ I2, D3
18	Si(SI, T.Suave) y (SD Apagado) y (MI. abierto) y (MD. cerrado)	→ I2, D3
19	Si(SI, T.Suave) y (SD Apagado) y (MI. centrado) y (MD. abierto)	→ I2, D4
20	Si(SI, T.Suave) y (SD Apagado) y (MI. centrado) y (MD. cerrado)	→ I2, D4
21	Si(SI, T.Suave) y (SD Apagado) y (MI. cerrado) y (MD. abierto)	→ I2, D4
22	Si(SI, T.Suave) y (SD Apagado) y (MI. cerrado) y (MD. centrado)	→ I2, D4
23	Si(SI, T.Suave) y (SD T. Suave) y (MI. abierto) y (MD. abierto)	→ I2, D4
24	Si(SI, T.Suave) y (SD T. Suave) y (MI. centrado) y (MD. centrado)	→ I2, D4
25	Si(SI, T.Suave) y (SD T. Suave) y (MI. cerrado) y (MD. cerrado)	→ I2, D4
26	Si(SI, T.Suave) y (SD T. Suave) y (MI. abierto) y (MD. centrado)	→ I3, D3
27	Si(SI, T.Suave) y (SD T. Suave) y (MI. abierto) y (MD. cerrado)	→ I3, D2
28	Si(SI, T.Suave) y (SD T. Suave) y (MI. centrado) y (MD. abierto)	→ I4, D2
29	Si(SI, T.Suave) y (SD T. Suave) y (MI. centrado) y (MD. centrado)	→ I2, D4
30	Si(SI, T.Suave) y (SD T. Suave) y (MI. cerrado) y (MD. abierto)	→ I2, D4
31	Si(SI, T.Suave) y (SD T. Suave) y (MI. cerrado) y (MD. centrado)	→ I2, D4
32	Si(SI, T.fuerte) y (SD T. fuerte) y (MI. centrado) y (MD. centrado)	→ I4, D4
33	Si(SI, T.fuerte) y (SD T. fuerte) y (MI. cerrado) y (MD. cerrado)	→ I4, D4
34	Si(SI, T.fuerte) y (SD T. fuerte) y (MI. centrado) y (MD. centrado)	→ I4, D4
35	Si(SI, T.fuerte) y (SD T. fuerte) y (MI. cerrado) y (MD. centrado)	→ I3, D4
36	Si(SI, T.fuerte) y (SD T. Suave) y (MI. centrado) y (MD. centrado)	→ I3, D4
37	Si(SI, T.fuerte) y (SD T. Suave) y (MI. cerrado) y (MD. cerrado)	→ I4, D4
38	Si(SI, T.fuerte) y (SD T. Suave) y (MI. centrado) y (MD. cerrado)	→ I3, D3
39	Si(SI, T.fuerte) y (SD T. Suave) y (MI. cerrado) y (MD. centrado)	→ I4, D4
40	Si(SI, T. Suave) y (SD T. fuerte) y (MI. centrado) y (MD. centrado)	→ I4, D4
41	Si(SI, T. Suave) y (SD T. fuerte) y (MI. cerrado) y (MD. cerrado)	→ I4, D4
42	Si(SI, T. Suave) y (SD T. fuerte) y (MI. centrado) y (MD. cerrado)	→ I3, D3
43	Si(SI, T. Suave) y (SD T. fuerte) y (MI. cerrado) y (MD. centrado)	→ I3, D3
44	Si(SI, T. fuerte) y (SD Apagado) y (MI. centrado) y (MD. centrado)	→ I3, D4
45	Si(SI, T. fuerte) y (SD Apagado) y (MI. cerrado) y (MD. cerrado)	→ I3, D4
46	Si(SI, T. fuerte) y (SD Apagado) y (MI. centrado) y (MD. centrado)	→ I3, D4
47	Si(SI, T. fuerte) y (SD Apagado) y (MI. cerrado) y (MD. centrado)	→ I4, D4
48	Si(SI, T. Apagado) y (SD T. Fuerte) y (MI. centrado) y (MD. centrado)	→ I4, D2
49	Si(SI, T. Apagado) y (SD T. Fuerte) y (MI. cerrado) y (MD. cerrado)	→ I4, D2
50	Si(SI, T. Apagado) y (SD T. Fuerte) y (MI. centrado) y (MD. cerrado)	→ I4, D2

CUADRO 5.2: Reglas para difuso Tipo-I.

$$\mu_{T.Fuerte}(\overline{SD}) = \begin{cases} 0 & (SD < 22) \text{ ó } (SD > 72) \\ (SD - 22)/(32 - 22) & 22 \leq SD < 32 \\ 1 & 32 \leq SD \leq 62 \\ (72 - SD)/(72 - 62) & 62 < SD \leq 72 \end{cases}$$

$$\mu_{Apretando}(\overline{SD}) = \begin{cases} 0 & SD < 58 \\ 1 & SD > 70 \\ (SD - 58)/(70 - 58) & 58 \leq SD \leq 70 \end{cases}$$

Así mismo la evaluación por intervalos para el sensor Derecho parte baja se muestra a continuación:

$$\mu_{Apagado}(SD) = \begin{cases} 0 & \underline{SD} > 8 \\ 1 & \underline{SD} < -1 \\ (8 - \underline{SD})/(8 + 1) & -1 \leq \underline{SD} \leq 8 \end{cases}$$

$$\mu_{T.Suave}(SD) = \begin{cases} 0 & (\underline{SD} < 2) \text{ ó } (\underline{SD} > 36) \\ (\underline{SD} - 2)/(10 - 2) & 2 \leq \underline{SD} < 10 \\ 1 & 10 \leq \underline{SD} \leq 28 \\ (36 - \underline{SD})/(36 - 28) & 28 < \underline{SD} \leq 36 \end{cases}$$

$$\mu_{T.Fuerte}(SD) = \begin{cases} 0 & (SD < 28) \text{ ó } (SD > 71) \\ (SD - 28)/(38 - 28) & 28 \leq SD < 38 \\ 1 & 38 \leq SD \leq 61 \\ (71 - SD)/(71 - 61) & 61 < SD \leq 71 \end{cases}$$

$$\mu_{Apretando}(SD) = \begin{cases} 0 & \underline{SD} < 62 \\ 1 & \underline{SD} > 73 \\ (\underline{SD} - 62)/(73 - 62) & 62 \leq \underline{SD} \leq 73 \end{cases}$$

En la Fig. 5.11 se muestran las funciones de pertenencia altas y bajas para el sensor izquierdo.

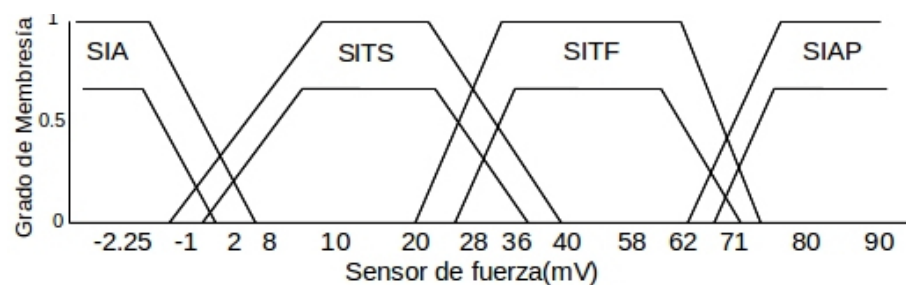


FIGURA 5.11: Funciones de pertenencia altas y bajas, para sensor y motor derecho.

La evaluación por intervalos para el sensor Izquierdo parte alta se muestra a continuación,

$$\mu_{\text{Apagado}}(\overline{SI}) = \begin{cases} 1 & SI < 0 \\ 0 & SI > 10 \\ (10 - SI)/(10) & 0 \leq SI \leq 10 \end{cases}$$

$$\mu_{T.Suave}(\overline{SI}) = \begin{cases} 0 & (SI < 0) \text{ ó } (SI > 40) \\ (SI - (0))/(10 - (0)) & 0 \leq SI < 10 \\ 1 & 10 \leq SI \leq 28 \\ (40 - SI)/(40 - 28) & 28 < SI \leq 40 \end{cases}$$

$$\mu_{T.Fuerte}(\overline{SI}) = \begin{cases} 0 & (SI < 22) \text{ ó } (SI > 69) \\ (SI - 22)/(33 - 22) & 22 \leq SI < 33 \\ 1 & 32 \leq SI \leq 60 \\ (69 - SI)/(69 - 60) & 60 < SI \leq 69 \end{cases}$$

$$\mu_{\text{Apretando}}(\overline{SI}) = \begin{cases} 0 & SI < 62 \\ 1 & SI > 70 \\ (SI - 62)/(70 - 62) & 62 \leq SI \leq 70 \end{cases}$$

La evaluación por intervalos para el sensor Izquierdo parte baja se muestra a continuación,

$$\mu_{\text{Apagado}}(\underline{SI}) = \begin{cases} 1 & \underline{SI} < 0 \\ 0 & \underline{SI} > 8 \\ (8 - \underline{SI})/(8) & 0 \leq \underline{SI} \leq 8 \end{cases}$$

$$\mu_{T.Suave}(\underline{SI}) = \begin{cases} 0 & (\underline{SI} < 6) \text{ ó } (\underline{SI} > 32) \\ (\underline{SI} - 6)/(14 - 6) & 6 \leq \underline{SI} < 14 \\ 1 & 14 \leq \underline{SI} \leq 22 \\ (32 - \underline{SI})/(32 - 22) & 22 < \underline{SI} \leq 32 \end{cases}$$

$$\mu_{T.Fuerte}(\underline{SI}) = \begin{cases} 0 & (\underline{SI} < 28) \text{ ó } (\underline{SI} > 68) \\ (\underline{SI} - 28)/(37 - 28) & 28 \leq \underline{SI} < 37 \\ 1 & 37 \leq \underline{SI} \leq 58 \\ (68 - \underline{SI})/(68 - 58) & 58 < \underline{SI} \leq 68 \end{cases}$$

$$\mu_{Apretando}(\underline{SI}) = \begin{cases} 0 & \underline{SI} < 63 \\ 1 & \underline{SI} > 71 \\ (\underline{SI} - 63)/(71 - 63) & 63 \leq \underline{SI} \leq 71 \end{cases}$$

Las funciones de pertenencia altas y bajas trapezoidal para el motor derecho se muestran en la Fig. 5.12.

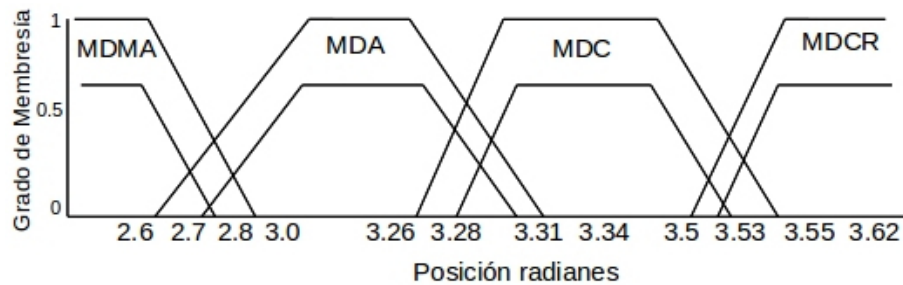


FIGURA 5.12: Funciones de pertenencia altas y bajas, para sensor y motor derecho.

La evaluación por intervalos para el motor Derecho parte alta se muestra a continuación,

$$\mu_{Muy_abierto}(\overline{MD}) = \begin{cases} 0 & MD > 3.0 \\ 1 & MD < 1 \\ (3.0 - MD)/(3.0 - 1.0) & 1.0 \leq MD \leq 3.0 \end{cases}$$

$$\mu_{Abierto}(\overline{MD}) = \begin{cases} 0 & (MD < 2.60) \text{ ó } (MD > 3.34) \\ (MD - 2.60)/(3.14 - 2.60) & 2.60 \leq MD < 3.14 \\ 1 & 3.14 \leq MD \leq 3.26 \\ (3.34 - MD)/(3.34 - 3.26) & 3.26 < MD \leq 3.34 \end{cases}$$

$$\mu_{centrado}(\overline{MD}) = \begin{cases} 0 & (MD < 3.26) \text{ ó } (MD > 3.6) \\ (MD - 3.26)/(3.30 - 3.26) & 3.26 \leq MD < 3.30 \\ 1 & 3.30 \leq MD \leq 3.5 \\ (3.35 - MD)/(3.5 - 3.6) & 3.5 < MD \leq 3.6 \end{cases}$$

$$\mu_{cerrado}(\overline{MD}) = \begin{cases} 0 & MD < 3.6 \\ 1 & MD > 3.66 \\ (MD - 3.6)/(3.66 - 3.60) & 3.4 \leq MD \leq 3.66 \end{cases}$$

La evaluación por intervalos para el motor Derecho parte baja se muestra a continuación,

$$\mu_{Muy_abierto}(MD) = \begin{cases} 0 & MD > 2.80 \\ 1 & MD < 0.8 \\ (2.80 - MD)/(2.80 - 0.8) & 0.8 \leq MD \leq 2.80 \end{cases}$$

$$\mu_{Abierto}(MD) = \begin{cases} 0 & (MD < 2.70) \text{ ó } (MD > 3.30) \\ (MD - 2.70)/(3.16 - 2.70) & 2.70 \leq MD < 3.16 \\ 1 & 3.16 \leq MD \leq 3.21 \\ (3.30 - MD)/(3.30 - 3.21) & 3.21 < MD \leq 3.30 \end{cases}$$

$$\mu_{centrado}(MD) = \begin{cases} 0 & (MD < 3.28) \text{ ó } (MD > 3.55) \\ (MD - 3.28)/(3.40 - 3.28) & 3.28 \leq MD < 3.40 \\ 1 & 3.40 \leq MD \leq 3.45 \\ (3.55 - MD)/(3.55 - 3.45) & 3.45 < MD \leq 3.55 \end{cases}$$

$$\mu_{cerrado}(MD) = \begin{cases} 0 & MD < 3.53 \\ 1 & MD > 4.0 \\ (MD - 3.53)/(4.0 - 3.53) & 3.53 \leq MD \leq 4.0 \end{cases}$$

Las funciones de pertenencia altas y bajas trapezoidal para el motor izquierdo se muestran en la Fig. 5.12.

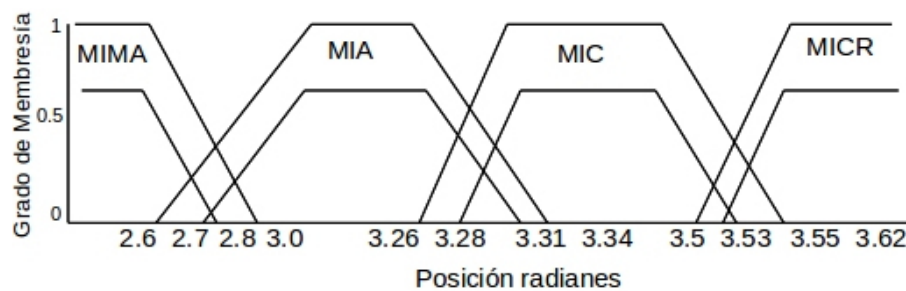


FIGURA 5.13: Funciones de pertenencia altas y bajas, para sensor y motor derecho.

$$\mu_{Muy_abierto}(\overline{MI}) = \begin{cases} 0 & MI < 1.98 \\ 1 & MI > 2.2 \\ (1.98 - MI)/(1.98 - 2.2) & 1.98 \leq MI \leq 2.2 \end{cases}$$

$$\mu_{Abierto}(\overline{MI}) = \begin{cases} 0 & (MI > 2.46) \text{ ó } (MI > 1.91) \\ (MI - 2.46)/(1.99 - 2.46) & 1.99 \leq MI \leq 2.46 \\ 1 & 1.94 \leq MI \leq 1.99 \\ (1.91 - MI)/(1.91 - 1.94) & 1.91 \leq MI < 1.94 \end{cases}$$

$$\mu_{centrado}(\overline{MI}) = \begin{cases} 0 & (MI > 1.95) \text{ ó } (MI < 1.70) \\ (MI - 1.95)/(1.93 - 1.95) & 1.93 < MI \leq 1.95 \\ 1 & 1.87 \leq MI \leq 1.93 \\ (1.70 - MI)/(1.70 - 1.87) & 1.70 \leq MI \leq 1.87 \end{cases}$$

$$\mu_{cerrado}(\overline{MI}) = \begin{cases} 0 & MI > 1.8 \\ 1 & MI < 1.7 \\ (MI - 1.8)/(1.7 - 1.8) & 1.7 \leq MI \leq 1.8 \end{cases}$$

La evaluación por intervalos para el motor Izquierdo parte baja se muestra a continuación,

$$\mu_{Muy_abierto}(MI) = \begin{cases} 0 & MI < 1.99 \\ 1 & MI > 2.1 \\ (1.99 - MI)/(1.99 - 2.1) & 1.99 \leq MI \leq 2.1 \end{cases}$$

$$\mu_{Abierto}(MI) = \begin{cases} 0 & (MI > 2.0) \text{ ó } (MI > 1.93) \\ (MI - 2.0)/(1.98 - 2.0) & 1.98 \leq MI \leq 2.0 \\ 1 & 1.95 \leq MI \leq 1.98 \\ (1.93 - MI)/(1.93 - 1.95) & 1.93 \leq MI < 1.95 \end{cases}$$

$$\mu_{centrado}(MI) = \begin{cases} 0 & (MI > 1.94) \text{ ó } (MI < 1.8) \\ (MI - 1.94)/(1.91 - 1.94) & 1.91 < MI \leq 1.94 \\ 1 & 1.88 \leq MI \leq 1.91 \\ (1.8 - MI)/(1.8 - 1.88) & 1.8 \leq MI \leq 1.88 \end{cases}$$

$$\mu_{cerrado}(MI) = \begin{cases} 0 & MI > 1.86 \\ 1 & MI < 1.6 \\ (MI - 1.86)/(1.6 - 1.86) & 1.6 \leq MI \leq 1.86 \end{cases}$$

El motor de inferencia combina las reglas disparadas y hace un mapeo de los conjuntos de entrada difuso tipo-II a un conjunto de salida difuso tipo-II. En este motor de inferencia, los antecedentes en las reglas se conectan utilizando la operación *Meet*, los

grados de membresía en los conjuntos de entrada se combinan con las de los conjuntos de resultados usando la composición extendida sup-star y múltiples reglas se combinan usando la operación *Join*. Similar al sistema difuso de tipo-I, la fuerza de disparo se puede obtener mediante el siguiente proceso de inferencia:

$$F^i = [\underline{f}^i, \bar{f}^i] \quad (5.9)$$

donde \underline{f}^i y \bar{f}^i puede reescribirse como:

$$\bar{f}^i = \bar{\mu}_{\bar{F}_{SFD}}(SFD) * \bar{\mu}_{\bar{F}_{SFI}}(SFI) \quad (5.10)$$

$$\underline{f}^i = \underline{\mu}_{\underline{F}_{SFD}}(SFD) * \underline{\mu}_{\underline{F}_{SFI}}(SFI) \quad (5.11)$$

donde $\bar{\mu}$ y $\underline{\mu}$ indica el grado de las funciones pertenencia superior e inferior respectivamente. El símbolo * es el operador t-norm. En nuestro controlador la operación *meet* se lleva acabo utilizando un operador mínimo t-norm, el cual para nuestro diseño de muestra en la Fig. 5.14.

