



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Aplicación de modelos psicológicos  
formales en el desarrollo de robots  
móviles autónomos**

**TESIS**

Que para obtener el título de  
**Ingeniero en Computación**

**P R E S E N T A**

Victor Hugo Sánchez Correa

**DIRECTOR DE TESIS**

Dr. Marco Antonio Negrete Villanueva



Ciudad Universitaria, Cd. Mx., 2020

# Dedicatoria

Esta tesis se la dedico a las dos personas que son el motor de vida, son las dos personas que me inspiran a ser una mejor persona cada día que empieza, las dos personas que con su cariño, amor, comprensión, enojos, regaños, explicaciones y acciones me han convertido el hombre que soy hoy en día.

Ma. Elena, que te puedo decir sino es gracias por dedicarme todos los días de tu vida desde hace ya tantos años y esforzarte por cubrir todas mis necesidades desde que llegué a este mundo. Solo espero que te guste este trabajo que es fruto de todo lo que has hecho por mí.

Mara, desde que recuerdo has estado para oírme, aconsejarme, acompañarme a cada paso de mi vida y nunca tendré forma de pagar todo lo que has hecho por mí, debido a esto espero que este trabajo sea un pequeña compensación por las innumerables veces que has estado para mí en todos los aspectos de mi vida.

Por último, solo quiero decirles que son lo mejor que me pudo pasar en esta vida y espero estar ahí el resto de sus días para lo que me necesiten.

..... las amo con todo mi ser.

# Agradecimientos

Este trabajo se realizó con el apoyo de UNAM-DGAPA a través de los proyectos PAPIIT IG100818 y PAPIME PE116219, y de Toyota Motors Company.

A la UNAM, que a través de sus diversas instituciones (como la Facultad de Ingeniería), ha convertido mi experiencia como estudiante en la mejor que haya podido tener.

Mis más sinceros agradecimientos a mi director el Dr. Marco Antonio Negrete Villanueva, por todo el apoyo, conocimiento y sobre todo la gran paciencia que me brindó durante la realización de este trabajo, sin olvidar las extremadamente divertidas retas de futbolito.

A Fernanda Nova, Carolina Barrientos, Maira Morales y Roberto Salvador, por el apoyo incondicional desde hace muchos años, las aventuras y recuerdos inmemorables que recordaré por el resto de mi vida y su cariño incondicional. Por todo lo que nos falta por vivir y lo que me han hecho madurar desde que los conocí.

A Guillermo Pérez, Daniel Unda, Stefania Flores, Angélica Flores, Juan García y Abraham Manzano, por llenar cada uno de mis clases y días a lo largo de la carrera de risas, felicidad, aventuras y mucho conocimiento.

Al laboratorio de Biorrobótica en el cual he adquirido muchas habilidades y conocimientos que me serán de gran utilidad para mi futuro.

A cada uno de los colaboradores del laboratorio antes mencionado por que gracias al trabajo y esfuerzo de ellos es que podemos desarrollar diferentes sistemas debido al software ya generado. Me gustaría agradecer al Dr. Jesús Savage, por el apoyo brindado en el último año para la realización de diversas actividades que me han forjado como la persona que soy hoy. Al equipo Takeshi (Edgar Silva, Adrián Sarmiento, Edgar Vázquez, Oscar fuentes, Jesús Coyotzi, etc) por que gracias su paciencia, ayuda y consejos fue que pude involucrarme al mundo de la robótica y a su vez producir un trabajo más

completo, sin mencionar las grandes aventuras que hemos vivido en el último año y las grandes personas que son. A Manuel Pano por que desde el primer día que llegué al laboratorio me ha brindado su ayuda de diversas formas. Por último, quiero agradecer a Roberto Avilés no solo por ser la gran persona que es, sino también por la creación del material audio visual generado para este trabajo.

# Resumen

Los algoritmos tradicionales con los que se desarrollan tareas que realizan los robots móviles autónomos son efectivas, pero han mostrado ciertas limitaciones. Debido a esto, es necesario explorar ideas no tan convencionales si se quieren realizar tareas más complejas. El objetivo de este trabajo es aplicar los modelos psicológicos formales, que son modelos matemáticos que describen a fenómenos psicológicos, en el campo de la robótica. Esta idea, se muestra mediante la realización de 2 experimentos para evaluar su viabilidad. En el primero se aplica la Ley del Efecto (en la cual se fundamentó el condicionamiento instrumental) para enseñar a un robot (que fue construido para este experimento) a realizar un determinado comportamiento. Para el segundo experimento se implementó el Modelo de Contexto Generalizado en un sistema de reconocimiento de objetos ya existente, agregando información de la posición del robot con el fin de incrementar la tasa de reconocimiento del sistema antes mencionado.

Tras realizar el primer experimento, el robot logró realizar en repetidas ocasiones el comportamiento aprendido y se logró modificar la velocidad con la que aprende el robot manipulando unas constantes del modelo. Respecto al segundo experimento, los datos recabados muestran que logró una mejora en la tasa de reconocimiento del sistema, Por lo cual, se puede decir que la aplicación de modelos psicológicos formales es una herramienta útil en el área de los robots autónomos y si se aplica de la manera correcta puede ser usado para mejorar sistemas ya creados. Además, que los modelos psicológicos formales deberían ser considerados en el desarrollo de nuevos algoritmos, ya que están basados en como los seres vivos resuelven tareas que computadoras y robots aun nos son capaces de realizar.

# Abstract

Traditional algorithms for developing autonomous-mobile-robot tasks have proven their effectiveness, but also their limitations. Thus, implementing other ways is needed to achieve more complex behaviors in this kind of robot. The goal of this work is to take psychological formal models, which are mathematical models that psychological phenomena, and make use of them in robotics. This idea is shown by performing two different experiments, in order to evaluate the viability of the idea. The first experiment is a simple reinforcement learning experiment developed in a robot built for this project. It makes use of the Law of Effect, the idea where instrumental conditioning is based on, to teach a robot a specific behavior. The second experiment was implemented in an HSR robot. The Generalized Context Model was applied to an already used object recognizer system, by adding probabilistic information about the position of the robot to improve the recognition rate of this system.

After running the first experiment, the robot was able to repeat the behavior that was taught, and how fast the robot learns can be modified by the manipulation of some constant of the model. For the second experiment, data shows that the modification made to the system, improve the recognition rate of the system, So, it can be said that the application of formal psychological models in autonomous mobile robots is a useful tool that if it is correctly applied can be used to enhance the accuracy of the algorithms already developed. Furthermore, these models should be considered for developing new algorithms, because they are based on how living beings achieve tasks that computers and robots are not able to do by now.

# Índice general

<b>Dedicatoria</b>	<b>I</b>
<b>Agradecimientos</b>	<b>II</b>
<b>Resumen</b>	<b>IV</b>
<b>Abstract</b>	<b>V</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	3
1.2. Hipótesis . . . . .	4
1.3. Objetivos . . . . .	4
1.4. Descripción del documento . . . . .	5
<b>2. Antecedentes</b>	<b>6</b>
2.1. ¿Qué es un robot móvil autónomo? . . . . .	6
2.2. Paradigmas y primitivas de la robótica . . . . .	7
2.2.1. Primitivas . . . . .	7
2.2.2. Paradigma de la robótica . . . . .	7
2.3. Robótica basada en comportamientos . . . . .	9
2.4. Sensores y actuadores . . . . .	10
2.4.1. Actuadores . . . . .	10
2.4.2. Sensores . . . . .	10
2.5. Visión computacional para robots de servicio . . . . .	11
2.6. Modelos psicológicos formales . . . . .	12
2.6.1. La Ley del Efecto . . . . .	12
2.6.2. El Modelo de Contexto Generalizado . . . . .	14

<b>3. Construcción de un robot móvil autónomo</b>	<b>16</b>
3.1. Actuadores . . . . .	16
3.1.1. Motores de corriente continua . . . . .	17
3.2. Sensores . . . . .	22
3.2.1. Cámara . . . . .	22
3.2.2. Unidad de medición Inercial (IMU) . . . . .	24
3.3. Unidades de procesamiento . . . . .	25
3.3.1. Arduino . . . . .	25
3.3.2. Raspberry . . . . .	25
3.4. ROS . . . . .	26
3.5. OpenCV . . . . .	26
3.6. Conexión de los componentes . . . . .	27
<b>4. Implementación de comportamientos</b>	<b>29</b>
4.1. Segmentación por color . . . . .	29
4.1.1. Espacios de color . . . . .	30
4.1.2. Espacio RGB . . . . .	30
4.1.3. Espacio HSV y cómo transformar entre ambos . . . . .	31
4.1.4. Operadores morfológicos . . . . .	33
4.2. Control de los actuadores . . . . .	37
4.3. Interfaz gráfica . . . . .	39
4.4. Implementación de la Ley del Efecto . . . . .	41
<b>5. Visión con Modelo de Contexto</b>	<b>44</b>
5.1. Cálculo de la probabilidad de estar en un área . . . . .	45
5.1.1. Localización por filtro de partículas . . . . .	45
5.2. Reconocedor de objetos actual . . . . .	48
5.3. Reconocedor de objetos con <i>GCM</i> . . . . .	49
<b>6. Resultados</b>	<b>55</b>
6.1. Robots de este trabajo . . . . .	55
6.1.1. El robot de servicio HSR . . . . .	55
6.2. Comportamientos con Ley del Efecto . . . . .	56
6.3. Reconocimiento de objetos con <i>GCM</i> . . . . .	59
<b>7. Discusión</b>	<b>63</b>
7.1. Conclusiones . . . . .	63
7.2. Trabajo futuro . . . . .	64



VIII

*ÍNDICE GENERAL*

**Referencias**

**65**

# Índice de figuras

1.1. Autonomía vs complejidad . . . . .	2
2.1. Paradigma jerárquico . . . . .	8
2.2. Paradigma reactivo . . . . .	8
2.3. Paradigma híbrido . . . . .	9
2.4. Modelo en el centro . . . . .	13
3.1. Estator . . . . .	17
3.2. Rotor . . . . .	18
3.3. Esquema simplificado de las fuerzas . . . . .	18
3.4. Esquema de control del motor . . . . .	19
3.5. Puente H . . . . .	20
3.6. Ciclo de trabajo de una señal PWM . . . . .	21
3.7. Modelo de Pinhole . . . . .	23
3.8. Modelo de Pinhole equivalente . . . . .	23
3.9. Conexión de los elementos del robot . . . . .	27
3.10. Robot construido . . . . .	28
4.1. Cubo RGB . . . . .	31
4.2. Cono HSV . . . . .	32
4.3. Dilatación . . . . .	35
4.4. Erosión . . . . .	36
4.5. Apertura VS Cierre . . . . .	38
4.6. Interfaz Gráfica . . . . .	40
4.7. Interfaz Gráfica 2 . . . . .	40
5.1. Mapa del Laboratorio de Biorrobótica . . . . .	47
5.2. Mapa de celdas de ocupación . . . . .	51
5.3. Localización por filtro de partículas . . . . .	52

5.4.	Reconocedor de objetos actual . . . . .	53
5.5.	Reconocedor de objetos con GCM . . . . .	54
6.1.	Robot HSR . . . . .	56
6.2.	Probabilidades de visitar cada zona . . . . .	57
6.3.	Constante $K_r$ grande . . . . .	58
6.4.	Constante $K_r$ pequeña . . . . .	58
6.5.	Objetos usados para el sistema de visión . . . . .	60
6.6.	Objetos Similares . . . . .	60
6.7.	Distribución de los objetos . . . . .	61
6.8.	Matrices de confusión . . . . .	62

# Capítulo 1

## Introducción

El origen de la robótica como una disciplina formal se dio con la creación del robot Unimate, el cual era un brazo mecánico diseñado con el fin de asistir en la línea de producción de vehículos de General Motors, de esta forma surgió el primer robot industrial. La Federación Internacional de Robótica (IFR) refiere a la ISO para definir a un robot industrial como: manipulador controlado automáticamente, reprogramable y multipropósito, que puede ser programado en tres o más ejes y el cual puede ser fijo o móvil para uso en aplicaciones de automatización industrial (ISO).

Con el evento previamente mencionado se empezaron a desarrollar más robots en conjunto con otras disciplinas y se han logrado muchos avances, lo cuales se pueden abstraer en dos grandes ejes: la autonomía que el robot tiene para realizar las tareas sin necesidad de ayuda u operación humana y la complejidad del entorno en el que el robot se desenvuelve. Los primeros robots, como el antes mencionado, tenían un alto grado de autonomía ya que podían realizar las tareas que se deseaba, pero el ambiente en el cual se desenvolvían era controlado. Al desarrollarse otro tipo de robots como los de servicio, se incrementó la necesidad de responder a ambientes en los cuales pueden pasar diferentes eventos e incluso cambiar ciertas condiciones rápidamente (ambientes más dinámicos o complejos), lo cual lleva a crear un balance entre estas dos categorías según el propósito (autonomía y el ambiente en el que el robot estará operando) como lo ejemplifica Pineda et al. (2017) en la figura 1.1.

En los últimos años se han desarrollado robots que requieren un alto grado de autonomía y se desenvuelven en un ambiente muy complejo, como el robot “ASIMO”, que es un humanoide diseñado para funcionar en interiores

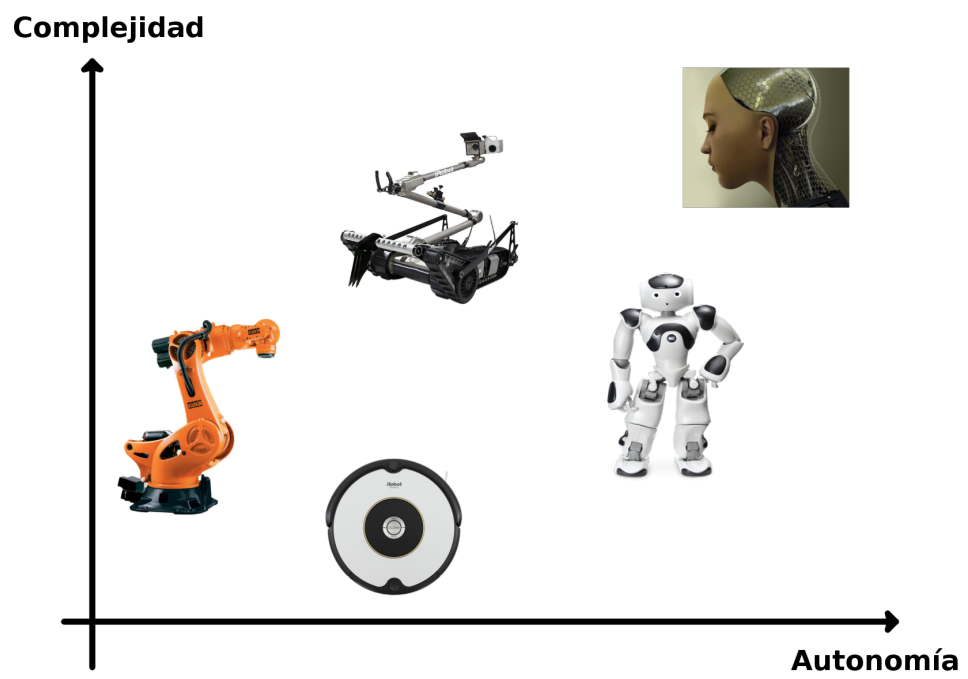


Figura 1.1: Autonomía vs complejidad

(Sakagami et al., 2002) y fue rentado a compañías como recepcionista. Otro ejemplo es el robot “Justina”, que es un robot de servicio en desarrollo para interactuar en ambientes complejos y realizar las tareas que una persona le pueda pedir (Savage, Martell, et al., 2018). Por último, “Takeshi” es otro ejemplo de robot de servicio doméstico cuyo hardware fue diseñado por la empresa Toyota y cuyo software está siendo desarrollado en la Facultad de Ingeniería de la UNAM (Savage, Vazquez, et al., 2018).

Estos robots llevan a cabo tareas que para los seres vivos son cotidianas, como la planeación de los pasos a seguir para alcanzar un objetivo, la forma de ir de un lugar a otro sin hacerse daño, o encontrar objetos que se necesiten entre otras tareas. Estos comportamientos que deben realizar los robots son de un alto nivel cognitivo, y es por esto que es conveniente revisar los estudios y hallazgos que las ciencias cognitivas han desarrollado. Las ciencias de la computación y la psicología cognitiva han cruzado caminos al grado que hay libros cuyo propósito es exponer los avances computacionales, matemáticos y psicológicos que han impactado en los avances científicos en estas áreas (Busemeyer, 2015). Mucho de este trabajo ha sido desarrollado a causa de la inteligencia artificial, ya que como mencionan Gershman, Horvitz, y Tenenbaum (2015), las ciencias cognitivas y neurociencias han compartido puntos de vista con los fundamentos de la inteligencia artificial y han promovido intercambios multidisciplinarios en cuestión de preguntas, métodos y resultados. Aunque se ha aplicado este enfoque en diferentes proyectos aún falta mucho por mejorar, un ejemplo es el de Banino et al. (2018), en el cual los autores usan redes neuronales y aprendizaje por reforzamiento para poder optimizar la navegación y se han lograron resultados prometedores, pero aún no tan buenos como los de los mamíferos.

## 1.1. Planteamiento del problema

La autonomía en un robot móvil, entendida como la capacidad de este de realizar una tarea sin intervención humana, es aún un problema abierto. La robótica basada en comportamientos es un paradigma que propone que un robot móvil puede ser autónomo mediante la interacción de múltiples pares entrada-salida o estímulo-respuesta llamados comportamientos. Por otro lado, la habilidad de reconocer objetos es importante para la autonomía de un robot debido a las diferentes tareas que debe de cumplir como la manipulación de objetos.

