

# Robótica de Servicio

Enrique Sucar, Yasmín Hernández  
Editores



ACADEMIA MEXICANA DE COMPUTACIÓN, A, C.

II

Robótica de Servicio

Coordinador: Luis Enrique Sucar Succar

Editores: Luis Enrique Sucar Succar, Yasmín Hernández Pérez

Segunda edición: 2019

Academia Mexicana de Computación, A.C.

Todos los derechos reservados conforme a la ley

ISBN: 978-607-97357-6-0

Corrección de estilo: Yasmín Hernández Pérez, Luis Enrique Sucar Succar

Diseño de portada: Mario Alberto Vélez Sánchez

Cuidado de la edición: Yasmín Hernández Pérez, Luis Enrique Sucar Succar

Este libro se realizó con el apoyo del CONACyT, Proyecto No. I1200/28/2019

Queda prohibida la reproducción parcial o total, directa o indirecta, del contenido de esta obra, sin contar con autorización escrita de los autores, en términos de la Ley Federal del Derecho de Autor y, en su caso, de los tratados internacionales aplicables.

**Editores:**

Luis Enrique Sucar Succar

Yasmín Hernández Pérez

**Con la colaboración de:**

Yasmín Hernández Pérez

Mauricio Matamoros

Marco Antonio Morales Aguirre

Marco Negrete

Luis Alberto Pineda Cortés

Caleb Rascón Estebané

David A. Rosenblueth

Jesús Savage Carmona

Sergio A. Serrano

Luis Enrique Sucar Succar



## Agradecimientos

La Amexcomp agradece a *Yasmín Hernández, Mauricio Matamoros, Marco Antonio Morales Aguirre, Luis Alberto Pineda Cortés, Caleb Rascón Estebané, Jesús Savage Carmona, Sergio A. Serrano y Luis Enrique Sucar Succar*, por sus aportaciones técnicas y científicas, así como por su entusiasmo y esfuerzo en la edición de este libro.

Asimismo, la Amexcomp agradece al Conacyt por su valioso apoyo para la creación de esta obra.



## Prólogo

Hace casi 100 años que el mundo conoció la palabra *Robot*. En 1920, Karel Čapek presentó el término en su obra *R.U.R. (Robots Universales Rossum)*, para referirse a entidades mecánicas que realizaban trabajo pesado. Desde entonces, hemos soñado con robots que nos ayuden en las tareas diarias, a ir a lugares peligrosos o lejanos y a realizar tareas repetitivas.

La ciencia ficción ha motivado nuestra fascinación por los robots, mostrándonos a seres autónomos e independientes capaces de comportarse como un ser humano. En el cine, los robots que toman conciencia de sí mismos han sido un tema favorito: desde el icónico HAL-9000, hasta Ava de *Ex-máquina*, pasando por R2-D2, David y Wall-e, encontramos robots que conversan con los humanos y que realizan actividades que sólo los humanos pueden hacer. Estos personajes logran que los humanos creen que están conversando con otro humano, e incluso siembran en los humanos la duda sobre su propia condición humana.

En el mundo real, los científicos han buscado construir robots que sean inteligentes, es decir, están en la búsqueda de robots que hablen, piensen, aprendan, decidan, incluso que simulen tener sentimientos. Es importante resaltar que el comportamiento inteligente en un robot no solamente implica imitar al ser humano, sino que incluye capacidades de predicción y toma de decisiones en ambientes complejos y con incertidumbre sin la intervención de un humano. Actualmente, los robots se utilizan ampliamente en diferentes sectores: salud, industria, educación, transporte, entre otros. Se aplican principalmente en tareas que requieren exactitud, y en trabajos peligrosos o rutinarios. Los robots se usan en exploraciones espaciales, cirugías, laboratorios y en la producción de bienes. Algunas aplicaciones incluyen la limpieza de residuos tóxicos, minería, y búsqueda y rescate de personas.

En México, el interés en la robótica surgió hace varias décadas, cuando se incluyeron las primeras materias relacionadas con inteligencia artificial en programas de licenciatura y posgrado, y también se usaron los primeros brazos robóticos en la industria mexicana. Posteriormente, se abrieron programas de estudios enfocados en robótica. Entre las actividades más vistosas y divertidas de la robótica se encuentran los torneos de robótica. Desde hace más de una década, profesores, investigadores y estudiantes han formado equipos para prepararse y participar en

competencias nacionales e internacionales. En nuestros días, México es uno de los principales competidores en torneos de prestigio, y ha obtenido los primeros lugares y reconocimientos en eventos internacionales. Sin embargo, aunque el interés y el entusiasmo sigue en aumento, hace falta difundir las oportunidades, beneficios y aplicaciones de la robótica, y sobre todo hace falta atraer estudiantes a este fascinante campo para aprovechar el potencial que tenemos en esta materia.

En 2016, la Academia Mexicana de Computación editó el libro *La computación en México por especialidades académicas*, en un esfuerzo por reunir en un único volumen, el estado del arte y de la práctica de las diferentes especialidades de la computación y presentar a los principales científicos de México. Dicho libro incluye el capítulo *Robótica de Servicio*, del cual nace este extraordinario libro. La 1a edición se publica en 2018 mientras que la 2a edición se publica en 2019.

La academia decide escribir este libro para tender un puente entre la comunidad científica, academia y sociedad, y así dar a conocer los avances de la investigación en robótica en México. Está dirigido a profesionistas de diferentes ramas y a estudiantes de preparatoria que busquen conocer los fundamentos de la robótica. Este libro busca ser una referencia de primera mano, incluso para profesores de nivel medio y medio superior que quieran alentar en sus estudiantes la curiosidad por los robots.

El campo de la robótica es muy amplio e involucra muchas disciplinas, por lo que es difícil incluir todos los aspectos y tipos de robots en un volumen. Por ello, el presente libro se enfoca en la robótica de servicio, ya que en este tipo de robots, la computación tiene una gran relevancia; además de ser un área con desarrollos importantes en nuestro país. Se describen los diferentes aspectos del diseño y programación de los robots de servicio, que se ilustran con ejemplos del trabajo de los grupos de investigación mexicanos, por lo que se presentan los robots mexicanos más conocidos. En cada capítulo se proporcionan ligas a sitios Web en donde se puede conocer más sobre los diversos temas. Muchas de las ligas incluyen videos de demostraciones de las capacidades de los robots.

El libro se compone de seis capítulos y tres apéndices. En el capítulo 1 se presenta brevemente la historia de la robótica y un panorama de los aspectos que intervienen en el diseño y programación de un robot. El capítulo 2 hace una descripción de las funciones y diseño de los robots de servicio. Del capítulo 3 al 5 se describen los diferentes niveles en la construcción de los robots de servicio. El



capítulo 6 plantea los retos de la robótica principalmente en el ámbito mexicano, y sugiere algunas líneas de acción para el progreso de esta área. El apéndice A presenta algunos de los grupos de investigación más importantes de nuestro país. El apéndice B ofrece una sinopsis de los principales torneos de robótica en México, y hace énfasis en el Torneo Mexicano de Robótica. Mientras que el apéndice C presenta el diseño y funcionamiento del robot *Justina*.

Este libro es el resultado del trabajo constante de la Amexcomp por difundir los logros y retos de la computación, en especial de su presidente, el Dr. Luis Enrique Sucar Succar y de su presidente fundador, el Dr. Luis Alberto Pineda Cortés. Así también es producto de la entusiasta participación de los doctores: Jesús Savage Carmona, Mauricio Matamoros, Marco Antonio Morales Aguirre, Caleb Rascón Estebané, Sergio A. Serrano, David A. Rosenblueth y Marco Negrete. De la misma manera, esta obra no sería posible sin el valioso apoyo del Conacyt.

Esperamos que este libro ayude a que muchos estudiantes se interesen en el tema y se conviertan en los roboticistas que México y el mundo necesita.

Yasmín Hernández



# Abreviaturas

<b>AEF</b>	Autómata de Estados Finitos .....	26
<b>AIRA</b>	<i>Acoustic Interactions for Robot Audition</i> .....	50
<b>BUAP</b>	Benemérita Universidad Autónoma de Puebla.....	80
<b>CIC</b>	Centro de Investigación en Computación .....	80
<b>CICESE</b>	Centro de Investigación Científica y de Educación Superior de Ensenada .....	80
<b>CIMAT</b>	Centro de Investigación en Matemáticas .....	40
<b>Cinvestav</b>	Centro de Investigación y de Estudios Avanzados.....	40
<b>CIDI</b>	Centro de Investigación en Diseño Industrial .....	59
<b>CLIPS</b>	<i>C Language Integrated Production System</i> .....	110
<b>DARPA</b>	<i>Defense Advanced Research Projects Agency</i>	
<b>DOF</b>	<i>Degree of freedom</i>	
<b>FPGA</b>	<i>Field Programmable Gate Array</i> .....	27
<b>FIRST</b>	<i>For Inspiration and Recognition of Science and Technology</i> .....	103
<b>FMR</b>	Federación Mexicana de Robótica.....	104
<b>GPS</b>	<i>Global Positioning System</i> .....	40
<b>GPU</b>	<i>Graphics Processing Unit</i> .....	16
<b>HSR</b>	<i>Human Support Robot</i> .....	96
<b>IA</b>	Inteligencia Artificial.....	80
<b>IFR</b>	<i>International Federation of Robotics</i> .....	57

<b>IIMAS</b>	Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas	25
<b>IMM</b>	<i>Interacting Multiple Model</i>	47
<b>IMU</b>	<i>Inertial Measurement Unit</i>	60
<b>INAOE</b>	Instituto Nacional de Astrofísica, Óptica y Electrónica	25
<b>INEEL</b>	Instituto Nacional de Electricidad y Energías Limpias	
<b>IOCA</b>	<i>Interaction-Oriented Cognitive Architecture</i>	30
<b>IPN</b>	Instituto Politécnico Nacional	40
<b>ITAM</b>	instituto Tecnológico Autónomo de México	40
<b>LAIDETEC</b>	<i>Laboratorio de Innovación y Desarrollo Tecnológico</i>	58
<b>LIDAR</b>	<i>Laser Imaging Detection and Ranging</i>	16
<b>MDP</b>	<i>Markov Decision Process</i>	31
<b>MIT</b>	<i>Massachusetts Institute of Technology</i>	18
<b>MOPED</b>	<i>Multiple Object Pose Estimation and Detection</i>	43
<b>MPC</b>	<i>Model predictive Control</i>	41
<b>NP</b>	<i>Non-deterministic Polynomial Time</i>	45
<b>PID</b>	<i>Proportional-integral-derivative</i>	111
<b>POMDP</b>	<i>Partially Observable Markov Decision Process</i>	31
<b>PRM</b>	<i>Probabilistics Roadmap</i>	51
<b>PWM</b>	<i>Pulse-Width Modulation</i>	
<b>RGB</b>	<i>Red Green Blue</i>	64
<b>RGBD</b>	<i>Red Green Blue Depth</i>	64
<b>ROS</b>	<i>Robot Operating System</i>	16
<b>RRT</b>	<i>Rapidly Exploring Random Tree</i>	44
<b>RyM</b>	Robótica y Mecatrónica	97
<b>RyMA</b>	Robótica y Manufactura Avanzada	99
<b>SIFT</b>	<i>Scale-Invariant Feature Transform</i>	39
<b>SLAM</b>	<i>Simultaneous Localization and Mapping</i>	39

<b>SPL</b>	<i>Standard Platform League</i> .....	80
<b>SSL</b>	<i>Small Size League</i> .....	80
<b>TMR</b>	Torneo Mexicano de Robótica.....	81
<b>UNAM</b>	Universidad Nacional Autónoma de México.....	8
<b>UPAEP</b>	Universidad Popular Autónoma del Estado de Puebla.....	80



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Referencias	5
<b>2. Robots de Servicio</b>	<b>7</b>
2.1. De robots industriales a robots de servicio	8
2.2. Aplicaciones	11
2.2.1. Robots en el hogar	11
2.2.2. Robots y adultos mayores	12
2.2.3. Robots en hospitales	13
2.2.4. Robots en oficinas	14
2.2.5. Otros usos	14
2.3. Capacidades básicas de robots de servicio	15
2.3.1. Limitantes	17
2.4. Paradigmas	18
2.5. Modelo conceptual y niveles de sistema	19
2.6. Conclusiones	21
2.7. Referencias	23
<b>3. Nivel Funcional</b>	<b>25</b>
3.1. Máquinas de estado finito	26
3.2. Especificación e interpretación de tareas	27
3.3. Procesos de decisión de Markov	31
3.4. Referencias	33
<b>4. Nivel de Dispositivos y Algoritmos</b>	<b>35</b>
4.1. Percepción	36
4.2. Acción motora	37
4.3. Mapeo, localización y control	38
4.3.1. Seguimiento	41
4.3.2. Reconocimiento y manipulación de objetos	42
4.3.3. Reconocimiento de personas	46

4.3.4. Monitoreo y vigilancia . . . . .	47
4.4. Audición robótica . . . . .	47
4.5. Representación del conocimiento y razonamiento . . . . .	50
4.6. Referencias . . . . .	53
<b>5. Nivel de Implementación</b>	<b>57</b>
5.1. Plataformas de robots de servicio . . . . .	57
5.1.1. Robots móviles . . . . .	59
5.1.2. Robots humanoides . . . . .	62
5.2. Sensores y actuadores . . . . .	63
5.2.1. Sensores . . . . .	63
5.2.2. Sensor RGBD . . . . .	64
5.2.3. Sensor láser . . . . .	65
5.2.4. Actuadores . . . . .	65
5.2.5. Motor de corriente directa . . . . .	66
5.2.6. Motor a pasos . . . . .	67
5.2.7. Servomotor . . . . .	67
5.3. Programas de apoyo . . . . .	68
5.4. <i>Robot Operating System</i> . . . . .	69
5.4.1. Funcionamiento de ROS . . . . .	70
5.4.2. Aprendiendo ROS . . . . .	72
5.5. Conclusiones . . . . .	72
5.6. Referencias . . . . .	73
<b>6. Retos y Perspectivas</b>	<b>75</b>
<b>A. Investigación en Robótica en México</b>	<b>79</b>
A.1. Grupo de Robótica del ITAM . . . . .	80
A.1.1. Robot ITAM-EKBot . . . . .	81
A.1.2. Simulador de robot AutoNOMOs . . . . .	82
A.1.3. Videos . . . . .	82
A.2. Grupo Golem . . . . .	83
A.2.1. Robot Golem . . . . .	84
A.2.2. Robot Golem II+ . . . . .	84
A.2.3. Robot Golem III . . . . .	85
A.2.4. Videos . . . . .	86
A.3. Grupo Markovito . . . . .	87
A.3.1. Robot Sabina . . . . .	88
A.3.2. Robot Markovito . . . . .	89
A.3.3. Robot Markovito Jr. . . . .	90
A.3.4. Videos . . . . .	91
A.4. Grupo Nanisha . . . . .	91
A.4.1. Robot Nanisha . . . . .	92



A.4.2. Robot Donaxi . . . . .	92
A.4.3. Videos . . . . .	93
A.5. Grupo Pumas . . . . .	93
A.5.1. Robot Justina . . . . .	96
A.5.2. Robot Takeshi . . . . .	96
A.5.3. Videos . . . . .	96
A.6. Grupo RyM . . . . .	97
A.6.1. Videos . . . . .	98
A.7. Grupo RyMA . . . . .	99
A.7.1. Videos . . . . .	100
<b>B. Torneos de Robótica en México</b>	<b>103</b>
<b>C. El robot <i>Justina</i></b>	<b>107</b>
C.1. Capa de entrada . . . . .	107
C.2. Capa de planeación . . . . .	109
C.3. Capa de representación del conocimiento . . . . .	110
C.4. Capa de ejecución . . . . .	111
C.4.1. Máquinas de estados . . . . .	112
C.5. Referencias . . . . .	114



# Capítulo 1

## Introducción

L. Enrique Sucar Succar, [INAOE](#)

Jesus Savage Carmona, Facultad de Ingeniería, [UNAM](#)

Luis A. Pineda Cortés, [IIMAS](#), [UNAM](#)

Yasmín Hernández, [INEEL](#)

La palabra *robot* apareció por primera vez en 1920 en la obra de teatro R.U.R. (*Robots Universales Rossum*) de Karel Čapek. Está relacionada con nuestra palabra *trabajo*.

La idea de máquinas que realicen trabajo físico, quizás para permitir que los humanos se dediquen a actividades más intelectuales, implica de inmediato la imitación de *acciones* humanas. Como resultado, la metodología de la robótica ha sido normalmente guiada por demostraciones físicas de máquinas que exhiban comportamientos similares a los de los humanos. El énfasis en la *construcción* física de máquinas, a diferencia de la obtención resultados teóricos, no tolera aproximaciones (como lo hacen las ciencias más teóricas). Los jueces, finalmente, son las leyes de la física.

La robótica opera entonces en un mundo que, por su complejidad, no podemos conocer completamente. Por un lado, pueden aparecer ventajas inesperadas, resultantes de aspectos que un enfoque más teórico no modelaría. Por ejemplo, es posible construir una máquina con forma similar a la del cuerpo humano, pero que camine descendiendo por un plano inclinado sin coordinar las piernas (Pfeifer &

Bongard, 2006). Por otro lado, las leyes físicas son jueces severos, de los que emergen dificultades ocultas en enfoques teóricos. Por ejemplo, un robot debe actuar en presencia de ruido en sus sensores.

La robótica como disciplina<sup>1</sup> se inició con la invención de los brazos autónomos para asistir la manufactura en líneas de producción, especialmente en la industria automotriz, con el robot Unimate en 1961.<sup>2</sup> Esta invención dio lugar al desarrollo de una industria de grandes dimensiones con gran impacto económico en el mundo.

A partir de estos robots iniciales, en conjunto con el desarrollo de las disciplinas relacionadas, como el procesamiento de señales, la teoría de control y la inteligencia artificial, durante las últimas décadas se han diseñado y construido una gran cantidad de robots que se pueden clasificar en dos dimensiones: i) autonomía y ii) complejidad del entorno.

El grado de autonomía de un robot depende de varios factores. Por un lado, dicha autonomía está determinada por la riqueza de los sensores del robot y por la variedad de sus mecanismos motores y mentales. Por otro lado, la autonomía es función de la variedad de los tipos de acciones que puede realizar un robot.

La complejidad del entorno, en cambio, corresponde a la variedad del ambiente que el robot es capaz de enfrentar e incluso transformar. Esta dimensión no se refiere a la extensión espacial donde los robots pueden funcionar e interactuar, sino a la cantidad y complejidad de entornos cualitativamente diferentes. Por ejemplo, aunque las hormigas, las cucarachas y los osos de agua son ubicuos en el planeta, sus microentornos locales, en relación con su escala, son similares en todos lados. En contraste, los seres humanos, por ejemplo, residen en entornos globales complejos, cualitativamente diferentes entre sí.

Se ha dicho que el grado de autonomía de un robot y su capacidad de actuar en entornos complejos están relacionadas con su inteligencia (Ashby, 1958).

De acuerdo con estas consideraciones, es posible describir el mapa de la robótica en el espacio definido por estas dos dimensiones, como se ilustra en la [Figura 1.1](#). En el origen están los robots con un solo tipo de acción en un entorno fijo, como los brazos industriales y en la esquina superior derecha los robots humanoides de la ciencia ficción. Asimismo, la gran mayoría de los robots actuales y potenciales estarían sobre una nube a lo largo de la diagonal principal ya que la complejidad

---

<sup>1</sup><https://en.wikipedia.org/wiki/Robotics>

<sup>2</sup><https://en.wikipedia.org/wiki/Unimate>

del robot debe ser coherente con la complejidad de su entorno, aunque habría muchas excepciones, como los robots completamente autónomos dirigidos a actuar en entornos de complejidad baja o moderada.

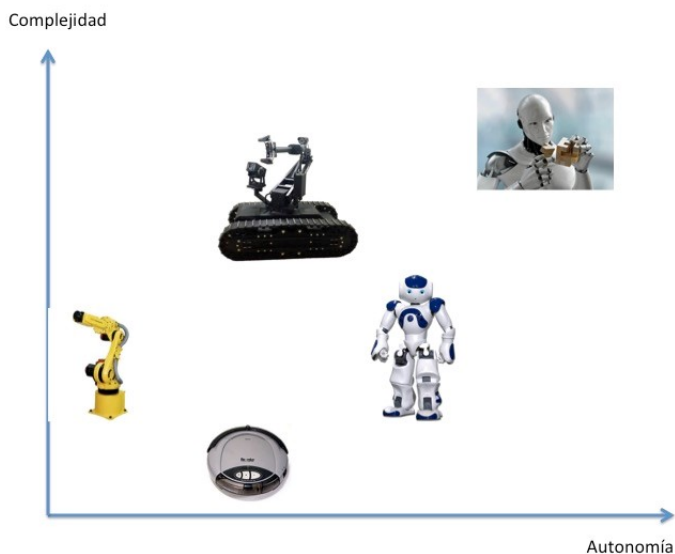


Figura 1.1: Variabilidad del ambiente versus autonomía. Se ilustran algunos ejemplos de robots en este espectro, desde robots con poca autonomía que operan en ambientes controlados (manipulador industrial, robot aspiradora) hasta los robots humanoides *inteligentes* que interactúan en una diversidad de ambientes.

Por otra parte, desde una perspectiva funcional y utilitaria, el espacio de los robots se puede dividir en tres grandes regiones o semiplanos paralelos verticales de izquierda a derecha: i) industriales, ii) controlados de manera remota y iii) completamente autónomos. La primera región corresponde a los robots de las líneas de producción cuyo entorno no varía o varía muy poco y su protocolo de acción está definido de antemano, por lo que sus procesos de percepción, toma de decisiones y acción intencional son muy limitados. La segunda incluye mecanismos con la capacidad de explorar y manipular el entorno pero la información que obtienen a través de sus sensores se envía a un operador humano quien los controla de manera remota y decide sus acciones. Estos robots tienen grandes aplicaciones actuales y potenciales, ya que pueden operar en ambientes peligrosos o inaccesibles para

los seres humanos, como inspeccionar el núcleo de un reactor nuclear o buscar sobrevivientes después de un desastre. La tercera categoría corresponde a robots capaces de percibir e interpretar el mundo y tomar decisiones de manera autónoma, principalmente en entornos complejos. No obstante, salvo algunas excepciones, como los vehículos autónomos, los robots de esta última categoría están todavía en los laboratorios de investigación. En la [Figura 1.1](#) se ilustran algunos ejemplos de robots en este espacio.

Para que un robot pueda llevar a cabo la tarea para la que fue construido, se requiere de diversas tecnologías trabajando en conjunto. Por ejemplo, si se le ordena a un robot de servicio traer una bebida de la cocina, es necesario, primero, reconocer el comando de voz y convertir los datos de audio emitidos por el usuario a texto. Después, dicho texto debe descomponerse en los pasos que deberá llevar a cabo para completar la tarea. El texto *tráeme un refresco de la cocina*, por ejemplo, implica que el robot navegue a la cocina, encuentre el refresco por medios visuales, que lo recoja, y que regrese a la posición del usuario para entregarlo. Para realizar estos pasos, el robot debe poder navegar, lo que a su vez precisa saber su posición en todo momento, calcular una ruta a la cocina, y moverse de tal manera que no choque con ningún obstáculo. También necesita un sistema de reconocimiento de objetos, que incluye no sólo la clasificación de los objetos que ve en su cámara, sino también la posición de cada uno de ellos. Asimismo, utiliza un sistema de manipulación para controlar sus brazos para poder recoger el refresco. Cada uno de los sistemas necesarios para ejecutar las acciones de este ejemplo (voz, navegación, visión, manipulación, etc.), representa un área de estudio por sí solo. En la robótica, dichos sistemas involucran un alto nivel de integración para que el robot lleve a cabo su tarea. Por ejemplo, el sistema de visión necesita saber del sistema de voz cual objeto localizar; la información resultante del sistema de visión es esencial para que el sistema de manipulación funcione; y el sistema de manipulación depende del sistema de navegación si el robot está lejos del refresco.

Se puede considerar que, habiendo establecido la funcionalidad global de un robot, este es el resultado de la integración de un gran gama de subsistemas todos dirigidos a contribuir con dicha funcionalidad global.

A la luz de lo anterior, este libro presenta al lector un punto de partida de los conceptos básicos de la robótica, desde la perspectiva de la funcionalidad de los robots de servicio, hasta la integración de la percepción y actuado. También se

describen las técnicas que se utilizan en los diversos subsistemas potenciales, desde el punto de vista funcional, de dispositivos y algoritmos, y de implementación. Finalmente se presentan algunos retos y perspectivas. En cada capítulo se describe el trabajo que se ha llevado a cabo en México como una forma de compendio del avance nacional en el campo de la robótica.

## 1.1. Referencias

- Ashby, W. R. (1958). Requisite variety and its implications for the control of complex systems. *Cybernetica*, 1 (2), 83-99.
- Pfeifer, R. & Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.





# Capítulo 2

## Robots de Servicio

Jesús Savage Carmona, Facultad de Ingeniería, [UNAM](#)

L. Enrique Sucar Succar, [INAOE](#)

Luis A. Pineda Cortés, [IIMAS](#), [UNAM](#)

Mauricio Matamoros, *University of Koblenz*

David A. Rosenblueth, [IIMAS](#), [UNAM](#)

Marco Negrete, Facultad de Ingeniería, [UNAM](#)

Este capítulo se enfoca en aquellos robots con un grado de autonomía media o alta diseñados para asistir a las personas y que son capaces de operar en los mismos espacios en los que se desenvuelven los seres humanos, es decir, entornos altamente dinámicos. La necesidad de coexistir con los humanos será un factor que determinará las capacidades y limitaciones de los robots de servicio, tal como se explica en la [Sección 2.2](#).

Antes de adentrarse de manera formal en el estudio de los robots de servicio, es pertinente establecer la diferencia entre estos y los robots tradicionales, conocidos también como robots industriales. Una vez establecida esta diferencia, se analizarán tres tipos de robots de servicio con base en su aplicación, a saber: I) destinados a trabajar en hogares, II) orientados a asistir a personas mayores, y III) especializados en auxiliar a enfermos; de los cuales se hablará en la [Sección 2.2](#).

Como un ejemplo detallado de un robot de servicio, en el [Apéndice C](#) se explica el funcionamiento de *Justina*, un robot desarrollado en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México ([UNAM](#)).

## 2.1. De robots industriales a robots de servicio

Desde sus orígenes, el hombre ha soñado con construir máquinas capaces de imitar ciertos aspectos del pensamiento humano consciente, como Talos, el autómatas de bronce programado por Hefestos para defender Creta, por no mencionar a la Galatea de Pigmalión, o a los robots guardianes de Ajatasatru. Esto llevaría a la creación de numerosos autómatas y otros prodigios mecánicos que hoy día siguen maravillando a los mejores ingenieros. Sin embargo, la automatización no sería posible sino hasta después de la revolución industrial con la llegada de las primeras máquinas eléctricas.

