



Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado

Guía de Personas con Robots Móviles
Basado en la Interacción Humano-Robot

T E S I S

Para obtener el grado de:

Doctora en Ingeniería Eléctrica
con opción en Control Automático

Presenta:

M.I. Karla Lourdes Luna Gallegos

Asesores:

Dra. Elvia Ruth Palacios Hernández
Dr. Antonio Marín Hernández

San Luis Potosí, S. L. P.

Febrero de 2018





UASLP

Universidad Autónoma
de San Luis Potosí

16 de marzo de 2017

**M.I. KARLA LOURDES LUNA GALLEGOS
P R E S E N T E**

En atención a su solicitud de Temario, presentada por los **Dres. Elvia Ruth Palacios Hernández y Antonio Marín Hernández** Asesora y Co-asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de **Doctora en Ingeniería Eléctrica**, me es grato comunicarle que en la Sesión del H. Consejo Técnico Consultivo celebrada el día 16 de marzo del presente año, fue aprobado el Temario propuesto:

TEMARIO:

“Guía de Personas con Robots Móviles Basado en la Interacción Humano-Robot”

Introducción.

1. Robots Móviles de Servicio: Antecedentes y Evolución.
2. Estudio del escenario de robots móviles e interacción humano-robot.
3. Desarrollo de algoritmos inteligentes implementados en robots móviles.
4. Experimentación y resultados.

Conclusiones.

Apéndices.

Referencias.

“MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO”

A T E N T A M E N T E



M. I. JORGE ALBERTO PÉREZ GONZÁLEZ
DIRECTOR
UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
DIRECCION



**FACULTAD DE
INGENIERÍA**

Av. Manuel Nava 8
Zona Universitaria • CP 78290
San Luis Potosí, S.L.P.
tel. (444) 826 2330 al39
fax (444) 826 2336
www.uaslp.mx

Copia. Archivo
*etn.

*Dedicado a mi familia,
con todo mi cariño, respeto y admiración.
A mi hijo, por ser la motivación más grande para terminar este proyecto.*

Agradecimientos

A mi familia, por el apoyo incondicional que me han brindado y por acompañarme en cada etapa de mi vida.

A mis asesores de tesis por transmitirme sus conocimientos y experiencia, por su ayuda y paciencia.

A mis profesores de posgrado quienes fueron formadores de habilidades y capacidades durante el doctorado.

A mis sinodales por los consejos y observaciones.

Al Consejo Nacional de Ciencia y Tecnología por haberme becado durante mis estudios.

A mis compañeros y amigos, al igual que a todas aquellas personas que colaboraron directa o indirectamente en la elaboración de esta tesis.

Resumen

En este documento se presenta el desarrollo e implementación del marco de trabajo de una nueva técnica de interacción humano-robot para la guía de personas a través de plataformas móviles con ruedas con el fin de facilitar la integración de los robots en la sociedad.

Actualmente, existe un problema de control de movimiento para el seguimiento de objetivos con robots móviles. Este problema se ha abordado desde dos enfoques diferentes: el robot sigue a alguien o algo y el robot desea que lo sigan. Las leyes de control utilizadas para resolver estos problemas, generalmente utilizan retroalimentación mediante la percepción visual para ajustar la velocidad y posición del robot. En el caso de un robot guía de turistas, no solo debe considerar el seguimiento de trayectoria, también debe tener en cuenta otro objetivo de control, el cual es el asegurar que el humano lo viene siguiendo.

La técnica de interacción propuesta consiste en un sistema de control que regule la velocidad del robot durante el recorrido tomando en cuenta la velocidad estimada del usuario en tiempo real. Para llevar a cabo esta tarea se deben desarrollar diferentes habilidades en el robot. Tareas como el reconocimiento y localización del usuario, la navegación autónoma, el control de movimiento, la percepción del entorno y la comunicación con el usuario deben integrarse y ejecutarse de forma paralela, en caso de que una falle, no se logrará el objetivo final. El robot desarrollado en este trabajo de tesis es llamado “Tbot” y opera en los pasillos de un edificio universitario en donde se encuentran las oficinas de algunos docentes de la institución. El robot cuenta con una variedad de sensores que le permiten obtener información de su entorno.

La comunicación entre las diferentes tareas y módulos en el robot se realiza mediante el Sistema Operativo de Robots (ROS); el cual permite el paso de información entre un proceso y otro de manera paralela. Gracias a esto, el software desarrollado en este trabajo de investigación, posee la capacidad de ejecutarse en diferentes plataformas robóticas con ruedas, es decir se tiene *portabilidad*. Para comprobar esta portabilidad, se utilizó el robot “UVerto” desarrollado en el Centro de Investigación en Inteligencia Artificial de la Universidad Veracruzana (CIIA – UV), que opera en los pasillos y laboratorios del CIIA.

La actividad de guía de personas inicia con la detección y localización de un usuario en el entorno, con el fin de interactuar con el usuario, en este espacio tridimensional se calcula la velocidad de este con base en la distancia que recorre en intervalos de tiempo determinados. Además, el robot cuenta con la información del mapa de su entorno, el cual es previamente reconocido, conoce su punto inicial y la ubicación de los lugares a los que el usuario puede dirigirse. Con esta información se realiza el cálculo de trayectorias; para lo cual inicialmente se genera una trayectoria global del punto inicial al punto final realizando un cuadrículado en el mapa

y definiendo las celdas libres por las que puede pasar el robot. En caso de existir obstáculos no previstos dentro del mapa, como lo pueden ser otros humanos u objetos depositados anteriormente, se calculan trayectorias locales. Este cálculo consiste en conocer la velocidad lineal y angular del robot, así como la proximidad de obstáculos, del punto meta y de la trayectoria global y con esta información, se realiza una simulación de la posición próxima del robot al cabo de un corto periodo de tiempo, para predecir si el robot colisionará.

Para poder ejecutar dichas trayectorias, pero sobre todo para hacer corrección de errores durante el proceso de guía, es necesario conocer la localización del robot dentro del mapa en todo momento. Esta localización se realiza mediante en algoritmos probabilísticos (odometría, telemetría, localización de Monte Carlo aumentada). El movimiento del robot depende de sus velocidades lineal y angular, estas variables se ajustan por medio de un controlador. Para incluir al humano en el lazo de control se diseñó un proceso inteligente adaptativo para el seguimiento de trayectorias de robots móviles basado en la lógica difusa con un lazo interno y otro externo. La lógica difusa se utiliza debido a que al incluir al humano dentro del lazo de control, el proceso de seguimiento se complica debido a que no existen modelos matemáticos precisos que expresen el comportamiento de las personas. El lazo interno consiste en un controlador PI+D difuso para adaptar la velocidad lineal del robot a la velocidad estimada del usuario. Para el lazo externo se tiene como entrada la velocidad lineal del robot obtenida en el lazo interno y el error en orientación; el cual se calcula a partir de la trayectoria deseada y la posición actual del robot. La interacción del robot con las personas se realiza por medio de normas sociales como el respeto al espacio personal, el límite de la velocidad máxima del movimiento del robot, su aspecto físico y la modulación de la velocidad del robot durante el recorrido. Además, para entablar la comunicación con el usuario se utiliza una pantalla y un sintetizador de voz. La adaptación de velocidad del robot en función de la del usuario, busca brindar un mejor servicio a este último, ya que incrementará las probabilidades de que el usuario lo siga y ambos lleguen a su objetivo. Debido a que el control de las velocidades del robot está basado en reglas sociales de interacción entre humanos y robots, es decir considera el comportamiento de las personas de su entorno, evitará accidentes y brindará mayor seguridad y comodidad a las personas. Este sistema puede implementarse para robots guía en distintos entornos como museos, hospitales, universidades, etc., en donde puede aplicarse para la tarea de guía turístico, para el cuidado de adultos mayores, rescate de personas donde el robot debe guiar al usuario a la salida, entre otras.

Índice general

Lista de figuras	X
Lista de tablas	XI
Nomenclatura	XIV
Introducción	1
Objetivos de la tesis	3
Contribuciones de la tesis	4
Organización del documento	5
1. Robots móviles de servicio: antecedentes y evolución	7
1.1. Conclusiones	10
2. Estudio del escenario de robots móviles e interacción humano-robot	11
2.1. Plataformas robóticas	11
2.1.1. Tbot	11
2.1.2. UVerto	13
2.2. Cinemática del robot	13
2.3. Control visual	15
2.4. Interacción Humano-Robot	16
2.5. Proceso inteligente adaptativo para el seguimiento de trayectorias	21
2.6. Arquitectura de software	21
2.6.1. Primera capa: dispositivos	22
2.6.2. Segunda capa: percepción y control	22
2.6.3. Tercera capa: habilidades	23
2.6.4. Cuarta capa: actividades	26
2.6.5. Quinta capa: funcionamiento	26
2.6.6. Sexta capa: supervisor	26
2.7. Sistema Operativo de Robots (ROS)	27
2.8. Entorno	28
2.9. Conclusiones	29
3. Desarrollo de algoritmos inteligentes implementados en robots móviles	31
3.1. Algoritmos de visión para la detección y localización del usuario	31
3.1.1. Detección de rostros	32
3.1.2. Clasificación del usuario	34

3.1.3. Segmentación y localización del usuario	34
3.1.4. Reconocimiento de rostro	35
3.1.5. Implementación	35
3.1.6. Estimación de la velocidad del usuario	36
3.2. Navegación reactiva y pasiva	37
3.2.1. Odometría, modelo de muestra del movimiento	37
3.2.2. Modelo de telémetro	37
3.2.3. Modelo de probabilidad del sensor de profundidad	38
3.2.4. Localización de Monte Carlo aumentada	38
3.3. Metodología para la interacción humano-robot	38
3.3.1. Navegación con HRI	38
3.3.2. Plan de búsqueda de usuario	39
3.3.3. Apariencia y mecanismo para la interacción	40
3.4. Control visual difuso	41
3.4.1. Control difuso	41
3.4.2. Control PI+D difuso; control de velocidad lineal	42
3.4.3. Control difuso tipo sugeno para seguimiento de trayectoria	46
3.5. Supervisor	47
3.6. Conclusiones	47
4. Experimentación y resultados	49
4.1. Resultados del algoritmo de detección y localización de usuario	49
4.2. Estudio sobre la marcha humana	52
4.2.1. Estimación de la velocidad del usuario	55
4.2.2. Algoritmo del supervisor	55
4.2.3. Resultados de la navegación del robot guía	57
4.3. Pruebas con UVerto	61
4.3.1. Conclusiones	63
Conclusiones	65
Referencias	67
A. Características de los robots guía (1993-2018)	83
B. Pseudocódigos	99

Índice de figuras

1.1. Línea cronológica de los robots guía (1993-2018)	8
2.1. iRobot Create, componentes	12
2.2. Robot Tbot	12
2.3. Sensor Kinect	13
2.4. Robot UVerto	14
2.5. Robot móvil de tipo diferencial	14
2.6. Modelo de un robot con una cámara montada en el centro de este	16
2.7. Familiaridad con los robots	17
2.8. Interacción humano-robot centrada en robot y usuario	17
2.9. Zonas de contacto espacial de una persona	20
2.10. Lazo cerrado de control para el accionamiento del robot	21
2.11. Arquitectura de software propuesta	23
2.12. Marco referencial del mundo	24
2.13. Desplazamiento del centro de gravedad durante la marcha	24
2.14. Navegación de robots móviles en entornos conocidos	25
2.15. Actividades que realiza el robot	26
2.16. Distribución de ROS	27
2.17. A) Mapa del entorno de Tbot. B) Mapa del entorno de UVerto	28
2.18. Mapa del entorno obtenido por láser A) de Tbot. B) de UVerto	29
3.1. Imagen integral	32
3.2. Filtros Haar	33
3.3. Entorno de trabajo, mapa	36
3.4. Seguidor-Líder a) Robot-Humano, b) Humano-Robot	39
3.5. Kinect superior y altura del robot	40
3.6. Aplicación de selección en la tablet	41
3.7. Esquema de control para las velocidades lineal y angular	42
3.8. Esquema de control PI+D difuso	43
3.9. Regiones de combinación para el controlador PI	43
3.10. Funciones de pertenencia para las entradas y salida del control PI	44
3.11. Regiones de combinación para el controlador D	45
3.12. Esquema de control para la velocidad angular	46
3.13. Funciones de pertenencia de las variables de entrada θ y v	46
3.14. Supervisor	48
4.1. Algoritmo de localización de usuario	49

4.2. Colores que mejor distingue el algoritmo de visión de clasificación HSV	50
4.3. Detección de rostros distintos tonos de piel y posiciones.	51
4.4. Clasificación por color.	51
4.5. Ubicación del usuario.	51
4.6. Localización de la persona en un espacio de 3D.	52
4.7. Ubicación del usuario en existencia de más personas.	52
4.8. Interpolación de Lagrange y splines cúbicos.	53
4.9. Comparación de aproximación de la trayectoria con 12 y 6 datos de posición. . .	55
4.10. A. Velocidad con ec. cúbica de Bézier. B. Trayectoria realizada por el usuario. .	56
4.11. Procesos del robot.	56
4.12. Navegación reactiva de un robot móvil en un mapa.	58
4.13. Posición XY de las trayectorias seguidas a lo largo del trayecto.	59
4.14. Distancia entre el usuario y el robot comparada con la velocidad lineal del robot. .	59
4.15. Entorno real del experimento.	60
4.16. A. Seguimiento de velocidad Tbot. B. Trayectorias realizadas.	60
4.17. Seguimiento de velocidad del usuario en la plataforma móvil.	60
4.18. Trayectoria deseada vs trayectoria realizada.	61
4.19. Entorno real.	61
4.20. Seguimiento de trayectoria en plataforma UVerto.	62
4.21. Seguimiento de usuario en plataforma UVerto.	62
4.22. A. Seguimiento de velocidad en UVerto. B. Trayectorias realizadas.	62
4.23. Distancia de separación entre el robot UVerto y el usuario.	63
4.24. A. V lineal para trayectoria con obstáculos en UVerto. B. Trayectorias realizadas. .	63

Índice de tablas

3.1. Reglas del controlador Sugeno para w .	47
4.1. Posiciones con separación de 1 m .	54
4.2. Posiciones con separación de 0.5 m .	54
A.1. Historia de los robots guía (1993-2018).	84

Nomenclatura

- AMCL* Localización adaptativa de Monte Carlo (por sus siglas en inglés Adaptive Monte Carlo Localization)
- ARIA* Interfaz avanzada de robot para aplicaciones (por sus siglas en inglés Advance Robot Interface for Applications)
- BCM* Método de curvatura de rayo (por sus siglas en inglés Beam Curvature Method)
- CA* Autómatas celulares
- CC* Corriente Continua
- CdG* Centro de Gravedad
- CIIA* Centro de Investigación en Inteligencia Artificial
- CMU* Universidad Central de Michigan
- CP* Planificador de colaboración (por sus siglas en inglés collaboration Planner)
- CTMap* Mapa de Transición Condicional
- DWA* Algoritmo de ventana dinámica (por sus siglas en inglés Dynamic Window Algorithm)
- EIF* Filtro de información extendida (por sus siglas en inglés Extended Information Filter)
- FROG* Diversión guía robótica al aire libre (por sus siglas en inglés Fun Robotic Outdoor Guide)
- gdl* Grados de libertad
- GUI* Interfaz gráfica de usuario (por sus siglas en inglés Graphical User Interface)
- HOG* Histograma de degradados orientados (por sus siglas en inglés Histogram of Oriented Gradients)
- HRI* Interacción humano-robot (por sus siglas en inglés Human-Robot Interaction)
- HSV* Valor, saturación, tono (por sus siglas en inglés Hue Saturation Value)
- IFRR* Federación internacional de robótica (por sus siglas en inglés International Federation of Robotics)

II Imagen Integral

IPA Instituto de ingeniería de producción y automatización (por sus siglas en alemán Institut für Produktionstechnik und Automatisierung)

IR Infra Rojo

LRF Localizador láser por sus siglas en inglés Laser Range Finder

MCL Localización de Monte Carlo (por sus siglas en inglés Monte Carlo Localization)

MDP Proceso de decisión de Markov (por sus siglas en inglés Markov Decision Process)

MOMDP Proceso de decisión de Markov de observabilidad mixta (por sus siglas en inglés Mixed Observability Markov Decision Process)

NDTOM Transformaciones de distribuciones normales en mapas de ocupación por sus siglas en inglés (Normal Distributions Transform Occupancy Maps)

PCA Análisis de componentes principales (por sus siglas en inglés Principal Component Analysis)

PCL Biblioteca de nube de puntos

PERSES Sistema de servicio personal (por sus siglas en inglés PERSONAL SERVICE SYSTEM)

PID Proporcional, Integral y Derivativo

POMDP Proceso de decisión de Markov parcialmente observable (por sus siglas en inglés Partially Observable Markov Decision Process)

RFID Identificación por radiofrecuencia (por sus siglas en inglés Radio Frequency Identification)

RGB Rojo, verde, azul (por sus siglas en inglés Red Green Blue)

ROS Sistema operativo para robots (por sus siglas en inglés Robot Operating System)

SA Arquitectura de software (por sus siglas en inglés Software Architecture)

SLAM Localización y mapeo simultáneos (por sus siglas en inglés Simultaneous Localization And Mapping)

TDH Histograma por distancia de transferencia (por sus siglas en inglés Transfer Distance Histogram)

UV Universidad Veracruzana

VSLAM Localización y mapeo visuales simultáneos por sus siglas en inglés (Visual Simultaneous Localization And Mapping)

Introducción

Entre las aplicaciones de los robots de servicio en el mundo, el uso de los robots guías en museos, exhibiciones, universidades, etc. es una tarea muy interesante ya que involucra avanzar en varios campos del conocimiento tales como la navegación autónoma en ambientes dinámicos, interacción humano robot (HRI), modelado de ambientes interiores con localización simultánea y construcción de mapas, etc., todos esto interactuando en tiempo real.

Una plataforma robótica móvil genérica debe tener un conjunto muy amplio de habilidades que incluyen entre otras percepción avanzada, planeación de movimiento, y planeación de tareas. Estas habilidades son insuficientes para un robot guía, el cual se desenvuelve en ambientes dinámicos, por lo que necesitan ser complementadas con comportamientos conscientes del ser humano. Por lo tanto los robots guía deben de comportarse de tal forma que los humanos perciban al robot como un humano más. Los métodos tradicionales para el movimiento de un robot guía que navega cerca de las personas podrían ser rechazados por los humanos, ya que no seguiría las normas sociales. Así que cuando un robot guía tiene un comportamiento social no esperado la interacción humano robot falla. Además, las habilidades de navegación del robot son limitadas a un simple seguimiento de trayectorias con acciones reactivas para evitar colisiones con humanos y obstáculos. Así mismo, ciertas acciones particulares donde el usuario no sigue al robot no son consideradas en los métodos tradicionales, siendo esto un aspecto de estudio en este trabajo de tesis.

Para abordar estos problemas, y proponer alternativas de mejora, en este trabajo tesis se desarrolla una metodología que permite al robot interactuar de forma natural con los humanos y su ambiente mientras el robot navega, guía al usuario en ambientes sociales. En este sentido, el movimiento del robot está inspirado por el movimiento del usuario y convenciones sociales tal como espacio personal.

Motivación y planteamiento del problema

El problema de control de movimiento para el seguimiento de objetivos con robots móviles se puede dividir en dos grandes grupos: a) el robot sigue a alguien o algo o b) el robot desea que lo sigan.

En el primer caso las leyes de control se establecen para que el robot pueda seguir a un objeto o persona dentro de un ambiente determinado, este problema es generalmente conocido como “sigue al líder” y en función del objetivo o metas, el control se puede establecer en posición o velocidad.

Las leyes de control generalmente son retro-alimentadas mediante algún medio de percep-

ción, generalmente mediante el uso de sistemas de visión computacional; en donde se pueden usar objetos binarios grandes o BLOBS (por sus siglas en inglés Binary Large Objects) de color, marcas o etiquetas como códigos QR (código de respuesta rápida), o mediante métodos más sofisticados de segmentación del objetivo. En algunos trabajos, sobre todo en seguimiento de robot a robot, estos se pueden comunicar mediante señales visuales o enviando la posición en el mundo de cada uno de los robots, ejemplo de ello son los prototipos de autos autónomos que se siguen en una autopista.

Por otro lado el problema del robot como guía, es diferente, debido a que, el robot tiene un objetivo; llegar a un sitio en particular, lo cual se puede establecer mediante el control en posición o en velocidad pero adaptado a un perfil específico de velocidad para poder llegar a su objetivo

Un tercer caso, es el comúnmente llamado “persecución - evasión”, en donde se considera un robot tratando de seguir a un objetivo, que puede ser otro robot, y este a su vez trata de no ser alcanzado.

En el caso de un robot guía de turistas, además de considerar que el robot debe realizar una trayectoria específica determinada para llegar a su objetivo y considerando que esta puede ser deformada o modificada por posibles obstáculos u obstrucciones, se debe tener en cuenta otro objetivo de control, el cual es el asegurar que el humano u otro robot lo vienen siguiendo.

Actualmente existen múltiples robots guía, siendo presentada una breve descripción de cada uno de ellos en el Capítulo 1, que no consideran dentro de sus objetivos de control la detección del usuario, por ejemplo en [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28], solo se generan trayectorias planificadas y es en tiempos bien establecidos en donde se intercambia información, generalmente la navegación se realiza por láser. La detección y seguimiento visual del usuario, generalmente se ha usado solo para la parte de interacción estática, es decir cuando el robot esta parado o tiene movimientos limitados. Esto se puede hacer mediante cámaras embarcadas como webcams, cámaras omnidireccionales y dispositivos RGB-D como el Kinect. En [29] se realiza una segmentación con la información del color de piel, el contorno de la cabeza y hombro asegurando con esto la interacción con una persona y no con algunos falsos negativos de los detectores de rostros. Solo el color de piel para ubicar al usuario dentro del campo de visión es utilizado en [10]. Otros robots guía realizan un control del movimiento del mismo con una distancia constante entre el robot y el usuario. En [30, 29, 31, 32] realizan un reconocimiento de rostros y voz y se mantiene a una distancia constante del usuario. En [33] se utiliza la información de vestimenta y el esqueleto de la persona para mantener una distancia constante entre el usuario y el robot. Otros consideran la distancia entre el punto meta y el robot junto con la distancia del usuario y el robot con la información del rostro. Estos solo tienen dos velocidades, si la distancia esta dentro de un rango de 50 cm tiene una velocidad constante, si es mayor o igual a 1.2 m la velocidad se reduce a $\frac{1}{2}$ [34, 35, 36]. En [37, 38] utilizan datos RGB-D para determinar la distancia entre el usuario y el robot y mantenerla constante durante el trayecto.

Algunos otros trabajos utilizan cámaras externas (vigilancia) o sistemas láser distribuidos en el ambiente para determinar la distancia entre el usuario y el robot [27, 19, 20]. En [39, 40] se obtienen características faciales para detectar expresiones junto con el cálculo de los ángulos de Euler para detectar la rotación del rostro determinando el interés del usuario, solo para interacción. En [41] los usuarios son detectados a través de una diferenciación de mapas además de un sensor de rango y [42, 43] detecta al usuario a través de una segmentación por nube de puntos con 17 características del cuerpo del usuario junto con la información de 2 sensores láser.

El filtro de Kalman predice el estado de la persona con un modelo de velocidad constante. En [44] utilizan un filtro de Kalman para determinar la velocidad del usuario pero solo la usan para controlar el movimiento vertical y horizontal de la cámara por medio de un “pan-tilt” y tener al usuario siempre en el campo de visión.

Solo el robot guía de [45] trata de ajustar su velocidad a la del usuario por medio de un controlador de movimiento adaptativo del robot que se basa en el método de campo de potencial artificial [46]. En este método se generan fuerzas atractivas y de repulsión, donde los campos de atracción provienen del usuario y del punto meta o submetas; y los campos repulsores provienen de los obstáculos y del mismo usuario. Para decidir si el usuario sera una fuerza de repulsión o de atracción se define un umbral de distancia, el cual cambia el modo de acción del robot para mantener su relación de distancia social con los usuarios si estos se desvían. El cálculo de trayectorias se realiza mediante el algoritmo de Dijkstra [47], para calcular la ruta más corta. Sobre esta ruta, se definen sub metas como referencias para los campos de atracción. Sin embargo, la referencia para el control de velocidad del robot esta determinada por medio de los campos de fuerzas repulsivas y no por la velocidad del usuario. Incluso, la estimación de la velocidad del usuario solo se utiliza para verificar su controlador. Lo anterior ocasiona que el error entre velocidades (usuario - robot) no este regulado, por lo que las variaciones en la distancia social son grandes (aprox 1.4 m), lo cual excede la velocidad cómoda para los usuarios de 1 m/s [48] y presenta cambios bruscos de velocidad que oscilan entre 0 y 1 m/s. Además, la técnica de navegación esta centrada en calcular la ruta más corta, lo cual no resulta ser lo más conveniente [12, 49].

Hasta el momento no se ha considerado introducir la velocidad del usuario dentro de la ley de control de movimiento del robot guía. En este trabajo de investigación, la velocidad del usuario de determina en tiempo en real (al momento de iniciar la aplicación) y utilizarla como retroalimentación en un controlador inteligente, diseñado específicamente para asegurar que el usuario lo viene siguiendo.

Con base en la información anterior, se presenta la siguiente hipótesis.

Hipótesis de la tesis

La incorporación del usuario en un proceso inteligente adaptativo del seguimiento de trayectorias de un robot guía mejora la interacción humano-robot.

Objetivo general

El principal objetivo de este trabajo de investigación es diseñar, implementar y analizar una metodología que permite la interacción de varios procesos para mejorar la interacción humano robot en sistemas móviles utilizando la información del usuario. En el diseño propuesto centrado en el usuario para la interacción humano robot, el objetivo es introducir la información de velocidad del usuario dentro del lazo de control de movimiento del robot.

Objetivos particulares

Para alcanzar el objetivo planteado se presentan los objetivos específicos del trabajo de tesis:

- Planear trayectorias locales y globales que le permitan navegar de manera autónoma.
- Evitar obstáculos y recalcular trayectorias a lo largo del recorrido para llegar a su objetivo.
- Reconocer personas y distinguir a un solo usuario por medio de algoritmos de visión.
- Mantener una interacción con el usuario durante el trayecto.
- Ajustar la velocidad del robot a la del usuario.
- Realizar una estrategia de búsqueda, en caso de pérdida del usuario.

Contribuciones de la tesis

Esta tesis provee cuatro contribuciones principales.

- El diseño de un proceso inteligente adaptativo para incluir al humano en el lazo de control de seguimiento de trayectorias de robots móviles.
- La implementación en un sistema distribuido (no lineal) de comunicación hardware-software (ROS).
- La adaptación de algoritmos de reconocimiento de personas en datos RGB-D.
- La evaluación de todos los algoritmos propuestos con usuarios en ambientes reales.
- Mejorar la interacción con los robots (HRI).

Publicaciones

- Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández y Antonio Marín Hernández. “A Fuzzy Speed Controller for a Guide Robot Using an HRI Approach”, en IEEE Latin America Transactions (aceptado 2018).
- Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández y Antonio Marín Hernández. “A Survey on Navigation Strategies for Tour Guide Robots”, enviado a la revista Robotics and Autonomous Systems - Elsevier (en revisión).
- Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández y Antonio Marín Hernández. “Análisis del Primer Contacto entre un Robot Asistente Didáctico y Niños de Educación Básica”, en Congreso Nacional de Control Automático 2017 (AMCA 2017), Monterrey, México, Oct 2017.
- Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández y Antonio Marín Hernández. “Diseño e implementación de un robot guía basado en la interacción humano-robot”, en Congreso Nacional de Control Automático (AMCA 2016), Querétaro, México, Sep 2016.
- K. L. Luna-Gallegos, E. R. Palacios-Hernandez, S. Hernandez-Mendez y A. Marin-Hernandez. “A proposed software architecture for controlling a service robot”, en IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), páginas 1–6, Guerrero, México, Nov 2015.

- Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández y Antonio Marín Hernández. “Una propuesta de arquitectura para el control de un robot guía”, en XI Semana Nacional de Ingeniería Electrónica (SENIE 2015), no. 112, páginas 1155–1171, San Luis Potosí, México, May 2015.
- Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández, and Antonio Marín Hernández. “Detección y seguimiento de personas con análisis de color en datos RGB-D”, en XVI Congreso Latinoamericano de Control Automático (CLCA 2014), páginas 630–635, Quintana Roo, México, Oct 2014.

Tesis dirigidas (co-asesoría)

- Argel Moreno Hernández, Diseño y Construcción de un Robot Móvil para Tareas de Enseñanza e Investigación. Para obtener el título de Ingeniero Electrónico, Facultad de Ciencias, UASLP. 2018.

Organización del documento

Este documento de tesis está organizado de la siguiente manera:

Capítulo I. Se describen las principales características de los robots guía en hardware, navegación e interacción desarrolladas a través del tiempo.

Capítulo II. En este Capítulo se explica el desarrollo de las plataformas robóticas donde se implementó la metodología propuesta, así como su análisis matemático. Además, se describen los diferentes estudios de interacción que se consideraron para el comportamiento del robot.

Capítulo III. En este Capítulo se describen los algoritmos de visión utilizados para el reconocimiento y localización de las personas, así como para la navegación. Se presenta el diseño de los controladores propuestos y la implementación de estos para el movimiento del robot. También se establece en este Capítulo, la arquitectura de software y el supervisor.

Capítulo IV. Aquí se detallan las características de los experimentos realizados y los resultados obtenidos en cada etapa de la metodología.

Finalmente, se describen las conclusiones y trabajo futuro del trabajo de tesis.

Capítulo 1

Robots móviles de servicio: antecedentes y evolución

Los robots son actualmente objetos de gran interés. Estos pueden tener diferentes modalidades para establecer una mejor interacción con los humanos. Por lo tanto no es sorpresa que una de las primeras aplicaciones de los robots de servicio haya sido la guía de usuarios en laboratorios, museos y exhibiciones. En este Capítulo se describen los robots guía que existen en la actualidad, estableciendo sus diferencias en hardware, tipo de navegación y las técnicas de HRI. En la Figura 1.1 se muestra una línea del tiempo de los robots guía desarrollados, desde el año 1993 hasta el año 2018.

Polly [1], fue el primer robot guía. Se basaba en una visión de bajo costo para el reconocimiento del entorno. Además, busca usuarios en los corredores, se presenta y ofrece un recorrido por los diferentes laboratorios. Las personas aceptan que el robot los guíe agitando los pies, porque era lo único que el robot podía detectar. Después en el año 1998 y 1999 aparecen los robots Rhino [2, 3] y Minerva [4, 5] los cuales son considerados como pioneros en la aplicación de robot guía. Ambos ofrecían una guía turística interactiva en museos y contaban con 4 sensores de última generación (láser, sonar, infrarrojo y táctil), también tenían una cámara estéreo, una base móvil RWI B2. Para estos robots se desarrollaron algunos algoritmos para navegar entre multitudes y evadir obstáculos, básicamente, la navegación se divide en dos módulos: la percepción estima la ubicación del robot y los obstáculos cercanos. El módulo de control realiza la evasión de obstáculos mediante la planificación de rutas en tiempo real. Se utilizó una versión modificada del algoritmo de localización de Markov junto con un filtro de entropía [50] para aproximar la posición del robot. Además, se proporcionó un mapa métrico del museo al robot y debido a que el entorno cambia constantemente, se utilizó un algoritmo de cuadrícula de ocupación probabilística [51] para modificar el mapa inicial en tiempo real para evitar obstáculos. Cada vez que el robot llega a su objetivo, este mapa se restablece al original. El movimiento del robot se realiza controlando la dirección y la velocidad de este, en función de la información de los sensores de entrada y la ubicación del objetivo. Rhino usa una extensión del algoritmo de ventana dinámica (DWA) [52] que le permite navegar sin problemas alrededor de los obstáculos hacia su objetivo. Debido a la presencia de obstáculos invisibles, el robot debe consultar el mapa proporcionado antes de comenzar, expandiendo el enfoque DWA para estimar la posición del robot dentro del mapa y generar mediciones virtuales de proximidad

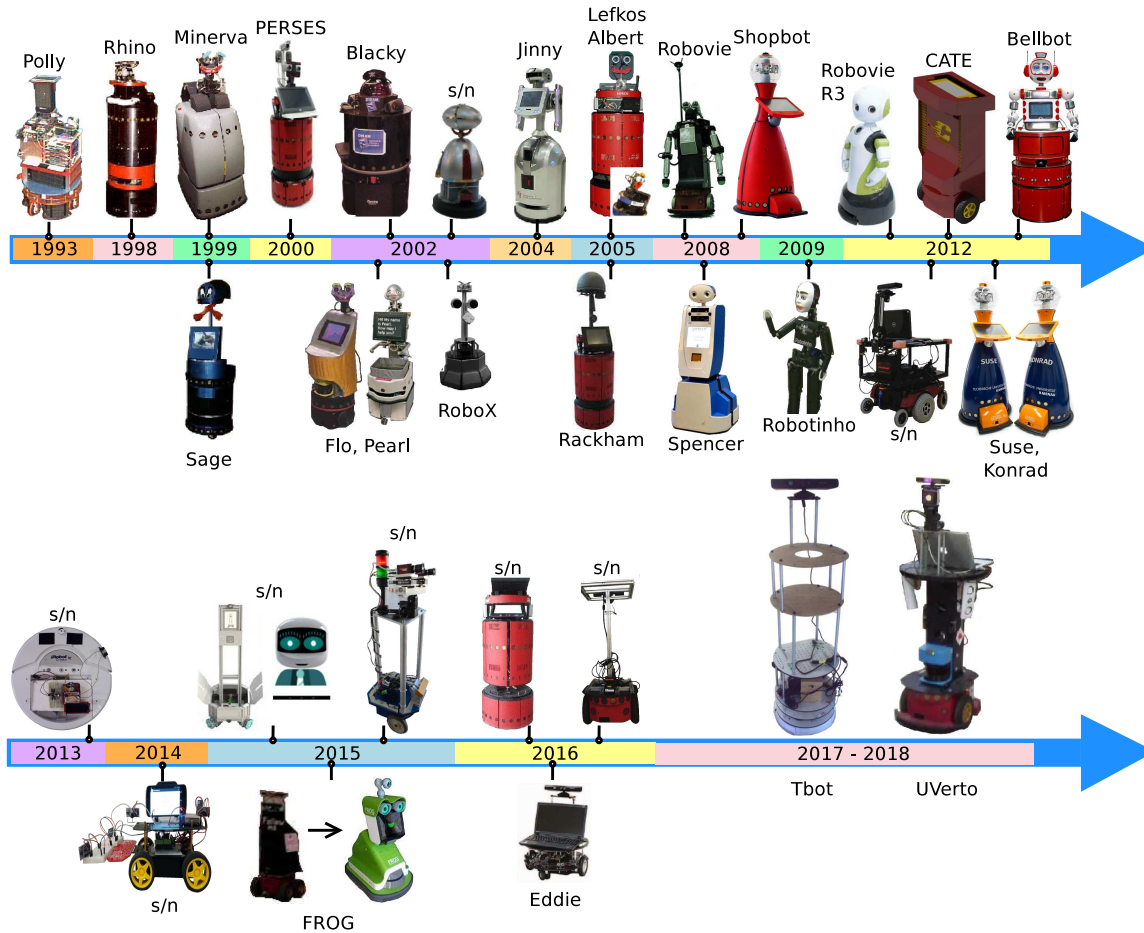


Figura 1.1: Línea cronológica de los robots guía (1993-2018).

las cuales se integran con las mediciones reales obtenidas de los sensores del robot. Un rostro se adicionó a Minerva, este le permite hacer expresiones faciales y diferentes tonos de voz para interactuar con los visitantes. Los estados de ánimo variaron de felices a enojados, dependiendo de la persistencia de las personas que bloqueaban su camino. Los robots posteriores a estos utilizaron otras técnicas de navegación, algunas para evitar el costo computacional de la navegación probabilística [6, 21]. También se adicionaron más y distintos sensores para mejorar el posicionamiento del robot, ejemplo de esto son sensores ubicados en el entorno como: RFID, bandas magnéticas, cámaras, etc. Una de las técnicas que se destaca para la localización es la Localización Monte Carlo (MCL) implementada por primera vez en el robot ‘PERSES’ (PERSONAL SERVICE SYSTEM) [29] para la tarea de robot guía. Respecto al movimiento del robot, se han utilizado controladores de bajo nivel [7, 6], PID [28], difusos. También se realizaron mejoras en la interacción y comunicación con las personas. Algunos robots incluyeron pantallas, síntesis de voz, a otros se les dotó de personalidad a través de posturas, gestos y expresiones [4, 53]. Para detectar a las personas, se ha incluido el reconocimiento de las mismas mediante detección de voz y algoritmos de visión [29, 31]. Y dependiendo de sus tareas se adicionaron partes extras a la estructura del robot, por ejemplo ‘Pearl’ [41] cuenta con dos manijas sólidas para brindar soporte a las personas mayores residentes de un asilo.