Los comportamientos de un robot móvil autónomo tradicionalmente son diseñados usando técnicas como máquinas de estados, redes neuronales y métodos reactivos como campos potenciales. En el área de visión computacional, los métodos para detección de objetos comúnmente utilizados son extracción de puntos característicos como: *Scale Invariant Feature Transform* (SIFT), *Speed up Robust Feature* (SURF), *Robust Independent Elementary Features* (BRIEF), *Oriented FAST and Rotated BRIEF* (ORB) (Karami, Prasad, y Shehata, 2017); segmentación por color, forma, agrupamiento, etc.

En este trabajo se abordarán estos dos problemas: en el diseño de comportamientos, se propone utilizar la Ley del Efecto para el desarrollo e implementación de éstos en un robot móvil; en el reconocimiento de objetos, se propone utilizar un modelo de categorización para mejorar el desempeño del sistema que se tiene implementado en los robots Justina y Takeshi del Laboratorio de Biorrobótica de la Facultad de Ingeniería.

## 1.2. Hipótesis

En este trabajo se considerarán las siguientes hipótesis:

- El uso de modelos psicológicos formales puede ayudar en el desarrollo de robots móviles autónomos.
- Se pueden generar comportamientos mediante la aplicación de modelos de aprendizaje como la Ley del Efecto.
- Se puede mejorar el reconocimiento de objetos mediante el uso de modelos de categorización.

## 1.3. Objetivos

- Construir un robot con los sensores y actuadores necesarios para la implementación de comportamientos con base en la Ley del Efecto.
- Aplicar el modelo formal de la Ley del Efecto en el diseño de comportamientos para un robot móvil autónomo.
- Evaluar el desempeño del robot móvil con el comportamiento basado en la Ley del Efecto.

- Aplicar el Modelo de Contexto Generalizado en la mejora de un sistema de visión activa para reconocimiento de objetos.
- Evaluar el desempeño del sistema de visión propuesto en un robot de servicio y compararlo con el sistema actual.

## 1.4. Descripción del documento

El contenido de este trabajo está organizado de la manera siguiente: En el capítulo 2 se presentan los conceptos teóricos y definiciones que son la base para la realización de este trabajo y se describen los modelos psicológicos formales que se implementaron. En el capítulo 3 se describen todos los componentes tanto de software como de hardware que se utilizaron para construir el robot autónomo móvil usado para la implementación de la Ley del Efecto y se muestra cómo es que estos elementos interactúan en conjunto. En el capítulo 4 se habla sobre el proceso que se siguió para la realización del experimento con base en la Ley del Efecto y como es que está fue implementada. En el capítulo 5 se describe la forma en que funciona el segundo experimento, en el cual se implementó una mejora del sistema de visión basada en el Modelo de Contexto Generalizado e información de la localización en un robot HSR fabricado por Toyota y se describen todos los elementos necesarios para esta implementación. Para finalizar este trabajo, se encuentran los resultados y conclusiones obtenidas con la realización de estos experimentos, así como la propuesta de trabajo futuro.



# Capítulo 2

## Antecedentes

En este capítulo se plantean los conceptos básicos para contextualizar los elementos de este trabajo e ideas que implementarán. Primero se habla de qué es un robot móvil autónomo y cuáles son sus funciones generales para dar paso dar una breve explicación de qué es lo que se usa para obtener información del ambiente y como afectarlo. Después se explica qué es la visión computacional y su importancia. por último, se habla sobre los modelos psicológicos formales y las ecuaciones que los describen.

### 2.1. ¿Qué es un robot móvil autónomo?

El concepto de robot es un concepto que en general es mal interpretado, ya que cuando se dice esta palabra, la primer imagen que viene a la cabeza es la de un robot con apariencia similar a la de un humano (extremidades, torso, cabeza, etc.), pero existen diferentes tipos de robots con objetivos, capacidades y formas diferentes. Para poder resolver la pregunta de ¿qué es un robot?, primero se hablará un poco de cómo surgió esta palabra, después se mostrarán diferentes acepciones de esta palabra y por último se presentarán algunos conceptos para comprender qué es un robot móvil autónomo.

La palabra robot se popularizó gracias a la obra checoslovaca Rossum's Universal Robots escrita por Karel Capek, en 1921, en la cual se usa la palabra "robota" y que se puede traducir como trabajo no especializado o de poca importancia. La Real Academia Española define a un robot como "máquina o ingenio electrónico programable que es capaz de manipular objetos y realizar diversas operaciones", la Organización Internacional de Estandarización

(ISO) lo define como un mecanismo con actuadores, que se puede programar en dos o más ejes con un grado de autonomía(ISO). La definición que se usará para este trabajo es la propuesta por Arkin (2000): un robot inteligente es un máquina capaz de extraer información del ambiente y usar el conocimiento que tiene del mundo para moverse de manera segura, significativa y de acuerdo con un propósito.

Un robot móvil es aquel robot que puede navegar bajo su propio control (ISO), esto quiere decir que es una base móvil, que como puede tener algún manipulador o no tenerlo. Por último, cuando se habla de autonomía se refiere a la capacidad de realizar una tarea sin intervención de un humano (ISO).

## 2.2. Paradigmas y primitivas de la robótica

### 2.2.1. Primitivas

Las primitivas de la robótica son las funciones en general que tiene un robot. Las más aceptadas son tres, sensado, planeación y actuación, como se muestra en Murphy (2000).

**Sensar:** es la parte en donde se procesa la información del ambiente del robot y se tiene como salida información útil.

**Planear:** es la parte donde la información de los sensores o del conocimiento que se tiene del mundo se usa para generar las tareas que se ejecutarán.

**Actuar:** Son las funciones donde se generan los comandos a los actuadores para realizar la tarea que se requiere.

### 2.2.2. Paradigma de la robótica

La RAE define paradigma como “teoría o conjunto de teorías que se aceptan sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento”, esto quiere decir que es una forma de abordar un problema. Diferentes modelos pueden mostrar mejores o peores resultados que otros dependiendo del enfoque o paradigma que usaron, pero en general ningún paradigma es incorrecto.

Los paradigmas de la robótica se diferencian por la forma en la cual interactúan las primitivas entre ellas, existen tres tipos de paradigmas: jerárquico, reactivo e híbrido.

### Paradigma jerárquico

El paradigma jerárquico es el más viejo, está inspirado en como se creía que el humano pensaba. Como primer paso se sensa, después se planea y hasta el último se ejecuta y se empieza de nuevo como lo muestra la figura 2.1. Para tomar decisiones se necesitan modelos globales del mundo donde el robot se desempeña e ignora las pruebas biológicas que muestran que la información sensada puede resultar directamente en una acción.

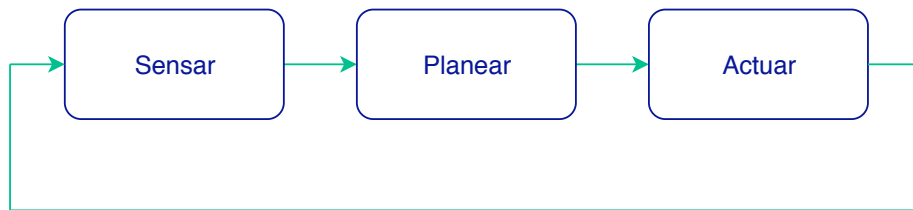


Figura 2.1: Paradigma jerárquico

### Paradigma reactivo

El paradigma reactivo surgió a raíz de los inconvenientes que causaba el paradigma jerárquico. En este modelo se elimina la sección de planeación, por lo tanto, cada acción dependerá de la información sensada. Entonces, si se empareja lo sensado con una acción, el robot solo podría realizar una tarea, pero si se tienen muchas de estas parejas, se obtiene al final una interacción que puede resultar en un comportamiento inteligente.

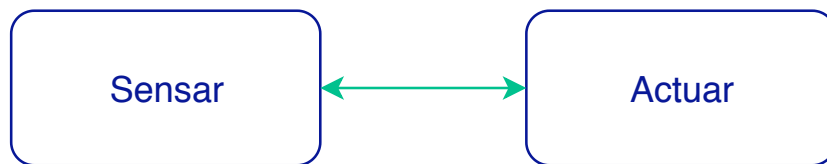


Figura 2.2: Paradigma reactivo

### Paradigma híbrido

El modelo reactivo trajo consigo ventajas como un tiempo de ejecución más corto, por ese motivo se usó como base del paradigma híbrido, en el cual el robot primero planea cuál es la mejor forma de descomponer las

tareas en subtareas y después cuáles comportamientos o parejas de “sensado-actuado” son las que mejor se ajustan para cada subtarea. Se puede decir que el paradigma híbrido es una combinación del jerárquico y reactivo.

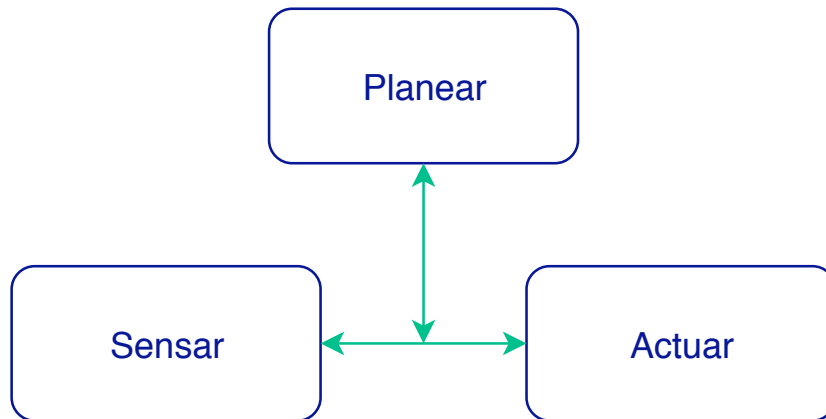


Figura 2.3: Paradigma híbrido

## 2.3. Robótica basada en comportamientos

Es otra forma de diseñar software para robots, en la cual se hace uso de diferentes módulos que interactúan constantemente entre sí, conocidos como comportamientos, que en su conjunto logran un cierto nivel de inteligencia. Para un observador externo, estos comportamientos solo son reacción a la interacción del medio ambiente y el robot. Desde el punto de vista de un programador, estos comportamientos son un conjunto de restricciones para lograr una meta.

Los comportamientos reciben señales de los sensores, pero también pueden recibir información de otro comportamiento; sus salidas pueden mandar señales a los actuadores o a otros comportamientos. Para programar este tipo de controles se construye una red estructurada de comportamientos interactuando entre ellos. Este tipo de control se adapta muy bien para ambientes en constante cambio, en los cuales se necesita una respuesta rápida y adaptación al medio, sin la necesidad de tener una completa abstracción del mundo real, pero con más capacidad para planear que un paradigma puramente reactivo.

Se puede empezar con comportamientos sencillos y al ir agregando más, se pueden generar comportamientos más complejos como seguir una pared, o un

objetivo cualquiera, encontrar un objeto, evadir luz, por mencionar algunos ejemplos.

## 2.4. Sensores y actuadores

### 2.4.1. Actuadores

Los actuadores son dispositivos de un robot que le permiten afectar el medio en el que se encuentran ejerciendo movimiento o generando fuerzas (Bekey, 2005). Por lo regular realizan una transformación de un tipo de energía a otra, debido a esto una forma de clasificarlos es:

**Neumáticos:** este tipo de actuadores transforman la energía del aire que se comprime en un depósito creando una presión que genera una fuerza sobre algún elemento mecánico. Se usan como cilindros neumáticos para producir movimientos lineales en procesos de automatización y como motores rotacionales, que tienen una gran velocidad, pero entregan un par bajo.

**Hidráulicos:** son muy parecidos a los actuadores neumáticos, pero la principal diferencia radica en el fluido que usan, en los actuadores neumáticos se usa aire y en los hidráulicos se hace uso cualquier otro tipo de fluido, un aceite por lo regular. Al compararlos, el actuador hidráulico tiene una relación potencia/peso mayor a la salida, sin embargo, su instalación se vuelve más compleja porque se manejan presiones muy altas y estas aumentan el riesgo de fugas.

**Eléctricos:** estos usan la corriente eléctrica como fuente de energía y son los más usados, debido a que los otros tipos de actuadores requieren de un compresor para su funcionamiento. Son muy versátiles y su operación o tiempo de respuesta es más rápido en comparación con los otros dos, además de su sencilla instalación. Los ejemplos más usados en robótica son los motores de corriente continua, servomotores y motores a pasos.

### 2.4.2. Sensores

De acuerdo con Solutions (2007), un sensor es un dispositivo de entrada que adquiere una medida de una variable física. La señal que entrega puede ser fácilmente convertida a otra representación eléctrica.

Algunas formas de clasificarlos son:

- Propioceptivos y exteroceptivos

- Pasivos y activos

**Propioceptivos:** son sensores que miden variables físicas relacionadas con el estado interno del robot.

**Exteroceptivos:** son aquellos que toma información del ambiente externo del robot.

**Activos:** estos sensores emiten energía para realizar la medición.

**Pasivos:** son sensores que no requieren emitir energía para realizar la medición.

Una fotorresistencia cae en la categoría de pasiva, porque varía su resistencia dependiendo de la intensidad luminosa que recibe y es exteroceptivo porque la fuente de luz no pertenece al robot, sino a algún lugar cercano. Si se habla de un sensor óptico de herradura, este es activo ya que emite una luz infrarroja (a través de un led de luz infrarrojo) que capta gracias a un fototransistor. EL sensor herradura se cataloga como propioceptivo, porque su principal uso en la robótica es en los encoders, los cuales se usan para medir el ángulo de giro de un motor.

Las cámaras se catalogan como exteroceptivas, ya que son usadas para obtener información sobre colores o formas que no pertenecen al robot. Sin embargo, una cámara tradicional (RGB) es pasiva, ya que la luz que capta es emanada por el ambiente, pero una cámara RGB-D es un sensor activo ya que tiene que emitir luz para poder adquirir la nube de puntos del ambiente.

## 2.5. Visión computacional para robots de servicio

La visión en términos biológicos para un humano es la transformación de una sensación visual en una percepción visual y al proceso análogo para las computadoras se le conoce como visión computacional (Grigorescu, 2010). Esta desarrolla técnicas para que una computadora pueda obtener información de una escena y es usada para resolver diferentes tareas en diversas áreas como puede ser en la industria de manufactura, sistema de seguridad, el área médica y por supuesto en robots de servicio. Para la última área, tiene muchas aplicaciones como: identificar caras, identificar algunos gestos para indicaciones no verbales, en navegación se puede usar para estimar la posición de un robot como lo hizo Sarmiento et al. (2019) y para detección,

reconocimiento y localización de objetos que generalmente dan paso a tareas de manipulación.

En competencias donde se evalúan las capacidades de los robots de servicio existen tareas donde son necesarias la implementación de estos tipos de algoritmos, un ejemplo es la RoboCup donde en su libro de reglas (Matamoros et al., 2019) existen diversas pruebas que involucran visión computacional. Un ejemplo es la prueba *storing groceries*, en la cual se deben de detectar objetos de una mesa y colocarlos en la repisa de un estante que corresponde a la categoría del objeto manipulado.

Como parte de este trabajo se propone mejorar un sistema de visión agregando información obtenida del sistema de navegación mediante el Modelo de Contexto Generalizado que está descrito en el capítulo 5.

## 2.6. Modelos psicológicos formales

Para poder definir que es un modelo psicológico formal, primero se debe de entender que es un modelo formal. Este concepto proviene de las ciencias de la computación y Busemeyer (2015) lo define como un modelo que usa métodos matemáticos y estadísticos, lógica formal, o simulación por computadora para representar algún fenómeno, situación, o sistema en el mundo real. Por lo cual, se puede decir que un modelo psicológico formal es obtener un modelo que explique algún fenómeno psicológico. Este tipo de modelos han sido una herramienta muy poderosa para poder integrar las teorías de la psicología cognitiva y la neurociencia cognitiva Rodríguez Villagra (2014).

Un ejemplo de la importancia de esta herramienta se puede observar en la figura 2.4 tomada de Forstmann, Wagenmakers, Eichele, Brown, y Serences (2011), en el cual presentan la relación que tienen diferentes áreas que tienen como objeto de estudio la cognición humana y su objetivo es obtener conclusiones de los procesos cognitivos. Donde hablan de un enfoque llamado el “modelo en medio”, mostrado en la figura 2.4, en donde concluyen que está relación dada por los modelos formales acelera la búsqueda de explicaciones de los procesos cognitivos.

### 2.6.1. La Ley del Efecto

Es una ley que nació de una hipótesis empírica para poder explicar la capacidad de adaptación y aprendizaje de los animales que otras teorías no

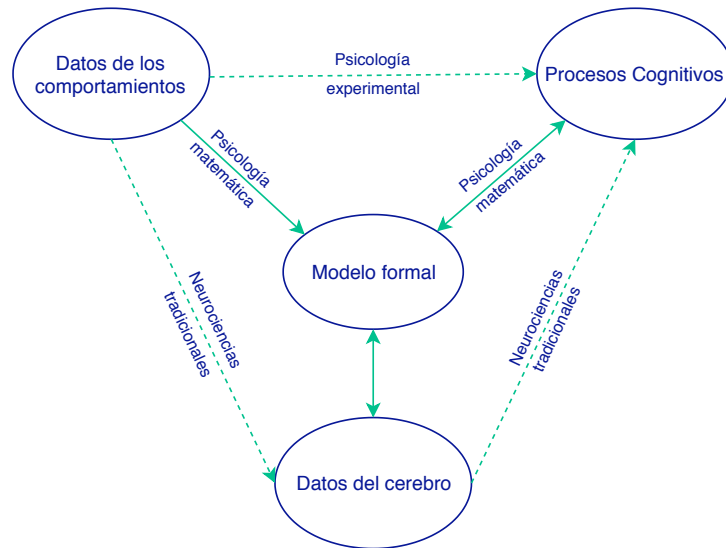


Figura 2.4: Modelo en el centro

podían explicar (Thorndike, 1927). Fue propuesta por Edward L. Thorndike a finales del siglo XIX y esta plantea “que si una respuesta (R) dada en presencia de un estímulo (E) es seguida de un evento satisfactorio, la asociación entre el estímulo y respuesta (E-R) es fortalecida. Si la respuesta es seguida de un evento molesto, la asociación E-R se debilita” (Domjan, Grau, Krause, Ortiz Salinas, y Avila Santibáñez, 2010). En otras palabras, en la asociación de un E-R se tiene que estampar o reforzar (ya sea positiva o negativamente) para provocar que se repita o no se repita este comportamiento o asociación E-R.