Las primeras máquinas autónomas se enfocaron principalmente en tareas de manipulación, repitiendo incansables tareas simples y repetitivas una y otra vez. Así nacen los primeros robots industriales en la década de los sesenta (véase [Figura 2.1](#)), en su mayoría brazos estacionados en un solo lugar que pronto se volverían mucho más rápidos, fuertes y precisos que los seres humanos. Durante décadas, la robótica hace a un lado el ideal literario del sirviente antropomórfico para enfocarse en aumentar la eficiencia de los ingenios mecánicos. No obstante, el principal factor limitante sigue siendo la percepción del entorno. Al entrar a un cuarto un humano construye un mapa mental de dicho cuarto, es decir, genera un modelo del mundo que lo rodea y utilizar este modelo (normalmente de forma inconsciente) para tomar decisiones. De igual forma, los robots requieren de una representación del entorno para funcionar, lo que requiere de complejos sensores para percibir, abstraer y modelar un gran número de variables mediante procesos computacionalmente costosos. La solución adoptada para superar esta limitante fue limitar el número de variables y minimizar en lo posible eventos inesperados, descartando ambientes dinámicos como hogares, oficinas, tiendas y hospitales. Los robots industriales encuentran entonces su nicho en fábricas, donde los entornos cambian poco y son por lo tanto más predecibles, pues estos robots sólo pueden operar eficientemente en entornos controlados (véase [Capítulo 1](#)).

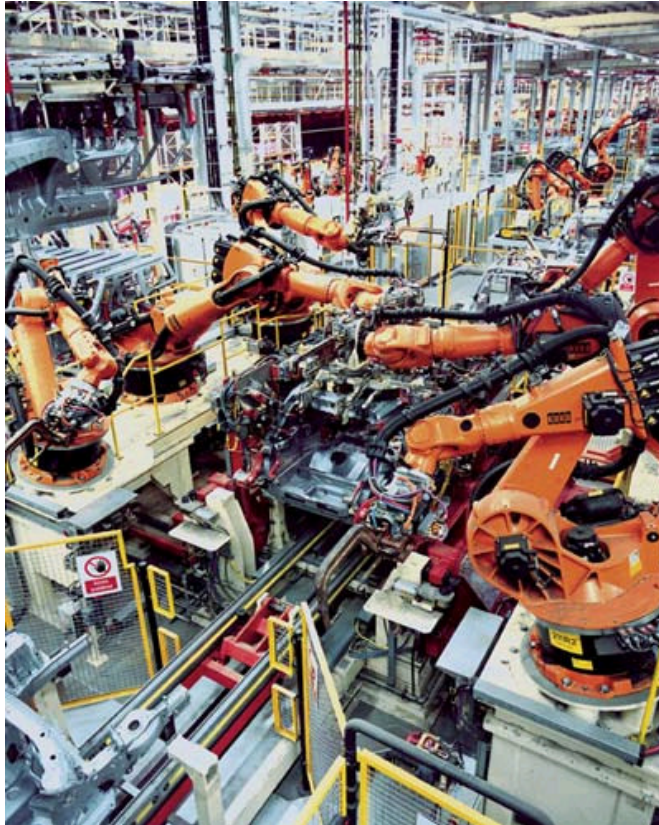


Figura 2.1: Robots manipuladores ensamblando automóviles

Con esto, la robótica industrial se separa del resto de la robótica a tal grado que las definiciones de robot tanto de la *International Standard Organization* en el estándar ISO 8373 como de la *Robot Institute of America* reducen el término robot a un manipulador.<sup>3</sup> Mientras tanto, los robots seguirían desarrollándose lentamente en las universidades hasta el final de la década de los noventa.

---

<sup>3</sup>El estándar ISO 8373 define robot como «Un manipulador automáticamente controlado, reprogramable, multiuso, programable en tres o más ejes, que pueden estar fijos en un lugar o movilizarse para ser usado en aplicaciones de automatización industrial»; mientras que la RIA (1979) considera a un robot como «un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de diversos movimientos programados para la realización de diversas tareas».

Nuevos sensores y computadoras más rápidas han permitido la construcción de robots capaces de realizar trabajo en entornos cada vez más dinámicos. Uno de dichos avances fue la idea revolucionaria de robots carentes de entornos premodelados que, en cambio, usan al entorno mismo como el propio modelo del mundo. Otro avance importante fue la incorporación de modelos probabilísticos que permiten al robot predecir qué sucederá a continuación con base en sus observaciones y elegir el plan de acción con mayor probabilidad de éxito. Estos avances permitieron el desarrollo de bases móviles autónomas, dando origen a los primeros robots móviles. Al mismo tiempo, el nuevo poder de cómputo disponible permitió probar, fuera del laboratorio y en tiempo real, algoritmos de control e inteligencia artificial (planeación, visión computacional, reconocimiento de voz, etcétera) que llevaban décadas sin ser explotados. En pocos años, dispositivos de corte, succión, inyección e incluso brazos manipuladores fueron acoplados a bases móviles y empezaron a recorrer el mundo para tratar de realizar sus tareas, abandonando los espacios controlados de las fábricas y empezando poco a poco a adentrarse en nuestros entornos para brindar asistencia. Nacen así los robots de servicio.

Este progreso en la robótica ha sido fulgurante, al grado que se espera, en un futuro cercano, una demanda masiva de los robots de servicio, para el cual se provee la siguiente definición:

*Un robot de servicio es una máquina móvil reprogramable autónoma o semiautónoma diseñada para operar en entornos dinámicos de manera segura, confiable y robusta, y ser capaz realizar tareas específicas.*

El objetivo de estos robots es reducir o eliminar el trabajo físico de las personas en lugares como casas, oficinas y tiendas, es decir, entornos propensos a cambiar de forma abrupta. Es importante resaltar que, a diferencia de los robots industriales, los robots de servicio por lo regular navegan a través de los espacios que les han sido asignados. Su autonomía consiste en la habilidad de tomar decisiones basadas en una representación interna del mundo. Así como televisores, radios, computadoras y los teléfonos celulares se incorporaron a la vida cotidiana, es factible suponer que los robots de servicio también lo harán en algún momento, y llegarán a ser familiares. La [Figura 2.2](#) muestra el tipo de robots que se espera tener en el futuro no muy lejano. Por ejemplo, los robots desempeñarán un papel importante en diferentes actividades físicas e incluso en actividades de

compañía para los ancianos. Los dispositivos robóticos ayudarán a las personas con discapacidad a moverse, apoyarán a soldados, trabajadores de la construcción y profesionales médicos. Los robots darán mantenimiento a máquinas industriales peligrosas y manipularán materiales peligrosos. Permitirán a los trabajadores de la salud diagnosticar y tratar a pacientes que pudieran estar a kilómetros de distancia, y serán una característica central de los sistemas de seguridad y las operaciones de búsqueda y rescate. (Gates, 2006).

## 2.2. Aplicaciones

Las tendencias tecnológicas de vanguardia indican que en un futuro cercano una casa contará con varios robots, cada uno especializado para resolver una tarea específica.

### 2.2.1. Robots en el hogar

Se tendrán robots que trabajen en zonas aledañas a la casa, por ejemplo un robot se dedicará a cortar el pasto mientras que otro hará rondines de vigilancia. Así mismo, dentro de la casa habrá robots móviles pequeños que limpien, aspiren y trapeen el piso, mientras que robot fijo podría encargarse del lavado y planchado de la ropa. Al mismo tiempo, un robot tipo humanoide ayudará a las personas a traer y llevar objetos de un lugar a otro. Dichos robots humanoides tendrán capacidades de comunicarse con los miembros de la casa usando lenguaje natural y planeará las acciones y los movimientos necesarios para hacer sus funciones.

Obviamente un robot humanoide podría tomar la aspiradora y realizar la limpieza. Sin embargo, este tendría en principio las mismas limitantes que un ser humano, mientras que una roboaspiradora podría realizar la misma función de manera más eficiente (por ejemplo pudiendo entrar abajo de los muebles) y con un menor consumo de energía, dejando libre al robot antropomórfico para tareas más complejas como cocinar, hacer la compra, o pasear al perro.

Finalmente, los robots pueden ayudarnos haciendo nuestros espacios más seguros, desde recoger objetos en el piso para evitar tropezones hasta verificar que hemos cerrado las llaves del gas.



Figura 2.2: Robots del futuro. Las nuevas máquinas serán tan especializadas y ubicuas, y a medida que estos dispositivos se vuelven asequibles para los consumidores, tendrán cada vez mayor impacto en la vida diaria.

### 2.2.2. Robots y adultos mayores

Otra función importante de los robots humanoides es la de asistir a personas de edad avanzada o a enfermos. Por ejemplo, un robot puede llevar un registro de las medicinas que debe tomar una persona, así como recordarle cuándo debe tomarlas y verificar que tome las dosis prescritas. Este tipo de recordatorios son

una función fundamental para atender a adultos mayores con enfermedad de Alzheimer o demencia senil, quienes pueden olvidar en dónde están, el camino para llegar a algún lugar deseado, o cómo operar aparatos domésticos, como lavadoras y secadoras. De igual manera, e incluyendo al cualquier tipo de usuario, un robot podría fácilmente llevar la agenda de su dueño, recordándole por ejemplo un cumpleaños, que debe hacer ejercicio o hacer una llamada telefónica.

Muchas veces, los adultos mayores y los enfermos no necesitan otra cosa que compañía. Es aquí donde entran los robots conversacionales, mismos que explotan nuestra propensión a atribuir características humanas a animales y cosas para dar mayor calidez a las interacciones y aliviar la soledad del usuario. Por ejemplo, el uso de investigación de punta en análisis de sentimientos en lenguaje natural, permitirá a los robots detectar cuando una persona se siente triste, deprimida o enferma, enviar una notificación a otra persona a cargo tal como un familiar o al médico de confianza (si así estuviere configurado) para que este pueda enlazar una video llamada mediante el robot mismo.

Incluso hoy día existen robots de propósito específico para asistir a los ancianos. Por ejemplo, los robots-andadera permiten a los ancianos moverse con mayor facilidad en rampas, adaptando su velocidad y frenado a la pendiente para reducir el esfuerzo de la persona, además de reconocer la voz del usuario y navegar de manera autónoma desde su estación de recarga cuando son llamados.

### 2.2.3. Robots en hospitales

El alcance de los robots de servicio no se ve limitado única y exclusivamente al hogar, estos también tienen múltiples oportunidades de ser útiles en clínicas y hospitales. De inmediato salta a la vista el transporte de medicinas, de resultados de análisis, y de alimentos, aunque en unos años es posible que los robots aprendan a leer el instrumental médico e incluso puedan monitorear los signos vitales de los pacientes. A la fecha existen robots que incorporan cámaras infrarrojas para detectar personas, mismas que podrían recalibrarse para saber si una persona tiene fiebre o presenta patrones anormales de temperatura en el cuerpo que puedan ayudar en el diagnóstico.

Robots más especializados con mucha mayor fuerza podrían ayudar en rehabilitación, así como ayudando a caminar o a hacer ejercicio a los enfermos. De igual manera, un robot podría actuar localmente reemplazando a un médico que

se encuentra lejos, inclusive complementando o aumentando las capacidades del médico gracias a sus sensores, es decir, fungiendo como intermediario en consultas vía telepresencia. A este respecto, es importante señalar que existen una gran cantidad de estudios que utilizan procesamiento de lenguaje natural y sistemas expertos como recursos en el desarrollo de sistemas de diagnóstico médico computarizado. En principio, un robot de servicio podría incorporar estas funciones y diagnosticar enfermedades menores en pacientes, además de suministrar recetas, medicamentos e incluso aplicar inyecciones, reduciendo así la carga de trabajo de los médicos y reservando su tiempo para atender casos más complejos o graves. No obstante, antes habría de resolverse la cuestión ética subyacente dado que un robot no puede hacerse responsable por las consecuencias de un mal diagnóstico.

#### 2.2.4. Robots en oficinas

Otro espacio en los que los robots podrían ser de utilidad es en las oficinas. Igual que en el hogar, un grupo de robots especializados podría encargarse de tareas de limpieza. Al mismo tiempo, los robots de servicio de propósito general se encargarían de de otras actividades más complejas como envío y distribución de paquetes, sacar copias e incluso organizar archivos físicos, por no mencionar el acarreo de café y bocadillos.

En otros tipos de oficinas como las gubernamentales donde se brinda atención al cliente, un robot de servicio podría encargarse de agendar citas, recibir clientes y guiarlos al departamento u oficina correspondiente, o incluso dar informes sobre los trámites y procedimientos a realizar de manera similar a como agentes inteligentes automatizados atienden en los *call centers* hoy día.

#### 2.2.5. Otros usos

Un robots de servicio de propósito general puede asistir a las personas en virtualmente cualquier entorno donde se desenvuelven los seres humanos. Sus aplicaciones incluyen comprando o como vendedores en supermercados y centros comerciales, sirviendo como meseros en restaurantes y cafés, acomodando objetos en tiendas, bibliotecas y librerías, cuidando y entreteniendo niños en guarderías, guiando visitantes en museos, o tan solo manteniendo los espacios limpios y en buen estado.



## 2.3. Capacidades básicas de robots de servicio

Las capacidades básicas de un robot pueden darse desde varias perspectivas y niveles de abstracción. A fin de evitar ambigüedades, estas capacidades serán divididas en dos grupos: I) un conjunto de facultades necesarias definidas en abstracto cuya implementación puede llegar a ser muy compleja pero que son afines a todos los robots de servicio doméstico, y II) un conjunto de elementos atómicos programáticos concretos pero no necesariamente obligatorios y que pueden ser especializados o generalizados según convenga a la aplicación.

Para realizar cualesquiera de las actividades mencionadas en la [Sección 2.2](#), un robot de servicio deberá tener las siguientes características:

- a) **Reacción:** Es imprescindible que un robot pueda reaccionar de forma oportuna, segura y apropiada ante imprevistos, adaptándose a cambios en el entorno. Es decir, deberá contar con un conjunto de comportamientos reactivos auxiliares, especialmente en lo tocante a la evasión de obstáculos y a preservar la integridad física tanto del robot como de las personas.
- b) **Planeación:** Un robot debería poder resolver las tareas de manera inteligente o, dicho de otra manera, elaborar un plan que traduzca objetivos de alto nivel en acciones concretas ejecutables por el robot y, preferentemente de manera óptima. Por ejemplo un robot aspiradora tendrá que planear una ruta óptima para limpiar toda la superficie de una habitación en el menor tiempo posible y evitando pasar dos veces por el mismo lugar a menos que sea necesario. En contraste, para robot de servicio de propósito general la planeación de rutas de movimiento o manipulación será sólo un componente asumido como parte de la ejecución de tareas más complejas.
- c) **Aprendizaje:** Es deseable que un robot de servicio utilice su experiencia (procedimientos probados y optimizados con anterioridad) para solucionar situaciones similares. Esto quiere decir que un robot debería poder utilizar planes previamente generados como base para crear nuevos planes, ya sea modificándolos o combinando varios para formar un plan de mayor complejidad. A este respecto es importante destacar que un robot sólo podrá aprender aquello que pueda expresarse en términos de lo que el robot puede percibir y hacer.

Nótese que tanto la planeación de acciones como el aprendizaje dependen de la descomposición de tareas complejas en tareas mucho más sencillas. Es innecesario reiterar que todo proceso de descomposición tiene un límite, lo que implica que todo robot deberá contar con un conjunto de elementos atómicos a partir de los cuales se construyen todos los planes y el conocimiento del robot. Este conjunto de elementos atómicos programáticos concretos al que denominaremos primitivas (Misra, Sung, Lee & Saxena, 2016; Spangenberg & Henrich, 2015), pero que puede ser referido en la literatura especializada también como funciones o núcleos de acción (Pomarlan, Koralewski & Beetz, 2017), forma parte integral del software del robot y determina lo que este puede hacer y percibir, estando por tanto intrínsecamente relacionadas a las tareas para las cuales está diseñado el robot.

La [Tabla 2.1](#) presenta algunas primitivas comunes en robots de servicio de propósito general. Se asume un robot móvil con configuración de par diferencial, un brazo manipulador con efector final tipo *gripper*, cámara RGBD y micrófono acoplados sobre una unidad de paneo-cabeceo (pan-tilt), un sensor de rango por láser, *Laser Imaging Detection and Ranging* (LIDAR) y bocinas. Por fines didácticos, las primitivas fueron agrupadas en tres categorías:

- I) *acción*, para aquellas primitivas que interactúan con el entorno o cambian el estado físico del robot;
- II) *percepción*, para las primitivas que representan aquello que el robot puede percibir; y
- III) *razón* que comprende las primitivas que operan sobre planes y modifican el conocimiento del robot, incluyendo representaciones internas y el estado del mundo.

Normalmente, las implementaciones reales agrupan las primitivas en módulos organizados ya sea por su complejidad o por compartir acceso a recursos físicos, por ejemplo, la *Graphics Processing Unit* (GPU), o frecuentemente, nodos de *Robot Operating System* (ROS). Es necesario aclarar, no obstante, que estas primitivas son meramente funcionales y carecen de valor semántico. Esto quiere decir que el planeador de acciones deberá tener asociada para cada primitiva un conjunto de reglas de funcionamiento de mundo (funciones lambda, dependencias conceptuales, probabilidades de transición espacio-estado, etc.) que garanticen consistencia y congruencia en el modelo del mundo y base de conocimientos cada vez que una primitiva es ejecutada o sus consecuencias evaluadas a fin de armar un plan.

Tabla 2.1: Ejemplos de primitivas comunes en robots de servicio de propósito general

Acción	Percepción	Razón
ir a	detectar obstáculo	evaluar
mirar a	detectar persona	re-planear
tomar objeto	reconocer objeto	memorizar persona
colocar objeto	reconocer persona	memorizar objeto
abrir puerta	identificar gesto	memorizar lugar
abrir gripper	escuchar	memorizar plan
estrechar mano	localizar sonido	planear trayectoria
hablar	medir batería	olvidar persona

Por último, es importante mencionar que las capacidades básicas incluyen interacciones llamadas *positivas*, que contribuyen a la realización de los deseos de un ser humano, e interacciones *negativas*, aquellas que van en contra de los deseos humanos, tales como prevenir un delito e intentos de dañar al mismo robot.

### 2.3.1. Limitantes

La autonomía y adaptabilidad necesaria para desempeñar sus funciones impone varias restricciones a la cantidad de recursos que tendrán disponibles los robots de servicio durante su operación, lo que se traduce en importantes desafíos de ingeniería y abre un sin fin de áreas de estudio para la ciencia. Por ejemplo, al requerir desplazarse en su entorno, un robot de servicio no podrá estar conectado de forma permanente a una infraestructura de soporte, por lo que la potencia disponible para su operación estará limitada por la cantidad de energía almacenada en su batería. Así mismo, el acceso a recursos de cómputo remotos podría verse interrumpido o experimentar problemas de latencia mientras el robot se desplaza, por lo que todo proceso crítico deberá ejecutarse siempre dentro del robot mismo.

Por otro lado, es importante recordar que el entorno compartido impondrá restricciones que limitarán las capacidades de un robot de servicio, especialmente en materia de seguridad. Por ejemplo, un robot de servicio no debería desplazarse mucho más rápido que una persona al caminar y, en general, ninguna de sus partes debería moverse de manera brusca, y aplicando siempre la menor cantidad de fuerza necesaria. A este respecto, el uso de actuadores hidráulicos y neumáticos deberá

evitarse a pesar de ser mucho más fuertes que los electromecánicos por cuestiones de seguridad y confort. Este tipo de actuadores son considerados bloqueantes, es decir el torque aplicado no disminuye (o disminuye muy lentamente) al cortar el suministro de corriente, además de el ruido generado por los sistemas neumáticos suelen ser molesto para muchas personas. Lo mismo aplica para los sensores. El uso de ciertos tipos de láser o luces ultra brillantes podría no sólo molestar sino incluso lastimar a las personas, por lo que su uso debe ser cuidadosamente estudiado. Finalmente, el entorno compartido con los humanos impondrá restricciones de diseño a los robots. Por ejemplo, el ancho de un robot deberá considerar al estándar de la puerta más estrecha. De manera similar, el espacio manipulable deberá ser afín al utilizado por las personas, mimo que varía de país a país.

## 2.4. Paradigmas

En el desarrollo de los robots de servicio, se observan tres paradigmas que definen las arquitecturas de robots móviles:

- **Tradicional:** El primer paradigma, llamado *tradicional* y desarrollado en los años sesenta, utiliza modelos simbólicos para representar el medio ambiente y el uso de técnicas básicas de inteligencia artificial (búsquedas en árboles como  $A$ ) representaciones espacio-estado para la planeación de acciones. Un ejemplo de este tipo de arquitecturas es la del robot Shakey<sup>4</sup>, desarrollado en la Universidad de Stanford.
- **Reactivo:** Veinte años después, en el *Massachusetts Institute of Technology* (MIT) de los años ochenta, Rodney Brooks propuso un paradigma reactivo donde varios módulos responden a estímulos específicos generando salidas inmediatas las cuales son combinadas o seleccionadas por un arbitro para generar lo que el robot debe hacer bajo determinadas circunstancias. Nótese que en contraste con el paradigma tradicional, no se cuenta con una representación del medio ambiente, ni tampoco con módulos de planeación de acciones ni movimiento, y los módulos pueden utilizar máquinas de estados, redes neuronales o campos potenciales. Un ejemplo de este tipo de arquitecturas es la del robot Roomba<sup>5</sup>, desarrollado por la empresa iRobot.

---

<sup>4</sup><http://www.ai.sri.com/shakey/>

<sup>5</sup><https://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx>

- **Probabilístico:** El paradigma más reciente, llamado *probabilístico o estocástico*, parte de que tanto la percepción como los movimientos del robot pueden ser modelados como variables aleatorias y por tanto ser aproximados y manipulados utilizando técnicas probabilísticas. Un ejemplo de estas arquitecturas es la del automóvil autónomo que obtuvo el primer lugar en el [DARPA Grand Challenge del 2005](#), desarrollado por Sebastian Thrun de la Universidad de Stanford (Thrun et al. [2006](#)).

No obstante, ningún sistema es perfecto y pronto resultó evidente que un enfoque híbrido era la manera más inmediata de solucionar los problemas intrínsecos de cada uno de estos enfoques. Por ejemplo, los enfoques tradicionales requieren de un experto y requieren de mucho tiempo para su desarrollo, presentando además grandes problemas para manejar datos numéricos carentes de significado como nubes de puntos. En contraste, los sistemas reactivos y los probabilísticos son relativamente fáciles de desarrollar, simples y manejan con facilidad datos no simbólicos, pero tienden a estancarse en mínimos locales y es muy difícil estructurar con ellos planes complejos. Luego entonces es común toparse con sistemas donde los comportamientos reactivos siguen un paradigma reactivo con un árbitro probabilístico, los datos de entrada son normalmente procesados por sistemas estocásticos que realizan abstracción simbólica, y finalmente la planeación es orquestada por un sistema experto bajo paradigma tradicional. Un ejemplo de estas hibridaciones se puede encontrar tanto en el software Siri de Apple (Bellegarda, [2014](#)), como en el del robot *Justina* de la Facultad de Ingeniería de la [UNAM](#) (véase [Apéndice C](#)). Sin embargo, es importante recalcar que las nuevas tendencias de investigación se enfocan en el uso de redes neuronales de aprendizaje profundo (enfoque probabilístico) para realizar la planeación de acciones, presentando una alternativa que promete ser robusta y tolerante a fallas pero que requiere de mucho poder de cómputo.

## 2.5. Modelo conceptual y niveles de sistema

Desde el punto de vista de su uso, diseño y construcción, los robots se pueden estudiar desde tres perspectivas distintas ordenados de forma decreciente por su nivel de abstracción a las que nos referiremos en adelante como niveles de sistema,

a saber: I) funcional, II) de dispositivos y algoritmos, y III) de implementación (Pineda, Rodríguez, Fuentes, Rascon & Meza, 2015).

El nivel funcional estudia a los robots desde la perspectiva de sus usuarios, los seres humanos o usuarios que interactúan con el robot. En este nivel es relevante qué servicios ofrece el robot en términos prácticos, cómo se estructuran las tareas que puede realizar y cuál es la estructura de la comunicación o la interacción con los agentes en su entorno. Supóngase que fuera posible comprar un robot de servicio en una agencia de manera similar a como se compra un automóvil. Dicho robot incluiría un manual de usuario con las especificaciones de sus funciones básicas y comportamientos, así como ejemplos sobre cómo instruir al robot en la combinación de estas funciones para que realice tareas complejas. Este manual hipotético correspondería con la especificación del nivel funcional. Luego entonces, se puede decir que el nivel funcional impacta directamente en la arquitectura del robot al determinar cómo se utilizarán los dispositivos, qué algoritmos se seleccionarán, y la forma en que el robot interactuará con otros agentes, afectando la respuesta del mismo a los eventos que ocurren en su entorno, tanto naturales como los producidos por la acción de otros agentes de manera intencional.

El nivel de dispositivos y algoritmos corresponde al diseño y construcción de artefactos robóticos junto con los algoritmos o programas que los habilitan: brazos con manos capaces de tomar objetos frágiles con la fuerza suficiente para sostenerlos pero sin romperlos; cabezas con cámaras y algoritmos de visión para reconocer objetos y determinar su posición y orientación, que en conjunto con los brazos y las manos habiliten al robot para tomar objetos; plataformas móviles o piernas que le permitan al robot trasladarse de una ubicación a otra, asociados a algoritmos que le permitan construir un mapa de su entorno y moverse evadiendo obstáculos; dispositivos y algoritmos que lo habiliten para interactuar o comunicarse a través del lenguaje hablado, o reconocer ruidos y voltear hacia la fuente sonora y ponerla en su foco de atención. En este nivel también se incluyen sistemas de representación del conocimiento que le permiten hacer inferencias conceptuales y sistemas de razonamiento deliberativo para hacer diagnósticos, tomar decisiones y planear sus acciones futuras. Este nivel de sistema es el que se asocia más directamente con la práctica de la robótica, o de manera más coloquial, con lo que hacen los científicos y tecnólogos *roboticistas* o los *roboteros*. Asimismo, muchos dispositivos y algoritmos en este nivel se apoyan en técnicas de análisis de señales, reconoci-

miento de patrones, aprendizaje de máquina y teoría de control, entre otras, por lo que desde la perspectiva de la robótica estas disciplinas se subordinan al nivel algorítmico.

Finalmente, el nivel de implementación corresponde a los programas de apoyo que se requieren para habilitar los diversos dispositivos y funciones, pero que no son tecnología robótica propiamente. Se pueden mencionar aquí los sistemas operativos y sistemas de comunicaciones que se utilizan para integrar las partes físicas o computacionales del robot como un todo. Esta tarea la realizan programadores de sistemas como apoyo a los roboticistas.