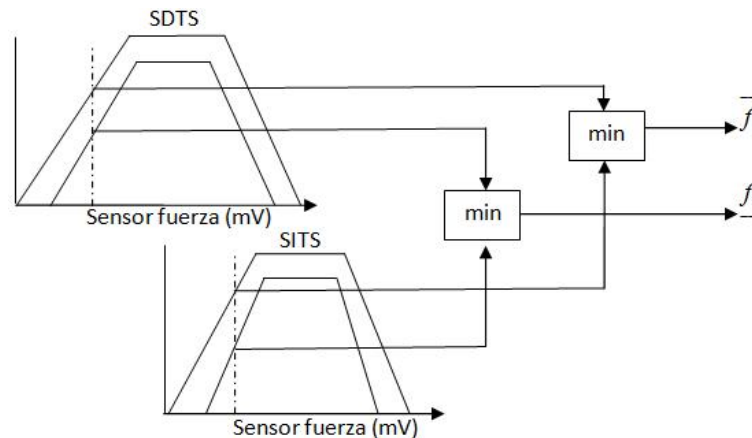


FIGURA 5.14: Operacion meet usando el operador mínimo t-norm en nuestro controlador difuso tipo II.

Se tomaron las mismas 50 reglas que el controlador difuso tipo-I, pero ahora con los valores para función de pertenencia alta y pertenencia baja, es decir sus intervalos. En la tabla 5.3 se muestran solo las primeras 16 reglas de las 50 reglas empleadas para controlador difuso tipo-II.

Posteriormente se tiene que obtener un valor nítido, igual como el difuso tipo-I, pero antes se debe de aplicar un tipo de reductor. Existen diversos métodos para reducirlo por ejemplo: centroide, centro de conjuntos, altura modificada para realizar el proceso

de reducción. Como se mencionó en la sección de 3.5.1 del capítulo 3, el método de reducción será el KM que ocupa el centro de conjuntos. De acuerdo al algoritmo tenemos que y_r^i y y_l^i instanciados a nuestro controlador es \underline{y}_I^i y \bar{y}_I^i respectivamente. Estos representan el intervalo de salida, que de acuerdo con la ec. 3.14 se obtiene el comando de salida del motor izquierdo.

Este mismo procedimiento se emplea para el motor derecho, es decir, que y_r^i y y_l^i se instancian con \underline{y}_D^i y \bar{y}_D^i respectivamente y representan el intervalo de salida, que de acuerdo con la ec. 3.14 se obtiene el comando de salida del motor derecho.

En este capítulo se explicó el diseño y la arquitectura de cada uno de los controladores. Para el controlador híbrido se mostró el diagrama de control que conmuta de posición a fuerza y viceversa. En los controladores difusos se detalló el procedimiento para diseñar el conjunto de reglas a partir de los casos propuestos para tomar el objeto. Las funciones de pertenencia de tipo trapezoidal son empleadas en ambos controladores difusos. En el siguiente capítulo se mostrará la implementación de los controladores en los experimentos propuestos.

$F^1 = [\underline{f}^1, \bar{f}^1] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.Muy.abierto \text{ Y } MD.Muy.abierto]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.Muy.abierto \text{ Y } MD.Muy.abierto]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^1, \bar{y}_I^1]=[0.4, 0.8] \\ [\underline{y}_D^1, \bar{y}_D^1]=[0.5, 0.9] \end{matrix}$
$F^2 = [\underline{f}^2, \bar{f}^2] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.Muy.abierto]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.Muy.abierto]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^2, \bar{y}_I^2]=[0.4, 0.8] \\ [\underline{y}_D^2, \bar{y}_D^2]=[0.5, 0.9] \end{matrix}$
$F^3 = [\underline{f}^3, \bar{f}^3] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MI.centrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MI.centrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^3, \bar{y}_I^3]=[0.4, 0.8] \\ [\underline{y}_D^3, \bar{y}_D^3]=[0.5, 0.9] \end{matrix}$
$F^4 = [\underline{f}^4, \bar{f}^4] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^4, \bar{y}_I^4]=[0.0, 0.2] \\ [\underline{y}_D^4, \bar{y}_D^4]=[0.0, 0.3] \end{matrix}$
$F^5 = [\underline{f}^5, \bar{f}^5] =$	$\frac{[SI, Apagado \text{ Y } SDT.Suave \text{ Y } MI.abierto \text{ Y } MD.abierto]}{[SI, Apagado \text{ Y } SDT.Suave \text{ Y } MI.abierto \text{ Y } MD.abierto]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^5, \bar{y}_I^5]=[0.4, 0.8] \\ [\underline{y}_D^5, \bar{y}_D^5]=[2.0, 2.8] \end{matrix}$
$F^6 = [\underline{f}^6, \bar{f}^6] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.centrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.centrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^6, \bar{y}_I^6]=[0.4, 0.8] \\ [\underline{y}_D^6, \bar{y}_D^6]=[1.1, 1.8] \end{matrix}$
$F^7 = [\underline{f}^7, \bar{f}^7] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^7, \bar{y}_I^7]=[0.4, 0.8] \\ [\underline{y}_D^7, \bar{y}_D^7]=[1.1, 1.9] \end{matrix}$
$F^8 = [\underline{f}^8, \bar{f}^8] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.centrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.centrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^8, \bar{y}_I^8]=[0.4, 0.8] \\ [\underline{y}_D^8, \bar{y}_D^8]=[2.0, 2.7] \end{matrix}$
$F^9 = [\underline{f}^9, \bar{f}^9] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.cerrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.cerrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^9, \bar{y}_I^9]=[0.4, 0.8] \\ [\underline{y}_D^9, \bar{y}_D^9]=[2.0, 2.8] \end{matrix}$
$F^{10} = [\underline{f}^{10}, \bar{f}^{10}] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.abierto]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.abierto]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^{10}, \bar{y}_I^{10}]=[0.4, 0.8] \\ [\underline{y}_D^{10}, \bar{y}_D^{10}]=[2.1, 2.8] \end{matrix}$
$F^{11} = [\underline{f}^{11}, \bar{f}^{11}] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.cerrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.cerrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^{11}, \bar{y}_I^{11}]=[0.4, 0.8] \\ [\underline{y}_D^{11}, \bar{y}_D^{11}]=[2.0, 2.8] \end{matrix}$
$F^{12} = [\underline{f}^{12}, \bar{f}^{12}] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.abierto]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.abierto]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^{12}, \bar{y}_I^{12}]=[0.4, 0.8] \\ [\underline{y}_D^{12}, \bar{y}_D^{12}]=[2.0, 2.8] \end{matrix}$
$F^{13} = [\underline{f}^{13}, \bar{f}^{13}] =$	$\frac{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}{[SI, Apagado \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^{13}, \bar{y}_I^{13}]=[0.4, 0.8] \\ [\underline{y}_D^{13}, \bar{y}_D^{13}]=[2.1, 2.9] \end{matrix}$
$F^{14} = [\underline{f}^{14}, \bar{f}^{14}] =$	$\frac{[SI, T.Suave \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.abierto]}{[SI, T.Suave \text{ Y } SDApagado \text{ Y } MI.abierto \text{ Y } MD.abierto]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^{14}, \bar{y}_I^{14}]=[0.4, 0.8] \\ [\underline{y}_D^{14}, \bar{y}_D^{14}]=[2.0, 2.8] \end{matrix}$
$F^{15} = [\underline{f}^{15}, \bar{f}^{15}] =$	$\frac{[SI, T.Suave \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.centrado]}{[SI, T.Suave \text{ Y } SDApagado \text{ Y } MI.centrado \text{ Y } MD.centrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^{15}, \bar{y}_I^{15}]=[0.4, 0.8] \\ [\underline{y}_D^{15}, \bar{y}_D^{15}]=[2.0, 2.8] \end{matrix}$
$F^{16} = [\underline{f}^{16}, \bar{f}^{16}] =$	$\frac{[SI, T.Suave \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}{[SI, T.Suave \text{ Y } SDApagado \text{ Y } MI.cerrado \text{ Y } MD.cerrado]}$	\rightarrow	$\begin{matrix} [\underline{y}_I^{16}, \bar{y}_I^{16}]=[0.4, 0.8] \\ [\underline{y}_D^{16}, \bar{y}_D^{16}]=[2.1, 2.9] \end{matrix}$

CUADRO 5.3: Ejemplo de reglas para el controlador difuso Tipo-II.

Capítulo 6

Experimentos y Resultados

En este capítulo se describirán los experimentos que se realizaron para mostrar el desempeño de los controladores implementados en la pinza robótica.

Se realizaron diversos experimentos de manipulación a continuación se reportan dos de ellos cada experimento propuesto se ejecuto varias veces.

El primer experimento consistió en dirigir únicamente la pinza hacia el objeto, centrarlo en medio de la pinza y levantarlo de la mesa como se observa en la Fig. 6.1. El objetivo de este experimento es poder centrar el objeto en medio de la pinza. Esto permite tomarlo correctamente, ya que de lo contrario el objeto puede resbalarse y caerse al momento de levantarlo. Los tres controladores se probaron con este experimento, primero se explica el comportamiento del controlador híbrido y posteriormente los controladores difusos.

En el segundo experimento nuevamente los tres controladores son verificados y consistió en dirigir la pinza hacia el objeto pero ahora ocupando el brazo robótico montado sobre una base móvil, con el objetivo de manipular el objeto y llevarlo a otra posición. El propósito de este experimento es observar, si las perturbaciones ocurridas en el traslado del objeto afectan el desempeño de los controladores. A continuación se detallará cada uno de los experimentos, así como se discutirán los resultados obtenidos.

6.1. Experimento 1

Para este experimento el control híbrido de posición-fuerza aplicado sobre al objeto, se tomaron dos criterios de evaluación, la señal del sobre tiro y el tiempo de establecimiento, al tomar y mover un objeto. La secuencia del experimento es la siguiente:

- Tomar el objeto con la pinza el cual debe estar alineado en el centro de la pinza, con el fin de que la fuerza aplicada al objeto sea uniforme y el riesgo de dañarlo o soltarlo sea mínimo. Para esta parte se utiliza el controlador de posición.
- Mover el objeto con la fuerza que aplica la pinza al objeto, sea la suficiente para que no se resbale, el controlador de fuerza es usado.

6.1.1. Controlador Híbrido

Este controlador híbrido posición-fuerza es ejecutado en ROS en un nodo llamado controlador, como se había mencionado en la sección 4.3. El controlador híbrido PID de posición fue sintonizado por el método de prueba y error. Los parámetros de las ganancias quedaron con los siguientes valores, $K_p = 0.000001$, $K_i = 0.0001$ and $K_d = 0.001$. Para esta parte del experimento el objeto es colocado en frente de la pinza, éste lo toma y lo centra en medio de la pinza (Fig. 6.1).

De acuerdo con el diagrama de la Fig. 5.1 sección 5.1, una vez que el objeto se coloca en frente de la pinza, el PID de posición se activa para centrar el objeto y el PID de fuerza permanece apagado. En la Fig. 6.2, en la parte superior se muestran las lecturas del sensor de posición que se usan para ajustar el PID de posición durante 4.19 segundos (el PID de posición está encendido). La tabla 6.1 muestra los resultados del controlador PID. El tiempo de subida es largo, por que el objeto está lejos del centro de la pinza. En la Fig 6.1, el centro está marcado sobre la mesa con una línea negra. En este PID de posición, el tiempo de establecimiento es más corto que el PID de fuerza, esto es por que cuando los dedos de la pinza están cercanos al objeto, el objeto no cambia su posición.

El PID de fuerza también fue sintonizado de la misma manera que el PID de posición y los parámetros de las ganancias son los siguientes: $K_p = 0.5$, $K_i = 0.005$ y $K_d = 0.001$.

Para esta parte del experimento se muestra en la Fig. 5.1 en la segunda fila como se levanta el objeto de la mesa. El controlador híbrido PID de fuerza se ejecuta para que la fuerza que aplica sobre el objeto sea la suficiente para no soltarlo cuando se levanta de la mesa. Una vez que el objeto está centrado, el PID de posición se apaga y el PID de fuerza se enciende. En la gráfica 6.2 en la parte inferior se muestran las lecturas del sensor de fuerza que se utilizan para ajustar el PID de fuerza después de 4.19s (PID de posición desactivado). En la Fig. 6.3 se muestran las señales de error y salida del PID de fuerza después de 4.19s. En la tabla se muestran los resultados del controlador PID en el lazo de control. El tiempo de subida es corto ya que los dedos están cerca del objeto, entonces de manera casi instantánea los dedos aplican la suficiente fuerza al objeto para manipularlo. El tiempo de asentamiento es grande, ya que la pinza esta levantando al objeto y por lo tanto la fuerza aplicada sobre el es ajustada constantemente.

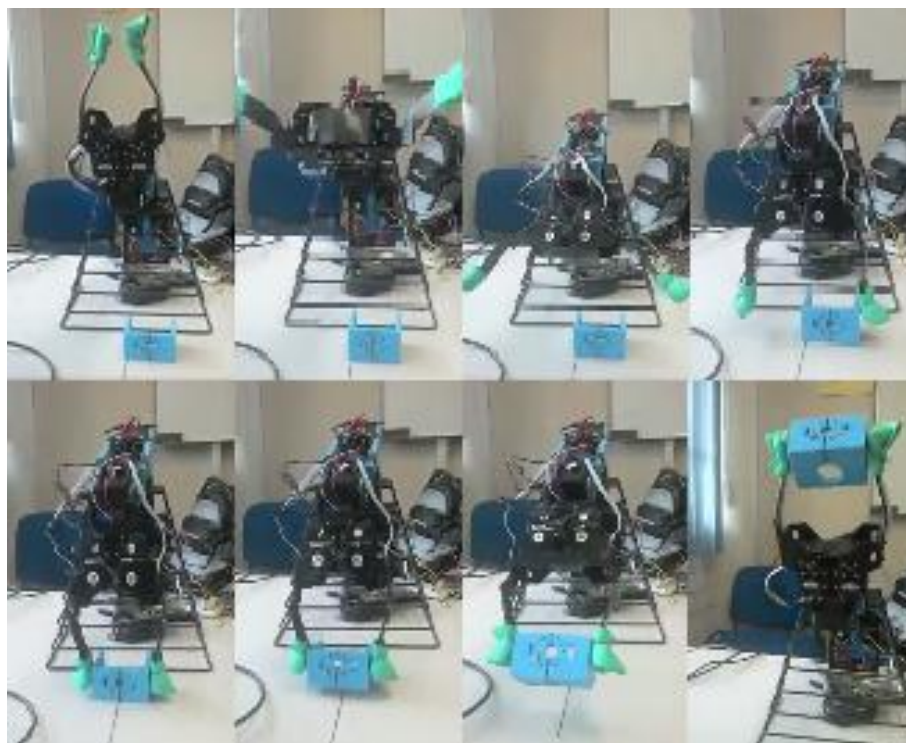


FIGURA 6.1: En la primera fila se muestra cómo el objeto está centrado en el medio de la pinza. En la segunda fila se muestra cómo se levanta el objeto de la mesa.

CUADRO 6.1: Comparación de los parámetros de desempeño

PID	Posición	Fuerza
Tiempo de levantamiento (sec)	4.08	0.61
Valor pico	3.471	53.5
Tiempo de asentamiento	7	12.81
Valor final	3.47	41.25
Sobre tiro	0.001	7.25

6.1.2. Controladores Difusos Tipo-I y Tipo-II

Como se mencionó anteriormente, este experimento también fue implementado con los controladores difusos Tipo-I y Tipo-II. Primero se muestra en la Fig. 6.4 la posición de los motores derecho e izquierdo. Los valores de referencia son los mismos 3.5 radianes para el motor Derecho y 1.8 radianes para el motor Izquierdo. Se observa en las gráficas como durante el tiempo de 0s hasta 2s. aproximadamente mantienen su velocidad constante hasta que los dedos hacen contacto con el objeto. En el tiempo de 3s los controladores llegan a la posición deseada y se mantienen estables en esta posición incluso aun cuando se levanta el objeto en el tiempo de 5s.

En la Fig. 6.6 se muestran lecturas de los sensores de fuerza derecho e izquierdo. En los dos primeros segundos se observa que ambos sensores tienen una lectura de cero y esto se debe a que la pinza apenas se está cerrando y aún no detectan el objeto. Después

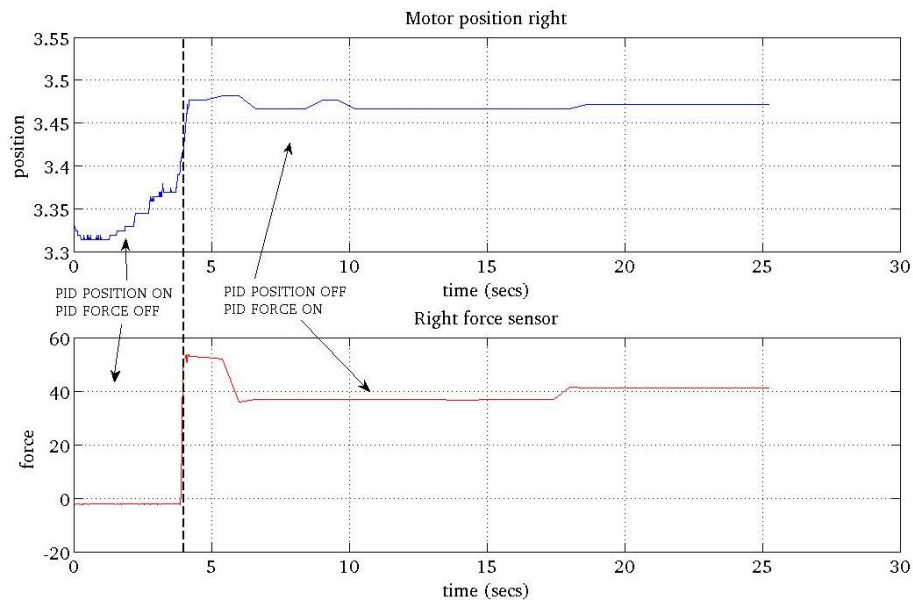


FIGURA 6.2: En la gráfica de arriba se muestran las lecturas del sensor de posición que se usan para ajustar el PID de posición. En la gráfica de abajo se muestran las lecturas del sensor de fuerza que se utilizan para ajustar el PID de fuerza.