Con estos trabajos experimentales y la publicación de la Ley del Efecto creó una nueva corriente de condicionamiento conocida como condicionamiento instrumental o dirigido a metas, en la cual se cree que la conducta se produjo por que anteriormente esa misma conducta ayudó a producir determinadas consecuencias. Sus experimentos consistían en el diseño de alguna caja con un mecanismo que un animal hambriento pudiera activar para escapar de ella y poder obtener la comida que estaba fuera de la caja. Estos experimentos originalmente estaban encaminados a entender “inteligencia de los animales”, y lo que medía era el tiempo en que el sujeto (animal) tardaba en salir, a lo que se denomina latencia, y observó que conforme se repetía el experimento la latencia era menor. A esta inteligencia no se le dio una



interpretación antropomórfica, dado que la latencia menor no se atribuía a que el sujeto hubiera comprendido la tarea necesaria para escapar, sino que era el reflejo del aprendizaje de una asociación del E-R (Domjan et al., 2010).

Con el nacimiento de esta corriente, se empezaron a realizar más experimentos en los que en vez de escapar de una caja, ahora se pretendía que ratas llegaran de un punto o caja inicial a una final de un laberinto para poder medir algunos otros parámetros y ver cómo el ambiente los afecta (procedimiento de ensayo discreto) y con esto se dio pie los procedimientos de condicionamiento operante libre que se le atribuyen a Burrhus Frederic Skinner.

En este trabajo se usó el modelo matemático (4.12) propuesto por Staddon (2014) para lograr que un robot tuviera un comportamiento deseado (acercarse a una marca deseada por el usuario) como se explica en la sección 4.4.

### 2.6.2. El Modelo de Contexto Generalizado

Categorizar o clasificar hace referencia al proceso de organizar experiencia sensorial en grupos (Pothos y Wills, 2011), se considera como un tema de estudio de la psicología cognitiva, ciencias cognitivas y neurociencias cognitivas y se relaciona con muchos procesos cognitivos como el aprendizaje, adquisición y reproducción del lenguaje, toma de decisiones y razonamiento inductivo. Para poder describir este fenómeno se han creado modelos formales que son teorías que permiten predicciones cuantitativas sobre el comportamiento de categorización de los participantes (Pothos y Wills, 2011).

El Modelo de Contexto Generalizado o GCM (por sus siglas en inglés), es un modelo de caracterización supervisada que fue propuesto Robert Nosofsky en 1984. El GCM representa a cada modelo como un punto en un espacio psicológico multidimensional y la similitud con el modelo es una función de distancia entre los objetos en este espacio (Busemeyer, 2015). Este modelo se basa en que los individuos guardan una serie de ejemplos para representar una clase, es decir, para representar la categoría “perro” se tiene una colección grande de diferentes razas y si el objeto tratando de clasificar se parece lo suficiente el individuo lo clasificaría como perro, aunque no fuera una raza

que el conociera. Esto se representa con la ecuación:

$$d_{ij} = \left[ \sum_{k=1}^K w_k |x_{ik} - x_{jk}|^\rho \right]^{\frac{1}{\rho}} \quad (2.1)$$

donde por cada característica que se va a comparar ( $k$ ), se obtiene la diferencia con los modelos almacenados ( $x_j$ ). Se le puede dar prioridad a la característica que se está evaluando con la constante ( $w_k$ ) y a esto se le asocia una distancia.

Se calcula una similitud dada por la ecuación 2.2, con la que se obtiene una función continua decreciente y para cada distancia previamente obtenida. El parámetro  $c$  sirve para afectar qué tanto la similitud es disminuida o aumentada por la distancia, es decir, si  $c$  es grande la similitud decae rápidamente con el aumento de distancia y viceversa.

$$s_{ij} = e^{-cd_{ij}} \quad (2.2)$$

El modelo indica que estas similitudes se almacenan con un peso de memoria, que hace referencia a que tanto se ha presentado este objeto recientemente como se muestra en la siguiente ecuación:

$$a_{ij} = m_j s_{ij} \quad (2.3)$$

Con lo que, cambiando este valor de memoria, se puede lograr que un objeto tienda activarse con mayor intensidad que el resto. Para este trabajo se propone calcular el valor de memoria a partir de información obtenida de la localización del robot, es decir, en el momento en que el robot recupere los ejemplos (objetos entrenados) estos se ponderaran con la información de la ubicación del robot y del objeto (ver en 5.1).

En el siguiente capítulo se describen los componentes de hardware y software que se usaron para la construcción y programación de un robot móvil en el que se implementaron comportamientos con base en la Ley del Efecto y en el capítulo 5 se describe la mejora del sistema de visión de un robot de servicio usando el Modelo de Contexto Generalizado.

# Capítulo 3

## Construcción de un robot móvil autónomo

En este capítulo se describen los sensores y actuadores empleados en la construcción del robot móvil autónomo. Se comienza con el motor de corriente directa, su principio de funcionamiento y el modelo que describe la velocidad en función del voltaje en sus terminales. Se presentan la cámara, incluyendo un modelo simple de cómo se obtiene la imagen, la unidad de medición inercial y los elementos que la componen. Se describen las unidades de procesamiento utilizadas, sus capacidades y por qué fueron elegidas. Al final, se presenta la forma en que interactúan todos estos componentes.

### 3.1. Actuadores

Como se menciona en el capítulo 2.4.1, gracias a los actuadores los robots pueden generar movimiento, en la robótica móvil los actuadores más comunes son los motores. Hay diferentes tipos de motores como: los servomotores, que son motores de corriente continua con un conjunto de engranes y un sistema de control de posición; motores a pasos, los cuales mediante pulsos en un orden se genera un campo que mueve el inducido cierto número de grados a la vez. Para este trabajo se decidió usar motores de corriente continua, porque para los comportamientos que se desarrollaron y considerando que es una base diferencial, este tipo de motores ofrecían la manera más fácil de controlar la velocidad de giro.

### 3.1.1. Motores de corriente continua

Este tipo de motores se componen de una armadura donde se generan campos magnéticos fijos conocida como estator. Dentro de esta armadura está un rotor, en el cual se inducen las fuerzas electromagnéticas que lo hacen girar.

En el estator, también conocido como el inductor, hay una armadura redonda hecha de un material ferromagnético llamada yugo, a este van atornillados los polos, que pueden ser imanes permanentes y enrollados a los polos unas bobinas por las cuales circulará la corriente que genera el campo magnético y con los polos alternados se producen polaridades norte y sur.

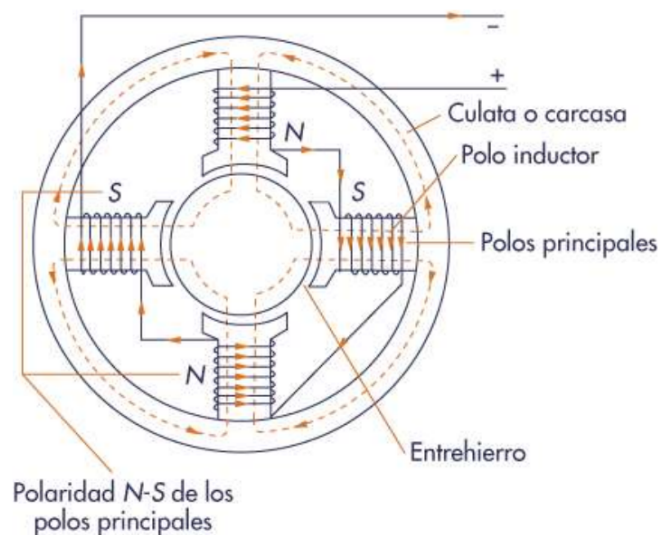


Figura 3.1: Estator

El rotor, también conocido como inducido, es un eje en el cual se enrollan un conjunto de bobinas de filamento de cobre que se van conectando el inicio de una con el final de la otra. En el colector del motor están las delgas, que son placas metálicas aisladas una de la otra y que esta conectadas a una de las bobinas del inducido. Por último, están las escobillas que están hechas de grafito y ellas tienen contacto directo con el colector.

### 18CAPÍTULO 3. CONSTRUCCIÓN DE UN ROBOT MÓVIL AUTÓNOMO



Figura 3.2: Rotor

Conociendo las partes que conforman al motor de corriente continua, se puede entender el funcionamiento en conjunto. El estator genera un campo magnético que, aunque está alternado en sus polos se mantiene constante y dentro del estator se encuentra el rotor que a través de las escobillas hace contacto con el colector dejando fluir una corriente por sus embobinados. Esta corriente al pasar por el campo magnético genera una fuerza perpendicular y como resultado se obtiene el giro en el rotor hacia un sentido dependiendo de la dirección de la corriente, como se muestra en el siguiente esquema:

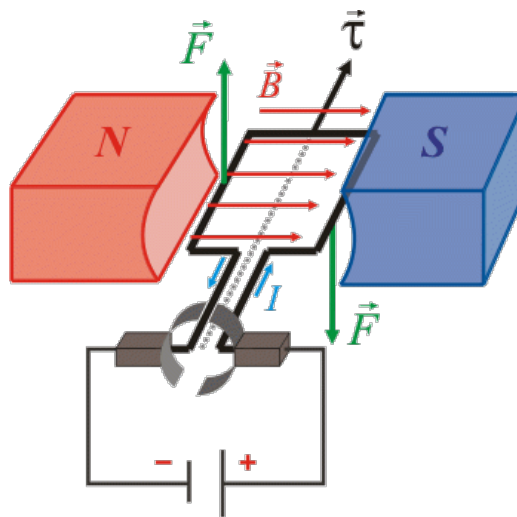


Figura 3.3: Esquema simplificado de las fuerzas

Ahora que se sabe cómo es que funciona el motor, se puede obtener un modelo matemático partiendo del siguiente diagrama (Ogata, 2004):

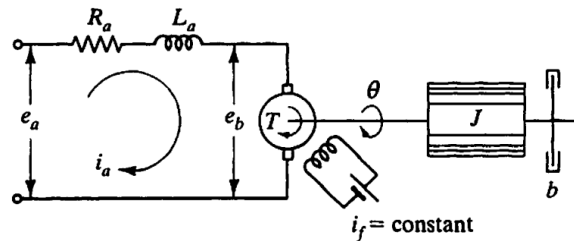


Figura 3.4: Esquema de control del motor

donde:

- $R_a$  = resistencia de la armaduta [ $\Omega$ ]
- $L_a$  = inductancia de la armaduta [H]
- $i_a$  = corriente de la armaduta [A]
- $i_f$  = corriente del campo [A]
- $e_a$  = voltaje aplicado a la armaduta [V]
- $e_b$  = fuerza contra-electromotriz [V]
- $\Theta$  = desplazamiento angular del eje del motor [rad]
- $T$  = Par generado por el motor [N m]

El par generado por el motor es proporcional al producto de la corriente de la armadura y al flujo del aire que a su vez es proporcional a la corriente del campo, dicho de otra forma:

$$\Psi = K_f i_f \quad (3.1)$$

donde  $k_f$  es una constante, por lo tanto, el par se puede escribir de la siguiente forma

$$T = K_f i_f K_1 i_a \quad (3.2)$$

para una corriente constante del campo el flujo se vuelve constante y el par es directamente proporcional a la corriente

$$T = K i_a \quad (3.3)$$

donde  $K$  es una constante del motor. Se observa que si el signo (sentido) de la corriente  $i_a$  se invierte, el par resultará opuesto y por lo tanto el rotor girará hacia la dirección contraria.

## 20CAPÍTULO 3. CONSTRUCCIÓN DE UN ROBOT MÓVIL AUTÓNOMO

Cuando la armadura está girando, el voltaje es proporcional al producto del flujo y la velocidad angular inducida en la armadura. Para un flujo constante, el voltaje inducido es directamente proporcional a la velocidad angular

$$e_b = K_b \frac{d\theta}{dt} \quad (3.4)$$

donde  $e_b$  es la fuerza contra-electromotriz y  $k_b$  es la constante de fuerza contra-electromotriz.

Dado el modelo anterior, se sabe que la corriente es directamente proporcional al par que entregará el motor y de igual manera el voltaje respecto a la velocidad. También que si se cambia la polarización el motor girará en sentido opuesto, pero estas velocidades normalmente son muy altas y los pares un poco bajos, así que se le agrega una reducción de engranes, de radios pequeños a radios más grandes con el propósito de aumentar el par de salida.

### Puente H

Para cambiar la polaridad del motor se tienen que cambiar físicamente los cables, así que se creó un dispositivo que permite cambiar la polaridad, si se hace uso de un microcontrolador solo se debe poner las señales correspondientes en alto o bajo (tabla 3.1). Como medio principal, usa un dispositivo que puedan servir como switch, así que se puede usar relevadores, transistores, diodos, etc. En la figura 3.5 se observa el ejemplo de este circuito implementado con transistores.

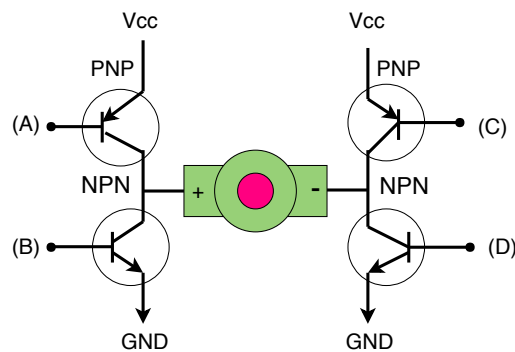


Figura 3.5: Puente H

A	B	C	D	Función
1	0	0	1	Adelante
0	1	1	0	Reversa
1	1	0	0	Frenar
0	0	1	1	Frenar

Tabla 3.1: Combinación de las entradas al puente H

Para regular la velocidad del motor se debe de poner a las entradas (A, B, C o D) una señal con un cierto voltaje regulado, pero se usa una técnica conocida como modulación de ancho de pulso (PWM, por sus siglas en inglés), que en vez de regular el voltaje que se tiene de entrada, prende y apaga la señal rápidamente, así que el tiempo que está activado el circuito es menor o mayor y con esto se obtienen diferentes velocidades. Esto se recrea mediante una señal cuadrada con un periodo constante y lo que se varía es el ciclo de trabajo de la misma.

El ciclo de trabajo de una señal es el tiempo que una señal está activa (o en alto), así que una señal cuadrada con ciclo de trabajo del 100% estaría prendida todo el tiempo, una del 50% la mitad del tiempo prendida y la mitad apagada como se muestra en la siguiente imagen:

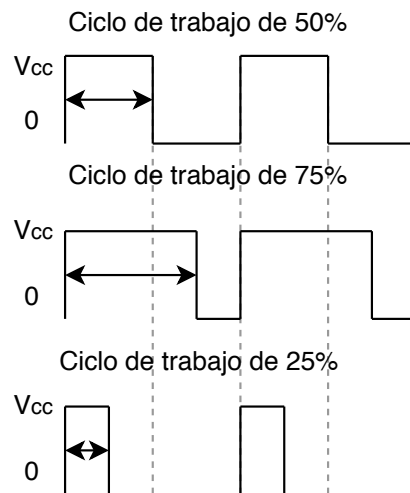


Figura 3.6: Ciclo de trabajo de una señal PWM



## 22CAPÍTULO 3. CONSTRUCCIÓN DE UN ROBOT MÓVIL AUTÓNOMO

El voltaje promedio de este tipo de señales, como se puede observar, solo depende del tiempo que esta activa la señal( el ciclo de trabajo), por lo tanto se puede calcular de la siguiente forma:

$$V_{prom} = V_{cc} * Ciclo\_de\_trabajo \quad (3.5)$$

### 3.2. Sensores

Para que el robot pueda adquirir información del medio en el que se encuentra hace uso de los sensores, como se menciona en el capítulo 2.4.2. Para este robot se usarán en particular 2 tipos: una cámara, una unidad de medición de inercial.

#### 3.2.1. Cámara

Para poder usar visión computacional necesitamos un dispositivo que obtenga información de la luz del ambiente, esta es emanada por una fuente y absorbida por los objetos en el espacio. La que no llega a ser absorbida es rebotada y lo percibimos como el color. Con el modelo de Pinhole, que es un modelo muy simplificado para entender como es que se proyecta la luz, se explicará cómo es que esta luz pasa por un pequeño orificio imitando la función de nuestra retina.

El modelo Pinhole, como se explica en Bradski y Kaehler (2011), es una pared imaginaria que no deja pasar ningún rayo de luz excepto los que pasan a través del orificio central. El haz de luz que entra proviene de algo en el ambiente o un objeto distante, pero este proviene de un cierto punto. En una cámara de Pinhole real el punto es proyectado a otra superficie. Como resultado se tiene un plano proyectado y el tamaño de este plano o imagen es relativo a la distancia del objeto y el parámetro de la cámara conocido como distancia focal. Considere el diagrama de la figura 3.7, donde  $f$  es la distancia focal de la cámara,  $Z$  la distancia entre el objeto y la cámara,  $X$  el tamaño del objeto y  $x$  el tamaño del objeto proyectado.