Todos los robots se pueden conceptualizar en términos de estos tres niveles de sistema pero es frecuente que sólo el nivel de dispositivos y algoritmos se considere explícitamente, en particular en el diseño y construcción de robots simples. Sin embargo, cuando el robot está constituido por varios dispositivos y requiere de varios algoritmos complejos, que tienen que operar de manera coordinada, es indispensable distinguir los tres niveles de manera explícita y abordarlos con metodologías específicas y grupos de investigación y desarrollo enfocados a los niveles funcional y de implementación, de manera adicional a los grupos enfocados a cada una de las especialidades en el nivel de dispositivos y algoritmos.

## 2.6. Conclusiones

En este capítulo se establecieron las diferencias entre los diversos tipos de robots que existen, y con base en estas se propuso una definición que sirve de base para el estudio de los robots de servicio de propósito general en este capítulo y los subsiguientes. Posteriormente, se detallaron las posibles aplicaciones de los robots de servicio (recordemos que estos aún no son un producto final sino que se encuentran en una etapa temprana de investigación y desarrollo), las capacidades de los mismos y, finalmente, los paradigmas bajo los cuales se estudian. Aquí se presenta un breve resumen con las ideas principales introducidas en éste capítulo:

1. Los primeros **robots industriales** son brazos reprogramables estacionados en un solo lugar enfocados principalmente en tareas simples y repetitivas.

2. Un **robot de servicio** es una máquina móvil reprogramable autónoma diseñada para operar en entornos dinámicos mientras realiza tareas específicas.
3. En el futuro, los hogares podrían contar con **múltiples robots de servicio**, cada uno especializado en una tarea particular.
4. La capacidad de **interactuar con humanos** es un aspecto fundamental para los robots de servicio doméstico.
5. El **objetivo de los robots de servicio** es reducir o eliminar el trabajo físico de las personas en entornos propensos a cambios abruptos.
6. Entre las **aplicaciones** que tienen los robots de servicio se encuentran el cuidado de los adultos mayores, personas con discapacidad y enfermos, además de asistir en las labores del hogar y la oficina.
7. Las **capacidades básicas** fundamentales que debe tener un robot de servicio son las de **reacción, planeación y aprendizaje**.
8. Un **robot reactivo** es aquel capaz de responder de manera oportuna ante imprevistos.
9. La **planeación de acciones** consiste en descomponer la tarea asignada en una secuencia de acciones no especializables conocidas como **primitivas**.
10. La **imprevisibilidad** de los entornos dinámicos hace necesario que un robot de servicio abstraiga elementos clave de sus acciones pasadas a fin de desempeñarse mejor en el futuro, un proceso conocido como **aprendizaje**.
11. Entre los factores **limitantes** de un robot de servicio se encuentran su hardware, software, y las restricciones de seguridad impuestas por el entorno.
12. Los **paradigmas** bajo los cuales se estudian a los robots de servicio son el paradigma **tradicional**, el paradigma **reactivo**, el paradigma **probabilístico**, y, finalmente, el paradigma **híbrido**.
13. Los **niveles** desde los cuales se pueden estudiar a los robots de servicio son el nivel **funcional**, el nivel de **dispositivos y algoritmos**, y el nivel de **implementación**.



Con estas bases establecidas, es posible proceder con las formalizaciones del estudio de los robots de servicio que se presentan en el siguiente capítulo.

## 2.7. Referencias

- Bellegarda, J. R. (2014). Spoken Language Understanding for Natural Interaction: The Siri Experience. En *Natural Interaction with Robots, Knowbots and Smartphones* (pp. 3-14). Springer.
- Gates, B. (2006). A Robot in Every Home: Overview/The Robotic Future. *Scientific American*. Recuperado desde <https://www.scientificamerican.com/article/a-robot-in-every-home-ove/>
- Misra, D. K., Sung, J., Lee, K. & Saxena, A. (2016). Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research*, 35(1-3), 281-300.
- Pineda, L. A., Rodríguez, A., Fuentes, G., Rascon, C. & Meza, I. V. (2015). Concept and Functional Structure of a Service Robot. *International Journal of Advanced Robotic Systems*, 12(2), 6. doi:10.5772/60026
- Pomarlan, M., Koralewski, S. & Beetz, M. (2017). From Natural Language Instructions to Structured Robot Plans. En G. Kern-Isberner, J. Fürnkranz & M. Thimm (Eds.), *KI 2017: Advances in Artificial Intelligence* (pp. 344-351). Cham: Springer International Publishing.
- Spangenberg, M. & Henrich, D. (2015). Grounding of actions based on verbalized physical effects and manipulation primitives. En *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (pp. 844-851). IEEE.
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., ... Hoffmann, G. et al. (2006). Stanley: The robot that won the DARPA Grand Challenge. *Journal of field Robotics*, 23(9), 661-692.



# Capítulo 3

## Nivel Funcional

L. Enrique Sucar Succar, [INAOE](#)

Jesús Savage Carmona, Facultad de Ingeniería, [UNAM](#)

Luis A. Pineda Cortés, [IIMAS](#), [UNAM](#)

En este capítulo se presentan tres métodos para organizar y estructurar robots de servicio en el nivel funcional: i) máquinas de estado finito, ii) lenguajes de especificación de tareas y iii) procesos de decisión de Markov. Estos enfoques han sido adoptados y desarrollados por los grupos mexicanos que trabajan con robots de servicio. El primer método ha sido adoptado por el grupo Pumas<sup>6</sup> de la Facultad de Ingeniería de la [UNAM](#) y se implementa en el robot *Justina*. El segundo enfoque fue desarrollado por el Grupo Golem<sup>7</sup> del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas ([IIMAS](#)), de la [UNAM](#) y se ha implementado en los robots Golem, Golem-II+ y Golem-III. Y el tercer método ha sido adoptado y desarrollado por el grupo Markovito<sup>8</sup> del Instituto Nacional de Astrofísica, Óptica y Electrónica ([INAOE](#)), y se ha implementado en los robots Markovito y Sabina. Estos robots se muestran en la [Figura 3.1](#)

---

<sup>6</sup><https://biorobotics.fi-p.unam.mx/>

<sup>7</sup><http://golem.iimas.unam.mx/>

<sup>8</sup><https://ccc.inaoep.mx/~markovito/>

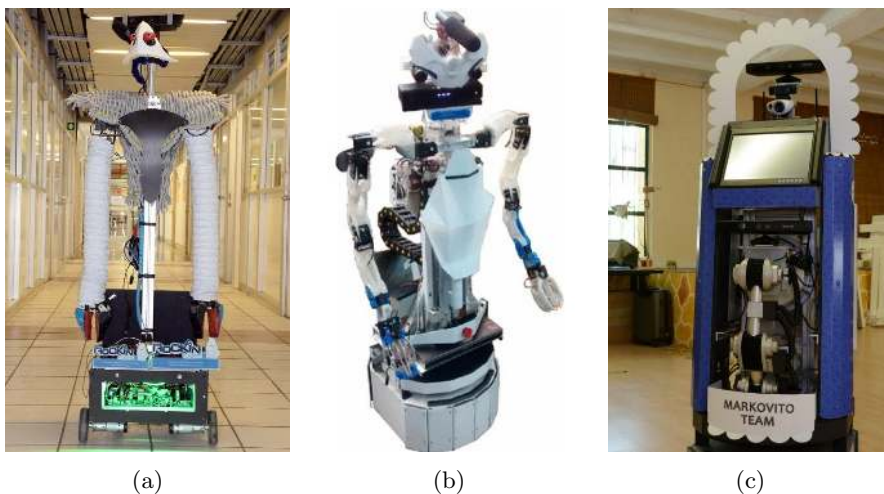


Figura 3.1: Robots de servicio desarrollados en México: (a) Justina de la Facultad de Ingeniería de la UNAM, (b) Golem del IIMAS-UNAM, y (c) Sabina del INAOE

### 3.1. Máquinas de estado finito

La forma más directa de programar la conducta de un robot de servicio es mediante el uso de máquinas de estado finito. En estas se tiene una representación del estado del sistema, y dadas ciertas condiciones de entradas y el estado presente se calcula el estado siguiente. La parte fundamental de las máquinas de estado finito es Autómata de Estados Finitos (AEF) que especifica y guía la conducta. Por ejemplo, la conducta de un robot omnidireccional con dos motores que le permiten ir hacia adelante, hacia atrás y hacer giros de 45 grados hacia la derecha e izquierda, que cuente con dos sensores (izquierdo y derecho) que le permiten detectar obstáculos y capaz de realizar un conjunto finito de acciones, se puede especificar de la siguiente manera. Si el robot no detecta ningún obstáculo avanza en la dirección en que está orientado; si detecta un obstáculo con el sensor izquierdo se mueve hacia atrás y gira hacia la derecha 45 grados; si detecta un obstáculo con el sensor derecho se mueve hacia atrás y gira hacia la izquierda 45 grados; si detecta un obstáculo con los dos sensores entonces se mueve hacia atrás, gira dos veces hacia la izquierda, avanza hacia adelante y finalmente gira hacia la derecha dos veces. Las entradas y salidas del autómata se pueden asociar con diferentes

dispositivos y algoritmos, como redes neuronales implementadas en hardware *Field Programmable Gate Array (FPGA)* (Savage et al. 2016) para interpretar una señal de entrada o campos de potenciales, para decidir una acción de navegación que se ejecuta cuando el control de estados se mueve de un estado al siguiente. Estas conductas pueden ser externas, como moverse hacia atrás o girar, o internas, como consultar a una base de datos de conocimiento. En el caso de que la interpretación y la conducta sean concretas el robot será *reactivo* y en la medida en que sus conductas internas sean más ricas tendrá una orientación más representacional y deliberativa. El robot *Justina*, el cual se ilustra en la Figura 3.1a, es una realización de este modelo conceptual en el nivel funcional. La Figura 3.2 y la Figura 3.3 ilustran el modelo conceptual máquinas de estado finito.

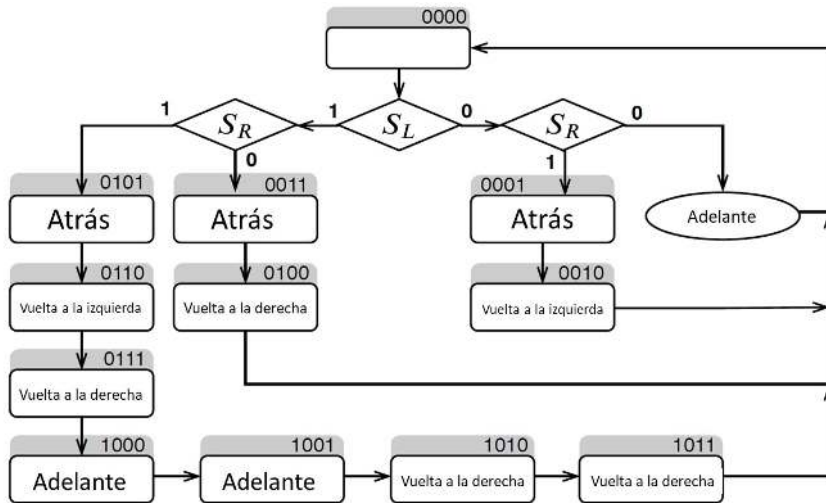


Figura 3.2: Máquina de estado finito que codifica comportamientos de un robot que evade obstáculos.

## 3.2. Especificación e interpretación de tareas

El modelo basado en autómatas de estado finito es limitado cuando la complejidad de la tarea y la comunicación aumenta, por lo que es útil incrementar el nivel de abstracción para especificar y ejecutar tareas complejas. Esto se puede lograr substituyendo la noción de *estado* por la noción más estructurada de *situa-*

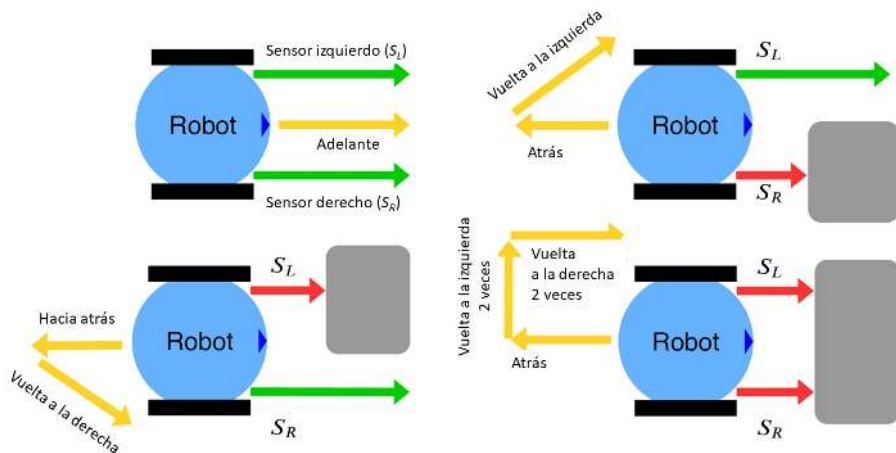


Figura 3.3: Algoritmo para una máquina de estado finito.

*ción*. La diferencia radica en que mientras un estado tiene un contenido nulo de información, excepto por las etiquetas asociadas a sus transiciones y sus siguientes estados, una situación es rica en información, como en los cuadros de historietas en las caricaturas, donde hay uno o más personajes en el entorno o contexto. Una situación se puede caracterizar computacionalmente mediante la especificación de las expectativas de quien juega el papel del *yo*, que en este caso es el robot, las cuales corresponden con sus creencias acerca de los eventos naturales que pueden ocurrir en el mundo y a las creencias e intenciones que adscribe a sus interlocutores. La situación contiene también las acciones intencionales que el robot debe realizar cuando una de las expectativas se cumple. En este modelo la acción del robot se concibe como un tránsito continuo entre situaciones, como cuando se lee un libro de historietas, que corresponde con las tareas que el robot realiza en el mundo.

Por ejemplo, la conducta de un mesero se puede caracterizar de manera simplificada por un conjunto de situaciones como: estar en la puerta del restaurante esperando al cliente, llevar al cliente a su mesa, darle la carta, tomarle la orden, servirle, vigilar que todo esté en orden, llevarle la cuenta, cobrarle, llevarle el cambio y despedirle. En cada situación el mesero tiene un conjunto, de expectativas generalmente pequeño, así como la capacidad de realizar de manera inmediata la

acción que se requiere cuando una de estas se cumple. La noción de situación es una abstracción con una extensión temporal y espacial relativamente amplia y de ahí su utilidad: el mesero puede estar toda una hora moviéndose nerviosamente en el vestíbulo del restaurante en la situación *esperando al cliente* y sólo pasará a la situación de *llevando cliente a su mesa* cuando llegue un cliente. De hecho el agente cambia de situación justamente cuando cambian sus expectativas. Por supuesto, hay situaciones complejas como *ver si todo está en orden*, pero estas se pueden partir, a su vez, en situaciones más simples, hasta llegar a las situaciones en que las expectativas correspondan con actos de interpretación concretos y las acciones se realicen directamente, como moverse o dar una orden. Por estas razones, tareas relativamente complejas, como la conducta de un mesero, de un asistente en un supermercado o de un recepcionista en un hotel, se pueden modelar como una gráfica prototípica de situaciones, que a su vez se extiende o *desdobla* como una gráfica concreta cada vez que el robot lleva a cabo una tarea.

En este modelo las expectativas y las acciones guían a los dispositivos *percepción* y *acción*; sin embargo, las acciones pueden ser también internas o *mentales* y pueden consistir en hacer inferencias conceptuales o invocar a mecanismos deliberativos, como los sistemas de diagnóstico, toma de decisiones y planeación, que se utilizan de manera oportunista según los requerimientos del ciclo de la comunicación.

La pregunta obligada en este punto es ¿Qué pasa si no se cumple ninguna expectativa o sucede un evento inesperado, que además es relevante? En este punto, el robot pasa de estar situado a no estarlo y no sabe qué hacer. No estar situado es también perder momentáneamente la capacidad de interpretar y actuar en relación con la tarea o perder el contexto. Por lo mismo, cuando no se cumple ninguna expectativa u ocurre un evento inesperado se activa un modelo de diálogo cuyo propósito es simplemente volver a situarse, o alternativamente, invocar un ciclo de razonamiento para saber qué pasó y actuar de manera coherente para volver a situarse y realizar sus objetivos. Un robot que adopta este modelo conceptual es un robot situado.

Para implementar este modelo conceptual se pueden emplear lenguajes de programación orientados a la definición de tareas. Por ejemplo, el Grupo Golem desarrolló el lenguaje de programación de tareas robóticas SitLog (Pineda, Salinas, Meza, Rascon & Fuentes, 2013). Este lenguaje ofrece dos tipos de datos abstractos

para especificar e interpretar tareas robóticas: el tipo *situación* y el tipo *modelo de diálogo*. El primero permite declarar directamente las expectativas y acciones de una situación y el segundo especificar una gráfica de situaciones que corresponde con la estructura prototípica de la tarea y, de manera paralela, a la estructura de la comunicación entre el robot y otros agentes en el entorno. El intérprete de SitLog consiste del intérprete de una red de transición recursiva (RTR),<sup>9</sup> que extiende a los AEFs con una estructura de pila o *stack*, y del intérprete de un lenguaje funcional para la especificación e interpretación de expectativas y acciones, donde ambos intérpretes funcionan de manera coordinada. El intérprete de SitLog es el componente central de la arquitectura *Interaction-Oriented Cognitive Architecture (IOCA)* (Pineda et al. 2011) utilizada en la serie de robots Golem y, en particular, en el robot Golem-III, el cual se ilustra en la Figura 3.1b. El intérprete de SitLog utiliza a su vez servicios de una base de conocimiento para llevar a cabo inferencias conceptuales<sup>10</sup> para explorar la base de datos de conocimiento, así como inferencias deliberativas de manera oportunista.<sup>11</sup> El modelo de especificación e interpretación de tareas se ilustra en Figura 3.4

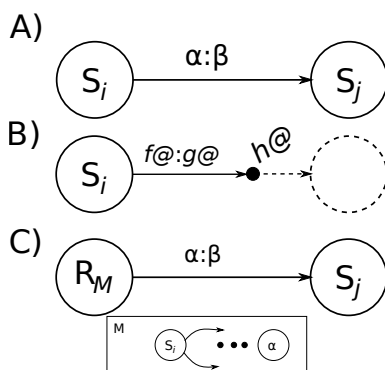


Figura 3.4: Ejemplo de codificación de tareas por medio de SitLog utilizando a) un arco simple, b) un arco funcional, y c) una situación recursiva.

<sup>9</sup>[https://en.wikipedia.org/wiki/Recursive\\_transition\\_network](https://en.wikipedia.org/wiki/Recursive_transition_network)

<sup>10</sup>Ver video en: <http://golem.iimas.unam.mx/lightkb/>

<sup>11</sup>Ver video en: [http://golem.iimas.unam.mx/opportunistic\\_inference](http://golem.iimas.unam.mx/opportunistic_inference)



### 3.3. Procesos de decisión de Markov

A medida que los robots de servicio interactúan en ambientes más variables y complejos se tienen que enfrentar a la incertidumbre tanto en la percepción del ambiente como en el resultado de sus acciones. Una forma de considerar y lidiar con dicha incertidumbre es a través de los procesos de decisión de Markov *Markov Decision Process* (MDP). Un MDP se puede ver como una extensión de las máquinas de estado finito que considera que las transiciones de un estado a otro, dada cierta acción, son probabilistas. Por ejemplo, si el robot realiza cierto movimiento como desplazarse 10 metros en línea recta, lo más probable es que no avance exactamente 10 metros dado que puede *patinar*, haber imperfecciones en el piso, imprecisión en los motores, etc. Los MDPs consideran la incertidumbre en el resultado de las acciones del robot; si también existe incertidumbre sobre el estado actual del robot (que tiene que ver con su percepción del ambiente) existe otro modelo llamado MDP parcialmente observable *Partially Observable Markov Decision Process* (POMDP). Mediante MDPs y POMDPs se pueden modelar los estados y acciones de un robot para realizar cierta tarea de forma que su comportamiento sea más robusto.

Los MDPs se pueden utilizar para representar y resolver problemas de la robótica en diferentes niveles, tanto problemas muy básicos, por ejemplo, desplazarse en su ambiente, como de alto nivel, por ejemplo, coordinar sus diferentes capacidades y resolver tareas complejas (Elinas, Sucar, Reyes & Hoey, 2004). En esta sección nos enfocamos a este último aspecto. Para realizar una tarea compleja un robot requiere utilizar diferentes capacidades en el nivel de dispositivos y algoritmos. Se requiere además que dichas capacidades se ejecuten de una forma coordinada para realizar una tarea, de preferencia en forma óptima (menor tiempo, mayor efectividad, etc.).

Mediante MDPs (o POMDPs) se pueden coordinar las diferentes capacidades básicas de un robot para realizar tareas complejas bajo incertidumbre. Para ello se define un *modelo* que especifica:

1. Un conjunto de estados  $S$  que define la situación actual del robot a alto nivel; por ejemplo su ubicación, la meta, lo que conoce del ambiente, los comandos de sus usuarios, etcétera.

2. Un conjunto de acciones  $A$ , que establece las posibles decisiones que puede realizar el robot, como localizarse, desplazarse de un lugar a otro, percibir el ambiente con sus cámaras u otros sensores, preguntar algo, etcétera.
3. Una función de transición  $P(S_i, A, S_j)$  que define las probabilidades de que el robot pase a un nuevo estado  $S_j$  dado que realizó cierta acción  $A$  y se encontraba en cierto estado  $S_i$ . Estas probabilidades se pueden establecer en forma aproximada por una persona (subjetivas) o se pueden aprender a través de interactuar con el ambiente mediante algoritmos de aprendizaje por refuerzo.
4. Una función de recompensa que establece que tan deseable es cada estado (o estado-acción) de acuerdo con los objetivos de la tarea. Normalmente, se establece una recompensa alta para el estado meta de la tarea; por ejemplo, el que el robot entregue a la persona cierto objeto, si la tarea es que el robot vaya y busque dicho objeto, lo tome y se lo lleve a una persona.

Una vez establecido el modelo de una tarea mediante un **MDP** existen algoritmos que permiten establecer una *política* o plan general para que el robot realice la tarea de forma óptima, es decir, que maximice la utilidad esperada, que normalmente es la suma de las recompensas que recibe de acuerdo con la función establecida previamente. La política establece para cada estado cuál es la mejor acción que debe hacer el robot de forma que se lleve a cabo la tarea de la mejor manera, incluso habiendo incertidumbre en los resultados de sus acciones.

Este esquema permite que se pueda hacer un desarrollo modular y eficiente de robots de servicio para realizar tareas. Si se cuenta con un conjunto de capacidades básicas, desarrollar nuevas tareas implica simplemente cambiar el modelo del **MDP** y resolverlo para tener el plan correspondiente. Una analogía sería el del director de una orquesta que con los mismos músicos e instrumentos puede interpretar diferentes obras cambiando la partitura.

Existen diversas extensiones al esquema básico descrito anteriormente. Una es considerar incertidumbre en el estado actual del robot de forma que se modela como un **POMDP**. Esto hace más compleja la solución ya que hay que considerar la probabilidad de cada estado (estado de creencia), lo que en principio implica un número infinito de estados; la solución exacta es sólo posible para problemas muy pequeños pero se han desarrollado soluciones aproximadas que permiten resolver

casos más complejos como los de robótica. Otra extensión es el considerar que el robot puede realizar varias acciones a la vez, como desplazarse, comunicarse y reconocer personas. Para este fin se han desarrollado MDPs concurrentes que toman en cuenta las restricciones para evitar conflictos entre los diversos tipos de acciones (Corona-Xelhuantzi, Morales & Sucar, 2009).

El robot Markovito,<sup>12</sup> el cual se ilustra en la Figura 3.1c, utiliza MDPs para representar y realizar diferentes tareas, entre las cuales están: i) llevar mensajes y objetos entre personas; ii) buscar un objeto en un ambiente doméstico; iii) desplazarse a diferentes cuartos en una casa; iv) recibir y reconocer a diferentes personas, entre otras. El enfoque de procesos de decisión de Markov se esquematiza en la Figura 3.5 a través de una red de decisión dinámica en 4 etapas, donde  $S$  es el estado,  $E$  las observaciones,  $D$  las decisiones o acciones y  $U$  las recompensas inmediatas.

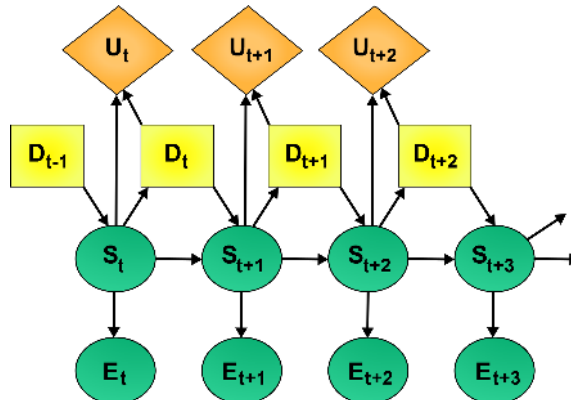


Figura 3.5: Proceso de decisión de Markov ilustrado a través de una red de decisión dinámica

### 3.4. Referencias

Corona-Xelhuantzi, E., Morales, E. F. & Sucar, E. (2009). Executing concurrent actions with multiple Markov decision processes. En *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (pp. 82-89). IEEE.

<sup>12</sup><http://ccc.inaoep.mx/~markovito/>

- Elinas, P., Sucar, E., Reyes, A. & Hoey, J. (2004). A decision theoretic approach for task coordination in social robots. En *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No. 04TH8759)* (pp. 679-684). IEEE.
- Pineda, L. A., Meza, I., Aviles, H., Gershenson, C., Rascon, C., Alvarado, M. & Salinas, L. (2011). IOCA: Interaction-Oriented Cognitive Architecture. *Research in Computer Science*, 54, 273-284.
- Pineda, L. A., Salinas, L., Meza, I. V., Rascon, C. & Fuentes, G. (2013). SitLog: A Programming Language for Service Robot Tasks. *International Journal of Advanced Robotic Systems*, 10(10), 358. doi:[10.5772/56906](https://doi.org/10.5772/56906)
- Savage, J., Cruz, J., Matamoros, M., Rosenblueth, D. A., Muñoz, S. & Negrete, M. (2016). Configurable Mobile Robot Behaviors Implemented on FPGA Based Architectures. En *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (pp. 317-322). IEEE.

## Capítulo 4

# Nivel de Dispositivos y Algoritmos

L. Enrique Sucar Succar, [INAOE](#)

Jesús Savage Carmona, Facultad de Ingeniería, [UNAM](#)

Luis A. Pineda Cortés, [IIMAS](#), [UNAM](#)

Caleb Rascón Estebané, [IIMAS](#), [UNAM](#)

Marco Morales Aguirre, [ITAM](#)

Este nivel se organiza en relación con las modalidades principales de la percepción, tales como la visión y el lenguaje, así como de la acción motora incluyendo la navegación y manipulación. Cada una de estas facultades se habilita mediante sensores y actuadores físicos en conjunto con algoritmos especializados que interpretan la señal proveniente del entorno para construir una representación del mundo. Es importante recalcar que el robot no mantiene representaciones del mundo directamente sino de las interpretaciones del mundo que es capaz de crear con su equipamiento sensorial y sus procesos perceptuales. En este capítulo se describen las facultades robóticas principales que han sido estudiadas por la comunidad mexicana en i) percepción y acción motora, ii) audición y lenguaje y iii) representación del conocimiento y razonamiento.