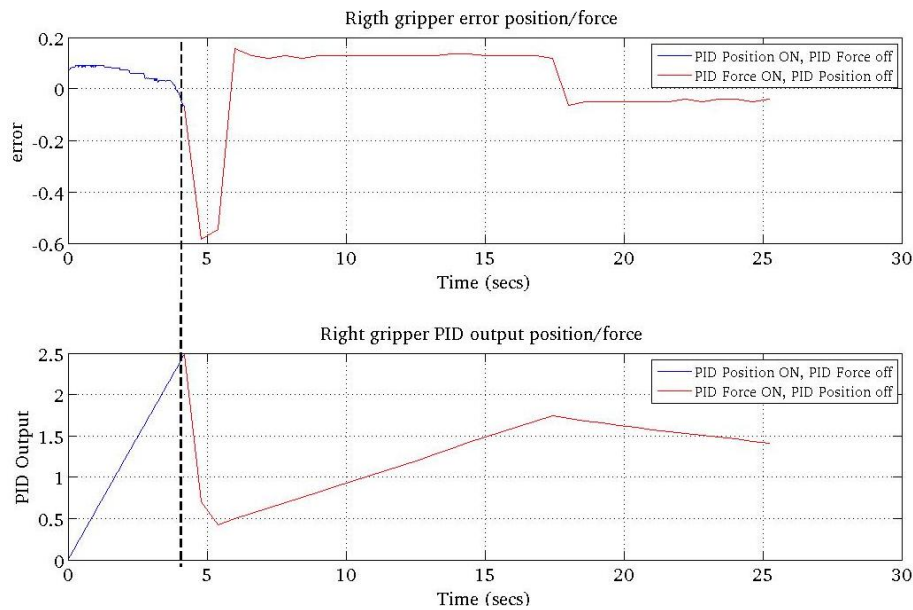


FIGURA 6.3: En la gráfica de arriba se muestra el error de la posición y fuerza del PID. En el siguiente gráfica se muestra la salida de la posición PID y la fuerza.

del tiempo de 2s. se puede apreciar que ambos controladores tratan de llegar al valor de referencia el cual era de $60mV$. La lectura del sensor de fuerza derecho muestra que el controlador difuso Tipo-I queda por debajo de este valor. Por otra parte el controlador difuso Tipo-II está cercano a $50mV$, aun debajo del valor de referencia, pero mejor que el difuso Tipo-I. En el tiempo de 5s. se levanta el objeto de la mesa, se observan como existen diversas variaciones en las lecturas del sensor izquierdo.

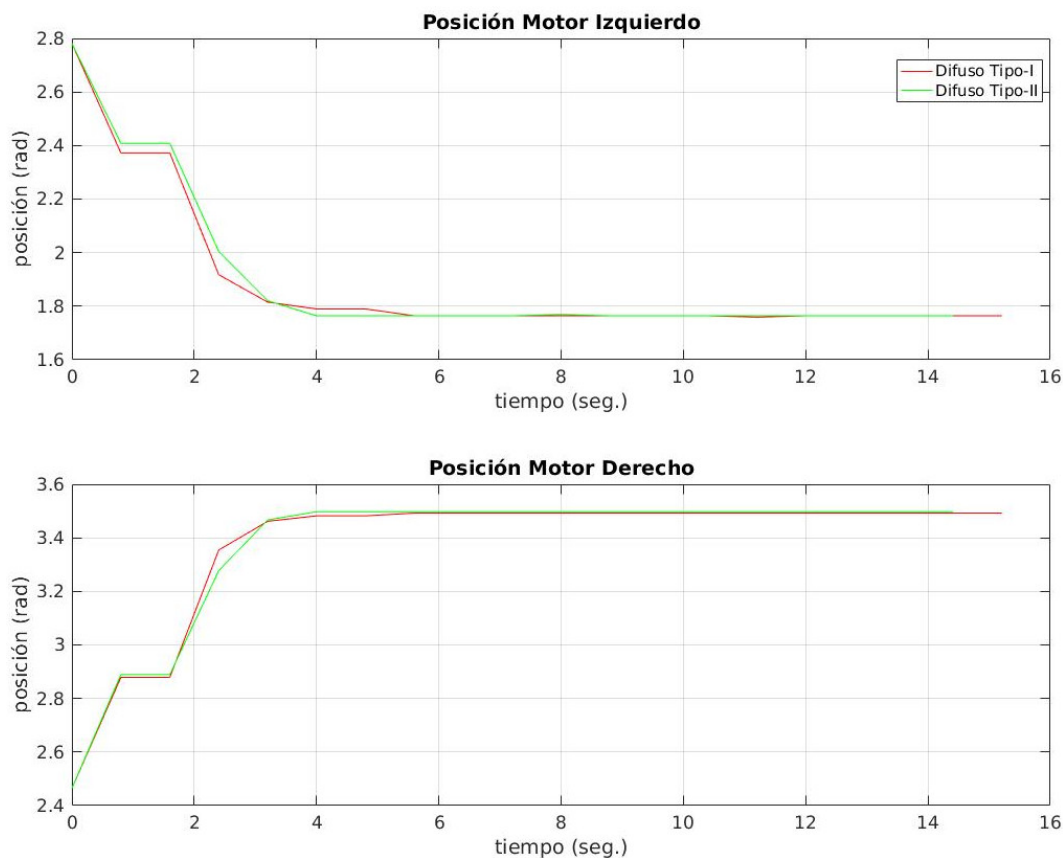


FIGURA 6.4: Posiciones de los motores durante el experimento. En la gráfica de arriba se muestran las lecturas de posición del motor izquierdo. En la parte de abajo se muestran las posiciones del motor derecho.

CUADRO 6.2: Comparación de los parámetros de desempeño del controlador Difuso Tipo-I para Motor y Sensor Derecho

Difuso Tipo-I	Controlador Posición Derecho	Controlador Fuerza Derecho
Tiempo de levantamiento (sec)	4	3.5
Valor pico	3.492	46.75
Tiempo de asentamiento	8	12
Valor final	3.492	43.75
Sobre tiro	0	0.25

CUADRO 6.3: Comparación de los parámetros de desempeño del controlador Difuso Tipo-I para Motor y Sensor Izquierdo

Difuso Tipo-I	Controlador Posición Izquierdo	Controlador Fuerza Izquierdo
Tiempo de levantamiento (sec)	3.8	3.5
Valor pico	1.78	56.0
Tiempo de asentamiento	8	12
Valor final	1.76	33.75
Sobre tiro	0.02	22.25

Finalmente se muestran las salidas de los dos controladores difusos Tipo-I y Tipo-II. con el comando de velocidad que se publica en ROS.

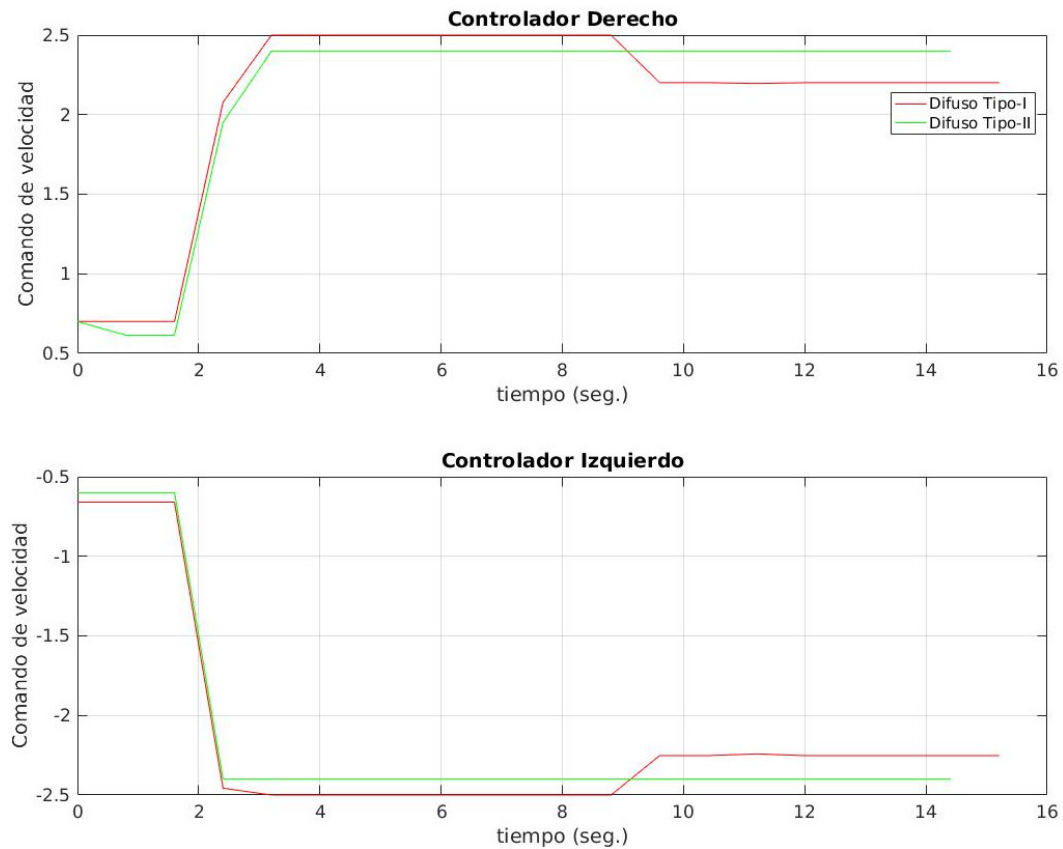


FIGURA 6.5: Salidas de los controladores difusos Tipo-I y Tipo-II. Estas salidas son comandos de velocidad que se envían a los motores. En la parte de arriba se muestran las salidas de los controladores para el motor derecho. En la parte de abajo se observan las salidas de los controladores para el motor izquierdo

CUADRO 6.4: Comparación de los parámetros de desempeño del controlador Difuso Tipo-II para Motor y Sensor Derecho

Difuso Tipo-II	Controlador Posición Derecho	Controlador Fuerza Derecho
Tiempo de levantamiento (sec)	4	3.5
Valor pico	3.497	49.25
Tiempo de asentamiento	8	12
Valor final	3.497	46.75
Sobre tiro	0	2.25

CUADRO 6.5: Comparación de los parámetros de desempeño del controlador Difuso Tipo-II para Motor y Sensor Izquierdo

Difuso Tipo-II	Controlador Posición Izquierdo	Controlador Fuerza Izquierdo
Tiempo de levantamiento (sec)	3.8	3.5
Valor pico	1.8	55.25
Tiempo de asentamiento	8	8
Valor final	1.76	47.5
Sobre tiro	0	7

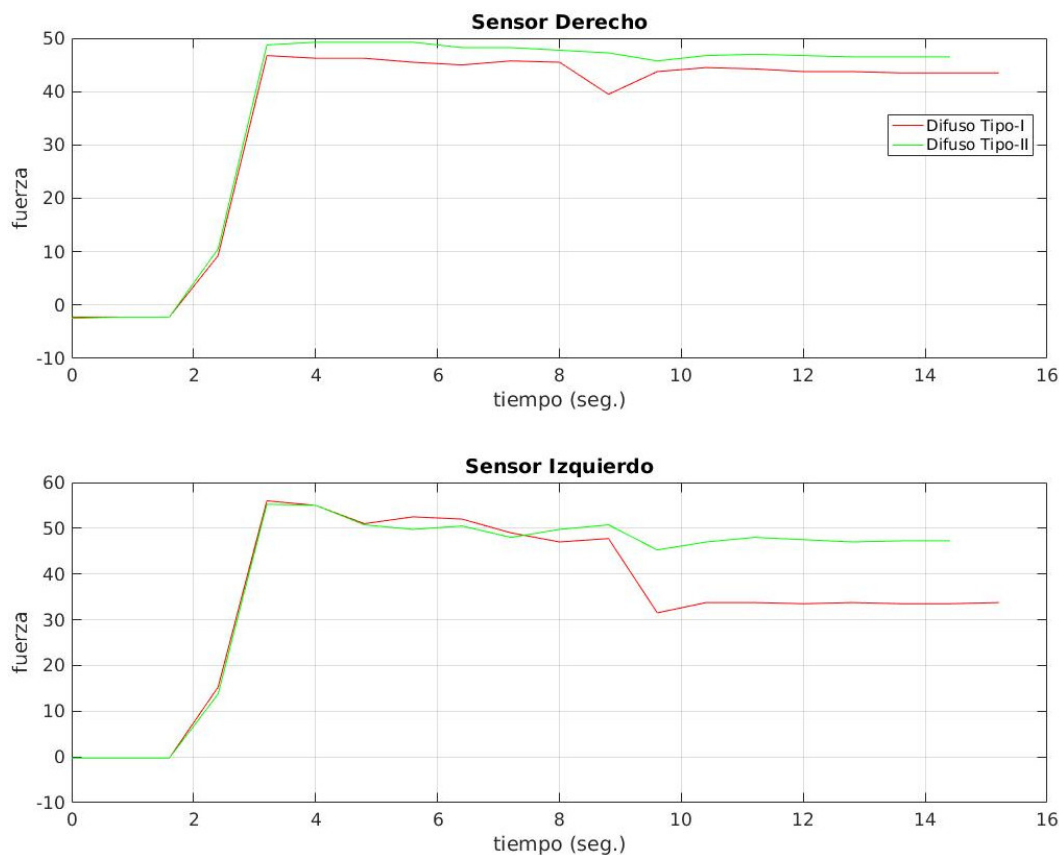


FIGURA 6.6: Lecturas de los sensores, en la parte de arriba se muestran las lecturas del sensor derecho, en la parte de abajo las lecturas del sensor izquierdo.

Se podría concluir de este experimento que de los tres controladores el controlador difuso Tipo II, tuvo un mejor comportamiento. Para este controlador las lecturas del sensor de fuerza derecho presenta menos oscilaciones que el izquierdo. Este se debe a la posible forma en que tomo al objeto, ya que en el tiempo de 9s se observa como cae la lectura del sensor.

6.2. Experimento 2.

Para este experimento los tres controladores (híbrido, difuso Tipo-I, Difuso-Tipo-II) propuestos son ejecutados nuevamente en un nodo de ROS y además la pinza, se monta con el brazo sobre una base móvil (Fig. 6.7). El objetivo es evaluar el desempeño de cada uno de los controladores sobre la pinza al manipular objetos y el experimento que consiste en manipular un objeto, levantarlo y dejarlo en otra ubicación. Este experimento se divide en tres etapas:

Etapas 1: Se coloca la pinza en posición cercana al objeto, asumiendo que la posición del objeto se conoce. La pinza empieza a cerrar hasta hacer contacto con el objeto y

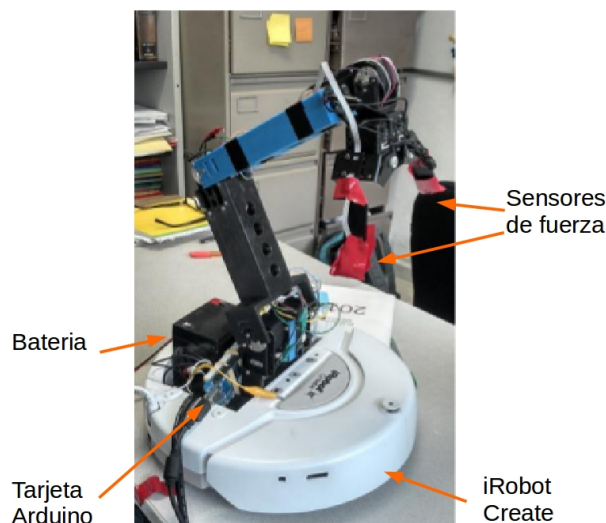


FIGURA 6.7: Brazo de tres grados de libertad, colocado en un robot móvil (iRobot Create). Sobre la parte posterior del robot se coloca una placa de adquisición de datos, así como una batería de 12V para alimentar el sistema.

centrarlo. Ésta etapa dura 5.6s para los tres controladores (PID posición, tipo I y tipo II) con el objetivo de evaluar cómo se comporta la salida del controlador, (Fig. 6.8 a-c).

Etapa 2: Una vez centrado el objeto en la pinza, éste se levanta con la suficiente fuerza para no soltarlo (PID fuerza, tipo I y tipo II), y la base del robot empieza a girar durante 4.6s el cual es el tiempo suficiente para que la base gire aproximadamente 180° , (Fig. 6.8 d-h).

Etapa 3: Una vez que la base del robot se detuvo, la pinza baja junto con el objeto, hasta que éste es depositado en la mesa, (Fig. 6.8 i-j).

Estas etapas se pueden apreciar en las Figuras 6.9 y 6.10. Empezando con la parte de posición en la Fig. 6.9 en los tres controladores la referencia en posición para este objeto es 1.8 radianes del dedo izquierdo, y para el dedo derecho 3.4 radianes. En la Fig.6.9 se observa en la parte de arriba la posición del motor izquierdo. Ésta empieza desde la posición de apertura es decir de 2.8 radianes hasta ir cerrando y por lo tanto se acerca la posición de referencia de 1.8 radianes. Los tres controladores parten de la misma posición de inicio y hasta que tocan al objeto para centrarlo tardan 2.7 segundos aproximadamente, este tiempo es similar para los tres controladores. Para el caso del controlador Híbrido desde el inicio actúa el controlador de posición. Una vez que llegan a la posición de referencia los controladores difusos no presentan oscilaciones y se mantienen estables. Por otra parte el controlador Híbrido PID en la última etapa presenta fuerte oscilaciones y esto es debido a que hace el cambio de control de fuerza a posición. Cuando la condición sobre el error de posición no se satisface (ec.5.4) se realiza la conmutación al PID de posición. La posición del motor derecho también es mostrada en la parte de abajo de la Fig.6.9. El motor parte de la posición inicial de

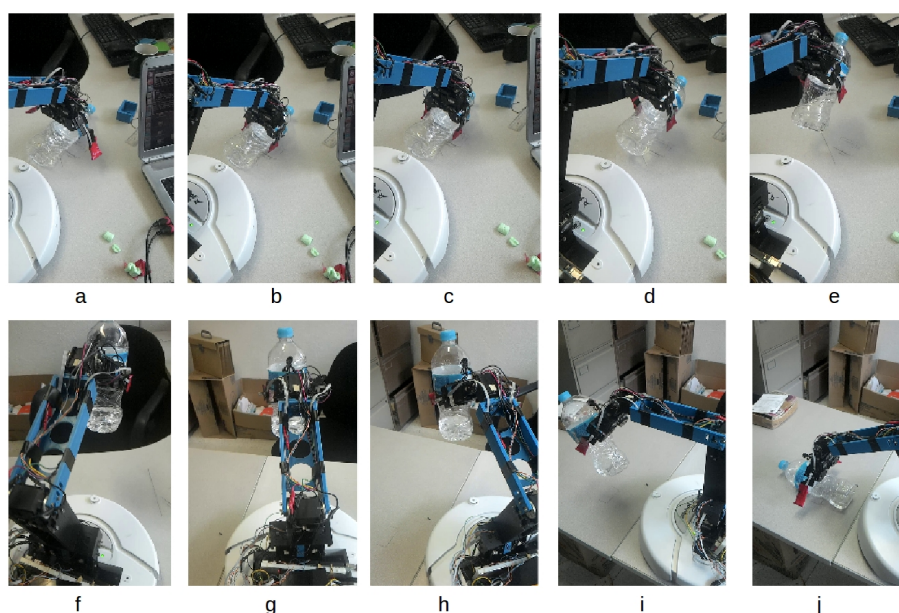


FIGURA 6.8: Secuencia del experimento de tomar un objeto. En la fila de arriba se aprecian las etapas 1 (a-c) y parte de la etapa 2 (d-h), en la fila de abajo se aprecia al final la etapa 3 (i-j).

apertura de grifer de 2.4 radianes hasta ir cerrando y llegar a la posición de referencia de 3.4 radianes. Nuevamente los controladores difusos no presentan oscilaciones en las tres etapas, mientras que el controlador híbrido PID en la etapa 3 presenta oscilaciones y esto es nuevamente debido al cambio de que hace el controlador híbrido de fuerza a posición.