De la imagen 3.7 se puede ver que por semejanza de triángulos,  $-x/f = X/Z$  o escrito de otra manera:

$$-x = f \frac{X}{Z} \quad (3.6)$$

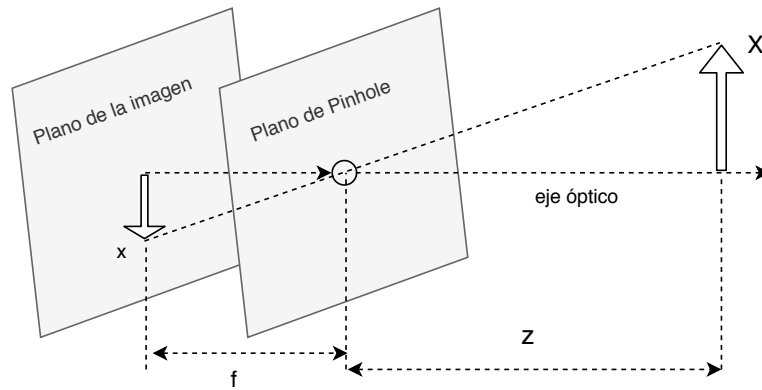


Figura 3.7: Modelo de Pinhole

Intercambiando el plano de la imagen con el orificio, las operaciones resultan más sencillas y la imagen proyectada no queda de cabeza, como se muestra en la figura 3.8. En la cual podemos observar que al orificio ahora se le llama centro de proyección y que el eje óptico queda referenciado al punto principal. Por lo tanto, la relación entre los triángulos está definida por  $x/f = X/Z$ .

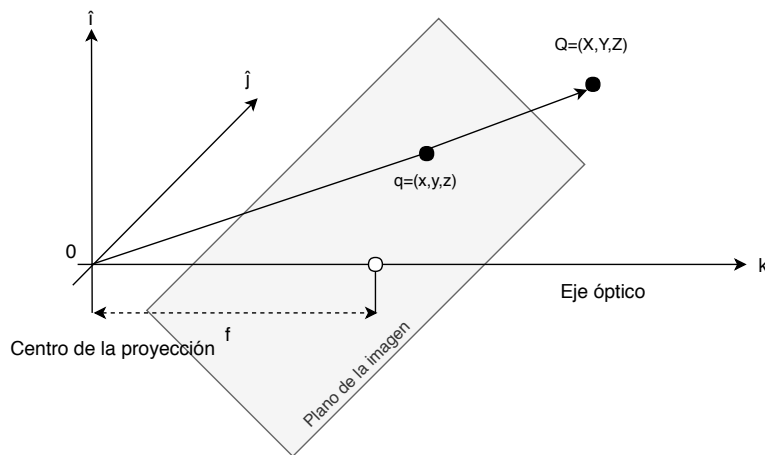


Figura 3.8: Modelo de Pinhole equivalente

Como muestra el diagrama 3.8, se puede observar que el orificio principal estaría en el centro del sensor, pero esto implicaría que deberían de estar colocados con una precisión muy alta, por lo tanto, se añaden dos nuevos

## 24CAPÍTULO 3. CONSTRUCCIÓN DE UN ROBOT MÓVIL AUTÓNOMO

parámetros  $c_x$  y  $c_y$ . Estos corresponden a un posible desplazamiento del centro del plano de la proyección con respecto del eje óptico. Con lo cual se obtienen un modelo en el cual un punto del mundo físico con coordenadas  $(X, Y, Z)$  es proyectado a un pixel de acuerdo con las siguientes ecuaciones:

$$x_{pantalla} = f_x \left( \frac{X}{Z} + c_x \right), y_{pantalla} = f_y \left( \frac{Y}{Z} + c_y \right) \quad (3.7)$$

Es importante mencionar que en las ecuaciones se tienen dos diferentes parámetros de la distancia focal, ya que usualmente el sensor es rectangular y no cuadrado.

La cámara utilizada en este robot es una RaspiCam v2, que tiene una resolución de 3280x2464 pixeles, con una distancia focal de 3.04mm, su campo de vista horizontal es de 62.2 grados y el vertical es de 48.8 grados. En video puede tomar alta definición (1080p30).

### 3.2.2. Unidad de medición Inercial (IMU)

Son dispositivos usados para navegación cuando no se tienen referencias externas, ya que gracias a los dispositivos que contiene calcula el cambio de posición con respecto a su alineación inicial. Las IMU's se componen típicamente de un acelerómetro, que como su nombre lo dice, mide aceleraciones, no solo las que se experimentan al tener un cambio de velocidad y posición, sino también las que experimentan los cuerpos por acción de la gravedad. Otro elemento típico es el giroscopio, que detecta las velocidades a las que giran los cuerpos. Un elemento no tan común en el diseño de este tipo de sensores es el magnetómetro, este dispositivo mide los campos magnéticos a los que se encuentra sometido, puede tener muchas interferencias ya que todos los dispositivos electrónicos que tenga cercanos generarían un campo magnético que lo afectaría, pero es muy útil para ubicar el norte magnético de la tierra (usarlo como brújula).

Estos tres dispositivos, puede medir las magnitudes antes mencionadas, pero usualmente lo hace respecto a uno, dos o tres ejes. El sensor utilizado en este robot (LSM9DS1) tiene estos tres dispositivos con tres ejes, la aceleración la mide en [g], la velocidad de giro en DPS (grados por segundo) y el campo magnético en Gauss. Tiene una interfaz I2C para él envió de los datos y un tamaño muy compacto (0.9 in X 0.4 in).

## 3.3. Unidades de procesamiento

### 3.3.1. Arduino

Es una plataforma abierta creada para el manejo sencillo de hardware y software (<https://www.arduino.cc/en/Guide/Introduction>). En principio fue un proyecto desarrollado en el Instituto de Diseño de Interacción de Ivrea, Italia, como una herramienta de prototipado rápido que ayudaba a los estudiantes sin un gran conocimiento o experiencia en programación o en electrónica. Pero a través de los años y del amplio uso que se le dio se fue adaptando a nuevas necesidades y retos, ofreciendo productos para tecnología IoT, manufactura aditiva y sistemas embebidos entre otras.

Para comprender Arduino se tiene que abordar desde dos puntos de vista. Desde el hardware es una tarjeta de bajo costo que contiene un microprocesador programable y tiene la capacidad de leer y generar señales (ya sean digitales o analógicas) para la lectura de sensores o la generación de una respuesta con los actuadores. Desde el enfoque del software, brinda un entorno de desarrollo integrado (IDE, por sus siglas en inglés) que permite la programación del microprocesador de una manera simple en un lenguaje parecido a C y si se quiere, se puede hacer uso de bibliotecas de C++. El IDE puede usarse en diferentes sistemas operativos lo cual lo hace accesible para muchas personas.

Se hizo uso de esta herramienta para la lectura y procesamiento de señales de sensores, así como para la generación de señales para el control de los motores del robot.

### 3.3.2. Raspberry

Es una computadora de bajo costo del tamaño de una tarjeta de crédito, que se puede conectar a una pantalla o monitor y a un ratón y teclado estándar (<https://www.raspberrypi.org/help/what-is-a-raspberry-pi>). Fue creada en el Reino Unido por una fundación del mismo nombre, con el propósito de acercar a los jóvenes a la programación. Al ser una pequeña computadora de propósito general y de pequeño tamaño, lo que la hace portable, se ha usado por todo el mundo en diferentes tipos de proyectos. Tiene algunos pines de entrada y salida para generar y leer señales y se le pueden conectar diferentes tipos de periféricos.

Para este robot, la Raspberry Pi 3 se usó como unidad principal de proce-

samiento, dado que su capacidad cubre las necesidades y, como se mencionó, por su tamaño es ideal para estar en el robot, además de que hace uso de sus dispositivos de red inalámbrica, que permiten el monitoreo y envío de ordenes sin tener la necesidad de estar conectados por un cable.

### **3.4. ROS**

Es un meta sistema operativo de código abierto para robots (<http://wiki.ros.org/ROS/Introduction>), que ayuda a la creación de software mediante una colección de bibliotecas y convenciones que permiten usar dispositivos para el control a bajo nivel, enviar mensajes entre códigos y proporciona herramientas para escribir, compilar y correr el código creado. ROS nació gracias al esfuerzo de muchos investigadores que tenían la necesidad de crear una infraestructura de colaboración abierta. En la universidad de Stanford ya se tenían proyectos encaminados a lograr esta meta, pero no fue hasta el 2007 que una incubadora interesada en robótica aportó recursos y gracias a la comunidad de investigadores que ayudó, se creó ROS.

Al ser una creación de investigadores, la infraestructura está diseñada para poder reusar código y por su estructura de paquetes, también es fácil de compartirse. En este proyecto es usado como herramienta principal en el desarrollo del software, porque permite comunicar diferentes dispositivos que generan las señales o reciben las lecturas de los sensores (el bajo nivel) con los algoritmos de alto nivel en el cual se toman las decisiones.

### **3.5. OpenCV**

Es una librería de código abierto para visión computacional y aprendizaje de máquina (<https://opencv.org/about.html>), que brinda una estructura para la creación fácil de aplicaciones de visión computacional, disponible para diferentes sistemas operativos y varios lenguajes de programación como, C++, Python y Java. Está escrita en C y C++, además estar muy optimizada y de poder tomar ventaja de computadoras con multinúcleos.

Algunas de sus aplicaciones son: reconocimiento de objetos y caras, clasificación acciones de humanos, rastreo de objetos en movimiento, entre otras. En este proyecto OpenCV fue utilizado para poder adquirir datos con la cámara RaspPiCam, crear un rastreador de color (color tracker) y obtener

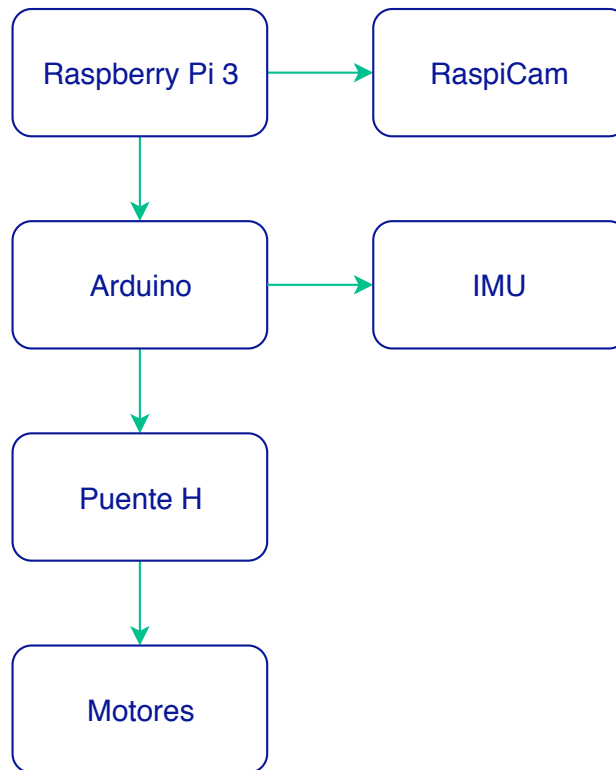


Figura 3.9: Conexión de los elementos del robot

información sobre los centroides de las marcas de colores.

### 3.6. Conexión de los componentes

La forma en la que se conectaron los componentes que conforman al robot es la siguiente: los motores se conectan al puente H, en las entradas del puente se conectan 2 señales PWM por motor provenientes del Arduino. La IMU es conectada al Arduino, esta conexión se realiza por el puerto I2C. El Arduino se conecta por medio del puerto serial a la Raspberry Pi 3, en la cual, haciendo uso de ROS, se envía información de las velocidades de los motores y se recibe la información de los sensores. A la Raspberry también está conectada la cámara (Raspicam) por medio del puerto CSI que tiene la tarjeta para esta cámara, el diagrama se muestra en la figura 3.9.

El robot físicamente conectado se puede observar en la ilustración 3.10,

### 28CAPÍTULO 3. CONSTRUCCIÓN DE UN ROBOT MÓVIL AUTÓNOMO

donde se pueden apreciar los componentes físicos descritos en esta sección.

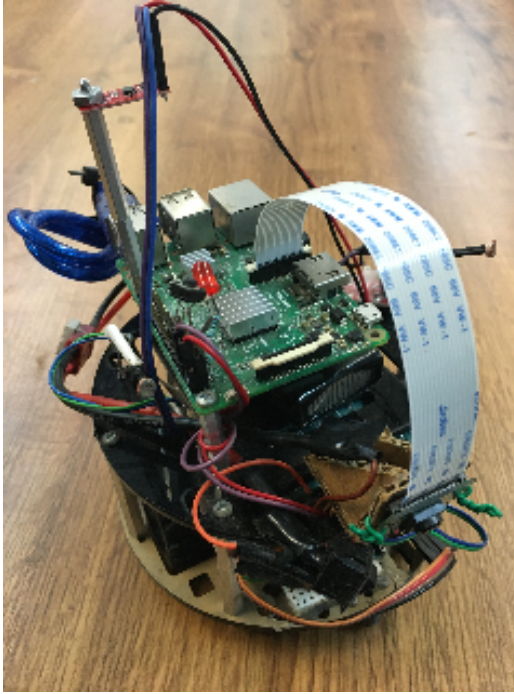


Figura 3.10: Robot construido

Conociendo los componentes del robot y como es que interactúan entre ellos, en el siguiente capítulo se hablará de cómo se programaron los nodos para la implementación de los comportamientos en el robot.

# Capítulo 4

## Implementación de comportamientos con base en la Ley del Efecto

En este capítulo se explica todo el software desarrollado para implementar la Ley del Efecto, empezando por explicar que es un segmentador de color y los pasos que se llevan a cabo como la transformación de espacio de color y el uso de ciertos operadores morfológicos. Se presentan las ecuaciones de control para poder mover al robot al punto deseado, la interfaz gráfica que se desarrolló para la interacción del humano con el robot al realizar los experimentos y para concluir como es que se implementó la Ley del Efecto descrita en el capítulo 2.6.1.

### 4.1. Segmentación por color

La segmentación, en el procesamiento de imágenes, se encarga de aislar partes de la misma, ya que dependiendo del propósito se necesita solo encontrar las partes que aportan información deseada. En el caso de este trabajo la información es el color, el cual es una característica muy poderosa para identificación y extracción de objetos en una imagen (Bradski y Kaehler, 2011). Se puede decir que el procesamiento de imágenes se puede dividir en dos grandes áreas: el procesamiento a todo color, que regularmente se refiere a la adquisición de la imagen gracias a un dispositivo o sensor como cámaras o scanners, y un procesamiento de pseudocolor, prácticamente todo el proce-



samiento digital se realiza a este nivel y consiste en asignar un valor a cierta intensidad o lecturas de estos sensores.

### **4.1.1. Espacios de color**

El proceso por el cual los humanos perciben e interpretan el color aún no es completamente entendido, se tienen varios experimentos de los cuales se pueden tener bases teóricas. Los colores, que son percibidos por humanos y por algunos animales, dependen de la naturaleza de la luz reflejada por el objeto, esta luz corresponde a una longitud de onda que está dentro de un rango de luz visible (aproximadamente entre 400 y 700 [nm]) (Gonzalez y Woods, 2002), los sensores en los ojos que perciben este tipo de luz se llaman conos y estos se pueden dividir categorías: los que perciben rojo, verde y azul. Por esto se puede decir que el color es una combinación de los tres primarios.

Los espacios de color, también conocidos como modelo de color o sistemas de colores, es la herramienta usada para poder representar de una manera estandarizada estas combinaciones de color y poderlo describir o caracterizar de manera general a un color en particular. Existen varios de estos espacios como: RGB (rojo, verde y azul), CMY (cian, magenta, amarillo), CMYK (cian, magenta, amarillo y negro), etc. En general los modelos están orientados para hacer uso de un hardware o uso en una aplicación donde la manipular el color es parte esencial. Para este trabajo se hablará de dos en particular RGB y HSV.

### **4.1.2. Espacio RGB**

En este sistema, cada color se representa por su componente en cada uno de los tres ejes que tiene, ya que está basado en un sistema de referencia cartesiano. Entonces se genera un área de interés como se muestra en la figura 4.1, en la cual se observa que en los vértices del cubo se encuentran los tres colores primarios (rojo, verde y azul), otros secundarios (amarillo, cian y magenta) y por último en vértices opuestos el negro (en el origen de los ejes) y blanco en el extremo más distante. Si se traza una línea entre los dos últimos puntos mencionados, en ella se encuentra la escala de grises. Por lo que se puede decir es que todos los colores pertenecen al cubo y que el color es el vector que resulta de la suma de sus componentes.

Este espacio es muy usado para mostrar imágenes en monitores, así que al combinarse en la pantalla producen el color deseado en este pixel. Para

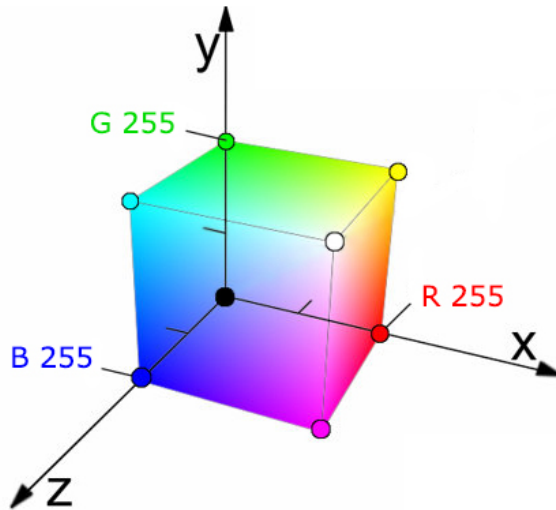


Figura 4.1: Cubo RGB

poderse representar se dice que cada canal de color tiene hasta 8 bits para representar su valor, de esta forma un pixel de RGB de 24 bits (8 bits por color) puede mostrar hasta  $(2^8)^3 = 16,777,216$  colores. Aunque este espacio es muy útil para mostrar imágenes o adquirir (en términos del hardware requerido) y dar un valor estandarizado a los colores, cuando se trata de darle un uso práctico para la interpretación humana no es la mejor opción, ya que no se ajusta a la forma en la que describimos los colores y es muy susceptible a cambios en la iluminación.

### 4.1.3. Espacio HSV y cómo transformar entre ambos

Cuando un humano describe un color no lo hace en términos de la cantidad de cada uno de los colores primarios, sino en términos del matiz, que se refiere al color en sí (naranja, amarillo, rojo, morado, etc.); la saturación, que se refiere al grado de disolución que tiene el color con la luz blanca, y finalmente el brillo, que es una medida difícil de cuantificar, ya que es subjetiva y depende de la percepción de quien está describiendo el color (Gonzalez y Woods, 2002), pero se puede decir que es qué tanto se acerca al negro o al blanco.