## 4.1. Percepción

La percepción robótica involucra dos tareas principales: i) el sensado de una señal proveniente del mundo mediante algún tipo de dispositivo, y ii) asignar una interpretación a dicha señal de manera coherente con el contexto o situación en que se hace la observación. Las señales pueden ser de cualquier modalidad de la percepción. Normalmente, se presentan al sistema de cómputo como arreglos o matrices de datos, donde cada celda contiene un número que codifica a una propiedad de la señal sensada; por ejemplo, en el caso de las cámaras, la escena fotografiada se registra en una matriz de píxeles, donde cada celda contiene un número que representa el color del punto correspondiente en el espacio; a partir de estas informaciones se inicia el proceso de interpretación (Gomez-Gil, 2018). Este tipo de dispositivos de alta tecnología se desarrollan en México sólo de manera muy limitada; por lo mismo, la investigación en robótica en nuestro país se hace comúnmente con componentes básicos disponibles en el mercado internacional. Entre los tipos de sensores más utilizados en robótica se encuentran los siguientes:

- Sonares: emiten una señal sonora y a partir del tiempo de ida y vuelta se puede estimar la distancia a los obstáculos (análogo a la manera en que los murciélagos detectan obstáculos).
- Láser: emiten rayos láser y también a partir del tiempo de ida y vuelta (o la fase) se estima la distancia a los objetos; son más precisos que los sonares.
- Cámaras: obtienen imágenes/videos del entorno que pueden utilizarse para múltiples propósitos.
- Cámaras de profundidad: como el Kinect, no sólo obtienen la información de color de cada píxel de la imagen sino también un estimado de su distancia.
- Micrófonos: permiten percibir los sonidos en el ambiente, en particular la voz humana para la interacción con las personas.
- Narices electrónicas: recientemente se han desarrollado sensores olfativos que permiten detectar ciertos compuestos como el alcohol.<sup>13</sup>

---

<sup>13</sup><http://tecreview.itesm.mx/esta-nariz-robotica-podria-salvarte-la-vida/>

Dado su bajo costo, tamaño y versatilidad, las cámaras y las cámaras de profundidad se han vuelto muy populares recientemente. En muchos casos se combinan diferentes sensores, ya sea del mismo o diferente tipo, mediante lo que se conoce como fusión sensorial [figura 4.1](#).

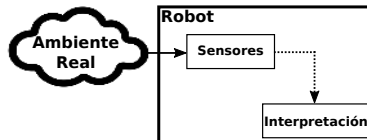


Figura 4.1: Descripción global de la percepción robótica.

## 4.2. Acción motora

Por su parte, la acción motora consiste en moverse en el entorno, caminar o tomar objetos con las manos, entre muchas otras formas. Estos procesos son inversos con respecto a la percepción ya que su entrada es una señal producida por un proceso computacional y la salida es la acción mecánica producida por un servo-motor.<sup>14</sup>

No todas las tareas robóticas requieren que la escena se interprete completamente y es común que la información proporcionada por el sensor sólo permita una interpretación muy limitada; sin embargo, esta puede ser suficiente para producir la conducta motora. Por ejemplo, para evadir objetos durante la navegación es suficiente detectar que hay un objeto enfrente para que el robot cambie de trayectoria sin tener que saber cuál es el objeto particular que se evade o a qué clase pertenece. Conductas de este tipo son reactivas en oposición a aquellas que involucran un proceso de interpretación profundo, que son deliberativas. Dentro de la robótica clásica, la mayoría de las conductas son reactivas e involucran al menos un sensor y un dispositivo de acción, por lo que la percepción y la acción motora se tienen que abordar juntas. Algunas de las tareas paradigmáticas de carácter reactivo son la navegación, el seguimiento o *tracking* y el movimiento supervisado por la visión o control visual. Por su parte, si el proceso incluye interpretaciones e inferencias deliberativas es posible diferenciar más claramente a las tareas de la percepción de

---

<sup>14</sup>Los servo-motores de calidad se producen también por corporativos internacionales y en México se usan como componentes básicos

las acciones motoras y estudiar ambos problemas por separado. Algunos procesos con un carácter más perceptual son la detección y el reconocimiento de personas, el reconocimiento de objetos y el análisis de escenas.

También hay tareas que involucran la construcción de representaciones pero que a su vez están estrechamente vinculadas con conductas motoras, por lo que la percepción y la conducta motora se abordan también en conjunto. Un ejemplo es la construcción de mapas al tiempo que se explora el espacio, aunada a la localización del robot en el mismo mapa; otro es el reconocimiento visual y la manipulación de objetos. Para enfrentar estos retos se han desarrollado una diversidad de algoritmos, algunos más genéricos, por ejemplo en el campo de la visión computacional, y otros más específicos para la robótica. A continuación se mencionan algunos desarrollos de la comunidad mexicana en i) mapeo, auto-localización y navegación, ii) seguimiento, iii) reconocimiento y manipulación de objetos, iv) detección y reconocimiento de personas y v) monitoreo y vigilancia [figura 4.2](#).

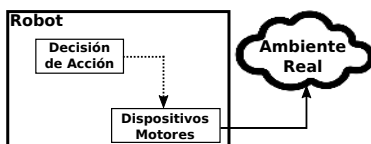


Figura 4.2: Descripción global de la acción motora.

### 4.3. Mapeo, localización y control

Uno de los problemas fundamentales para los robots de servicio es la movilidad, lo que incluye la construcción de mapas, la localización y la planificación y el control de movimientos:

1. Construcción de mapas: los robots móviles requieren una representación de su entorno o mapa para moverse en el entorno espacial; en general es conveniente que ellos mismos lo construyan por medio de sus capacidades sensoriales y algorítmicas. Existen diferentes tipos de mapas, incluyendo los métricos (rejillas de ocupación espacial, representaciones geométricas, en 2 y 3 dimensiones), los topológicos, los semánticos (Ver [Sección 3.1](#)) y los de



configuraciones. Estos últimos tienen tantas dimensiones como los grados de libertad del robot **DOF**.

2. **Localización**: este problema consiste en que el robot determine su ubicación en el mapa de manera dinámica. La localización tiene dos variantes principales: i) global, en la cual el robot se tiene que localizar de manera absoluta en relación con el mapa (por ejemplo cuando no sabe donde está al iniciar una tarea); ii) local o relativo a una posición de referencia durante su desplazamiento. Para localizarse, el robot puede utilizar información distintiva del ambiente, como marcas (naturales o artificiales), objetos específicos o puntos característicos (por ejemplo, esquinas). El modelo *Scale-Invariant Feature Transform* (**SIFT**) es uno de múltiples esquemas para identificar puntos característicos salientes y estables.<sup>15</sup> Es importante destacar que el robot tiene que actualizar su posición continuamente ya que si sólo utiliza el registro de su propio movimiento, lo que se conoce como odometría, el error de localización se incrementa continuamente hasta que eventualmente el robot se pierde.
3. **SLAM**: frecuentemente la construcción del mapa y el cálculo de la localización o ubicación se tienen que realizar de manera simultánea; este proceso coordinado se conoce como *mapeo y localización simultáneos*, *Simultaneous Localization and Mapping* (**SLAM**).
4. **Planificación y Control de Movimientos**: Los robots requieren construir un plan o definir una trayectoria para desplazarse de un lugar a otro; esto se conoce como planeación de trayectorias o de movimiento (*motion planning*). Para esta tarea se han desarrollado diversos algoritmos, incluyendo determinísticos y probabilísticos. Una vez establecido, el plan debe ejecutarse, lo que involucra el control de los actuadores del robot para poder seguir la trayectoria deseada, como las ruedas en robots móviles, las articulaciones en brazos robóticos y las piernas en robots humanoides. En ambientes variables deben detectarse posibles cambios en el mundo u obstáculos dinámicos y evitarlos o ajustar el plan.

---

<sup>15</sup>[https://en.wikipedia.org/wiki/Scale-invariant\\_feature\\_transform](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform)

Para resolver estas tarea, en el [INAOE](#) y el Centro de Investigación en Matemáticas ([CIMAT](#)) se han desarrollado algoritmos probabilísticos que utilizan sensores láser para obtener una representación del espacio libre (en 2D) basados en rejillas de ocupación espacial, donde cada celda contiene un número real que representa la probabilidad de que esté libre u ocupada; con base en el mapa resultante se han desarrollado técnicas de navegación robustas ante las limitaciones perceptuales del robot (Romero, Morales & Sucar, 2001b). Asimismo, dicho mapa se utiliza para que el robot se localice mediante la detección de marcas naturales en el ambiente, por ejemplo esquinas, puertas y paredes (Romero, Morales & Sucar, 2001a). También se ha trabajado en técnicas de navegación mediante el aprendizaje de reglas telereactivas a partir de ejemplos provistos por una persona (Vargas & Morales, 2009).

Se ha trabajado además en la navegación de robots con ruedas basada en un conjunto de imágenes objetivo las cuales se capturan previamente y constituyen una representación del ambiente que se denomina “memoria visual”, la cual le permite al robot navegar posteriormente de forma autónoma (H. M. Becerra, Sagues, Mezouar & Hayet, 2014). Se tienen ya resultados para robots humanoides. Otro reto es hacer mapas utilizando una sola cámara, lo que se conoce como [SLAM](#) monocular, ya que no se tiene información de profundidad; sin embargo, esta se puede estimar mediante conocimiento previo del mundo (Mota-Gutierrez, Hayet, Ruiz-Correa, Hasimoto-Beltran & Zubieta-Rico, 2015).

Un reto más, abordado en el Centro de Investigación y de Estudios Avanzados ([Cinvestav](#)), Instituto Politécnico Nacional ([IPN](#)) y el instituto Tecnológico Autónomo de México ([ITAM](#)), es la navegación para coches autónomos que requiere de la integración a alta velocidad de componentes de localización basada en entrada visual y de lidiar con planificación de tareas y movimientos. En el [ITAM](#) desarrollaron AutoNOMOs, un simulador de coche a escala,<sup>16</sup> con el que se ha abordado el problema de conducción autónoma sin sensores de localización, como el *Global Positioning System* ([GPS](#)) (Delgado, Granados, Martínez, Poblete & Morales, 2017; Meraz, Osegueda & Morales, 2017). También se ha usado en un proyecto conjunto entre el [ITAM](#) y el [INAOE](#) para combinar el uso de robots terrestres y aéreos (Rill-Garcia, Martinez-Carranza, Granados & Morales, 2020).

---

<sup>16</sup>[https://github.com/ITAM-Robotica/EK\\_AutoNOMOS\\_Sim](https://github.com/ITAM-Robotica/EK_AutoNOMOS_Sim)

Otro problema es la caminata estable de robots humanoides. Se han hecho desarrollos para simultáneamente realizar el control visual y controlar la caminata dinámica de un robot humanoide (Garcia et al. 2015). Para este efecto se utiliza un generador de patrón de movimiento basado en control predictivo *Model predictive Control* (MPC) que adapta los pasos de la caminata y las trayectorias del centro de masa del robot para seguir perfiles de velocidad provistos por el usuario. La contribución de este trabajo es de reformular el problema de MPC al considerar información de retroalimentación visual, en lugar de usar la velocidad de referencia. Un problema relacionado es la regulación de postura por medio de visión para robots humanoides (Delfin, Becerra & Arechavaleta, 2016). Para este efecto se utiliza un esquema de control capaz de lograr la convergencia de la tarea visual en un tiempo predefinido. El esquema es un control jerárquico basado en tareas que puede manejar al mismo tiempo la tarea visual y una tarea de evasión de obstáculos, los cuales se detectan con una cámara de profundidad. Genera velocidades continuas para el robot y considera las capacidades de caminata omnidireccional de los robots humanoides.

#### 4.3.1. Seguimiento

El seguimiento de personas en ambientes dinámicos donde pueden transitar libremente vehículos y personas, como en las pruebas de navegación de la competencia RoboCup, es una tarea sumamente compleja. Este problema se ha abordado por los robots Markovito, Justina y Golem con diferentes estrategias y algoritmos.

El método de seguimiento de personas del equipo Markovito del INAOE combina información del esqueleto obtenido con el Kinect con información de color de la cámara del robot. Inicialmente se detecta a la persona mediante el algoritmo de detección de esqueletos del Kinect y luego se aprende un modelo de color de la ropa de dicha persona. Para el seguimiento se integra la información de las observaciones (basadas en color) con la estimación de movimiento mediante un filtro de Kalman.

La estrategia que se desarrolla actualmente por el Grupo Golem se basa en el uso combinado de un láser que localiza el torso de la persona a seguir y el esqueleto que se obtiene del Kinect. El método utiliza adicionalmente el parche de colores del objeto a seguir, como la camisa o el pantalón del sujeto, para distinguirlo de

otras personas u objetos, fijos o móviles, que se encuentren en la trayectoria de seguimiento.

Un problema relacionado que se considera clásico en robótica móvil desde el punto de vista de los sistemas de control es el de persecución–evasión, donde un robot debe seguir y alcanzar a un blanco móvil. Este problema se ha analizado en el [CIMAT](#) desde el punto de vista teórico para la captura de un evasor omnidireccional usando un robot de manejo diferencial en un ambiente sin obstáculos (Ruiz, Murrieta-Cid & Marroquin, 2013). El método propuesto utiliza técnicas de control óptimo para obtener representaciones de las primitivas de movimiento y estrategias (en equilibrio de Nash) de tiempo mínimo para seguidor y evasor. También aborda el problema de decisión del juego y las condiciones que definen al ganador. Esta metodología se ha aplicado además al problema de persecución–evasión en un ambiente con obstáculos (Israel Becerra, Murrieta-Cid, Monroy, Hutchinson & Laumond, 2016) con atención especial al problema combinatorio que surge para mantener visibilidad de un evasor que visita varias ubicaciones.

### 4.3.2. Reconocimiento y manipulación de objetos

Los robots de servicio deben ser capaces de tomar y llevar objetos de un sitio a otro, lo que se conoce como el problema de manipulación. Para ello los robots móviles y humanoides incorporan brazos robóticos con pinzas o manos para sujetar diferentes tipos de objetos. La manipulación presupone a su vez que el robot debe ver los objetos, para lo cual es necesario identificarlos y determinar su posición y orientación espacial. Por esta razón, los problemas de reconocimiento y manipulación se relacionan de manera muy estrecha. Desde el punto de vista computacional, el problema de manipulación involucra varios subproblemas:

- Reconocimiento y ubicación del objeto: el primer paso para manipular un objeto es reconocerlo y ubicar su posición en el ambiente (coordenadas 3D).
- Modelado: en el caso de objetos complejos se requiere conocer su forma; es decir, obtener un modelo tridimensional que permita identificar posibles puntos de agarre. Para ello se obtienen diferentes vistas del objeto con algún sensor de profundidad, como el Kinect, que se integran en una representación 3D. El determinar las mejores posiciones del sensor para obtener dichas vistas

es un problema de planificación que se conoce como *siguiente mejor vista* (*next best view*).

- Determinar puntos de sujeción (*grasping*): con base en el modelo 3D del objeto se deben establecer puntos de sujeción estables (normalmente dos o tres, dependiendo del tipo de pinza o mano); es decir, que permitan sujetar y levantar al objeto en forma segura.
- Planeación/ejecución de movimientos: una vez que se establece la posición y punto de agarre del objeto es necesario llevar al efector final del brazo a las coordenadas deseadas, para lo cual se establece un plan de los movimientos del brazo/base móvil que luego se ejecuta, normalmente con cierta retroalimentación sensorial que asegure tomar al objeto.

Enseguida resumimos algunas aportaciones de grupos en nuestro país.

El Grupo Golem del IIMAS, UNAM, aborda la tarea de reconocimiento de objetos mediante el algoritmo *Multiple Object Pose Estimation and Detection* (MOPED) (Collet, Martinez & Srinivasa, 2011) el cual requiere crear un modelo de antemano para cada objeto conocido. Los modelos se forman a partir de un conjunto de fotografías tomadas alrededor del objeto desde diferentes distancias y perspectivas para configurar una envolvente. Las fotografías se procesan mediante el algoritmo SIFT (Ver Sección 4.5 Con estos descriptores se forma una nube de puntos alrededor del objeto, la cual se calibra con respecto a un punto externo de observación. Esta calibración permite determinar cuál será el tamaño del objeto con relación con un nuevo punto de observación. Para efectos de reconocimiento, la vista del objeto desde el punto de observación se procesa también mediante el algoritmo SIFT y sus descriptores se alinean parcialmente con respecto a alguna vista arbitraria del modelo. Si este proceso es exitoso el objeto se reconoce en conjunto con su posición y orientación o *pose* en el espacio 3D en relación con el punto de observación, que corresponde con la cámara del robot.

Para efectos de la manipulación, el robot alinea su brazo a la misma altura que el objeto y, dado que conoce el tamaño de su brazo y la distancia al objeto, le es posible dirigir su brazo hacia la posición del objeto por un método de triangulación directo.

El algoritmo MOPED es muy preciso (tiene muy pocos falsos positivos) pero sólo funciona con objetos ricos en textura, además de que no puede reconocer

objetos para los que no se cuente con modelos, por lo que su cobertura es pobre y puede tener muchos falsos negativos (es decir objetos que no ve). Para superar esta limitación el grupo Golem utiliza una segunda estrategia basada en la nube de puntos, que corresponden a objetos en la escena, que se obtiene a través del sensor de distancia del Kinect. Las nubes se segmentan en la imagen y se asocian a una máscara de color, la cual se utiliza como un identificador único para objetos de baja textura. Con esta nueva funcionalidad, el robot Golem-III pondera los objetos que reconoce y otorga un mayor peso a los objetos reconocidos con [MOPED](#), un peso menor a los objetos modelados por nube de puntos y máscara de color, y un peso aún más bajo a los objetos que reconoce por la nube de puntos, pero cuya máscara de color no corresponde con ninguno de los modelos conocidos. Estos últimos objetos se catalogan como *no conocidos*. Con esta estrategia en cascada se espera incrementar significativamente la cobertura visual y dar al robot Golem-III la capacidad de manipular una gama amplia de objetos tanto conocidos como desconocidos.

Sin embargo, es difícil que un robot de servicio cuente con modelos previos de todos los posibles objetos en el mundo, por lo que en el [INAOE](#) se ha desarrollado un método de reconocimiento de objetos que aprende nuevos modelos a partir de ejemplos de imágenes que obtiene de Internet (Navarrete, Morales & Sucar, 2012). Para tener más ejemplos se generan nuevas imágenes mediante transformaciones de las imágenes originales y luego se aprende un clasificador del nuevo objeto a partir de los ejemplos, que se puede usar para reconocerlos en el ambiente.

En el [INAOE](#) se desarrolló también un algoritmo para resolver el problema de planificación de vistas para modelado tridimensional de objetos; este toma en cuenta la incertidumbre en la posición final del sensor, el cual está montado en el eslabón final de un brazo sobre un robot móvil (Vasquez-Gomez & Sucar, 2016). El desplazamiento de la base móvil introduce errores en la posición deseada del sensor, lo que puede ocasionar que no se obtenga una vista adecuada. Para compensar esto y tener posiciones robustas ante el error de posicionamiento, el planificador ejecuta varias veces el algoritmo de planeación basado en *Rapidly Exploring Random Tree* (RRT) simulando errores y selecciona aquel plan que da un valor esperado óptimo. Para ello se define una función de utilidad que establece un compromiso entre varios aspectos deseables: i) observar superficies no vistas del objeto, ii) garantizar un traslape mínimo con las vistas anteriores (para poder

juntar las vistas en un modelo 3D), y iii) minimizar la distancia que el robot debe moverse.

En el campo del control visual de robots móviles se han propuesto esquemas genéricos de control robusto que usan restricciones geométricas y relacionan múltiples vistas de una escena. Por ejemplo, el problema de regulación de postura se ha abordado mediante un esquema de control visual basado en imágenes (*image-based visual servoing*) que usa restricciones geométricas para conducir a un robot de manejo diferencial a una posición y orientación deseadas (H. M. Becerra, Hayet & Sagués, 2014). Un problema relevante en este contexto es el seguimiento de objetos, por ejemplo, para tomar un objeto con la mano. En el CIMAT se ha investigado también el problema de confirmar la identidad de un objeto candidato; para este efecto se propuso un método que mezcla la localización del robot relativa al objeto candidato y la confirmación de que dicho objeto es el blanco buscado (I. Becerra, Valentín-Coronado, Murrieta-Cid & Latombe, 2016). El proceso de confirmación con esta meta dual se modela como un POMDP que se resuelve con Programación Dinámica.

Otro problema relevante es la búsqueda de objetos; por ejemplo si el usuario hace una petición (*robot, tráeme una manzana*) en una casa es necesario combinar aspectos de percepción, planeación y navegación. Se desea que el robot encuentre el objeto en el menor tiempo posible, para lo cual debe estimar los lugares más probables (por ejemplo, una manzana puede encontrarse en la cocina o el comedor) y establecer una estrategia de búsqueda que minimice el tiempo esperado. En el INAOE se desarrolló una estrategia que inicialmente estima la probabilidad de encontrar cierto objeto en los diferentes tipos de habitaciones; esta se basa en estimar la correlación entre objeto-habitación mediante el uso de diversas bases de datos en Internet (Izquierdo-Córdova, Morales & Sucar, 2016). Combinando la probabilidad de las habitaciones, la distancia del robot a las habitaciones y el área de las habitaciones, se establece una heurística que permite planear una ruta para explorar las habitaciones en el menor tiempo posible, la cual da resultados cercanos al óptimo (Izquierdo-Cordova, Morales, Sucar & Murrieta-Cid, 2016).<sup>17</sup> Finalmente, el robot busca el objeto dentro de cada habitación enfocándose en superficies planas donde se espera encontrarlo.

---

<sup>17</sup>Garantizar el tiempo óptimo es un problema difícil de resolver eficientemente, lo que se conoce en computación como problemas *Non-deterministic Polynomial Time* (NP) (no existe algoritmo conocido que pueda resolverlo en un tiempo polinomial respecto al tamaño del problema).

### 4.3.3. Reconocimiento de personas

El reconocimiento de personas forma parte de las tareas que un robot de servicio debe realizar. Este problema se puede descomponer en tres subproblemas: detección, localización e identificación, cuyo objetivo es determinar cuántas personas hay, dónde están y quiénes son respectivamente (Cielniak & Duckett, 2004).

La detección está relacionada con la capacidad de clasificar cuáles objetos en el ambiente son personas y cuáles no. Una estrategia común es primero detectar caras en las imágenes (Reyes-García, Manzanares, Escalante-Balderas & Torres-García, 2018)<sup>18</sup> que permite determinar si un pixel forma parte de una cara o no. La localización, por su parte, consiste en determinar la ubicación exacta de la persona detectada. Por ejemplo, si la detección se realiza con Kinect es posible obtener las coordenadas 3D de los puntos más relevantes de su cuerpo, incluyendo las manos, los codos, los hombros y la cabeza. Finalmente, la identificación consiste en determinar si la persona detectada está incluida en la base de datos del robot. Este último proceso se realiza principalmente mediante el uso de rasgos biométricos (A. K. Jain, Dass & Nandakumar, 2004); por ejemplo, mediante la identificación de puntos característicos en la zonas de los ojos-nariz-boca. El reconocimiento se hace más robusto al combinar varias imágenes mediante un esquema bayesiano (Cruz, Sucar & Morales, 2008), que es la estrategia que usa el robot Markovito para reconocer personas.

En particular, el robot Golem-III realiza la detección de personas mediante el algoritmo de Viola-Jones para detectar rostros. También puede detectar la región completa de la cabeza y los hombros utilizando un algoritmo desarrollado por el grupo de investigación Golem, el cual se basa en la técnica de Histogramas de Gradientes Orientados (Dalal & Triggs, 2005). El robot utiliza un Kinect 2 para poder detectar cuerpos y sus respectivos esqueletos, los cuales permiten clasificar la posición en la que se encuentra una persona –de pie, sentada o recostada en el piso– o determinar si está realizando un gesto –ya sea apuntar o alzar la mano para llamar la atención. Para la identificación de personas, el robot utiliza la técnica llamada Eigenrostros (Turk & Pentland, 1991), la cual es un enfoque biométrico que utiliza algoritmos de aprendizaje.

---

<sup>18</sup> <http://amexcomp.mx/index.php?r=site/page&view=libros>



En años recientes, un tipo especial de rasgos llamados biométricos suaves (como el color de la ropa) han atraído la atención de distintas investigaciones debido a su utilidad en escenarios no controlados (A. Jain, Ross & Prabhakar, 2004). En particular, el robot Golem-III puede identificar a una persona a partir de su ropa con la información que provee el detector de esqueletos de Kinect 2 y sin necesidad de ver el rostro. Para ello, durante la fase de aprendizaje, el robot guarda múltiples vistas de la misma persona en distintas orientaciones y extrae parches de color que se etiquetan semánticamente de acuerdo con la parte del cuerpo en la que se encuentran.

#### 4.3.4. Monitoreo y vigilancia

Los robots no sólo deben ser capaces de analizar imágenes estáticas sino también secuencias de imágenes o videos. En el CIMAT se desarrolló una metodología para aprender y utilizar modelos de movimiento probabilísticos de múltiples objetivos, incorporada a un esquema de rastreo visual por filtro de partículas en secuencias de video (Madrigal & B., 2015). Dada una secuencias de video para entrenamiento se extraen primero las posiciones (u *objetivos*) donde los peatones entran/salen de la escena observada, o simplemente donde cambian de dirección con frecuencia. Posteriormente, se aprende un modelo de movimiento a partir de estadísticas para cada uno de estos objetivos. Finalmente, se utilizan los modelos de movimiento en un esquema de tipo *Interacting Multiple Model (IMM)* para el seguimiento y la estimación del objetivo de cada persona rastreada.

### 4.4. Audición robótica

Una gran parte de la información disponible en el ambiente está en el dominio del audio. Es de gran interés que un robot de servicio pueda extraer información de dicho dominio. Por un lado, la voz es uno de los vehículos principales de interacción entre humanos, por lo que es natural que el humano quiera interactuar de esta manera con el robot (Roe & Wilpon, 1994). El avance actual del reconocimiento de voz es evidente en productos como Alexa de Amazon, Google Voice, y Siri de Apple.

Por otro lado, hay una cantidad considerable de información relevante para dicha interacción que va más allá del reconocimiento de voz. Esta información es lo que se define como la escena auditiva, una representación básica de la cual se muestra en la Figura 4.3.