Algunos trabajos utilizan la proxémica como norma de interacción [16, 34, 45]. Otro método utilizado para la interacción con las persona es proporcionando acceso individual a las exhibicio-

nes de los museos a través de Internet [14, 2]. Los entornos más comunes en los que actualmente se emplea a los robots guía son: universidades [33, 44] y pasillos de laboratorios [1, 21], museos [4, 6], comercios [29, 30], y exposiciones [10, 14]. Algunos de estos trabajos implementaron arquitecturas de software para la manipulación de las actividades del robot [12, 13, 15].

La investigación y el desarrollo de robots guías móviles creció día a día, algunos empleaban plataformas ya desarrolladas las cuales con base en estudios anteriores, mejoraban su mecanismo, apariencia, control y comportamiento del robot. También se incrementaron los estudios enfocados al desempeño de los robots en un entorno social, implementando diferentes métodos de interacción entre humanos y robots.

En la actualidad existen guías robóticas muy sofisticadas que ya se encuentran operando en entornos sociales. Otros trabajos de investigación, comenzaron a realizar robots guía de bajo costo adaptando técnicas de navegación desarrolladas anteriormente con nuevos dispositivos electrónicos. En el Apéndice A, se muestran las principales características de estos robots guía móviles desarrollados desde 1993 hasta 2018. Se define su espacio de trabajo (entorno), sus componentes mecánicos y electrónicos, sus algoritmos utilizados para la navegación y percepción del entorno, además del tipo de interacción que se establece con las personas.

Actualmente, el trabajo de investigación más parecido al presentado en este documento, es el desarrollado en [45] el cual, como ya se mencionó, la velocidad del usuario no se retro-alimenta en el lazo de control de movimiento. Sino que se consideran los campos de fuerzas atractivas y repulsivas.

En este trabajo de tesis, se desarrollo el robot Tbot. El cual opera en los pasillos del edificio 2 de la Facultad de Ciencias donde se encuentran las oficinas de algunos docentes de la institución. Consiste en una base móvil de tipo diferencial *iRobot*, la cual cuenta con un giroscopio y dos codificadores para el cálculo de la odometría, la velocidad lineal máxima es de 0.5 m/s , la velocidad angular es de $180^\circ/\text{seg}$. Además, el robot tiene dos sensores Microsoft Kinect para la percepción del entorno, estos sensores consisten en una cámara RGB y un emisor/receptor de luz infrarroja para la construcción del entorno en tres dimensiones. La altura del robot es de 120 cm y realiza la comunicación entre las diferentes tareas mediante el Sistema Operativo de Robots (ROS) el cual permite el paso de información entre un proceso y otro de manera paralela. La actividad de guía de personas inicia con la detección y localización del usuario en un espacio tridimensional. Con esta información se calcula la velocidad del usuario con base en la distancia que recorre en un intervalo de tiempo. El robot cuenta con la información del mapa de su entorno previamente reconocido, el punto inicial y la ubicación de los lugares a los que el usuario puede dirigirse. Con esta información se realiza el cálculo de trayectorias, donde se genera una trayectoria global del punto inicial al punto final realizando un cuadrículado en el mapa y definiendo las celdas libres por las que puede pasar el robot. Además, es necesario conocer la localización del robot dentro del mapa la cual está basada en algoritmos probabilísticos (odometría, telemetría, localización de Monte Carlo aumentada). Con la posición actual del robot y la información del entorno es posible calcular trayectorias locales para la evasión de obstáculos, este cálculo consiste en conocer la velocidad lineal y angular del robot, así como la proximidad de obstáculos, del punto meta y de la trayectoria global. Con esta información, se realiza una simulación de la posición próxima del robot al cabo de un corto periodo de tiempo, para predecir si el robot colisionará. El movimiento del robot depende de la velocidad lineal y la velocidad angular, estas variables se ajusta por medio de un controlador difuso con un lazo interno y otro externo. El lazo interno consiste en un controlador PI+D difuso para adaptar la velocidad lineal del robot a la velocidad estimada del usuario. El

lazo externo tiene como entrada la velocidad lineal del robot obtenida en el lazo interno y el error en orientación, el cual se calcula a partir de la trayectoria deseada y la posición actual del robot. Por medio de normas sociales como la proxémica, una pantalla y con un sintetizador de voz se entabla la comunicación con el usuario. La adaptación de velocidad del robot busca brindar un mejor servicio al usuario, ya que incrementará las probabilidades de que el usuario lo siga y ambos lleguen a su objetivo. El control de las velocidades del robot está basado en normas reglas sociales de interacción entre humanos y robots [54, 55, 56, 57, 58].

El entorno en el que el robot trabaja implica el tipo de personas que van a interactuar con él, por ejemplo: niños, adultos mayores, personas con conocimientos de ingeniería. Así como las complicaciones que puedan surgir en la navegación, i.e., lugares con mucha gente, pasillos estrechos o muy anchos. Además de las tareas que realizará el robot por ejemplo: proporcionar información o solicitarla, modificar su movimiento y acción, entre otras. Por ejemplo, el área de los laboratorios de una institución es un ambiente controlado debido a que las personas que interactúan con el robot tienen en su mayoría conocimiento sobre este u otro tipo de tecnología. Los experimentos realizados en entornos reales no controlados donde pueden encontrarse personas inexpertas como hoteles, hospitales y museos, a menudo no funcionan como se esperan [25]. Debido a esto, es importante que los investigadores realicen experimentos en entornos reales para verificar el verdadero funcionamiento del robot.

Las técnicas de navegación son muy variadas, se determinan dependiendo de las capacidades del robot. Algunas técnicas necesitan adaptar el entorno para su funcionamiento de acuerdo con los dispositivos y la adquisición de datos (etiquetas de visión, RFID, nodos de anclaje, etiquetas IR). Una de las técnicas más utilizadas es con la utilización de mapas cuadrículados (también llamadas celdas de ocupación) y por detección de objetos por medio de sensores. Anteriormente se utilizaban controladores de bajo nivel, pero con el paso del tiempo, fue necesario implementar controladores de alto nivel para cumplir con las exigencias de tareas más complicadas y precisas. En este trabajo de tesis se diseñó e implementó un supervisor de alto nivel.

1.1. Conclusiones

En este Capítulo se presentó una revisión de las técnicas de navegación e interacción utilizadas por los robots guía móviles. Los diferentes artículos revisados demuestran que es posible abordar la tarea de guía de personas con plataformas robóticas desde diferentes enfoques y características.

Debido a que la presencia de robots guía en lugares comunes para las personas va en aumento. Es importante estudiar y analizar las diferentes técnicas utilizadas para la interacción del robot con el humano, a fin de conocer aquellas técnicas que ofrecen una mejor integración de los robots en un entorno social. Es decir, la relación entre ellos debe ser amigable, intuitiva y segura. El comportamiento del robot se rige por normas sociales aceptadas entre humanos y robots, la elección de este comportamiento está sujeta a la información del entorno, la autoinformación y las reglas establecidas para la interacción. Por esta razón, es importante conocer los diferentes dispositivos de adquisición de datos del entorno y la planificación del movimiento.

Con la introducción de robots en la vida cotidiana de las personas, se definirán reglas y características sociales más aceptables para el comportamiento del robot.

Capítulo 2

Estudio del escenario de robots móviles e interacción humano-robot

En la actualidad, los robots guía se utilizan en oficinas, museos, hospitales y áreas de entretenimiento. Las tareas que desempeñan estos robots no solo incluyen la guía de personas también realizan otras funciones como facilitar información, transportar objetos, evacuar edificios, asistir a adultos mayores, entre otras cosas.

Para el desarrollo de este robot guía, se consideran diversas disciplinas. El robot debe navegar de manera autónoma evitando obstáculos y llegando a su destino. También debe ser capaz de reconocer personas a las que brindará sus servicios y distinguir a un solo usuario por medio de clasificadores. Para esto es deseable implementar módulos que permitan al robot obtener la información necesaria para poder relacionarse con el usuario y con el entorno. Esto se logra combinando tres áreas; HRI, sistemas de control, control visual y procesamiento de imágenes.

En este Capítulo se describe el diseño e implementación de los distintos algoritmos necesarios para los robots, así como el procedimiento y las necesidades para lograr la ejecución de la aplicación de guía de personas.

2.1. Plataformas robóticas

En este trabajo de investigación se utilizaron dos robots móviles para la aplicación de robot guía, el robot prototipo desarrollado en este trabajo de tesis llamado “Tbot” y un robot de la Universidad Veracruzana (UV) llamado “UVerto”. En esta sección se describen ambas plataformas robóticas.

2.1.1. Tbot

El robot Tbot utiliza como base el robot de la compañía Clearpath Robotics, el Turtlebot. Como plataforma móvil tiene al robot iRobot Create que se muestra en la Figura 2.1, esta es alimentada por una batería de $3000mAh$, cuenta con sensores de desnivel, un receptor IR omnidireccional, dos ruedas con motores, dos ruedas de soporte, un giroscopio y se comunica con una computadora a través de un puerto serial. El robot Turtlebot tiene además un sensor

de 3D Microsoft Kinect orientado hacia delante para la navegación, una laptop con procesador Dual Core y la estructura propia del robot.

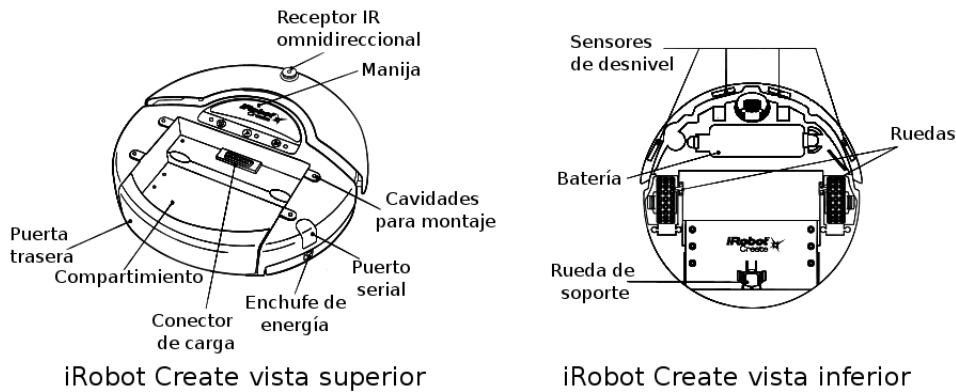


Figura 2.1: iRobot Create, componentes.

Al robot Turtlebot utilizado en este trabajo, se le añadieron 2 pisos de madera para posicionar otro sensor Kinect, este orientado hacia atrás para la interacción con los usuarios. También se adicionó una batería para alimentar a los dos sensores Kinect y una laptop Dell Inspiron 14R con procesador i7. El diseño final del robot Tbot se muestra en la Figura 2.2. Además, para la HRI se utilizó unas bocinas y un dispositivo móvil Acer Tablet con Android el cual invitará al usuario a utilizar sus servicios. Brindará información y pondrá a disposición del usuario una visita guiada al lugar que desee dirigirse, ofreciendo durante el recorrido un trato agradable a la persona para que esta lo siga. La altura del robot es de 120 cm con un diámetro de 35 cm, la velocidad de desplazamiento máxima es de 0.5 m/s, mientras que su velocidad angular máxima es de 150°/s.

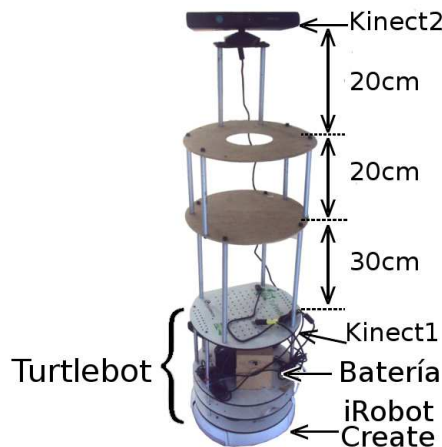


Figura 2.2: Robot Tbot.

El sensor Microsoft Kinect consiste en una cámara RGB, un emisor de luz infrarroja (IR) y un sensor IR de profundidad, un arreglo múltiple de micrófonos y un acelerómetro en un motor de inclinación (tilt), como se muestra en la Figura 2.3.

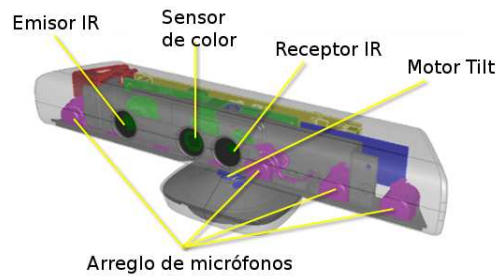


Figura 2.3: Sensor Kinect [59].

La secuencia de video RGB usa una resolución VGA de 8 bits (640x480 píxeles) con un filtro Bayer de color. La transmisión de video de detección de profundidad monocromática está en resolución VGA (640x480 píxeles) con una resolución de profundidad de 11 bits, que proporciona 2,048 niveles de sensibilidad. El sensor Kinect tiene un límite de autonomía práctico de 1.2 – 3.5 *m* de distancia, aunque el sensor puede mantener el seguimiento a través de un rango extendido de aproximadamente 0.7 – 6 *m*.

El sensor tiene un campo de visión angular de 57° horizontalmente y 43° verticalmente. El campo horizontal del sensor Kinect a la distancia mínima de visualización es de 87 *cm*, y el campo vertical es de 63 *cm*, lo que da como resultado una resolución de poco más de 1.3 *mm* por píxel.

2.1.2. UVerto

El robot UVerto se muestra en la Figura 2.4, cuenta con una base robótica llamada Pioneer 3-DX cuya velocidad de desplazamiento máxima es 1.2 *m/s* y su velocidad angular es de 300°/*s*. Este robot utiliza un entorno de software llamado interfaz avanzada de robot para aplicaciones ARIA (por sus siglas en inglés Advance Robot Interface for Applications), con el cual se pueden controlar y recibir datos de la plataforma. Cuenta con un sonar para localización y navegación, además se adicionó un láser de 2D para navegación y mapeo (ARNL), un sensor Xtion PRO de Asus. El sensor Xtion PRO, al igual que el sensor Kinect, cuenta con una cámara RGB y un sensor de profundidad.

Ambos robots son soportados por el sistema operativo para robots ROS (por sus Robot Operating System), este software provee librerías y herramientas para ayudar a los desarrolladores de software para crear aplicaciones para el robot. Tales como acceso al Hardware, controlador de dispositivos, visualizadores, simuladores, paso de mensajes entre procesos, manejo de paquetes y más. ROS es de código libre y da soporte a gran variedad de robots e integra otros proyectos populares de código libre como Gazebo, OpenCV, Point Cloud Library (PCL) y MoveIt! [60]. En la Sección 2.7 se detalla este sistema.

2.2. Cinemática del robot

Las dos plataformas robóticas utilizan un modelo cinemático de tipo diferencial, el cual consiste de dos ruedas con motores y una o dos ruedas giratorias como soporte. En la Figura 2.5 se muestra este tipo de configuración.



Figura 2.4: Robot UVerto.

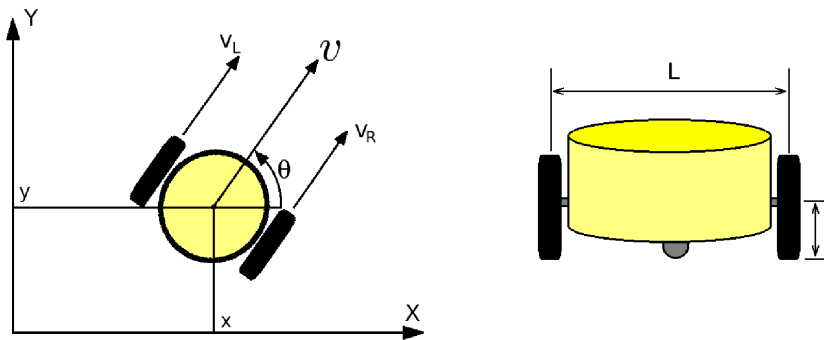


Figura 2.5: Robot móvil de tipo diferencial.

Un robot móvil con ruedas es un sistema no lineal, multivariable y variante en el tiempo. La configuración diferencial tiene limitaciones de movimiento no-holonómicas que resultan en un sistema subactuado [61, 62]. Se deben considerar las restricciones no-holonómicas del sistema, es decir, cada rueda gira hacia adelante y hacia atrás, pero no puede trasladarse a los lados.

Para conocer la posición del robot con respecto a una posición inicial se utiliza la odometría, para obtenerla, se realiza el análisis cinemático. En la Ecuación 2.1, v_L y v_R representan las velocidades tangenciales de cada rueda las cuales se obtienen a través de sensores propioceptivos (encoders).

$$v_L = w_L * r; \quad v_R = w_R * r \quad (2.1)$$

donde r es el radio de las ruedas, w_L y w_R son las velocidades angulares de las llantas.

Para que el robot avance en línea recta, la velocidad de sus ruedas debe ser igual. Debido a esto, se puede definir la velocidad lineal del robot como el promedio de las velocidades de las ruedas siendo proporcional al radio de estas. A partir de esta información se obtiene la velocidad angular (w) y lineal (v) del robot, Ecuaciones 2.2.

$$\begin{aligned} v &= \frac{v_R + v_L}{2} = \frac{(w_R + w_L) * r}{2} \\ w &= \frac{v_R - v_L}{L} = \frac{(w_R - w_L) * r}{L} \end{aligned} \quad (2.2)$$

donde L representa la distancia entre las ruedas y r es el radio de las mismas.

La cinemática del movimiento del robot se indica en la siguiente Ecuación 2.3 y describe el desplazamiento del robot en 2D.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (2.3)$$

donde \dot{x}, \dot{y} es la trayectoria que realiza el robot y $\dot{\theta}$ el cambio en la orientación, esta última es calculada con base en la información del giroscopio.

La odometría se calcula con sensores propioceptivos, estos solo miden las variables internas del robot, por lo que el error de cálculo en la posición se incrementa con el paso del tiempo al no existir una retroalimentación de la verdadera posición del robot. Para conocer el entorno, se utilizan sensores exteroceptivos los cuales miden variables externas, de esta manera se puede tener una mejor localización del robot dentro de su espacio de trabajo.

2.3. Control visual

Durante la tarea de navegación, el robot debe responder a tres preguntas: ¿dónde estoy?, ¿cómo se ve el mundo?, ¿cómo llegar a un lugar deseado?. Las dos primeras preguntas se relacionan con la percepción y la última es tratada por el sistema de control [61]. El control visual (visual servoing) representa una buena opción para el control de robots pues es capaz de realizar el proceso de dirigir un robot hacia un objetivo utilizando la retroalimentación visual en un lazo cerrado de control. Para que el controlador calcule el movimiento del robot necesita dos componentes principales: un modelo del entorno, ya que el entorno va a cambiar después de un comando de control; y una ley de control, como se determina el comando siguiente dada la medición de una imagen actual y un modelo.

Se han propuesto sistemas de control visual basadas en Jacobiano para 6 gdl [63]. En [64] se utiliza el esquema de control visual basado en imágenes que linealiza la dinámica de los carros no-holónicos transformándolos al plano de la imagen. Una estrategia para la planificación del movimiento visual en el espacio de la imagen se presenta en [65]. Se han propuesto algunos esquemas de control visual basado en posición por ejemplo, el controlador de conmutación de [66] utiliza una reconstrucción basada en homografía euclidiana para estimar la posición del robot.

Para diseñar la ley de control se deben tener la información del modelo del sistema. El modelo cinemático de muchas plataformas móviles con ruedas se puede caracterizar por un modelo diferencial de tracción que se muestra en la Figura 2.6.

La cámara se coloca en el eje central del robot, a una distancia l a lo largo del eje y_R . El movimiento del marco del robot $\{R\}$ con respecto al marco del entorno de trabajo $\{W\}$ puede expresarse usando el modelo $\dot{x} = -v \sin \phi$, $\dot{y} = v \cos \phi$ y $\dot{\phi} = w$ donde v y w representan la

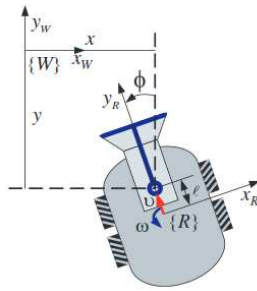


Figura 2.6: Modelo de un robot con una cámara montada en el centro de este [61].

translación y velocidad rotacional, respectivamente. El comportamiento de la cámara es descrita por el siguiente modelo (ecuación 2.4):

$$\begin{aligned}\dot{x} &= [g_1(x) \quad g_2(x)]u, \\ y &= h(x).\end{aligned}\tag{2.4}$$

Este sistema tiene el vector de estado $x = [x, y, \phi]^T$, el vector de entrada $u = [v, w]^T$ y los campos vectoriales de entrada $g_1(x) = [-\sin \phi, \cos \phi, 0]^T$ y $g_2(x) = [-l \cos \phi, -l \sin \phi, 1]^T$. La medición de la salida y es dada por una función no lineal adecuada $h(x)$ y se define a través de la información extraída de imágenes [61]. Conociendo el modelo del sistema se puede diseñar la ley de control basado en visión.

2.4. Interacción Humano-Robot

La interacción humano-robot es un campo multidisciplinario, algunas de estas disciplinas son inteligencia artificial, robótica, comprensión de lenguaje, ciencias sociales, psicología, entre otras. La HRI se dedica al entendimiento, diseño y evaluación de sistemas robóticos para usarlos por o con humanos [67].

Actualmente, para que el robot pueda interactuar de forma segura con humanos, se ha limitado el área de trabajo que comparten los robots y las personas, además se establecen normas de seguridad. Con la inteligencia artificial y el estudio de la HRI se pretende que los robots de manera autónoma sean capaces de mantener una interacción natural, confiable y eficiente.

La mayoría de los robots disponibles actualmente pueden interactuar únicamente con sus creadores o con un pequeño grupo de personas especialmente entrenados. Por lo tanto, de acuerdo a la familiaridad que tienen los usuarios con los robots se les puede asignar un nivel, esto se muestra en la Figura 2.7 [68].

En el nivel más alto están los expertos, especialistas y técnicos en robótica. En el siguiente nivel se encuentran los trabajadores de otras áreas que tienen una relación de colaboración con el robot. Luego están los trabajadores inexpertos quienes encuentran al robot como parte de su entorno aunque no trabajen directamente con él. En otro nivel están las personas con capacidades diferentes que tienen poca o ninguna experiencia de computadoras y/o robots. Después está el público en general que tiene poca o ninguna experiencia y finalmente los niños que tienen menos habilidad y experiencia.



Figura 2.7: Familiaridad con los robots.

Al estudio de la HRI se le puede dividir en dos enfoques diferentes: la HRI centrada en el robot y la HRI centrada en el usuario. La primera se enfoca en controlar al robot con el fin de mejorar su reconocimiento y percepción de los seres humanos. El enfoque centrado en el usuario, da prioridad a como los usuarios perciben y responden a los robots para de esta forma ajustar su comportamiento. Ejemplo de lo anterior incluye el estudio de los vínculos emocionales o sentimientos que los usuarios desarrollan en el transcurso de la interacción con robots [69] (ver Figura 2.8).

*Percepción, Cognición, Acción (PCA)

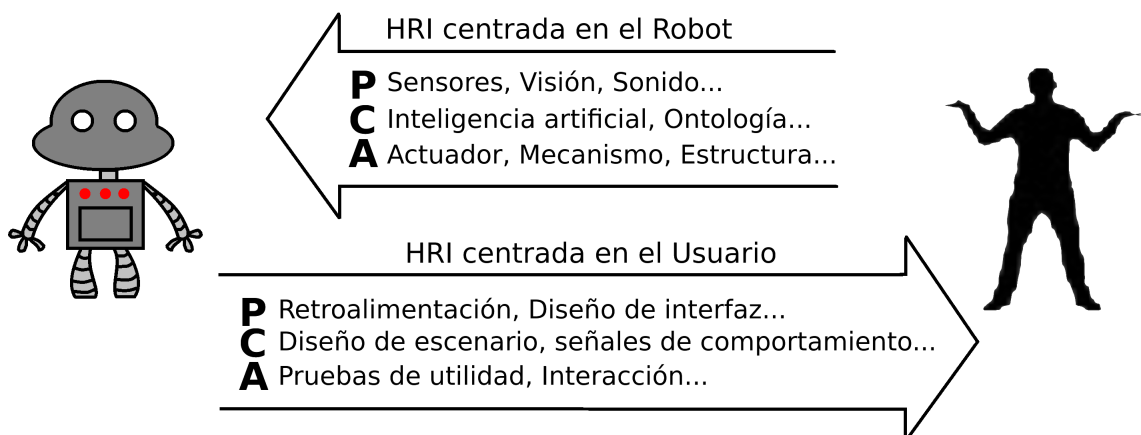


Figura 2.8: Interacción humano-robot centrada en robot y usuario.

Para el problema atacado en este trabajo, los dos paradigmas anteriores no satisfacen plenamente los requerimientos deseados y por lo tanto se implementará una combinación de estos dos paradigmas.

Además, es importante analizar los patrones de interacción existente y adoptar la perspectiva del diseñador, es decir, considerar el problema de HRI en los factores que intervienen en el proceso de comunicación entre el Humano y el Robot que son [67]:

- Nivel y comportamiento de la autonomía.
- La naturaleza del intercambio de información.
- Estructura del equipo.

- Adaptación, aprendizaje y capacitación de las personas y el robot.
- ¿Cuál es la forma en la que se debe realizar la tarea? (task-shaping).

Los factores anteriores dependen en gran medida de los procesos cognitivos humanos que se quiere transmitir al robot. A estos procesos cognitivos se le asignan tres elementos [70, 71, 72]: a la tecnología de sensores que permite al robot percibir su entorno se le denomina *percepción*, a la tecnología de IA que permite a un robot pensar utilizando las bases de datos de información y conocimientos percibidos, además de controlarse de manera autónoma se le denomina *cognición* y a la tecnología de actuador que permite al robot moverse y entrar en *acción*.

Entonces, para realizar esta interacción se deben estudiar técnicas de comportamiento, técnicas de comunicación y técnicas de extracción e interpretación de información.

A) Percepción: para la comunicación y comprensión de las personas, el robot debe obtener datos de dispositivos que le proporcionen la información necesaria. Estos pueden ser teclado, ratón, joystick, pantallas táctiles, interfaces hápticas, sistemas de reconocimiento de voz, sensores ópticos, etc.

B) Cognición: sirve para conectar las entradas de información obtenidas en la percepción con las salidas que accionan a los actuadores. Dentro de la cognición existen dos tipos importantes de memorias: la memoria operativa que mantiene la información actual y la memoria a largo plazo que almacena la información para un futuro. La memoria operativa consiste en un conjunto de elementos de la memoria a largo plazo que han sido activadas. La memoria a largo plazo consta de una red de fragmentos relacionados, accedida a partir de los requerimientos de la memoria de trabajo. Esta fase de conocimiento debe ser paralela a la fase de percepción y serial con la fase de acción [72].

La emoción humana es un dispositivo comunicativo, permite reflejar sensaciones de la persona, por ejemplo manifestar hambre. Dado que algunas necesidades son más urgentes que otras, las emociones reflejan un conjunto multidimensional complejo de disparos, su reflexión es un conjunto similarmente complejo de comportamientos. Por ejemplo, las expresiones faciales, son una manera rica y sofisticada para comunicar mensajes complejos. La etología estudia los comportamientos de los animales, en especial los domésticos. Este estudio junto con el estudio de las emociones pueden servir como base para desarrollar aplicaciones para robots que convivan con humanos [68].

Otro factor importante es la personalidad del robot, para ello combina conductas, creencias e intenciones en un comportamiento cohesivo y consistente. Siendo este una simplificación drástica de la personalidad humana. Proporcionar una personalidad a un robot puede dar a los seres humanos buenos modelos y un buen entendimiento del comportamiento del robot [68].

C) Acción: cuando el robot entra en movimiento, se utilizan actuadores tales como motores de corriente continua, motores paso a paso, cilindros neumáticos, actuadores hidráulicos, motores de engranajes, servomotores, etc. Si además se le proporciona al robot la capacidad de interactuar con el humano de una manera más amigable, también debe contar con dispositivos que le permitan ejecutar esta comunicación. Ejemplo de ello puede ser dotar al robot de voz, un rostro que gesticule, pantallas, entre otras cosas.

El robot deberá reconocer e interpretar su entorno realizando un análisis similar al que realiza una persona. Para esto se han estudiado normas sociales, comportamientos, respeto de espacios, criterios culturales, comunicación, etc. para poder adaptar al robot a nuestro entorno social. También es importante el estudio de la percepción que el usuario tiene respecto a los robots, es decir, que grado de aceptación tiene la integración de los robots en la sociedad

así como evaluar los beneficios que ofrece. A continuación se muestran algunos trabajos de investigación en el área de HRI.

En [73] el robot toma el papel del alumno y aprende el léxico de un maestro humano. En esta investigación se concluyó que un robot activo en interacción bilateral puede crear más y mejores datos de aprendizaje y a su vez motiva al humano a involucrarse más en la interacción con el robot. La sincronía no intencional es un factor importante en la interacción humano-robot, deben incluirse otros comportamientos para una comunicación natural. Por ejemplo movimiento de manos, gestos, etc. [74].

Otros factores que influyen en la HRI son la edad, personalidad, familiaridad con robots y género. Los niños tienden a alejarse más del robot que los adultos [75]. Las personas extrovertidas son más tolerantes a la invasión del espacio personal como se muestra en [76] y [77]. La gente que está familiarizada con los robots también muestran más cercanía que los demás [78]. Respecto al género, las mujeres prefieren que el robot se aproxime de frente y los hombres por un costado [79, 80].

Además de los anteriores, los adultos tienden a acercarse más a un robot con voz sintetizada que a robots con voz femenina, masculina o sin voz [81]. También el diseño del robot influye pues las personas prefieren un modelo mecanizado a uno humanoide [82, 48]. Un robot móvil con una velocidad de $1m/s$ es incómodo para las personas [48], mientras que su altura no influye de manera significativa en la aceptación del robot [81].

En [83] se realizó un estudio para ver qué tan comfortable era un robot, si el robot era del gusto de la persona, la mirada de este no importaba. Cuando el robot no agradaba al usuario, este se incomodaba cuando la mirada del robot era dirigida hacia él.

Otro estudio exploró las interacciones humano-robot centrándose en los efectos psicológicos que experimentan las personas. Con base en los resultados obtenidos, las personas se sienten más cómodas con una velocidad menor a $1m/s$, con una aproximación indirecta del robot, sin cambios bruscos de comportamiento y con una estructura no humanoide.

En este trabajo de tesis, se realizó una metodología para la interacción del usuario con el Tbot. La estrategia de HRI elegida para la interacción es la proxémica, donde el robot debe respetar el espacio personal del usuario.

El término “proxémica”, introducido en 1959 por el antropólogo Edward T.Hall, refiere al estudio de los patrones culturales que se usan para construir, manejar y percibir el espacio social y personal [84]. Hall divide en cuatro zonas el contacto del espacio con los demás [85]:

- Distancia íntima: 0 - 0.45 *m*.
- Distancia personal: 0.45 - 1.20 *m*.
- Distancia social: 1.2 - 3.7 *m*.
- Distancia pública: 3.7 - 7.6 *m*.

En la Figura 2.9 se muestran las cuatro zonas de contacto. Para nuestro caso el robot Tbot operará en el espacio social (1.2 - 3.7 *m*) para dar espacio de movimiento al usuario sin salir de su espacio de interacción social. Además, para mejorar la interacción con las personas, se implementó un algoritmo de voz para que el Tbot pueda comunicarse con el usuario.

Dentro del estudio de la HRI es importante conocer la percepción y apreciación que tiene el usuario por parte del robot en diferentes aspectos. Hay cuatro métodos principales de evaluación utilizados para el estudio de HRI:

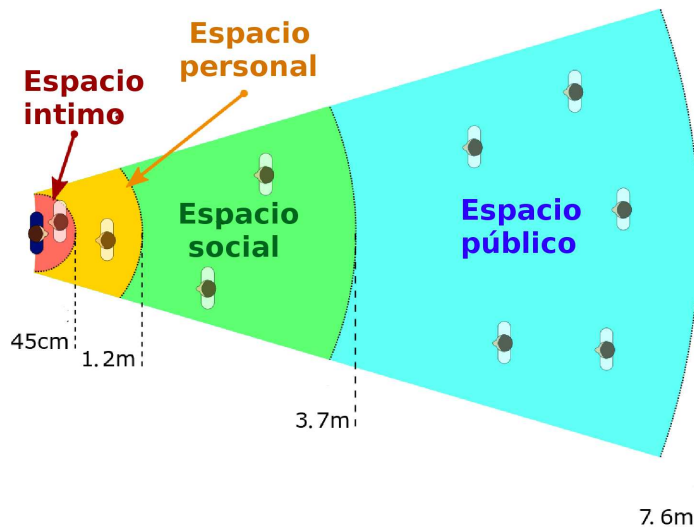


Figura 2.9: Zonas de contacto espacial de una persona.

- Medidas de auto-informe.
- Medidas de comportamiento.
- Medidas psicofisiológicas.
- La ejecución de tareas [86, 87].

Los métodos más comunes utilizados en la mayoría de los estudios HRI son auto-informe y las medidas de comportamiento [88].

El método de medición de auto-informe incluye papel y lápiz, escalas psicométricas, cuestionarios y/o encuestas. Los participantes proporcionan un informe personal de sus motivos y sentir acerca de un objeto o situación [88]. Los auto-informes proporcionan información valiosa pero no exacta, dada la falta de validez y corroboración. Podría incluso depender del estado de ánimo de los participantes [89, 90], por esto, es importante añadir otro tipo de mediciones que confirmen los datos obtenidos.