Un espacio que hace referencia a estas tres propiedades es el sistema

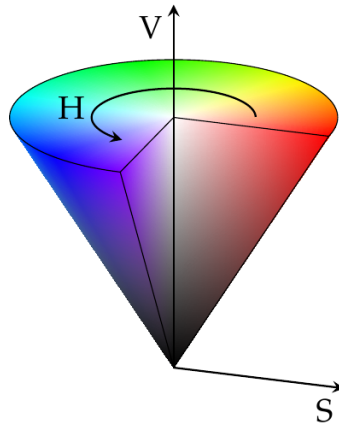


Figura 4.2: Cono HSV

Matiz, Saturación y Valor (HSV, por su siglas en inglés), el cual deja de ser un espacio lineal ya que mapea el cubo RGB en un cono donde el matiz se define por el ángulo que se tiene con respecto a la referencia del ángulo  $0^\circ$  (el cual corresponde al color rojo), la saturación es la distancia del eje que pasa por el centro hacia afuera (entre más cerca del vértice menor saturación), y el valor es la distancia del vértice del cono hacia la base (altura del cono), donde el vértice es el color negro y el centro de la base, el blanco, como se muestra en la figura 4.2:

La relación para transformar de espacio RGB a HSV está dada por las siguientes ecuaciones: se normaliza los valores RGB originales dividiendo entre 255, que corresponde al valor máximo que se tiene en el canal de 8 bits:

$$R' = R/255; G' = G/255; B' = B/255 \quad (4.1)$$

Se obtiene el valor mínimo y máximo de los canales ya normalizados y se calcula el delta con ellos:

$$C_{max} = \max(R', G', B'); C_{min} = \min(R', G', B'); \Delta = C_{max} - C_{min} \quad (4.2)$$

Para el matiz, se aplica la ecuación según los valores previamente obteni-

dos:

$$H = \begin{cases} 0^\circ, & \Delta = 0 \\ 60(\frac{G'-B'}{\Delta} \bmod 6), & C_{max} = R' \\ 60(\frac{B'-R'}{\Delta} + 2), & C_{max} = G' \\ 60(\frac{R'-G'}{\Delta} + 4), & C_{max} = B' \end{cases} \quad (4.3)$$

Para la saturación, se usa la siguiente ecuación y solo depende del valor máximo:

$$S = \begin{cases} 0, & C_{max} = 0 \\ 60(\frac{\Delta}{C_{max}}), & C_{max} \neq 0 \end{cases} \quad (4.4)$$

Para el valor, solo se asigna el valor máximo:

$$V = C_{max} \quad (4.5)$$

#### 4.1.4. Operadores morfológicos

Morfología es un concepto que generalmente se refiere a una rama de la biología que estudia la forma y estructura de las plantas y animales. En este contexto morfología matemática, son operaciones matemáticas que se usan para poder aislar componentes que son de utilidad para representar y describir una región (Gonzalez y Woods, 2002). Estas operaciones se aplicarán a imágenes binarias, las cuales solo tienen como valor de pixel cero o uno. Se empezará por describir dos operaciones morfológicas básicas, que son la dilatación y erosión, para proceder a describir la cerradura y apertura.

##### Dilatación y Erosión

La dilatación es una operación que se puede decir que expande, agranda o engruesa los objetos en una imagen binaria. Este proceso depende directamente de la forma y tamaño del kernel que es otra imagen binaria del tamaño que convenga según los propósitos y que tiene un punto de ancla. Las formas más comunes del Kernel son cruz, elipse y rectángulos. Como lo muestra la figura 4.3, en la cual se puede observar que con dos diferentes tamaños de kernel se obtienen diferentes resultados aplicando la misma operación a la misma imagen. En el inciso a, se tiene la imagen binaria a la cual se le aplicó la dilatación, en el caso del Kernel 1 (inciso b) es de tamaño de 17X17[px] en forma de elipse, con el punto de anclaje en el centro. Mientras que el kernel 2 (inciso c), tiene las mismas características que el kernel 1 exceptuando que es del doble de tamaño. La marca de interés son los pixeles en blanco, la

cual se ocluyo con una mano para poder mostrar el efecto del operador, por lo que la mancha negra medio ovalada (un mano), no interesa para nuestros propósitos.

Lo que se observa en el inciso d, es que a aplicar la operación con el kernel más pequeño la mano fue reducida de su tamaño o se agrando la marca de interés y al aplicarse la operación con el kernel más grande (inciso e), incluso la mano perdió su forma y la marca de interés casi se convierte en un rectángulo. Por lo que se puede decir que la dilatación nos sirve para quitar elementos no deseados y tiende a juntar zonas de interés o fusionarlas.

La operación está definida por la ecuación 4.6, donde  $A$  es la imagen binaria,  $B$  el kernel y  $z$  todos los puntos donde  $B$ , trasladado  $z$  veces sigue perteneciendo a  $A$ . Esta operación se parece a una convolución, pero al estar basada en operaciones de conjuntos no es una operación lineal.

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\} \quad (4.6)$$

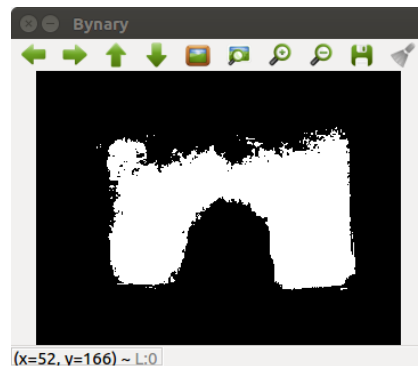
La erosión es la operación inversa, ya que esta encoje, adelgaza o empequeñece los elementos. En la figura 4.4, al igual que el ejemplo de la dilatación se le aplico la misma operación a la misma imagen con kernels distintos, de las características previamente descritas. En el resultado se puede observar que este hace más pequeña la zona de interés, ya que hace crecer la mancha central, este efecto se ve mas notorio en el inciso e. Esta operación es útil cuando se quieren separar zonas de interés en la misma imagen.

La erosión también se define por una ecuación (4.7), donde  $A$  vuelve a ser la imagen binaria;  $B$  el kernel y  $z$  el espacio por donde puede desplazar el kernel.

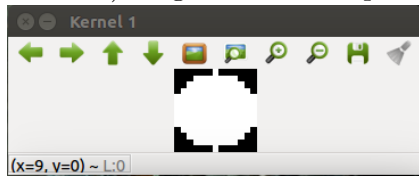
$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (4.7)$$

### **Apertura y cierre**

Estas operaciones son definidas por una combinación de aplicar las operaciones de erosión y dilatación en cierto orden. La operación de apertura es generalmente usada para suavizar el contorno de objetos o desaparecer líneas delgadas entre elementos. La operación de cierre también suaviza los contornos, pero al contrario de la apertura esta elimina pequeños hoyos y llena los huecos en el contorno, sin mencionar que junta pequeños salientes y los hace uno (Gonzalez y Woods, 2002).



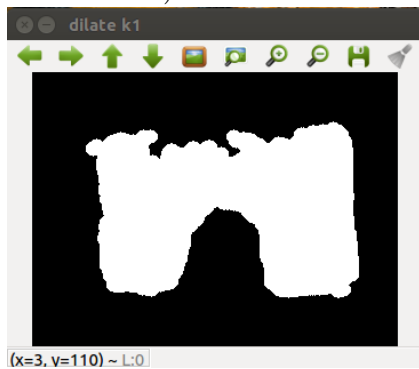
a) Imagen binaria después de la segmentación de color



b) Kernel 1



c) Kernel 2

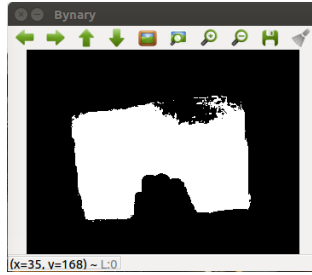


d) Imagen dilatada con Kernel 1

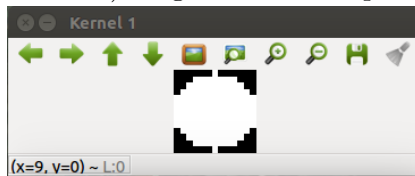


e) Imagen dilatada con Kernel 2

Figura 4.3: Dilatación



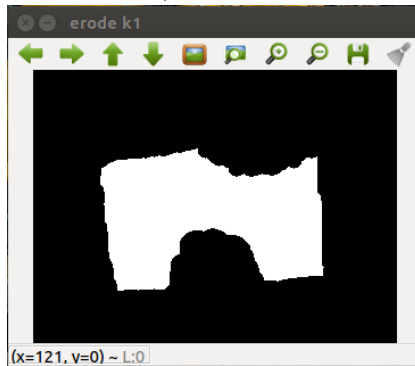
a) Imagen binaria después de la segmentación de color



b) Kernel 1



c) Kernel 2



d) Imagen dilatada con Kernel 1



e) Imagen dilatada con Kernel 2

Figura 4.4: Erosión

La apertura se define como:

$$A \circ B = (A \ominus B) \oplus B \quad (4.8)$$

La cerradura esta dada por:

$$A \forall B = (A \oplus B) \ominus B \quad (4.9)$$

Para este trabajo se usó la operación de apertura ya que uno de sus efectos más importantes es eliminar elementos aislados que tiene un valor más alto que su vecindario y los pequeños elementos que se presentaban al hacer la segmentación de color los elimina de forma contundente. Mientras que la operación de cierre elimina los elementos que tienen un valor menor a sus vecinos. Como se muestra en la figura 4.5, donde se observa claramente como de la imagen binaria, queda una mancha de color sin los pequeños elementos aislados de la parte superior derecha a diferencia que con el cierre donde no se eliminan.

Una vez segmentada la región de color y después de aplica la operación de apertura, se obtiene el centroide  $(p_x, p_y)$  promediando las coordenadas de todos los pixeles en la región.

## 4.2. Control de los actuadores

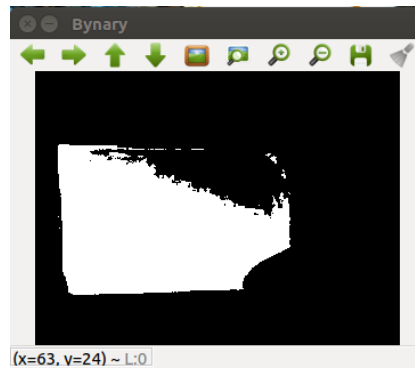
Ya que se obtiene la información de la región de color (el centroide), se tiene que lograr que el robot se acerque al punto que se requiere. Para ello, se implementó un nodo de control que con base en un error de ángulo se acerca a la marca de color deseada. El error se calcula como se describe a continuación.

Sean  $c_x$  y  $c_y$  el centro de la cámara obtenido de la resolución de la imagen y  $p_x$ , y  $p_y$  el centroide de de la marca de color. El error esta dado por:

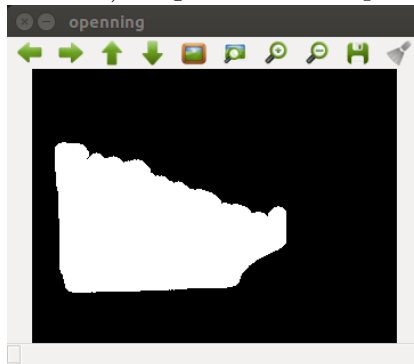
$$e_\theta = c_x - p_x$$

En otras palabras, es la diferencia entre el pixel central de la cámara y el centroide dado. Es importante remarcar que solo se toma en cuenta el error horizontal, ya que al ser una base diferencial no se puede hacer un movimiento vertical para compensar el error vertical. Haciendo uso de este parámetro se calculan las velocidades de los motores  $v_i, v_d \in [-1, 1]$  donde el signo indica el sentido del motor y la magnitud es el porcentaje de la velocidad máxima





a) Imagen binaria después de la segmentación de color



b) Imagen después de aplicar operación de apertura



c) Imagen después de aplicar operación de cierre

Figura 4.5: Apertura VS Cierre

del motor. El control implementado es un control de tipo proporcional dado por las siguientes ecuaciones:

$$v_i = \begin{cases} v_{o_i} + Ke_\theta, & e_\theta > 10 \\ v_{o_i}, & \text{otro caso} \end{cases} \quad (4.10)$$

$$v_d = \begin{cases} v_{o_d} + Ke_\theta, & e_\theta < -10 \\ v_{o_d}, & \text{otro caso} \end{cases} \quad (4.11)$$

donde  $K$  es la constante proporcional,  $v_{o_i}$  y  $v_{o_d}$  son las señales mínimas que se necesitan para que los motores salgan de la zona muerta.

Estas velocidades son calculadas hasta que se obtiene un cierto tamaño en pixeles de la marca, que debido al modelo de Pinhole (ver sección 3.2.1), se puede establecer que es una distancia entre la el área de la marca y la cámara. Este nodo solo recibe la información del nodo de visión para realizar los cálculos, pero existe otro nodo que le indica cual es el lugar al que debe de llegar.

### 4.3. Interfaz gráfica

Para este trabajo se realizó una interfaz gráfica como lo muestra la figura 4.6, la cual está conformada por 3 elementos: el primero a la izquierda de la ventana es una representación de la arena en donde la zona roja, corresponde a la marca roja, la zona verde no corresponde a ninguna marca en particular y la tercera zona corresponde a la marca de color azul. En esta parte se muestra cual es la marca que el robot tiene que alcanzar; el segundo elemento se encuentra la parte superior derecha y corresponde a un indicador de que zona debe de llegar, es decir, la zona roja, se le asignó el número 1, a la verde el número 2 y por último el número 3 a la zona azul; Para finalizar se puede observar el tercer elemento, que es el botón con el cual se usará en las tareas de reforzamiento (ver sección 4.4). Cuando el observador ve que la posición del robot concuerda con lo que se muestra en esta interfaz y además es el comportamiento que se quiere obtener en el robot, se debe de dar un clic y con esto queda reforzada la acción.

Como se puede observar en la figura 4.7, y solo por dar un ejemplo, el robot se le mando ir a la zona uno, donde debe de quedar muy cerca de la marca de color rojo.

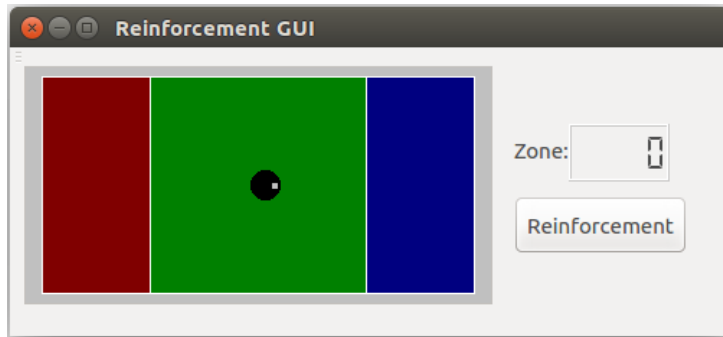


Figura 4.6: Interfaz Gráfica

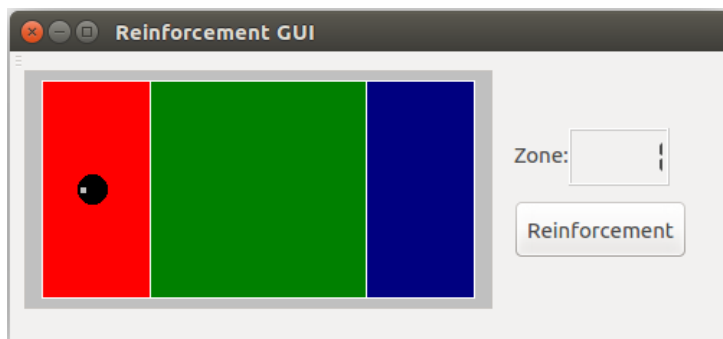


Figura 4.7: Interfaz Gráfica 2

Movimiento	Descripción
0	No hacer nada
1	Acercarse al rojo
2	Quedar lejos del rojo
3	Acercarse al azul
4	Quedar lejos del azul
5	Media vuelta y acercarse al rojo
6	Media vuelta y quedar lejos del rojo
7	Media vuelta y acercarse al azul
8	Media vuelta y quedar lejos del azul
9	Quedarse cerca del rojo
10	Quedarse cerca del azul

Tabla 4.1: Lista de movimientos

## 4.4. Implementación de la Ley del Efecto

Como se explica en el capítulo 2.6.1 la Ley del Efecto afirma que un comportamiento tiende a repetirse si es seguido de un estímulo satisfactorio. En este experimento tenemos los estímulos, que es la información que se tiene del nodo de visión, las respuestas que son el conjunto de movimientos que puede realizar el robot (tabla 4.1) y gracias a esta pareja, se pueden observar comportamientos en el robot, es decir, el robot dependiendo de lo que “observa” genera un movimiento (respuesta) y esto se refleja en el comportamiento de llegar a una de las zonas (roja, neutra o azul).

Para elegir este movimiento y poder reforzarlo se programó en un servicio de ROS, que cuando es llamado devuelve un número correspondiente a una de las 3 zonas del experimento. Estas zonas están representadas por un vector de probabilidades y al inicio están distribuidas uniformemente (un tercio cada probabilidad), por lo tanto, al elegir una de las zonas todas tiene la misma probabilidad de ser elegidas, lo cual se ve reflejado en que el robot visita las 3 zonas un número similar de veces.

Cuando se quiere que el robot empiece a visitar una zona en particular se debe de reforzar el comportamiento cuando aparezca, es decir que se tenga el evento satisfactorio previamente mencionado. Cuando este tipo de experimentos es realizado en seres vivos como ratas o palomas, se les da algún

alimento dulce, pero en el caso de un robot interpreta señales, así que el refuerzo se aplica dando clic en el botón de refuerzo que está en la interfaz gráfica.