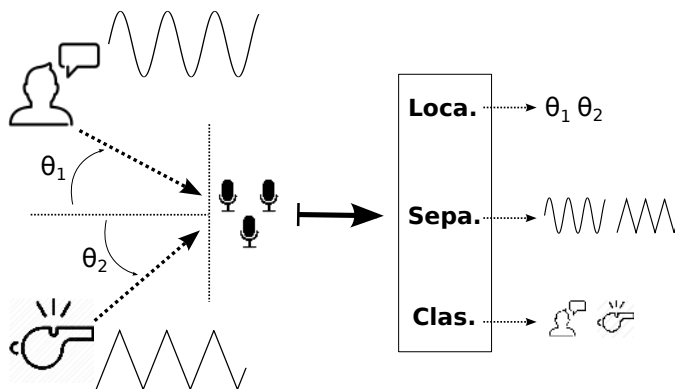


Figura 4.3: Escena auditiva.

En una forma general, el análisis de una escena auditiva se puede dividir en i) localizar, ii) separar y iii) clasificar las fuentes sonoras en el ambiente. Esto a partir de la mezcla capturada en un arreglo de micrófonos instalados sobre el robot.

El problema de la localización de fuentes sonoras involucra posicionar en la escena auditiva a todas estas. Hay una cantidad considerable de metodologías para resolver este problema sobre una base robótica (Rascon & Meza, 2017), y la gran mayoría se basan en métodos tradicionales de análisis estadístico y filtrado. Desafortunadamente, se sufren de diversas limitantes, pero la más relevante aquí es que requieren de un alto número de micrófonos para localizar múltiples fuentes sonoras en ambientes reales. Actualmente en la UNAM, se ha abordado el problema con una configuración de hardware conformado por un arreglo de tres micrófonos que puede localizar más fuentes que el número de micrófonos utilizados, descrito en la Figura 4.4.

Dicho desarrollo fue integrado en el robot de servicio Golem (Meza, Rascon, Fuentes & Pineda, 2016) y se utiliza en varias aplicaciones como recuperarse al perder visualmente a la persona que se sigue, tomar asistencia en una clase, jugar

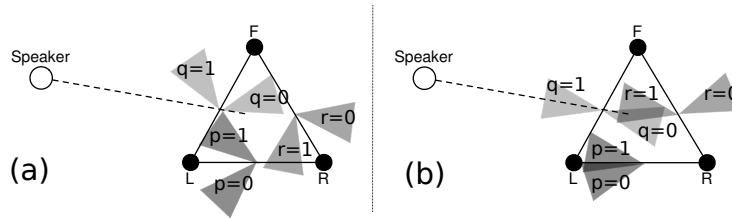


Figura 4.4: Localización con sólo 3 micrófonos (Rascon, Fuentes & Meza, 2015), donde (a) es un caso incoherente, y (b) es un caso coherente.

el juego Marco Polo (versión americana del juego *Gallinita Ciega*) y ser mesero en un restaurante (Rascon, Meza, Fuentes, Salinas & Pineda, 2015).

Por lo que va de separación de fuentes sonoras, uno de los métodos más utilizados actualmente se basa en filtrado espacial (o *beamforming*) donde se pretende descartar toda la información auditiva que no tenga alineada su información de fase. Al llevar a cabo esto después de un proceso de desfasamiento artificial que depende de la localización de la fuente de interés, los datos auditivos de esta son preservados mientras que el resto son reducidos substancialmente. Desafortunadamente, dicha reducción es proporcional al número de micrófonos utilizados y es muy sensible a errores de localización. Actualmente, dicho reto se aborda en la UNAM por medio de técnicas de *Aprendizaje Profundo* que refinan la salida del filtro espacial.

El último componente de la escena auditiva descrita aquí es la clasificación de la fuentes sonoras localizadas y separadas. Dicha clasificación puede ser desde identificar si la fuente es una persona o no, hasta identificar quién es el/la que está hablando. La clasificación también puede ser aún más amplia, como identificar entre tipos de ruido, estimar el estado emocional de la persona, etc. Desafortunadamente, los métodos tradicionales de clasificación o identificación de fuentes sonoras (así como de *Aprendizaje Automático* en general) requieren conocer previamente las clases a seleccionar, lo cual hace impráctico considerar nuevas clases en el ambiente, por ejemplo, cuando hay nuevos hablantes que interactúan con el robot con los que el sistema de clasificación no fue entrenado. En la UNAM se abordó este problema por medio de crear un sistema de identificación el cual no requiere re-entrenarse cuando hay un nuevo usuario, descrito en la Figura 4.5.

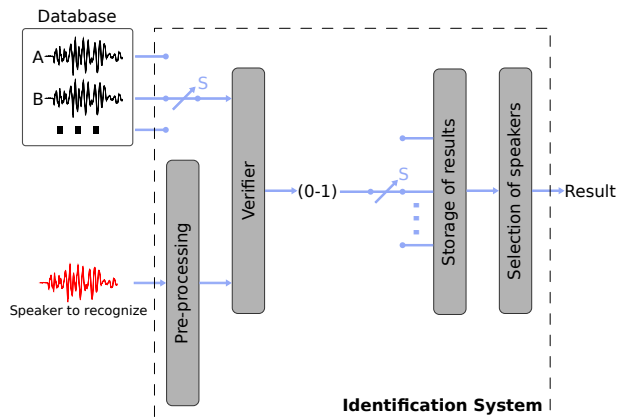


Figura 4.5: Identificación de nuevos usuarios (I. Velez, 2018)

Para la evaluación de algoritmos de análisis de escenas auditivas, centrada en la Audición Robótica, en la UNAM se capturó el corpus *Acoustic Interactions for Robot Audition (AIRA)* (Rascon et al. 2018), basado en el corpus DIMEx100 (L. Pineda et al. 2010). Ha sido utilizado exitosamente para evaluar diversos algoritmos y tiene un gran potencial para ser utilizado como un corpus de entrenamiento para metodologías basadas en Aprendizaje Automático que localicen, separen y clasifiquen fuentes sonoras en un ambiente real.

## 4.5. Representación del conocimiento y razonamiento

En robótica de servicio es muy importante contar con un sistema de representación de conocimiento y razonamiento. Entre los diversos aspectos que debe conocer el robot destaca la necesidad de contar con un mapa del entorno físico. Los principales tipos de mapas utilizados en robótica son:

1. *Mapas métricos*. Consisten en una representación geométrica del espacio en dos o tres dimensiones en la que se distinguen al menos las zonas libres (por donde el robot puede desplazarse) de las zonas ocupadas (obstáculos). Esto le permite al robot conocer su ubicación en el mundo y navegar. Existen

diversas formas de representación incluyendo: i) mapas de celdas (voxels en 3D) en el cual el espacio se divide en una rejilla de 2 ó 3 dimensiones a cuyas celdas se les asigna un valor de ocupado o libre (cuantificado por una probabilidad); ii) mapas que se representan con elementos geométricos básicos, como líneas, superficies, cilindros, etc., que al integrarse proveen un modelo aproximado del mundo; iii) mapas con base en puntos característicos que se extraen de las imágenes y cuyo objetivo principal es permitir al robot mantener su ubicación, sin incluir un modelado preciso de los objetos en el mundo.

2. *Mapas topológicos*. El ambiente se representa en forma más abstracta con base en un grafo de conectividad, donde los nodos representan áreas o espacios (como cuartos en ambientes interiores) y los arcos denotan relaciones de adyacencia o conectividad (como pasillos y puertas). Los mapas topológicos le permiten al robot planear sus trayectorias sin necesidad de información precisa del mapa.
3. *Mapas de configuraciones*. Representan los movimientos posibles que puede realizar un robot. Una configuración es una abstracción del robot que representa su pose en forma única y tiene un parámetro por cada uno de sus grados de libertad **DOF**. Por ejemplo, un robot móvil sin manipulador tiene usualmente tres, dos para desplazamiento y uno para orientación, y una configuración es una combinación única de estos tres parámetros. Estos mapas se representan como grafos cuyos nodos son configuraciones y sus aristas secuencias de configuraciones para transitar entre los nodos. La construcción de estos mapas es intratable computacionalmente por lo que se obtienen mediante muestreo ya sea en forma global, como los *Probabilistic Roadmaps* (**PRMs**), o incremental, como los **RRTs**.
4. *Mapas semánticos*. Expresan información de nivel más alto, como el tipo de ambiente (interiores, exteriores) o la clase de habitación (sala, comedor, recámara) que permiten dar comandos en forma más natural a los robots, por ejemplo: *robot, tráeme una manzana de la cocina*.

La construcción del mapa por el propio robot permite que la representación sea coherente con sus capacidades sensoriales; esto ha motivado un desarrollo significa-

tivo de las técnicas de **SLAM**, las cuales se estudian en México desde los noventa. Por ejemplo, además de contar con conocimiento del mundo físico en el que habita, el robot puede tener otros tipos de conocimiento (Madrigal & B, 2000), tal como:

- Conocimiento sobre los objetos en su ambiente y sus propiedades, incluyendo objetos pequeños (manipulables), objetos fijos, entre otros.
- Conocimiento sobre diversos agentes, como personas u otros robots (por ejemplo, para reconocerlos).
- Conocimiento de lenguaje que le permita comunicarse con personas y posiblemente con otros robots.

Para representar este conocimiento se utilizan los diversos formalismos que se han desarrollado en inteligencia artificial, incluyendo lógica de predicados, reglas de producción, representaciones estructuradas, modelos gráficos probabilistas, entre otros.

En particular, el robot Golem-III cuenta con una base de conocimiento no-monotónica en la que se pueden representar clases de individuos en una estructura jerárquica, con sus propiedades y relaciones, así como individuos concretos de cada clase, también con sus propiedades y relaciones particulares (L. A. Pineda, Rodríguez, Fuentes, Rascón & Meza, 2017; Cantú & Aldeco, 2018). El sistema permite la expresión de *valores por omisión o defaults*, así como excepciones, por lo que permite algunas formas de razonamiento no-monotónico (L. A. Pineda et al. 2017; Cantú & Aldeco, 2018). La base de datos de conocimiento se embebe en la conducta perceptual y motora del robot, y permite una gran expresividad y flexibilidad. El robot Golem-III cuenta también con un sistema de inferencia que le permite hacer diagnósticos, tomar decisiones (en relación con un conjunto de valores o preferencias preestablecidas) así como hacer planes para satisfacer dichos objetivos. Por ejemplo, si en la tarea de asistente del supermercado Golem-III no encuentra un objeto en el estante en el que debería estar, es capaz de diagnosticar que el proveedor lo puso en un estante incorrecto, de tomar la decisión de ordenar los estantes y/o atender los requerimientos del cliente, y de construir un plan y llevarlo a cabo para lograr este objetivo.<sup>19</sup> Este ciclo de razonamiento se puede utilizar de manera directa como parte de la estructura de una tarea, o

---

<sup>19</sup><http://golem.iimas.unam.mx/opportunistic.inference>

dinámicamente, cuando las expectativas del robot no se cumplen, por lo que se sale de contexto; en este caso el ciclo inferencial tiene por objetivo identificar qué pasó y cómo contextualizarse nuevamente para completar la tarea.

## 4.6. Referencias

- Becerra, H. M., Hayet, J. B. & Sagués, C. (2014). A single visual-servo controller of mobile robots with super-twisting control. *Robotics and Autonomous Systems*, 62(11), 1623-1635.
- Becerra, H. M., Sagues, C., Mezouar, Y. & Hayet, J. B. (2014). Visual navigation of wheeled mobile robots using direct feedback of a geometric constraint. *Autonomous Robots*, 37(2), 137-156.
- Becerra, I. [I.], Valentín-Coronado, L. M., Murrieta-Cid, R. & Latombe, J. C. (2016). Reliable confirmation of an object identity by a mobile robot: A mixed appearance/localization-driven motion approach. *International Journal of Robotics Research*, 35(10), 1207-1233.
- Becerra, I. [Israel], Murrieta-Cid, R., Monroy, R., Hutchinson, S. & Laumond, J. P. (2016). Maintaining strong mutual visibility of an evader moving over the reduced visibility graph. *Autonomous Robots*, 40(2), 395-423.
- Cantú, F. & Aldeco, R. (Eds.). (2018). *Conocimiento y Razonamiento Computacional*. Amexcomp. Recuperado desde <http://amexcomp.mx/files/ConocimientoRazonamiento.pdf>
- Cielniak, G. & Duckett, T. (2004). People Recognition by Mobile Robots. *Journal of Intelligent and Fuzzy Systems*, 15, 21-27.
- Collet, A., Martinez, M. & Srinivasa, S. (2011). The MOPED framework: Object Recognition and Pose Estimation for Manipulation. *The International Journal of Robotics Research*, 30, 1284-1306.
- Cruz, C., Sucar, L. E. & Morales, E. (2008). Real-Time Face Recognition for Human-Robot Interaction. En *IEEE International Conference on Automatic Face and Gesture Recognition*.
- Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*, 1, 886-893.

- Delfin, J., Becerra, H. M. & Arechavaleta, G. (2016). Visual servo walking control for humanoids with finite-time convergence and smooth robot velocities. *International Journal of Control*, 89(7), 1342-1358.
- Delgado, J. A., Granados, E., Martínez, E. D., Poblete, K. L. & Morales, M. (2017). Integrated Task and Motion Planning for Instances of Autonomous Driving. En *Workshop on Integrated Task and Motion Planning at Robotics:Science and Systems XIII (RSS)*. Robotics Science y Systems Foundation.
- García, M., O., S., Hayet, J. B., Dune, C., Esteves, C. & Laumond, J. P. (2015). Vision-guided motion primitives for humanoid reactive walking: Decoupled versus coupled approaches. *International Journal of Robotics Research*.
- Gomez-Gil, P. (Ed.). (2018). *El reconocimiento de patrones y su aplicación a las señales digitales*. Amexcomp. Recuperado desde <http://amexcomp.mx/files/ReconocimientoPatronesAppSenalesDigitales.pdf>
- I. Velez, G. F., C. Rascon. (2018). One-Shot Speaker Identification for a Service Robot using a CNN-based Generic Verifier. Recuperado desde <https://arxiv.org/abs/1809.04115>
- Izquierdo-Córdova, R., Morales, E. & Sucar, L. (2016). Object Location Estimation in Domestic Environments through Internet Queries. En *RoboCup Symposium*.
- Izquierdo-Cordova, R., Morales, E., Sucar, L. & Murrieta-Cid, R. (2016). Searching Objects in Known Environments: Empowering Simple Heuristic Strategies. En *IJCAI Workshop on Service Robots*.
- Jain, A. K., Dass, S. C. & Nandakumar, K. (2004). Soft Biometric Traits for Personal Recognition Systems. En *International Conference on Biometric Authentication* (pp. 731-738).
- Jain, A., Ross, A. & Prabhakar, S. (2004). An Introduction to Biometric Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 4-20.
- Madrigal, F. & B, H. J. (2000). Learning Probabilistic Grid-Based Maps for Indoor Mobile Robots Using Ultrasonic and Laser Range Sensors. En *Mexican International Conference on Artificial Intelligence MICAI* (Vol. 1793, pp. 158-169). Springer-Verlag.



- Madrigal, F. & B., H. J. (2015). Goal-oriented visual tracking of pedestrians with motion priors in semi-crowded scenes. En *IEEE International Conference on Robotics and Automation ICRA* (pp. 720-725).
- Meraz, J. A. D., Osegueda, E. G. & Morales, M. (2017). An Application of POMDPs for Autonomous Driving. En *Workshop on Integrated Task and Motion Planning at Robotics:Science and Systems XIII (RSS)*. Robotics Science y Systems Foundation.
- Meza, I., Rascon, C., Fuentes, G. & Pineda, L. (2016). On Indexicality, Direction of Arrival of Sound Sources, and Human-Robot Interaction. *Journal of Robotics*.
- Mota-Gutierrez, S. A., Hayet, J. B., Ruiz-Correa, S., Hasimoto-Beltran, R. & Zubieta-Rico, C. E. (2015). Learning depth from appearance for fast one-shot 3-D map initialization in VSLAM systems. En *International Conference on Robotics and Automation ICRA* (pp. 2291-2296). IEEE.
- Navarrete, D., Morales, E. & Sucar, L. E. (2012). Unsupervised learning of visual object recognition models. En *13th Ibero-American Conference on Artificial Intelligence, IBERAMIA* (pp. 511-520). IberoAmerican Society of Artificial Intelligence.
- Pineda, L. A., Rodríguez, A., Fuentes, G., Rascón, C. & Meza, I. (2017). A Light non-monotonic knowledge-base for service robots. *Intel Serv Robotics*, 10(3), 159-171. Recuperado desde <https://link.springer.com/article/10.1007/s11370-017-0216-y>
- Pineda, L., Castellanos, H., Cuétara, J., L. Galescu, J. J., Llisterri, J., Pérez-Pavón, P. & Villaseñor, L. (2010). The Corpus DIMEx100: Transcription and Evaluation. *Language Resources and Evaluation*, 44(4).
- Rascon, C., Fuentes, G. & Meza, I. (2015). Lightweight multi-DOA tracking of mobile speech sources. *EURASIP Journal on Audio, Speech, and Music Processing*, 11.
- Rascon, C. & Meza, I. (2017). Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96.
- Rascon, C., Meza, I., Fuentes, G., Salinas, L. & Pineda, L. (2015). Integration of the Multi-DOA Estimation Functionality to Human-Robot Interaction. *International Journal of Advanced Robotic Systems*, 12(8).

- Rascon, C., Meza, I., Millán, A., Velez, I., Fuentes, G., Mendoza, D. & Espitia, O. (2018). Acoustic interactions for robot audition: a corpus of real auditory scenes. *The Journal of the Acoustical Society of America*, 144(5).
- Reyes-García, C. A., Manzanares, E. F. M.-., Escalante-Balderas, H. J. & Torres-García, A. A. (2018). *El reconocimiento de patrones y su aplicación a las señales digitales*. Amexcomp. Recuperado desde <http://amexcomp.mx/files/AprendizajeInteligenciaComputacional.pdf>
- Rill-García, R., Martínez-Carranza, J., Granados, E. & Morales, M. (2020). Path Planning and Following for an Autonomous Model Car Using an Eye in the Skin. En A. Koubaa (Ed.), *Robot Operating System (ROS) The Complete Reference (Volume 4)* (pp. 25-51). "Springer".
- Roe, D. B. & Wilpon, J. G. (1994). *Voice Communication Between Humans and Machines*. The National Academies Press.
- Romero, L., Morales, E. & Sucar, L. (2001a). A Hybrid Approach to Solve the Global Localization Problem for Indoor Mobile Robots Considering Sensor's Perceptual Limitation's. En *International Joint Conference on Artificial Intelligence IJCAI* (pp. 1411-1416). AAAI.
- Romero, L., Morales, E. & Sucar, L. (2001b). A robust exploration and navigation approach for indoor mobile robots merging local and global strategies. En *International Conference on Robotics and Automation ICRA* (pp. 3092-3097). IEEE.
- Ruiz, U., Murrieta-Cid, R. & Marroquin, J. L. (2013). Time-optimal motion strategies for capturing an omnidirectional evader using a differential drive robot. *IEEE Transactions on Robotics*, 29(5), 1180-1196.
- Turk, M. & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71-86.
- Vargas, B. & Morales, E. (2009). Learning Navigation Teleo-Reactive Programs using Behavioural Cloning. En *2009 IEEE International Conference on Mechatronics*. IEEE.
- Vasquez-Gomez, J. & Sucar, R., L.E. ans Murrieta-Cid. (2016). View/state planning for three-dimensional object reconstruction under uncertainty. *Autonomous Robots*, 41(1), 89-109.

# Capítulo 5

## Nivel de Implementación

L. Enrique Sucar Succar, [INAOE](#)

Jesús Savage Carmona, Facultad de Ingeniería, [UNAM](#)

Luis A. Pineda Cortés, [IIMAS](#), [UNAM](#)

Sergio A. Serrano, [INAOE](#)

El nivel de implementación está constituido por la plataforma de hardware del robot, que incluye las partes mecánicas y electrónicas, así como por los programas de apoyo que permiten el acceso a los dispositivos de hardware (sensores y actuadores) y la interacción entre programas y datos (sistema operativo). En este capítulo se presentan algunas plataformas robóticas orientadas a robots de servicio, y los principales programas de apoyo, en particular los sistemas operativos.

### 5.1. Plataformas de robots de servicio

De acuerdo con datos recabados por la *International Federation of Robotics* (IFR), en el periodo 2018–2020 se espera que las ventas de todo tipo de robots para tareas domésticas alcance un total de 11 mil millones de dólares a nivel global<sup>20</sup>, lo cual refleja el creciente interés del público por la robótica de servicio. Las diversas opciones para plataformas de robots de servicio se clasifican en dos categorías principales: i) robots móviles y ii) robots humanoides o bípedos. Estas plataformas

---

<sup>20</sup><https://ifr.org/news/why-service-robots-are-booming-worldwide>

comparten un conjunto significativo de características físicas y funcionales. Las plataformas más comunes para robots de servicio son los robots móviles.

En la [Figura 3.1](#) se ilustran diversas plataformas robóticas de algunos grupos en México. Estas se caracterizan por desplazarse sobre ruedas utilizando diversas estructuras, las más comunes son las estructuras diferenciales y omnidireccionales, que incluyen los siguientes elementos:

- Motores para el control de las ruedas que permiten el desplazamiento del robot sobre el piso.
- Baterías que proveen la energía eléctrica a los diversos dispositivos.
- Sensores internos que permiten conocer el estado del robot, como el nivel de la batería y su desplazamiento (odometría).
- Sensores externos para percibir su ambiente, como sonares, láser, infrarrojos, cámaras de video, cámaras de profundidad y micrófonos.
- Equipo de cómputo interno para el control de sensores, actuadores y la ejecución de los programas que proveen funcionalidad al robot.
- Equipo de comunicación que permite conectarse a la red (Internet) y usar otro equipo de cómputo externo si es necesario.
- Brazos manipuladores (generalmente uno o dos brazos) para poder tomar y transportar objetos.
- Otros elementos tales como simular una cara que proporcione al robot una cierta apariencia humanoide y le facilite la interacción con las personas.

En México, la mayoría de estos dispositivos se importan, aunque varias compañías mexicanas han logrado desarrollos de calidad en diversos componentes robóticos. Por ejemplo, el *Laboratorio de Innovación y Desarrollo Tecnológico (LAIDETEC)*, empresa que surgió del Departamento de Probabilidad y Estadística del [IIMAS, UNAM](#), por iniciativa de Hernando Ortega Carrillo, con apoyo de la incubadora de empresas de base tecnológica, InnovaUNAM, ha desarrollado brazos, manos y torsos. Las líneas principales de esta empresa son el diseño y construcción de prótesis de manos robóticas y el desarrollo de una plataforma para robots de

servicio con fines educativos y de investigación. Los brazos, el torso, las pinzas, y manos de Golem-III fueron diseñados y construidos por Hernando Ortega dentro del contexto de su participación y colaboración con el Grupo Golem. También se han desarrollado de manera experimental cabezas y rostros con la capacidad de hacer gestos y expresar emociones, como alegría, tristeza o sorpresa. Por ejemplo, la cabeza y el rostro del robot Golem-III, con la capacidad de expresar emociones, fueron desarrollados por Mauricio Reyes Castillo del Centro de Investigación en Diseño Industrial (CIDI) de la Facultad de Arquitectura de la UNAM, durante su investigación doctoral en el Posgrado de Ciencia e Ingeniería de la Computación de la UNAM y su participación en el Grupo Golem.

Por otra parte, los robots de servicio del tipo humanoide se caracterizan por utilizar un par de piernas robóticas para desplazarse. La principal ventaja de este tipo de plataforma con respecto a los robots móviles, es que dispone de una mayor flexibilidad al no verse limitado a navegar únicamente en superficies planas, sino que su diseño les permite usar escaleras, pasar por encima de objetos pequeños, y realizar casi cualquier actividad que una persona puede llevar a cabo. Sin embargo, los robots humanoides tienen la desventaja de ser menos estables y, en general, se mueven más lento que los robots móviles, por lo que su uso como robots de servicio aún es muy limitado.

En las siguientes secciones se presentan las principales plataformas de robots de servicio, móviles y humanoides, disponibles en el mercado.

### 5.1.1. Robots móviles

Pepper,<sup>21</sup> desarrollado por la compañía SoftBank Robotics, es un robot orientado a la interacción humano-robot (ver Figura 5.1). Mediante técnicas de procesamiento de lenguaje natural y procesamiento de imágenes, es capaz de iniciar y mantener conversaciones con personas de manera natural, realizando ademanes y gestos que ayudan a generar en la persona interés por la conversación. Gracias a sus habilidades sociales, algunas empresas han comenzado a emplear a Pepper como recepcionista y asistente en restaurantes, agencias de ventas y hospitales. Dentro del campo de la investigación, Pepper ha sido adoptado como una de las

---

<sup>21</sup><https://www.softbankrobotics.com/us/pepper>

principales plataformas de desarrollo, convirtiéndose en una de las plataformas estándar en la competencia RoboCupHome.<sup>22</sup>

RB-1<sup>23</sup> es un robot móvil desarrollado por la empresa Robotnik (ver Figura 5.1). Orientado al campo de la investigación, RB-1 es una plataforma reconfigurable que dispone de una base con tracción diferencial, un torso móvil, un cabezal capaz de girar e inclinarse y un brazo con 6 grados de libertad **DOF**. Entre los sensores de RB-1 se encuentran 2 cámaras RGBD (uno en la cabeza y otro en la base móvil), un sensor láser en su base, un sistema de odometría basado en el giro de sus llantas y un sensor de tipo *Inertial Measurement Unit* (**IMU**) (Ahmad, Ghazilla, Khairi & Kasi, 2013). Dadas estas características de hardware, RB-1 puede emplearse en aplicaciones de investigación y desarrollo, así como en tareas de asistencia en el hogar y manipulación de objetos en entornos interiores.

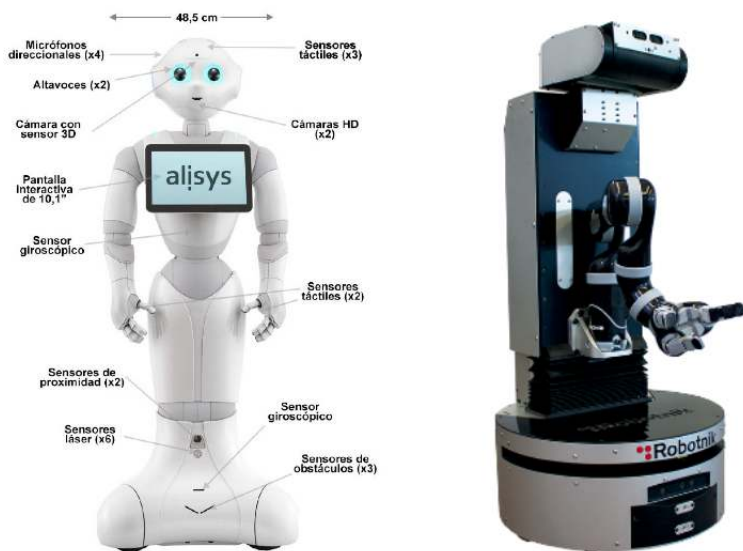


Figura 5.1: Pepper (izquierda) y RB-1 (derecha), robots de servicio móviles manufacturados por las empresas SoftBank Robotics y Robotnik, respectivamente.