Existen diferentes artículos que evalúan el grado de aprobación que tienen las personas en la interacción con un robot. Por ejemplo, en [91] se evalúan las conductas socialmente aceptables de un robot no-antropomórfico. En [92] se revisa el efecto que causa la apariencia de un robot recepcionista y el sonido de la voz de este, una de las apariencias era dada por gráficos de ordenador y la otra era un robot humanoide real. En [93] se realizaron encuestas por internet, los resultados mostraron que la mayoría de las personas prefiere un robot humanoide para fines domésticos. En [94] se realizó un estudio con 5 participantes a lo largo de 5 semanas para conocer las diferencias en la conducta de las personas cuando conocen por primera vez al robot y cuando se habitúan a él. Estos estudios de interacción son importantes para controlar la acción del robot y sea aceptable para el usuario y las personas que utilizan el mismo entorno.

2.5. Proceso inteligente adaptativo para el seguimiento de trayectorias

Para este proceso se propone un sistema de control inteligente para la guía de personas, el cual debe considerar las áreas anteriores (HRI, navegación y visión) y combinarlas en un solo lazo, incluyendo todas las variables del sistema (ver Figura 2.10).

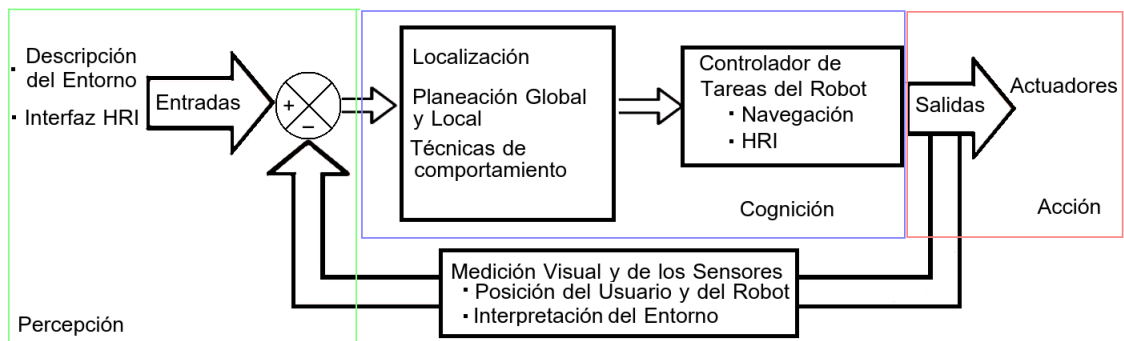


Figura 2.10: Lazo cerrado de control para el accionamiento del robot.

En la etapa de percepción, el robot obtiene los datos necesarios a partir de dispositivos para obtener una descripción del entorno se utilizan cámaras, sensores de profundidad, de radiofrecuencia, etc. La interfaz HRI brinda a las personas un servicio e interpreta el objetivo del usuario, esto se realiza con teclados, joystick, pantallas táctiles, controles de mando, entre otras. La retroalimentación de la medición visual y de otros sensores brinda la información de donde se localiza el robot y el usuario dentro del área de trabajo. También proporciona los datos del entorno para una navegación segura, por ejemplo para evadir obstáculos.

En el área de cognición se cuenta con la información obtenida en la etapa de percepción, esta se procesa junto los datos que se le proporcionan al robot para que realice una planificación hacia su punto objetivo y sea capaz de prestar un buen servicio al usuario con técnicas de comportamiento establecidas, tales como, esperar al usuario, búsqueda en caso de pérdida, etc. Una vez que el robot cuenta con una tarea a realizar, entra a una fase de control, donde las leyes de control determinan que acción tomar. Estas leyes influyen en el movimiento y el comportamiento del robot.

Finalmente las señales de salida accionarán a los actuadores para ejecutar el movimiento del robot en el mundo real.

2.6. Arquitectura de software

La actividad final de este trabajo de investigación es el de ofrecer guías turísticas a los visitantes de un edificio a través de un robot móvil. Para realizar esta tarea se implementó una Arquitectura de Software (SA) para controlar al robot. Existen muchas formas de SA para el control de un sistema, estas permiten controlar los comportamientos del robot en distintos niveles y de forma paralela, lo cual permite realizar una tarea compleja. A continuación se detallan algunas de las propuestas que existen en la literatura.

En [95] se propone una arquitectura de control que consiste en cuatro capas: manejo de dispositivos, control, coordinación y organización, cada capa se controla de manera autónoma y su procesamiento es de forma paralela, está compuesta por cuatro módulos generales: navegación con control, planeación de trayectoria global, operaciones de control y comunicación. La arquitectura en [96] se compone de seis capas: hardware, dispositivos, middleware, componentes, servicios y aplicaciones. Una arquitectura diferente se propone en [97] basada en toma de decisiones por control, que consiste en un sistema de gestión de tareas, un sistema de procesamiento de información y un sistema de navegación, cada uno independiente y paralelo de los otros.

En [98] se presenta una arquitectura jerárquica para el control de un robot humanoide, consta de cinco capas de implementación: dos capas de comportamiento, dos de tareas y una de estado del robot. El controlador del robot juzga el estado del robot y proporciona al robot comportamientos basados en la percepción y el control autónomo de los actuadores. Una arquitectura de estructura modular se presenta en [99], aquí los módulos están separados por las características y responsabilidades, primero se encuentra el nivel básico donde están los dispositivos y sensores, después un nivel reactivo para mantener la integridad del robot, los siguientes módulos controlan las funciones de movimiento de bajo nivel (velocidad, aceleración, posición) y alto nivel (navegación, teleoperación, manipulación).

En este proyecto de investigación se propone una arquitectura de 6 capas para controlar al robot móvil, esta se presenta en la Figura 2.11. En la primera capa se encuentran los dispositivos de entrada y salida, en la segunda capa se procesa la información que se extrae de los sensores y la que se envía a los actuadores. La tercera capa muestra las habilidades del robot, es decir, los procesos que el robot realiza ya sea de forma individual o en conjunto. Después, en la cuarta capa se encuentra la operación que ejecutará el robot, dependiendo del comportamiento asignado en la quinta capa. Finalmente, en la última capa se encuentra el supervisor. Todas las capas están conectadas al middleware ROS para la transferencia de información entre procesos, el cual se describirá en la Sección 2.7.

2.6.1. Primera capa: dispositivos

Los dispositivos empleados son aquellos sensores y actuadores con los que cuenta el robot que se utilizan para tal aplicación, estos fueron descritos en la Sección 2.1.1. En general, se dispone de una base móvil de tipo diferencial, sensor de rango y RGB, giroscopio, codificadores, motores, bocinas y pantalla.

2.6.2. Segunda capa: percepción y control

En esta capa se procesa la información obtenida de los sensores, para la localización del robot, trazado de trayectorias, estimación de posición y rastreo del usuario. Además se proporciona el movimiento al robot por medio de comandos de control de velocidad, los cuales son traducidos a señales eléctricas que accionarán los motores del robot de acuerdo a la velocidad establecida. En esta parte también se accionan las bocinas y se recibe el dato de la posición destino. Estos datos son obtenidos o enviados por medio de ROS.

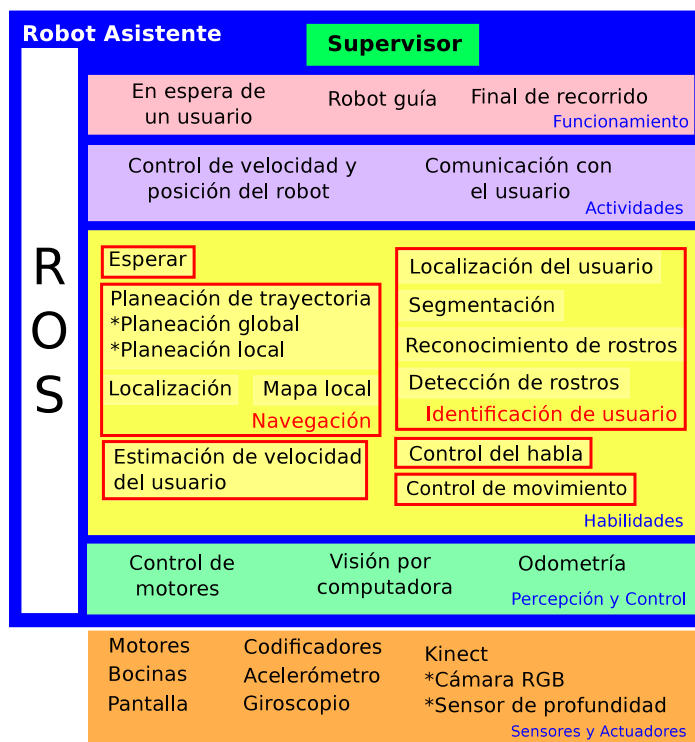


Figura 2.11: Arquitectura de software propuesta.

2.6.3. Tercera capa: habilidades

Una vez obtenida la información de los dispositivos, la siguiente capa contiene las capacidades que tiene el robot y que realiza de manera separada o en forma paralela. A continuación se describen cada una de ellas.

El **estado de espera** es con el que inicia la aplicación de guía, cuando el robot está en la recepción y en espera de un usuario. El robot debe conocer el mapa en el que se encuentra y conocer su posición actual. El movimiento del robot es nulo mientras que la detección de las personas se realiza. El algoritmo de detección comprueba si existe algún rostro en el área de visión, si detecta uno, el robot comenzará la interacción invitando al posible usuario a utilizar sus servicios.

La persona que desee utilizar el servicio deberá colocarse frente al robot y utilizar las opciones de una pantalla, esto permitirá que el **algoritmo de identificación y localización de usuario** almacene su información. Esta es la primera actividad que realiza el robot luego de iniciar la aplicación, utiliza los datos que proporciona el sensor Kinect de la parte superior (RGB y profundidad) y por medio de un control visual, es posible determinar cuando alguien se encuentra dentro del campo de visión de la cámara. Este proceso se realiza con la información del sensor RGB y la localización de personas se realiza en un plano bidimensional a lo largo de la imagen (640x480).

Para este propósito se generó un algoritmo que se detalla en el próximo Capítulo, las personas son encontradas a través de su rostro. La clasificación entra como auxiliar para la ubicación de la persona en caso de perder la detección del rostro y después se realiza una segmentación con la información de la distancia obtenida del sensor depth para la localización del usuario en un

espacio tridimensional.

Al realizar la detección del usuario, este se ubica dentro de la imagen y posteriormente se localiza en un espacio tridimensional. La posición del usuario (x, y, z) es en referencia al marco referencial de la cámara.

La **velocidad del usuario** es estimada con base en la distancia que recorre en un intervalo de tiempo. Ya que el robot se encontrará en movimiento, la posición obtenida en el apartado anterior (localización de usuario) no brinda la velocidad real del usuario pues este punto se moverá con él. Entonces, para el cálculo de la distancia recorrida por el usuario, la posición de este se transforma del marco referencial de la cámara al marco referencial del mapa.

Como se muestra en la Figura 2.12, el usuario es localizado en la posición $(1, -0.6)$ respecto al marco referencial de la cámara (Mr), esta posición se transforma al marco referencial del mundo (Mm). La persona ahora es ubicada en el punto $(3.8, 1.2)$.

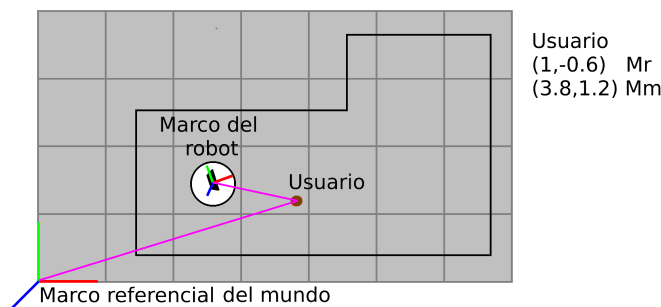


Figura 2.12: Marco referencial del mundo.

Por otra parte, es importante el estudio de la marcha humana para conocer y adaptarnos mejor a ella. En la marcha humana, el cuerpo debe desplazarse conservando el equilibrio. El centro de gravedad (CdG) durante la marcha describe una trayectoria sinusoidal tanto en el plano horizontal (Figura 2.13) como en el sagital. El desplazamiento del CdG tiene una amplitud pico - pico de aproximadamente 5 cm , esta medida es necesaria para que la marcha sea eficiente [100]. Además de acuerdo a [101], la velocidad promedio de una persona adulta sin dificultades físicas es de 1.2 m/s .

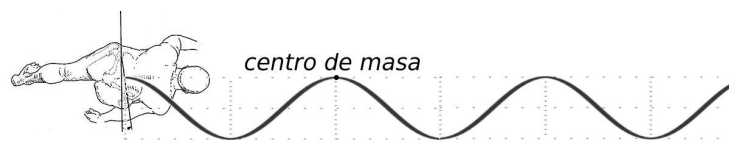


Figura 2.13: Desplazamiento del centro de gravedad durante la marcha [100].

Debido al balanceo y otros factores, la marcha no es constante en posición ni en velocidad. Lo anterior ocasiona movimientos erráticos e inseguros en el robot, ya que trata de igualar la velocidad del usuario, poniendo en riesgo la integridad de ambos. Además, la velocidad debe modificarse cuando se encuentre con obstáculos y/o la trayectoria sea curva. También, la supervisión constante de estos valores provoca demasiado cálculo computacional y variaciones significativas en la velocidad calculada. Para solucionar estos problemas, la señal de la velocidad estimada utilizada por el controlador debe ser suavizada para que el robot no presente cambios

bruscos de velocidad que pongan en riesgo la integridad de las personas y de él mismo. También se evaluó el periodo de toma de muestra más conveniente para nuestra aplicación.

En la **navegación** se realiza el cálculo de trayectorias, para esto, se cuenta con la información del punto inicial y el mapa de su entorno, así como la ubicación dentro del mapa de los lugares a los que el usuario puede dirigirse. El punto inicial que se tiene como referencia será la recepción, el robot regresará a esta posición cada que termine un recorrido. La elección del punto final se dará por el usuario a través de una aplicación de Android que muestre la lista de los lugares disponibles.

Entonces, se genera una trayectoria global del punto inicial al punto final, esta trayectoria se calcula realizando un cuadrículado en el mapa y definiendo las celdas libres por las que puede pasar el robot. También se calcula una trayectoria local para la evasión de obstáculos, la cual consiste en conocer la velocidad lineal y angular del robot, así como la proximidad de obstáculos, del punto meta y de la trayectoria global. Con esta información, el algoritmo realiza una simulación de la posición próxima del robot al cabo de un corto periodo de tiempo, para predecir si el robot colisionará.

En la Figura 2.14 se muestra un ejemplo de la navegación que realizan los robots dentro de un mapa conocido, las paredes y mobiliario son marcados con color rojo. Estas zonas son inaccesibles para el robot y se amplían (color azul marino) para calcular las trayectorias global (verde) y local (naranja) por las celdas libres. En color magenta se presenta un obstáculo que no es parte del mapa y por lo tanto no se conoce al iniciar la tarea de navegación. En el momento en el que el objeto es visible para el robot, la trayectoria local se desvía de la trayectoria global para lograr esquivar el obstáculo. Ambas trayectorias son calculadas en tiempo real y se actualizan de acuerdo a la posición actual del robot y la información del entorno, por lo que no necesita tener un conocimiento previo de la posición de los obstáculos en el mapa.

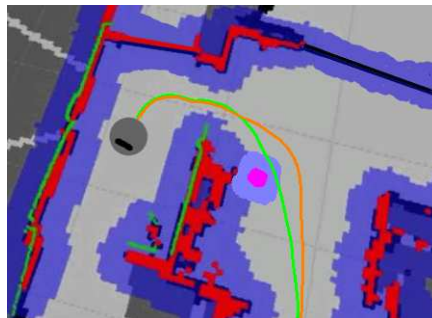


Figura 2.14: Navegación de robots móviles en entornos conocidos.

El **control de movimiento** del robot depende de la velocidad lineal y la velocidad angular. En este trabajo de tesis, se diseñó e implementó un controlador difuso diferente para cada una de las variables, dadas las necesidades de la aplicación de robot guía. Para adaptar la velocidad del robot a la velocidad del usuario, se implementó un controlador PI+D difuso. Este controlador tiene como referencia la velocidad estimada del usuario y como salida la velocidad lineal del robot. El controlador de lógica difusa Sugeno, manipula la velocidad angular para el seguimiento de trayectorias. Tiene como entrada la velocidad lineal del robot y el error en orientación. El diseño de estos controladores se describe en la Sección 3.4.

2.6.4. Cuarta capa: actividades

Las actividades que realiza el robot se resumen en dos acciones: el control de velocidad/posición del robot y la comunicación con el usuario. Algunas de las habilidades mencionadas en la Sub-sección 2.6.3 realizan ya sea de manera conjunta o separada estas acciones. La comunicación con el usuario se activa cuando se realizan las siguientes habilidades: espera, identificación de usuario, navegación, control del habla, control de movimiento. La acción de control de velocidad/posición del robot se encontrará activa en las habilidades de: navegación, identificación de usuario, estimación de velocidad y control de movimiento. Estas actividades se ejecutan de manera paralela y se explicará de manera clara en la capa de supervisor.

2.6.5. Quinta capa: funcionamiento

Para la tarea de robot asistente se consideran tres estados de funcionamiento básicos. Como se explicó anteriormente, el robot inicia en una etapa de espera en la que los actuadores de la base están inactivos hasta que encuentre un usuario; aquí el robot puede tratar de establecer una comunicación con un posible usuario, invitándolo a utilizar sus servicios con el algoritmo de control de voz, si la persona se encuentra a una distancia cercana. El funcionamiento del robot como guía, inicia al momento en que un usuario requiere la asistencia del robot para llegar a algún punto en específico, el robot reconoce a su usuario y una vez conocido el destino al que se desea llegar traza una trayectoria y modula su velocidad por medio de un sistema de control con la información estimada de la velocidad del usuario. Sin perder la comunicación con el usuario para ofrecer un recorrido eficiente y confortable. Al final del recorrido, el robot se despide del usuario y traza una trayectoria para volver a su origen. En este punto de operación la HRI se desactiva.

2.6.6. Sexta capa: supervisor

La capa del supervisor se encarga de dirigir todas las capas anteriores, es decir, en que momento debe accionarse o desactivarse cada una de ellas. A continuación se detalla paso a paso las actividades que realiza el robot, las cuales pueden verse en el diagrama de la Figura 2.15.

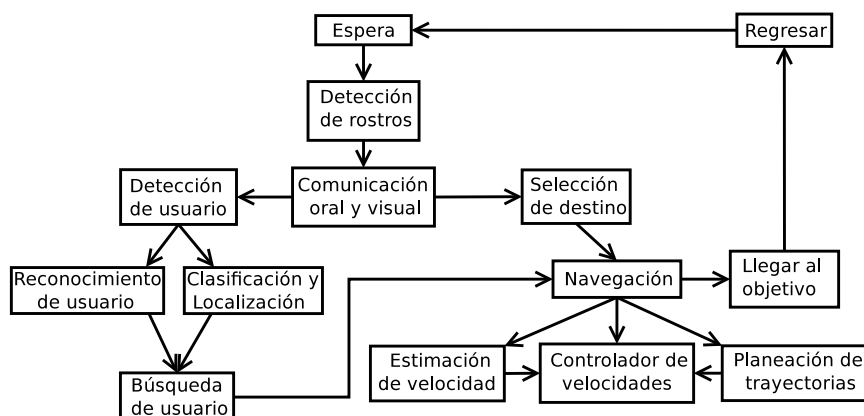


Figura 2.15: Actividades que realiza el robot.

Al comenzar, el robot se encontrará en una posición de inicio en espera de un usuario

invitando a los posibles participantes a disponer de sus servicios por medio de la detección de rostros y comunicación oral. Una vez que se elija un destino en la pantalla incorporada al robot (comunicación visual), el robot detecta, reconoce y ubica al usuario en un espacio tridimensional. La navegación traza las trayectorias para llegar al punto destino elegido por el usuario. El controlador utiliza la información de estas trayectorias y de la velocidad estimada del usuario para ajustar la velocidad del robot, además observa constantemente la posición del usuario para que en caso de que este se aleje de la zona de interacción (0.5 m a 1.5 m) cambie su acción de guía a búsqueda para acercarse al usuario. Finalmente, el robot llega a su destino y regresa a recepción.

2.7. Sistema Operativo de Robots (ROS)

ROS brinda soporte a ambas plataformas robóticas. Como se mencionó anteriormente, ROS es un conjunto de software de código abierto que provee librerías y herramientas para ayudar a los desarrolladores de software a crear aplicaciones para robots. ROS funciona como un sistema operativo, proporciona controladores de dispositivos y acceso al Hardware obteniendo la información de los sensores. A través de paso de mensajes entre procesos se puede manipular la información obtenida y accionar el robot por medio de velocidad lineal y angular, además ofrece algoritmos de aplicación para múltiples robots, visualizadores, simuladores, manejo de paquetes, etc. [102].

ROS opera sobre la distribución Ubuntu del sistema operativo Linux (y otras distribuciones). ROS inicia en el año 2007 y cuenta con 12 diferentes versiones, la última versión fue liberada en Mayo del 2017 y se llama ROS Lunar. En este trabajo de tesis se utilizaron las versiones Fuerte e Hydro.

ROS es un software distribuido de procesos, también conocidos como nodos, que permite a los ejecutables ser diseñados de forma individual y acoplados en tiempo real. Estos procesos se pueden agrupar en paquetes y pilas, que pueden ser fácilmente compartidos y distribuidos. Esta distribución se muestra en la Figura 2.16.

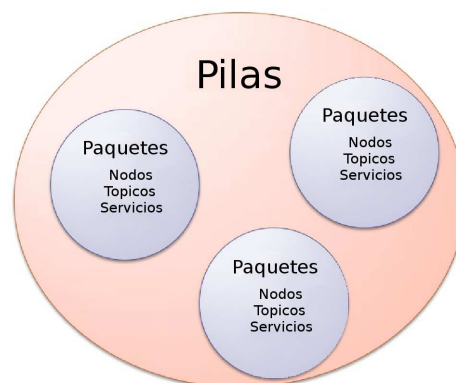


Figura 2.16: Distribución de ROS.

Los paquetes son la principal unidad de organización en ROS, las pilas son colecciones de paquetes con funcionalidad relacionada, los nodos son los procesos ejecutables. Tanto los paquetes como las pilas contienen archivos con metadatos, los repositorios son una colección

de paquetes y pilas disponibles en línea. También es necesario mencionar a los temas, estos son nombres que identifican el contenido de un mensaje. Los nodos pueden enviar y recibir mensajes por medio de los topics. El divulgador publica mensajes de un determinado topic y el suscriptor se suscribe a mensajes de un determinado topic.

Cuenta con una arquitectura de cliente/servidor entre los nodos donde se utilizan dos mensajes, uno en la solicitud y otro en la respuesta. Si el nodo cliente realiza una solicitud, el nodo servidor realiza procesamiento y responde al cliente. De esta manera se pueden realizar diferentes procesos de manera conjunta o por separado y agrupar distintas actividades a los comportamientos del robot.

2.8. Entorno

Para el diseño del robot deben considerarse las dimensiones del lugar donde podrá transitar, cuantas personas circulan en el mismo espacio, la iluminación del lugar, que tipo de personas podrían utilizar los servicios del robot, entre otras.

Para esta investigación, el robot desarrollará su actividad en el segundo piso del edificio 3 de la Facultad de Ciencias de la UASLP donde se encuentran las oficinas de algunos docentes de esta institución. En este piso se localiza un total de 24 oficinas, dos baños, una sala de maestros y escaleras que conducen al primer y tercer piso. En la Figura 2.17A se muestra el entorno donde actualmente se encuentra el robot Tbot. En el diseño del robot se consideró que los pasillos del edificio son estrechos (2 m de ancho), el robot no debe ocupar mucho espacio para dar oportunidad de movimiento al usuario y poder desplazarse libremente por los pasillos aun en presencia de obstáculos. Además, el conocimiento del espacio de trabajo ayudará al robot a trazar las trayectorias de su punto actual al punto meta.

El entorno donde opera el robot UVerto es el segundo piso del Centro de Investigación en Inteligencia Artificial (CIIA), el cual se muestra en la Figura 2.17B. El robot se ubica en la entrada al piso, frente a las escaleras (punto rojo), donde esperará por un usuario. Esta es la posición donde el robot debe volver una vez terminado su recorrido. Las zonas naranjas son espacios inaccesibles para el robot, las zonas azules son mobiliario (sillas, mesas, gabinetes) y espacio en blanco es por donde el robot puede desplazarse.

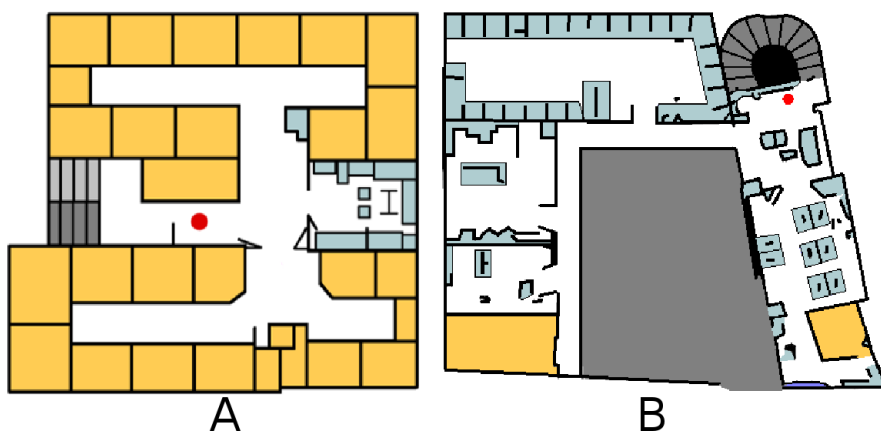


Figura 2.17: A) Mapa del entorno de Tbot. B) Mapa del entorno de UVerto.

ROS proporciona un paquete de navegación que puede generar un mapa usando el algoritmo GMapping, que por medio de un láser realiza una técnica llamada SLAM que consiste en utilizar un filtro de partículas en la que cada partícula lleva un mapa individual del entorno. Se calcula una distribución teniendo en cuenta no solo el movimiento del robot sino también la observación más reciente. Los mapas obtenidos al utilizar este algoritmo en los mapas mostrados anteriormente, se muestran en la Figura [2.18](#).

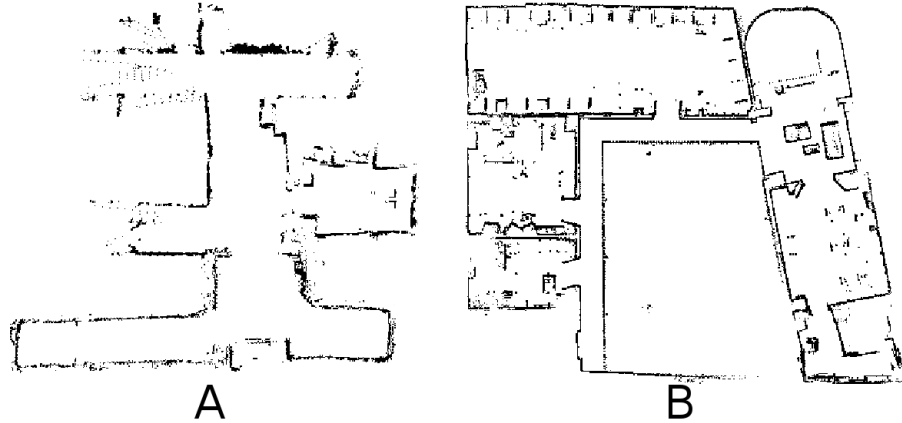


Figura 2.18: Mapa del entorno obtenido por láser A) de Tbot. B) de UVerto.

Los mapas obtenidos con el algoritmo GMapping de ROS (Figura [2.18](#)) se utilizan para comparar la información actual de los sensores con la información del mapa pre-grabado y así conocer su ubicación dentro del entorno.

2.9. Conclusiones

En este Capítulo se abordaron las diversas disciplinas y conocimientos necesarios para el desarrollo del robot guía, estas incluyen:

- Plataformas robóticas.
- Movimiento del robot.
- Navegación.
- Visión.
- Control visual.
- Control de movimiento.
- Arquitectura de software.
- HRI.
- Entorno.

Se utilizaron dos plataformas móviles, Tbot y UVerto; ambas con tracción diferencial no-holonómica. Se explicó la cinemática del robot, donde las variables a controlar son la velocidad lineal v y la velocidad angular w . Estas variables se controlan para obtener el movimiento deseado en el robot y dirigirlo a una posición en específico con una velocidad determinada. Algunos de los métodos de navegación implementados en los robots móviles guías se mencionan en el Capítulo II.

También se expuso la necesidad de utilizar visión tanto para la interacción con los humanos, como para el proceso de navegación. Para este último se implementa además el control visual, el cual facilita y mejora la auto-localización del robot. Para que esta localización resulte más sencilla, un mapa del área de trabajo es proporcionado al robot previo a la acción de guía.

El sistema de control de movimiento del robot necesita información de la navegación y las normas de comportamiento establecidas para la interacción con las personas. Estas reglas son elegidas con base en el estudio realizado sobre la HRI.

Los módulos implementados permiten al robot obtener la información necesaria para relacionarse con el usuario y con el entorno, para lograr la ejecución de la aplicación de guía de personas.

Capítulo 3

Desarrollo de algoritmos inteligentes implementados en robots móviles

En este Capítulo se describe el desarrollo de los algoritmos implementados en cada etapa del proceso del robot guía. Esta tarea inicia con la identificación y localización del usuario presentado en la Sección [3.1](#).

3.1. Algoritmos de visión para la detección y localización del usuario

La detección de personas es un caso especial dentro de los distintos grupos de algoritmos existentes en la comunidad de visión por computadora. Estos algoritmos extraen características tales como los contornos, tonos de piel, puntos de interés, etc., con el fin de hacer la detección. En [\[103\]](#) se propuso un método de detección extremadamente rápido, que se basa en la extracción de características tipo Haar para el reconocimiento de objetos donde algunos grupos de variaciones de contraste forman una característica Haar. También utiliza Adaboost para la clasificación, el cual utiliza el valor de cambio en el contraste para determinar las áreas de luz y oscuridad. Los clasificadores Haar son de los métodos más utilizados, ya que permiten realizar una segmentación en tiempo real.

El objetivo en la detección de rostros es básicamente reconocer si una persona se encuentra en el área de trabajo y localizarla con respecto a la posición 3D del robot. Existen muchos métodos utilizados para la detección de rostros, reconocimiento y seguimiento. El algoritmo para el seguimiento de rostros utilizado en este proyecto está basado en [\[103\]](#), en donde el clasificador Haar extrae características y el AdaBoost los clasifica.

El algoritmo propuesto, se divide en tres etapas; la primera realiza el reconocimiento de rostros con el algoritmo de Viola and Jones [\[103\]](#), el cual es validado por un umbral de color de piel y el porcentaje de dicho color en la región identificada como rostro. Después de detectar un rostro, se genera un perfil de color del usuario mediante una estadística del color de ropa usada por este. El perfil de color seleccionado de una región determinada (unos centímetros abajo del rostro) es el color predominante en el espacio de color: matiz, saturación y valor, conocido como HSV (por sus siglas en inglés Hue, Saturation, Value). Finalmente se realiza una segmentación

de los puntos correspondientes al usuario, con los datos de profundidad proporcionados por sensor Kinect. Con estos datos se ubica a la persona en un espacio de 3D.

3.1.1. Detección de rostros

El algoritmo para la detección de rostros realiza una primera transformación de la imagen mediante la generación de una nueva imagen llamada Imagen Integral (II). Se realiza una extracción de características usando filtros Haar, y finalmente se utiliza el Boosting para construir clasificadores en cascada.

La II fue introducida a la visión por computadora por [103], permite una forma rápida de calcular las sumas sobre subregiones de imágenes para extraer características a diferentes escalas. En la II, cada píxel es la suma del píxel actual y de los píxeles arriba y a la izquierda del mismo. Se puede calcular con la Ecuación 3.1, donde $II(x, y)$ es la Imagen Integral, $i(x, y)$ es la imagen original, x' va de 0 a x y y' va de 0 a y .

$$II(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3.1)$$

En la Figura 3.1 se muestra la imagen original y la II correspondiente. Para calcular la suma de una subregión de una imagen, se utiliza la región correspondiente de su II. Por ejemplo, en la Figura 3.1, la suma de la región verde se calcula usando cuatro valores de referencia de la región rectangular en la II correspondiente: $65 - 33 - 36 + 17 = 13$. El cálculo suma el píxel final de la subregión en la II (65) y resta las regiones arriba (33) y a la izquierda (36) de la región sombreada. El área de superposición (17) se agrega de nuevo para compensar la resta doble. De esta forma, puede calcular sumas en regiones rectangulares rápidamente, independientemente del tamaño del filtro.

3	5	1	9	2
2	7	3	3	8
5	4	6	2	1
8	2	4	1	6
7	1	3	6	4

Imagen original

0	0	0	0	0	0
0	3	8	9	18	20
0	5	17	21	33	43
0	10	26	36	50	61
0	18	36	50	65	82
0	25	44	61	82	103

Imagen integral

Figura 3.1: Imagen integral.

Las características de cada objeto son extraídas con la implementación de ciertos patrones, esto se realiza mediante la aplicación de filtros de tipo Haar los cuales realizan una detección de la diferencia de intensidad, que generan curvas de nivel, puntos y líneas, mediante la localización de zonas de contrastes, estos pueden ser calculados eficientemente sobre la II. La percepción se procesa para encontrar lineamientos que se pueden utilizar para clasificar un objeto en particular. Algunos de los filtros usados para la extracción de características de muestran en la Figura 3.2.

La clasificación asigna una clase de características similares, de acuerdo con un modelo obtenido mediante un entrenamiento. El método de boosting presentado en [104], consiste de

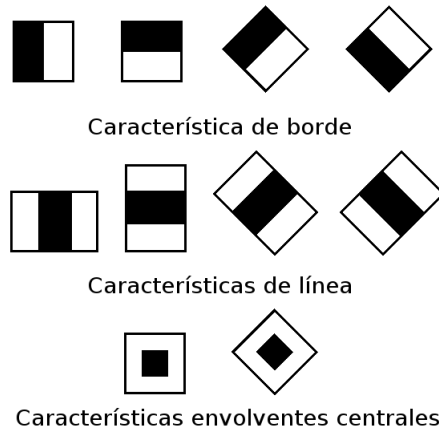


Figura 3.2: Filtros Haar.

varios clasificadores esenciales embebidos en un clasificador más complejo, siempre que tengan un número suficiente de muestras (en este caso 6000). Este algoritmo utilizado por Viola and Jones en 2001 [103] es conocido como AdaBoost y su procedimiento se muestra a continuación.