La referencia en fuerza para el sensor derecho e izquierdo es de $60mV$. En la etapa 1 del controlador híbrido se observa en la gráfica del sensor izquierdo que tiene variaciones en muy poco tiempo y esto nos indica que ya está haciendo contacto con el objeto y que trata de desplazarlo hacia el centro de la pinza. Cuando la condición sobre el error de posición se satisface (ec.5.4) se realiza la conmutación al PID de fuerza (a partir de 3.67s hasta 6.073s). Ya en la etapa 2, cuando se está levantando el objeto de la mesa, se observa un sobre tiro en el sensor de fuerza izquierdo y en el sensor de fuerza derecho cae su valor y esto se debe a que vuelve a conmutar el controlador Híbrido de fuerza a posición. Quiere decir que el objeto no estaba centrado en medio de la pinza y la condición sobre el error de posición no se satisface (ec.5.4) por lo tanto se realiza la conmutación al PID de posición. En tiempo de 7.8s el controlador híbrido vuelve a activar el control de fuerza el sensor izquierdo está cerca del valor de referencia, presenta más oscilaciones y no alcanza a llegar al valor de referencia. Finalmente en la etapa 3 el objeto no se encuentra completamente en el centro de la pinza y vuelve a entrar en control de posición, es por eso que la lectura de ambos sensores de fuerza cae.

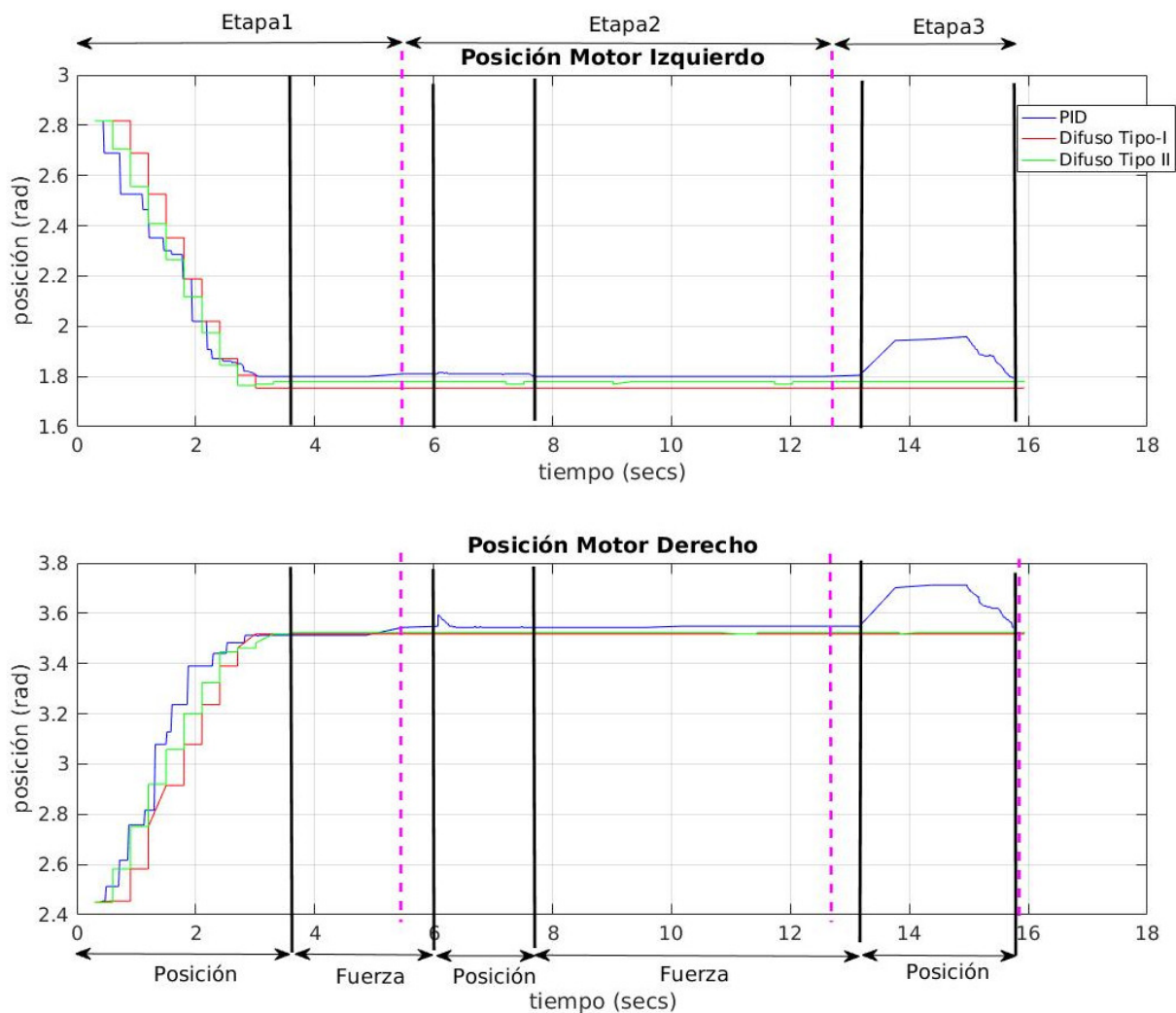


FIGURA 6.9: Posición de cada motor en las diferentes etapas del experimento.

Por otra parte para los controladores difusos, en la etapa 1 se observa que controlador difuso tipo-I en el sensor izquierdo rápidamente alcanza el valor de referencia, mientras que el controlador difuso tipo-II no lo alcanza llegando solo al valor de 50 mV. Para el sensor derecho el controlador difuso tipo-I nuevamente alcanza el valor de referencia, y el controlador difuso tipo-II queda muy abajo llegando al valor de 22.5 mV. Esto ocurre en el tiempo de 2.4 s a 5.7 s.

En la etapa 2 los controladores tipo-I y tipo-II se mantienen muy similares en el sensor de fuerza izquierdo. Mientras que en esa misma etapa pero en el sensor de fuerza derecho, solo el controlador difuso tipo-I se mantiene cerca del valor referencia y el controlador difuso tipo-II se sigue manteniendo en el rango de los 20-25 mV.

Finalmente en la etapa 3 nuevamente los controladores difusos tipo-I y tipo-II se mantienen cerca del valor de referencia en el sensor izquierdo, mientras que en el sensor derecho nuevamente el controlador difuso tipo-I se mantiene cercar de la referencia, en

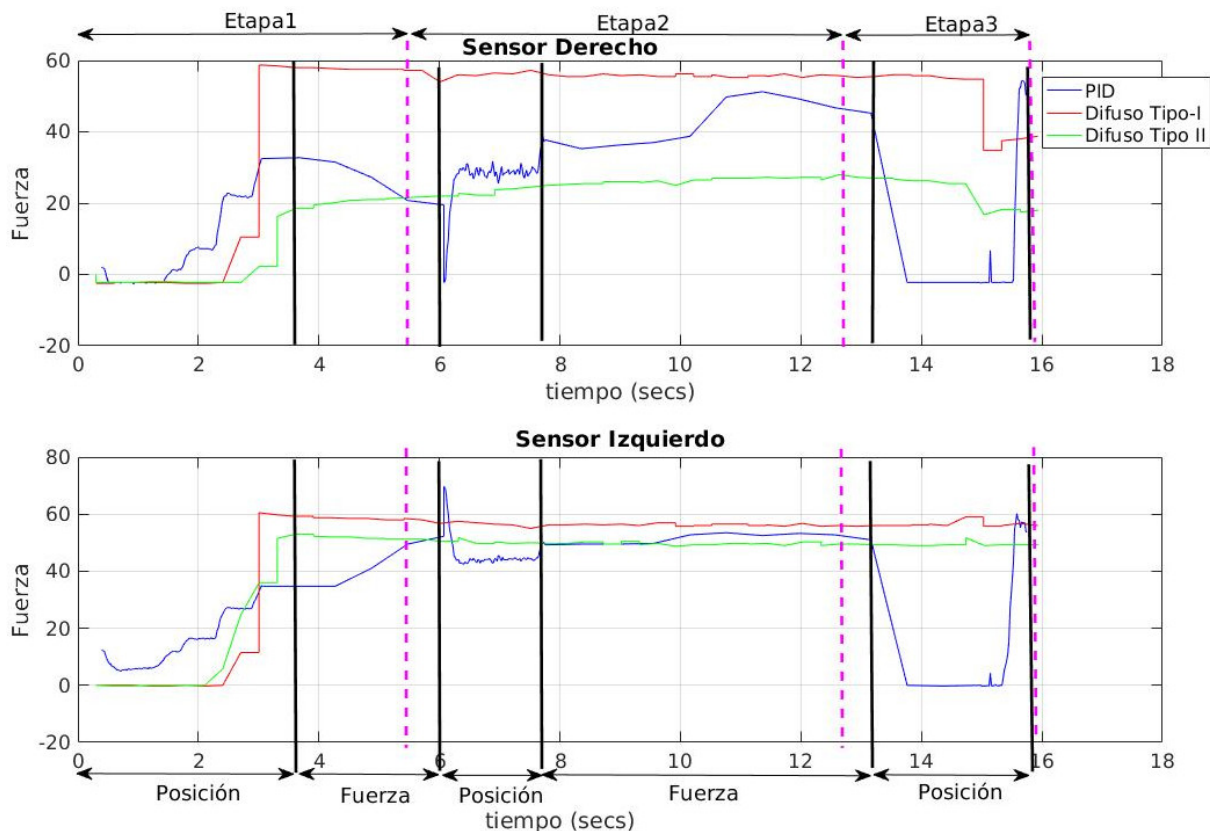


FIGURA 6.10: Desempeño de cada controlador por cada etapa. En la gráfica de arriba se muestra la respuesta del sensor de fuerza derecho. En la gráfica de abajo, se muestra la respuesta del sensor de fuerza izquierdo.

el tiempo de 15.3 s cae la lectura del sensor pero es debido al momento de dejarlo el objeto sobre la mesa.

Debemos notar que durante las tres etapas, mientras el controlador híbrido PID de posición y de fuerza están conmutando durante todo el proceso, los controladores difusos tipo-I y tipo-II trabajan de forma continua desde que toman al objeto. Claro que el difuso tipo-I lo hace mejor que el tipo-II.

En la Fig. 6.11 se muestra la salida de cada uno de los controladores en cada motor, es decir el comando de velocidad que se publica en ROS para que cada motor lo ejecute. Se observa como en la etapa 1 durante en el tiempo 0s. hasta los 2s. aproximadamente los tres controladores mantienen una velocidad constante en cada dedo hasta que hacen contacto con el objeto. Posterior a eso cada controlador incrementa el comando de velocidad hasta llegar al valor deseado en fuerza o posición según sea el caso. El controlador Híbrido de posición izquierdo incrementa su velocidad mientras que el controlador Híbrido derecho se mantiene constante, esto es por que el objeto se encuentra ligeramente desplazado a la izquierda, y este seguirá incrementando su velocidad hasta que el objeto se encuentre centrado en la pinza. Por lo tanto, en este caso cada vez que el objeto se

desplaza más hacia la izquierda, el control de posición izquierdo incrementara su velocidad para llevarlo a la posición del centro de la pinza. Cuando se activa el control de fuerza se observa que ambos motores incrementan su velocidad para tratar de llegar al valor deseado de referencia. Si en algún momento un dedo aplica mas fuerza que otro y esto provoca que el objeto se desplace más hacia un lado, vuelve a entrar el control de posición y cambias las velocidades en cada motor de la pinza.

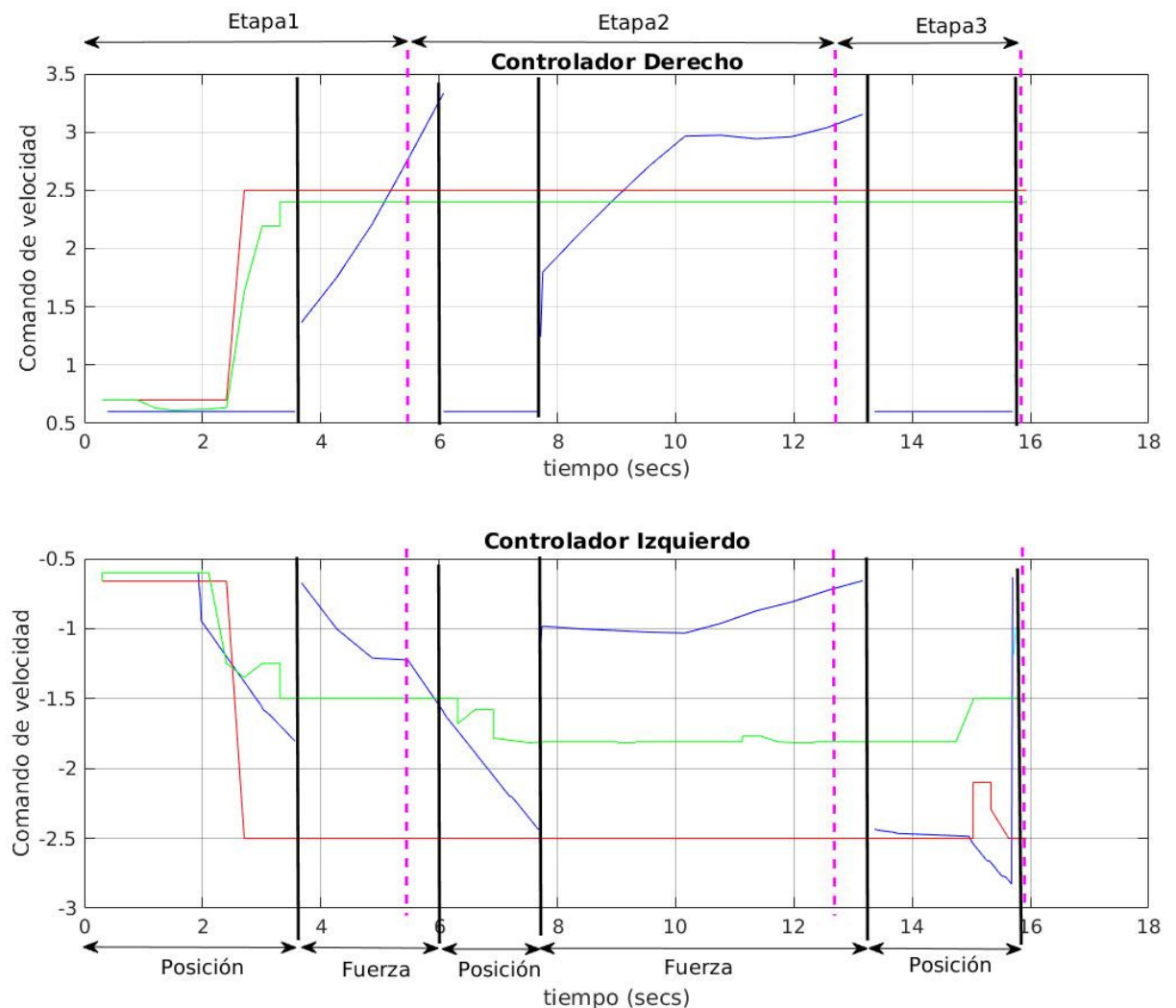


FIGURA 6.11: Salida de los controladores en comandos de velocidad. En la parte de arriba se muestra la gráfica de los comandos de velocidad del motor derecho para cada uno de los controladores. En la parte de abajo se muestra la gráfica velocidad del motor izquierdo para cada uno de los controladores

	Controlador Dedo Izquierdo		Controlador Dedo Derecho	
	IAE	ITAE	IAE	ITAE
Híbrido PID	0.08330776	0.3820714	0.33537232	1.116447603
Difuso Tipo-I	0.07411874	0.496096701	0.0670152	1.05220179
Difuso Tipo-II	0.09851675	0.22070650	0.01394075	1.1569658

CUADRO 6.6: Mediciones de peso y voltaje para el sensor de fuerza.

Controlador	Sensor de fuerza Izquierdo		Sensor de fuerza Derecho	
	IAE	ITAE	IAE	ITAE
Híbrido PID	0.860155	0.557759	0.205725	0.109171
Difuso Tipo-I	8.82	84.953571	5.98	47.39870
Difuso Tipo-II	51.1325	457.0258	15.2875	134.1986

CUADRO 6.7: Mediciones de peso y voltaje para el sensor de fuerza.

En general, el controlador que tuvo mejor desempeño fue el difuso tipo I ya que fue más fácil de diseñar e implementar y aunque la cantidad de reglas es grande fue más fácil que el difuso tipo II. El inconveniente del controlador difuso tipo II es el tiempo de procesamiento de todas las reglas superiores e inferiores, aunque éste fue el menos sensible a los cambios en las mediciones. El problema principal con el PID híbrido (posición - fuerza) es lo sensible que es ante cambios o perturbaciones durante la manipulación del objeto, además que el error en estado estable con los dos PID fue más grande que el que presentaron los difusos tipo I y II.

Capítulo 7

Conclusiones y Trabajo Futuro

El problema de manipulación robótica es un tema complejo y en este trabajo se presento un esquema de control basado en lógica difusa y control clásico.

Para que un robot pueda manipular objetos sin dañarlos y de manera autónoma, en este trabajo se presentó el desarrollo e implementación de tres diferentes controladores (PID híbrido, difuso tipo I y difuso tipo II) para la manipulación de objetos con una pinza angular de dos dedos independientes. El objetivo era lograr centrar objetos en medio de la pinza y posteriormente aplicar la suficiente fuerza para levantarlo.

Uno de los objetivos era caracterizar un sensor de fuerza que permitiera a los controladores ajustar la fuerza aplicada al objeto sin dañar los motores de la pinza. El sensor de fuerza piezoresistivo dio un buen resultado al detectar pequeñas variaciones de peso durante la caracterización de este.

Por otra parte el análisis cinemático de la pinza permitió establecer una serie de ecuaciones para obtener las configuraciones posibles de manipulación de un objeto.

Otro de los objetivos era que la pinza debía ejecutarse en ROS, por lo tanto se creo un nodo cada uno de los controladores y este nodo debía comunicarse con los nodos que publicaban información de los sensores de fuerza y posición de los motores.

Finalmente el desarrollo de los controladores los cuales se probaron con diversos experimentos para verificar la funcionalidad de los controladores en lazo cerrado. En el último experimento se mostró el desempeño de los 3 controladores, que consistió en 3 etapas (sujetar, depositar y liberar el objeto en otro lugar). El primer controlador que se desarrolló fue el PID Híbrido posición-fuerza. Este controlador PID híbrido (posición-fuerza) presenta oscilaciones durante la manipulación. La sintonización de las ganancias de este controlador Híbrido fueron calculadas mediante el método *prueba y error* debido

a la complejidad del sistema a controlar. En el primer experimento con este método se logró una sobreelongación pequeña a la salida, así como también un tiempo de asentamiento pequeño. Sin embargo, en el segundo experimento no tuvo un buen desempeño, esto es debido a la conmutación del controlador de posición al de fuerza y viceversa, además de que tarda en restablecerse ante las perturbaciones. Este controlador híbrido podría servir para una pinza paralela de un solo motor ya que en éste no se tiene la posibilidad de mover los dedos de manera independiente y por lo tanto tendría menos perturbaciones.

El controlador difuso tipo I tiene un buen desempeño ya que no presento grandes oscilaciones durante la manipulación y tuvo menos error en estado estable. El conjunto de reglas definido fue suficiente para lograr reducir las perturbaciones que el Híbrido no puedo eliminar. Desafortunadamente no se pudieron graficar las 4 variables de entrada (Sensor de fuerza derecho-izquierdo y motor derecho-izquierdo) en una sola gráfica de superficie para observar la salida continua del sistema. Sin embargo, se gráfico de manera independiente y se observó que la superficie de salida era continua y por tanto el controlador difuso iba a tener un desempeño aceptable.