Al presionar este botón, se llama a una función que actualiza el vector de las probabilidades de las zonas de acuerdo con la ecuación 4.12 donde  $k_n$  es la constante de no recompensa,  $K_r$  constante de recompensa.

$$X(t + 1) = k_N X(t) + K_R [1 + X(t)] \quad (4.12)$$

Realmente, lo que quiere decir es que cuando se quiere reforzar un comportamiento se le aplica la ecuación 4.13 y a los comportamientos que no se quieren reforzar se les aplica la ecuación 4.14. Las constantes solo hacen que se olviden ( $k_n$ ) o se aprendan ( $K_r$ ) más rápido o lento los comportamientos.

Refuerzo:

$$X(t + 1) = K_R [1 + X(t)] \quad (4.13)$$

No refuerzo:

$$X(t + 1) = k_N X(t) \quad (4.14)$$

En la tabla 4.2 se muestra un ejemplo de cómo se van calculando las probabilidades cuando se trata de reforzar el comportamiento de ir a la zona 1 (roja) y  $K_r = K_n = 0.9$ . En la primera petición el servicio respondió con la zona 1, tomando la decisión con la probabilidad uniforme (renglón 1). Al llegar a la zona 1 se aplicó el refuerzo y como se observa en la segunda petición las probabilidades ya cambiaron, aumentó la probabilidad de la zona 1 de disminuyeron el resto. Aun así, este servicio respondió con la zona 2. Para la tercera petición se mantuvieron las probabilidades igual ya que al visitar la zona 2 no se reforzó el comportamiento. En la última petición, se modifican de nuevo las probabilidades ya que en el intento pasado se visitó la zona 1 y se aplicó el refuerzo.

Con el ejemplo anterior, se puede observar cómo es que la Ley del Efecto va modificando la probabilidad de que un comportamiento deseado se repita, En el siguiente capítulo se describe como es que se implemento el Modelo de Contexto Generalizado para mejorar un sistema reconocimiento de objetos.

Intento	Zona devuelta	p(zona 1)	p(zona 2)	p(zona 3)
1	1	0.333333	0.333333	0.333333
2	2	0.608695	0.195652	0.195652
3	1	0.608695	0.195652	0.195652
4	3	0.731788	0.134105	0.134105

Tabla 4.2: ejemplo de las probabilidades

# Capítulo 5

## Sistema de visión con base en el Modelo de Contexto Generalizado

En éste capítulo se presenta todo el software tanto desarrollado como usado para la implementación de Modelo de Contexto General descrito en el capítulo 2.6.2 de este trabajo. Se empieza por especificar como obtiene la probabilidad de estar en una determinada área haciendo uso de un enfoque probabilístico como lo es un filtro de partículas. Posteriormente se detalla cómo es que funciona el sistema actual de visión y las modificaciones que se le hicieron a dicho sistema.

En materia de detectores, se han desarrollado varios conocidos como SURF, SIFT, BRIEF Y ORB, pero estos algoritmos funcionan con objetos ricos en texturas (Bradski y Kaehler, 2011; Lowe y others, 1999). En los hogares, muchos de los objetos que se encuentran carecen de texturas, como los vasos, platos, cubiertos, tazas, etc. Debido a esto, se beben de utilizar características del objeto como el color, tamaño y forma. Por lo cual, Cruz Navarro (2016) creó un detector de objetos sobre planos horizontales basado en las características antes mencionadas. El sistema consiste en obtener una nube de puntos de un sensor RGB-D, calcular los planos horizontales y crear máscaras binarias del objeto que esté sobre el plano. Con esto ya se puede obtener la información necesaria para implementar el reconocimiento. Se calculan 3 valores de error (tamaño, forma y color) y con estos valores se clasifica qué tipo de objeto es. Aunque este sistema mejoró el detector que antes se usaba en el robot, aún sigue teniendo fallos al clasificar. Si se

hace uso de la información de navegación del robot, es decir su pose estimada, se podría mejorar la clasificación de los objetos. Se propone incluir esta información mediante el Modelo de Contexto Generalizado. El cual esta compuesto por tres grandes pasos.

Primero se obtiene la probabilidad de que el robot este en una determinada área, lo siguiente es calcular la similitud de acuerdo con las ec 2.1 y 2.2. Finalmente se corrige probabilidad a las distancias para reconocer el objeto con la ecuación 2.3.

## 5.1. Cálculo de la probabilidad de estar en un área

En el Modelo de Contexto Generalizado se propone que se debe de calcular una activación dado un factor de memoria, que en general representa que tanto se ha observado el objeto recientemente. Dado que no se tiene un nodo de memoria implementado en el robot, se propone que este ponderador dependa de en que espacio debería de estar el objeto y en que espacio se encuentra el robot. Es decir, un rollo de papel de baño tiende a estar en el baño o en el cuarto donde se guarda y si un rollo de servitoallas tiende a estar en la cocina o comedor. Entonces, haciendo uso de esta información apriori se calcula una probabilidad de que el robot se encuentra en la área correspondiente y está probabilidad se usa como valor de memoria. El efecto que esto produce es que al estar detectando objetos y dependiendo del área en que se encuentre el robot se va a tener que un ponderador más grande para el objeto que se debería de encontrar en esta área pero no erradica la posibilidad de encontrar un objeto que no deba de estar en ese lugar.

### 5.1.1. Localización por filtro de partículas

Hay diferentes formas de localizar o estimar la posición de un robot, un ejemplo de ellas y la que esta implementada en el robot Takeshi es mediante un filtro de partículas. Este filtro es un modelo probabilístico que Thrun (2002) define como una técnica de aproximación para el cálculo de distribuciones a posteriori para cadenas de Markov controlables y parcialmente observables de tiempo discreto. Este consiste en tener una distribución de probabilidad y de esta se calculan un número  $N$  determinado de muestras o



partículas que tendrá cierto peso o probabilidad. Cada una de estas partículas se desplazan una pequeña distancia (dado que en este caso representan una posición) y se les agrega ruido, con estas nuevas partículas y la lectura del sensor (un sensor láser para este caso) se calcula una nueva función de distribución, que al repetir este ciclo varias veces crea partículas cada vez más condensadas a la posición estimada o más probable. Entonces si se obtiene el promedio de sus valores se puede estimar la posición.

La explicación a detalle de esta técnica esta fuera del alcance de este trabajo, pero se puede consultar en Siciliano y Khatib (2008) y Thrun (2002). A partir de este filtro de partículas se puede calcular una probabilidad de que el robot este localizado en un área determinada.

La representación del ambiente usada es conocida como un mapa de celdas de ocupación, y se define como una técnica en donde para cada celda de una malla discreta se le asigna un valor de probabilidad que el área correspondiente en el espacio este ocupada por un obstáculo (Siciliano y Khatib, 2008). En la figura 5.1 se observa el mapa ocupado para la realización de este experimento y corresponde al Laboratorio de Biorrobótica. Como se puede observar en la figura las zonas blancas representan un espacio libre, las zonas negros un espacio ocupado y la zona gris es un espacio inexplorado.

A lo que llamaré área o zona de la casa, es una representación de una habitación como lo muestra la figura 5.2, en la cual se puede ver que sobre el mapa previamente construido se delimitan las zonas. Esto se consigue con los vértices de cada espacio en coordenadas del mapa.

Teniendo el mapa, el filtro de partículas y las zonas delimitadas se procede a calcular la probabilidad de encontrarse en la habitación. La forma en que se obtiene este valor consiste lo siguiente: Cada partícula  $P_i$  tiene asociada una posición  $([P_{i_x}, P_{i_y}])$  y una covarianza  $cov(P_i)$  con distribución normal y dado a que se tiene un conjunto de ellas se calcula su función de densidad dado todo este conjunto  $N$  de partículas para cada una de las celdas  $C_{xy}$  con coordenadas  $[C_x, C_y]$  del mapa como se muestra en la siguiente ecuación:

$$P_{C_{xy}} = \frac{\sum_{i=1}^N f(C_{xy}, P_i, cov(P_i))}{N} \quad (5.1)$$

donde  $f$  es la función de densidad de probabilidad asociada a cada partícula, definida como:

$$f(C_{xy}, P_i, cov(P_i)) = \frac{1}{\sqrt{(2\pi)^2 cov(P_i)}} e^{(-\frac{1}{2}(P_{i_{xy}} - C_{xy})^T cov^{-1}(P_i))} \quad (5.2)$$

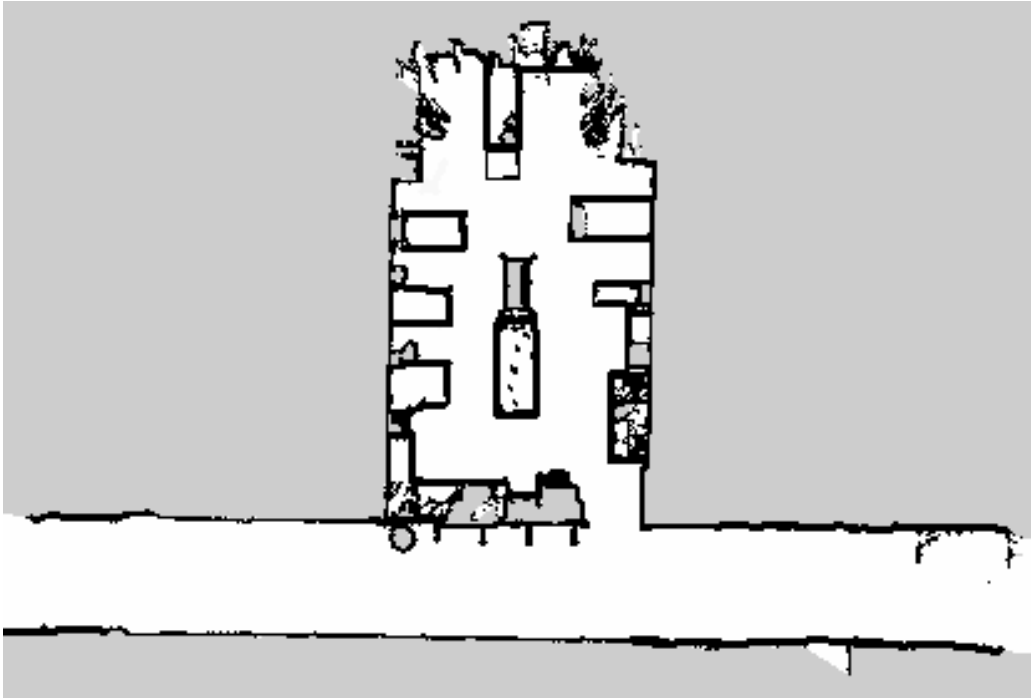


Figura 5.1: Mapa del Laboratorio de Biorrobótica

Una vez obtenida la probabilidad para cada celda ( $P_{C_{xy}}$ ) se aproxima la probabilidad de una zona multiplicando por un pequeña área el valor de cada celda creando un prisma y sumando todos los prismas que pertenecen a la región de interés. Se muestra un ejemplo en la figura 5.3, donde en el inciso a se puede observar como es que al iniciar el nodo del filtro de partículas, las partículas (las flechas) están distribuidas por toda una parte del mapa.

Al realizar el cálculo de probabilidades para estas dos áreas existentes se obtuvo que la probabilidad de que el robot se encuentre en la cocina es de 0.33 y para la oficina es de 0.24, se puede decir que el algoritmo dice que lo más seguro es que se encuentre en la cocina incluso cuando claramente el robot está en la oficina, lo cual, es normal dado que no ha iterado. Al navegar una distancia pequeña, como lo muestra la línea de la trayectoria que siguió, El filtro empieza a iterar y se puede observar en el inciso b de la figura 5.3 como las partículas tienden a conglomerarse alrededor del robot, en este punto se recalcularon las probabilidades y se obtuvo que la probabilidad de estar en la cocina es 0.33 y para la oficina es 0.36.

Por estos datos, se puede decir que el filtro esta empezando a converger y ahora si se piensa que es estar más probable estar en la oficina. Por último, en el inciso c de la misma figura (5.3) se recorrió una trayectoria más larga y ahora se llevo a la cocina, el cúmulo de partículas ya se encuentra muy cerca de la posición del robot. Las probabilidades calculadas son las siguientes: para la cocina se estimo en 0.81 y para la oficina en 0.008, por lo que se puede observar que cuando el algoritmo converge la información es fidedigna. Una vez obtenida la probabilidad para cada celda ( $P_{C_{xy}}$ ), se obtiene la suma de todas las celdas que pertenecen a la región y esta suma es la probabilidad de que el robot esté dentro de esa área.

## 5.2. Reconocedor de objetos actual en los robots de servicio Justina y Takeshi

En la figura 5.4 modificado de Cruz Navarro (2016) se muestra como se implementó el reconocedor de actual. El proceso inicia calculando dos tipos de error y una similitud, donde el error  $e$  que representa una altura  $h$  denominado como  $e_h$  se obtiene como una diferencia de altura, para todos los elementos  $O_i$  de la base de datos, contra el el objeto detectado  $O_d$  como se muestra en la siguiente ecuación:

$$e_h = |h_{O_d} - h_{O_i}| \quad (5.3)$$

Después, se obtiene el error de forma  $e_s$  del objeto detectado contra todos los objetos en la base de datos:

$$e_s = I(O_d, O_i) = \sum_{j=1}^{j=7} \left| \frac{1}{m_j^{O_d}} - \frac{1}{m_j^{O_i}} \right|$$

donde :

$$\begin{aligned} m_j^{O_d} &= \text{sign}(h_j^{O_d}) \log h_j^{O_d} \\ m_j^{O_i} &= \text{sign}(h_j^{O_i}) \log h_j^{O_i} \end{aligned} \quad (5.4)$$

y  $h_i^{O_d}$  y  $h_j^{O_i}$  son los momentos de Hu de cada imagen

En el siguiente paso se calcula la similitud de color, comparando los his-

togramas de color:

$$s_c = d(H_{O_d}, H_{O_i}) = \sum_I \min(H_{O_d}(I), H_{O_i}(I)) \quad (5.5)$$

A este valor se le conoce como la intersección entre histogramas y entre más parecidos son, el valor resultante es mayor, dado que se crea una intersección más grande de pixeles. Por último, se evalúa si el error de color y forma están dentro un umbral, y si la similitud de color es mayor a otro umbral. Si alguno de estos no esta dentro de estos limites se considera como un elemento no entrenado. Si el objeto esta dentro de los parámetros, el que tenga mayor similitud de color se considera como el objeto encontrado de la base de datos. Cómo se está asumiendo que se tiene un conocimiento previo para este trabajo se utilizo una función de este sistema donde se hace la consideración que todos los objetos que va a encontrar están en la base de datos, así que no habrá objetos desconocidos.

### 5.3. Reconocedor de objetos con GCM

Las adaptaciones que se hicieron para la implementación del algoritmo de clasificación tienen un costo computacional un poco más alto que solo las comparaciones que previamente se describen. El diagrama del nuevo proceso de reconocimiento se muestra en la figura 5.5 y es el siguiente:

Si se tienen  $j$  ejemplos en la base de objetos entrenados, entonces para cada objeto detectado ( $i$ ) se calcula el error de altura  $e_{h_j}^i$  con la ecuación 5.3, el error de forma  $e_{s_j}^i$  con la ecuación 5.4 y el error de color  $e_{c_j}^i$  con la ecuación 5.6. Se obtiene un “inverso” de este error, ya que como se explicó previamente, el valor es mayor si es que son más parecidos.

$$e_{c.inv} = e^{-s_c} \quad (5.6)$$

Una vez obtenidos estos valores, se calcula una distancia (ver sección 2.1), en el espacio psicológico resultante de los errores con la ecuación 5.7, donde la distancia euclidiana del objeto detectado  $i$ , es calculada para cada elemento  $j$  de la base de datos con los errores  $k$

$$d_{ij} = \left[ \sum_{k=1}^{k=3} |x_{ik} - x_{jk}|^2 \right]^{\frac{1}{2}} \quad (5.7)$$

Ya que se tienen las distancias se obtiene la similitud con la ecuación 5.8 donde el valor  $c$  (de sensibilidad) se estableció como 20.

$$s_{ij} = e^{-20d_{ij}} \quad (5.8)$$

Después de obtener el valor de similitud se aplica la ecuación de la activación 2.3 y se obtiene  $a_{ij}$ , donde el valor de memoria como se explica en 2.6.2 sustituyo por la probabilidad de estar en el lugar donde el debería de estar el objeto y con esto el que tuvo el valor mayor corresponde al objeto identificado.

$$O_{d_i} = \operatorname{argmax}_j(a_{ij}) \quad (5.9)$$

Una vez realizada todas las implementaciones para que estuviera en línea el sistema de visión propuesto en este trabajo, se recabaron los datos para poder comparar ambos sistemas de visión, los cuales junto con los resultados del primer experimento descrito en el capítulo 4 se muestran en el siguiente capítulo.