- Dadas unas muestras de imágenes $(x_1, y_1), \dots, (x_n, y_n)$ donde $y_i = 0, 1$ para muestras negativas y positivas, respectivamente.
- Inicializa los pesos $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ para $y_i = 0, 1$, respectivamente; donde m y l son los números de muestras negativas y positivas, respectivamente.
- Para $t = 1, \dots, T$.:

1. Normaliza los pesos $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$.
2. Para cada característica, j , entrenar un clasificador h_j el cual está restringido al uso de una sola característica. El error es evaluado respecto a $w_t, \epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
3. Escoge el clasificador, h_t , con el más pequeño ϵ_t .
4. Actualiza los pesos:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}. \quad (3.2)$$

donde $e_i = 0$ si la muestra x_i es clasificada correctamente, $e_i = 1$ en otro caso, y $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- El clasificador final es:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{en otro caso} \end{cases} \quad (3.3)$$

donde $\alpha_t = \log \frac{1}{\beta_t}$.

Por último se realiza una validación del algoritmo para asegurar que se encontró a una persona, con un umbral de predefinido del color de tez de piel y con un porcentaje mínimo de 40 % de dicho color de piel en el la zona del rostro [105]. El algoritmo ubica el rostro de hasta seis usuarios, tomando como referencia el punto central del rostro.

3.1.2. Clasificación del usuario

Después de detectar el rostro se realiza una clasificación del color de ropa del usuario en el espacio de color HSV para la localización de la persona de una forma más robusta. El primer paso es convertir la imagen obtenida por el sensor (RGB) al espacio HSV. En el espacio de color (HSV) el matiz proporciona el valor de cromaticidad, la saturación representa la pureza e intensidad de un color y el valor o luminancia corresponde a la claridad u oscuridad. Este espacio de color ofrece una forma más sencilla e intuitiva de segmentación a diferencia del espacio RGB como se explica en [106], por ejemplo, las zonas de sombra y brillo pueden provocar segmentaciones incorrectas en el espacio de RGB.

El algoritmo de detección de rostros es capaz de localizar a tres personas en el área de visión, siendo el más cercano el objetivo a clasificar y segmentar del resto de la imagen. En caso de que el sensor pierda de vista al usuario objetivo, entra en operación el algoritmo de clasificación para la localización del usuario por medio del color de su ropa. La metodología que utiliza se muestra a continuación.

1. Adquiere la información de la imagen en RGB y la convierte al formato de HSV.
2. Partiendo de la ubicación del rostro encontrado con el algoritmo anterior se obtiene una región de interés, la zona debajo del rostro del usuario.
3. Se detecta el color predominante de esta región mediante el cálculo del histograma en el espacio de color HSV.
4. Se segmenta este color en el campo de visión del sensor.
5. Se ubica al usuario en un espacio de 2D (u, v) , es decir, sobre los píxeles de la imagen (640, 480) dando prioridad al punto de referencia del rostro.

3.1.3. Segmentación y localización del usuario

Finalmente, se segmenta al usuario con el método de crecimiento de regiones, pero en esta ocasión en el espacio 3D correspondiente a los datos de profundidad proporcionados por el sensor. El método de crecimiento de regiones, trata de la selección de puntos de semilla inicial, examina los píxeles vecinos de los “puntos de semilla” iniciales y determina si los píxeles vecinos deben añadirse a la región. Esta adición depende de un criterio de región de pertenencia que podría ser, intensidad de los píxeles, la textura de nivel de gris, el color, la profundidad del objeto, etc.

El pseudocódigo del algoritmo se presenta a continuación:

Entradas:

- Lista de regiones= $\{R_i\}$.
- Puntos disponibles= $\{P\}$.
- Punto semilla= $\{S\}$.
- Región actual $\{R_c\} \rightarrow \emptyset$.

Algoritmo:

- Mientras $\{P\}$ no sea cero:
 - Leer $\{S\}$.
 - Encontrar vecinos más cercanos del punto semilla actual $\{S_c\} \rightarrow \{S_c\} \cup \{P\}$.
 - Si $\{S_c\}$ cumple con criterio de pertenencia agregar a $\{R_c\} \rightarrow \{R_c\} \cup \{P\}$.
 - Sino $\{R_{c2}\} \rightarrow \{R_{c2}\} \cup \{P\}$.

Donde R_i debe cumplir con:

- La segmentación debe ser completa, es decir, cada píxel debe pertenecer a una región.
- Los puntos de una región deben estar conectados de algún modo predefinido.
- Las regiones deben ser disjuntas.

El punto de semilla inicial está dado por los algoritmos anteriores, el cual corresponde al punto central del rostro o de la persona. El criterio para la adición de los píxeles vecinos será la distancia a la que se encuentra el usuario que es proporcionada por el sensor Kinect. Además, el sensor brinda la ubicación de los objetos en su área de visión en un espacio de 3D (x, y, z) , haciendo posible la reconstrucción del lugar de trabajo y la localización del usuario en este.

Para la localización en 3D se utiliza las librerías de nubes de puntos (PCL), estas librerías son independientes y de código abierto [107].

3.1.4. Reconocimiento de rostro

El algoritmo de visión puede confundirse al encontrarse dos o más personas con un color similar en su vestimenta. Para esto, se adaptó un algoritmo de reconocimiento de rostros proporcionado por ROS, llamado “face recognition”. El reconocimiento de rostros se realiza utilizando Eigenfaces también llamado “Análisis de Componentes Principales” o PCA (por sus siglas en inglés Principal Component Analysis) al utilizar el código fuente de C++ proporcionado por Shervin Emami en [108].

Se utiliza una base de datos de 25 imágenes, obtenidas del usuario, al inicio de la aplicación de guía. Esta base de datos se utiliza para hacer la comparación de la persona encontrada y decidir si pertenece o no a un nombre en la base de datos, se utiliza también un umbral de probabilidad con valores desde 0 hasta 1, este umbral es ajustable.

Las imágenes de entrenamiento se almacenan en el directorio de datos y se enumeran en un archivo. El programa se entrena a partir de los ejemplos de entrenamiento enumerados en el archivo guardado y crea una base de datos de Eigenfaces que se almacenan en otro archivo. La detección de rostros se realiza utilizando un clasificador de haarcascade (que es el mismo que se utiliza para la detección de rostros).

3.1.5. Implementación

OpenNI y OpenCV son proyectos enfocados en la integración de sensores PrimeSense con ROS. El controlador ROS es compatible con los dispositivos de PrimeSense (PSDK5.0) y con el Kinect [109]. Los algoritmos son implementados ROS, el cual permite obtener información del sensor y procesarla.

Con los datos que proporciona el sensor (RGB y profundidad) es posible determinar, cuando alguien se encuentra dentro del campo de visión de la cámara (2D) donde serán detectados y localizados en un espacio de tres dimensiones. A continuación se debe realizar una transformación de esta ubicación (punto donde se encuentra el usuario) con respecto al mundo. El usuario es localizable en el entorno de trabajo del robot y del mapa, ver Figura 3.3, donde se realiza una transformación de la siguiente manera:

$$H = \sqrt{Zu^2 + Xu^2} \quad (3.4)$$

$$\alpha = \tan^{-1}\left(\frac{Xu}{Zu}\right) \quad (3.5)$$

$$PoseX = \cos(\theta + \alpha) + Xr \quad (3.6)$$

$$PoseY = \sin(\theta + \alpha) + Yr \quad (3.7)$$

donde: $Zu, Xu =$ posición a la que se encuentra el usuario del sensor, α es la orientación de la persona respecto al robot, H es la distancia a la que se encuentra el robot de la persona, Xr, Yr es la posición en X, Y del robot respecto al mapa, θ es la orientación del robot respecto al mapa, $PoseX, PoseY$ es la posición del usuario respecto al mapa (transformación final).

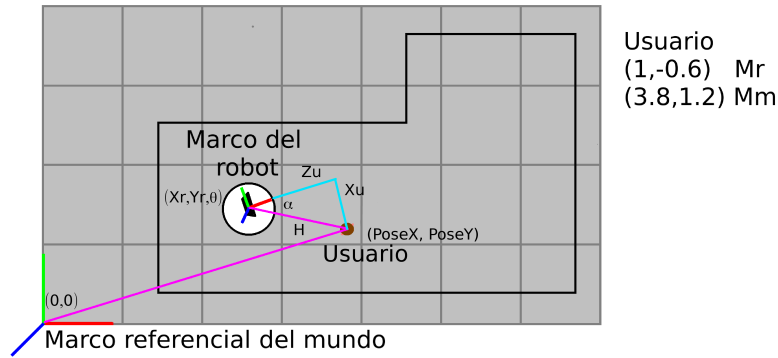


Figura 3.3: Entorno de trabajo, mapa.

El algoritmo de detección de personas en combinación con el algoritmo de segmentación 3D, son capaces de seguir a una persona dentro del espacio visual del sensor. En el Capítulo 4 se presentan varios casos en los que el algoritmo es sujeto a condiciones reales aplicado a la guía de personas.

3.1.6. Estimación de la velocidad del usuario

Por medio del algoritmo de visión se obtiene la localización del usuario en un espacio tridimensional con coordenadas dentro del marco referencial del mundo, esta información es utilizada para estimar la velocidad del usuario que será la señal que proporcionará la velocidad lineal al robot. Pero, debido a que la señal de salida presenta cambios bruscos de velocidad se utilizan las curvas de Bézier para suavizar la señal que entra al controlador. La ecuación de las curvas cúbicas de Bézier [110] se muestra en Ecuación 4.8.

$$B(t) = P_0(1 - t)^3 + 3P_1t(1 - t)^2 + 3P_2t^2(1 - t) + P_3t^3, \quad t \in [0, 1] \quad (3.8)$$

Además, en [111] se estudia la marcha humana a lo largo de trayectorias curvas, en este estudio concluyeron que la cabeza gira antes que el torso y que la persona reduce su velocidad en curvas para poder dar vuelta. Estos datos son considerados al momento de realizar el control de velocidad del robot.

3.2. Navegación reactiva y pasiva

Cuando el robot realiza un seguimiento de trayectoria con una navegación pasiva, esta proporciona al robot una trayectoria de un punto inicial a un punto final, sin considerar posibles obstáculos en el camino. La navegación reactiva, permite al robot esquivar obstáculos sin la necesidad de conocer la ubicación de dicho obstáculo.

ROS provee un paquete de navegación para el robot [112], el cual utiliza el láser como sensor para ubicarse en un espacio dentro del mapa. Este paquete utiliza el algoritmo AMCL que es un sistema de localización probabilística para mover un robot en un espacio de 2D, se utiliza esta librería para la localización del robot en el mapa. Esta librería implementa el enfoque Adaptativo de la Localización de Monte Carlo, el cual utiliza un filtro de partículas para rastrear la posición del robot en un mapa conocido, las funciones esenciales para la navegación se describen con detalle en [113] y se explicarán brevemente en las siguientes subsecciones, estos son: odometría, modelo de telémetro, modelo de probabilidad del sensor de profundidad, localización de Monte Carlo aumentada.

3.2.1. Odometría, modelo de muestra del movimiento

Esta función utiliza el método de muestreo, donde se acepta una posición inicial x_{t-1} y la información de odometría u_t como entrada y proporciona una salida aleatoria de la siguiente posición x_t , esta función es más sencilla de implementar pues adivina de manera aleatoria una nueva posición, a diferencia de otros que calculan la probabilidad de una nueva posición. También evita la necesidad de utilizar un modelo inverso.

3.2.2. Modelo de telémetro

Para este modelo se incorporan cuatro tipos de errores en las mediciones del telémetro: pequeños ruidos de medición, errores debido a objetos inesperados, errores debido a fallos en la detección de objetos y ruido inexplicable. La entrada de la función es una exploración completa del espacio z_t , la posición del robot x_t y un mapa m . La probabilidad de la posición del robot respecto a un mapa se obtiene a partir del producto de las mediciones de probabilidad individuales (Ecuación 3.9), donde las cuatro distribuciones del error se mezclan de acuerdo a una función de peso para obtener cada probabilidad de z_t^k .

$$p(z_t|x_t, m) = \prod_{k=1}^K p(z_t^k|x_t, m) \quad (3.9)$$

La función entrega la probabilidad de los objetos que se encuentran en el campo de visión de la cámara.

3.2.3. Modelo de probabilidad del sensor de profundidad

La función anterior tiene dos inconvenientes mientras más se vincule con la geometría y la física de los telémetros: la falta de suavidad, por ejemplo al existir múltiples objetos pequeños que puedan desorientar los cálculos del robot y la complejidad computacional del cálculo de cada medición del sensor de profundidad.

La función de modelo de probabilidad del sensor de profundidad entra como refuerzo del anterior, esta calcula la probabilidad de la posición (x, y) de los objetos más cercanos dentro del mapa de manera rápida. Los inconvenientes que presenta este modelo son: no es capaz de percibir correctamente objetos que causen lecturas cortas (ej. una persona cruzando); debido a que obtiene los cálculos de los objetos más cercanos, ignora los más lejanos haciéndolo incapaz de determinar trayectorias a un punto sin ser interceptado por algún obstáculo; finalmente, no puede manejar áreas inexploradas o no especificadas en el mapa.

3.2.4. Localización de Monte Carlo aumentada

El algoritmo básico de Localización de Monte Carlo (MCL) representa la suposición de una ubicación dado un conjunto de partículas. La posición global inicial es localizada a través de un conjunto de partículas posicionadas de manera aleatoria y uniforme sobre todo el espacio de posición.

La función de MCL Aumentada, es capaz de re-localizarse en caso de que el robot sea secuestrado o que tenga fallas de localización global, esto se logra añadiendo partículas al azar a los conjuntos ya existentes. Utilizando las nuevas partículas junto con los factores de peso correspondientes se suponen nuevas posiciones, es decir, las partículas que mejor se acoplan con lo que el sensor de profundidad observa tienen mayor peso que las demás.

Todas estas funciones en conjunto con otros parámetros que proporciona el paquete de navegación de ROS proveen al robot la información necesaria para navegar en un mapa conocido. Si se proporciona un punto meta al algoritmo MCLA, calculará una trayectoria global y una local además de la velocidad lineal y angular a lo largo del recorrido.

3.3. Metodología para la interacción humano-robot

3.3.1. Navegación con HRI

Se establecen dos enfoques como estrategias de HRI para la navegación (Figura 3.4 a y b).

- Seguidor-Líder (robot-humano): teniendo la transformación de la posición del usuario en el mapa (3D), el robot puede ir a encontrarlo con velocidad constante ubicándose en los alrededores del punto donde se localiza el usuario ($PoseX, PoseY$). En la Figura 3.4 a), la línea color verde representa la trayectoria que realizará el robot.
- Seguidor-Líder (humano-robot): el robot realizará la tarea de guía de personas siguiendo una trayectoria y utilizando como criterio para avanzar la distancia a la que se encuentra el usuario del robot, distancia H Ecuación 3.4. El rango de distancia a la que debe estar el usuario para que el robot avance es de 0.5 a 1 m, una vez que se excede este umbral el robot se detiene y prosigue su camino al volver a tener cerca al usuario. El robot se maneja a una velocidad constante y se detiene bruscamente si el usuario se aleja del rango antes

mencionado. En la Figura 3.4 b), el área en el que el usuario debe encontrarse para que el robot avance se encuentra limitada por la línea naranja y la línea verde es la trayectoria que realizará el robot mientras el criterio de distancia se cumpla.

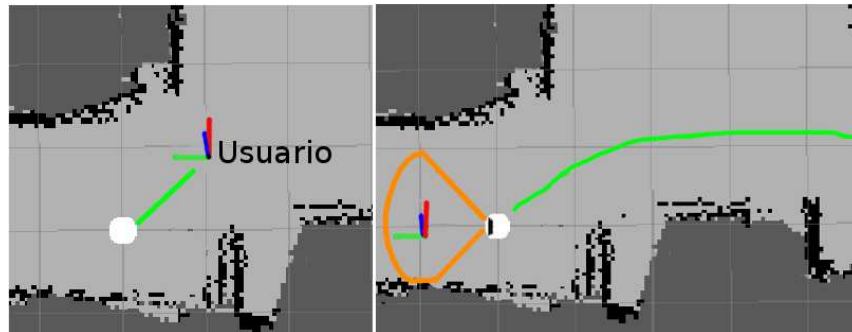


Figura 3.4: Seguidor-Líder a) Robot-Humano, b) Humano-Robot.

En la metodología de seguidor-líder, existen dos opciones básicas: esperar que la persona (líder) complete su recorrido o simplemente seguir la posición actual del líder. Para cualquier opción, el robot debe de ser capaz de guardar la información de la posición del líder y la propia en coordenadas del mundo [114, 115]. Para después repetir la trayectoria del líder usando un algoritmo de seguimiento de trayectoria tal como seguimiento puro [116]. Pero, seguir la posición del líder en coordenadas del mundo no es un problema fácil. Además, este método se hace más difícil para el seguidor tratando de mantener en el campo de visión de la cámara, a menos que el robot esté equipado con una cámara pan-tilt [117, 118].

Para robots simples, es más práctico seguir al líder (usuario) directamente, varios trabajos se han realizado al respecto [119, 120]. Si la posición y orientación del líder es conocida, el seguidor puede calcular una trayectoria que alcance la posición usando el método de seguimiento vectorial [121] o trayectorias de Bezier [118, 122].

Si la orientación del líder no está disponible, un método de “atrapa colas” o tail chase se adopta donde la posición actual o posición/velocidad del líder son considerados como objetivo de seguimiento utilizando el modelo cinemático del vehículo [123, 124].

Una metodología más simple reduce el problema de controlar el movimiento del robot en coordenadas cartesianas a través de un problema de servocontrol visual [125, 126, 127, 128]. Aquí el robot siempre trata de mantener la imagen del líder centrada en su campo de visión controlando las llantas del robot. La desventaja es que no sigue perfectamente la trayectoria del líder.

Se optó por considerar el enfoque de seguidor-líder (humano-robot) para la acción de guía de personas y el enfoque seguidor-líder (robot-humano) cuando el usuario se pierda y el robot deba buscarlo para llegar a una posición cercana y recuperarlo. Siendo esta, parte de la estrategia de recuperación de usuario.

3.3.2. Plan de búsqueda de usuario

Cuando el usuario sale del campo de visión del robot, este comenzará el plan de búsqueda que consiste en 2 etapas:

Etapas 1

El robot tratará de buscar al usuario, girando sobre su propio eje. Si localiza el color deseado

o un rostro, el robot se acercará a esa posición manteniendo la distancia de interacción. Esta primera etapa de búsqueda se realiza con la suposición de que el usuario no se ha alejado mucho del campo debido a que el robot, a lo largo del trayecto, siempre trata de mantener al usuario en la mira.

Etapa 2

Cuando esté lo suficientemente cerca, detectará por medio del algoritmo de reconocimiento de rostro, si es el usuario en turno. En esta etapa se tendrán dos casos: el primero se presenta cuando la identificación del usuario es correcta, el robot esperará que el usuario vuelva a acercarse para continuar el recorrido. El segundo caso se presenta cuando la persona identificada no es la correcta y entonces el robot regresará a la etapa 1 para seguir buscando. En caso de no localizar al usuario en un periodo de tiempo de 5 minutos, el robot volverá a la recepción en busca de otro usuario.

3.3.3. Apariencia y mecanismo para la interacción

En el ámbito mecánico, la altura del robot Tbot se estableció en relación a las personas que normalmente visitan el área de cubículos, estudiantes y profesores en su mayoría (adultos). Se tomó como referencia la altura promedio de los mexicanos que oscila entre los 157 cm y 169 cm [129]. Debido a que el ángulo de apertura del sensor Kinect es de 47° , el campo vertical es de aproximadamente 63 cm por lo que se eligió una altura para Tbot de 120 cm colocando el Kinect superior de forma que el vector normal a la cámara tenga una inclinación de 23.5° hacia arriba. Esta posición se muestra en la Figura 3.5, a los 120 cm de la altura del robot se suman los 63 cm del campo de visión vertical. Esto permite tener un rango de visión para personas de estatura entre los 140 cm y 183 cm .

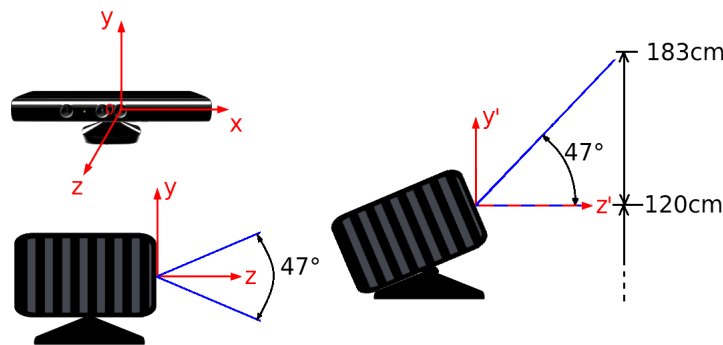


Figura 3.5: Kinect superior y altura del robot.

En la cima del robot se colocó una pantalla con una aplicación de selección de destino, la cual muestra en una lista los docentes que se ubican en ese piso, el usuario selecciona el docente al cual desea visitar y a continuación selecciona el botón de aceptar. Al momento de aceptar, la coordenada predeterminada se manda como punto meta al algoritmo de navegación para obtener la trayectoria global y comenzar el recorrido. En la Figura 3.6 se muestra la aplicación de selección de destino, además de mostrar un mapa con la ruta trazada hacia su destino.

La velocidad máxima de navegación es de 0.5 m/s para ambos robots con base a un estudio realizado en [48], donde establecen que un robot móvil con una velocidad de 1 m/s es incómodo

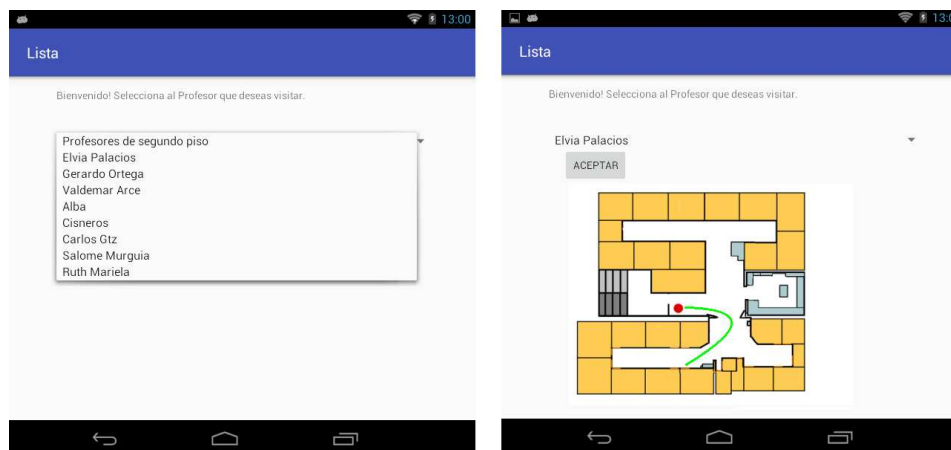


Figura 3.6: Aplicación de selección en la tablet.

para las personas. Y debido a que la velocidad máxima del robot Tbot es de 0.5 m/s se limitó la velocidad del robot UVerto para tener el mismo funcionamiento.

Además, se propuso introducir la velocidad del usuario al lazo de control para mantener cerca al usuario y evitar perderlo. La velocidad lineal del robot se controlará y ajustará a la velocidad del usuario, suavizada por las curvas de Bézier [4.8](#).

3.4. Control visual difuso

El movimiento del robot depende de la velocidad lineal y la velocidad angular. La velocidad lineal del robot a su vez, depende de la velocidad estimada del usuario a diferencia de la velocidad angular del robot que depende de la velocidad lineal del mismo y del tipo de trayectoria que debe realizar (recta, curva). Por lo tanto, para este trabajo de investigación, se decidió realizar dos lazos de control: un lazo interno para controlar la velocidad lineal del robot y un lazo externo para la velocidad angular.

Debido a que los controladores implementados utilizan información de los otros procesos que se realizan en forma paralela, todos los algoritmos utilizados deben tener la característica de ser rápidos, de fácil implementación y que trabajen con señales obtenidas de diferentes tareas.

Por lo tanto, para el control de movimiento, se utilizaron dos tipos de controladores difusos ya que tienen la ventaja de no necesitar el modelo matemático del sistema. El diagrama de bloques del controlador global (velocidad lineal, velocidad angular) se muestra en la Figura [3.7](#).

3.4.1. Control difuso

La lógica difusa se basa en modos de razonamiento aproximados, la importancia de esto se deriva del hecho que la mayoría de los modos de razonamiento humano y de sentido común son aproximados [130](#). En la lógica difusa: el razonamiento exacto es visto como un caso límite, cualquier sistema lógico puede ser “fuzzificado”, el conocimiento se interpreta como una colección de restricciones difusas equivalentes. La lógica difusa se diferencia de los sistemas lógicos tradicionales ya que en estos últimos, la respuesta puede tener sólo dos valores: verdadero o falso, y en la lógica difusa el valor puede ser un elemento dentro un conjunto finito. En un sistema tradicional los predicados son claros y en la lógica difusa son imprecisos. En los sistemas

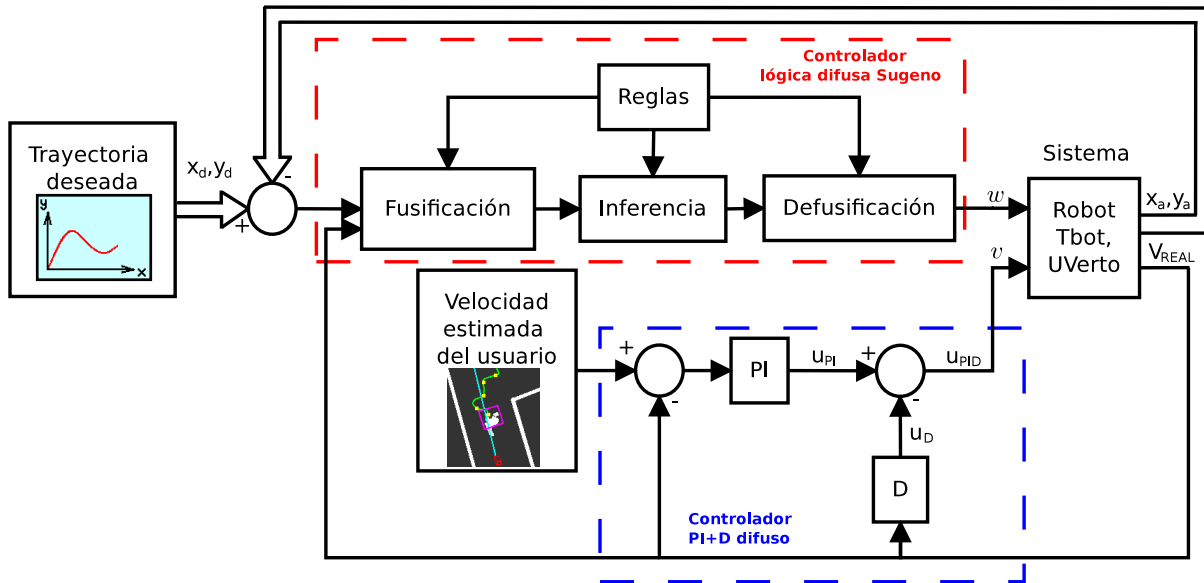


Figura 3.7: Esquema de control para las velocidades lineal y angular.

clásicos, la única forma de modificar un enunciado es con la negación. En la lógica difusa existe una variedad de modificaciones (más, menos, poco, mucho, etc.).

El control difuso utiliza expresiones lingüísticas para formular las reglas que controlarán un sistema. La lógica difusa sugiere un cierto grado de pertenencia para los datos permitiendo tomar diferentes grados de acción en un sistema, a esto se le llama inferencia. En la inferencia se genera un mapeo para asignar a una entrada una salida utilizando lógica difusa. Este proceso involucra la utilización de funciones de pertenencia y las reglas generadas en la base de conocimiento. Existen diferentes métodos de inferencia, los más comunes son los de Mamdani y Takagi Sugeno-Kang [131].

Como ya se mencionó se considerarán dos tipos de controladores difusos para este trabajo. Para el lazo interno de control se diseñó e implementó un PI+D difuso [132] para adaptar la velocidad del robot a la velocidad del usuario, ya que la velocidad lineal del robot además de ser similar a la del usuario debe de ser robusta a cambios imprescindibles en la entrada. Por otro lado, con este controlador se asegura que el error en estado estable sea mínimo, ya que este controlador tiene todas las grandes ventajas de un PID clásico pero con las características de un difuso. Este controlador tiene como entrada la velocidad estimada del usuario y como salida la velocidad lineal del robot.

Para la navegación del robot en un entorno conocido, se cuenta con una trayectoria global del punto inicial al punto final y una trayectoria local para la evasión de obstáculos. Debido a que el lazo interno controla la velocidad lineal, el control de la velocidad angular que regulará el seguimiento de trayectorias del robot se diseñó e implementó como un controlador difuso de tipo Sugeno. Este tiene como entrada la velocidad lineal del robot y el error en orientación.

3.4.2. Control PI+D difuso; control de velocidad lineal

El lazo interno de control: control PI+D difuso se muestra en la Figura 3.7 (sección azul). Donde el PI+D difuso utiliza un esquema de control discreto como el que se muestra en la

Figura 3.8. Este controlador, como su nombre lo indica es un PI (Proporcional- Integral) añadido con un D (Derivativo). Es decir, son dos controladores difusos por separado (PI,D). $K_I, K_P, K_D, K_{u,PI}, K_{u,D}$ son los parámetros ajustables de controlador, $r(nT)$ es la referencia, $\Delta u_{PI}(nT)$ es la salida del controlador PI difuso, $u_D(nT)$ es la salida del controlador D difuso y $u_{PID}(nT)$ es la salida global del controlador PI+D difuso.

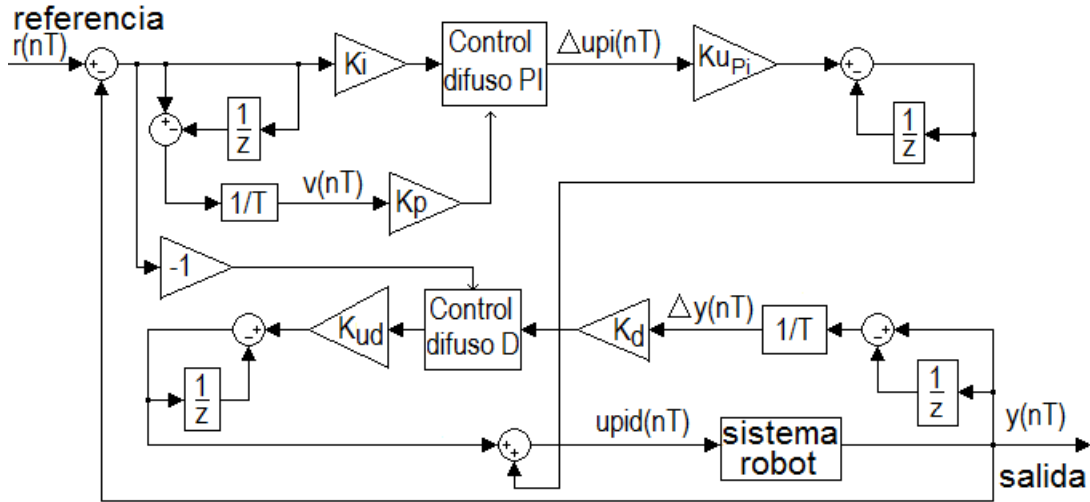


Figura 3.8: Esquema de control PI+D difuso.

Por lo tanto, la ley del controlador global PI+D difuso se presenta en 3.10.

$$\begin{aligned}
 u_{PID}(nT) &= u_{PI}(nT) - u_D(nT) \\
 &= u_{PI}(nT - T) + K_{u,PI}\Delta u_{PI}(nT) + u_D(nT - T) + K_{u,D}\Delta u_D(nT)
 \end{aligned}
 \tag{3.10}$$

Para determinar $\Delta_{PI}(nT)$ se introduce un parámetro constante $L > 0$ en el plano generado por $K_i e(nT)$ y $K_p v(nT)$ (error y la variación del error) como lo muestra la Figura 3.9, obteniéndose así 20 regiones ($R_i, i = 1...20$) de este plano, las cuales representan las combinaciones de estas dos señales de entrada ($K_i e(nT)$ y $K_p v(nT)$).

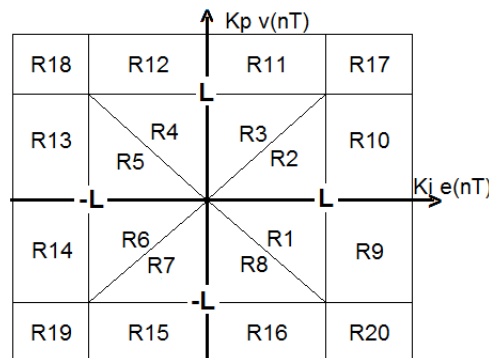


Figura 3.9: Regiones de combinación para el controlador PI.

Las funciones de pertenencia para las dos entradas (error $e(nT)$ y variación del error $v(nt)$)

y la salida (output negative o_n , output positive o_p , output zero o_z) se muestran en la Figura 3.10 y en las ecuaciones 3.11.

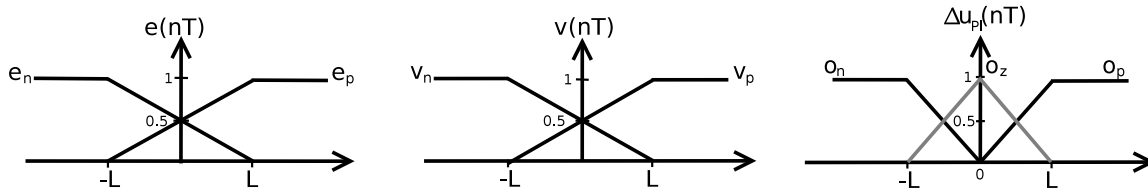


Figura 3.10: Funciones de pertenencia para las entradas y salida del control PI.

$$e_n = \frac{L - k_I e(nT)}{2L}, \quad e_p = \frac{L + k_I e(nT)}{2L}, \quad v_n = \frac{L - k_P v(nT)}{2L}, \quad v_p = \frac{L + k_P v(nT)}{2L}, \quad (3.11)$$

donde e_n sería el error negativo y e_p el error positivo, de igual forma v_n y v_p se define para la variación del error negativo y positivo, respectivamente.