Fetch<sup>24</sup>, desarrollado por la compañía Fetch Robotics (ver Figura 5.2), es un robot móvil muy parecido a RB-1 en términos de hardware, con la diferencia

<sup>22</sup><http://www.robocupathome.org/athome-spl>

<sup>23</sup><https://www.robotnik.eu/manipulators/rb-one/>

<sup>24</sup><https://fetchrobotics.com/robotics-platforms/fetch-mobile-manipulator/>

importante de que no tiene cámara RGBD en su base móvil. Además del desarrollo y la investigación, el diseño de Fetch está orientado a tareas en las que se puedan aprovechar sus capacidades de movilidad y manipulación, por ejemplo, en tareas de logística dentro de un almacén.

PR2<sup>25</sup> es un robot móvil de servicio desarrollado por la empresa Willow Garage (ver Figura 5.2). PR2 es uno de los robots móviles comerciales con mejor equipamiento, tanto por sus actuadores como sus sensores. Dispone de una base móvil omnidireccional, un torso ascendente, una cabeza capaz de girar e inclinarse y 2 brazos (con 7 grados de libertad DOF cada uno). Con respecto a los sensores, cuenta con un sensor RGBD, varias cámaras en la cabeza y brazos, acelerómetros tri-axiales y sensores de presión en sus pinzas, así como un sensor láser en la base. Por su gran cantidad de sensores y grados de libertad DOF, PR2 podría desempeñar prácticamente cualquier tarea dentro de ambientes interiores.



Figura 5.2: Fetch (izquierda) y PR2 (derecha), robots de servicio móviles manufacturados por las empresas Fetch Robotics y Willow Garage, respectivamente.

---

<sup>25</sup><http://www.willowgarage.com/pages/pr2/overview>

### 5.1.2. Robots humanoides

TALOS<sup>26</sup> es un robot humanoide (ver Figura 5.3) desarrollado por la compañía PAL Robotics. TALOS dispone de un total de 32 grados de libertad **DOF** que le proveen de una gran flexibilidad para andar en superficies irregulares, así como para mantener el equilibrio en pendientes. TALOS dispone de sensores de torque en todas sus articulaciones, así como con un sensor **IMU** para monitorear su orientación y desplazamiento en el ambiente. Con respecto a sus posibles aplicaciones, TALOS podría emplearse para desarrollar investigación en planeación de movimientos e interacción humano-robot.

ROMEO<sup>27</sup>, desarrollado inicialmente por la empresa SoftBank Robotics en conjunto con varias instituciones de Francia y de otras partes de Europa, es un robot humanoide orientado al cuidado de personas de edad avanzada o aquellas que comienzan a perder su autonomía (ver Figura 5.3). ROMEO cuenta con diversos modos de operación, los cuales varían con el objetivo de asistir al paciente con recordatorios y promover la actividad mental del mismo.



Figura 5.3: TALOS (izquierda) y ROMEO (derecha), robots de servicio humanoides manufacturados por las empresas PAL Robotics y SoftBank Robotics, respectivamente.

<sup>26</sup><http://pal-robotics.com/es/robots/talos/>

<sup>27</sup><https://projetromeo.com/>



THORMANG<sup>28</sup> es un robot humanoide desarrollado por la compañía ROBOTIS (ver Figura 5.4). La estructura de THORMANG fue diseñada de manera modular, lo que significa que, en caso de dañarse, cualquiera de sus articulaciones puede substituirse. En cuanto a sensores, cada uno de sus motores cuenta con sensores de posición y de torque, dispone de una unidad IMU, un sensor de profundidad, una cámara RGB y un sensor LIDAR, el cual le permite construir mapas del ambiente en entornos exteriores. Dada su capacidad para operar en exteriores, además de aplicaciones de robótica de servicio, THORMANG tiene el potencial para operaciones de rescate, como lo mostró con sus participaciones en el *DARPA Robotics Challenge*.<sup>29</sup>

REEM-C<sup>30</sup> es un robot humanoide desarrollado por la empresa PAL Robotics destinado a la investigación en robótica de servicio (ver Figura 5.4). REEM-C es muy parecido a TALOS en la configuración de actuadores y sensores, sin embargo, REEM-C cuenta con 19 grados de libertad DOF en cada mano, lo que le provee de la habilidad para manipular objetos pequeños con suma precisión. Esta clase de habilidades hacen que REEM-C sea idóneo para desempeñar tareas de asistencia en entornos como oficinas y hogares.

## 5.2. Sensores y actuadores

Así como las plataformas móviles y humanoides han contribuido al desarrollo de sistemas aplicados a la robótica de servicio, la disponibilidad de sensores y actuadores en el mercado, ha proporcionado flexibilidad de diseño para que estudiantes e investigadores, que no tienen acceso a una plataforma de servicio, puedan proponer sus propios diseños. En esta sección se presentan algunos de los sensores y actuadores más utilizados en la construcción de robots orientados a la robótica de servicio.

### 5.2.1. Sensores

Un sensor es un dispositivo cuyo propósito es medir algún aspecto del entorno, con el objetivo de que el robot pueda recopilar información que le sirva para tomar

---

<sup>28</sup><http://www.robotis.us/thormang/>

<sup>29</sup><https://archive.darpa.mil/roboticschallenge/teams.html>

<sup>30</sup><http://pal-robotics.com/es/robots/reem-c/>

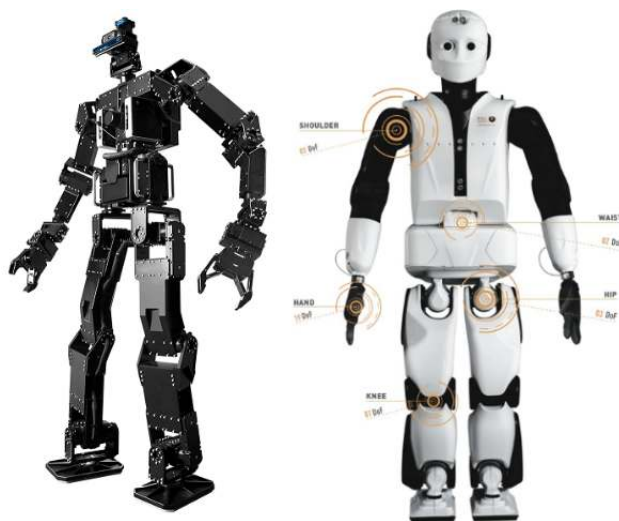


Figura 5.4: THORMANG (izquierda) y REEM-C (derecha), robots de servicio humanoides manufacturados por las empresas ROBOTIS y PAL Robotics, respectivamente.

decisiones. Actualmente, es posible encontrar una gran variedad de sensores, que son capaces de medir variables como la temperatura y humedad del entorno, hasta la aceleración del robot. Sin embargo, para un robot de servicio, el aspecto físico del entorno y su ubicación relativa al mismo, son las principales variables que debe tomar en cuenta al momento de llevar a cabo una tarea. En las siguientes secciones se presentan algunos de los sensores más utilizados.

### 5.2.2. Sensor RGBD

Los sensores *Red Green Blue Depth* (**RGBD**) comúnmente se utilizan para obtener una descripción visual del ambiente, tanto de su apariencia como de su estructura (ver Figura 5.5). El sensor **RGBD** se compone de una cámara *Red Green Blue* (**RGB**) y un sensor de profundidad empujados uno a lado del otro. En la imagen obtenida por la cámara **RGB**, el valor de sus píxeles representa el color, mientras que en la imagen de profundidad los valores de los píxeles representan la distancia entre el sensor y algún objeto.



Figura 5.5: Sensor RGBD marca Kinect (Zhang, 2012). Esta clase de sensor genera simultáneamente imágenes de color y profundidad, que se utilizan junto con técnicas de procesamiento de imágenes para inferir información de alto nivel acerca del entorno, tal como clasificar objetos, seguir personas y detectar eventos (Han, Shao, Xu & Shotton, 2013).

### 5.2.3. Sensor láser

Los sensores láser se utilizan para obtener una descripción en dos dimensiones del entorno (ver Figura 5.6). Con esta clase de sensor es posible generar una imagen que contiene los contornos de los objetos con los que colisiona el láser emitido por el sensor.

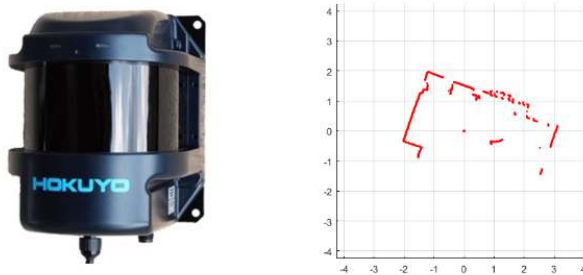


Figura 5.6: Sensor láser marca Hokuyo (Kneip, Tâche, Caprari & Siegwart, 2009). Esta clase de sensor con frecuencia se utiliza con algoritmos de localización y mapeo para construir un mapa del entorno y llevar un seguimiento de la posición del robot dentro del entorno en todo momento.

### 5.2.4. Actuadores

En robótica se considera como actuador a cualquier dispositivo que sea capaz de generar alguna señal de salida que modifique algún aspecto del entorno. Es decir,

si dentro del contexto de la tarea que un robot debe resolver hay un dispositivo que puede cambiar el estado del ambiente, hasta tal punto de solucionar la tarea, entonces es un actuador para el robot (*e.g.* un motor o una resistencia calentadora). En particular, en la robótica de servicio los actuadores más empleados son aquellos que permiten al robot desplazarse, así como mover elementos en su ambiente. Para lograrlo, generalmente se recurre a motores de corriente directa. En la siguiente sección se presentan algunos de los motores más utilizados.

### 5.2.5. Motor de corriente directa

Los motores de corriente directa (Fristedt, 2009) se usan en aplicaciones en las que se requiere de un eje girando de manera ininterrumpida por largos periodos, a alta velocidad y con un torque bajo (ver Figura 5.7). Esta clase de motores usualmente tienen 3 estados: inactivo, girando en sentido horario y en sentido antihorario. Sin embargo, también es posible regular la velocidad de giro mediante un control PWM (*pulse-width modulation*).



Figura 5.7: Motor de corriente directa, comúnmente utilizado para el desplazamiento de bases móviles y como rotor en drones.

### 5.2.6. Motor a pasos

Los motores a pasos (Markkanen, Blackwell, Knaust, Kropac & Mccall, 1971) se utilizan en aplicaciones que requieren de una velocidad de giro, torque y precisión media (ver Figura 5.8). Están constituidos por varios circuitos (embobinados) independientes, que permiten que el eje realice desplazamientos angulares predefinidos por la distribución de los embobinados en la carcasa, a diferencia del motor de corriente directa.

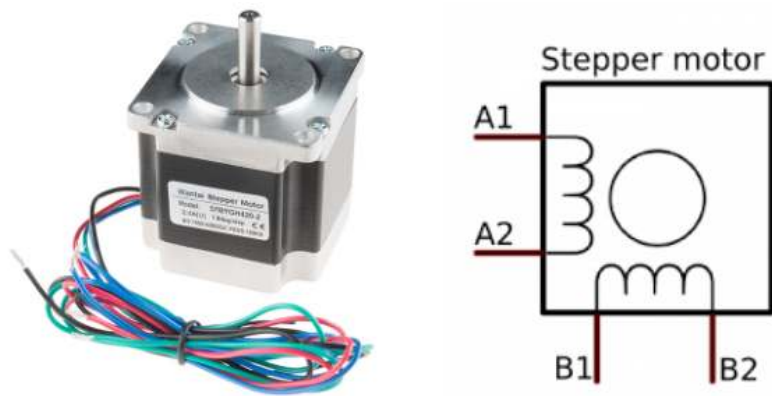


Figura 5.8: Motor a pasos con dos bobinas independientes: A y B, las cuales permiten al eje del motor tomar 4 posibles posiciones, separadas por  $90^\circ$ .

### 5.2.7. Servomotor

Los servomotores (Murphy, McMullin, Egan & Brosnan, 1987) se utilizan en aplicaciones que requieren de una velocidad de giro baja, y torque y precisión altos. Un servomotor está compuesto por un motor de corriente directa cuyo eje de salida se encuentra acoplado a una caja de engranes. La caja de engranes tiene la función de reducir la velocidad de giro e incrementar el torque. El servomotor también cuenta con un circuito de control, cuyo propósito es ajustar la posición angular del eje, en ocasiones con una resolución de fracción de grado (ver Figura 5.9).



Figura 5.9: Servomotor analógico controlado por una señal de entrada de tipo PWM y cuyo eje es capaz de tomar posiciones dentro del intervalo  $[0^\circ, 180^\circ]$ .

### 5.3. Programas de apoyo

Los programas de apoyo proveen la interfaz entre los algoritmos de alto nivel y el hardware del robot, en particular los sensores y actuadores (ver Figura 5.10). Consisten de un conjunto de manejadores (*drivers*) y bibliotecas que facilitan el acceso a cada tipo de dispositivo del robot y, normalmente, dependen de la plataforma robótica y dispositivo específico. Adicionalmente, los programas de apoyo deben incluir los servicios análogos a los que proveen los sistemas operativos en las computadoras, como el manejo de usuarios, etc. Para ello se han utilizado sistemas operativos desarrollados para equipo de cómputo genérico, en particular sistemas basados en UNIX, como LINUX orientados a los sistemas que funcionan en tiempo real.

Debido a la gran variedad de sensores y actuadores que se encuentran disponibles en el mercado, y a la necesidad de homogeneizar el desarrollo de programas para aplicaciones en robótica, recientemente han surgido sistemas operativos enfocados en robots, dentro de los cuales se encuentra ROS<sup>31</sup> que se ha convertido en un estándar en la comunidad de robótica, en especial entre académicos (Quigley et al. 2009).

ROS provee un conjunto de bibliotecas que, una vez instaladas, pueden incorporarse a programas escritos en C++ y Python para comunicarse con otros programas que también integran ROS, con el objetivo de resolver problemas con

<sup>31</sup><https://www.ros.org/>

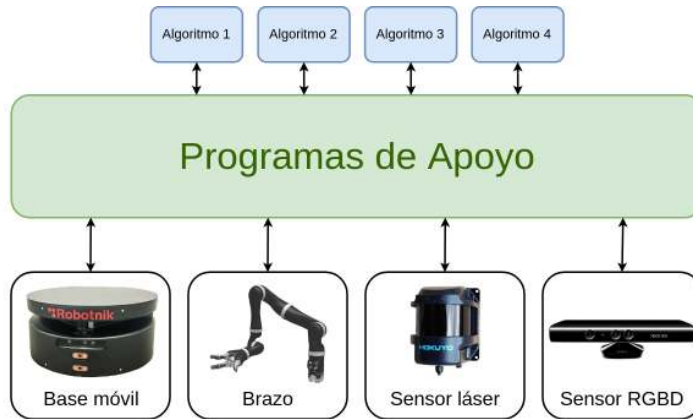


Figura 5.10: Interacción de los programas de apoyo con los algoritmos y los dispositivos del robot. Mientras que los sensores recaban datos, los programas de apoyo los transportan hacia los algoritmos, que los procesan para decidir cuál acción tomar. Una vez que los algoritmos han definido el comando que se ejecutará, este se transmite al correspondiente actuador.

un enfoque distribuido. Además, [ROS](#) facilita la interoperabilidad de los programas en diferentes plataformas y permite compartir algoritmos entre los grupos de investigación, promoviendo la colaboración y el desarrollo de la robótica. En la sección siguiente se presentan los conceptos principales de [ROS](#), así como una lista de sitios en los que se pueden encontrar material introductorio, con documentación y ejemplos de [ROS](#).

## 5.4. *Robot Operating System*

[ROS](#) está conformado por un conjunto de herramientas cuyo propósito es proveer un medio común en el que desarrolladores puedan colaborar para construir sistemas robóticos robustos y de alta complejidad. Con [ROS](#) se busca desarrollar sistemas modulares, con distribución de las tareas y que promuevan la reutilización de código. Es decir, en lugar de implementar en un solo programa todos los algoritmos que un robot podría necesitar, estos se distribuyen en varios programas más sencillos y se establece cuáles programas dependen de la salida de otros programas para realizar su propia tarea.

Al distribuir el problema original en varios programas, se tienen varias ventajas, en comparación con la implementación de un solo programa:

1. Varios desarrolladores pueden trabajar simultáneamente en distintos módulos.
2. Debido a que el sistema está constituido por varios módulos independientes, es más sencillo identificar errores de implementación en el sistema.
3. De ser necesario, es posible integrar en un sistema varias instancias de un mismo módulo.
4. Ya que aROS es un medio de comunicación entre programas, puede servir para integrar en un solo sistema, bibliotecas que resulten ser incompatibles entre sí, por ejemplo, bibliotecas que son compatibles con distintos lenguajes de programación. Actualmente, ROS ofrece soporte para *C++* y *Python*.

### 5.4.1. Funcionamiento de ROS

En general, en ROS los sistemas funcionan como redes de sub-sistemas interconectados, los cuales están diseñados para resolver sub-partes del problema completo y enviar dicha solución a los sub-sistemas que la requieran. En ROS existen cuatro conceptos básicos que, *grosso modo*, describen los componentes elementales sobre los cuales se puede desarrollar cualquier sistema.

Se le llama *nodo* a un programa, el cual puede estar implementado en *C++* o *Python*. Los nodos son los responsables de resolver subtareas y comunicar las soluciones para que otros nodos las puedan utilizar como entradas y resolver otras subtareas. En la Figura 5.11 hay cuatro nodos interactuando entre sí para implementar un sistema robótico que involucra dos actuadores (base móvil y brazo) y dos sensores (láser y RGBD).

En ROS, se le llama *tópico* a los canales de comunicación mediante los cuales los nodos comparten información entre ellos. Cada tópico puede comunicar un sólo tipo de dato, ya sea numérico, booleano, o una estructura arbitraria. De esta manera, los nodos pueden estar seguros que cuando reciban información mediante cierto tópico, el mensaje recibido cumplirá con cierto formato que el nodo receptor requiere para desempeñar su tarea. En la Figura 5.11 se muestra como el sistema cuenta con tres tópicos que comunican a cuatro nodos.



Se le llama *publicar* a la acción de un nodo de enviar información a un tópico. De esta forma, cada vez que un nodo publique un dato en un tópico, todos los nodos que se encuentren suscritos a dicho tópico recibirán el dato recién publicado. En la Figura 5.11, los nodos 3 y 4 publican las mediciones obtenidas por sus sensores después de aplicarles una etapa de preprocesamiento, en los tópicos A y B, respectivamente. De la misma manera, los nodos 1 y 2 publican en el tópico C para comunicarse entre sí.

Se le llama *suscribirse* al proceso de registrar un nodo como receptor a un tópico. De esta forma, cada nodo recibirá únicamente la información que es relevante para la tarea que es responsable de resolver. En la Figura 5.11, el nodo 1 está suscrito al tópico A porque requiere de información del sensor láser, para que la base móvil no colisione. A su vez, el nodo 2 se encuentra suscrito al tópico B para utilizar la información del sensor de profundidad. Esta información se requiere para identificar la posición de un objeto y definir la secuencia de movimientos que le permitirán agarrarlo.

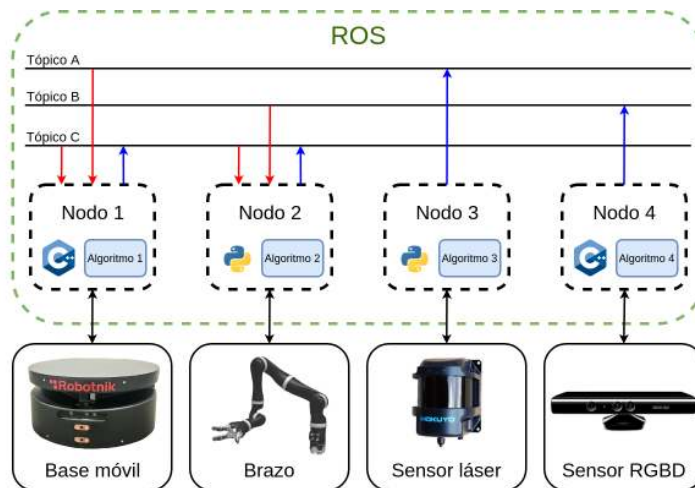


Figura 5.11: Estructura de un sistema implementado como un conjunto de nodos en ROS. En este ejemplo, los nodos 3 y 4 se encargan de publicar la información proveniente de los sensores del robot. Después, los nodos 1 y 2 se suscriben a los tópicos donde se publica la información sensorial, para realizar (posiblemente) tareas de navegación y manipulación de objetos. Además, el tópico C sirve para que los nodos 1 y 2 se comuniquen entre sí para coordinar sus acciones.

En el ejemplo de la Figura 5.11, se puede apreciar como se puede modularizar un sistema en cuatro nodos, como podría ser uno de los robots de servicio móviles presentados en la sección 5.1.

### 5.4.2. Aprendiendo ROS

Gracias a que la comunidad de desarrolladores ha mostrado un creciente interés en ROS desde su nacimiento, existen una gran cantidad de recursos para aprender a implementar sistemas dentro el entorno de ROS. Para poder utilizar ROS, es necesario disponer de alguna versión de UBUNTU.

Para aprender los conceptos de ROS, en internet se pueden encontrar diversos tutoriales que instruyen al lector, mediante ejemplos, como puede obtener el máximo provecho de las herramientas disponibles en ROS para el desarrollo de sistemas.<sup>32</sup>

Debido a que muy pocos desarrolladores disponen de una plataforma robótica sobre la cual probar sus sistemas, surgió GAZEBO<sup>33</sup> (Koenig & Howard, 2004) el cual es un simulador en donde se pueden cargar modelos de robots reales. El modelo de un robot incluye modelos de sus sensores y actuadores, de esta forma es posible simular el funcionamiento completo del robot. En internet se pueden encontrar diversos tutoriales para conectar ROS con GAZEBO con el propósito de simular robots.<sup>34</sup>

## 5.5. Conclusiones

Con empresas como *SoftBank Robotics* y *PAL Robotics*, las cuales han desarrollado plataformas comerciales, se ha creado una oportunidad de acelerar el trabajo de investigación en el área de robótica de servicio. Con plataformas de robots estándar es posible que los desarrolladores localizados en distintos institutos puedan compartir su trabajo, que otros lo utilicen como punto de partida y, a su vez, compartan su contribución al trabajo original. De esta manera, la comunidad entera de investigadores y desarrolladores pueden beneficiarse de un proceso

---

<sup>32</sup><http://wiki.ros.org/ROS/Tutorials>

<sup>33</sup><http://gazebosim.org/>

<sup>34</sup><http://gazebosim.org/tutorials>

continuo de retroalimentación. En ROS podemos encontrar un ejemplo perfecto de las ventajas que trae consigo una tecnología estándar de código abierto y multiplataforma.

Además de disponer de plataformas comunes de desarrollo, es igual de importante el acceso a dispositivos independientes, como sensores y actuadores. De esta manera, es posible diseñar y proponer nuevos sistemas orientados a aplicaciones particulares que las plataformas estándar no, necesariamente, toman en cuenta. Por otra parte, la integración de sensores y actuadores a plataformas estándar puede significar una alternativa viable para compensar las limitantes de los sistemas actuales, conforme nuevos problemas son abordados.

## 5.6. Referencias

- Ahmad, N., Ghazilla, R. A. R., Khairi, N. M. & Kasi, V. (2013). Reviews on various inertial measurement unit (IMU) sensor applications. *International Journal of Signal Processing Systems*, 1(2), 256-262.
- Fristedt, T. (2009). Method and arrangement for control of direct current motor. US Patent 7,567,045. Google Patents.
- Han, J., Shao, L., Xu, D. & Shotton, J. (2013). Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5), 1318-1334.
- Kneip, L., Tâche, F., Caprari, G. & Siegwart, R. (2009). Characterization of the compact Hokuyo URG-04LX 2D laser range scanner. En *2009 IEEE International Conference on Robotics and Automation* (pp. 1447-1454). IEEE.
- Koenig, N. & Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. En *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)* (Vol. 3, pp. 2149-2154). IEEE.
- Markkanen, C. O., Blackwell, D. B., Knaust, G. A., Kropac, C. R. & McCall, J. M. (1971). Stepper motor control system. US Patent 3,586,953. Google Patents.
- Murphy, J., McMullin, F., Egan, M. & Brosnan, M. (1987). Servomotor control systems. US Patent 4,661,756. Google Patents.

- 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, 3.2, p. 5). Kobe, Japan.
- Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2), 4-10.

# Capítulo 6

## Retos y Perspectivas

L. Enrique Sucar Succar, [INAOE](#)

Jesús Savage Carmona, Facultad de Ingeniería, [UNAM](#)

Luis A. Pineda Cortés, [IIMAS](#), [UNAM](#)

Los robots son máquinas que pueden percibir y actuar en el mundo, controlados por procesos computacionales para realizar diversas tareas. Se pueden distinguir diferentes tipos de robots de acuerdo con su autonomía y la complejidad de su entorno, destacando los robots de servicio de media o alta autonomía que operan en entornos de complejidad baja o moderada. Estos robots están orientados a asistir a las personas en diversas actividades y ambientes, los cuales se pueden conceptualizar en tres niveles de sistema: i) funcional, ii) de dispositivos y algoritmos y iii) de implementación. El nivel funcional corresponde a la especificación y coordinación de las capacidades del robot; se han desarrollado diferentes esquemas a nivel funcional incluyendo las máquinas de estado, los lenguajes de especificación de tareas y los procesos de decisión de Markov. El nivel de algoritmos incluye los elementos computacionales que proveen las principales capacidades al robot: i) percepción y conducta motora, ii) audición y lenguaje, iii) conocimiento y razonamiento. El nivel de implementación está constituido por los elementos mecánicos y electrónicos del robot, y por los programas de apoyo y sistema operativo.

En México hay varios grupos activos en investigación y desarrollo en robótica, que han hecho aportaciones en los diferentes niveles y aspectos, con énfasis en los

robots de servicio. Una forma de impulsar y evaluar los desarrollos en robótica es la participación en competencias, donde destaca RoboCup<sup>35</sup> a nivel internacional, en la cual participan regularmente los grupos mexicanos, y el Torneo Mexicano de Robótica<sup>36</sup> en nuestro país.