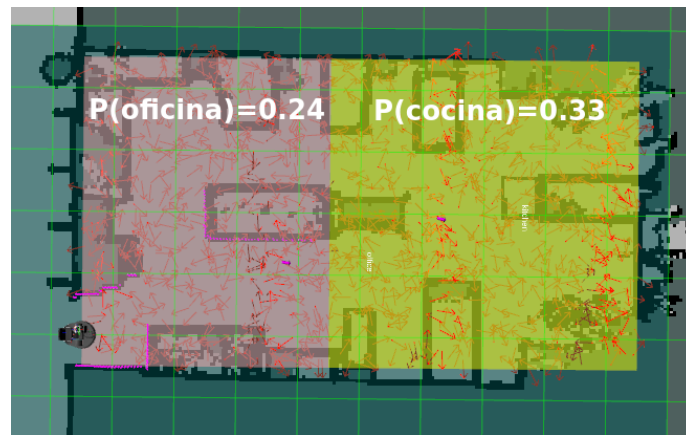


a) Mapa sin áreas

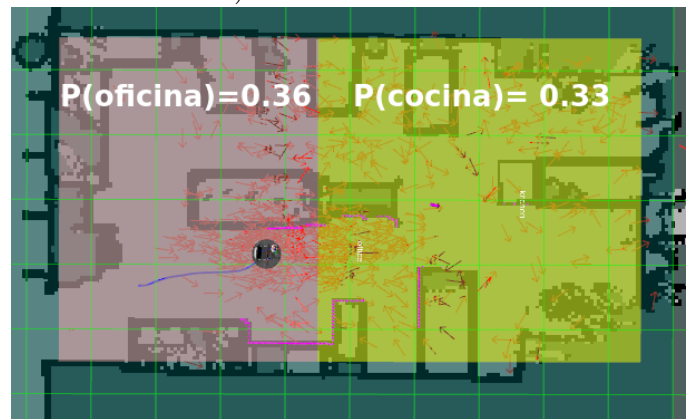


b) Mapa con áreas

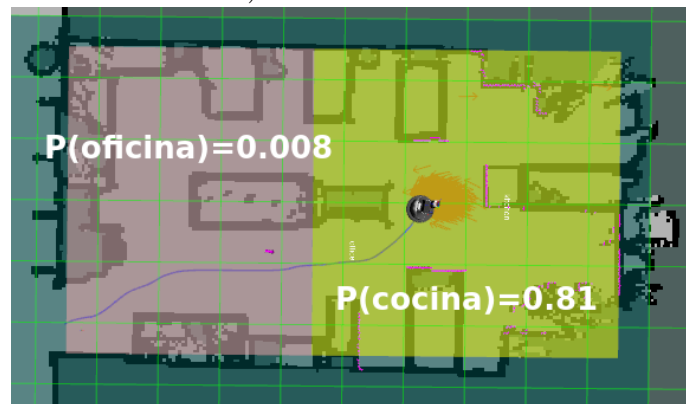
Figura 5.2: Mapa de celdas de ocupación



a) Distribución inicial



b) Pocas iteraciones



c) Completamente localizado

Figura 5.3: Localización por filtro de partículas

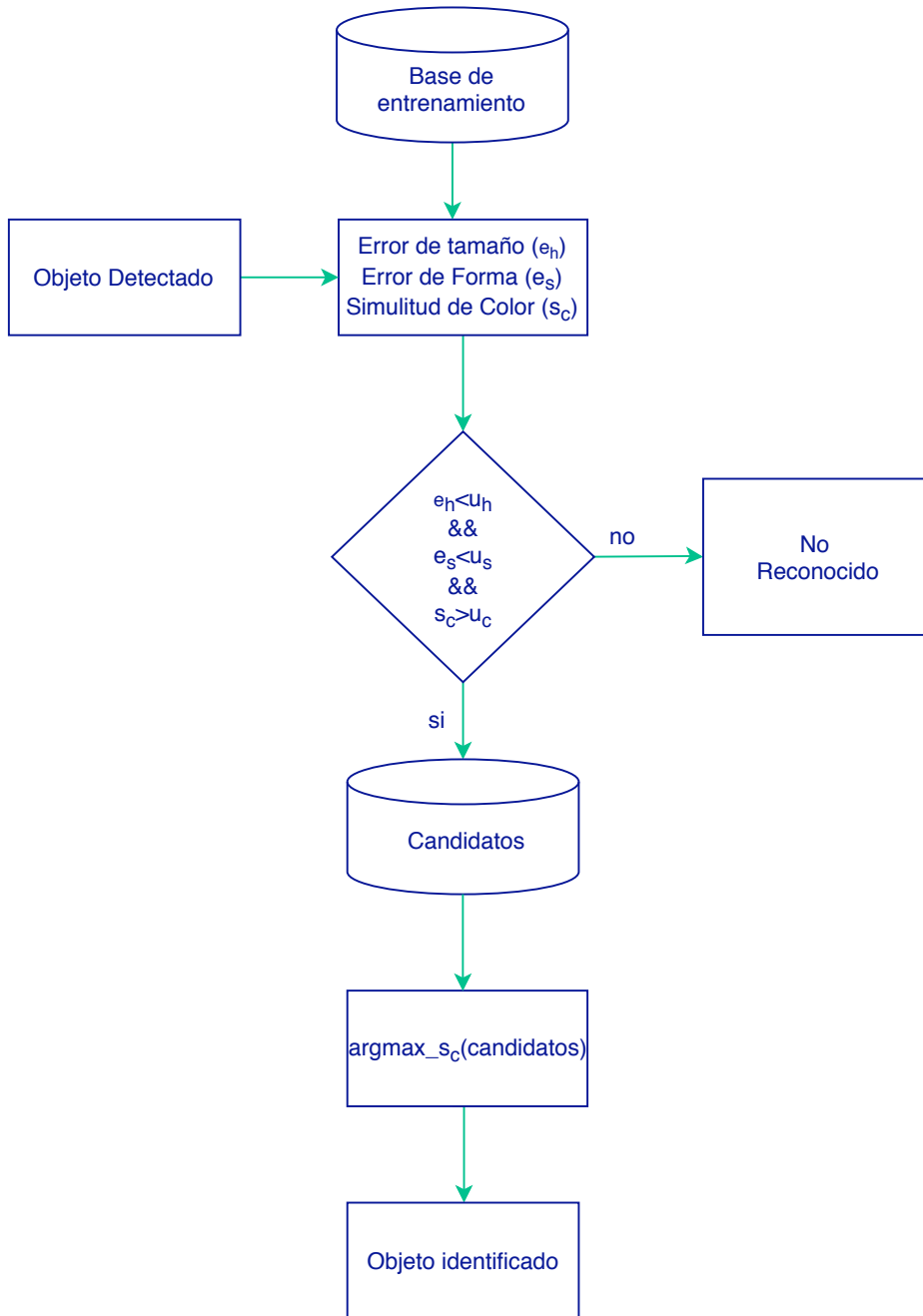


Figura 5.4: Reconocedor de objetos actual



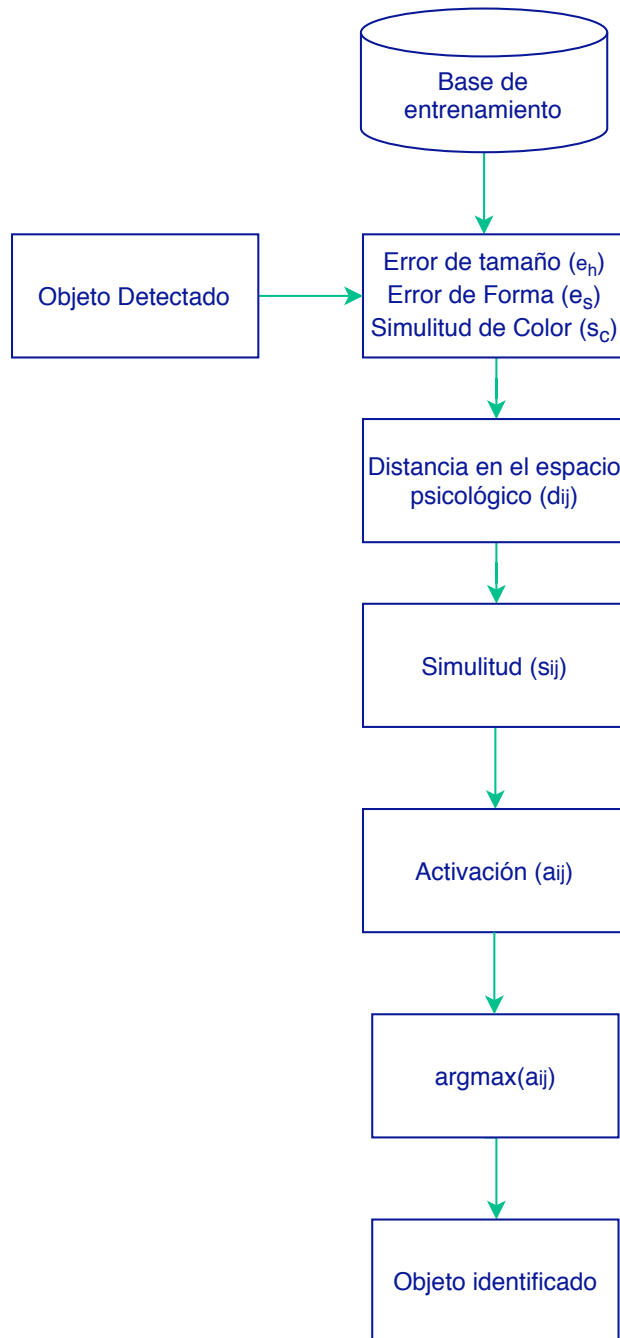


Figura 5.5: Reconocedor de objetos con GCM

# Capítulo 6

## Resultados

A continuación, se relatan los experimentos realizados y se muestran los resultados obtenidos tras haber realizado repetitivamente dichos experimentos. Para seguir con el orden de este trabajo primero se aborda el experimento referente a la Ley del Efecto y posteriormente se muestran los resultados y comparación de la modificación realizada al aplicar el Modelo de Contexto Generalizado.

### 6.1. Robots de este trabajo

Cada uno de los modelos descritos en este trabajo se implementó en un robot diferente, para el caso de la Ley del Efecto se construyó el robot descrito en el capítulo 3 y para el Modelo de Contexto Generalizado se usó un robot fabricado por la empresa Toyota y que fue facilitado por el Laboratorio de Biorrobótica de la Facultad de Ingeniería de la UNAM, del cual se profundiza más en la sección 6.1.1, dado que cuenta con el hardware necesario para la implementación del modelo previamente mencionado.

#### 6.1.1. El robot de servicio HSR

El robot denominado Takeshi es un robot fabricado por la empresa Toyota en su línea de robots de apoyo humano (HSR). Yamamoto et al. (2019) lo definen como un robot manipulador doméstico móvil con funciones para realizar tareas físicas dentro de un espacio doméstico y de comunicación para monitoreo. Este proyecto nació debido a que la tasa de nacimiento de diversos

países (incluido Japón) ha decrecido, al envejecer la población necesitará ayuda para realizar tareas básicas en su casa debido a discapacidades que vienen con la edad. Este robot ha sido elegido para ser el estándar en la liga @HOME-DSPL de la Robocup desde el 2017 y para el World Robot Summit (WRS).

Para la realización de este trabajo ya se tenía una versión del sistema de visión implementado (descrito en la sección 5.2). A está, se le realizaron las modificaciones necesarias para la implementación de la mejora haciendo uso del Modelo de Contexto Generalizado.



Figura 6.1: Robot HSR

## 6.2. Comportamientos con Ley del Efecto

El experimento realizado para probar la Ley del Efecto consiste en colocar al robot dentro del arena (donde están las 3 zonas contiguas propuestas para este experimento) viendo hacia una marca de color. El robot empieza a navegar aleatoriamente por las 3 áreas. El operador elige un área y cada vez que el robot llegue a esa zona debe de oprimir el botón que refuerza el comportamiento. Para leer sobre como es que funciona esté experimento ver el capítulo 4.

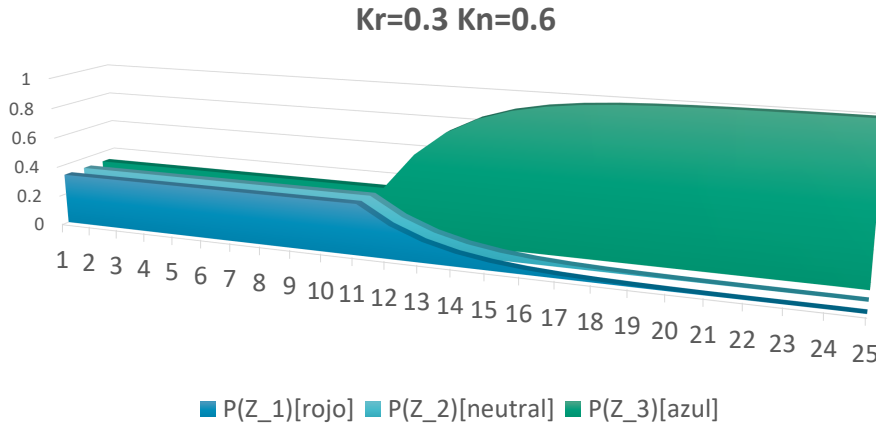


Figura 6.2: Probabilidades de visitar cada zona

En la figura 6.2 se tiene una gráfica en la cual en el eje horizontal se muestra el número de veces que el robot llamó al servicio que le indica a que zona debe de dirigirse, en el eje vertical la probabilidad de que esa zona sea elegida, en el mismo gráfico se muestran los datos de las tres zonas para que se puedan comparar de una manera sencilla. En este gráfico se pueden observar dos fases del experimento. La primera donde el robot empieza a visitar cualquier zona con la misma probabilidad dado que se tiene una distribución uniforme y la segunda (a partir de la décimo primera que llamó al servicio en este ejemplo) donde se muestra como es que se comportan las probabilidades una vez que se vieron afectadas. Estas probabilidades son afectadas por dos constantes que son muy importantes para modificar la rapidez con la que el modelo va a aprender ( $K_R$  y  $K_N$ ) como se puede observar en las ecuaciones 4.13 y 4.14.

Se puede decir que dependiendo de lo rápido que se desea que el robot aprenda se tiene que fijar una  $K_r$  más alta, como se muestra en el ejemplo de la figura 6.3, en la cual se puede observar que en pocos intentos (aproximadamente 3 después del primer refuerzo), se aprendió a ir a la zona deseada y este evento pasará con una probabilidad muy alta y se podría decir que prácticamente deshecho las otras opciones a la primera.

En caso de que se requiera que el robot aprenda de una forma más lenta o que explore más el resto de las opciones, se puede fijar una  $K_r$  más baja

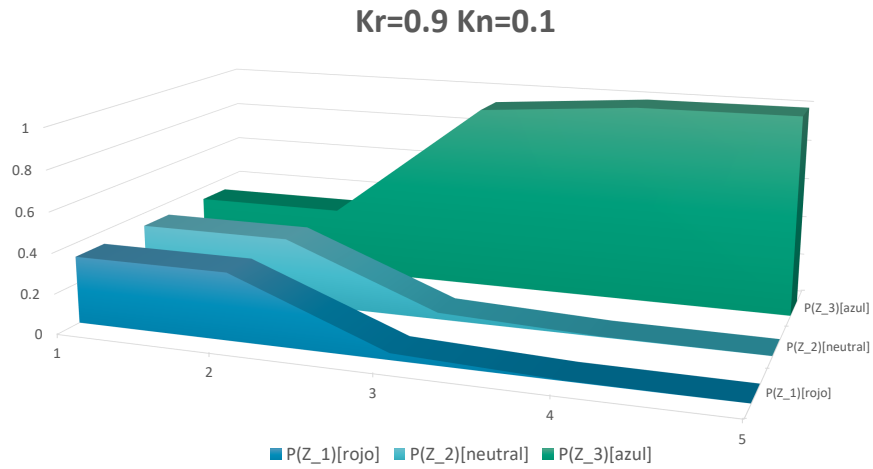


Figura 6.3: Constante  $K_r$  grande

y este proceso de aprendizaje se hará de forma mas lenta y gradual como se observa en la figura 6.4, donde se requirió llamar al servicio muchas veces más que en el ejemplo anterior (6.3).

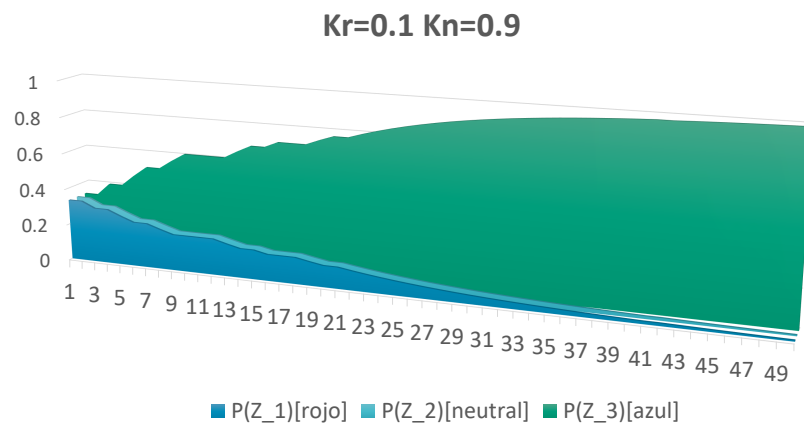


Figura 6.4: Constante  $K_r$  pequeña

### 6.3. Reconocimiento de objetos con GCM

Para realizar las pruebas para este experimento se eligieron 10 artículos que son muy parecidos en tamaño, forma y color y que podrían estar en la cocina o en la oficina de una casa. Estos artículos están definidos en la tabla 6.1 y se pueden observar en la figura 6.5.

No.	Nombre del artículo	Cuarto
1	Taza azul	Cocina
2	Porta plumas azul	Oficina
3	libro de recetas	Cocina
4	Tabla de picar	Cocina
5	Taza verde	Cocina
6	Porta plumas verde	Oficina
7	Juego de mesa	Oficina
8	Taza roja	Cocina
9	Porta plumas rojo	Oficina
10	Pizarrón	Oficina

Tabla 6.1: Lista de objetos

Un ejemplo puede ser el libro y el juego de mesa, como se observa en la figura 6.6, en la cual se puede observar que las medias de los objetos son muy parecidas, los dos son de color azul relativamente parecido y si hablamos de su forma los dos son unos rectángulos, realmente pueden confundir al sistema de visión. Por lo cual, se pretende que teniendo información a priori de donde es que ese objeto se encuentra frecuentemente (esté libro en la cocina ya que se sabe que es un recetario y el juego de mesa en la oficina ya que ahí se encuentran los juegos de mesa) y conociendo la posición del robot se encuentra (en qué zona de la casa) se pueda hacer una diferencia y reconocer de manera más certera el objeto.