Entonces, el comportamiento de un PI se establece en las siguientes reglas difusas, Ecuaciones 3.12.

$$\begin{aligned} R_1 : & \text{ Si } e = e_n \text{ Y } v = v_n \text{ entonces Salida}_{PI} = o_n \\ R_2 : & \text{ Si } e = e_n \text{ Y } v = v_p \text{ entonces Salida}_{PI} = o_z \\ R_3 : & \text{ Si } e = e_p \text{ Y } v = v_n \text{ entonces Salida}_{PI} = o_z \\ R_4 : & \text{ Si } e = e_p \text{ Y } v = v_p \text{ entonces Salida}_{PI} = o_p \end{aligned} \quad (3.12)$$

En donde $e = r - y$ es el error de velocidad lineal y $v = e = 0 - y = -y$ es la tasa de cambio del error. Las funciones de pertenencia de las entradas e_n, e_p, v_n, v_p son: error negativo, error positivo, derivada del error negativo y positivo, respectivamente, y o_n, o_z, o_p son la salida negativa, cero y positiva. Determinando la salida para cada región en la Figura 3.9, se obtienen las Ecuaciones 3.13 que en conjunto representan la salida del control PI ($\Delta u_{PI}(nT)$).

$$\begin{aligned} \Delta u_{PI}(nT) &= \frac{L[K_I e(nT) + K_P v(nT)]}{2(2L - K_I |e(nT)|)} \\ &= \frac{L[K_I e(nT) + K_P v(nT)]}{2(2L - K_P |v(nT)|)} \\ &= \frac{L[K_I e(nT) + K_P v(nT)]}{2(2L - K_I |e(nT)|)} \\ &= \frac{L[K_I e(nT) + K_P v(nT)]}{2(2L - K_P |e(nT)|)} \\ &= 0.5[L + K_P v(nT)] \end{aligned} \quad (3.13)$$

$$\begin{aligned}
&= 0.5[L + K_I e(nT)] \\
&= 0.5[-L + K_P v(nT)] \\
&= 0.5[-L + K_I e(nT)] \\
&= 0 \\
&= L \\
&= -L
\end{aligned}$$

Para la acción de control $\Delta u_D(nT)$ se realiza el mismo procedimiento que para el PI difuso, las entradas del controlador D difuso son $y_d(nT)$ (salida del sistema) y $\Delta y(nT)$ (señal negativa del error) con las combinaciones de entrada $K y_d(nT)$, $K_d \Delta y(nT)$. En la Figura 3.11, se muestran las regiones del control difuso D.

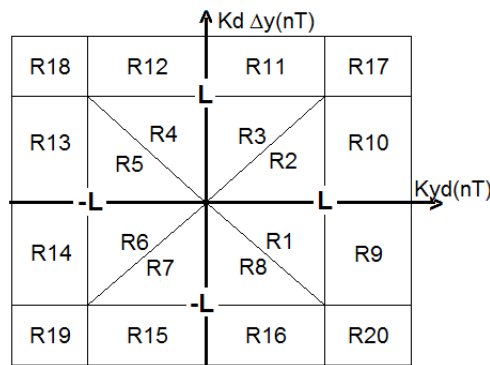


Figura 3.11: Regiones de combinación para el controlador D.

Las salidas de cada una de las regiones del plano de la Figura 3.11 para el controlador D difuso $\Delta u_D(nT)$, se muestran en las Ecuaciones 3.14.

$$\begin{aligned}
\Delta u_D(nT) &= \frac{L[K y_d(nT) - K_d |\Delta y(nT)|]}{2(2L - K |y_d(nT)|)} \\
&= \frac{L[K y_d(nT) - K_d |\Delta y(nT)|]}{2(2L + K |y_d(nT)|)} \\
&= 0.5[L - K_D \Delta y(nT)] \\
&= 0.5[-L + K y_d(nT)] \\
&= 0.5[-L - K y_d(nT)] \\
&= 0.5[L + K y_d(nT)] \\
&= 0 \\
&= L \\
&= -L
\end{aligned} \tag{3.14}$$

Para el ajuste de las ganancias, primero se realizó un controlador PID clásico y se ajustaron las ganancias K_P , K_I , K_D del controlador y con base en estas, se ajustaron las variables del controlador PI+D difuso. Finalmente, las variables se establecieron de la siguiente manera: $T = 0.15$, $K_{u,D} = T$, $K_{u,pi} = T$, $K = 0.9$, $L = 10$, $K_p = 0.5$, $K_d = 0.3$, $K_i = 0.0005$.

La ley de control global PI+D difuso [3.10](#), determinará la velocidad lineal del robot, la cual estará limitada por la velocidad máxima del robot.

3.4.3. Control difuso tipo sugeno para seguimiento de trayectoria

Para el seguimiento de trayectoria se considera el lazo externo de control, el cual modificará la velocidad angular para el ajuste de orientación del robot. El controlador está basado en la lógica difusa tipo Sugeno, las entradas del controlador son: la velocidad lineal obtenida en el controlador anterior ($\Delta u_{PID}(nT)$) y el error entre la orientación deseada y la actual. El esquema de control de este controlador se muestra en la Figura [3.12](#). Donde v y w son la velocidad lineal y angular que controlarán al robot, (x, y, ϕ) es la posición y orientación del robot, (x_d, y_d) es la posición deseada y θ es el error entre la orientación deseada y la actual del robot.

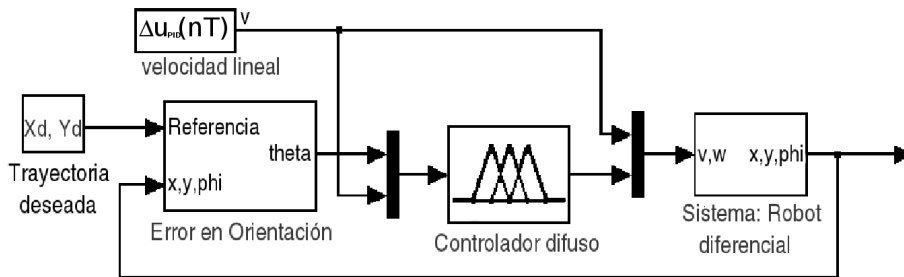


Figura 3.12: Esquema de control para la velocidad angular.

Los valores de pertenencia de las variables de entrada se presentan en la Figura [3.13](#). Los valores de salida de la variable w son Singletons con los siguientes valores: $N_a = -1$, $N_1 = -0.55$, $N_2 = -0.25$, $Z = 0$, $P_2 = 0.25$, $P_1 = 0.55$, $P_a = 1$.

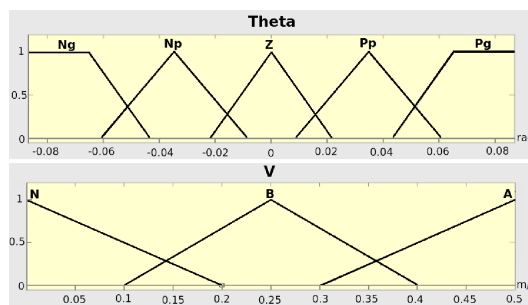


Figura 3.13: Funciones de pertenencia de las variables de entrada θ y v .

Las reglas del controlador difuso se presentan en la Tabla [3.1](#).

Ambos controladores se diseñaron para que, de manera conjunta, ajusten las velocidades del robot tanto lineal como angular a los diferentes objetivos. Con esto se logra dirigir al robot a un punto dentro del mapa para el seguimiento de trayectorias, a una velocidad estimada del usuario, como parte del método de HRI implementado.

Tabla 3.1: Reglas del controlador Sugeno para w .

v/θ	N_g	N_p	Z	P_p	P_g
N	Z	Z	Z	Z	Z
B	N_a	P_1	Z	N_1	P_a
A	N_2	Z	Z	Z	P_1

3.5. Supervisor

Como se mencionó en el Capítulo anterior, el supervisor se encarga de dirigir las acciones del robot con base en un procedimiento y una asignación de prioridades. Todas las actividades descritas en este Capítulo se llevan a cabo de forma paralela, en la Figura 3.14 se muestra la operación del supervisor en un diagrama de flujo estableciendo así, un orden y prioridad en la realización de las tareas.

El funcionamiento del robot se inicia con la localización del robot en un mapa local, mientras que se encuentra inmóvil en la recepción (estado de espera). El robot inicia la búsqueda de rostros en su campo de visión, una vez que detectó un rostro se establece una comunicación con el posible usuario, cuando la persona solicita los servicios del robot se identifica y localiza al usuario. Luego de recibir el destino deseado se ejecutan los algoritmos de navegación, planeación de trayectorias y estimación de velocidad del usuario además se mantiene una comunicación con el usuario a lo largo del recorrido por medio del algoritmo de control del habla.

Mientras la posición del usuario sea localizada dentro de los parámetros de proxémica establecidos para la HRI, el robot continuará con el seguimiento de trayectoria. Si el usuario sale de esta zona (1.5 m) el robot comenzará con la búsqueda de usuario. El proceso de búsqueda consiste en los siguientes 3 pasos:

- El robot girará sobre su eje para localizar en las cercanías el color predominante del usuario.
- El robot se acercará a esta zona para realizar el reconocimiento de rostro.
- Si la persona localizada es el usuario, el robot volverá a planear las trayectorias hacia el destino desde la posición actual de este. Si la persona localizada no es el usuario, el robot seguirá girando para ubicar una persona que corresponda a las características del usuario.

Finalmente, cuando se llega al destino se emite un mensaje de despedida y se traza una nueva trayectoria para volver al origen.

El objetivo principal para esta tarea de robot guía es no perder al usuario de vista y llevarlo a su objetivo, siendo este el de prioridad más grande del supervisor.

3.6. Conclusiones

En este Capítulo se presentan los algoritmos implementados para la aplicación de robot móvil guía basado en la interacción humano-robot.

Para la localización del usuario, se adaptaron diferentes métodos de visión utilizados en la identificación y ubicación de objetos. Este método se compone de tres algoritmos principales enunciados a continuación:

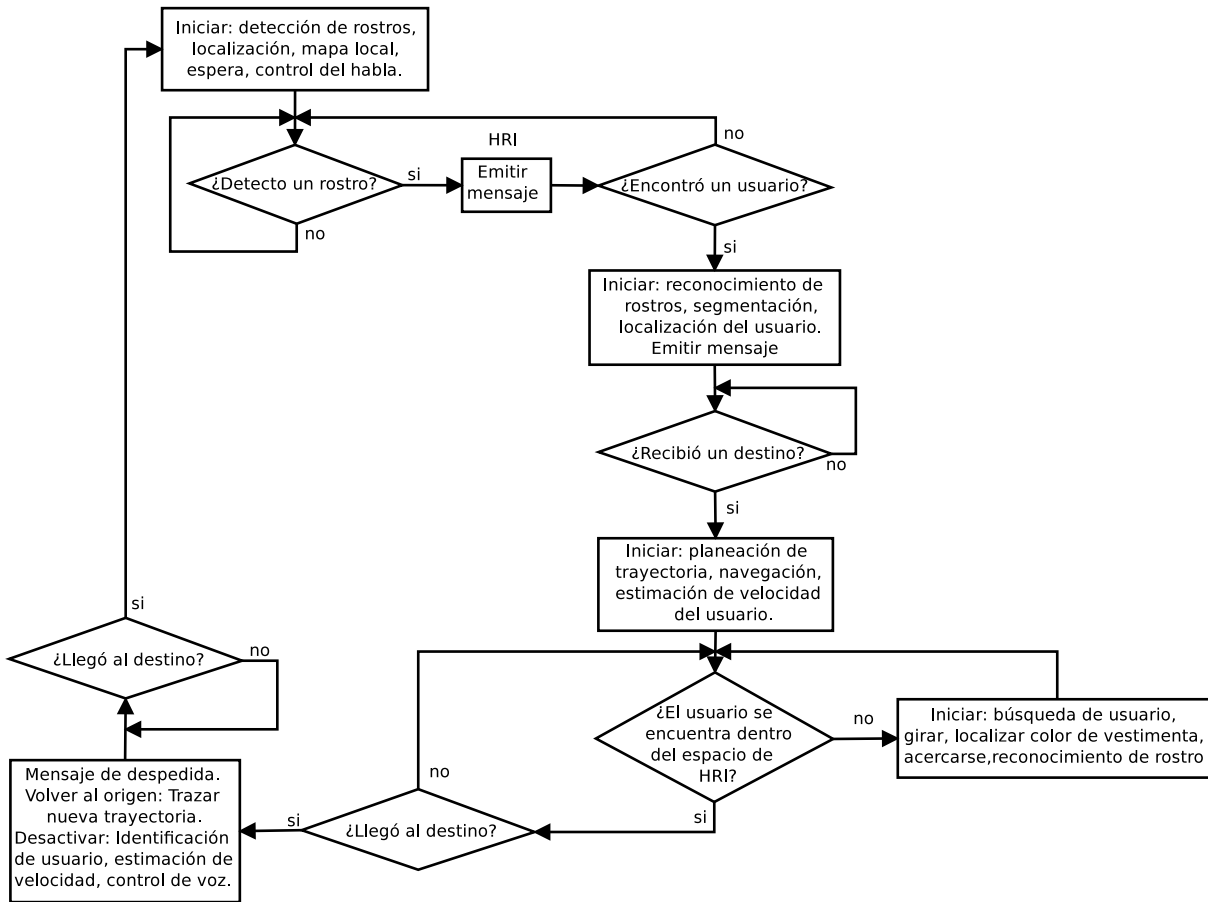


Figura 3.14: Supervisor.

- Detección y validación de rostros.
- Clasificación y segmentación por color.
- Segmentación por distancia y localización en un espacio tridimensional.

Y uno adicional que realiza reconocimiento de rostro para saber si el usuario sigue dentro del área de visión.

Además, con la información de los algoritmos de visión para la detección de usuario, se estima la velocidad del usuario, esta información se convierte en una señal suavizada que servirá como referencia para el control de velocidad lineal del robot.

Se diseñaron e implementaron dos controladores que trabajan de manera simultánea, uno para el seguimiento de velocidades del usuario las cuales se estimaron a partir de un algoritmo de visión (controlador difuso PI+D). El segundo controlador está basado en la lógica difusa tipo Sugeno para el control de la velocidad angular el cual dará la orientación necesaria para el seguimiento de trayectorias. Con el control de ambas velocidades, el robot debe seguir una trayectoria deseada con la velocidad estimada del usuario. Las pruebas se realizaron a nivel simulación y de manera experimental, los resultados obtenidos se muestran en el siguiente Capítulo.

Capítulo 4

Experimentación y resultados

En este Capítulo se presentan los resultados obtenidos de cada algoritmo adaptado o diseñado implementado para realizar la tarea de robot guía. Los resultados se muestran de manera individual en cada sección y después se expone el resultado final de todos los algoritmos funcionando en conjunto para la aplicación final de guía de personas.

4.1. Resultados del algoritmo de detección y localización de usuario.

Para la localización del usuario, se diseñó un algoritmo de visión que identifica al usuario por el rostro con el algoritmo de Viola y Jones [12], donde para evitar falsos positivos, valida la detección de un rostro con el color de su piel [13]. Después, el usuario es caracterizado en función del color predominante en su vestimenta en el espacio de color HSV. Posteriormente, es segmentado del resto de la imagen y ubicado en un espacio tridimensional gracias a la información de profundidad que proporciona el Kinect en la parte superior del robot [14]. En la Figura 4.1, se muestran las etapas del algoritmo de detección.

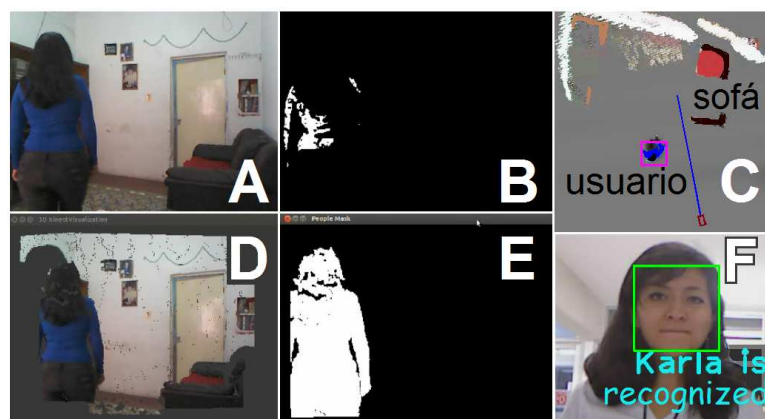


Figura 4.1: Algoritmo de localización de usuario. A. Imagen real, B. Segmentación por color (azul), C. Localización del usuario en 3D, D. Nube de puntos, E. Segmentación por distancia, F. Reconocimiento de rostro.

La detección de las personas se realiza al inicio de la aplicación de guía, cuando el robot está en la recepción y en espera de un usuario. El algoritmo de detección comprueba si existe algún rostro en el área de visión, si detecta uno, el robot comenzará la interacción invitando al posible usuario a utilizar sus servicios. El usuario que desee utilizar el servicio deberá colocarse frente al robot y utilizar las opciones de una pantalla, esto permitirá que el algoritmo de visión detecte y almacene el color predominante del usuario que será la persona más cercana al robot. El algoritmo de detección de rostros presenta un funcionamiento a una frecuencia promedio de $10Hz$.

Por medio de una toma de muestras se obtuvo que el algoritmo de detección de rostros tiene un porcentaje de detección acertado de 94.4% , donde el porcentaje total corresponde a las muestras detectadas como posibles rostros. Los errores que presenta el algoritmo se refieren a la detección de un posible rostro, se tienen tres clasificaciones: falso positivo (2.4%), falso negativo (3.2%) y errores correctos (1.9%). Donde falso positivo significa que se realizó una detección de rostro donde no existe, se obtiene un falso negativo cuando la detección de un posible rostro se clasifica como errónea pero es correcta y finalmente la detección de errores correctos es cuando el algoritmo descarta la ubicación de un posible rostro cuando este no existe. El porcentaje de los errores correctos se considera parte del porcentaje de detección acertado. Después, se ejecuta el algoritmo de clasificación por color dominante de la vestimenta del usuario, esto permite que el proceso de localización sea más rápido al enfocarse en el color y no en las características de un rostro. Se realizaron pruebas de color para localizar los rangos de los valores de HSV que obtienen la mejor clasificación para el usuario, donde el rango de matiz (H) varía 10 puntos de una escala total de 180, el rango de saturación (S) varía 100 unidades de 256 y el rango de brillo o luz (V) varía 90 puntos de 256. Para este último valor se realizaron además experimentos con diferentes intensidades de luz. También se obtuvieron los colores mejor detectados por el algoritmo los cuales se muestra una gama de colores en la Figura 4.2.



Figura 4.2: Colores que mejor distingue el algoritmo de visión de clasificación HSV.

Además, se confirma la posición del usuario con una segmentación realizada por el proceso de crecimiento de regiones, en la que el criterio de región de pertenencia es la profundidad del objeto. Mediante los algoritmos anteriores, se obtiene la posición del usuario en 2D (plano del piso), el algoritmo de localización se basa en esta información para ubicar los datos de profundidad dada la posición del usuario y obtener su ubicación en un espacio tridimensional, este algoritmo tiene una frecuencia de muestreo de $25 Hz$. El porcentaje de acierto final del conjunto de algoritmos de visión para la detección y localización de usuario es de 85.9% .

La Figura 4.3 muestra como el algoritmo distingue la cara de la persona y la señala con un recuadro verde, incluso si la persona gira un poco el rostro, el algoritmo sigue detectando, también puede reconocer distintos tonos de piel.

Después de detectar el rostro se busca una región de importancia (zona del pecho) por medio de una máscara en la imagen, se aplica un histograma en esta región para obtener el color (en el espacio de HSV) de mayor predominancia y así localizar al usuario incluso si el rostro no es ubicado (ver Figura 4.4). Las características deseadas en la ropa que debe usar el usuario para una eficiente clasificación por color son: ropa con color distinto al fondo y con uno o dos

4.1. RESULTADOS DEL ALGORITMO DE DETECCIÓN Y LOCALIZACIÓN DE USUARIO.51

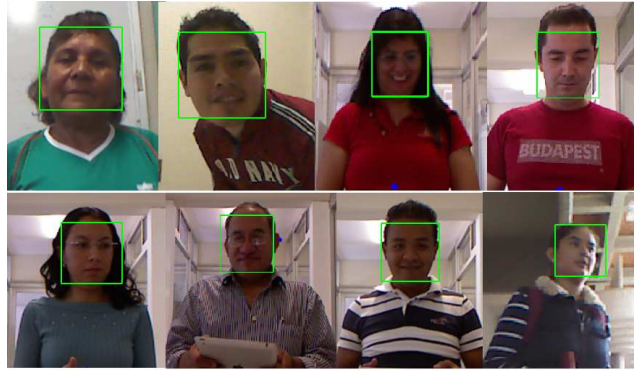


Figura 4.3: Detección de rostros distintos tonos de piel y posiciones.

colores dominantes, las texturas en la ropa pueden ser pequeñas. Otro tipo de vestimenta puede aumentar el error en la detección de posición del usuario y por lo tanto su velocidad.



Figura 4.4: Clasificación por color.

En la fase de segmentación, se toma la posición del usuario, en un espacio de dos dimensiones, obtenido con el algoritmo de detección de rostro junto a la clasificación de color, entonces con el método de región de crecimiento es posible hacer la reconstrucción del cuerpo completo y separarlo del fondo. Las tres partes del algoritmo mejoran la localización del usuario y se muestran en la Figura [4.5](#).

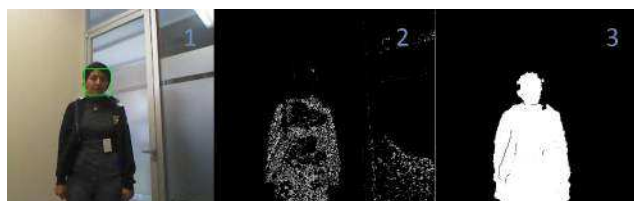


Figura 4.5: Ubicación del usuario.

Con la información de profundidad que el sensor ofrece, se puede conocer la posición de la persona en un espacio de tres dimensiones. La Figura [4.6](#) señala con un cuadro color magenta

donde se localiza el usuario, la línea color cyan es la referencia (ortogonal al plano de visión) y el cubo rojo representa el sensor.

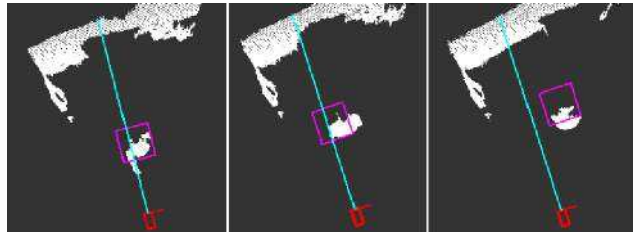


Figura 4.6: Localización de la persona en un espacio de 3D.

El algoritmo de detección de rostros localiza a los usuarios en el área de visión, asignando el título de “usuario” al primer usuario detectado o al más cercano, en caso de existir más de una persona al iniciar la aplicación. Es decir, en caso de existir otra persona en el campo de visión el usuario será distinguido, como se muestra en la Figura 4.7.



Figura 4.7: Ubicación del usuario en existencia de más personas.

4.2. Estudio sobre la marcha humana

Un spline cúbico es una función polinómica segmentaria de grado 3 con primera y segunda derivada continuas [133].

Un método clásico de interpolación es el polinomio de Lagrange. El término general de la fórmula de Lagrange puede escribirse de la siguiente forma:

$$P(x) = \sum_{i=1}^n y_i \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j} \quad (4.1)$$

donde (x,y) es la posición del usuario en el espacio.

El spline cúbico elimina dos problemas de la interpolación lineal: su lentitud de convergencia y el que presenta ángulos en los nodos, lo cual es inadecuado en ciertas aplicaciones.

Debido a que el seguimiento de trayectorias considera que el robot debe adaptarse a la trayectoria del usuario, modificándola a su conveniencia, se realiza una estimación de la trayectoria que seguirá, por medio de splines cúbicos.

Si se aproxima la siguiente función mediante el polinomio de Lagrange y mediante splines cúbicos se puede observar los resultados en la Figura 4.8 donde la interpolación de Lagrange se muestra en la parte superior y los splines cúbicos en la parte inferior, con 4, 5 y 10 puntos de izquierda a derecha.

$$y = \frac{12}{x^2 + 2x + 5} \quad (4.2)$$

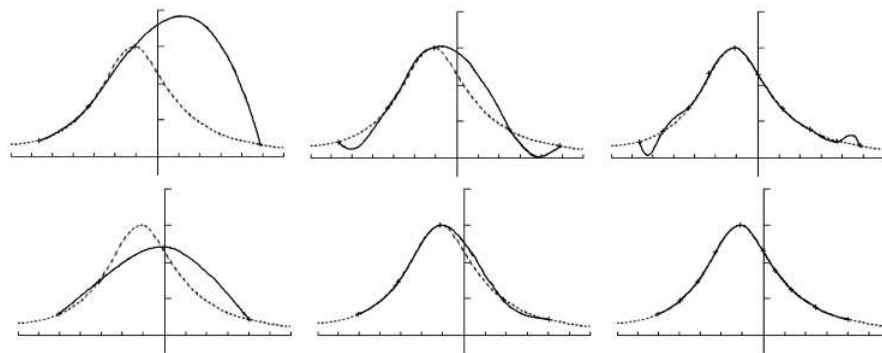


Figura 4.8: Interpolación de Lagrange (superior) y splines cúbicos (inferior), a 4,5 y 10 puntos [134].

El método de Lagrange con 10 puntos presenta ondulaciones lejos del máximo de la función, a este fenómeno se le denomina fenómeno de Runge [135], aumentando su actividad cuando se incrementa el número de puntos. El método de splines cúbicos muestra un mejor comportamiento con solo 5 puntos y con 10 es muy cercana a la función original.

El algoritmo para obtener los splines dada una función es el siguiente. Dados $n + 1$ puntos (x_k, y_k) , $k = 1, 2, \dots, n + 1$, con $x_1 < x_2 < \dots < x_{n+1}$, tales que:

$$s(x) = q_k(x), \quad x \in [x_k, x_{k+1}], \quad k = 1, 2, \dots, n \quad (4.3)$$

que verifiquen las siguientes condiciones de interpolación

$$\begin{aligned} q_k(x_k) &= y_k, & k &= 1, 2, \dots, n \\ q_k(x_{k+1}) &= y_{k+1}, & k &= 1, 2, \dots, n \end{aligned} \quad (4.4)$$

y también las condiciones de conexión

$$\begin{aligned} q'_k(x_{k+1}) &= q'_{k+1}(x_{k+1}), & k &= 1, 2, \dots, n - 1 \\ q''_k(x_{k+1}) &= q''_{k+1}(x_{k+1}), & k &= 1, 2, \dots, n - 1 \end{aligned} \quad (4.5)$$

Las primeras condiciones (4.4) establecen que s es continua e interpola los puntos (x_k, y_k) . Las segundas (4.5) implican que s tiene primera y segunda derivadas continuas [133]. Se dice entonces que $s(x)$ es spline interpolador para P_0, P_1, \dots, P_n . Denotando con $h_k = x_{k+1} - x_k$ para $k = 0, 1, \dots, n - 1$ y $\sigma_k = s''(x_k)$ para $k = 0, 1, \dots, n$. se tiene:

$$\begin{aligned} q_k(x) &= \frac{\sigma_k}{6} \left[\frac{(x_{k+1} - x)^3}{h_k} - h_k(x_{k+1} - x) \right] + \frac{\sigma_{k+1}}{6} \left[\frac{(x - x_k)^3}{h_k} - h_k(x - x_k) \right] \\ &+ y_k \left[\frac{x_{k+1} - x}{h_k} \right] + y_{k+1} \left[\frac{x - x_k}{h_k} \right] \quad \text{para } k = 0, 1, \dots, n - 1 \end{aligned} \quad (4.6)$$

que es la ecuación del spline $q_k(x)$.

Para obtener $\sigma_0, \sigma_1, \dots, \sigma_n$ se debe resolver la siguiente ecuación:

$$h_{k+1}\sigma_{k-1} + 2(h_{k-1} + h_k)\sigma_k + h_k\sigma_{k+1} = 6 \left(\frac{y_{k+1} - y_k}{h_k} - \frac{y_k - y_{k+1}}{h_{k+1}} \right) \text{ para } k = 1, \dots, n - 1 \quad (4.7)$$

Se realizó un estudio sobre la marcha humana donde se tomaron los datos de cinco personas al caminar por un pasillo para obtener su trayectoria. Para la primera estimación, se obtuvo la posición del usuario en un espacio tridimensional, cada que el usuario recorría aproximadamente 1 m de distancia. Los valores utilizados se muestran en la Tabla 4.1, donde el plano formado por los ejes X y Z representan el suelo (Z es la distancia del sensor al usuario).

Persona 1		Persona 2		Persona 3	
Z	X	Z	X	Z	X
1.15	0.4698	1.873	-0.1373	0.683	0.0722
2.101	0.5982	2.167	0.6666	1.69	-0.2108
3.341	-0.2004	3.073	0.5531	2.69	-0.0589
4.296	-0.4377	3.821	-0.1492	3.583	0.1262
5.282	-0.2867	4.838	0.0046	4.463	0.0807
6.124	0.5074	5.915	0.1295	5.364	-0.0051

Tabla 4.1: Posiciones con separación de 1 m .

En las primeras gráficas de la Figura 4.9 A, se muestra el recorrido de las personas y la trayectoria estimada a partir de los valores anteriores, se hace una comparación con un mayor número de datos (12 posiciones) para determinar qué cantidad de muestras obtenidas logran una mejor estimación de trayectoria y velocidad en tiempo real.

Las segundas gráficas de la Figura 4.9 B, muestra la trayectoria estimada para 6 posiciones en el mismo recorrido del usuario. La separación entre cada posición es de aproximadamente 0.5 m y se muestran en la Tabla 4.2.

Persona 1		Persona 2		Persona 3	
Z	X	Z	X	Z	X
0.991	0.05946	1.873	-0.1373	0.683	0.0722
1.15	0.4698	1.958	0.2137	1.36	-0.1651
1.603	0.6549	2.167	0.6666	1.69	-0.2108
2.101	0.5982	2.629	0.8087	2.38	-0.1459
3.341	-0.2004	3.073	0.5531	2.69	-0.0589
3.864	-0.3496	3.373	0.1317	3.1	0.0864
4.296	-0.4377	3.821	-0.1492	3.583	0.1262
4.771	-0.4134	4.35	-0.2692	4.01	0.097
5.282	-0.2867	4.838	0.0046	4.463	0.0807
5.626	-0.0053	5.364	0.1685	4.85	0.0201
6.018	0.1776	5.915	0.1295	5.364	-0.0051
6.124	0.5074	6.124	0.1224	5.364	-0.0051

Tabla 4.2: Posiciones con separación de 0.5 m .

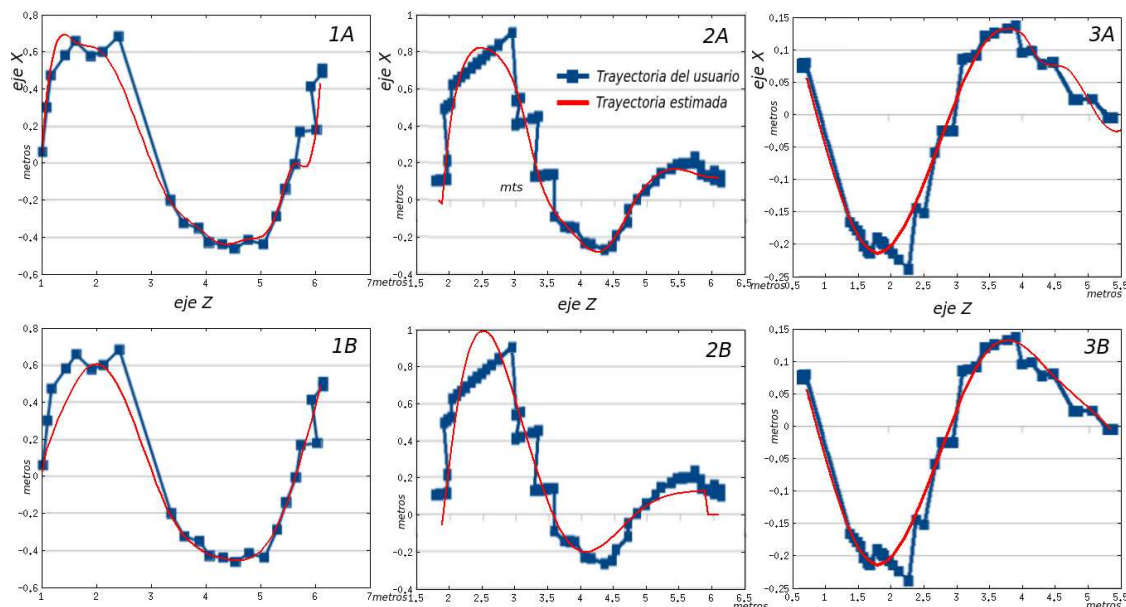


Figura 4.9: Comparación de aproximación de la trayectoria con 12 y 6 datos de posición.

4.2.1. Estimación de la velocidad del usuario

Gracias a los algoritmos de visión, es posible localizar al usuario dentro del área de trabajo del sensor Kinect. La posición del usuario se obtiene con respecto al marco referencial del sensor, debido a que el robot estará en movimiento, este punto no será referencia para obtener la velocidad del usuario. Entonces se debe de hacer una transformación de la posición del usuario al marco referencial del mapa. Una vez que se obtiene el punto de localización del usuario dentro del mapa, se procede a realizar el cálculo de velocidad el cual se obtiene de dividir la distancia recorrida en intervalos de 0.5 s . La velocidad estimada presenta variaciones bruscas debido a que las personas no realizan una marcha lineal [100]. La velocidad calculada del usuario se suaviza con las curvas cúbicas de Bézier [110] Ecuación (4.8).

$$B(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, \quad t \in [0, 1] \quad (4.8)$$

En la Figura 4.10A se observa la velocidad estimada del usuario en línea punteada y la velocidad obtenida a través de la ecuación cúbica de Bézier se presenta línea continua. En la Figura 4.10B se muestra la trayectoria que realizó el usuario en un pasillo recto, para esta velocidad. La velocidad obtenida se limitará con base en la velocidad máxima del robot, es decir, se ajustará a 0.4 m/s .

4.2.2. Algoritmo del supervisor

El supervisor se encarga de conocer de manera permanente la ubicación del usuario, la cual es la variable de mayor prioridad para esta aplicación. Por medio de interrupciones, ejecuta una pausa en el algoritmo de control que se encarga de seguir una trayectoria, para realizar la tarea de búsqueda de usuario.

Antes de comenzar el recorrido, el robot clasifica al usuario por color y almacena una base de datos con su rostro, la cual le permitirá reconocer a la persona que solicitó el servicio en

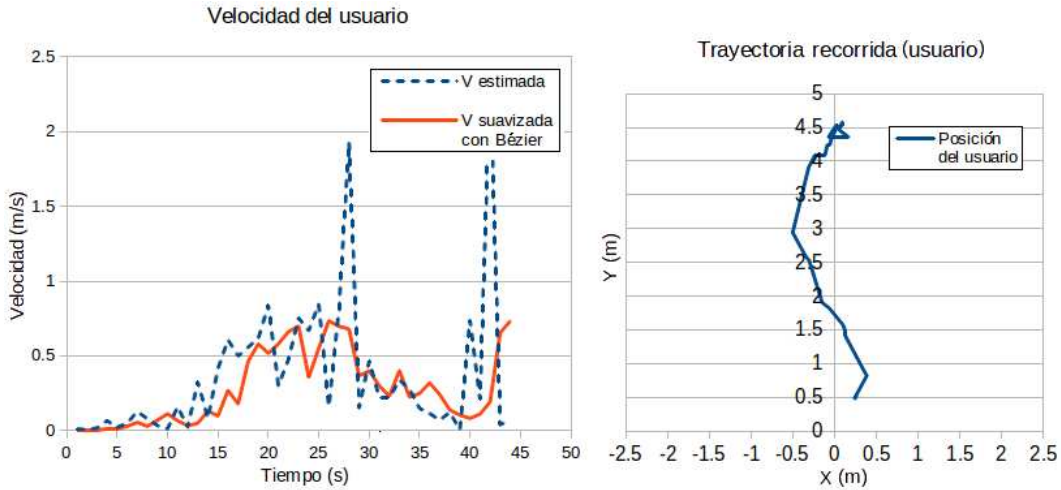


Figura 4.10: A. Velocidad con ec. cúbica de Bézier. B. Trayectoria realizada por el usuario.

caso de pérdida. Una vez localizado el usuario la interrupción se desactiva y se vuelve a trazar una trayectoria al mismo punto meta.