Por su parte el controlador difuso tipo II exhibe un comportamiento similar al difuso tipo I, solo el controlador del dedo izquierdo presento mayores perturbaciones que el derecho, sin embargo el objeto se mantuvo estable en la pinza. Uno de lo inconvenientes que se observó en este controlador fue que se tuvo que aumentar el costo de computo debido a la gran cantidad de reglas. A pesar de eso tuvo un desempeño similar al difuso Tipo-I. Tal vez se podría modificar las funciones de pertenencia y revisar si se mejora el comportamiento o se reduce el número de reglas en el controlador.

De lo anterior se puede decir que la gran ventaja que ofrecen los controladores difusos es que pueden proveer una eficiente metodología para implementar de forma experimental un controlador no lineal sin la necesidad de usar herramientas matemáticas avanzadas. Para hacer los controladores difusos tipo I, y tipo II se necesitó de una descripción lingüística de la ley de control, la cual es capturada por los conjuntos difusos trapezoidales, los operadores difusos y las respectivas reglas difusas. Estos controladores no requirieron explícitamente del modelo matemático del proceso a controlar es decir la pinza. Para el caso de la pinza usado en este trabajo el, cual contaba con dedos independientes con movimiento angular el problema de manipular un objeto aplicando lo suficiente fuerza pudo ser solucionado de manera más efectiva que el control clásico.

Por último, cualquiera de los dos controladores difusos tipo-I ó tipo-II se puede utilizar para manipular un objeto sin dañarlo o soltarlo, ya que si llegan a los valores deseados de referencia. Sin embargo, implementar un difuso Tipo-I es más sencillo que uno Tipo-II.

7.0.1. Trabajo futuro

Como trabajo futuro se proponen los siguientes puntos a esta investigación:

- considerar algún otro método de sintonización de las ganancias del PID Híbrido para que éste tenga un desempeño óptimo ante las posibles perturbaciones en las etapas de manipulación.
- agregar un dedo a la pinza y aplicar el controlador difuso-Tipo-I, esto es para tener más puntos de contacto durante la manipulación.
- modificar las funciones de pertenencia para los controladores difusos, y compararlo con los trapezoidales empleados en éste trabajo. Modificando las funciones de pertenencia se podría mejorar el desempeño de éste a las perturbaciones.
- montar el brazo sobre el robot uerto.
- agregar un modulo de visión para identificar el objeto a manipular.

Apéndice A

Apéndice A

El DYNAMIXEL es un sistema de actuador inteligente desarrollado para ser las uniones de conexión exclusivas en un robot o estructura mecánica. Los DYNAMIXELS están diseñados para ser modulares y encadenados en cualquier robot o diseño mecánico para movimientos robóticos potentes y flexibles. El DYNAMIXEL es un actuador de alto rendimiento con motor de corriente continua (corriente directa) totalmente integrado + reductor de engranaje + controlador + controlador + red, todo en un actuador de servo modulo (Fig. A.1). El estado del actuador programable y conectable en red puede leerse y supervisarse a través de un flujo de paquetes de datos.



FIGURA A.1: Motores Dynamixel.

A.1. B

Servos Dynamixel

	AX-12W	AX-12A	AX-18A	RX24F	RX-28	RX-64	MX-28T MX-28R	MX-64T MX-64R	MX-106T MX-106R	EX-106
										
Peso (g)	52.9 g	54.5g	54.5g	67g	72g	125g	72g	126g	153g	154g
Dimensión(mm)	32x50x40 (mm)	32x50x40 (mm)	32x50x40 (mm)	35.6x50.6x3 5.5 (mm)	35.6x50.6x3 5.5 (mm)	40.2x61.1x4 1 (mm)	35.6x50.6x 35.5 (mm)	40.2x61.1x 41 (mm)	40.2x65.1x 46 (mm)	40.2x65. 1x46 (mm)
Proporción de engranaje	32:1	254:1	254:1	193:1	193:1	200:1	193:1	200:1	255:1	184:1
Material engranaje	Enpla	Enpla	Enpla + Metal	Metal	Metal	Metal	Metal	Metal	Metal	Metal
Red	TTL	TTL	TTL	RS-485	RS-485	RS-485	TTL/RS-485	TTL/RS-485	TTL/RS-485	RS-485
Sensor posición	Potencióme- tro	Potencióme- tro	Potencióme- tro	Potencióme- tro	Potencióme- tro	Potencióme- tro	Encoder	Encoder	Encoder	Encoder
Angulo de operación	300°	300°	300°	300°	300°	300°	360°	360°	360°	251°
Resolución (pasos)	1024	1024	1024	1024	1024	1024	4096	4096	4096	4096
Motor	Motor sin núcleo	Motor sin núcleo	Motor sin núcleo	Motor sin núcleo	Motor Maxon	Motor Maxon	Motor Maxon	Motor Maxon	Motor Maxon	Motor Maxon
Volt de operación	9-12	9-12	9-12	9-12	12-18.5	12-18.5	10-14.8	10-14.8	10-14.8	12-18.5
Toque	N/A	1.5 N.m	1.8N.m	2.6N.m	3.7N.m	5.3 N.m	3.1N.m	7.3N.m	10.0N.m	10.9 N.n
No. RPM	470 (a 12 v)	59 (a 12 v)	97(a 12 v)	126(a 12v)	85(a 18.5 v)	64(a 18.5 v)	67(a 14.8)	78(a 14.8)	55(a 14.8)	91(a 18.5)

FIGURA A.2: Carcaterísticas Motores Dynamixel.

Apéndice B

Apéndice B

Para montar la pinza sobre una estructura se diseñó e imprimió las piezas que conforman el brazo robot. Para esto se ocupó el software FreeCAD, que permite crear objetos 3D a partir de un bosquejo 2D. Se muestran a continuación cada una de las piezas que se ensamblaron con los motores del brazo.

Base. Esta pieza sujeta los dos motores que actúan como hombro a una base (mesa). Las dimensiones se muestran y están de acuerdo al ancho de los motores.

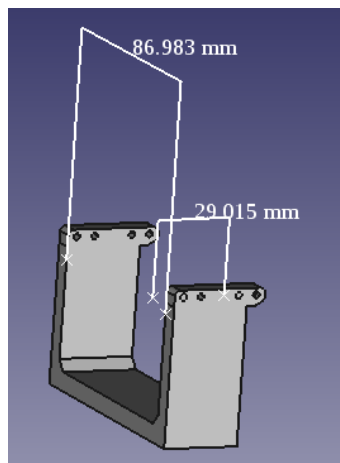


FIGURA B.1: Base

Unión-Hombro. Es la pieza que se encarga de unir los dos motores que actúan como hombro.

Brazo. Esta pieza conecta los dos motores del hombro con el motor que servirá de codo. Se diseñó lo bastante grueso para soportar el peso de las demás piezas.

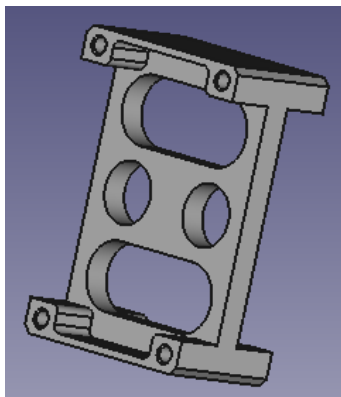


FIGURA B.2: Unión-Hombro

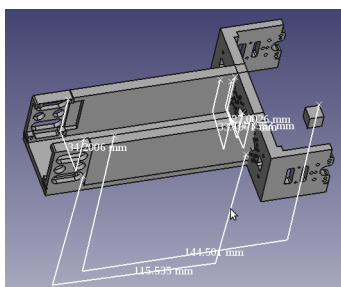


FIGURA B.3: Brazo

Antebrazo. La siguiente pieza sirve de unión entre el codo y la muñeca. Sus dimensiones están de acuerdo al tipo de motor que lleva (MT-64).

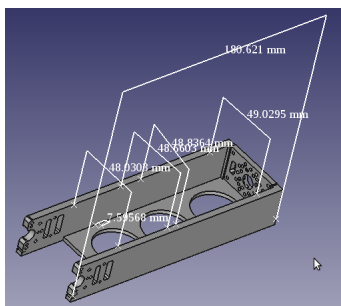


FIGURA B.4: Antebrazo

Base codo-muñeca. Esta pieza sirve de soporte para el motor que permitirá el movimiento de la muñeca del brazo. Es mas pequeña que las anteriores.

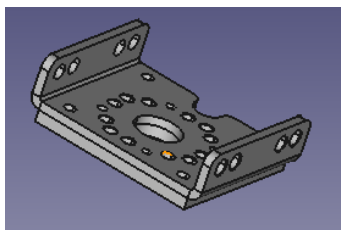


FIGURA B.5: Base codo-muñeca

Bracket de la muñeca. Esta pieza une el brazo completo con el gripper, y con esta logra tener 3 grados de libertad el brazo robot.

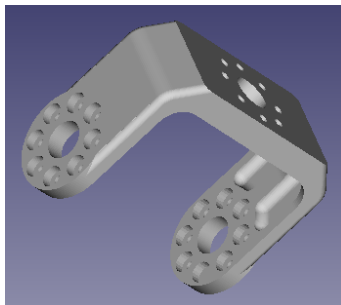


FIGURA B.6: Bracket de la muñeca

El diseño 3D del brazo completo se muestra en la Fig.B.7.

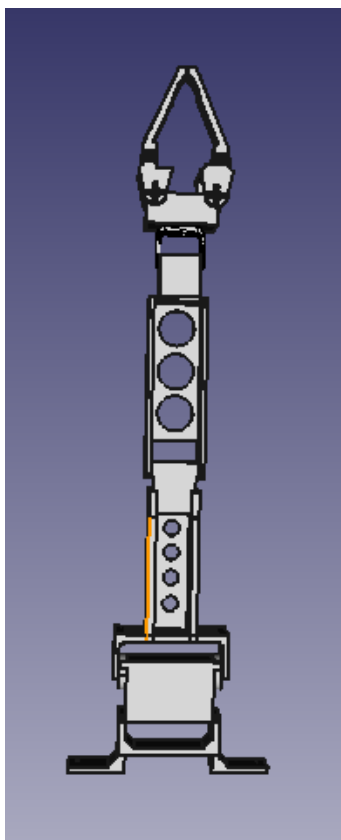


FIGURA B.7: Brazo con todas las piezas

En total se imprimió un número de 6 piezas lo cual llevo aproximadamente mas de 150 horas desde el diseño hasta la impresión, lijado, corte, y perforación de cada una de ellas.

En la Fig.B.8 se muestra una secuencia de la impresión de la pieza del antebrazo.

Una vez terminada esta pieza se coloco el motor y posteriormente el gripper, Fig.B.9

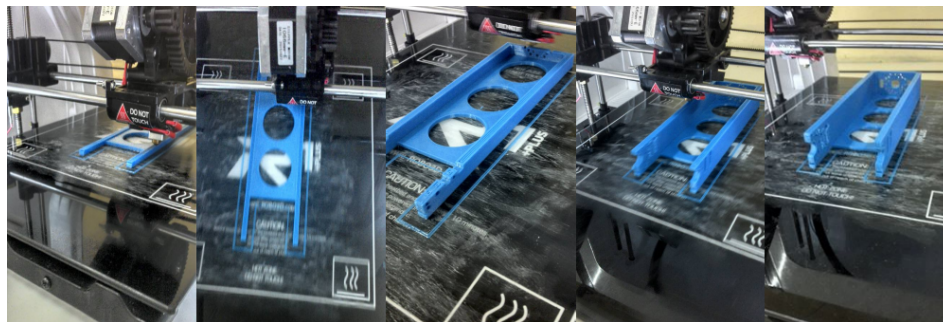
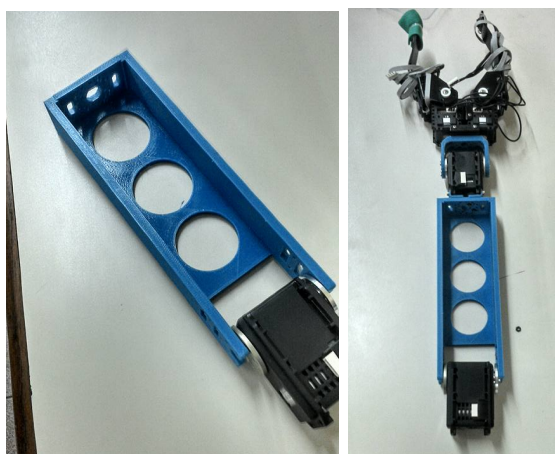


FIGURA B.8: Impresión de la pieza llamada antebrazo.



(a)

(b)

FIGURA B.9: a) Motor dynamixel conectado a la pieza antebrazo, b) pinza montada sobre la pieza antebrazo.

Por otra parte también se ensambló la pieza que une los dos motores grandes junto con la pieza llamada brazo y el motor que sirve de codo, Fig.B.10. Una vez ensamblado el prototipo se procedió a cablear todo el brazo, es decir, los cables de los motores (datos, alimentación) y los cables de los sensores de fuerza y acelerómetro (datos, alimentación) Fig.B.11.



FIGURA B.10: Ensamble de los dos motores dynamixel MX-106T



FIGURA B.11: Prototipo de brazo finalmente armado

Apéndice C

Implementación en ROS

Tomando el diseño 3D del brazo, se puede exportar a ROS para tener una visualización en todo momento del estado interno del brazo. En la Fig.C.1 se muestra la metodología empleada para este proceso. Para es necesario tener un archivo llamado

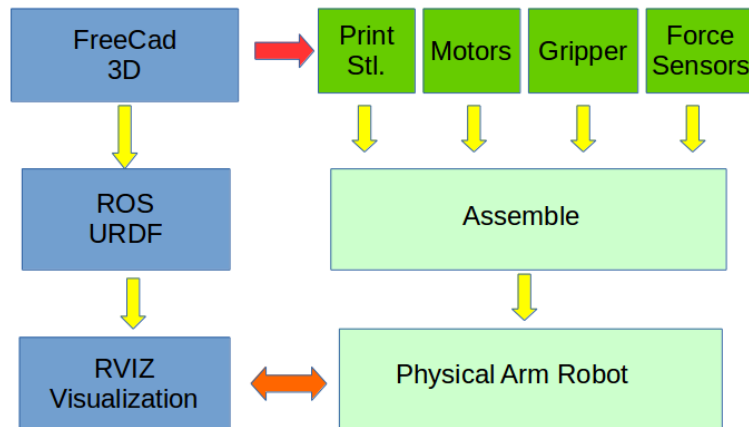


FIGURA C.1: Estructura del diseño del brazo y comunicación con ROS.

URDF (Unified Robot Description Format) [42] el cual es un archivo XML específico que describe el robot. Los archivos URDF se utilizan para registrar toda la información de un robot virtual. Por lo regular se crea un modelo CAD del robot con alguna herramienta de diseño en nuestro caso FreeCAD. Y se debe crear una transformación del modelo CAD al modelo de descripción URDF del robot. En ROS la estructura básica de un robot siempre está dividida en *Links* y *Joints*. Como ejemplo podemos observar la Fig. C.2

En el modelo de descripción URDF del robot, los elementos completos entre etiqueta `< link >` y etiqueta `< /link >` son la descripción de un link específico, lo mismo sucede con la descripción joint entre la etiqueta `< joint >` y la etiqueta `< /joint >`. Otro componente dentro del archivo URDF es el que se necesita importar para leer el objeto

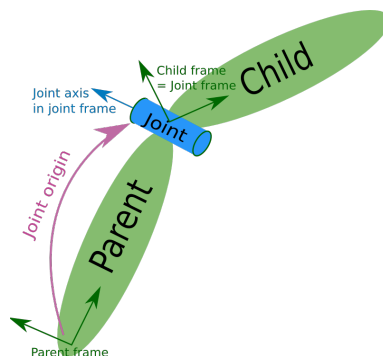


FIGURA C.2: Joint.

3D, es decir, debemos incluir un archivo con la extensión STL (STereoLithography), el cual fue creado en FreeCad. Cuando creamos un archivo URDF e importamos algún modelo en STL, básicamente lo que tenemos que hacer es unir diversos *Links*. La conexión entre estos *links* son los *Joints*, un joint es la unión entre dos diferentes links. Y como vimos en la Fig. C.2 cada link puede ser dividido en padre o en hijo y cada uno debe tener su propio marco de referencia y es importante colocar correctamente cada uno de los links hasta el link padre para tener una correcta transformación entre cada uno de ellos.

Cuando se tiene terminado el archivo URDF, se desea visualizar este archivo en RViz que es un interfaz gráfica para visualizar el robot en un ambiente 3D. Cuando se ejecuta el nodo de RViz se deben ejecutar también otros nodos que nos permitan establecer una red con ROS para poder controlar la posición de cada uno de los joints y además verificar la correcta conexión entre los links (Fig. C.3).

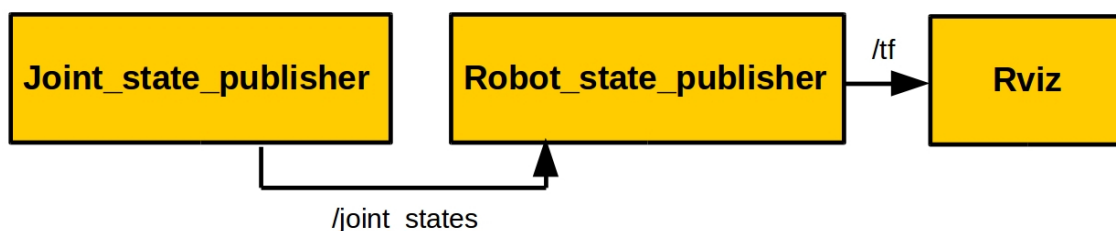


FIGURA C.3: Cuando RViz es ejecutado los nodos llamados *joint_state_publisher*, *robot_state_publisher* y *RViz*, y los topics llamados *joint_states* y *tf*, son transportados siguiendo la dirección de las flechas.

A continuación se explican brevemente cada uno de ellos.

- **joint state publisher:** Este nodo lee los parámetros de descripción del robot y publica toda la información de los joints no fijos, es decir información sobre velocidad, posición, aceleración a una frecuencia específica. También provee una interfaz gráfica para establecer la posición de cada uno de los joints.

- **robot state publisher**: Este nodo se suscribe a la posición actual de los joints en el brazo y publica los datos transformados de los marcos referenciales en cada link del brazo, por el método de cinemática inversa.
- **tf**: Esta librería fue diseñada para proveer una forma estándar de la transformación de los marcos de referencia en el sistema completo.

En nuestro prototipo de brazo robótico, se asignaron los siguientes nombres para identificara a cada uno de los links y joints que lo componen y esto se observa en la Fig. C.4

Links

- Base Arm
- Upper Arm Link
- Fore Arm Link
- RX-28

Joints

- shoulder_joint
- elbow_joint
- wrist_joint

En la Fig. C.5 se muestra una secuencia de como se fueron agregando cada uno de los links y joints del brazo hasta llegar al gripper.

Por otra parte para comprobar el estado del árbol de todas las transformaciones se ejecuta el comando `roslaunch tf view_frames` el cual nos arroja el resultado mostrado en la Fig. C.6. Finalmente la visualización del brazo en RViz se muestra en Fig. C.7, además son desplegados los *tf* de cada uno de los *links* de este. En la parte inferior aparece un cuadro donde se publica los *joint State Publisher* para cada uno de los motores, por lo tanto cada una de la articulaciones de puede mover a una posición deseada. Si el brazo físico estuviera conectado se podrían mover cada uno de los motores y visualizar el estado de cada motor en RViz.