Una vez definidos los objetos y el área al que pertenecen, se entrenan para tener sus características en la base de datos y poder compararse. Con esta información se puede correr el experimento en el cual primero se pone a navegar al robot para que pueda localizarse y ya que lo logra se manda distintos puntos pertenecientes a las dos áreas a reconocer objetos, estos objetos están colocados dentro del área que corresponden. Un ejemplo de la



Figura 6.5: Objetos usados para el sistema de visión



Figura 6.6: Objetos Similares

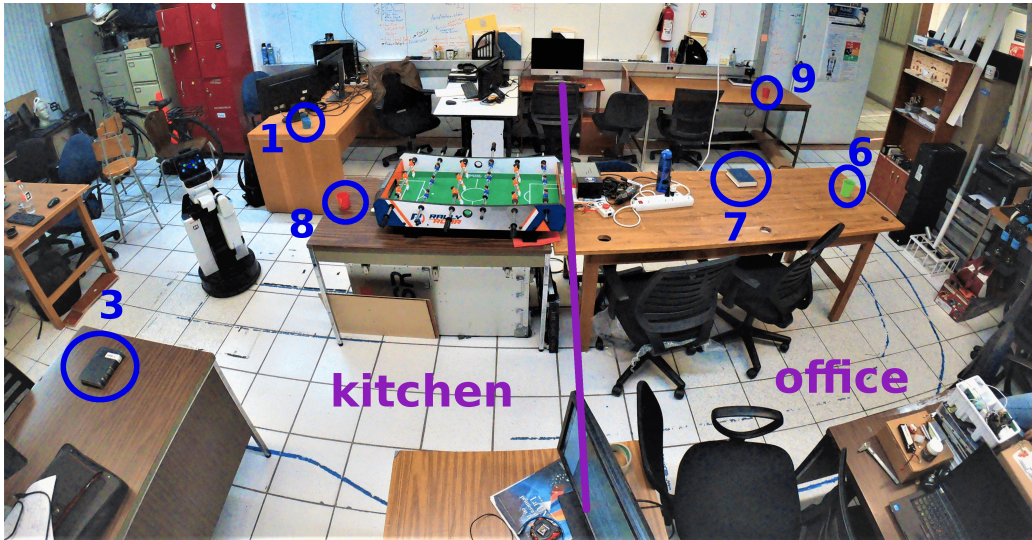


Figura 6.7: Distribución de los objetos

distribución de los objetos se muestra la figura 6.7.

Se realizaron 65 reconocimientos por objeto en su área correspondiente. Estos resultados se muestran en forma de una matriz de confusión (figura 6.8), la cual muestra cuantas veces se reconoció el objeto (que corresponde la suma de cada renglón) y en cada celda se puede ver el número de veces que cada objeto fue identificado consigo mismo o confundido con otro. El estado ideal sería que la diagonal principal de la matriz tuviera el número máximo de intentos debido a que el sistema no hubiera confundido ninguna vez al objeto y por lo tanto la diagonal principal estaría más obscura o con los valores más altos en comparación del resto de los elementos de la matriz.

Como se puede comparar en la figura 6.8, al aplicar el Modelo de Contexto Generalizado se genera un incremento en la diagonal principal en casi todos los elementos y se puede observar que las confusiones aunque siguen existiendo son menores. Es decir, se sigue confundiendo pero con menos frecuencia y no con tantos objetos.



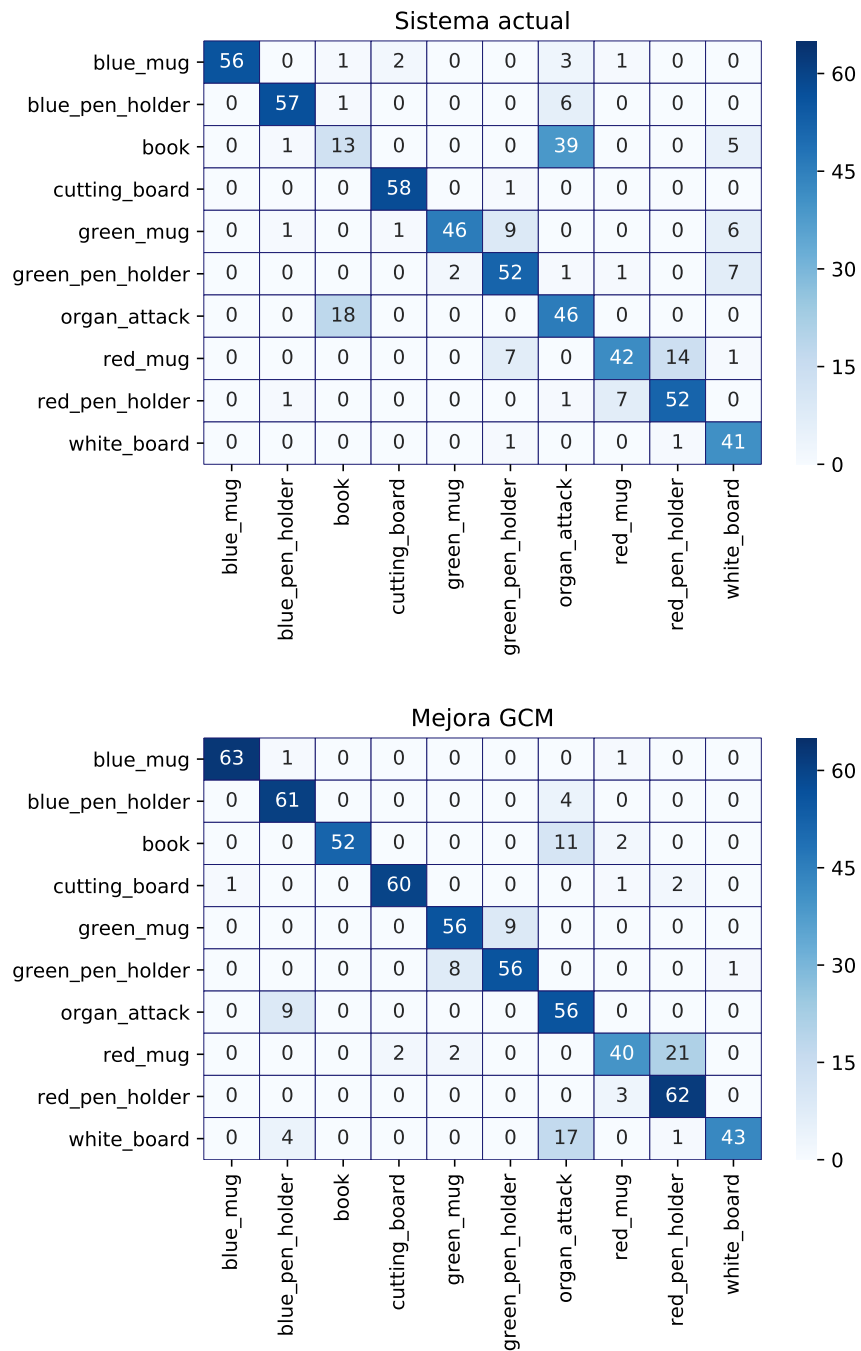


Figura 6.8: Matrices de confusión

# Capítulo 7

## Discusión

### 7.1. Conclusiones

Se construyó un robot móvil autónomo haciendo uso de diferentes componentes de hardware como: la tarjeta de prototipado rápido Arduino, una computadora compacta (Raspberry), una cámara, una unidad de medición inercial (IMU), y se generó todo el software necesario para el funcionamiento básico de robot, es decir, obtener la información de los sensores y mover los actuadores y con lo siguiente poder implementar los algoritmos de más alto nivel. Todo esto haciendo uso de distintos lenguajes de programación y diversas librerías de software. Una vez construido el robot, se realizó la implementación de la Ley del Efecto, para poder generar comportamientos y demostrar que es posible aplicar modelos psicológicos formales para diseñar un comportamiento deseado en un robot móvil autónomo.

Se realizó la implementación de software necesaria para aplicar el Modelo de Contexto Generalizado al sistema de visión que actualmente usan los robots Takeshi y Justina del Laboratorio de Biorrobótica. Que consistió en calcular la probabilidad de encontrarse en un determinado lugar gracias al sistema de navegación y con ello se propuso usarlo como valor de memoria en la ecuación de activación (ec. 2.3). Con lo cual, se generó un experimento para comparar la confiabilidad del reconocedor de objetos comparando el sistema original y la versión realizada para el presente trabajo que hace uso del modelo psicológico formal.

Los diez objetos que se usaron para realizar la comparación entre los dos sistemas se escogieron debido a la gran similitud que tienen, tomando en

cuenta que las 3 principales características en las cuales se basa el sistema para realizar la detección de objetos son: la altura, el color y la forma del objeto. Se encontró un aumento de confiabilidad del sistema en la mayoría de objetos propuestos. Por lo cual, se puede decir que incluir modelos psicológicos formales es una opción factible para el diseño de comportamientos en el área de la robótica y que se debería de explorar más esta área que podría ser de gran utilidad para la robótica.

## 7.2. Trabajo futuro

Para obtener la probabilidad de encontrarse en un área, se implementó calculando la integral definida para cada celda, es decir se calcula la función de probabilidad para cada celda dadas todas las partículas y se multiplica por el área de la celda, lo cual es funcional pero computacionalmente es muy caro. Así que se sería bueno explorar otra forma de obtener el mismo dato con un costo menor.

En este momento el conocer donde se debería de encontrar un objeto determinado esta preestablecido y si se cambia de habitación el objeto no tiene sentido calcular la activación, por lo cual se propone integrar con el módulo creado por Savage et al. (2019) para que conforme el robot realice diversas tareas se pueda actualizar el conocimiento o valor de donde se encuentra el objeto. Otro punto para desarrollar, es que al tener como actualizar la localización del objeto, se puede crear el nodo de “memoria” y se podría combinar en la ecuación de activación para que el el valor resultante sea dependiente tanto del valor propuesto de memoria como de el valor propuesto en este trabajo, es decir de la posición del objeto, lo cual podría generar resultados mejores a los ya generados en esta tesis.

# Referencias

- Arkin, R. C. (2000). *Behavior-based robotics* (3. printing ed.). Cambridge, Mass.: MIT Press. (OCLC: 248262157)
- Badamasi, Y. A. (2014, septiembre). The working principle of an Arduino. En *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)* (pp. 1–4). doi: 10.1109/ICECCO.2014.6997578
- Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., ... Kumaran, D. (2018, mayo). Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705), 429. Descargado 2019-03-05, de <https://www.nature.com/articles/s41586-018-0102-6> doi: 10.1038/s41586-018-0102-6
- Bekey, G. A. (2005). *Autonomous robots: from biological inspiration to implementation and control*. Cambridge, Mass: MIT Press.
- Bradski, G. R., y Kaehler, A. (2011). *Learning OpenCV: computer vision with the OpenCV library* (1. ed., [Nachdr.] ed.). Beijing: O'Reilly. (OCLC: 838472784)
- Busemeyer, J. R. (Ed.). (2015). *The Oxford handbook of computational and mathematical psychology*. Oxford ; New York: Oxford University Press.
- Cruz Navarro, J. (2016). *Detección y reconocimiento de objetos usando imágenes RGB y nubes de puntos organizadas* (Tesis de Master no publicada). Universidad Nacional Autónoma de México (UNAM), México.
- Domjan, M., Grau, J. W., Krause, M. A., Ortiz Salinas, M. E., y Avila Santibáñez, R. (2010). *Principios de aprendizaje y conducta*.
- Forstmann, B. U., Wagenmakers, E.-J., Eichele, T., Brown, S., y Serences, J. T. (2011, junio). Reciprocal relations between cognitive neuroscience and formal cognitive models: opposites attract? *Trends in Cognitive Sciences*, 15(6), 272–279. Descargado 2019-

- 10-18, de <http://www.sciencedirect.com/science/article/pii/S1364661311000726> doi: 10.1016/j.tics.2011.04.002
- Forsyth, D. A., y Ponce, J. (2003). *Computer vision: a modern approach* (Internat. ed., reprinted with corr ed.). Upper Saddle River, NJ: Pearson Prentice Hall. (OCLC: 248368757)
- Gershman, S. J., Horvitz, E. J., y Tenenbaum, J. B. (2015, julio). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245), 273–278. Descargado 2019-03-05, de <http://www.sciencemag.org/cgi/doi/10.1126/science.aac6076> doi: 10.1126/science.aac6076
- Gonzalez, R. C., y Woods, R. E. (2002). *Digital image processing* (2. ed., internat. ed ed.). Upper Saddle River, NJ: Prentice-Hall.
- Grigorescu, S. (2010, enero). Robust Machine Vision for Service Robotics. <http://elib.suub.uni-bremen.de/diss/docs/00011879.pdf>.
- ISO 8373:2012(en), *Robots and robotic devices — Vocabulary*. (s.f.). Descargado 2019-02-05, de <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>
- Karami, E., Prasad, S., y Shehata, M. (2017, octubre). Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. *arXiv:1710.02726 [cs]*. Descargado 2019-08-16, de <http://arxiv.org/abs/1710.02726> (arXiv: 1710.02726)
- Lowe, D. G., y others. (1999). Object recognition from local scale-invariant features. En *iccv* (Vol. 99, pp. 1150–1157).
- Matamoros, M., Rascon, C., Wachsmuth, S., Alexander, A. W., Kummert, J., Hart, J., ... St-Pierre, M. (2019). RoboCup@Home Rules and Regulations. , 92.
- Morrison, M. M. (1987). 54 INERTIAL MEASUREMENT UNIT. , 18.
- Murphy, R. (2000). *Introduction to AI robotics*. Cambridge, Mass: MIT Press.
- Ogata, K. (2004). *System dynamics* (4th ed ed.). Upper Saddle River, NJ: Pearson/Prentice Hall.
- Pernia, M. (2011, octubre). Conceptos Básicos de Máquinas de corriente continua..
- Pineda, L., Sucar, E., Savage, J., Aceves, A., Becerra, H., Fuentes, G., ... Rodríguez, A. (2017). Robótica de servicio. En *La computación en méxico por especialidades académicas* (p. 127-166). Asociación Mexicana de Computación.
- Pothos, E. M., y Wills, A. J. (Eds.). (2011). *Formal approaches in categori-*

- zation. Cambridge ; New York: Cambridge University Press.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... Ng, A. Y. (2009). ROS: an open-source Robot Operating System. En *ICRA workshop on open source software* (Vol. 3, p. 5). Kobe, Japan.
- Ramírez, L. G. C., Jiménez, G. S. A., y Carreño, J. M. (2014). *Sensores y Actuadores*. Grupo Editorial Patria. (Google-Books-ID: wMm3BgAAQBAJ)
- Rodríguez Villagra, O. (2014, noviembre). Modelos cognitivos formales: Una herramienta para la integración del conocimiento en psicología cognitiva y neurociencia cognitiva. *Actualidades en Psicología*, 28(117), 79–91. Descargado 2019-10-18, de <http://revistas.ucr.ac.cr/index.php/actualidades/article/view/14481> doi: 10.15517/ap.v28i117.14481
- Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., y Fujimura, K. (2002, septiembre). The intelligent ASIMO: system overview and integration. En *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 3, pp. 2478–2483 vol.3). doi: 10.1109/IRDS.2002.1041641
- Sarmiento, C., Savage, J., Juarez, A., Contreras, L., Pacheco, A., y Matamoros, M. (2019, abril). Feature detection using Hidden Markov Models for 3d-visual recognition. En *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (pp. 1–6). doi: 10.1109/ICARSC.2019.8733651
- Savage, J., LLarena, A., Carrera, G., Cuellar, S., Esparza, D., Minami, Y., y Peñuelas, U. (2008). ViRbot: A System for the Operation of Mobile Robots. En U. Visser, F. Ribeiro, T. Ohashi, y F. Dellaert (Eds.), *RoboCup 2007: Robot Soccer World Cup XI* (pp. 512–519). Springer Berlin Heidelberg.
- Savage, J., Martell, R., Estrada, H., Cruz, J., Negrete, M., Marquez, J., ... Cruz, J. (2018). *Pumas@Home 2019 Team Description Paper*.
- Savage, J., Rosenblueth, D. A., Matamoros, M., Negrete, M., Contreras, L., Cruz, J., ... Okada, H. (2019, abril). Semantic reasoning in service robots using expert systems. *Robotics and Autonomous Systems*, 114, 77–92. Descargado 2019-11-11, de <http://www.sciencedirect.com/science/article/pii/S0921889018302501> doi: 10.1016/j.robot.2019.01.007
- Savage, J., Vazquez, E., Silva, E., Hernandez, J., Sarmiento, A., y Nakayama, A. (2018). *DSPL Pumas@Home 2019 Team Description Paper*.

- Siciliano, B., y Khatib, O. (Eds.). (2008). *Springer handbook of robotics*. Berlin: Springer. (OCLC: ocn153562054)
- Solutions, A. K. (2007). *Robotics*. Hingham, Mass: Infinity Science Press. (OCLC: 255645881)
- Staddon, J. (2014). On Choice and the Law of Effect. *International Journal of Comparative Psychology*, 27(4). Descargado 2019-04-17, de <https://escholarship.org/uc/item/1tn9q5ng>
- Thorndike, E. L. (1927). The Law of Effect. *The American Journal of Psychology*, 39(1/4), 212–222. Descargado 2019-04-30, de <https://www.jstor.org/stable/1415413> doi: 10.2307/1415413
- Thrun, S. (2002). Particle Filters in Robotics. En *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence* (pp. 511–518). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Descargado 2019-09-19, de <http://dl.acm.org/citation.cfm?id=2073876.2073937> (event-place: Alberta, Canada)
- Torrente Artero, O. (2013). *Arduino: curso práctico de formación*. México D.F.: Alfaomega : Libros RC. (OCLC: 879738700)
- Yamamoto, T., Terada, K., Ochiai, A., Saito, F., Asahara, Y., y Murase, K. (2019, abril). Development of Human Support Robot as the research platform of a domestic mobile manipulator. *ROBOMECH Journal*, 6(1), 4. Descargado 2019-10-01, de <https://doi.org/10.1186/s40648-019-0132-3> doi: 10.1186/s40648-019-0132-3