El supervisor seguirá observando la separación entre el usuario y el robot a lo largo del recorrido para ejercer el plan de acción correspondiente y no perder al usuario.

En la Figura 4.11 se muestran los procesos y acciones que realiza el robot en la ejecución de la tarea de guía. También se muestra la información más relevante, que se obtiene en cada proceso.

Acción del robot	Procesos			Información
Estado de espera	Detección de rostros	Interacción HR	Localización en el mapa	Posición de origen
Reconocimiento de usuario	Reconocimiento de rostro	Segmentación por color	Localización de usuario	Rostro Color del usuario Posición del usuario
Navegación	Cálculo de trayectorias		Evasión de obstáculos	Posición actual y Posición deseada del robot
Estimación	Estimación de velocidad y ubicación del usuario		Suavizar señal de velocidad	Velocidad del usuario Localización del usuario en el mapa
Control de velocidad	Controlador PI+D (velocidad lineal)	Controlador Sugeno (velocidad angular)		Velocidad lineal Velocidad angular
Interrupción	Búsqueda de usuario, criterio (separación entre robot y usuario)			Interrupción verdadero/falso

Figura 4.11: Procesos del robot.

El pseudocódigo de este algoritmo se muestra a continuación:

Algoritmo 1 Supervisor.

```

1: Cargar mapa y localizarse dentro de él.
2: Iniciar algoritmos [2, 3, 4, 5].
3: Iniciar HRI.
4: if Detecta rostro = True then
5:     Esperar instrucción del usuario para iniciar recorrido.
6:     if Recibe instrucción= True then
7:         Iniciar algoritmos de localización y reconocimiento de usuario ([6, 7, 8]).
8:         while Usuario dentro de espacio HRI=True and Destino =False do
9:             Ejecutar algoritmos para la navegación ([9,10, 12, 13]).
10:        end while
11:        Ejecutar algoritmos de búsqueda ([10, 14]).
12:    end if
13: end if

```

4.2.3. Experimentos y resultados de la navegación autónoma en la tarea de robot guía

A continuación se explica todo el proceso que realiza el robot para ejecutar la tarea final de guía de personas y después se muestran los resultados obtenidos. El proceso comienza con la detección de las personas, la cual se realiza al inicio de la aplicación de guía, cuando el robot está en la recepción y en espera de un usuario. El algoritmo de detección comprueba si existe algún rostro en el área de visión, si detecta uno, el robot comenzará la interacción invitando al posible usuario a utilizar sus servicios. El usuario que desee utilizar el servicio deberá colocarse frente al robot y utilizar las opciones de una pantalla, esto permite que el algoritmo de visión detecte y almacene el color predominante del usuario que será la persona más cercana al robot.

Después se realiza la segmentación por color dominante de la vestimenta del usuario, esto permite que el proceso de localización sea más rápido al enfocarse en el color y no en las características de un rostro. Además, se confirma la posición del usuario con una segmentación realizada por el proceso de crecimiento de regiones, en la que el criterio de región de pertenencia es la profundidad del objeto. Mediante los algoritmos anteriores, se obtiene la posición del usuario en 2D (plano del piso), el algoritmo de localización se basa en esta información para ubicar los datos de profundidad dada la posición del usuario y obtener su ubicación en un espacio tridimensional.

Una vez hecha la solicitud, el robot comienza su trayecto a una velocidad de $0.4m/s$ para posteriormente ajustarse a la del usuario. La velocidad del usuario es calculada con base en la distancia recorrida por este en un intervalo de tiempo. Para el cálculo de la distancia recorrida, la posición del usuario se transforma del marco referencial de la cámara, al marco referencial del mapa. La señal que entra al controlador debe ser suavizada para que el robot no presente cambios bruscos de velocidad que pongan en riesgo la integridad de las personas y de él mismo.

La navegación se realiza con el cálculo de trayectorias, para esto, se cuenta con la información del punto inicial y el mapa de su entorno, así como la ubicación dentro del mapa de los lugares a los que el usuario puede dirigirse, por ejemplo, las oficinas de los docentes. El punto inicial que se tiene como referencia (home) será la recepción, el robot regresará a esta posición cada que termine un recorrido. La elección del punto final se dará por el usuario en una aplicación de Android que muestre la lista de los lugares disponibles.

Se genera una trayectoria global del punto inicial al punto final, esta trayectoria se calcula realizando un cuadrículado en el mapa y definiendo las celdas libres por las que puede pasar el robot. También se calcula una trayectoria local para la evasión de obstáculos la cual consiste en conocer la velocidad lineal y angular del robot, así como la proximidad de obstáculos, del punto meta y de la trayectoria global. Con esta información, realiza una simulación de la posición próxima del robot al cabo de un corto periodo de tiempo, para predecir si el robot colisionará. Estas trayectorias se calculan en tiempo real y se estarán actualizando de acuerdo a la posición actual del robot y la información del entorno, por lo que no necesita conocer la posición de los obstáculos en el mapa.

Las funciones mencionadas en la Sección 3.2 proveen al robot la información necesaria para navegar en un mapa conocido. Si se proporciona un punto meta al algoritmo, este calcula una trayectoria global y una local además de la velocidad lineal y angular a lo largo del recorrido. Para introducir el controlador difuso global se debe omitir la información de las velocidades que proporciona el paquete, sin embargo, se utilizarán las trayectorias (global y local) calculadas por este para la navegación. En la Figura 4.12, se muestran las trayectorias calculadas por los algoritmos de navegación (global y local). Donde la trayectoria global marca el camino de inicio a fin y la trayectoria local se encarga de evitar obstáculos en el recorrido.

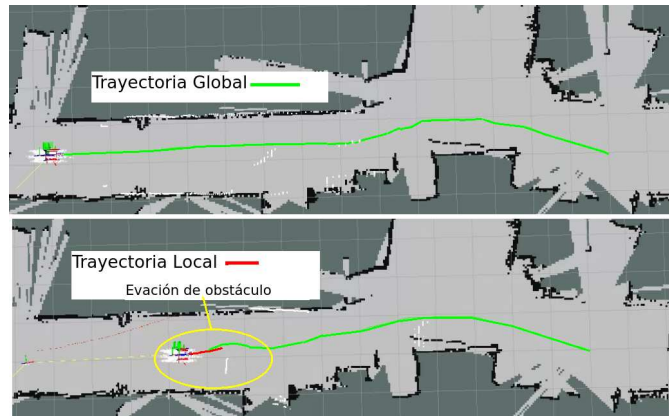


Figura 4.12: Navegación reactiva de un robot móvil en un mapa.

El control de ambas variables (velocidad lineal y angular) necesita de tres referencias las cuales son: la velocidad estimada del usuario suavizada con Bézier, la trayectoria deseada y la posición actual del robot. Las pruebas de los controladores funcionando juntos se realizaron primero en un simulador y luego se implementaron en las plataformas robóticas Tbot y UVerto. Para la implementación de los controladores en el mundo real, debe considerarse el entorno del robot (mapa), el retardo en la comunicación entre las diferentes computadoras, la frecuencia en el intercambio de mensajes y la respuesta de cada robot.

Las primeras pruebas de navegación se realizaron sin considerar la estimación de velocidad del usuario, esto con el objetivo de comparar la importancia de la localización y velocidad del usuario dentro del lazo de control. Con el estudio de la marcha humana, se obtuvo un promedio de la velocidad de una persona al seguir el robot a su velocidad máxima ($0.5m/s$), este promedio se fijó en ($0.3m/s$), a lo largo del recorrido la velocidad varía dadas las reglas de interacción provenientes del estudio de la proxémica. La Figura 4.13 muestra en la gráfica de lado derecho un plano bidimensional (X, Y) con las trayectorias que realizaron el usuario, el robot y la trayectoria deseada. De lado izquierdo se muestra nuevamente la distancia de

separación entre el usuario y el robot, y la velocidad lineal del robot.

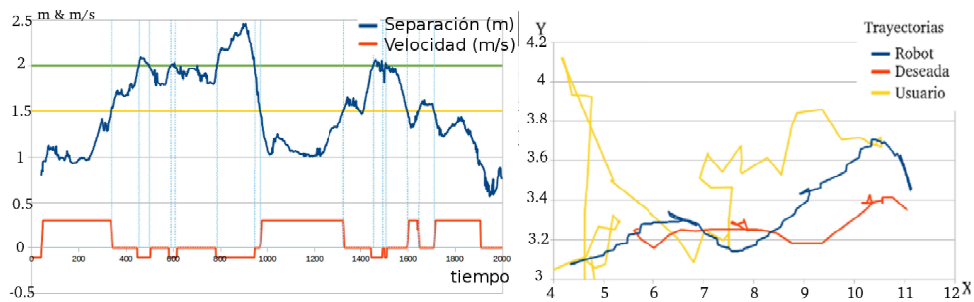


Figura 4.13: Posición XY de las trayectorias seguidas a lo largo del trayecto.

Debajo de la línea amarilla se especifica el rango de la distancia social que debe existir entre el robot y el usuario. Entre líneas verde y amarillo es el segundo rango de operación el robot se detiene para esperar al usuario. Como se observa, la distancia entre ambos es muy variable y sale constantemente de los márgenes especificados. Incluso las trayectorias del robot y usuario, difieren contra la deseada. En esta gráfica se muestra también el comportamiento de la estrategia de búsqueda. Cuando la distancia entre el robot y el usuario sale de los límites de distancia establecidos por la proxémica (por arriba de la línea verde), el robot retrocede. Sin embargo, a pesar de los inconvenientes, el robot logra llevar al usuario a la posición deseada.

En la Figura 4.14 se muestra la distancia del usuario así como la velocidad del robot para una trayectoria con curva, se mantienen las mismas especificaciones que la imagen anterior.

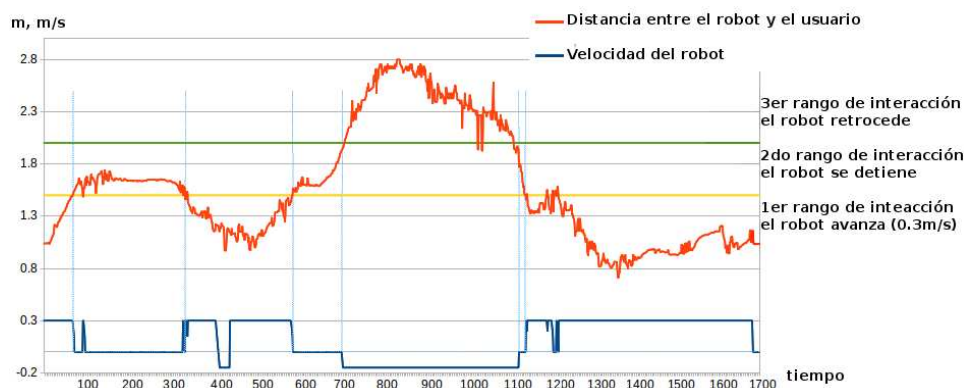


Figura 4.14: Distancia entre el usuario y el robot comparada con la velocidad lineal del robot.

En la Figura 4.15 se puede observar la visión del robot por medio del Kinect superior, la toma desde el marco referencial del mundo y la posición en el mapa del robot y del usuario.

Para el segundo experimento, se establecieron trayectorias rectas y sin obstáculos. Para el robot Tbot se realizó una trayectoria de 4 m a lo largo de un pasillo recto. En la Figura 4.16A, se muestra la referencia de velocidad en línea punteada y la salida del sistema (velocidad lineal del robot) en línea continua. El seguimiento de trayectoria se presenta en la Figura 4.16B, la trayectoria deseada en línea punteada larga, la trayectoria realizada por el robot en línea punteada corta y la trayectoria seguida por el usuario se muestra en línea continua.

Para el experimento final, la trayectoria que es más larga y compleja que en el experimento anterior, presenta obstáculos a lo largo del trayecto. En la Figura 4.17 se muestran las señales

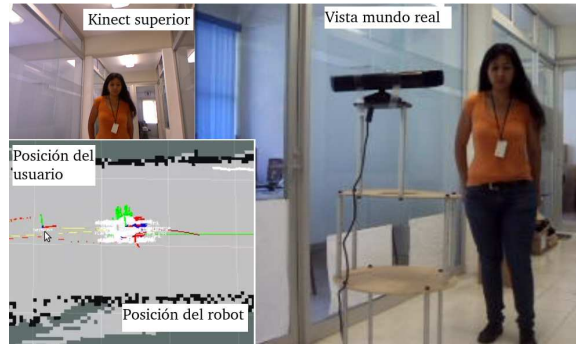


Figura 4.15: Entorno real del experimento.

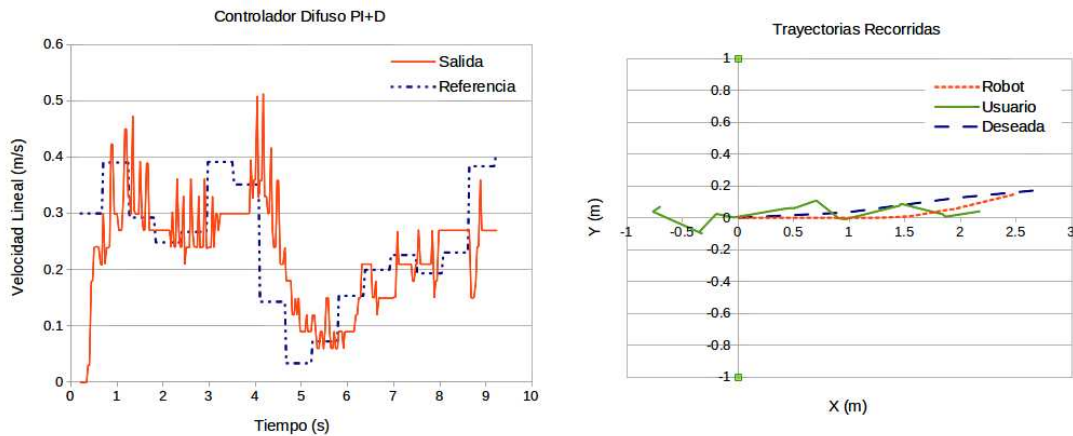


Figura 4.16: A. Seguimiento de velocidad Tbot, B. Trayectorias realizadas.

del controlador para las pruebas sobre la plataforma robótica, en línea punteada se muestra la señal de referencia y en línea continua la salida del sistema (velocidad lineal).

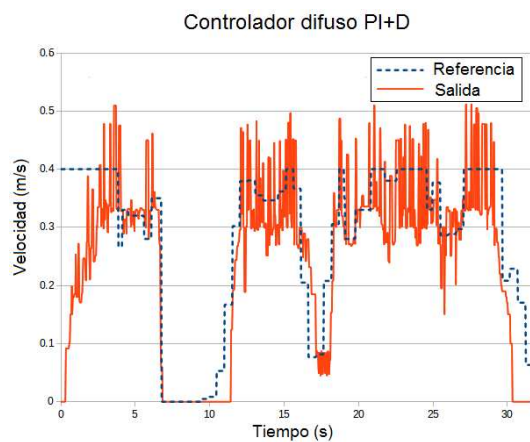


Figura 4.17: Seguimiento de velocidad del usuario en la plataforma móvil.

El controlador difuso para la velocidad angular, se ajusta dada la velocidad lineal y el error en orientación (orientación deseada menos orientación actual del robot). En la Figura 4.18, se muestra la trayectoria deseada en línea punteada y la realizada por el robot en línea continua.

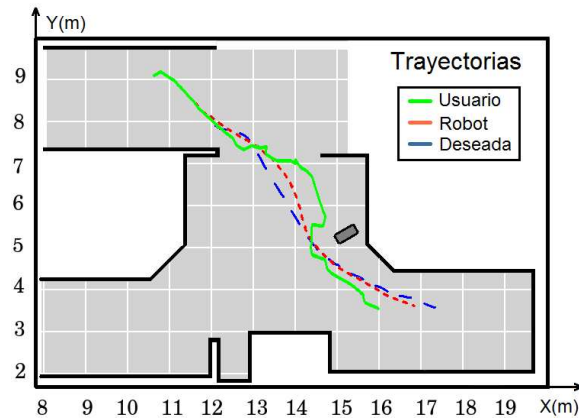


Figura 4.18: Trayectoria deseada vs trayectoria realizada.

En la Figura 4.19 se muestra el mapa donde se realizaron las pruebas y la visualización del robot, así como una imagen real del entorno.

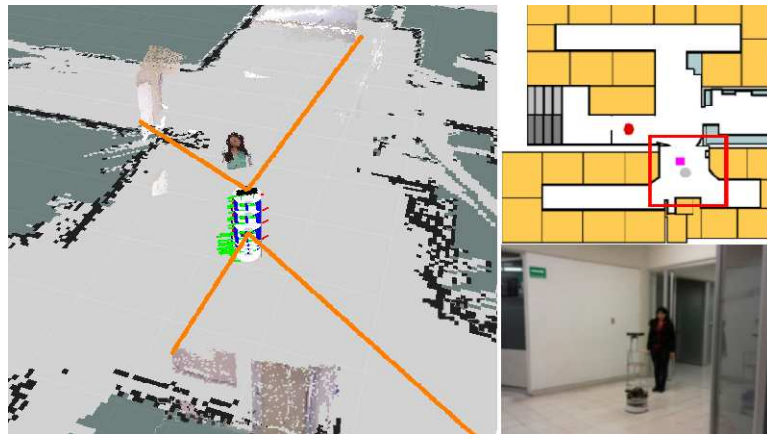


Figura 4.19: Entorno real.

Cuando el robot se aleja del usuario (más de 3 m) por más de 3 s, se interrumpe la operación de guía y comienza el estado de búsqueda.

4.3. Pruebas con UVerto

También se realizaron pruebas de compatibilidad de algoritmos entre los robots Tbot y UVerto de la UV. Se adaptaron los algoritmos implementados en el Tbot de seguimiento de trayectorias y el de seguimiento de usuario por distancia a la plataforma UVerto y se verificó el funcionamiento de los algoritmos en ambas plataformas. En la Figura 4.20 se muestra el recorrido que realizó el robot UVerto con el algoritmo de seguimiento de trayectoria. El algoritmo de seguimiento de usuario se muestra en la Figura 4.21 donde el robot sigue a un usuario en particular dado un criterio de distancia, la parte derecha de la imagen presenta la simulación de los datos obtenidos por medio de la cámara para una reconstrucción del espacio donde el robot se encuentra.

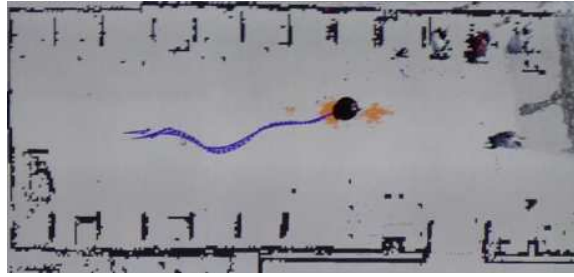


Figura 4.20: Seguimiento de trayectoria en plataforma UVerto.

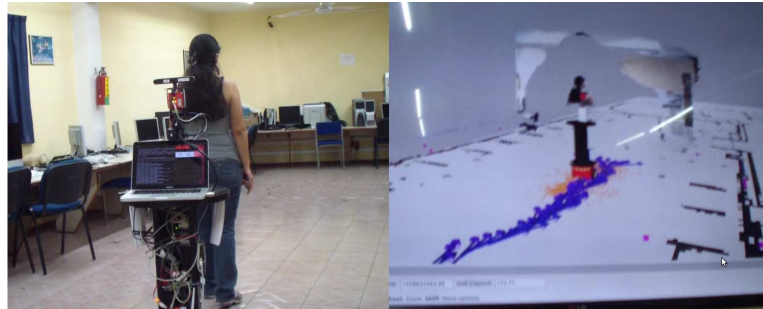


Figura 4.21: Seguimiento de usuario en plataforma UVerto.

Para el robot UVerto, se realizó una trayectoria recta de 6 m para los experimentos finales. En la Figura 4.22A, se muestra la referencia de velocidad en línea punteada y la salida del sistema (velocidad lineal del robot) en línea continua. El seguimiento de trayectoria realizado por el robot UVerto, se presenta en la Figura 4.22B. La trayectoria deseada en línea punteada larga, la trayectoria realizada por el robot en línea punteada corta y la trayectoria seguida por el usuario se muestra en línea continua.

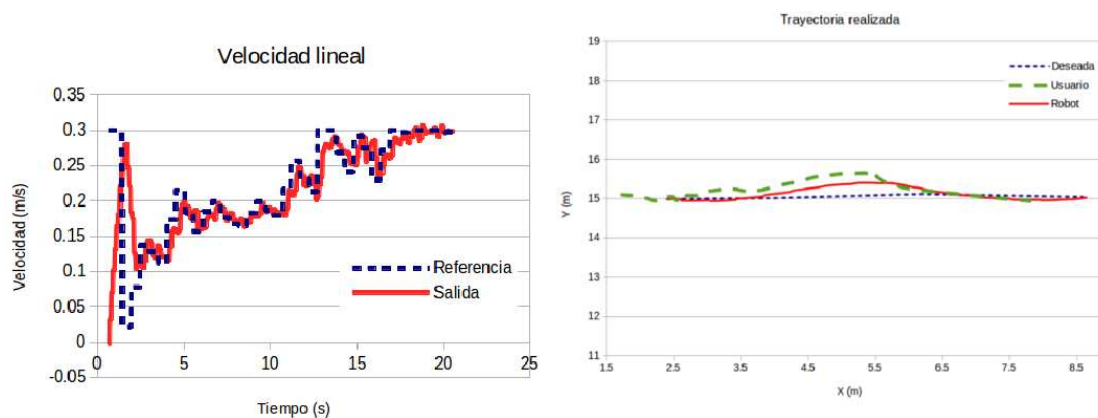


Figura 4.22: A. Seguimiento de velocidad en UVerto, B. Trayectorias realizadas.

La separación del usuario y el robot UVerto, se muestra en la Figura 4.23. Se puede observar que con la implementación del controlador que ajusta la velocidad del robot a la referencia de la velocidad estimada del usuario. La distancia de separación entre el robot y el usuario se mantiene con una media de 0.9 m y no se sale de los límites de interacción (0.5-1.5 m).

También se realizaron pruebas con obstáculos, donde la trayectoria es dinámica ya que

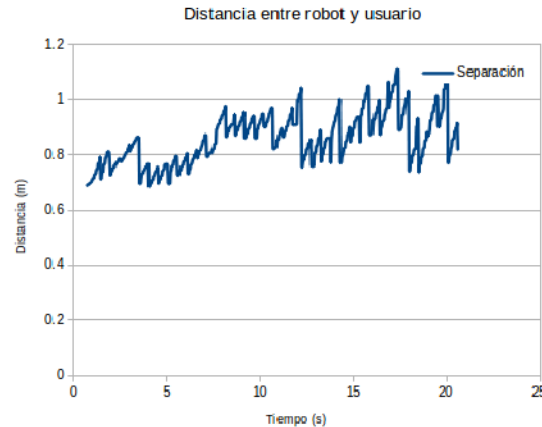


Figura 4.23: Distancia de separación entre el robot UVerto y el usuario.

cambia si es obstruida por algún objeto en su recorrido. En la Figura 4.24A, se puede observar la velocidad lineal que siguió el robot y el usuario en una trayectoria con obstáculos. En la Figura 4.24B se presentan las trayectorias realizadas por el robot y el usuario, así como la trayectoria deseada.

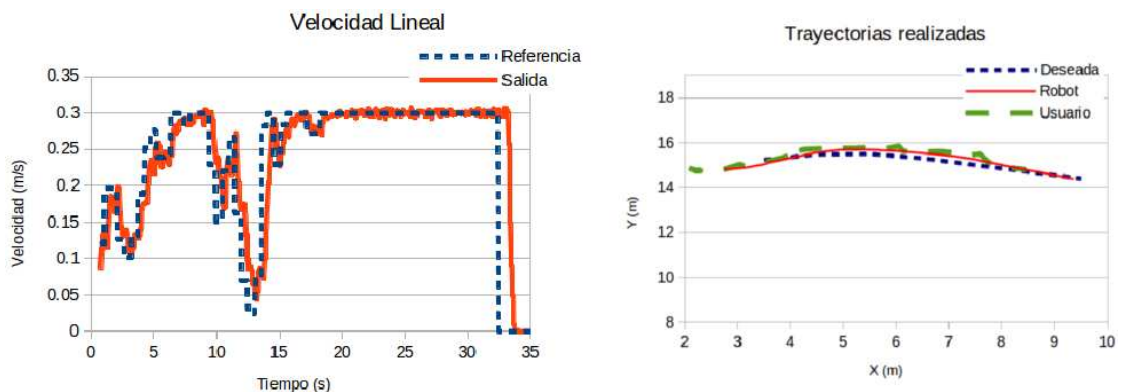


Figura 4.24: A. V lineal para trayectoria con obstáculos en UVerto. B. Trayectorias realizadas.

4.3.1. Conclusiones

El diseño e implementación de distintos algoritmos permiten al robot realizar una tarea de guía de personas. Para brindar un mejor servicio a los usuarios, el robot no debe perderlos durante el trayecto. Las herramientas utilizadas para el desarrollo de esta actividad, comprenden desde visión, navegación, interacción, control, entre otros.

La navegación puede evadir obstáculos y mantener cercano a su usuario, en caso de que el usuario salga del área de visión del robot, por medio de un supervisor que evalúa constantemente la separación entre el usuario y el robot, se iniciará un plan de búsqueda para localizar al usuario y después continuar con la trayectoria al punto deseado. Por lo que se cumple el objetivo general de la tarea de guía de personas basado en la HRI.

Conclusiones

La finalidad principal de este trabajo de investigación fue de desarrollar un robot guía prototipo de propósito general que realice la tarea de guía de personas en un ambiente semi-estructurado que mejore la principal desventaja que los robots guía actualmente tienen, la cual es que no modifican su velocidad a la del usuario. Si los robots guía no consideran la información de la velocidad del usuario podrían pasar varios escenarios no deseables en la tarea de guía, es decir, pérdida completa del usuario ya sea por:

- Decisión propia del usuario.
- Distracción del usuario.
- Diferente velocidad a la programada por el robot.
- Falta de paciencia del usuario, aburrimiento.

La contribución principal de este trabajo de investigación es la inclusión de la velocidad del humano dentro del lazo de control del movimiento del robot guía. Estimar la velocidad del usuario en tiempo real y retroalimentarla dentro del lazo de control de la velocidad lineal del robot, permite mejorar considerablemente la tarea de guía de personas en varios aspectos:

- Asegura que el objetivo se cumpla.
- Evita incidentes con las personas.
- Mejora la interacción humano-robot.
- Respeta normas sociales de interacción.
- Debido a que el robot tiene la misma velocidad que el usuario y gracias a la proxémica, es un compañero de trayecto.

Para realizar esto, el movimiento del robot es controlado tomando en cuenta la información del medio ambiente, la auto-información, las reglas establecidas para la interacción y la velocidad (presencia) del usuario.

En este trabajo se presenta la acción de un robot móvil guía basado en la interacción humano-robot. Se diseñaron e implementaron dos controladores difusos que trabajan de manera simultánea, uno para el seguimiento de velocidades del usuario (PI+D difuso), las cuales se estimaron a partir de un algoritmo de visión, donde se hace reconocido de personas en datos

RGB-D. El segundo controlador (Sugeno) está basado en la lógica difusa para el control de la velocidad angular el cual dará la orientación necesaria para el seguimiento de trayectorias. Con el control de ambas velocidades, el robot debe seguir una trayectoria deseada con la velocidad estimada del usuario. Todos los algoritmos necesarios para realizar la tarea final de guía de personas se implementaron de forma satisfactoria en un sistema distribuido (no lineal) de comunicación hardware - software (ROS).

Las pruebas se realizaron a nivel simulación y de manera experimental. Los resultados obtenidos muestran que el desempeño de los controladores permite al robot realizar un seguimiento de trayectorias con una modificación de su velocidad a lo largo del recorrido. La adaptación de velocidad del robot brindará un mejor servicio al usuario, ya que incrementará las probabilidades de que el usuario lo siga y ambos lleguen a su objetivo. Este sistema puede implementarse para robots guía en distintas áreas como museos, hospitales, universidades, etc., así como en el cuidado de adultos mayores, ya que la velocidad de marcha se reduce con los años. También puede implementarse en aplicaciones de rescate de personas donde el robot debe guiar al usuario a la salida, sin que este se pierda en el camino.

Bibliografía

- [1] I. Horswill. Polly: A vision-based artificial agent. In *AAAI*, pages 1999–2005. Proceedings of the 11th Conference of the American Association for Artificial Intelligence, 1993.
- [2] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 11–18, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.
- [3] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence, ELSEVIER*, 114(1-2):3–55, Oct 1999.
- [4] W. Burgard A. B. Cremers F. Dellaert D. Fox D. Hähnel C. Rosenberg N. Roy J. Schulte D. Schulz S. Thrun, M. Bennewitz. Minerva: A tour-guide robot that learns. In *Advances in Artificial Intelligence. KI 1999. Lecture Notes in Computer Science*, pages 144–149, Berlin, Heidelberg, Germany, 1999. Springer.
- [5] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B. Cremers, Frank Dellaert, Dieter Fox, Dirk Hähnel, Charles R. Rosenberg, Nicholas Roy, Jamieson Schulte, and Dirk Schulz. Minerva: A second-generation museum tour-guide robot. In *ICRA*, pages 1999–2005. IEEE Robotics and Automation Society, 1999.
- [6] S. Grange R. Lutz R. Meyer I. Nourbakhsh, J. Bobenage and A. Soto. An affective mobile educator with a full-time job. *Artificial Intelligence, ELSEVIER*, 114(1-2):95–124, 1999.
- [7] D. Rodriguez-Losada, F. Matia, R. Galan, and A. Jimenez. Blacky, an interactive mobile robot at a trade fair. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 4, pages 3930–3935, 2002.
- [8] R. D. Schraft, C. Schaeffer, and T. May. Care-o-bottm: the concept of a system for assisting elderly or disabled persons in home environments. In *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE*, volume 4, pages 2476–2481 vol.4, Aug 1998.
- [9] C. Schaeffer and T. May. Care-o-bot: A system for assisting elderly or disabled persons in home environments. In *Association for the Advancement of Assistive Technology in Europe AAATE-99*, 1999.

- [10] B. Jensen, G. Froidevaux, X. Greppin, A. Lorotte, L. Mayor, M. Meisser, G. Ramel, and R. Siegwart. The interactive autonomous mobile system roblox. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1221–1227 vol.2, 2002.
- [11] B. Jensen, N. Tomatis, L. Mayor, A. Drygajlo, and R. Siegwart. Robots meet humans-interaction in public spaces. *IEEE Transactions on Industrial Electronics*, 52(6):1530–1546, Dec 2005.
- [12] Gunhee Kim, Woojin Chung, Kyung-Rock Kim, Munsang Kim, Sangmok Han, and R. H. Shinn. The autonomous tour-guide robot jinny. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 4, pages 3450–3455, Sept 2004.
- [13] S. Kim, J. Choi, and M. Kim. Interactive software architecture for service robots. In *2006 SICE-ICASE International Joint Conference*, pages 4270–4275, Oct 2006.
- [14] P. Trahanias, W. Burgard, A. Argyros, D. Hahnel, H. Baltzakis, P. Pfaff, and C. Stachniss. Tourbot and webfair: Web-operated mobile robots for tele-presence in populated exhibitions. *IEEE Robotics Automation Magazine*, 12(2):77–89, June 2005.
- [15] A. Clodic, S. Fleury, R. Alami, M. Herrb, and R. Chatila. Supervision and interaction. In *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pages 725–732, July 2005.
- [16] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita. Interactive humanoid robots for a science museum. *IEEE Intelligent Systems*, 22(2):25–32, March 2007.
- [17] Eduardo Zalama Jaime Gómez-García-Bermejo Joaquín López, Diego Pérez. Bellbot - a hotel assistant system using mobile robots. *International Journal of Advanced Robotic Systems*, 10(40), 2013.
- [18] Stephen Sherbrook Johnathan Beckwith, Robert Lefief and Matthew Williams. Cate: Central automated tour experience. In *Proceedings of the 2012 American Society for Engineering Education (ASEE) North-Central Section Conference*, pages 1–12, 2012.
- [19] M. G. Rashed, R. Suzuki, A. Lam, Y. Kobayashi, and Y. Kuno. A vision based guide robot system: Initiating proactive social human robot interaction in museum scenarios. In *2015 International Conference on Computer and Information Engineering (ICCIIE)*, pages 5–8, Nov 2015.
- [20] M. Golam Rashed, R. Suzuki, A. Lam, Y. Kobayashi, and Y. Kuno. Toward museum guide robots proactively initiating interaction with humans. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts, HRI'15 Extended Abstracts*, pages 1–2, 2015.
- [21] J. MacDougall and G. S. Tewolde. Tour guide robot using wireless based localization. In *IEEE International Conference on Electro-Information Technology , EIT 2013*, pages 1–6, May 2013.

- [22] Kaname Sano, Keisuke Murata, Ryota Suzuki, Yoshinori Kuno, Daijiro Itagaki, and Yoshinori Kobayashi. Museum guide robot by considering static and dynamic gaze expressions to communicate with visitors. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, HRI'15 Extended Abstracts, pages 125–126, New York, NY, USA, 2015. ACM.
- [23] Yoshinori Kuno, Kazuhisa Sadazuka, Michie Kawashima, Keiichi Yamazaki, Akiko Yamazaki, and Hideaki Kuzuoka. Museum guide robot based on sociological interaction analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1191–1194, 2007.
- [24] K. Yasui, U. Masashi, W. Quan, X. Wu, and H. Furuhashi. One-to-one audio guidance system using human vision, designed for a guide robot. In *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 841–842, Dec 2015.
- [25] Daphne E. Karreman, Geke D.S. Ludden, Elisabeth M.A.G. van Dijk, and Vanessa Evers. How can a tour guide robot's orientation influence visitors' orientation and formations? In *Fourth International Symposium on New Frontiers in Human-Robot Interaction, HRI-AISB 2015*, Canterbury, UK, April 2015. University of Kent.
- [26] Daphne Karreman, Geke Ludden, and Vanessa Evers. *Visiting Cultural Heritage with a Tour Guide Robot: A User Evaluation Study in-the-Wild*, pages 317–326. Springer International Publishing, Cham, 2015.
- [27] Alpha Daye Diallo, Suresh Gobee, and Vickneswari Durairajah. Autonomous tour guide robot using embedded system control. *Procedia Computer Science*, 76:126 – 133, 2015.
- [28] S. J. Lee, J. Lim, G. Tewolde, and J. Kwon. Autonomous tour guide robot by using ultrasonic range sensors and qr code recognition in indoor environment. In *IEEE International Conference on Electro/Information Technology*, pages 410–415, June 2014.
- [29] H. M. Gross and H. J. Boehme. Perses-a vision-based interactive mobile shopping assistant. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 1, pages 80–85, 2000.
- [30] H. M. Gross, H. J. Boehme, C. Schroeter, S. Mueller, A. Koenig, C. Martin, M. Merten, and A. Bley. Shopbot: Progress in developing an interactive mobile shopping assistant for everyday use. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 3471–3478, Oct 2008.
- [31] H. M. Gross, A. Koenig, H. J. Boehme, and C. Schroeter. Vision-based monte carlo self-localization for a mobile service robot acting as shopping assistant in a home store. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 256–262 vol.1, 2002.
- [32] H. M. Gross, A. Koenig, C. Schroeter, and H. J. Boehme. Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 2, pages 1505–1511 vol.2, Oct 2003.