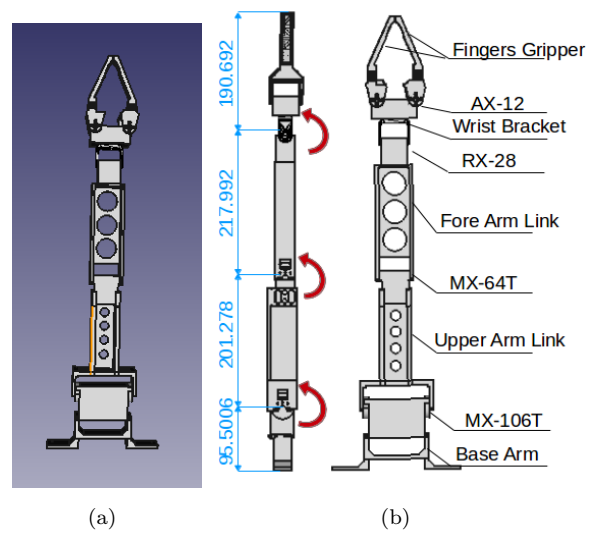


FIGURA C.4: a) Modelo 3D del brazo en FreeCad, b) Medidas y nombres de cada una de las partes que se asignan en ROS.

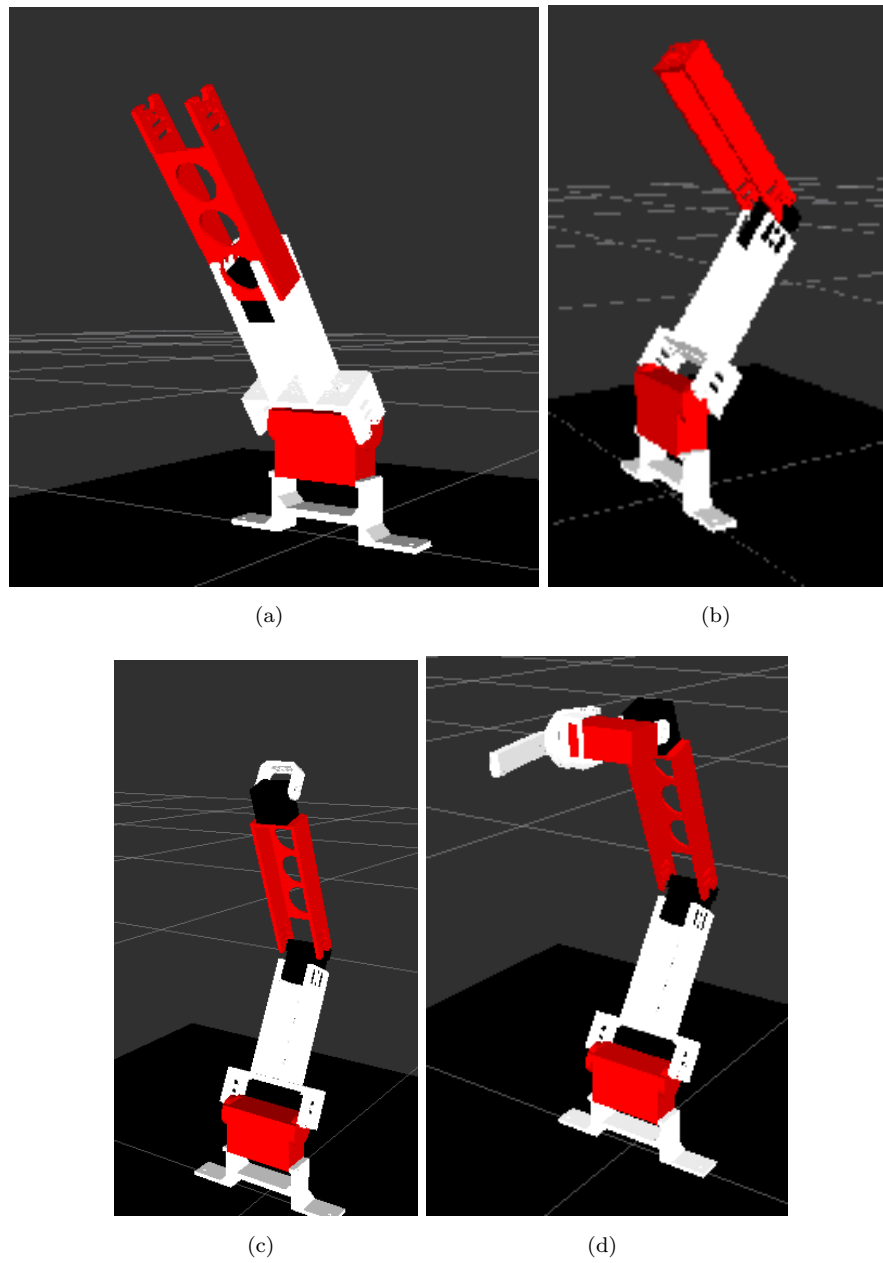


FIGURA C.5: Secuencia de armado de los joints y links del brazo robótico.

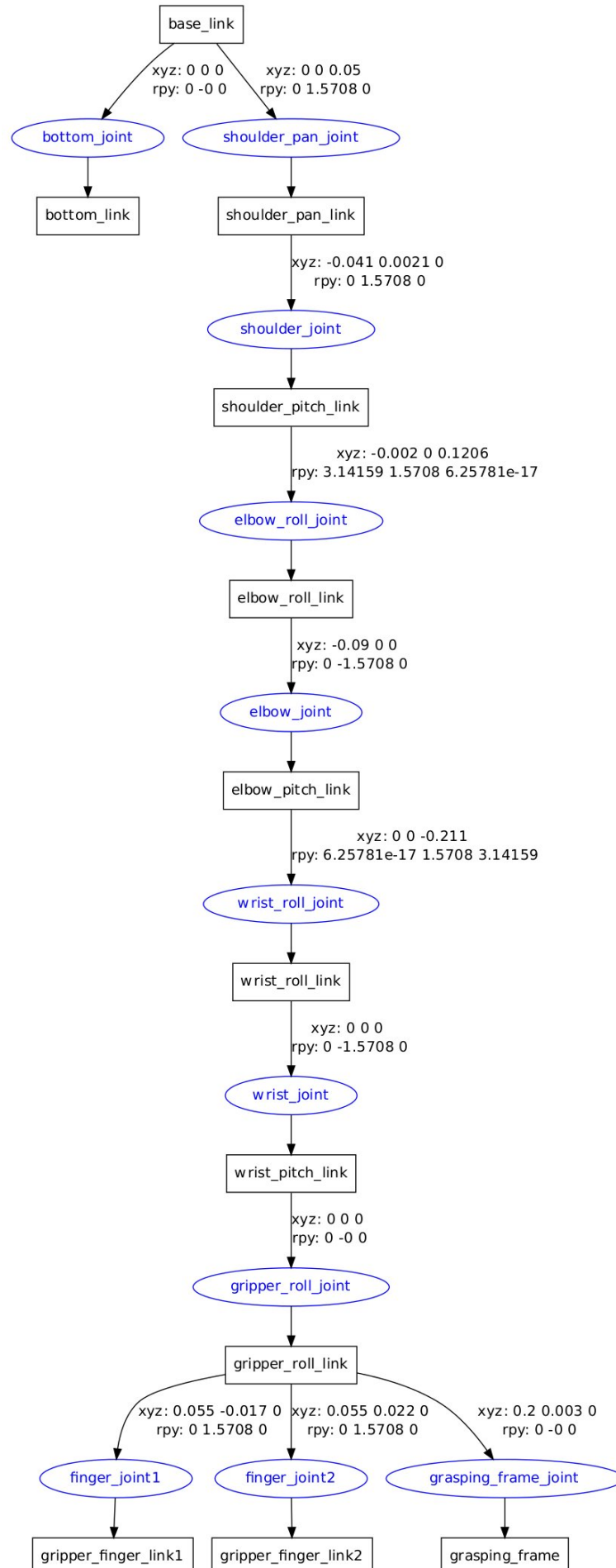


FIGURA C.6: Esquema de todas las transformaciones del brazo desde el *base_link* hasta *grasping_frame*.

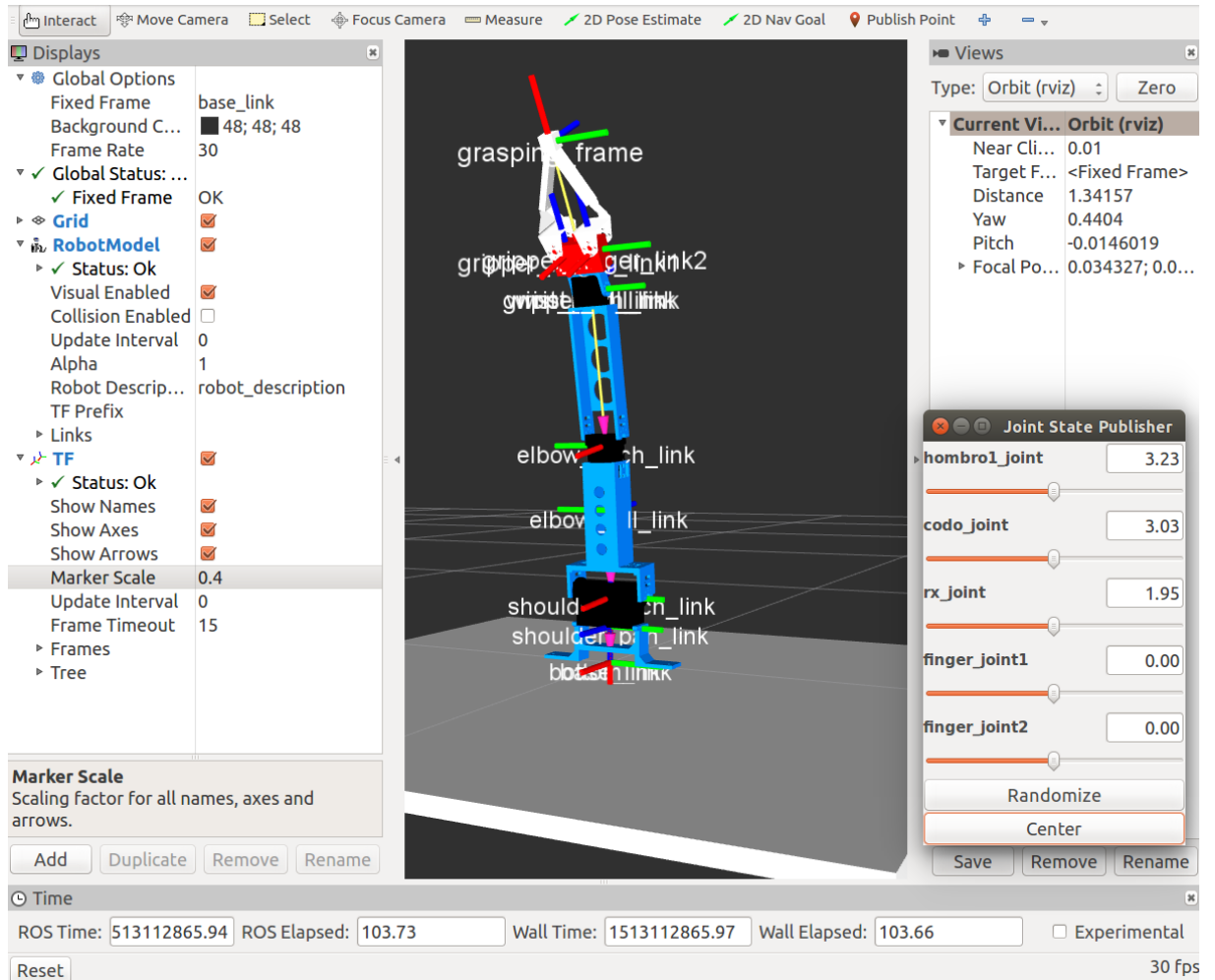


FIGURA C.7: Visualización en RViz del prototipo de brazo de 3 grados de libertad.

Apéndice D

Moveit!

Move it! [17] es un software de última generación para la manipulación móvil, incorporando los últimos avances en la planificación de movimiento, manipulación, percepción 3D, cinemática, y navegación. Proporciona una plataforma en ROS fácil de usar para desarrollar aplicaciones robóticas avanzadas, evaluar nuevos diseños de robots y construir productos robóticos integrados para investigación. El objetivo de usar Moveit! en este trabajo, es para poder dirigir el brazo hacia un objetivo específico, es decir, dirigirlo hacia el objeto que se desea manipular.



En la Fig.D.1 se muestran diversos robots en los que actualmente se esta empleando Moveit! como planificador de movimientos.

Moveit! emplea un *planning scene* para representar el mundo alrededor del robot y también guarda el estado interno del robot mismo. En la Fig. D.2 se muestra el diagrama de principal del *planning scene* de Moveit!. Este *planning scene* lee el topic de *joints_states* del robot, y también lee la información de algún otro sensor (vídeo,3D,temperatura,etc). Para nuestro brazo solo se leerán los *joints_states* los cuales corresponden a cada uno de los motores del brazo, descritos en la sección anterior.

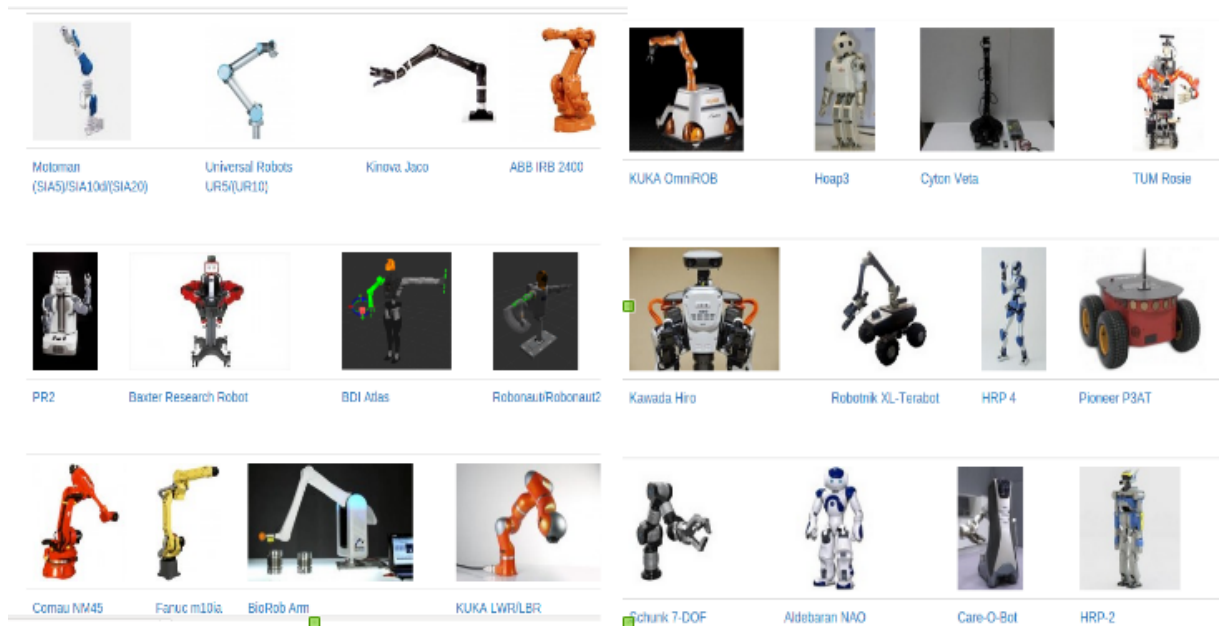


FIGURA D.1: Diversas plataformas robóticas que emplean moveit!.

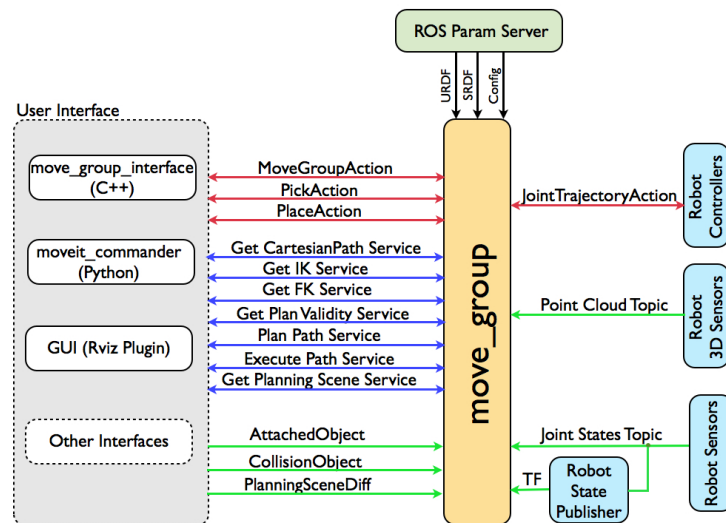


FIGURA D.2: Diagrama de la descripción del *planning scene* en Moveit!.

D.0.1. Planificadores Moveit

Moveit! emplea diversos planificadores, los cuales ofrecen soluciones aleatorias a los problemas de planificación. Por default ROS trae configurado OMPL, pero se puede instalar y configurar algún otro. A continuación se mencionan los cuatro principales:

- Open Motion Planning Library (OMPL).** Es una biblioteca de planificación de movimiento de código abierto que implementa principalmente planificadores de movimiento aleatorios.

- Stochastic Trajectory Optimization for Motion Planning (STOMP).
- Search-Based Planning Library (SBPL).
- Covariant Hamiltonian Optimization for Motion Planning (CHOMP)

D.0.2. Pasos para configurar el asistente de Moveit! con el brazo

A continuación se muestran los pasos que se necesitaron para configurar Moveit! usando el asistente, en el brazo robotico empleado en esta tesis.

- 1) Lanzar la herramienta del asistente.

Esta herramienta permite crear un nuevo paquete de configuración e editar uno ya existente (Fig. D.3). En este caso seleccionamos la opción de crear, *Create New MoveIt! Configuration Package*. En este paso, el asistente pregunta por el modelo *URDF* del nuevo nuestro robot. Presionamos el botón de buscar para encontrar el archivo *brazo_3g.xacro* y posteriormente se presiona el boton de *Load* para cargar nuestro archivo *URDF*. Esta herramienta nos permite trabajar ya sea con modelo *URDF* o *xacro*, ya que si cargamos un archivo *xacro*, el asistente lo convertirá en *URDF* automáticamente.

Si el modelo de robot se carga con éxito, podemos ver el modelo de robot en la ventana, como se muestra en Fig. D.4.