El desarrollo de los robots de servicio está en sus inicios y se espera que estos artefactos en el futuro sean ubicuos en nuestras casas, oficinas, hospitales, etc., como los televisores y las computadoras. Sin embargo, para lograrlo todavía existen retos importantes por conquistar, incluyendo los siguientes:

- En cuanto a los aspectos mecánicos y electrónicos, se necesitan robots que puedan desenvolverse en ambientes como los que habitamos los humanos, para lo que se requiere robots humanoides que sean más robustos y rápidos que los actuales, con capacidades de desplazarse en ambientes complejos y manipular diferentes tipos de objetos. Deben mejorarse los sistemas de almacenamiento de energía (baterías) para que tengan una mayor autonomía.
- Aunque las capacidades de percepción, planeación y manipulación han evolucionado en los últimos años todavía falta mucho por hacer; por ejemplo que los robots puedan doblar prendas de ropa como una persona, abrir cajones y tomar objetos ocluidos, o desplazarse en ambientes desconocidos y dinámicos, entre otros.
- Dado que los robots de servicio tendrán que resolver tareas muy diversas, una capacidad fundamental es que puedan aprender de su experiencia o con ayuda del usuario, de forma que puedan realizar nuevas tareas o mejorar las que ya conocen. También deberán poder utilizar, como nosotros, los recursos disponibles en Internet, incluyendo textos, fotos y vídeos.
- La interacción con los humanos es fundamental para los robots de servicio, incluyendo la interacción a través del lenguaje natural hablado apoyada por ademanes. Además, para que logren una mayor empatía con sus usuarios deberán ser capaces de reconocer su estado emocional y simular ciertas emociones.

---

<sup>35</sup><http://www.robocup.org/>

<sup>36</sup><https://femexrobotica.org/>

Actualmente ya que hay robots de servicio realizando ciertas tareas relativamente sencillas como aspirar la casa o cortar el pasto; y poco a poco van surgiendo robots que realizan tareas más complejas como recepcionistas en hoteles y oficinas, ayudantes en casas de personas de la tercera edad, entre otras. Podemos esperar que esta tendencia siga a corto y mediano plazo, con robots de servicio que realizan ciertas tareas específicas; y seguramente serán cada vez más comunes en diversos ambientes, incluyendo en las casas, oficinas, hospitales, tiendas, etc. Sin embargo, un robot de servicio con capacidades más genéricas que pueda realizar cualquier tipo de tarea como una persona está aún lejos de hacerse realidad, ya que esto implicaría la solución de la inteligencia artificial general, que se ve aún lejana.





## Apéndice A

# Investigación en Robótica en México

Yasmín Hernández, [INEEL](#)

La robótica ha demostrado ser una tecnología que puede ayudar en muchos aspectos de la vida diaria, en los procesos industriales, así como en actividades riesgosas. A nivel global, el mercado de la robótica ha ido en aumento por lo que también ha habido un incremento importante en la inversión en la investigación y desarrollo en robótica. En México ha habido un gran interés en la aplicación de la robótica y es uno de los principales compradores a nivel mundial. Sin embargo, México también es una potencia en el desarrollo de robots, ya que los investigadores mexicanos han desarrollado diferentes prototipos que permiten apreciar el potencial que tenemos en el diseño y construcción de robots. Es por esta razón que la academia se ha preocupado por impulsar el desarrollo y la investigación, con lo que se ha podido desarrollar tecnología propia.

A lo largo de los últimos años, en México han surgido diversos grupos de investigación en robótica. Muchos de ellos, además de dedicarse a la investigación, también se enfocan en el desarrollo de capacidades que les permitan ser competitivos en los torneos de prestigio. Todos estos grupos de investigación son semilleros de científicos en robótica que han seguido formando nuevos talentos y nuevos grupos de investigación. Entre las instituciones más reconocidas en es-

ta área se encuentran: Benemérita Universidad Autónoma de Puebla (BUAP), Centro de Investigación en Computación (CIC), Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), CIMAT, Cinvestav Campus Cd. de México, Cinvestav Campus Guadalajara, Cinvestav Campus Saltillo, Facultad de Ingeniería UNAM, IIMAS-UNAM, INAOE, ITAM, Tecnológico de Monterrey Campus Edo. de México, Tecnológico de Monterrey Campus Monterrey, Universidad LaSalle, Universidad Panamericana, Universidad Politécnica de Victoria, Universidad Tecnológica de la Mixteca y la Universidad Popular Autónoma del Estado de Puebla (UPAEP).

En este apéndice presentamos algunos de estos grupos de investigación.

## A.1. Grupo de Robótica del ITAM

El grupo de robótica del ITAM<sup>37</sup> se formó hace más de 20 años por parte del Dr. Alfredo Weitzenfeld con el apoyo de otros profesores del ITAM. Actualmente, el grupo trabaja bajo la coordinación del Dr. Marco Morales Aguirre. Algunos de los estudiantes que han participado en este grupo son: Paolo Aguilar Valiente, Andrés Akle Carranza, Carlos Jesús Alegría Ramírez, María Fernanda Borge Chavez, Alan Córdoba Posadas, Jorge Alfredo Delgado Meraz, Alejandro Escalante Arrieta, Edgar Alejandro Granados Osegueda, Mariana Raquel Hernández Rocha, Brandon Hernández Troncoso, Miriam Seth Martínez Guzmán, Eduardo Martínez Neri, Karen Poblete Rodríguez, Jorge Joaquín Popoca Herrera, Jorge Luis Pérez Rentería, Diego Pozo Barruel, Leonardo Miguel Ramos Morán, Gibran Ríos Tamayo, Ruiciro Rivera Serrano, Javier Sagastuy Breña, Humberto Isaac Téllez Benítez, Ernesto Torres Vidal, Héctor Armando Vidrio Morgado, y Erick Zetina Muciño.

El objetivo de este grupo es realizar proyectos de investigación y desarrollo en Robótica Autónoma e Inteligencia Artificial (IA) y participar en competencias como la RoboCup. Los antecedentes de este grupo datan de 1997, aunque se estableció formalmente en 2003 cuando inició su preparación para participar en competencias de robótica nacionales e internacionales con el nombre *Eagle Knights*. Desde 2005, ha calificado para participar en las ligas *Small Size League* (SSL) y *Standard Platform League* (SPL) de la Competencia Mundial RoboCup. También

---

<sup>37</sup><http://robotica.itam.mx>

ha participado en el Torneo Mexicano de Robótica (TMR), en la categoría *RoboCup Major Standard Platform League*, en donde obtuvieron el 1er Lugar en 2015 y el 2o Lugar en 2017 y 2019. Otra participación destacada de este equipo se dio en 2014 en el 1er Concurso de Robótica e Inteligencia Artificial NAO, en donde obtuvieron el 1er Lugar en la Categoría Tiro Penal. En la [Figura A.1](#) se muestra a algunos de los integrantes de *Eagle Knights* durante su participación en el TMR.



Figura A.1: Integrantes del Grupo *Equipo Eagle Knights* del grupo de robótica del ITAM.

### A.1.1. Robot ITAM-EKBot

Los robots ITAM-EKBot, que se muestran en la [Figura A.2](#), fueron diseñados en el ITAM para participar en la RoboCup SSL. Para el desarrollo de estos robots se han abordado retos que van desde la mecánica, electrónica, control y programación. Han habido múltiples versiones de estos robots; en las primeras versiones el diseño se basó en procesadores digitales de señales mientras que las versiones más recientes están basadas en micro-controladores y FPGAs. El desarrollo de este robot también incluye un simulador compatible con Gazebo y ROS.

Además de los robots ITAM-EKBots, el equipo del ITAM ha abordado retos con robots estándar que permiten enfocarse en retos de sensado, mapeo, navegación, planificación de movimientos y de tareas. Este es el caso de los perritos robots AIBO de Sony y los robots Nao de Aldebarán que se usan en [SPL](#).

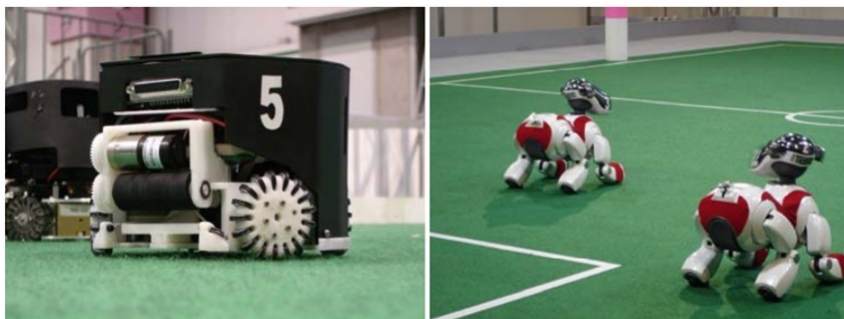


Figura A.2: Robots del Grupo de Robótica del [ITAM](#)

### A.1.2. Simulador de robot AutoNOMOs

En 2016, el [ITAM](#) recibió uno de los robots AutoNOMOs que donó Raúl Rojas de la *Freie Universität Berlin* a varias universidades Mexicanas para promover el desarrollo de coches autónomos. En el [ITAM](#) se desarrolló un simulador de este robot, que es un coche a escala 10:1, para permitir agilizar las pruebas de algoritmos para este robot antes de utilizar el robot físico. Además, esto ayuda en el aprendizaje de la plataforma aún sin contar con el robot.<sup>38</sup>

### A.1.3. Videos

- [Eagle Knights Qualification 2016](#)
- [Eagle Knights 2014](#)
- [Eagle Knights qualification RoboCup Austria 2009](#)

---

<sup>38</sup>El simulador se puede obtener en [https://github.com/ITAM-Robotica/EK\\_AutoNOMOS\\_Sim](https://github.com/ITAM-Robotica/EK_AutoNOMOS_Sim)

## A.2. Grupo Golem

El Grupo Golem<sup>39</sup> se creó en 1998 y es parte del Departamento de Ciencias de la Computación del IIMAS de la UNAM. El grupo es dirigido por el Dr. Luis Alberto Pineda Cortés. Entre los miembros del equipo se encuentran los investigadores: Dr. Gibrán Fuentes, Dr. Arturo Rodríguez, Dr. Mauricio Reyes, M. en C. Hernando Ortega, M. en C. Noé S. Hernández, y los estudiantes: Ricardo Cruz, Iván Torres, Dennis Mendoza, Uriel Ortíz. Este grupo cuenta con la colaboración de investigadores de otras instituciones, por ejemplo: Dr. Caleb Rascón, M.C. Ivette Vélez, Dr. Ivan Meza, Dra. Wendy Aguilar, M.C. Cinthya Ceja y el Dr. Jorge Garcia.

El objetivo del grupo Golem es desarrollar robots de servicio para asistir a las personas en su quehacer diario, con una orientación fuerte a la IA y a la interacción Humano-Robot. El trabajo de este grupo involucra investigación teórica, desarrollo de software y la construcción real de robots. Los robots creados por el grupo se exhiben frecuentemente en Universum, Museo de las Ciencias, de la UNAM. El grupo Golem participa en el TMR y en competencias de la RoboCup internacional en la categoría @Home. En la Figura A.3 se muestran algunos de los miembros de este equipo.



Figura A.3: Grupo Golem

---

<sup>39</sup><http://golem.iimas.unam.mx>

### A.2.1. Robot Golem

Golem es un robot de servicio capaz de mantener una conversación sencilla y entender comandos sencillos de movimiento. En 2006, la tecnología de administración de diálogos y de reconocimiento de voz en español fue suficientemente robusta, por lo que una versión de Golem fue presentado en el museo de la ciencia Universum en Junio de 2007. En la [Figura A.4](#) se muestra a Golem.



Figura A.4: Golem

### A.2.2. Robot Golem II+

Golem II+ es un robot de servicio apto para interpretar expresiones lingüísticas y visuales que da un visita guiada en el Departamento de Ciencias de la Computación, mostrando carteles de investigación de los proyectos de dicho departamento. Este robot tiene implementadas las tareas de la competencia RoboCup@Home. Se trata de un robot PeopleBot (TM) con el equipo siguiente: Tres arreglos de sonares con ocho sensores cada uno, dos sensores IR al frente, dos arreglos de sensores al tacto con cinco sensores cada uno, micrófonos y bocinas, computadora interna VersaLogic EBX-12. Adicionalmente, se le ha agregado el siguiente equipo: laptop Dell Precision M4600, brazos robóticos LAIDETEC-IIMAS, Webcam QuickCam Pro 9000, Cámara Microsoft Kinect, Láser Hokuyo UTM-30LX, Micrófono omnidireccional Shure Base x3, Micrófono direccional RODE VideoMic, Interface de sonido externa M-Audio Fast Track, Bocinas Infinity 3.5-Inch Two-Way x2. En la [Figura A.5](#) se muestra a Golem II+.



Figura A.5: Golem II+

### A.2.3. Robot Golem III

Golem III es un robot de servicio orientado al lenguaje y la comunicación. Este robot piensa, habla y tiene la capacidad de recibir comandos, sostener una conversación en lenguaje hablado con humanos, además de hacer diagnósticos de su entorno, tomar decisiones y resolver problemas sencillos de la vida cotidiana. Más allá de realizar labores de servicio como tomar objetos y transportarlos, Golem III también evalúa la situación presentada e intenta responder a la instrucción de la manera más eficiente.

Golem-III consta del siguiente hardware: Un arreglo de sonares con 8 sensores, base PatrolBot, dos arreglos protectores cada uno con 5 sensores al tacto, bocinas bidireccionales Infinity de 3.5 pulgadas, computadora Cobra EBX-12, láser Sick LMS-500. Además, se le ha añadido el siguiente hardware: dos computadoras laptop Dell Precision M7510, láser localizador Hokuyo SOKUIKI, switch ethernet Black Box alimentado por USB con 5 puertos, cámara Microsoft Kinect 2,

cámara de alta resolución Point Grey Flea USB 3, interfaz de audio 8SoundsUSB, tres micrófonos miniatura, micrófono direccional RODE VideoMic, torso, brazos y cuello construidos dentro del grupo. En la [Figura A.6](#) se muestra a Golem III.



Figura A.6: Golem III

#### A.2.4. Videos

- [Golem III en la prueba de Manipulación y Reconocimiento de Objetos, en la competencia RoboCup@Home 2016](#)
- [Golem III \(Rulebook RoboCup@Home 2015\)](#)
- [Golem II en la televisión](#)
- [Golem II, un robot que interactúa](#)



### A.3. Grupo Markovito

El grupo Markovito<sup>40</sup> se formó hace más de 12 años y está integrado por investigadores, técnicos y estudiantes del Laboratorio de Robótica de la Coordinación de Ciencias Computacionales del [INAOE](https://ccc.inaoep.mx/~markovito/).

Actualmente, el coordinador del grupo es el Dr. L. Enrique Sucar Succar, y los miembros del equipo son: Dr. Eduardo Morales, Dr. José Martínez Carranza, Víctor Lobato, Irving Bravo, Arquímedes Méndez, Reinier Oves, Adilene Palma, Esaú Eleazar, Sergio A. Serrano, David Carrillo López.

Este equipo ha participado con muy buenos resultados en competencias tales como: Concurso Mexicano de Robótica, Concurso Latinoamericano de Robótica y el Concurso RoboCup@Home. En la [Figura A.7](#) se muestra a algunos miembros del Grupo Markovito con el robot Markovito Jr.



Figura A.7: Integrantes del Grupo Markovito

---

<sup>40</sup><https://ccc.inaoep.mx/~markovito/>

### A.3.1. Robot Sabina

Este robot tiene como objetivo ayudar en las labores domésticas y está preparado para reconocer los patrones de su entorno y aprender a realizar de forma autónoma diversas tareas de la vida diaria, incluyendo ir a ciertos lugares, y llevar una bebida o un medicamento. Este robot está dotado de capacidades sensoriales, programas que le ayudan a interpretar lo que observa y un software para comprender simples órdenes. Está equipado con sensores láser, un Xbox Kinect, una cámara que le permite capturar la información en 3D, lo que le permite identificar a las personas y su posición. Sabina ganó el 1er. lugar en la categoría *@Home* en el TMR 2016 y 2015. En la [Figura A.8](#) se muestra a Sabina.



Figura A.8: Robot Sabina

### A.3.2. Robot Markovito

Es un robot basado en una plataforma PeopleBot. Markovito puede ver, escuchar e interactuar con humanos en un entorno de la vida real, como una sala de estar o una oficina. La función principal de Markovito es asistir en diferentes labores a personas de la tercera edad, discapacitadas y enfermas. Este robot está provisto de sensores, cámaras, tres computadoras, dos controladores Kinect, micrófonos, bocinas y un brazo robótico. De esta manera, Markovito puede calcular distancias, reconocer objetos, detectar obstáculos, manipular objetos, así como interactuar con usuarios. En la [Figura A.9](#) se muestra a Markovito.



Figura A.9: Robot Markovito

### A.3.3. Robot Markovito Jr.

Markovito Jr. es un robot de servicio que incorpora un conjunto de módulos de propósito general que proveen habilidades básicas tales como: i) construcción de mapas, ii) localización y navegación, iii) reconocimiento y seguimiento de objetos y personas, iv) interacción con las personas, a través del habla y la manipulación. Dichos módulos están integrados en una arquitectura en capas basada en el comportamiento e implementada en ROS. En la Figura [figura A.10](#) se presenta el nuevo robot Markovito Jr.



Figura A.10: Robot Markovito Jr. del INAOE

La plataforma robótica es de tracción diferencial (modelo RB-1 de Robotnik, modificada por el equipo Markovito) con tres ruedas onnidireccionales de carácter auxiliar (ó pasivas). La base tiene integrada una cámara de profundidad Astra 3D, una telémetro láser Hokuyo, una computadora Intel NUC i7, un *router* de red y una batería con química de litio que le dan 10 horas aproximadas de autonomía. Sobre la base se fija un torso telescópico y una cabeza articulada, junto con un brazo robótico de 6 grados de libertad y una mano con 3 dedos y falanges pasivas. Además, en la espalda del robot se integró una computadora laptop extra para ayudar en los procesos más demandantes. La cabeza articulada puede vol-*tear* y *mirar* hacia arriba y abajo dentro de ciertos límites. También cuenta con

un micrófono de alta direccionalidad para propósitos de reconocimiento de voz y bocinas que trabajan en conjunto con el sintetizador de voz. El proyecto *Aprendiendo de Internet*, consiste en el desarrollo de nuevos algoritmos que permiten a Markovito Jr aprender a buscar un objeto en Internet, reconocerlo, localizarlo en un espacio y finalmente entregarlo al usuario que se lo haya pedido. De esta manera, se le pueden solicitar objetos para las cuales no estaba programado y en lugar de que se pare, puede buscar en Internet y trata de solucionar el problema. Eso sirve para que el robot, una vez que conoce el objeto y sabe dónde está, ya no tenga que hacer de nuevo todas estas operaciones. Si le vuelven a pedir ese objeto ya la conoce y sabe dónde está.

#### A.3.4. Videos

- [Sabina en el torneo mexicano de robótica de 2015](#)
- [Sabina en Robocup@home 2014](#)
- [Markovito en la televisión](#)
- [Markovito en el Campus Party México 2009](#)
- [Markovito Jr. en acción](#)

### A.4. Grupo Nanisha

El grupo Nanisha está formado por estudiantes de la Universidad Popular Autónoma del Estado de Puebla bajo la dirección del Dr. Héctor Simón Vargas Martínez. Este grupo tiene como objetivo desarrollar tecnología que los haga competitivos en concursos regionales, nacionales e internacionales. Así mismo tienen como objetivo aplicar sus tecnologías en el cuidado de adultos mayores. Este equipo desarrolló diferentes prototipos, hasta que llegaron a Nanisha, un robot de servicio con el cual iniciaron sus participaciones en los torneos de robótica. Sin embargo, gracias al trabajo del equipo, el robot siguió evolucionando hasta llegar al robot Donaxi. En la [Figura A.11](#) se muestra a algunos de los miembros del grupo Nanisha.



Figura A.11: Grupo Nanisha

#### A.4.1. Robot Nanisha

Nanisha es un robot de servicio que integra sensores, una cámara, y programas en LabView que monitorean los sensores y llevan a cabo la toma de decisiones del robot. Dentro de las funciones que realiza este robot se encuentra la identificación de luz infrarroja por medio de los sensores que tiene en la parte superior y, gracias a la cámara que porta en el frente, puede seguir una línea e identificar un semáforo, así como el obedecer a los señalamientos de este. El Dr. Vargas eligió el nombre Nanisha por su significado: *algo que se hace con gusto*. En la [Figura A.12](#) se muestra a Nanisha.

#### A.4.2. Robot Donaxi

Donaxi es un robot de servicio para el hogar diseñado para ayudar a personas de la tercera edad. Donaxi, que en lengua zapoteca significa *princesa*, puede reconocer voces y personas, manipular objetos, limpiar, moverse en un ambiente conocido, obedecer órdenes, recordar rostros e identificarlos por su nombre, seguir a alguien evitando que se caiga y, en caso de caer, Donaxi puede llamar al número de emergencias. Donaxi fue desarrollado con el fin de que los investigadores y es-



Figura A.12: Robot Nanisha

tudiantes de la [UPAEP](#) pudieran probar y experimentar nuevos algoritmos en las diferentes áreas de la inteligencia artificial. En la [Figura A.13](#) se muestra a Donaxi.

#### A.4.3. Videos

- [Robot Nanisha, prueba seguimiento infrarrojo](#)
- [Robot Nanisha, prueba seguimiento infrarrojo](#)
- [Robot Nanisha, prueba seguimiento de personas y reconocimiento de voz](#)
- [Robot Nanisha, prueba reconocimiento de voz](#)
- [Robot Nanisha, prueba seguimiento de línea](#)
- [Donaxi en el TMR](#)

### A.5. Grupo Pumas

El Grupo Pumas<sup>41</sup> se formó hace más de 20 años y está integrado por investigadores y estudiantes de la Facultad de Ingeniería, del posgrado de Ingeniería

<sup>41</sup><https://biorobotics.fi-p.unam.mx/>



Figura A.13: Robot Donaxi

Eléctrica de la [UNAM](#), área de procesamiento de señales y del Posgrado de Ciencias e Ingeniería en Computación de la [UNAM](#). En él también participan profesores de la Facultad de Ingeniería y del [IIMAS](#) de la [UNAM](#), así como colaboradores que se encuentran en el extranjero. En la Figura [Figura A.14](#) se muestra el Grupo Pumas durante su participación en RoboCup@Home en Montreal 2018.

El grupo está bajo la coordinación del Dr. Jesús Savage, colaborando junto con los profesores: Dr. Boris Escalante, Dr. Carlos Rivera, Dr. David Rosenblueth, M.C. Stalin Muñoz y M.I. Yukihiro Minami y con los siguientes estudiantes de licenciatura, maestría y doctorado: Rubén Anaya, Carlos Munive, Marco Morales, Emmanuel Hernández, Gabriel Vázquez, Javier Jiménez, Sergio Cuellar, Adalberto Hernández, Gerardo Carrera, Francisco Rodríguez, David Cortes, David Esparza, Rafael Sobrevilla, Rogelio Cruz, Edna Marquez, Francisco Ayala, Luis Torres, Marco Negrete, Rommel Sánchez, Allan Reid, Rodrigo Savage, Saiph Savage, Luis Sannabra, Mauricio Matamoros, Marco Becerra, Luis Contreras, Ábel Pacheco, Ismael Castillo, Israel Figueroa, Iván Sánchez, Francisco Dorantes, Alejandra Sánchez, Juan Alvarez, Claudia Favela, Alejandro Macías, Angel Molina, Jaime Marquez, Laura López, Nonato Lagunas, Joaquín Díaz, Alejandro Bermúdez, Lisette Castro, Jesús Cruz, Cecilia Gómez, Adrián Revuelta, Lauro Vázquez, Samuel Vázquez, Germán Bastida, Luis Juárez, Luis Duran, Pablo Monroy, Reynaldo Martell, Julio





Figura A.14: Grupo Pumas: Sub-campeones de las categorías *Open Platform League* y *Domestic Standard Platform League* de RoboCup@Home en Montreal 2018

Cruz, Hugo León, José Cruz, Edgar Silva, Edgar Vázquez, Adrián Sarmiento, Luis González, Héctor Arce, Angélica Nakayama, Daniel Ruelas, Daniel Garces, Julio Martínez, Jesús Coyotzin y Manuel Pano.

Este grupo de investigación tiene como objetivo el desarrollo e investigación en robots de servicio, a través de promover el desarrollo de aplicaciones del mundo real y la interacción hombre-máquina con robots autónomos. Tradicionalmente, el grupo PUMAS construye o adapta sus robots (la mayoría son de manufactura propia), siendo su última creación la sexta iteración del robot Justina, cuya descripción se presenta en el el [Apéndice C](#). Desde 2006, el Grupo Pumas ha participado en el torneo internacional de robótica RoboCup@Home con muy buenos resultados, perfilándose de manera sostenida como finalista desde 2014, y coronándose como sub-campeón de liga en Montreal 2018 (véase [Figura A.14](#)) y Sydney 2019, y con un bien merecido bronce en Atlanta 2007.

### A.5.1. Robot Justina

Justina (véase [Figura A.15a](#)) fue desarrollado por estudiantes de posgrado dirigidos por el Dr. Savage Carmona. A este robot se le han hecho adecuaciones con diferentes tecnologías a lo largo de los años. Se trata de un robot en constante evolución cuyo hardware se encuentra ya en la sexta iteración y que recibe constantes mejoras en el software que lleva desarrollándose más de doce años. Uno de los elementos más interesantes de Justina es su capacidad de interactuar con las personas que encuentra en su entorno y darse cuenta de los cambios que suceden en el ambiente, lo que incrementa su capacidad para recibir y seguir órdenes. Un factor clave en este proceso es el avanzado reconocimiento facial para el cual ocupa un sensor [RGBD](#) tipo Kinect.

### A.5.2. Robot Takeshi

Takeshi (véase [Figura A.15b](#)) es un robot desarrollado en Japón por la empresa Toyota, con la cual la [UNAM](#) firmó un acuerdo de colaboración para lograr desarrollar sus capacidades a través de la programación y diferentes tecnologías implementadas por el Grupo Pumas. El *Human Support Robot* ([HSR](#)) de Toyota es un robot prototipo desarrollado por la división de investigación y desarrollo de la transnacional japonesa, el cual se otorga como préstamo a grupos de investigación experimentados tras un riguroso proceso de selección.<sup>42,43</sup> Cabe mencionar que Takeshi utiliza el mismo software de alto nivel que usa Justina, lo que permite verificar la robustez y adaptabilidad del mismo en diferentes plataformas. Se espera que Takeshi se comercialize en el corto plazo.

### A.5.3. Videos

- [Pumas@home Qualification Video RoboCup 2018](#)
- [Pumas@home Qualification Video RoboCup 2017](#)
- [Pumas@Home Robocup 2014 Qualification Video](#)
- [Takeshi ejecutando la prueba Help me carry en RoboCup@home2018](#)

---

<sup>43</sup> [https://www.toyota-global.com/innovation/partner\\_robot/robot/](https://www.toyota-global.com/innovation/partner_robot/robot/)

<sup>43</sup> <https://robots.nu/en/robot/human-support-robot-hsr-service-robot->



(a) Justina

(b) Takeshi

Figura A.15: Robots del Grupo Pumas

## A.6. Grupo RyM

El grupo Robótica y Mecatrónica (RyM)<sup>44</sup> del CIC del IPN trabaja bajo la coordinación del Dr. Juan Humberto Sossa Azuela, los miembros del grupo son: Dr. Carlos Fernando Aguilar Ibáñez, Dra. Elsa Rubio Espino, Dr. Jesús Yaljá Montiel Pérez y Erik Zamora Gómez, además de contar con la participación de estudiantes. En la Figura A.16 se muestran algunos de los miembros del Grupo RyM.