- [33] Kouhei Baba Seiji Sugiyama and Tsuneo Yoshikawa. Guide robot with personal identification method using dress color information via kinect. *Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics*, pages 2195–2200, 2012.
- [34] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita. A larger audience, please! encouraging people to listen to a guide robot. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 31–38, March 2010.
- [35] M. A. Yousuf, Y. Kobayashi, Y. Kuno, K. Yamazaki, and A. Yamazaki. Establishment of spatial formation by a mobile guide robot. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 281–282, March 2012.
- [36] Mohammad Abu Yousuf, Yoshinori Kobayashi, Yoshinori Kuno, Akiko Yamazaki, and Keiichi Yamazaki. *Development of a Mobile Museum Guide Robot That Can Configure Spatial Formation with Visitors*, pages 423–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [37] A. Bellarbi, S. Kahlouche, N. Achour, and N. Ouadah. A social planning and navigation for tour-guide robot in human environment. In *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, pages 622–627, Nov 2016.
- [38] A. Al-Wazzan, R. Al-Farhan, F. Al-Ali, and M. El-Abd. Tour-guide robot. In *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*, pages 1–5, March 2016.
- [39] Felix Faber, Maren Bennewitz, Clemens Eppner, Attila Görög, Christoph Gonsior, Dominik Joho, Michael Schreiber, and Sven Behnke. The humanoid museum tour guide Robotinho. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 891–896, Toyama, Japan, September 2009.
- [40] M. Nieuwenhuisen, J. Gaspers, O. Tischler, and S. Behnke. Intuitive multimodal interaction and predictable behavior for the museum tour guide robot robotinho. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 653–658, Dec 2010.
- [41] N. Roy S. Thrun M. Montemerlo, J. Pineau and V. Verma. In *Proceedings of the AAAI National Conference on Artificial Intelligence*.
- [42] Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore, Hayley Hung, Omar A. Islas Ramírez, Michiel Joosse, Harmish Khambhaita, Tomasz Kucner, Bastian Leibe, Achim J. Lilienthal, Timm Linder, Manja Lohse, Martin Magnusson, Billy Okal, Luigi Palmieri, Umer Rafi, Marieke van Rooij, and Lu Zhang. *SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports*, pages 607–622. Springer International Publishing, Cham, 2016.
- [43] Michiel Joosse and Vanessa Evers. A guide robot at the airport: First impressions. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, HRI '17*, pages 149–150, New York, NY, USA, 2017. ACM.

- [44] R. Stricker, S. Müller, E. Einhorn, C. Schröter, M. Volkhardt, K. Debes, and H. M. Gross. Interactive mobile robots guiding visitors in a university building. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 695–700, Sept 2012.
- [45] Masahide Kaneko Bin Zhanga, Tomoaki Nakamura. A framework for adaptive motion control of autonomous sociable guide robot. *IEEJ Transactions on Electrical and Electronic Engineering*, 11:786 – 795, 2016.
- [46] Kazushi Nakazawa, Keita Takahashi, and Masahide Kaneko. Movement control of accompanying robot based on artificial potential field adapted to dynamic environments. *Electrical Engineering in Japan*, 192(1):25–35, 2015.
- [47] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271, December 1959.
- [48] John Travis Butler and Arvin Agah. Psychological effects of behavior patterns of a mobile personal robot. *Autonomous Robots*, 10(2):185–202, Mar 2001.
- [49] Víctor Alvarez-Santos, A. Canedo-Rodriguez, Roberto Iglesias, Xosé M. Pardo, and C. V. Regueiro. *Route Learning and Reproduction in a Tour-Guide Robot*, pages 112–121. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [50] Dieter Fox, Wolfram Burgard, Sebastian Thrun, and Armin Cremers. Position estimation for mobile robots in dynamic environments. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.
- [51] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21 – 71, 1998.
- [52] D. Fox, W. Burgard, S. Thrun, and A. B. Cremers. A hybrid collision avoidance method for mobile robots. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 2, pages 1238–1243 vol.2, May 1998.
- [53] Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, Takayuki Kanda, and Ryohei Nakatsu. Robovie: an interactive humanoid robot. *Industrial Robot: An International Journal*, 28(6):498–504, 2001.
- [54] Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández, and Antonio Marín Hernández. Detección y seguimiento de personas con análisis de color en datos rgb-d. In *XVI Congreso Latinoamericano de Control Automático (CLCA 2014)*, pages 630–635, Oct 2014.
- [55] Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández, and Antonio Marín Hernández. Una propuesta de arquitectura para el control de un robot guía. In *XI Semana Nacional de Ingeniería Electrónica (SENIE 2015)*, number 112, pages 1155–1171, Nov 2015.

- [56] K. L. Luna-Gallegos, E. R. Palacios-Hernandez, S. Hernandez-Mendez, and A. Marin-Hernandez. A proposed software architecture for controlling a service robot. In *2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6, Nov 2015.
- [57] Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández, and Antonio Marín Hernández. Diseño e implementación de un robot guía basado en la interacción humano-robot. In *Congreso Nacional de Control Automático (AMCA 2016)*., Sep 2016.
- [58] Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández, and Antonio Marín Hernández. Análisis del primer contacto entre un robot asistente didáctico y niños de educación básica. In *Congreso Nacional de Control Automático (AMCA 2017)*., Oct 2017.
- [59] Kinect for windows sensor components and specifications (2018). <https://msdn.microsoft.com/en-us/library/jj131033.aspx>. Accesado: 2018-02-09.
- [60] Robot operating system. <http://www.willowgarage.com/pages/software/ros-platform>. Accessed: 2017-04-10.
- [61] Héctor M. Becerra and Carlos Sagüés. *Visual Control of Wheeled Mobile Robots: Unifying Vision and Control in Generic Approaches*. Springer Publishing Company, Incorporated, 1st edition, 2014.
- [62] Edgar Leonardo Rodríguez Vásquez Leonardo Enrique Solaque Guzmán, Manuel Alejandro Molina Villa. Seguimiento de trayectorias con un robot móvil de configuración diferencial. In *Ingenierías USBMed*, volume 5, pages 26–34, Ene-Jun 2014.
- [63] Y. Masutani, M. Mikawa, N. Maru, and F. Miyazaki. Visual servoing for non-holonomic mobile robots. In *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, volume 2, pages 1133–1140 vol.2, Sep 1994.
- [64] K. Hashimoto and T. Noritsugu. Visual servoing of nonholonomic cart. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1719–1724 vol.2, Apr 1997.
- [65] Hong Zhang and J. P. Ostrowski. Visual motion planning for mobile robots. *IEEE Transactions on Robotics and Automation*, 18(2):199–208, Apr 2002.
- [66] N. R. Gans and S. A. Hutchinson. A stable vision-based control scheme for nonholonomic vehicles to keep a landmark in the field of view. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2196–2201, April 2007.
- [67] Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275, January 2007.
- [68] A.M. Khamis. *Interacción remota con robots móviles basada en Internet*. PhD thesis, 2003.
- [69] Xiangyu Wang. *Mixed Reality and Human-Robot Interaction*, volume 47. Springer Netherlands, 1st edition, 2011.

- [70] D.A. Norman. *The Psychology of Everyday Things*. Basic Books, 1988.
- [71] D. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Books, 2007.
- [72] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983.
- [73] C. Yu, T. Xu, Y. Zhong, S. Foster, and H. Zhang. Learning to interact and interacting to learn: Active statistical learning in human-robot interaction. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 684–691, July 2014.
- [74] S. K. Hasnain, G. Mostafaoui, R. Salesse, L. Marin, and P. Gaussier. Intuitive human robot interaction based on unintentional synchrony: A psycho-experimental study. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–7, Aug 2013.
- [75] M. L. Walters, K. Dautenhahn, K. L. Koay, C. Kaouri, R. Boekhorst, C. Nehaniv, I. Werry, and D. Lee. Close encounters: spatial distances between people and a robot of mechanistic appearance. In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 450–455, Dec 2005.
- [76] D. S. Syrdal, K. Dautenhahn, S. Woods, M. L. Walters, and K. L. Koay. 'doing the right thing wrong' - personality and tolerance to uncomfortable robot approaches. In *RO-MAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 183–188, Sept 2006.
- [77] J.L. Williams. Personal space and its relation to extraversion-introversion. In *Canadian Journal of Behavioural Science*, volume 3, pages 156–160, Apr 1971.
- [78] M.L. Walters, University of Hertfordshire: Faculty of Engineering, and Information Sciences. *The Design Space for Robot Appearance and Behaviour for Social Robot Companions*. Ph.D Thesis. University of Hertfordshire, 2007.
- [79] J. D. Fisher and D. Byrne. Too close for comfort: Sex differences in response to invasions of personal space. In *Journal of Personality and Social Psychology*, volume 35, page 914, 1975.
- [80] D. S. Syrdal, K. Lee Koay, M. L. Walters, and K. Dautenhahn. A personalized robot companion? - the role of individual differences on spatial preferences in hri scenarios. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, pages 1143–1148, Aug 2007.
- [81] M. L. Walters, D. S. Syrdal, K. L. Koay, K. Dautenhahn, and R. te Boekhorst. Human approach distances to a mechanical-looking robot with different robot voice styles. In *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*, pages 707–712, Aug 2008.
- [82] Dag Sverre Syrdal, Kerstin Dautenhahn, Kheng Lee Koay, Michael L. Walters, and Wan Ching Ho. *Sharing Spaces, Sharing Lives – The Impact of Robot Mobility on User*

- Perception of a Home Companion Robot*, pages 321–330. Springer International Publishing, Cham, 2013.
- [83] J. Mumm and B. Mutlu. Human-robot proxemics: Physical and psychological distancing in human-robot interaction. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 331–338, March 2011.
- [84] S. Rulicki and M. Cherny. *Comunicación no verbal: Cómo la inteligencia emocional se expresa a través de los gestos*. Panorama Editorial, 2012.
- [85] A.M. Maqueo. *Lengua, aprendizaje y enseñanza: el enfoque comunicativo: de la teoría a la práctica*. Editorial Limusa, 2006.
- [86] C. D. Kidd and C. Breazeal. Human-robot interaction experiments: Lessons learned. In *Proceeding of AISB'05 Symposium Robot*, pages 141–142, 2005.
- [87] J. L. Burke, R. R. Murphy, D. R. Riddle, and T. Fincannon. Task performance metrics in human-robot interaction: Taking a systems approach. In *Performance Metrics for Intelligent Systems*, 2004.
- [88] C. L. Bethel, K. Salomon, R. R. Murphy, and J. L. Burke. Survey of psychophysiology measurements applied to human-robot interaction. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, pages 732–737, Aug 2007.
- [89] D.G. Elmes, B.H. Kantowitz, and H.L. Roediger. *Research Methods in Psychology*. Cengage Learning, 2011.
- [90] B. Johnson and L. Christensen. *Educational Research: Quantitative, Qualitative, and Mixed Approaches*. SAGE Publications, 2008.
- [91] J. Saez-Pons, H. Lehmann, D. S. Syrdal, and K. Dautenhahn. Development of the sociability of non-anthropomorphic robot home companions. In *4th International Conference on Development and Learning and on Epigenetic Robotics*, pages 111–116, Oct 2014.
- [92] G. Trovato, J. G. Ramos, H. Azevedo, A. Moroni, S. Magossi, H. Ishii, R. Simmons, and A. Takanishi. “olá my name is ana”: A study on brazilians interacting with a receptionist robot. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 66–71, July 2015.
- [93] M. M. A. de Graaf and S. Ben Allouch. The evaluation of different roles for domestic social robots. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 676–681, Aug 2015.
- [94] K. Lee Koay, D. S. Syrdal, M. L. Walters, and K. Dautenhahn. Living with robots: Investigating the habituation effect in participants’ preferences during a longitudinal human-robot interaction study. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, pages 564–569, Aug 2007.

- [95] G. Yasuda. Behavior-based autonomous cooperative control of intelligent mobile robot systems with embedded petri nets. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 1085–1090, Dec 2014.
- [96] Y. Zou, T. Wang, H. Wei, M. Liu, C. Li, and X. Lu. Robot software architecture based on ipv6. In *2011 6th IEEE Conference on Industrial Electronics and Applications*, pages 1666–1671, June 2011.
- [97] Li Xiaopen, Ma Hongwei, and Zou Zhihui. Research on open control architecture of autonomous mobile robot with multi-layer and modularization. In *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, volume 1, pages 511–515, March 2010.
- [98] J. Zhao, Q. Meng, W. Li, M. Li, and G. Chen. Ssvp-based hierarchical architecture for control of a humanoid robot with mind. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pages 2401–2406, June 2014.
- [99] S. A. Vukosavljev, D. Kukolj, I. Papp, and V. Kovacevic. A generic embedded robot platform for real time navigation and telepresence abilities. In *2011 Second Eastern European Regional Conference on the Engineering of Computer Based Systems*, pages 54–60, Sept 2011.
- [100] Dimas Carrasco Bellido and David Carrasco Bellido. *Biomecánica de la Actividad Física y del Deporte*. Biomecánica. Instituto Nacional de Educación Física, 2014.
- [101] Mark D. Latt, Hylton B. Menz, Victor S. Fung, and Stephen R. Lord. Walking speed, cadence and step length are selected to optimize the stability of head and pelvis accelerations. *Experimental Brain Research*, 184(2):201–209, Jan 2008.
- [102] Robot operating system. <http://www.ros.org>. Accessed: 2017-04-10.
- [103] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–511–I–518 vol.1, 2001.
- [104] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.
- [105] J. A. Rivera-Bautista, A. Marin-Hernandez, and L. F. Marin-Urias. Using color histograms and range data to track trajectories of moving people from a mobile robot platform. In *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, pages 288–293, Feb 2012.
- [106] Pablo Gil, Fernando Torres, and Francisco Gabriel Ortiz Zamora. Detección de objetos por segmentación multinivel combinada de espacios de color. In *XXV Jornadas de Automática, Comité Español de Automática*, pages 54–60, Sept 2004.
- [107] Pointclouds(2017). <http://pointclouds.org>. Accesado: 2017-11-10.

- [108] Algoritmo de reconocimiento de rostros(2017). http://wiki.ros.org/face_recognition. Accesado: 2017-11-10.
- [109] Rosintroduction(2017). <http://wiki.ros.org/ros/introduction>. Accesado: 2017-11-10.
- [110] Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. *Bezier and B-Spline Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
- [111] Manish N. Sreenivasa, Ilja Frissen, Jan L. Souman, and Marc O. Ernst. Walking along curved paths of different angles: the relationship between head and trunk turning. *Experimental Brain Research*, 191(3):313–320, Nov 2008.
- [112] Algoritmo de navegación(2017). <http://wiki.ros.org/amcl>. Accesado: 2017-11-10.
- [113] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [114] Michael C. McHenry Dan Helmick Chuck Bergh Stergios I. Roumeliotis Larry Henry Matthies Robert Hogg, Arturo L. Rankin. Sensors and algorithms for small robot leader/follower behavior. volume 4364, pages 4364 – 4364 – 14, 2001.
- [115] R. Sukthankar. Raccoon: A real-time autonomous car chaser operating optimally at night. In *Intelligent Vehicles '93 Symposium*, pages 37–42, Jul 1993.
- [116] R. Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Pittsburgh, PA, January 1992.
- [117] P. Hong, H. Sahli, E. Colon, and Y. Baudoin. Visual servoing for robot navigation. In *In Third International Conference on Climbing and Walking Robots*, 2001.
- [118] P. Renaud, E. Cervera, and P. Martinier. Towards a reliable vision-based mobile robot formation control. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 4, pages 3176–3181 vol.4, Sept 2004.
- [119] G. Weinberg and B. Blosser. A leader-follower turn-taking model incorporating beat detection in musical human-robot interaction. In *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 227–228, Marzo 2009.
- [120] D. Herrera, F. Roberti, M. Toibero, and R. Carelli. Human interaction dynamics for its use in mobile robotics: Impedance control for leader-follower formation. *IEEE/CAA Journal of Automatica Sinica*, 4(4):696–703, 2017.
- [121] Jeff Wit, Carl D. Crane, and David Armstrong. Autonomous ground vehicle path tracking. *Journal of Robotic Systems*, 21(8):439–449, 2004.
- [122] Siou Y Chiem, École Centrale De Nantes, and Enric Cervera. Vision-based robot formations with bézier trajectories. In *in Eighth Conference on Intelligent Autonomous System*, 2004.

- [123] W. Naeem, R. Sutton, and S. M. Ahmad. Pure pursuit guidance and model predictive control of an autonomous underwater vehicle for . . . In *14 DRDC SUFFIELD TM 2006-108*, 2003.
- [124] A. K. Das, R. Fierro, V. Kumar, B. Southall, J. Spletzer, and C. J. Taylor. Real-time vision-based control of a nonholonomic mobile robot. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 1714–1719 vol.2, 2001.
- [125] J. Smith. *A Tracking System for a Seismic Surveying Mobile Robot*. PhD thesis, University of Alberta, 2001.
- [126] J. Sattar, P. Giguere, G. Dudek, and C. Prahacs. A visual servoing system for an aquatic swimming robot. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1483–1488, Aug 2005.
- [127] N. Cowan, O. Shakerina, R. Vidal, and S. Sastry. Vision-based follow-the-leader. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1796–1801 vol.2, Oct 2003.
- [128] Ezio Malis and Selim Benhimane. A unified approach to visual tracking and servoing. *Robotics and Autonomous Systems*, 52(1):39 – 52, 2005. Advances in Robot Vision.
- [129] Estatura de la población mundial(2017). https://elpais.com/elpais/2016/07/21/media/1469127433_712478.html. Accesado: 2017-11-10.
- [130] Ronald R. Yager and Lotfi A. Zadeh, editors. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.
- [131] Diseño de un sistema experto para el enderezado de chasis en frío, capítulo 2 (2015). http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/maza_c_ac/indice.html. A. C. Maza Cabello Accesado: 2017-11-10.
- [132] G. Chen and T.T. Pham. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press, 2000.
- [133] A.C. Barbero. *Métodos numéricos con Matlab*. Editorial UPV, 2005.
- [134] Splines: Curvas y superficies(2017). <http://www.esi.uclm.es/www/cglez/>. Carlos González Morcillo Accesado: 2017-11-10.
- [135] Jae-Hun Jung and Wolfgang Stefan. A simple regularization of the polynomial interpolation for the resolution of the runge phenomenon. *Journal of Scientific Computing*, 46(2):225–242, Feb 2011.
- [136] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1):29–53, 1998.
- [137] Hans Moravec. Sensor fusion in certainty grids for mobile robots. *AI Mag.*, 9(2):61–74, July 1988.

- [138] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *J. Artif. Int. Res.*, 11(1):391–427, July 1999.
- [139] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 343–349, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.
- [140] Andrew P. Duchon, William H. Warren, and Leslie Pack Kaelbling. Ecological robotics. *Adapt. Behav.*, 6(3-4):473–507, October 1998.
- [141] James L. Crowley. *Mathematical foundations of navigation and perception for an autonomous mobile robot*, pages 7–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.
- [142] M. Bennewitz W. Burgard A.B. Cremers F. Dellaert D. Fox D. Hähnel C. Rosenberg N. Roy J. Schulte S. Thrun, M. Beetz and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.
- [143] S. Thrun, Langford. J., and V. Verma. Risk sensitive particle filters. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.
- [144] Pascal Poupart, Craig Boutilier, and Luis E. Ortiz. Value-directed sampling methods for monitoring pomdps. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 453–461, 2001.
- [145] Thomas G. Dietterich. The maxq method for hierarchical reinforcement learning. In *In Proceedings of the Fifteenth International Conference on Machine Learning*, pages 118–126. Morgan Kaufmann, 1998.
- [146] Birgit Graf and Oliver Barth. Entertainment robotics: Examples, key technologies and perspectives. In *in: IROS-Workshop Robots in Exhibitions*, 2002.
- [147] A. Traub and R. D. Schraft. An object-oriented realtime framework for distributed control systems. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4, pages 3115–3121 vol.4, 1999.
- [148] Kai O. Arras, Nicola Tomatis, Björn T. Jensen, and Roland Siegwart. Multisensor on-the-fly localization: Precision and reliability for applications. *Robotics and Autonomous Systems*, 34:2–3, 2001.
- [149] P. J. Prodanov, A. Drygajlo, G. Ramel, M. Meisser, and R. Siegwart. Voice enabled interface for interactive tour-guide robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1332–1337 vol.2, 2002.
- [150] Wolfram Burgard, Panos Trahanias, Dirk Hähnel, Mark Moors, Dirk Schulz, Haris Baltzakis, and Antonis Argyros. Tele-presence in populated exhibitions through web-operated mobile robots. *Autonomous Robots*, 15(3):299–316, Nov 2003.

- [151] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *The International Journal of Robotics Research*, 17(4):315–337, 1998.
- [152] Javier Minguez and L. Montano. Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, Feb 2004.
- [153] H. H. Lin, C. C. Tsai, and H. Y. Chang. Global posture estimation of a tour-guide robot using rfid and laser scanning measurements. In *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, pages 483–488, Nov 2007.
- [154] Yuan-Pao Hsu, Ching-Chih Tsai, Zeng-Chung Wang, Yi-Jiang Feng, and Hung-Hsing Lin. Hybrid navigation of a four-wheeled tour-guide robot. In *2009 ICCAS-SICE*, pages 4353–4358, Aug 2009.
- [155] Ching-Chih Tsai, Shu-Min Shish, Hsu-Chih Huang, Min-Yu Wang, and Chih Chung Lee. Autonomous navigation of an indoor tour guide robot. In *2008 IEEE Workshop on Advanced robotics and Its Social Impacts*, pages 1–6, Aug 2008.
- [156] H. H. Lin and W. Y. Tsao. Automatic mapping and localization of a tour guide robot by fusing active rfid and ranging laser scanner. In *The 2011 International Conference on Advanced Mechatronic Systems*, pages 429–434, Aug 2011.
- [157] S. Behnke. Online trajectory generation for omnidirectional biped walking. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1597–1603, May 2006.
- [158] F. Faber and S. Behnke. Stochastic optimization of bipedal walking using gyro feedback and phase resetting. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 203–209, Nov 2007.
- [159] J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1012–1017 vol.2, June 1991.
- [160] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 2, pages 1322–1328 vol.2, 1999.
- [161] C. Stachniss, M. Bennewitz, G. Grisetti, S. Behnke, and W. Burgard. How to learn accurate grid maps with a humanoid. In *2008 IEEE International Conference on Robotics and Automation*, pages 3194–3199, May 2008.
- [162] Robovie-R Ver.3, howpublished = <https://techcrunch.com/tag/robovie-r-ver-3/>, note = Accessed: 2017-10-14.
- [163] A. Kendon. *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Studies in Interactional Socio. Cambridge University Press, 1990.
- [164] Charles Goodwin. Restarts, pauses, and the achievement of a state of mutual gaze at turn-beginning. *Sociological Inquiry*, 50(3-4):272–302, July 1980.

- [165] R. Sanz A.R. Diéguez, J.L. Fernández. A global motion planner that learns from experience for autonomous mobile robots. In *Robotics & CIM*, volume 23, pages 544–552, Oct 2007.
- [166] J.L. Fernández, R. Sanz, J.A. Benayas, and A.R. Diéguez. Improving collision avoidance for mobile robots in partially known environments: the beam curvature method. *Robotics and Autonomous Systems*, 46(4):205 – 219, 2004.
- [167] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.
- [168] R. Ramón-Vigo, N. Pérez-Higueras, F. Caballero, and L. Merino. Transferring human navigation behaviors into a robot local planner. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 774–779, Aug 2014.
- [169] F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrilă. A probabilistic framework for joint pedestrian head and body orientation estimation. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1872–1882, Aug 2015.
- [170] I. Marras, J. A. Medina, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Online learning and fusion of orientation appearance models for robust rigid object tracking. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8, April 2013.
- [171] John D. Gould and Clayton Lewis. Designing for usability: Key principles and what designers think. *Commun. ACM*, 28(3):300–311, March 1985.
- [172] A. K. Pandey and R. Alami. A framework towards a socially aware mobile robot motion in human-centered dynamic environment. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5855–5860, Oct 2010.

Apéndice A

Características de los robots guía (1993-2018)

A continuación se presentan las características de los distintos trabajos desarrollados para la tarea de robot guía en orden cronológico. Para esta clasificación resaltan tres diferentes aspectos: el hardware del robot, la navegación, el lugar donde se desenvuelve y la manera en la que se relacionan con las personas. Esta información resulta relevante para el desarrollo del presente trabajo de investigación.

En la primera columna (Robot) se menciona el nombre del robot y su entorno de trabajo. En la segunda columna (Hardware) se presentan los sensores y actuadores utilizados por estos robots. La tercera columna (Navegación) describe el proceso de navegación. La cuarta columna (HRI) describe la manera en que el robot y el usuario se comunican e interaccionan.

Tabla A.1: Historia de los robots guía (1993-2018).

Robot	Hardware	Navegación	HRI	Ref.
<p>Polly 1993 Creado en el Laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachusetts (MIT), este robot ofreció sus servicios en el piso del laboratorio de Inteligencia Artificial (IA) de MIT (edificio de oficinas)</p>	<p>Polly tenía una base móvil síncrona RWI B12 capaz de viajar a $1.5m/s$ y se basaba en una visión de bajo costo para el reconocimiento del entorno con una cámara de 64×48 píxeles en escala de grises a 15 cuadros/segundo.</p>	<p>Estos sistemas controlan la velocidad aplicada a cada rueda para avanzar, girar o detenerse, con esto se lleva a cabo la navegación del robot. Para evitar obstáculos, el robot busca áreas limpias en frente de él a través de los píxeles inferiores de la imagen al restringir la textura del terreno y el plano de tierra. Finalmente decide entre tres formas posibles (izquierda, adelante, derecha). La ubicación del robot en el entorno se realiza a través de ejes de coordenadas para representar la posición en 2D. Cada lugar tiene una descripción en posición (x, y), dirección (norte, sur, este, oeste), rotación ($0^\circ, 45^\circ$) o alguna descripción (derecha, izquierda, pared).</p>	<p>Polly busca usuarios en los corredores, se presenta y ofrece un recorrido por los diferentes laboratorios. Las personas aceptan que el robot los guíe agitando los pies, porque era lo único que el robot podía detectar.</p>	<p>[1]</p>
<p>Rhino 1998-1999 Ofrecía una guía turística interactiva en el Deutsches Museum Bonn en Alemania</p>	<p>Cuenta con 4 sensores de última generación (láser, sonar, infrarrojo y táctil), también tenía una cámara estéreo, una base móvil RWI B21, tres PC's a bordo y tres estaciones de trabajo. Este robot cuenta con alrededor de 25 módulos independientes que trabajan en paralelo, algunos dedicados a la navegación y la ubicación y otros al HRI.</p>	<p>Se desarrollaron algunos algoritmos para navegar entre multitudes y evadir obstáculos a una velocidad máxima de $80cm/s$. Básicamente, la navegación se divide en dos módulos: la percepción estima la ubicación del robot y los obstáculos cercanos. El módulo de control realiza la evasión de obstáculos mediante la planificación de rutas en tiempo real. Se utilizó una versión modificada del algoritmo de localización de Markov junto con un filtro de entropía [50] para aproximar la posición del robot. Además, se proporcionó un mapa métrico del museo al robot y debido a que el entorno cambia constantemente, se utilizó un algoritmo de cuadrícula de ocupación probabilística [51] para modificar el mapa inicial en tiempo real para evitar obstáculos. Cada vez que el robot llegaba a su objetivo, este mapa se restablece al original. El movimiento del robot se realiza controlando la dirección y la velocidad de este, en función de la información de los sensores de entrada y la ubicación del objetivo. Rhino usa una extensión del algoritmo de ventana dinámica (DWA) [52] que le permite navegar sin problemas alrededor de los obstáculos hacia su objetivo. Debido a la presencia de obstáculos invisibles, el robot debe consultar el mapa proporcionado antes de comenzar, expandiendo el enfoque DWA para estimar la posición del robot dentro del mapa y generar mediciones virtuales de proximidad las cuales se integran con las mediciones reales obtenidas de los sensores del robot (táctil, infrarrojo, sonar, láser). El planificador de rutas de Rhino se basa en la iteración de valores, una programación dinámica y un algoritmo de aprendizaje de refuerzo. Las rutas generadas se ejecutan después de considerar la rutina de prevención de colisiones, generando ubicaciones de destino.</p>	<p>Rhino ofrecía una visita guiada interactiva y un recorrido de presencia virtual para usuarios de cualquier parte del mundo. El robot tenía una interfaz simple para usuarios sin experiencia, con una pantalla que mostraba texto, gráficos, voz y sonido, además de cuatro botones de selección. El usuario puede elegir entre iniciar/continuar la ruta o una explicación más amplia de las exposiciones. Para expresar la intensión o la insatisfacción, el robot hacía movimientos con su cuerpo y cabeza (cámara en la parte superior del robot) y sonidos.</p>	<p>[2, 3]</p>

Robot	Hardware	Navegación	HRI	Ref.
Minerva 1999, Se exhibió en el Smithsonian National Museum of American History en Washington durante dos semanas	Estaba equipado con una base móvil RWI B18 que tiene un accionamiento diferencial no holonómico, un láser, un sonar, un sensor infrarrojo y sensores táctiles, su velocidad máxima era de 163cm/s.	Minerva usa los mismos módulos de navegación que Rhino, las principales mejoras se presentan a continuación. Un mapa de ocupación [136] y mapas de textura del techo del museo se usan para orientarse. Los mapas se obtuvieron de los sensores moviendo al robot dentro de su entorno con un joystick. Esta técnica de mapeo se conoce comúnmente como mapeo y localización concurrente. Este enfoque expresa el problema de localización y localización concurrente como un problema de estimación de máxima verosimilitud, el mapa más probable se determina estimando los errores en la odometría del robot. El planificador de caminos de Minerva se llama planificador costero, el cual se mantiene cerca de las orillas para evitar perderse.	La interacción de Minerva con las personas tiene dos objetivos particulares: atraer a las personas para usar el robot y despejar su camino durante el recorrido. Un rostro se implementó en Minerva, este le permite hacer expresiones faciales y diferentes tonos de voz para interactuar con los visitantes. Los estados de ánimo variaron de felices a enojados, dependiendo de la persistencia de las personas que bloqueaban su camino.	[4] [5]
Sage 1999, En el Museo Carnegie de Historia Natural operó durante un período 9 meses	Sage es una modificación del robot Nomad XR4000, su sistema de transmisión contiene en cuatro ruedas que permiten un movimiento holonómico. Tiene codificadores para la medición de odometría, una cámara a color y para cubrir los puntos ciegos presentados en los artículos anteriores, el robot tiene dos anillos de sensores de sonar, con 24 transductores de sonar en cada anillo. El sistema de navegación de Sage solo usa visión de color e información odométrica.	Los trabajos citados anteriormente tienen un campo de visión limitado debido a los sensores y usan un mapa de ocupación de cuadrícula para evitar obstáculos. La planificación de ruta se basa en un mapa local que se actualiza de acuerdo con los obstáculos encontrados en el camino. Los sensores de Sage tienen una cobertura completa para eliminar el punto ciego del robot. Además, no crea una representación interna del entorno con obstáculos, sino que, en cada ciclo calcula la trayectoria de recorrido más libre. Los parámetros de aceleración de Sage que marcan el movimiento del robot son establecidos por un controlador de motor de bajo nivel. En este museo, el personal se encarga de mantener los pasillos libres de obstáculos, lo que facilita el sistema de prevención de obstáculos de Sage. La navegación de los robots guía Rhino y Minerva se basa en enfoques probabilísticos y la información del telémetro láser SICK. Sage utilizó una navegación visual para evitar el costo computacional de la navegación probabilística. El robot se limitó a un conjunto predefinido de rutas unidireccionales y seguras, además de tener un escenario confiable de reconocimiento de puntos de referencia visuales. Tres rectángulos rosas de alto contraste, de aproximadamente 60cm de lado, se instalaron en las paredes de los pasillos.	Sage consiste en un reproductor de video, un sistema de sonido y una pantalla para la interacción humano-robot. Uno de los objetivos de este proyecto fue crear un robot educador involucrado con la audiencia. Los desafíos presentados fueron el desarrollo del contenido educativo y la interacción educador-visitante. Sage tenía una personalidad afectiva a través de una arquitectura general como un esquema de transición del estado de ánimo. Se eligió una máquina de estado difusa, los eventos del entorno causan cambios en el estado del robot. Los estados de ánimo elegidos fueron feliz/ocupado, cansado, frustrado y confundido. Los educadores también presentaron escenarios que deberían causar el cambio entre los estados de ánimo.	[6]

Robot	Hardware	Navegación	HRI	Ref.
<p>PERSES 2000, PERsonal SERvice System un robot asistente de compras móvil interactivo en una tienda de mejoras para el hogar.</p>	<p>La base móvil de la plataforma es la misma que Rhino (B21 de RWI), además del equipamiento estándar de dos sonares (48 sensores) y una línea de sensores IR (32 sensores), el robot está equipado con una cámara omnidireccional a color con una vista panorámica de 360° y un telémetro láser 2D utilizado para localización y seguimiento del usuario, auto localización y navegación local, un binocular, una cabeza de 6 gdl con visión activa de 2 cámaras a color alineadas frontalmente utilizadas para la verificación y seguimiento del usuario, corrección de odometría y evasión de obstáculos, y un sistema auditivo para la localización acústica y seguimiento del usuario.</p>	<p>Para la creación de mapas, la auto localización y la navegación global, PERSES utilizó técnicas estadísticas y probabilísticas [137, 51, 3, 138, 139]. PERSES utilizó dos tipos de mapas para la auto localización y navegación: mapas de ocupación basados en el cuadrículado y otra cuadrícula de vistas panorámicas de los alrededores. Para atenuar el error de datos de odometría, se utilizó un método basado en la visión para la construcción de mapas. Para evitar confusión de auto localización en métodos basados únicamente en sensores de distancia, en este artículo se desarrolló un enfoque para la auto localización basada en la visión que combina vistas panorámicas de la omnicámara con la Localización Monte Carlo (MCL). Debido a que pueden existir gran cantidad de obstáculos, estos no se pueden percibir de manera confiable mediante sensores de distancia 2D (sonar, láser), por ello, se implementaron métodos complementarios basados en la visión para evitar obstáculos (flujo óptico y mapeos de perspectiva inversa de la imagen panorámica [140]). El control de movimiento del robot utilizó diferentes módulos: guía, seguimiento del usuario y módulo de evasión de obstáculos, que se activan dependiendo de la prioridad en la acción del robot. Para la ejecución de todas estas tareas, se usaron redes neuronales para controlar los motores.</p>	<p>PERSES detecta y contacta a usuarios potenciales, los guía a las áreas/artículos deseados o los sigue como un quiosco de información. La comunicación con el cliente es continua e intuitiva. Para localizar clientes, realiza una segmentación en primer plano del movimiento separándolo del fondo basado en la secuencia de imágenes y devuelve el ángulo al centro de gravedad de la región móvil más grande, y con el sistema auditivo es fácil para el usuario atraer la atención del robot aplaudiendo o gritando un comando, la hipótesis de ubicación del usuario se verifica mediante el uso del sistema de elevación de multi-señal que integra diferentes señales visuales: color de piel, contorno de cabeza y hombro y la estructura facial. La comunicación se realiza mediante una pantalla táctil, que solicita el objeto o área deseada. En este artículo, solo se presentan los resultados preliminares.</p>	<p>[29]</p>
<p>Blacky 2002, Denning Branch, Inc. desarrolló un robot que funcionó con éxito en 3 entornos diferentes (2 ferias, 1 concurso de robots).</p>	<p>La plataforma móvil MRV4 cuenta con un sistema de transmisión sincronizada de tres ruedas, un anillo de sonares, una conexión inalámbrica, un láser giratorio horizontal y altavoces para síntesis de voz.</p>	<p>Este robot usó la técnica de seguidor de pasillos y la evasión inteligente, basada en la percepción del robot para el movimiento y evasión de obstáculos con un controlador de bajo nivel. Para compensar la falta de percepción, Blacky utiliza un mapa virtual de los pasillos, simula la percepción y con un filtro de Kalman extendido calcula la localización. La arquitectura de control implementada en Blacky se presenta en [141] e incluye las siguientes habilidades: controlador de bajo nivel capaz de evitar colisiones, comportamientos reactivos que se ejecutan en paralelo con la localización (control reactivo), control de supervisión que controla los movimientos reactivos y la influencia del mapa virtual sobre el módulo reactivo.</p>	<p>Tiene Blacky puede realizar presentaciones orales y visitas guiadas, utiliza diferentes frases para saludos, bienvenida y auto presentación.</p>	<p>[7]</p>