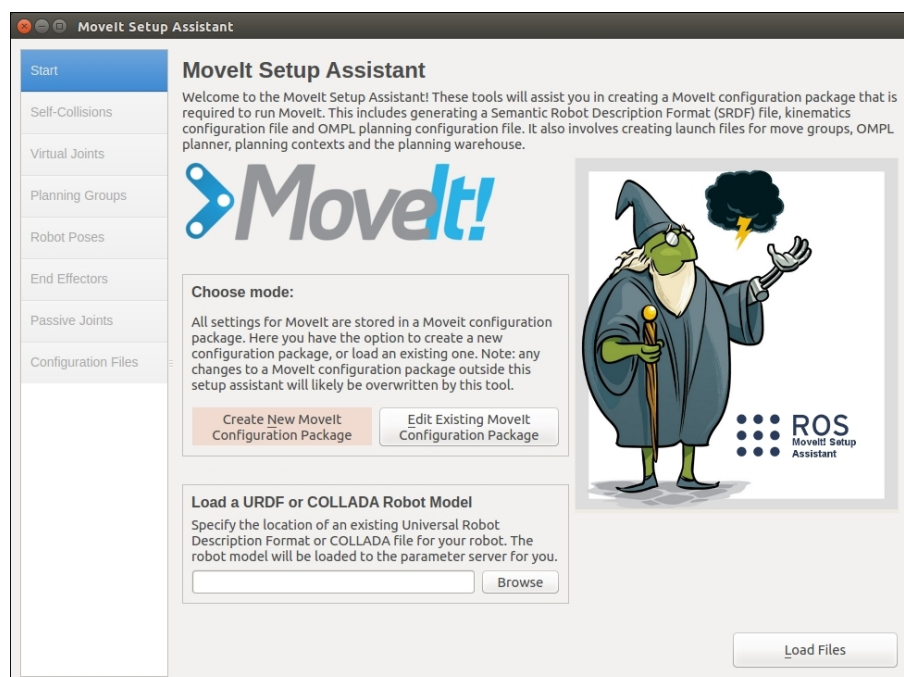


FIGURA D.3: Asistente de configuración de movit!.

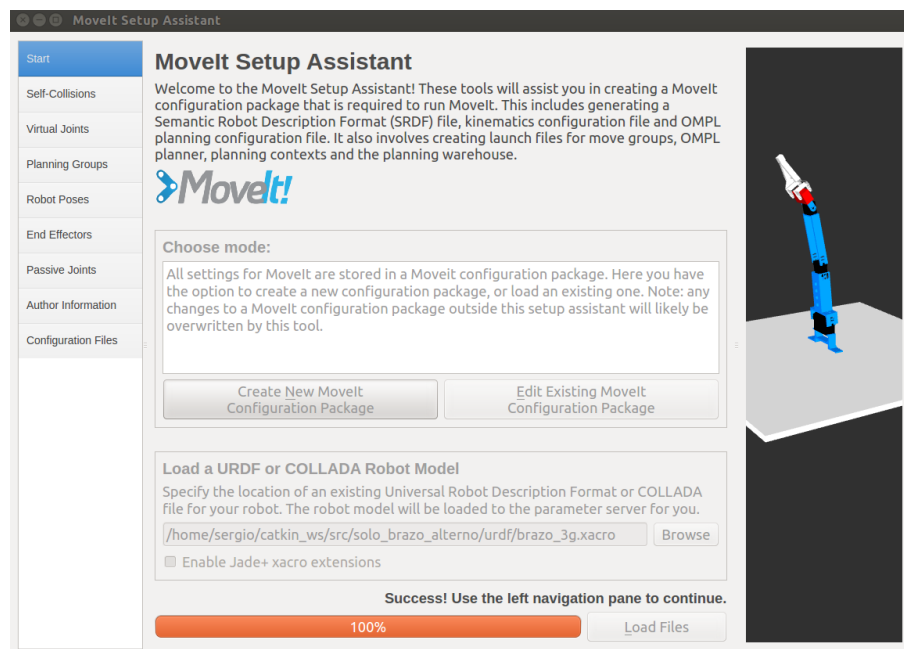


FIGURA D.4: Carga con éxito el modelo de robot en la herramienta Asistente de configuración.

■ 2) Generar matriz de auto-colisión

Moveit! busca un par de *links* en el robot que pueden ser desactivados de forma segura desde la verificación de colisión, y por lo tanto reducir el tiempo de procesamiento. Se revisa cada par de *links* y categoriza los *links* como siempre en *colisión*, *nunca en colisión*, *defecto de colisión*, *a veces en colisión* entre otros, en la Fig. D.5 se muestra las auto-colisiones.

La densidad de muestreo es el número de posiciones aleatorias para verificar la colisión. El cálculo es alto si la densidad es grande, pero la auto-colisión será menos. Por default el valor es de 10,000. En Fig. D.5 se observan las casillas que podemos desmarcar y volver a generar la matriz de colisión presionando el *Regenerate Default Collision Matrix*.

■ 3) Agregar grupos de planificación

Un grupo de planificación es un grupo de *joints/links* en un brazo robótico que ejecuta un plan para alcanzar una posición objetivo de un gripper. Para nuestro brazo se tiene que crear dos grupos de planificación, uno para todos los elementos del brazo hasta la muñeca y otro para el gripper. Para el planificador del brazo, le asignamos un nombre y seleccionamos el plugin por default de Moveit!, el cual es *kdl_kinematics_plugin/KDLKinematicsPlugin* Fig. D.5. Además a este grupo se le tiene que agregar toda la cadena cinemática, desde la base (*base.link*) hasta el gripper.

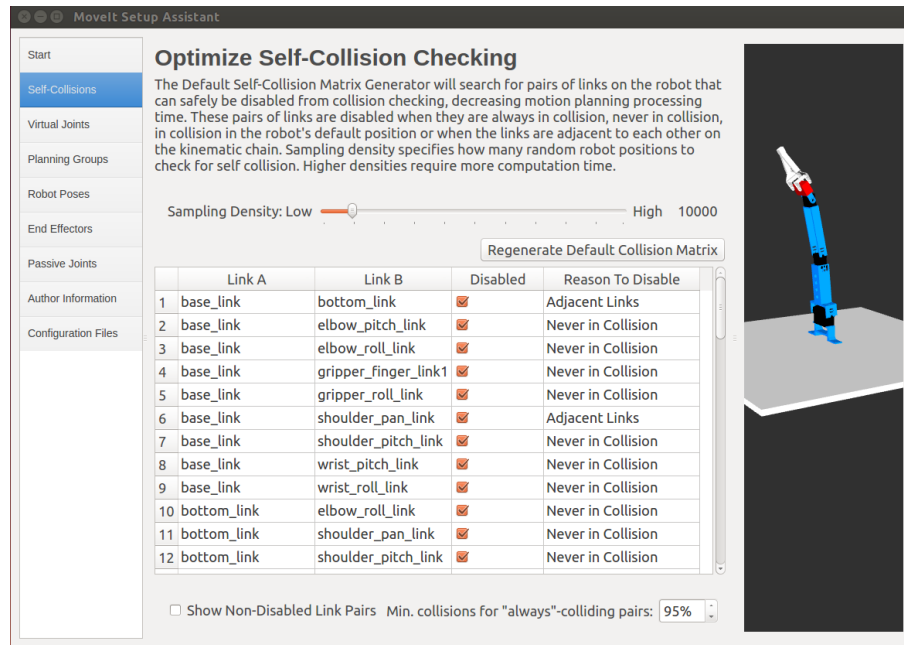


FIGURA D.5: Regenerando la matriz de auto colisión.

El segundo grupo, que corresponde al gripper, no se necesita ninguna cadena cinemática, solo se agrega los dos *joints* correspondientes a cada dedo del gripper angular robótico. Este se puede observar en Fig. D.7.

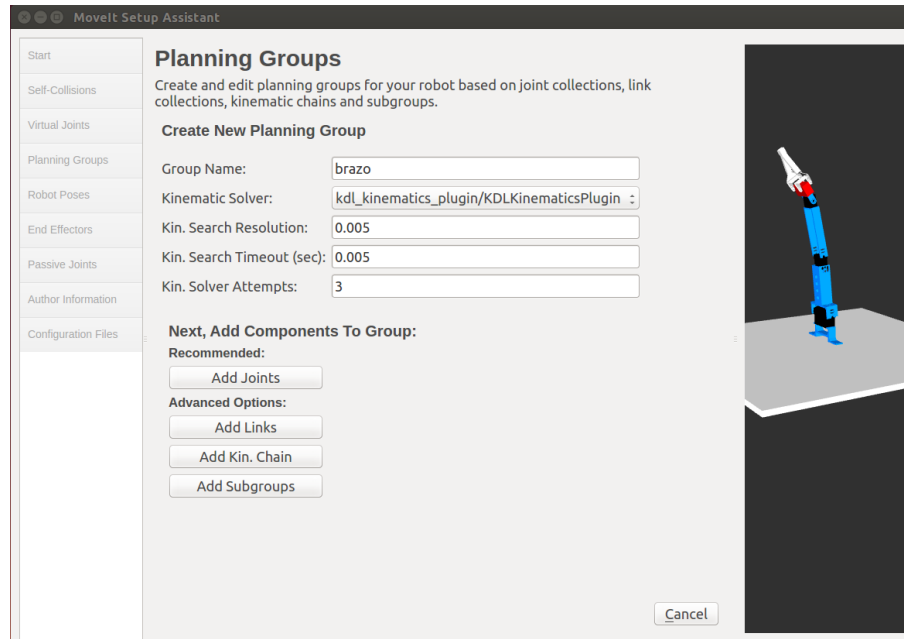


FIGURA D.6: Agregando grupo de planificación al brazo.

- 4) **Agregar pose** Con MoveIt! se puede agregar ciertas poses fijas en la configuración del brazo, las cuales pueden ser de inicio, o de cercanía a un objeto. Esto se puede apreciar en la Fig. D.8.

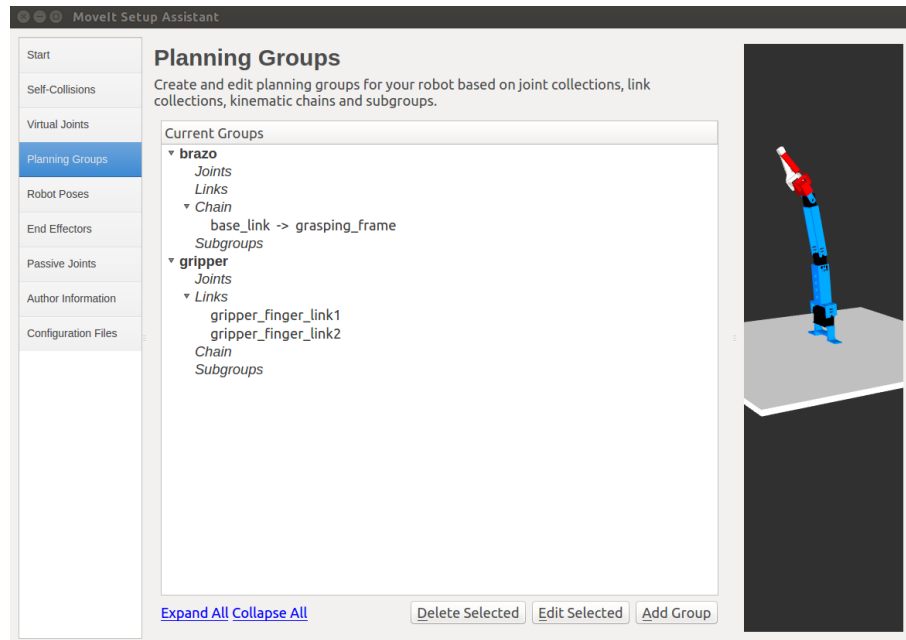


FIGURA D.7: Agregando grupo de planificación al brazo y el gripper.

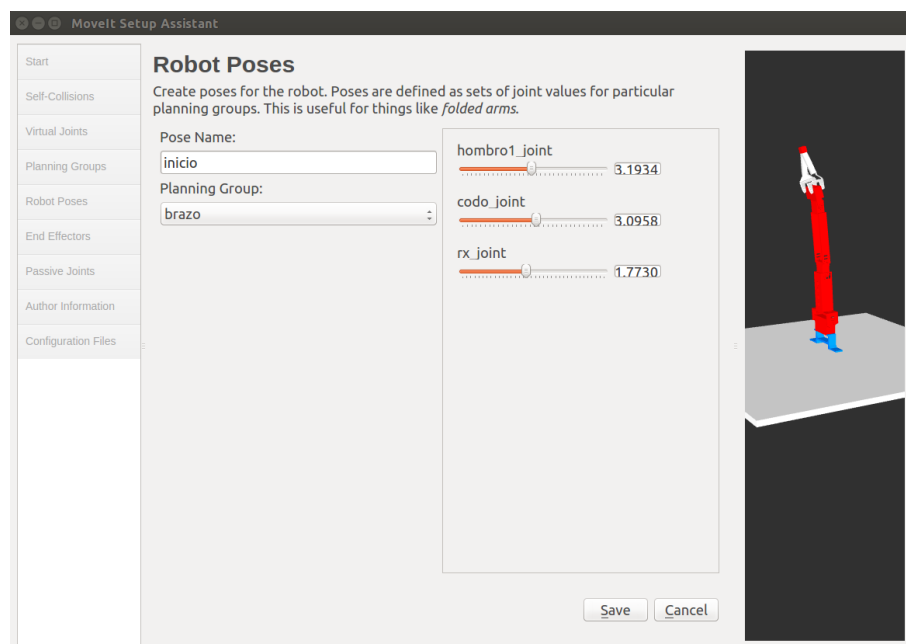


FIGURA D.8: Agregando pose inicial del brazo.

- **5) Configurar el efector final del robot** El brazo que se ocupó en este trabajo, se agregó un efector final (gripper) y le asignamos un grupo ya definido previamente, el link padre y el grupo padre. En la Fig. D.9 se puede apreciar esto.
- **6) Generar archivos de configuración** Una vez realizados los pasos anteriores se debe especificar una dirección en donde se creará el paquete con todo lo necesario para MoveIt!. Se recomienda crear el paquete en la carpeta src de ROS, para que se pueda compilar sin ningún problema. Esto se puede apreciar en Fig.D.10.

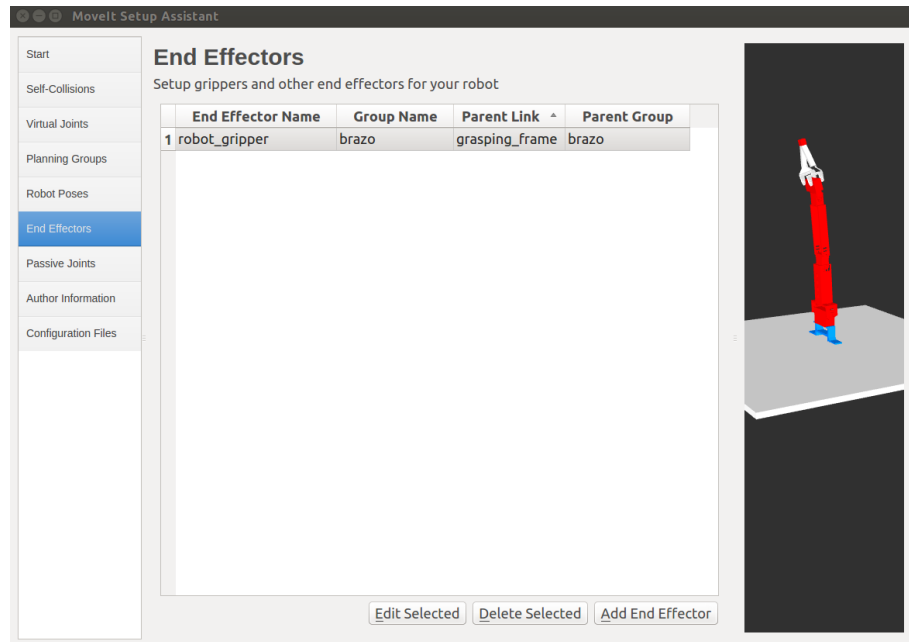


FIGURA D.9: Agregando end effector.

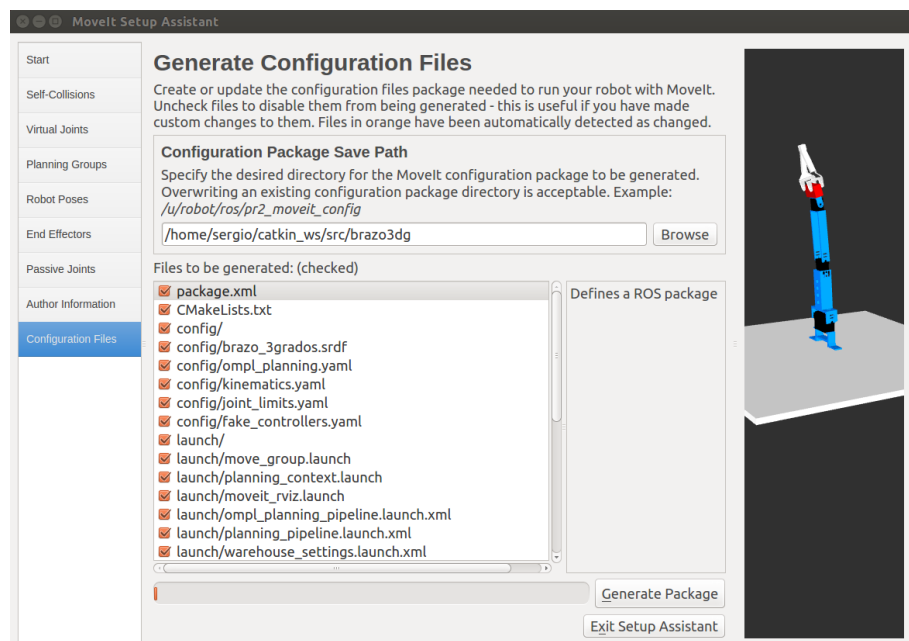


FIGURA D.10: Generando el paquete de configuración de Moveit!.

Finalmente creado el paquete solo se debe lanzar desde ROS y el brazo sera visualizado en RViz junto con Moveit!, Fig. D.11.

Todos los paquetes mencionados anteriormente permiten controlar el brazo en ROS, cada uno de los paquetes cumple una función específica y se comunican entre si en la Fig. D.12 se muestran todos los paquetes que se ejecutan en ROS.

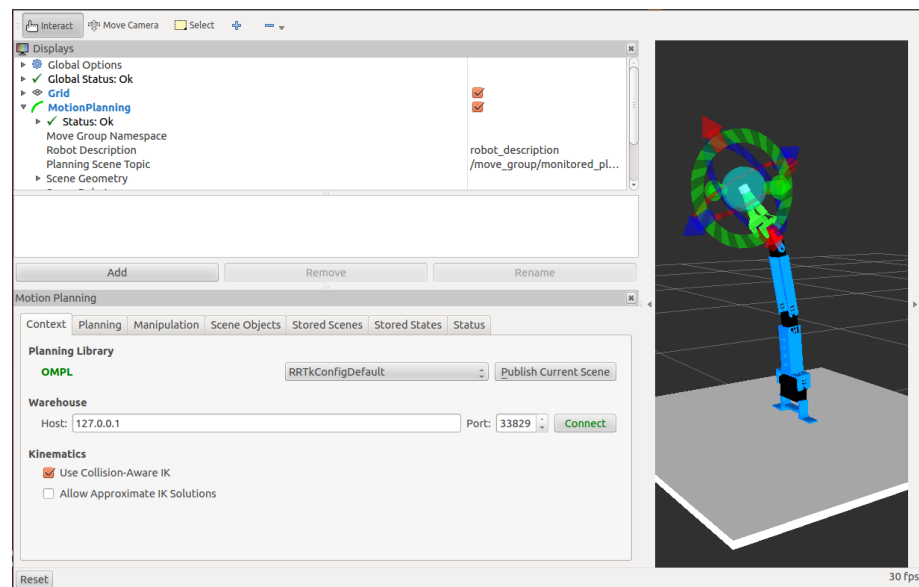


FIGURA D.11: Interfaz de planificación de movimiento de Moveit! en RViz.