<sup>44</sup><http://148.204.64.54/LabRM/LabRM/LabRM.html>



Figura A.16: Grupo RyM

El grupo RyM realiza investigación teórica y desarrolla aplicaciones en los campos de la robótica y la mecatrónica. Para esto combinan técnicas de cómputo suave como el control difuso, tratamiento y análisis digital de imágenes, reconocimiento de patrones, redes neuronales, memorias asociativas, modelado mediante redes de Petri y optimización no convencional, para ser aplicados en: control automático, modelado de sistemas lineales, no lineales y discretos, simulación y análisis de las dinámicas de los sistemas, visión por computadora, reconocimiento de patrones, control de robots humanoides, móviles y aéreos, así como; control por medio de técnicas de lógica difusa utilizando dispositivos de lógica programable FPGA.

El objetivo de este grupo es el desarrollo de modelos neuro-computacionales para aplicación en el diseño y control de robots. En la Figura A.17 se muestran algunos de los robots en los que el Grupo RyM ha probado sus modelos.

### A.6.1. Videos

- Seminario de Robótica impartido por el Dr. Sossa, coordinador del Grupo RyM
- Entrevista al Dr. Humberto Sossa, coordinador del Grupo RyM

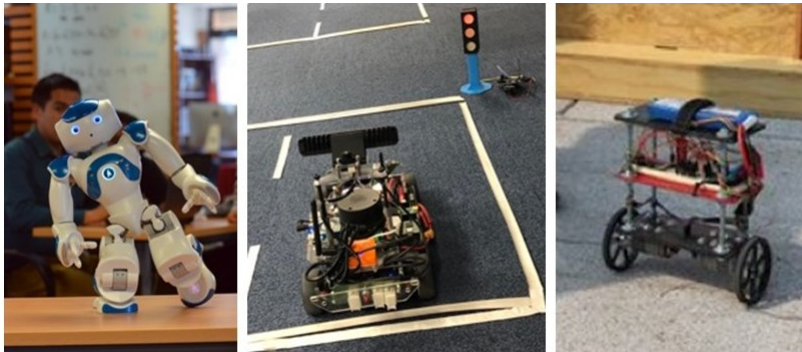


Figura A.17: Robots del Grupo RyM

## A.7. Grupo RyMA

El grupo Robótica y Manufactura Avanzada (RyMA)<sup>45</sup> (Robótica y Manufactura Avanzada) del Cinvestav del IPN está conformado por los investigadores: Dr. Gustavo Arechavaleta Servín, Dr. Arturo Baltazar Herrejón, Dr. Mario Castelán, Dra. Nadia Vanessa García Hernández, Dr. Ismael López Juárez, Dra. América Berenice Morales Díaz, Dr. Ernesto Olgún Díaz, Dr. Keny Ordaz Hernández, Dr. Vicente Parra Vega, Dr. Reyes Ríos Cabrera, Dr. Francisco José Ruiz Sánchez, Dr. Anand Eleazar Sánchez Orta y Dra. Luz Abril Torres Méndez, bajo la supervisión del Dr. Chidentree Treesatayapun. En la Figura [Figura A.18](#) se presenta un grupo de estudiantes del Grupo RyMA, quienes ganaron el primer lugar en la primera competencia de vehículos autónomos a escala. Estos vehículos fueron donados por el gobierno alemán y desarrollados por el Laboratorio de Inteligencia Artificial de la Universidad Libre de Berlín, dirigido por el científico mexicano Raúl Rojas. Los estudiantes son: Alejandro Maldonado Ramírez, Noel Cortés Pérez, María de la Paz Rico Fernández, Álvaro Paz, Jonathan Obregón.

El Grupo RyMA se especializa en tres líneas de investigación principales: Robótica, Manufactura Avanzada y Visión por Computadora, y se compone de cuatro laboratorios: i) Laboratorio de Visión Robótica Activa, RAVG, en el cual se desarrollan proyectos en donde la visión activa es esencial para la robótica e IA, tales como detección y reconocimiento de objetos, aprendizaje profundo, planeación de trayectorias, reconstrucción 3D, inspección visual, mejoramiento de

<sup>45</sup><http://ryma.cinvestav.mx/>

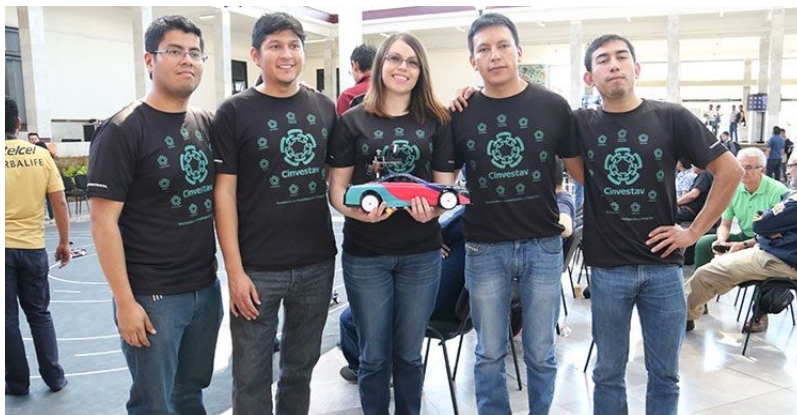


Figura A.18: Estudiantes del Grupo [RyMA](#)

imágenes acuáticas, robótica submarina basada en visión, análisis de formas; ii) Laboratorio de Manufactura Inteligente, LABMIC, se enfoca en robótica industrial y técnicas avanzadas de manufactura para resolver problemas reales, tales como el seguimiento de costura de soldadura, auto-reconfiguración de celda de manufactura, aprendizaje por demostración, robots colaborativos (robots móviles) para la Industria 4.0; iii) Laboratorio de Evaluación no Destructiva y Sensado Inteligente, LabENDSI, este laboratorio está dedicado a resolver problemas científicos y tecnológicos en el área de Evaluación No Destructiva (ultrasonido) y Sensado Inteligente; y iv) Laboratorio de Robótica no Inercial e Interfaces Hombre Máquina, RONIHM, el cual abarca líneas de investigación sobre control de drones, interfaces hombre-máquina para rehabilitación, simulaciones, control, entre otros.

Con estos laboratorios, [RyMA](#) se especializa en 3 líneas de investigación principales: Robótica, Manufactura Avanzada y Visión, en donde se desarrollan diversos proyectos sobre robótica aérea, robótica submarina, robots móviles, robots industriales, detección de objetos, visión robótica y visión acuática, entre otros. En la [Figura A.19](#) se muestran algunos de los robots del Grupo [RyMA](#).

### A.7.1. Videos

- Seminario *Exploración Robótica de Arrecifes Coralinos* impartido por la Dra. Luz Abril Torres Méndez del Grupo [RyMA](#)

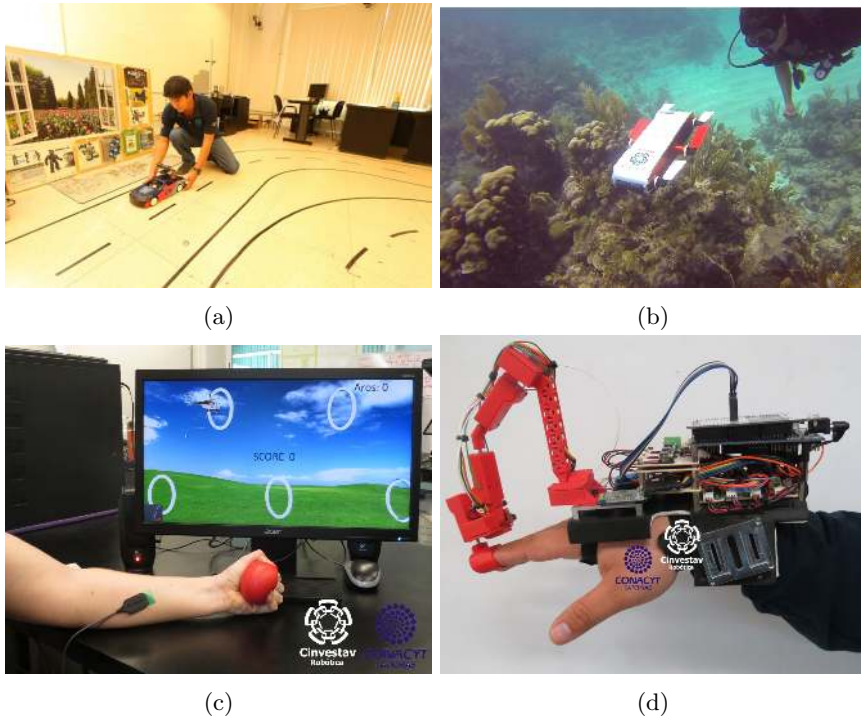


Figura A.19: Robots del Grupo RyMA: (a) Vehículo autónomo a escala, (b) Robot acuático Mexibot, (c) Exergame para ejercicios isométricos y fortalecimiento muscular, y (d) Exoesqueleto de mano portable.

- Proyectos desarrollados en la Nave de Manufactura por el grupo del Dr. Ismael López Juárez del Grupo RyMA





## Apéndice B

# Torneos de Robótica en México

Marco Morales Aguirre, [ITAM](#)

Las competencias de robótica han sido muy populares en México. Algunos de los mas destacados son: El [TMR](#), las competencias de *For Inspiration and Recognition of Science and Technology* ([FIRST](#)): [FIRST](#) Lego League Jr, [FIRST](#) Lego League, [FIRST](#) Tech Challenge y [FIRST](#) Robotics Competition y Guerra de Robots del [IPN](#). Debido a que el [TMR](#) aborda los retos mas cercanos a la Robótica de Servicio, se aborda con mas detalle a continuación.

Alrededor del año 2000, varios grupos de investigadores de universidades mexicanas iniciaron la organización de competencias de robótica en México con el propósito de fomentar el interés en la robótica en estudiantes de todo el país. Al paso de los años, estas competencias crecieron en participación tanto individual como institucional. Esto dio lugar a la creación del [TMR](#) en el 2008, con su primera edición en el Palacio de Minería de la [UNAM](#) en la Ciudad de México. La característica distintiva del [TMR](#) es que consiste en competencias de robots autónomos y restringen la intervención humana al mínimo dado que los robots deben tomar sus decisiones con base en el modelo que tengan de su entorno y a la interpretación de datos obtenidos mediante sus sensores.

En este mismo contexto, varios equipos mexicanos se formaron para participar en la competencia mundial de robótica RoboCup. La primera participación oficial de un equipo mexicano en esta competencia fue en 2002 y desde entonces la participación mexicana ha crecido principalmente en las siguientes ligas: *Four Legged*, *Small Size*, *Humanoid KidSize*, *Soccer Simulation*, *@Home*, *Junior Rescue*, *Junior Soccer*, *Junior Dance*, *Junior CoSpace* y *@Work*. La experiencia de estos equipos se llevó al **TMR** que desde su primera edición ha sido reconocido como el *RoboCup Mexican Open*.

Para canalizar estos esfuerzos, en el año 2010 se creó la Federación Mexicana de Robótica (**FMR**). Agrupa a investigadores de diversas instituciones que se han ido integrando para organizar competencias de robótica nacionales e internacionales. Sus principales objetivos incluyen: la organización de competencias regionales, nacionales e internacionales; establecer colaboraciones con grupos similares de todo el mundo; organizar redes de investigación en robótica; y diseminar entre el público en general resultados de investigación en robótica.

El **TMR** da un espacio a varias competencias de la RoboCup. En las de RoboCup Junior participan niños y jóvenes de educación básica y media. Las competencias de RoboCup Junior en el TMR son: *Dance*, *Soccer*, *Rescue*, y *CoSpace*. En las competencias de RoboCup Major participan estudiantes de educación superior y posgrado. Las competencias de RoboCup Major en el **TMR** son: *Humanoids KidSize*, *Standard Platform League*, *@Home*, y *Rescue*.

Otras competencias del **TMR** han sido motivadas por intereses más específicos de los investigadores involucrados. La competencia de Robots Limpiadores de Playa ha sido muy exitosa y ya ha dado lugar a su reproducción en otros países. También se incluye la eliminatoria nacional del *Latin American Robotics Competition* en las categorías de *Standard Educational Kit* y *Open Challenge* para ir a la correspondiente competencia internacional organizada por el *Latin American Robotics Council* de IEEE Región 9. Recientemente se incluyó la competencia de *AutoModel Car* para coches autónomos.

En el pasado se han incluido otras competencias como las competencias de Robothon (organizada por la *Seattle Robotics Society*).

En la mayoría de las competencias del **TMR**, el equipo ganador recibe apoyo económico para participar en la competencia internacional correspondiente.

Cada año, desde el 2008, el Torneo Mexicano de Robótica ha establecido *records* de participación alcanzando en la edición del 2018, 1055 participantes de 240 equipos de todo México (Figura B.1).



Figura B.1: TMR 2008

El éxito de la FMR se ha sostenido gracias a la participación de voluntarios en todos sus eventos. La mayoría de estos voluntarios son estudiantes.

En el 2012, la Federación de RoboCup y la FMR organizaron en conjunto la competencia mundial de RoboCup en la Ciudad de México con la participación de alrededor de 2500 competidores y una asistencia estimada de 35,000 espectadores. Esta competencia generó mucho interés en todo el país y tuvo una cobertura mediática enorme. En la Figura B.2 se muestra el cartel de RoboCup 2012 en México.



Figura B.2: RoboCup 2012 en México

# Apéndice C

## El robot *Justina*

Jesús Savage Carmona, Facultad de Ingeniería, [UNAM](#)

Mauricio Matamoros, *University of Koblenz*

David A. Rosenblueth, [IIMAS](#), [UNAM](#)

*Justina* es un robot de servicio diseñado en la Facultad de Ingeniería de la [UNAM](#). El sistema operativo que opera a *Justina*<sup>46</sup> está basado en una arquitectura híbrida de robots móviles, que consiste en cuatro capas: entrada, planeación, conocimiento y ejecución. Cada una de ellas cuenta a su vez con varios subsistemas que controlan en forma general la operación del robot. Operativamente, los componentes de software se comunican entre sí utilizando [ROS](#) (Jesús Savage et al. 2016). En la [figura C.1](#) se presenta el esquema de *Justina*.

### C.1. Capa de entrada

En la capa de entrada se procesan las señales provenientes de los sensores internos y externos con los que cuenta el robot, utilizando técnicas de procesamiento digital de señales, para generar una representación simbólica de ellas. Por ejemplo, si el robot está situado enfrente de una mesa en la que hay objetos, esta capa indicaría los tipos y nombres de objetos encontrados. Si alguien le da un comando de voz, el robot realiza primero el reconocimiento de palabras y después construye

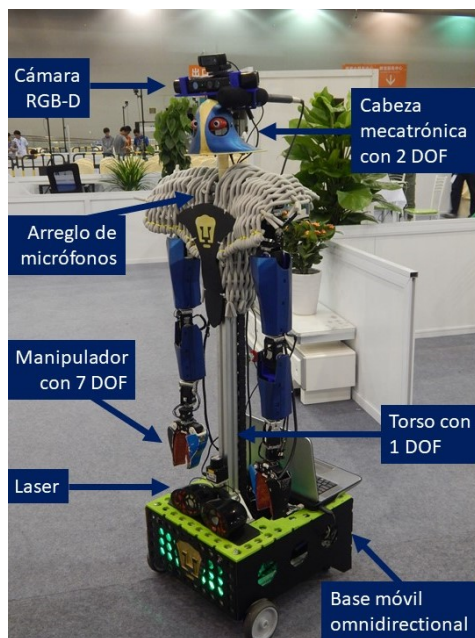


Figura C.1: Robot Justina.

una representación de la oración. Esta representación puede ser utilizada después por las siguientes capas para realizar lo que se solicita.

La capa de entrada se conforma de los siguientes subsistemas:

- **Sensores internos:** Provee una capa de abstracción que almacena el estado del robot y sirve de interfaz de los sensores internos, incluyendo: codificadores en las ruedas, sensor de carga de la batería, sensor de inclinación, sensor de temperatura, etc.
- **Sensores externos:** Provee una capa de abstracción sirve de interfaz de los sensores que detectan el medio ambiente, tales como: contacto, infrarrojos, cámaras, micrófonos, láser, sonares, etc.
- **Interpretación/representación simbólica:**  
 Genera representaciones simbólicas abstrayendo los datos proporcionados por los sensores utilizando técnicas de procesamiento digital de señales

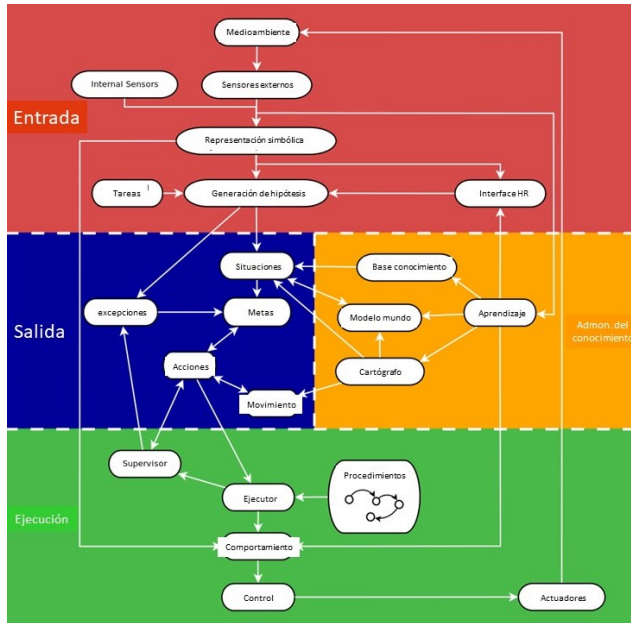


Figura C.2: Modelo Conceptual del Robot Justina.

- **Tareas del robot:** Permite programar tareas para ejecutarse a determinada hora o cuando ciertas condiciones se cumplan.
- **Interfaz humano/robot:** Comprende la serie de módulos que permiten al robot comunicarse de manera natural con los seres humanos, incluyendo reconocimiento y síntesis de voz, gestos, y expresiones faciales simples.
- **Percepción:** Genera una serie de creencias o hipótesis que representan los posibles estados del mundo con base en las representaciones simbólicas de los sensores, las tareas almacenadas, y los datos de interacción de la interfaz humano-robot.

## C.2. Capa de planeación

En la capa de planeación se planean, valga la redundancia, las acciones que tiene que realizar el robot para resolver una tarea o cumplir con un objetivo.

- **Validación de situaciones:** Valida las creencias generadas en la capa de entradas, permitiendo reconocer situaciones nuevas en el medio ambiente y descartar situaciones imposibles.
- **Activación de metas:** Activa una serie de metas u objetivos con base en las nuevas situaciones detectadas.
- **Planeador:** Genera un plan de acción para realizar la tarea u objetivo a partir de las funciones disponibles en el banco de procedimientos, incluyendo inferencias lógicas (planeador de acciones mediante búsquedas en representaciones espacio-estado) y cálculo y optimización de rutas con evasión de colisiones (planeador de movimientos *A*).
- **Reconocedor de excepciones:** Supervisa la ejecución del plan y lo modifica en caso de que sucedan cosas inesperadas.

### C.3. Capa de representación del conocimiento

La capa de conocimiento representa y administra todo conocimiento del robot, y está integrada por los siguientes módulos.

- **Modelo del mundo:** Contiene las descripciones de los objetos, las personas y los lugares con los que interactúa el robot, así como relaciones que existen entre ellos.
- **Cartógrafo:** Contiene los mapas de los entornos en los que se mueve el robot. Estos pueden ser creados manualmente, o de forma automática por el robot usando [SLAM](#).
- **Representación del conocimiento:** El resto del conocimiento del robot está codificado como un sistema basado en reglas construido en *C Language Integrated Production System (CLIPS)*. Emplea reglas de producción que corresponden a ciertas condiciones que ocurren en situaciones cotidianas, detallando la semántica de la física del mundo.
- **Aprendizaje:** El sistema de aprendizaje puede aprender a resolver problemas nuevos usando:



- Algoritmos genéticos y programación genética.
- Métodos probabilísticos usando cadenas de Markov.
- Agrupamiento utilizando cuantización vectorial.
- Redes neuronales artificiales.

## C.4. Capa de ejecución

La capa de ejecución lleva a cabo los planes generados en la capa de planeación.

- **Ejecutor:** Intenta ejecutar los planes de acción y de movimientos utilizando procedimientos que resuelven parcialmente problemas específicos.
- **Supervisor:** Utiliza una copia del plan enviado al Ejecutor y supervisa que este se lleve a cabo en tiempo y forma.
- **Comportamientos de reacción:** Comprende un conjunto de comportamientos reactivos alambrados para imprevistos no contemplados por el planeador de movimientos, tales como evasión de obstáculos. Los comportamientos pueden ser máquinas de estados, campos potenciales, redes neuronales, etc.
- **Banco de procedimientos:** Encapsula un conjunto de procedimientos que resuelven parcialmente problemas específicos (normalmente usando máquinas de estados). Estos incluyen para tomar objetos, limpiar mesas, encontrar personas, alinearse a una pared o muebles, etc.
- **Algoritmos de Control:** Conjunto de algoritmos encargados de realizar el control de bajo nivel de los actuadores. En su forma más simple, se utilizan controladores *Proportional-integral-derivative* (PID).
- **Actuadores:** Provee una capa de abstracción que permite el acceso al hardware del robot (que puede ser real o virtual). Una interfaz gráfica 3D permite probar diversos robots virtuales, es decir, simulaciones de robots reales. Esto permite probar primero en los robots virtuales y después en el real, usando Rviz y Gazebo bajo el sistema operativo de robots ROS. Los robots virtuales pueden ejecutar las mismas órdenes que los robots reales, así como los comportamientos, las ecuaciones de movimiento y las lecturas de los sensores.

### C.4.1. Máquinas de estados

En estas se tiene una representación del estado en el cual está el sistema y dadas ciertas condiciones de entradas y el estado presente se calcula el estado siguiente. La parte fundamental de las máquinas de estado es el algoritmo que estas ejecutan. En la [Figura C.3](#) se muestra un robot omnidireccional que cuenta con dos motores que le permiten ir hacia adelante, atrás y hacer giros hacia la derecha e izquierda, asumiendo que solamente puede hacer giros de 45 grados.

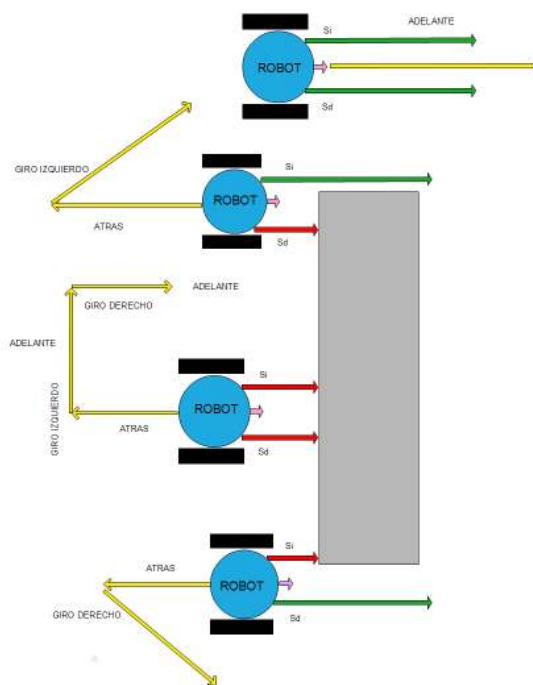


Figura C.3: Un robot omnidireccional que evade un obstáculo.

Por ejemplo, la forma más directa para programar la navegación de un robot móvil es a través del uso del concepto de máquinas de estados. Cuenta con dos sensores que le permiten detectar obstáculos, como se muestra en la [Figura C.3](#). Si el robot no detecta ningún obstáculo, seguirá avanzado; si detecta un obstáculo con su sensor izquierdo, primero se desplaza una distancia predeterminada hacia

atrás y después gira hacia la derecha 45 grados; si detecta un obstáculo con su sensor derecho, primero se desplaza hacia atrás y después gira hacia la izquierda 45 grados; si detecta un obstáculo con los dos sensores entonces el robot se desplaza hacia atrás, gira dos veces hacia la izquierda, avanza hacia adelante y finalmente gira hacia la derecha dos veces. Esta explicación se muestra en el algoritmo de la [Figura C.4](#).

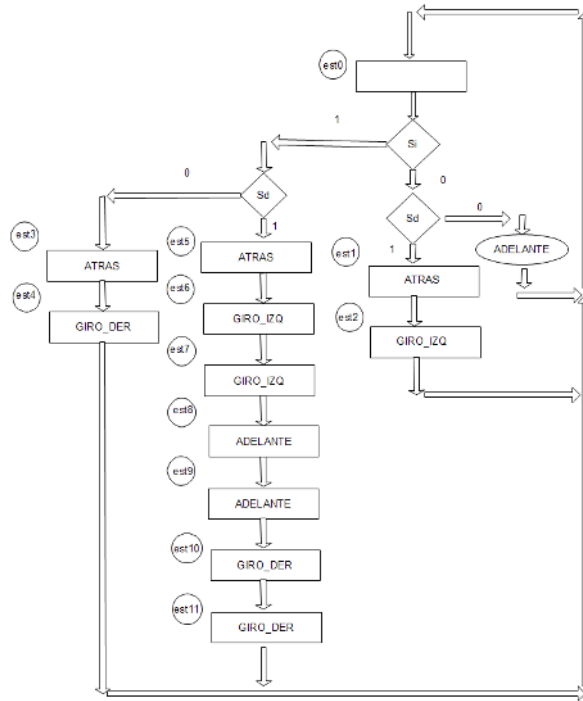


Figura C.4: Algoritmo para evadir obstáculos

Las máquinas de estados pueden programarse desde muy bajo nivel, construyéndolas usando *flip-flops* y compuertas *AND* y *OR*, hasta un nivel muy alto, usando lenguajes de programación como C++ y Python, entre otros. También se pueden utilizar para su instrumentación redes neuronales retroalimentadas en *FPGAs* (Jesus Savage et al. 2016).

## C.5. Referencias

- Savage, J. [Jesus], Cruz, J., Matamoros, M., Rosenblueth, D. A., Muñoz, S. & Negrete, M. (2016). Configurable Mobile Robot Behaviors Implemented on FPGA Based Architectures. En *16th IEEE Int. Conf. on Autonomous Robot Systems and Competitions ICARSC* (pp. 317-322).
- Savage, J. [Jesús], Negrete, M., Matamoros, M., Cruz, J., Lagunas, R. & Márquez, J. (2016). The Role of Robotics Competitions for the Development of Service Robots. En *IJCAI Workshop on «Autonomous Mobile Service Robots»*.

Robótica de Servicio  
Se terminó el 31 de octubre de 2019.  
A partir del 1 de diciembre del 2019  
está disponible en formato PDF  
en la página de la  
Academia Mexicana de Computación:  
<http://www.amexcomp.mx>