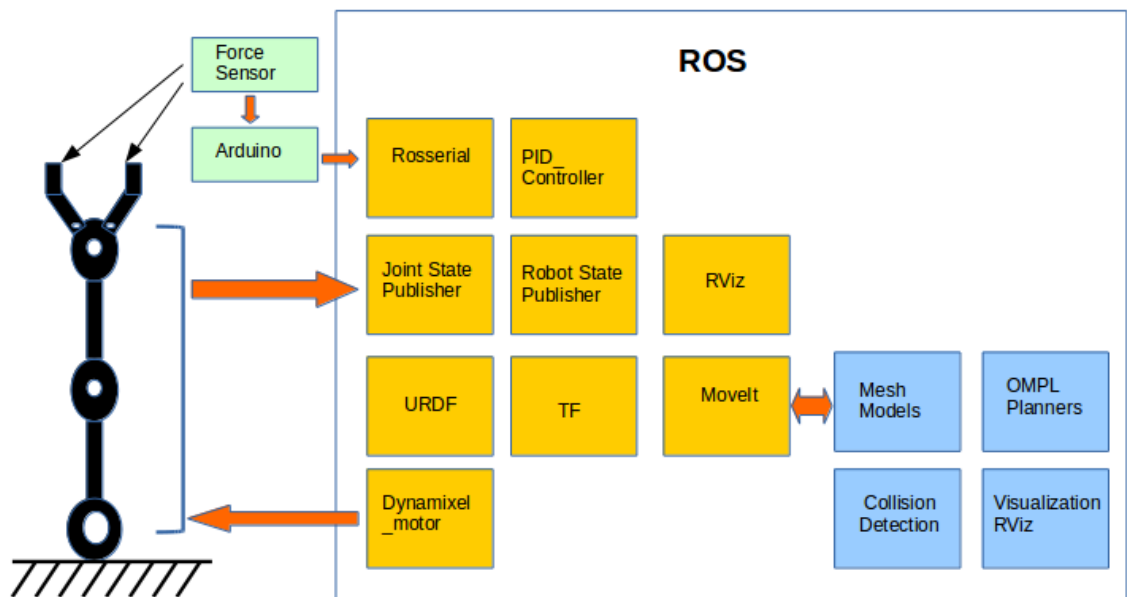


FIGURA D.12: Arquitectura del hardware y software con los paquetes de ROS.

Bibliografía

- [1] AHMAD, H., RAZALI, S., AND MOHAMED, M. R. Fuzzy logic controller design for a robot grasping system with different membership functions. In *IOP Conference Series: Materials Science and Engineering* (2013), vol. 53, IOP Publishing, p. 012051.
- [2] ALICI, G., AND SHIRINZADEH, B. Enhanced stiffness modeling, identification and characterization for robot manipulators. *IEEE Transactions on Robotics* 21, 4 (Aug 2005), 554–564.
- [3] ALMASSRI, A. M., WAN HASAN, W., AHMAD, S. A., ISHAK, A. J., GHAZALI, A., TALIB, D., AND WADA, C. Pressure sensor: state of the art, design, and application for robotic hand. *Journal of Sensors 2015* (2015).
- [4] ANTSAKLIS, P., ET AL. Defining intelligent control. *Report of the Task Force on Intelligent Control, IEEE Control Systems* (1994), 0272–1708.
- [5] BARRETT TECHNOLOGY, L. Bh8-series barretthand. url <http://www.barrett.com/products-hand.htm>, 2009.
- [6] BELTER, J. T., AND DOLLAR, A. M. Performance characteristics of anthropomorphic prosthetic hands. In *2011 IEEE International Conference on Rehabilitation Robotics* (June 2011), pp. 1–7.
- [7] BIAGIOTTI, L., LIU, H., HIRZINGER, G., AND MELCHIORRI, C. Cartesian impedance control for dexterous manipulation. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)* (Oct 2003), vol. 4, pp. 3270–3275 vol.3.
- [8] BICCHI, A., AND KUMAR, V. Robotic grasping and contact: a review. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)* (2000), vol. 1, pp. 348–353 vol.1.

-
- [9] BORST, C., FISCHER, M., HAIDACHER, S., LIU, H., AND HIRZINGER, G. Dlr hand ii: experiments and experience with an anthropomorphic hand. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on* (2003), vol. 1, IEEE, pp. 702–707.
- [10] BOUGHDIRI, R., BEZINE, H., AND ALIMI, A. M. Fuzzy logic control for grasping 3d objects with sliding contacts. In *Proceedings of International Conference on Control, Engineering and Information Technology* (2013), vol. 2, pp. 116–119.
- [11] BROWN, E., RODENBERG, N., AMEND, J., MOZEIKA, A., STELTZ, E., ZAKIN, M. R., LIPSON, H., AND JAEGER, H. M. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences* 107, 44 (2010), 18809–18814.
- [12] BUTTERFASS, J., GREBENSTEIN, M., LIU, H., AND HIRZINGER, G. Dlr-hand ii: Next generation of a dextrous robot hand. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (2001), vol. 1, IEEE, pp. 109–114.
- [13] CANNATA, G., AND MAGGIALI, M. Design of a tactile sensor for robot hands. In *Sensors: Focus on Tactile Force and Stress Sensors*. InTech, 2008.
- [14] CASTRO, D., MARQUES, L., NUNES, U., AND DE ALMEIDA, A. T. Tactile force control feedback in a parallel jaw gripper. In *Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on* (Jul 1997), pp. 884–888 vol.3.
- [15] CHARLEBOIS, M., GUPTA, K., AND PAYANDEH, S. On estimating local shape using contact sensing. *Journal of Field Robotics* 17, 12 (2000), 643–658.
- [16] CHEN, S. J., HUANG, J. L., WU, G. J., WU, C. L., AND PAN, S. S. Design and characterization of a pcb based capacitive shear force sensor for robotic gripper application. In *2013 Seventh International Conference on Sensing Technology (ICST)* (Dec 2013), pp. 884–888.
- [17] CHITTA, S., SUCAN, I., AND COUSINS, S. Moveit! [ros topics]. *IEEE Robotics Automation Magazine* 19, 1 (March 2012), 18–19.
- [18] CIOBANU, V., AND POPESCU, N. Tactile controller using fuzzy logic for robot inhand manipulation. In *System Theory, Control and Computing (ICSTCC), 2015 19th International Conference on* (2015), IEEE, pp. 435–440.
- [19] CIOCARLIE, M., MILLER, A., AND ALLEN, P. Grasp analysis using deformable fingers. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on* (2005), IEEE, pp. 4122–4128.

-
- [20] COMPANY, S. R. Design of a dextrous hand for advanced clawar applications. CLAWAR.
- [21] CUTKOSKY, M. R. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation* 5, 3 (Jun 1989), 269–279.
- [22] DAHIYA, R. S., AND VALLE, M. Tactile sensing for robotic applications. In *Sensors: Focus on Tactile Force and Stress Sensors*. InTech, 2008.
- [23] DARIO, P., BERGAMASCO, M., AND FIORILLO, A. Force and tactile sensing for robots. In *Sensors and sensory systems for advanced robots*. Springer, 1988, pp. 153–185.
- [24] DE SILVA, C. W. Applications of fuzzy logic in the control of robotic manipulators. *Fuzzy Sets and Systems* 70, 2-3 (1995), 223–234.
- [25] DIMEAS, F., SAKO, D. V., MOULIANITIS, V. C., AND ASPRAGATHOS, N. A. Design and fuzzy control of a robotic gripper for efficient strawberry harvesting. *Robotica* 33, 05 (2015), 1085–1098.
- [26] DOERSAM, T., AND MUNOZ UBANDO, L. Robotic hands: modelisation, control and grasping strategies. In *Meeting annuel de L’Institute Fanco-Allemand pour les Application de la recherche IAR* (1995).
- [27] DOMINGUEZ-LOPEZ, J. A., AND VILA-ROSADO, D. N. Hierarchical fuzzy control to ensure stable grasping. In *2006 Seventh Mexican International Conference on Computer Science* (Sept 2006), pp. 37–43.
- [28] DUBEY, V. N., CROWDER, R. M., AND CHAPPELL, P. H. Optimal object grasp using tactile sensors and fuzzy logic. *Robotica* 17, 6 (1999), 685–693.
- [29] EDSINGER, A., AND WEBER, J. Meka robotics h2. url <http://www.hizook.com/blog/2009/10/18/meka-robotics-humanoid-torso-and-anthropomorphic-hands>, 2009.
- [30] GAMA MELO, E. N., AVILES SANCHEZ, O. F., AND AMAYA HURTADO, D. Anthropomorphic robotic hands: a review. *Ingeniería y Desarrollo* 32, 2 (2014), 279–313.
- [31] GLOSSAS, N., AND ASPRAGATHOS, N. Fuzzy logic grasp control using tactile sensors. *Mechatronics* 11, 7 (2001), 899–920.
- [32] HA, X. V., HA, C., AND NGUYEN, D. K. A general contact force analysis of an under-actuated finger in robot hand grasping. *International Journal of Advanced Robotic Systems* 13, 1 (2016), 14.

- [33] HANG, C. C., ÅSTRÖM, K. J., AND HO, W. K. Refinements of the ziegler–nichols tuning formula. In *IEE Proceedings D (Control Theory and Applications)* (1991), vol. 138, IET, pp. 111–118.
- [34] HARADA, K., NAKANO, S., KANEKO, M., AND TSUJI, T. Manipulation of enveloped multiple objects. In *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies* (2000), vol. 1, pp. 422–427 vol.1.
- [35] HOWE, R. D., POPP, N., AKELLA, P., KAO, I., AND CUTKOSKY, M. R. Grasping, manipulation, and control with tactile sensing. In *Proceedings., IEEE International Conference on Robotics and Automation* (May 1990), pp. 1258–1263 vol.2.
- [36] JACOBSEN, S., IVERSEN, E., KNUTTI, D., JOHNSON, R., AND BIGGERS, K. Design of the utah/mit dextrous hand. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on* (1986), vol. 3, IEEE, pp. 1520–1532.
- [37] JOHNSTON, D., ZHANG, P., HOLLERBACH, J., AND JACOBSEN, S. A full tactile sensing suite for dextrous robot hands and use in contact force control. In *Proceedings of IEEE International Conference on Robotics and Automation* (Apr 1996), vol. 4, pp. 3222–3227 vol.4.
- [38] KAO, I., AND YANG, F. Stiffness and contact mechanics for soft fingers in grasping and manipulation. *IEEE Transactions on Robotics and Automation* 20, 1 (Feb 2004), 132–135.
- [39] KARNIK, N. N., MENDEL, J. M., AND LIANG, Q. Type-2 fuzzy logic systems. *IEEE transactions on Fuzzy Systems* 7, 6 (1999), 643–658.
- [40] KAWASAKI, H., KOMATSU, T., AND UCHIYAMA, K. Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii. *IEEE/ASME transactions on mechatronics* 7, 3 (2002), 296–303.
- [41] KEMP, C. C., EDSINGER, A., AND TORRES-JARA, E. Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14, 1 (2007), 20–29.
- [42] KUNZE, L., ROEHM, T., AND BEETZ, M. Towards semantic robot description languages. In *2011 IEEE International Conference on Robotics and Automation* (May 2011), pp. 5589–5595.
- [43] LASCHI, C., DARIO, P., CARROZZA, M. C., GUGLIEMELLI, E., TETI, G., TADDEUCCI, D., LEONI, F., MASSA, B., ZECCA, M., AND LAZZARINI, R. Grasping

- and manipulation in humanoid robotics. In *IEEE International Conference on Humanoid Robots (Humanoids 2003)* (2000).
- [44] LIANG, Q., AND MENDEL, J. M. Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy systems* 8, 5 (2000), 535–550.
- [45] LIU, G. J., AND GOLDENBERG, A. A. Robust hybrid impedance control of robot manipulators. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation* (Apr 1991), pp. 287–292 vol.1.
- [46] LOTTI, F. Design aspects for advanced robot hands: Mechanical design. In *Tutorial T1-Towards Intelligent Robotic Manipulation, IEEE/RSJ 2002 International Conference on Intelligent Robots and Systems* (2002).
- [47] M. F. A. JAMIL, J. JALANI, A. A. A new approach of active compliance control via fuzzy logic control for multifingered robot hand. *Proc.SPIE 10011* (2016), 10011 – 10011 – 9.
- [48] MA, R. R., AND DOLLAR, A. M. On dexterity and dexterous manipulation. In *2011 15th International Conference on Advanced Robotics (ICAR)* (June 2011), pp. 1–7.
- [49] MAEKAWA, H., TANIE, K., AND KOMORIYA, K. Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation* (May 1995), vol. 1, pp. 743–750 vol.1.
- [50] MAHESHWARI, V., AND SARAF, R. F. High-resolution thin-film device to sense texture by touch. *Science* 312, 5779 (2006), 1501–1504.
- [51] MALDONADO, A., KLANK, U., AND BEETZ, M. Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct 2010), pp. 2586–2591.
- [52] MANZ, M., BARTSCH, S., SIMNOFSKE, M., AND KIRCHNER, F. Development of a self-adaptive gripper and implementation of a gripping reflex to increase the dynamic payload capacity. In *Proceedings of ISR 2016: 47th International Symposium on Robotics* (June 2016), pp. 1–7.
- [53] MASON, M. T., RODRIGUEZ, A., SRINIVASA, S. S., AND VAZQUEZ, A. S. Autonomous manipulation with a general-purpose simple hand. *The International Journal of Robotics Research* 31, 5 (2012), 688–703.

-
- [54] MELCHIORRI, C., AND KANEKO, M. Robot hands. In *Springer Handbook of Robotics*. Springer, 2016, pp. 463–480.
- [55] MELCHIORRI, C., PALLI, G., BERSELLI, G., AND VASSURA, G. Development of the ub hand iv: Overview of design solutions and enabling technologies. *IEEE Robotics & Automation Magazine* 20, 3 (2013), 72–81.
- [56] MENDEL, J. M. *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall PTR Upper Saddle River, 2001.
- [57] MONKMAN, G. J., HESSE, S., STEINMANN, R., AND SCHUNK, H. *Robot grippers*. John Wiley & Sons, 2007.
- [58] NAKAZAWA, N., KIM, I.-H., INOOKA, H., AND IKEURA, R. Force control of a robot gripper based on human grasping schemes. *Control Engineering Practice* 9, 7 (2001), 735–742.
- [59] NAMIKI, A., IMAI, Y., ISHIKAWA, M., AND KANEKO, M. Development of a high-speed multifingered hand system and its application to catching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)* (Oct 2003), vol. 3, pp. 2666–2671 vol.3.
- [60] NAZEMIZADEH, M., TAHERI, M., AND NAZERI, S. The application of fuzzy-logic method to control of robots: a review study. *International Journal of Mechanical Engineering and Robotics Research* 3, 2 (2014), 229.
- [61] NGUYEN, P. T. A., AND ARIMOTO, S. Performance of pinching motions of two multi-dof robotic fingers with soft-tips. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)* (2001), vol. 3, pp. 2344–2349 vol.3.
- [62] NICHOLLS, H. R., AND LEE, M. H. A survey of robot tactile sensing technology. *The International Journal of Robotics Research* 8, 3 (1989), 3–30.
- [63] PALLI, G., AND MELCHIORRI, C. Optimal control of tendon-sheath transmission systems. *IFAC Proceedings Volumes* 39, 15 (2006), 73–78.
- [64] PATEL, R., AND CORRELL, N. Integrated force and distance sensing using elastomer-embedded commodity proximity sensors. In *Robotics: Science and Systems* (2016).
- [65] PLATT, R., FAGG, A. H., AND GRUPEN, R. A. Extending fingertip grasping to whole body grasping. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)* (Sept 2003), vol. 2, pp. 2677–2682 vol.2.

- [66] QUIGLEY, M., CONLEY, K., GERKEY, B., FAUST, J., FOOTE, T., LEIBS, J., WHEELER, R., AND NG, A. Y. Ros: an open-source robot operating system. In *ICRA workshop on open source software* (2009), vol. 3, Kobe, p. 5.
- [67] SACCOMANDI, P., ZOLLO, L., CIANCIO, A. L., SCHENA, E., FASANO, A., ODDO, C. M., CARROZZA, M. C., AND CAMBONI, D. Tactile piezoresistive sensors for robotic application: Design and metrological characterization. In *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (May 2017), pp. 1–6.
- [68] SALISBURY, J. K. Active stiffness control of a manipulator in cartesian coordinates. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on* (1980), vol. 19, IEEE, pp. 95–100.
- [69] SAUDABAYEV, A., AND VAROL, H. A. Sensors for robotic hands: A survey of state of the art. *IEEE Access* 3 (2015), 1765–1782.
- [70] SHAURI, R. L. A., SALLEH, N. M., AND HADI, A. K. A. Pid position control of 7-dof three-fingered robotic hand for grasping task. In *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)* (Nov 2014), pp. 70–74.
- [71] SHIKIDA, M., SHIMIZU, T., SATO, K., AND ITOIGAWA, K. Active tactile sensor for detecting contact force and hardness of an object. *Sensors and Actuators A: physical* 103, 1 (2003), 213–218.
- [72] SHIMOGA, K. B. Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research* 15, 3 (1996), 230–266.
- [73] SRINIVASA, S. S., BERENSON, D., CAKMAK, M., COLLET, A., DOGAR, M. R., DRAGAN, A. D., KNEPPER, R. A., NIEMUELLER, T., STRABALA, K., WEGHE, M. V., AND ZIEGLER, J. Herb 2.0: Lessons learned from developing a mobile manipulator for the home. *Proceedings of the IEEE* 100, 8 (Aug 2012), 2410–2428.
- [74] SU, K.-H., HUANG, S.-J., AND YANG, C.-Y. Development of robotic grasping gripper based on smart fuzzy controller. *International Journal of Fuzzy Systems* 17, 4 (2015), 595–608.
- [75] TAI, K., EL-SAYED, A.-R., SHAHRIARI, M., BIGLARBEKIAN, M., AND MAHMUD, S. State of the art robotic grippers and applications. *Robotics* 5, 2 (2016), 11.
- [76] TINCANI, V., GRIOLI, G., CATALANO, M. G., GARABINI, M., GRECHI, S., FANTONI, G., AND BICCHI, A. Implementation and control of the velvet fingers: A

- dexterous gripper with active surfaces. In *2013 IEEE International Conference on Robotics and Automation* (May 2013), pp. 2744–2750.
- [77] WANG, X., FAN, X., ZHAO, Y., AND MA, S. Parallel force control for a robot gripper based on grey prediction models. In *2012 Power Engineering and Automation Conference* (Sept 2012), pp. 1–5.
- [78] WIDHIADA, W., NINDHIA, T., AND BUDIARSA, N. Robust control for the motion five fingered robot gripper. *International Journal Of Mechanical Engineering And Robotics Research* 4, 3 (2015), 226.
- [79] WILLOWGARAGE. Pr2. url <http://www.willowgarage.com/pages/pr2/overview>, 2009.
- [80] YIN, Y., LUO, Z., SVININ, M., AND HOSOE, S. Hybrid control of multi-fingered robot hand for dexterous manipulation. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on* (Oct 2003), vol. 4, pp. 3639–3644 vol.4.
- [81] YUJI, J.-I., AND SONODA, C. A pvdf tactile sensor for static contact force and contact temperature. In *Sensors, 2006. 5th IEEE Conference on* (2006), IEEE, pp. 738–741.
- [82] ZAKI, A. M., SOLIMAN, A. M., MAHGOUB, O. A., AND EL-SHAFEI, A. M. Design and implementation of efficient intelligent robotic gripper. In *Proceedings of the 2010 International Conference on Modelling, Identification and Control* (July 2010), pp. 710–716.