Robot	Hardware	Navegación	HRI	Ref.
<p>Pearl y Flo 2002, estos robots asistían a los huéspedes en un asilo de ancianos. Su tarea consistía en guiar, interactuar y proporcionar información a los residentes..</p>	<p>Ambos robots poseen un sistema de accionamiento diferencial. Están equipados con buscadores de rango láser SICK, sonares, micrófonos para reconocimiento de voz, altavoces para síntesis de voz, pantallas gráficas sensibles al tacto, unidades de cabeza activadas y sistemas de cámara estéreo. Los comentarios de enfermeras y expertos médicos que utilizaron Flo sirvieron de base para la creación de un robot más optimizado, Pearl. Pearl, a diferencia de Flo, también cuenta con dos manijas sólidas añadidas para brindar soporte a las personas mayores, un diseño más compacto que permite espacio de carga y una bandeja extraíble, capacidad duplicada de la batería, un segundo buscador láser y una cabeza significativamente más sofisticada.</p>	<p>El sistema de navegación de ambos robots se basa en el sistema del robot Minerva [2] [142], con la adición de tres nuevos módulos relacionados con la interacción y el control de las personas. Un módulo aborda el problema de localización de personas determinando su ubicación (x, y) relativa al robot. Pearl detecta personas utilizando la diferenciación de mapas: el robot aprende un mapa y las personas se detectan por desviaciones significativas del mapa. También se implementó un módulo de navegación más segura, debido a que el robot navega en presencia de personas mayores. Se restringió el área de operación del robot para evitar regiones densamente pobladas, utilizando un mapa del entorno mejorado manualmente. Para mantenerse dentro de su área de operación, el robot necesita una localización precisa, el filtro de partículas de localización del robot fue aumentado por una estrategia de muestreo [143, 144]. Un algoritmo probabilístico para el control de alto nivel y la gestión de diálogo representan el nuevo módulo central en el software de Pearl. En Pearl se implementó una variable jerárquica de un proceso de decisión de Markov parcialmente observable (POMDP) como arquitectura de control de alto nivel. POMDP es una técnica para calcular acciones de control óptimas bajo incertidumbre. Las decisiones del controlador en Pearl se basan en una multitud de variables de objetivo y estado probabilístico multivalor, como la ubicación del robot, la ubicación de la persona, el estado de la persona, el objetivo de movimiento, el objetivo de recordatorio, el objetivo iniciado por el usuario. El problema de la toma de decisiones se dividió en una colección de problemas más pequeños para resolverlos de manera más eficiente. El enfoque es como la descomposición MAX-Q para MDPs (Markov decision processes) [145] definida sobre POMDP.</p>	<p>Para la interacción, Pearl cuenta con software de reconocimiento de voz, software de síntesis de voz, captura rápida de imágenes y software de compresión para transmisión de video en línea, software de localización de detección de rostros y un prototipo de sistema de recordatorio flexible que usa técnicas avanzadas de planificación y programación. El robot fue probado en cinco experimentos separados, cada uno con una duración de un día completo. En primer lugar, el robot interactuó verbal y espacialmente con personas mayores con la tarea específica de entregar dulces. Después de esto, el robot condujo de manera autónoma 12 rutas completas, que involucraban a 6 adultos mayores diferentes. Todos pudieron operar el robot después de menos de cinco minutos de presentación. Hubo algunos problemas con un sistema de reconocimiento de voz mal ajustado que se solucionó durante el curso de este proyecto. Otro problema surgió de la incapacidad inicial del robot para adaptar su velocidad al ritmo de caminar de la gente.</p>	<p>[41]</p>

Robot	Hardware	Navegación	HRI	Ref.
2002, tres robots basados en la arquitectura de hardware y software de Care-O-bot fueron desarrollados en el Instituto Fraunhofer de Ingeniería de Fabricación y Automatización (IPA) [146]. Estos interactúan y entretienen a personas en el Museum für Kommunikation en Berlín desde marzo de 2000.	Cada vehículo tiene un accionamiento diferencial, un giroscopio, un escáner láser 2D, un parachoques en la parte inferior y varios sensores IR que están integrados en el parachoques, un sensor magnético orientado hacia el suelo. El robot alcanza una velocidad de $1.2m/s$ y el área está limitada por una banda magnética ubicada en el suelo. El software de control para los robots móviles se basa en 'Realtime Framework' [147] y en la biblioteca de software 'Robotics Toolbox', ambos desarrollados en IPA.	Las habilidades de navegación en las plataformas de robots móviles consisten en auto-localización, movimiento del Robot, concepto de seguridad, interfaz de usuario. La auto localización se basa en los datos obtenidos de la odometría y el giroscopio. Además, la información de un mapa del entorno se compara con los datos del escáner láser y el robot puede corregir su posición, los datos adquiridos se combinan utilizando un filtro de Kalman. La planificación del movimiento del robot se realiza de 3 formas diferentes: navegación controlada por el programa, navegación reactiva y ruta planificada previamente. Se implementó un sistema de seguridad de tres niveles en las plataformas móviles. Si se detecta un obstáculo cerca del robot, la velocidad del vehículo se reduce dependiendo de la distancia al obstáculo, y se busca la trayectoria más libre de obstáculo considerando varios parámetros. Todos los factores relevantes se unen usando un enfoque de lógica difusa. Para evitar aceleraciones y desaceleraciones innecesarias causadas por la prevención de colisiones, se aplicó un módulo de seguridad. Además, cada robot está equipado con sensores magnéticos orientados hacia el suelo para garantizar que ningún robot salga de su área de operación. Se usa un joystick para configurar los robots y para apagarlos después.	Cada robot tiene un rol diferente, uno da la bienvenida a los visitantes; el segundo proporciona un recorrido dentro del edificio; y el tercero juega con una pelota. Además, los robots pueden interactuar entre sí y con los visitantes del museo. El robot dará una plática de acuerdo con su modo actual hasta que comience su modo de operación predeterminado. Durante la operación de estos robots no ocurrieron colisiones con los visitantes ni con el inventario del museo, los robots nunca abandonaron su área operativa. Los robots tuvieron una gran aceptación por parte de los visitantes del museo y un fácil manejo para el personal inexperto.	[8] [9]
RoboX 2002, estos robots interactuaron con los visitantes de la Exposición Nacional Suiza Expo.02	Su apariencia es entre antropomórfica y máquina, RoboX tiene la altura aproximada del visitante promedio y se realiza un seguimiento del rostro de las personas por medio de una matriz de leds que tiene en los ojos, cuenta además con movimientos oculares. El robot también cuenta con parlantes, botones y micrófono. Para la navegación, los robots usan un telémetro, un parachoques y sensores táctiles. Consiste en una base móvil con dos ruedas impulsadas diferencialmente.	La localización y la elusión de obstáculos se basa en [148]. RoboX compara la información obtenida por el telémetro con un mapa estático. La distribución gaussiana de la información ambiental se usa para analizar si algún elemento pertenece al mapa estático o si es un elemento dinámico, esto servirá para actualizar y validar el mapa actual. Como un medio adicional de seguridad, las placas sensibles al tacto y los parachoques de espuma aseguran que el robot se detenga si choca con algo. Dos escáneres láser SICK montados a la altura de la rodilla proporcionan información ambiental para la navegación y la interacción. Una cámara que mira hacia el techo complementa la navegación, otra cámara montada en uno de los ojos del robot proporciona información adicional para la interacción.	El robot realiza expresiones faciales (feliz, sorprendida y enojada) y mantiene el contacto visual con las personas a través de detección de rostros. Por medio de la detección del color de piel, una extracción del contorno y filtrado, un seguimiento para actualizar las posiciones de las regiones ya rastreadas. Además, el robot usa un sistema de voz, en [149] se diseñó una metodología para desarrollar interfaces activadas por voz para robots de guía turística. Este trabajo aborda el diseño de un robot, que ejecuta una visita pre-programada en una exposición pública y permite la interacción compleja y colaborativa con visitantes inexpertos.	[10]

Robot	Hardware	Navegación	HRI	Ref.
<p>RoboX 2006</p> <p>Durante la exhibición en la Exposición Nacional Suiza Expo.02</p>			<p>se incluyeron detalles de las capacidades de interacción del robot, los requisitos para el sistema robótico, la confiabilidad y la operabilidad segura del diseño, detección de movimiento, rastreo facial, reconocimiento de voz y el estado emocional del robot.</p>	[11]
<p>Jinny 2004,2006</p> <p>fue utilizado en el Museo Nacional de Ciencia</p>	<p>Cuenta con dos ruedas impulsadas diferencialmente, dos telémetros láser, dos escáneres de infrarrojos, un giroscopio de fibra óptica, topes de goma, micrófonos inalámbricos, una pantalla táctil, reconoce y sintetiza el sonido de voz en coreano, un LCD display y una matriz de LEDs que expresan sus estados emocionales en forma de iconos, su velocidad máxima es de $1m/s$.</p>	<p>Utiliza sensores de rango para construir mapas, planificación de rutas y ubicación propia. El sistema de navegación traza la mejor trayectoria hasta el objetivo en un mapa sin modificaciones ni puntos de referencia artificiales. Cada módulo de navegación analiza el estado actual del robot y la información del entorno, estos componentes influyen en la selección del comportamiento del robot. Luego, puede detectar errores en la detección, manejar diferentes eventos de cada módulo y descomponer tareas a través de la configuración basada en la red Petri. En función de las condiciones del robot y la información del entorno, se elige la acción del robot. El sistema de control de movimiento de Jinny consiste en cuatro estrategias de navegación: 1. AutoMove: Modifican el método de gradiente de Konolige, el robot realiza una planificación de ruta cuando la trayectoria calculada está bloqueada o el punto de destino está obstruido. 2. AutoMove sin actualización de ruta: según la información del entorno, a veces será mejor no desviarse de la ruta inicial, incluso si el robot debe esperar a que el obstáculo salga del camino. 3. Seguimiento de pared de globo virtual: basado en el algoritmo de cobertura completa, Jinny realiza la técnica de seguimiento de pared para tener una auto-localización más precisa. 4. Movimiento controlado por control remoto: en caso de que el robot se pierda o no encuentre una ruta hacia su objetivo, puede ser controlado por una red inalámbrica.</p>	<p>Jinny se centra en la HRI, se desarrolló una arquitectura de software que proporciona una interacción activa entre robots y humanos, y es tan intuitiva que las personas sin experiencia pueden manejar al robot. Para que HRI sea amigable y natural, la base de datos de robots se puede expandir o modificar.</p>	[12], [13]
<p>Lefkos, Rinho y Albert 2005, se instalaron en la exposición de la Fundación Hellenic World, el Museo Bizantino y Cristiano, el Deutsches Museum Bonn y el Centro de Exposiciones Belgioioso en Italia.</p>	<p>La plataforma móvil es un RWI B21 con 4 sensores de última generación (láser, sonar, infrarrojo y táctil), también tenía una cámara estéreo, una base móvil.</p>	<p>Son robots capaces de operar en entornos poblados denominados "robots en exposiciones". En este artículo se presentaron las técnicas desarrolladas, como la prevención de obstáculos en los casos de objetos que no son visibles mediante el escáner láser, el mapeo 3-D, el mapeo en entornos dinámicos, la navegación predictiva y la coordinación multirobot [150]. Para un sistema de navegación seguro y confiable, los robots están equipados con escáneres láser de rango, sonares y cámaras. Estos sistemas utilizan un método de localización alternativo que implica un modelo híbrido, es decir, una combinación de un modelo discreto (Markov oculto) y un modelo continuo (filtro de Kalman) [3].</p>	<p>Tiene una interfaz de voz para usuarios en el sitio y una interfaz web flexible con capacidades de visualización mejoradas para usuarios remotos a través de Internet. Estos experimentos han proporcionado datos útiles sobre la respuesta del público y las actitudes hacia los guías turísticos robóticos. Además, han facilitado la evaluación y validación de la solidez y efectividad del sistema en condiciones reales.</p>	[14]

Robot	Hardware	Navegación	HRI	Ref.
Rackham 2005, su entorno de trabajo es una exposición que simula los interiores de una nave espacial imaginaria en 14 elementos interactivos, incluida la atmósfera visual y acústica.	Es un robot B21r, se amplió con una cámara digital Sony sobre un pan-tilt, una pantalla táctil, un par de altavoces, un giroscopio de fibra óptica y Ethernet inalámbrico. Se diseñó un mástil alto para mantener las cámaras alejadas de los niños.	Este proyecto utilizó una arquitectura de software propuesta en [151] integrada por 15 módulos. Primero, se obtiene la posición calculada por la odometría y corregida por el giroscopio y usando un láser SICK. Los datos obtenidos se combinan con un mapa registrado con un procedimiento SLAM (Simultaneous Localization And Mapping) clásico. Además, otro módulo extrae, identifica y localiza los polígonos planos que aparecen en los muebles. El movimiento del robot se controla mediante velocidades de referencia, integra un procedimiento de evasión local basado en una instancia algebraica de los diagramas de profundidad [152] y un planificador de trayectoria basado en un gráfico de visibilidad algebraica optimizado.	Con el algoritmo de detección de rostros el robot ubica personas, se presenta a sí mismo o explica cómo usar sus servicios. El visitante debe seleccionar un destino para que el robot calcule y muestre su trayectoria e invita al visitante a seguirla, además proporciona información durante el viaje y usando diferentes botones, el visitante puede detener y cambiar la misión en curso.	[15]
Robovies 2007, se utilizaron dos robots y otros cuatro robots humanoides para orientar a los visitantes del Museo de Ciencias de Osaka	Los robots móviles tienen una altura de 120 cm, consta de una cabeza, un par de ojos, dos brazos, una plataforma móvil, un altavoz, un lector de etiquetas RFID (Radio-frequency identification) y un LED IR en la parte superior. Se instalaron 20 lectores de etiquetas RFID, cámaras de video y cámaras infrarrojas en el Museo.	Uno de los robots sirvió como guía para las exhibiciones, dos robots estacionarios explicaron las exhibiciones, un robot los saluda por su nombre y cuando los visitantes se preparan para partir otro robot les pide que devuelvan las etiquetas RFID y se despide. El robot puede guiar a las personas a cuatro tipos de exposiciones al determinar al azar el objetivo. Se presentó una metodología para la estimación global de la posición del robot guía con el dispositivo de identificación por radiofrecuencia (RFID) en [153], se aplica un método de mínimos cuadrados y un esquema de filtrado de información extendida (EIF). El robot de guía turístico propuesto se basa en el mecanismo de conducción diferencial. Está equipado con un sistema de localización global RFID (4 etiquetas), un escáner láser, un módulo de control de movimiento. El método propuesto proporciona una estimación precisa de las posiciones iniciales/continuas de los movimientos desconocidos y la localización del robot guía para la navegación.	Cada visitante llevaba una etiqueta RFID dentro del museo, que sirve para la interacción de los robots con los visitantes. Los robots interactuaban de forma autónoma con los visitantes a través de gestos y expresiones similares al juego libre de niños [53]. Los experimentos revelaron que la combinación de la interacción de juego libre y la guía de exhibición afectaron positivamente las experiencias de los visitantes en el museo de ciencias.	[16]
SHOPBOT 2008, tiene como objetivo servir como un asistente de compras móvil, es un robot inteligente e interactivo que se emplea en centros comerciales o tiendas de mejoras para el hogar	Se utilizó el robot SCITOS A5, cuenta con una cámara omnidireccional montada en la parte superior de la cabeza, un conjunto de 24 sonares en la parte inferior y un buscador de rango láser (LRF) SICK S300 montado en dirección frontal a una altura de 35 cm, una pantalla táctil, un sistema de sonido y una cabeza de 6 gdl. Todos estos sensores se usaron para la navegación e interacción con personas.	La localización se basa en la construcción en línea de mapas del área de operación mediante enfoques avanzados SLAM Rao-Blackwellized y en un enfoque probabilístico para la detección y seguimiento de usuarios durante el recorrido. En [154, 155] se propusieron metodologías y técnicas para el control del movimiento y la navegación autónoma de un robot guía con una plataforma móvil omnidireccional de cuatro ruedas en los años 2008, 2009 y 2011. Para la navegación se consideraron tres comportamientos: el seguimiento de trayectoria punto a punto, la función de evasión de obstáculos utilizando el método histograma de distancia de transferencia (TDH) para evitar barreras limitantes en museos y el control remoto. Para la posición y orientación del objetivo, se diseñaron leyes de control no lineal para ajustar la velocidad lineal y la velocidad angular. Para controlar el robot guía se utiliza una navegación híbrida difusa. Como una mejora en la navegación, se desarrolló un método de localización cartográfica y localización global para un robot guía, fusionando mediciones de un dispositivo activo RFID, un dispositivo odométrico y un LRF [156].	El HRI se desarrolla a través del sistema de expresión facial, un altavoz, una pantalla y un mecanismo de movimiento oscilante en los brazos. El sistema HRI diseñado tiene tres unidades importantes de software: teleoperación, transmisión remota de imágenes y voz, y diálogo simple en chino. Las primeras tres funciones ayudan al técnico a controlar y dirigir el robot de forma remota en caso de emergencia. El último módulo está diseñado para que el usuario se comunique convenientemente con el robot.	[30], [29], [31], [32]

Robot	Hardware	Navegación	HRI	Ref.
<p>Robotinho 2009, 2010, es un robot guía humanoide de cuerpo completo, este robot fue implementado en museos</p>	<p>Es capaz de ejecutar caminatas omnidireccionales, cuenta con un acelerómetro de doble eje y dos giroscopios, un par de altavoces, dos cámaras y para la localización y evasión de obstáculos, Robotinho está equipado con un pequeño LRF y un anillo de ocho sensores de ultrasónicos.</p>	<p>Usa diferentes técnicas de navegación tales como patrones para caminar [157], estabilización del tronco [158], control de velocidad y dirección de rotación, un enfoque de campo potencial [159], auto ubicación a través de MCL [160] en mapas de cuadrículado [161] y planificación de ruta.</p>	<p>Realiza detección de rostros utilizando una cascada potenciada de clasificadores Haar que rastrea las características faciales usando un aproximador de función neuronal. Reconoce algunos gestos humanos, usa un altavoz y micrófonos para localizar usuarios. El robot realiza varias expresiones faciales respaldadas por sonidos o movimientos del brazo. Se realizaron cuestionarios a los usuarios para mejorar las capacidades de HRI.</p>	<p>[39], [40]</p>
<p>Robovie 2010, se presenta un robot que avanza hacia atrás durante las guías turísticas en los corredores de un centro comercial en Japón.</p>	<p>La tarea consiste en cuatro comportamientos: inactividad, saludo, guía y publicidad. Para reutilizar estos cuatro comportamientos, se desarrollaron dos funciones: controlador de gestos y un algoritmo de navegación.</p>	<p>El algoritmo de navegación consiste en caminar hacia adelante y hacia atrás controlando su velocidad y rotación. Para caminar hacia adelante, Robovie se coloca del lado del usuario (en diagonal al frente). Y para caminar hacia atrás, el robot se coloca delante del usuario, en ambas condiciones, a una distancia del usuario de 50 cm. El robot reduce su velocidad a la mitad cuando la distancia entre el robot y el usuario excede los 1.5 m y no se detiene hasta que llega al objetivo o encuentra un obstáculo en su trayectoria.</p>	<p>La función del controlador de gestos controla la cara y los brazos del robot para saludar, estrechar la mano y señalar la tienda.</p>	<p>[34]</p>
<p>Robovie R3 2012, diseñado por ATR y VStone, se utilizó como un robot móvil guía para museo.</p>	<p>Cuenta con 11 sensores táctiles, 2 cámaras USB (ojos), 2 micrófonos (orejas), 17 articulaciones, un altavoz (boca), conexión con la red, velocidad de 2.5 km/h, un marco de aluminio y un LRF opcional [162]. Se le incorporaron 2 LRF y una cámara omnidireccional uno para obtener la posición y la orientación del robot, el otro se utiliza para rastrear la posición de los visitantes.</p>	<p>HRI - Establece la formación espacial conocida como "F-formation" [163] al comienzo de la explicación. Además, utiliza un procedimiento sistemático conocido como "pausa y reinicio" [164], que establece un marco de orientación mutua entre el hablante (robot) y los visitantes. Este sistema consta de tres unidades de software: la unidad de detección y seguimiento de rostros, la unidad de seguimiento del cuerpo y la unidad de control del robot. Para rastrear la posición y la orientación del cuerpo y la cabeza del visitante se empleó el filtro de partículas. Para la formación F se consideraron los siguientes datos: la distancia entre el robot y los visitantes fue entre los 110 cm y los 130 cm, la distancia entre el robot y los objetos expuestos fue de 110 cm y el campo de visión del robot tuvo un límite de 15m. Después de esto, si los visitantes no miran al robot al comienzo de la presentación, el robot emplea "pausa y reinicio". Durante la explicación de los objetos expuestos, el robot reproduce acciones corporales predeterminadas no verbales, como mirar hacia los visitantes, gestos con las manos y señalar.</p>	<p>El robot basado en el modelo propuesto se compara con un robot que no lo emplea. 4 grupos de 2 personas participaron en el experimento con cada robot. Los experimentos mostraron la efectividad de la formación F y de "pausa y reinicio" como método HRI.</p>	<p>[35], [36]</p>

Robot	Hardware	Navegación	HRI	Ref.
Jazzy1113 2012, El experimento se realizó en un edificio de oficinas en la Universidad de Ritsumeikan, Japón.	Configuración diferencial y una velocidad máxima de 6.44 km/h. Tiene 0.84 m de altura, tiene un LRF, una cámara Point Gray Research, una PC portátil y un dispositivo de control para juegos Kinect de Microsoft.	A través de una cámara, el robot puede guiarse a sí mismo, detecta una cinta de vinilo verde en el piso que es el camino a seguir. El control de velocidad de la navegación se calcula según un umbral de distancia entre el robot y la persona. Si la persona se aleja del robot, deja de esperarlo. La prevención de obstáculos se realiza con el LRF, en caso de detectar cualquier oclusión, el robot se detiene y, si desaparece, el robot continúa.	Se presenta un método de identificación de personas simple, usando la información del color de la vestimenta del usuario y la información del esqueleto de la persona, el robot puede rastrear al usuario y guiarlo hacia el destino, incluso si el robot pierde al usuario temporalmente. También, ha podido mantener el umbral de distancia con el usuario de acuerdo con la velocidad de marcha.	[33]
Konrad y Suse 2012, dos robots guía capaces de navegar en un edificio de oficinas de cuatro pisos. Los robots ofrecen visitas guiadas e información sobre los laboratorios, salas de juntas, oficinas y empleados de la Universidad Tecnológica de Ilmenau.	SCITOS A5 es la plataforma robótica utilizada para los robots guía Konrad y Suse, esta plataforma fue producida por Metralabs GmbH Ilmenau y se desarrolló en un inicio como asistente de compras [30]. Se adicionó con un transmisor de infrarrojos que llama al elevador y para la estimación del piso actual, el robot usa un acelerómetro integrado que permite estimar el movimiento vertical del robot.	Para la navegación, el robot utilizó un mapa topológico jerárquico, este mapa consta de nodos y puertas de enlace que están conectados por bordes dirigidos. Para la localización, se utiliza el algoritmo de MCL basada en filtro de partículas. El robot realiza el seguimiento de personas por medio de un filtro Kalman 6D y calcula la posición y la velocidad de las personas suponiendo una aceleración aleatoria incierta. Los robots están ubicados en la entrada principal del edificio	Si una persona se acerca a ellos el robot ofrece sus servicios de guía turística. La comunicación con el usuario se establece mediante una pantalla táctil y un sistema de síntesis de voz. Se implementó un seguidor de personas con múltiples hipótesis. También utiliza un detector de pies calculado con el láser basado en el enfoque clasificador reforzado, un detector de rostro por el detector AdaBoost de Viola and Jones, un detector de movimiento que utiliza un enfoque de diferencial de imagen rápido y simple y detectores de forma del cuerpo completo basados en Histogramas de gradientes orientados (HOG). La interacción tiene 20 estados asociados con la navegación, el seguimiento de trayectoria, la interacción con la pantalla táctil.	[44]
Bellbot and Sacarino 2012, dos robots asistente de hotel. Entre sus tareas está la orientación de personas, proporcionar información y llevar objetos pequeños.	Con apariencia humanoide, Bellbot posee un sistema diferencial, un parachoques, 16 sonares, un láser Hokuyo URG-04LX, un láser Sick LMS100/10000, dos brazos (4 gdl), cámaras en sus ojos, boca de leds, cuello (2 gdl), pantalla táctil, parlantes y micrófono.	El robot Bellbot construye mapas usando un algoritmo de búsqueda por escaneo. Entonces, la mejor ruta para llegar a los puntos objetivos es trazada usando el planificador de descenso de gradiente local de Konolige. El robot evita los obstáculos utilizando el método de control reactivo BCM (Beam Curvature Method), la localización dentro de un mapa se realiza mediante un filtro de partículas, para entornos poblados, puede usar etiquetas IR. El control de movimiento se divide en planificación de alto nivel (estratégica) [165] y prevención de colisión de nivel inferior (táctica) utilizando el método BCM [166].	Bellbot interactúa con personas que usan dos sensores táctiles en las mejillas del robot y en las manos, una pantalla táctil con una GUI (Graphical user interface) simple y un sensor láser. Aunque el robot tiene dispositivos para la interacción, en este trabajo de investigación el robot no tiene capacidades sociales.	[17]

Robot	Hardware	Navegación	HRI	Ref.
CATE 2012, opera en el edificio de Ingeniería y Tecnología en la Universidad Central de Michigan (CMU).	Cuenta con una base móvil Pioneer P3-DX y utiliza sensores IR, un RFID y sonares para una navegación autónoma.	Para la ubicación, el robot leerá etiquetas RFID conectadas a las paredes del edificio, si CATE se mueve 12 metros sin leer ninguna etiqueta RFID, se detendrá y enviará un código de error. Para que el robot esté en el rango de distancia de la etiqueta RFID, se implementó un seguidor de pared.		[18]
2013, el robot operó en los pasillos del laboratorio de Ingeniería Eléctrica e Informática (mismo entorno que en [23]).	La base móvil es un iRobot Create, se compone de un nodo ciego que se encuentra en el robot (el cual también se usa para controlar al robot) y múltiples nodos de anclaje que se encuentran en el entorno.	Para la navegación, el robot emplea una estrategia de seguimiento de pared simple. Utiliza un sensor de rango IR para detectar y seguir la pared del pasillo mientras realiza el recorrido. Mediante el método de la técnica del centroide ponderado y el uso de nodos de referencia de ZigBee distribuidos con ubicaciones conocidas, el robot se puede ubicar dentro de un edificio. El nodo ciego ejecuta un algoritmo de localización basado en los datos de los nodos de anclaje para determinar cuándo el robot ha llegado a su próximo destino.	Los métodos de interacción no se implementaron en esta investigación.	[21]
2014, El entorno de trabajo del robot se encontraba en los pasillos de los laboratorios de Ingeniería Eléctrica e Informática de una universidad.	El robot utiliza la plataforma de doble tracción DF (4WD), está equipado con un microcontrolador Arduino UNO R3, un módulo bluetooth y 4 sensores de rango ultrasónico.	El proceso de localización está basado en códigos QR que proporcionan referencias de ubicación para el robot, se realizó una aplicación móvil llamada "Aplicación de guía turística" en un teléfono inteligente. El entorno experimental se compone de varios códigos QR y señales de forma circular. El robot guía turístico utiliza el método de seguimiento de pared basado en un controlador PID para modificar la velocidad de las ruedas. Los obstáculos están limitados a las puertas abiertas, el robot encontrará puntos de referencia a lo largo de la ruta y los códigos QR almacenan información para ayudar a la navegación.	A través de la aplicación en el teléfono inteligente que reconoce los QR, se ejecuta la función "Hablar" y se explica automáticamente la información del laboratorio. La implementación de este robot es de bajo costo y mostró un rendimiento efectivo para guiar a los visitantes a los lugares de destino.	[28]
2015, En Asia Pacific University Engineering Labs	El robot mide 140 cm de alto y 50 cm de ancho, tiene un procesador Raspberry pi, tiene una cámara web Logitech HD, un solo altavoz usb, 4 sensores ultrasónicos, 4 ruedas omnidireccionales, 4 motores CC sin escobillas, 5 baterías y un microcontrolador Arduino.	El robot ejecuta la subrutina de ajuste a la pared y busca obstáculos en sus rutas. Si se detecta un obstáculo, primero espera 3 segundos y verifica si el obstáculo se mueve, si no, lo define como un objeto estático y se aleja del objeto. El robot usa la subrutina de procesamiento de imágenes para verificar si se usa el modelo de evacuación de multitudes basado en CA (Autómatas celulares). Esto proporciona una estimación del desarrollo de la evacuación. Y la "Evacuación de robots en condiciones reales", evalúa la situación actual del robot durante la evacuación. Se creó un mapa virtual del área a ser evacuada y con la ayuda de cámaras de vigilancia ubicadas en el edificio, se detecta y evalúa la actividad humana. Con los modelos mencionados anteriormente, se simula la progresión de la evacuación y se calcula la trayectoria menos congestionada. A través del módulo de ROS localización Visual y simultánea (VSLAM), el robot conocerá su ubicación dentro del entorno.	Se muestra una caricatura animada en la pantalla de una tableta. El robot debe encontrar personas y guiarlos hacia la salida. En esta investigación, los factores de HRI no se analizan. Sin embargo, se implementaron reglas simples que proporcionan seguridad a las personas, por ejemplo, atraer la atención de las personas para guiarlas hacia la salida mediante comandos de voz. Además, la velocidad del robot se limita a 0.89m/s para no intimidar o dañar a las personas.	[27]

Robot	Hardware	Navegación	HRI	Ref.
Robovie-R3 2015,	Se colocaron sensores de cámara de video USB encima de las pinturas para identificar el interés de una persona en una pintura específica.	Se programaron 4 caminos y posiciones posibles para la navegación del robot. El robot Robovie-R3 se encuentra a una distancia pública de las personas y se traslada a un espacio social, cerca de las exhibiciones.	Este sistema observa el interés de las personas hacia las pinturas en museos y les ofrece orientación. La orientación de la cabeza y la información de perfil de la persona se utilizan para conocer el interés de estas. El sistema de detección y seguimiento de personas se basa en el método de seguimiento 3D Head, basado en el método de Viola and Jones [167]. Una vez que se detecta el interés de la persona, el robot se moverá desde su punto inicial al lugar de la exposición (distancia social) para iniciar la interacción con el usuario (s) y ofrecer sus servicios como guía turística y proporcionar información sobre las pinturas.	[19], [20]
FROG 2015 , Fun Robotic Outdoor Guide es un guía turística en la Sala de Fiestas de Royal Alcázar en Sevilla, España.	El robot utilizado era una plataforma de cuatro ruedas, que fue diseñado con el fin de atraer a los visitantes, el robot tiene dos ojos como cámaras que detectan grupos de personas. Un brazo de puntero con otra cámara y luces LED, una pantalla táctil y un parachoques alrededor de la base para la detección de colisiones.	El robot puede navegar autónomamente evadiendo obstáculos en un entorno social [168]. Además, FROG puede buscar grupos, estimar su orientación y conducir hacia ellos [169]. Así como también puede ajustar el contenido de una visita al interés de los visitantes en función de sus expresiones faciales [170].	El robot mantiene una interacción con el usuario con dos tipos de narración: 1.- Proporcionando información durante la ejecución con frases pre-grabadas y voz de robot, 2.- Usando una narración humana pre-grabada para describir los lugares del edificio. Esta interacción es compatible con la pantalla táctil que muestra la cara de FROG y proporciona información gráfica. Con el fin de estudiar el comportamiento y la respuesta de las personas cuando interactúan con el robot, se realizaron estudios a través de entrevistas a las personas invitadas (aprox. 30 min) y a los usuarios imprevistos (5 a 10min). La información recopilada consiste en grabaciones de voz de los participantes invitados durante el recorrido del robot guía, además de grabaciones de video y audio. Durante el estudio se usó un enfoque de diseño iterativo centrado en el usuario [171] para el comportamiento del robot. Los resultados mostraron una primera impresión positiva, los usuarios encontraron el robot entretenido e informativo. La gente interactuaba con el robot cuando presentaba una voz robótica, aunque les parecía que sus oraciones eran repetitivas. A varios de los entrevistados les gustó que el robot ofreciera información en inglés. La característica que más le gustaba a la gente era la información que ofrecía y la novedad de ser guiado por un robot guía. Entre las características que menos agradaron a los usuarios fueron que en muchas ocasiones no entendieron la intención del robot ni hacia dónde se dirigía, además de su baja velocidad. Finalmente, el artículo explica los diferentes problemas que ocurrieron al cambiar de un entorno controlado a un entorno real con personas que actúan de forma imprevista.	[25], [26]
2016,	La plataforma móvil es un B21r (tracción síncrona) controlada por el sistema rFLEX de iRobot. Está equipado con un escáner láser Hokuyo y un Kinect.	Al definir un objetivo para el robot, se traza una ruta corta, sencilla y segura y se evitan los obstáculos dinámicos de un entorno social. Hay cuatro elementos principales para la navegación social: percepción, trayectoria de planificación, navegación y reglas sociales. La navegación del robot se realiza con un algoritmo disponible en la pila de navegación ROS,	El módulo de navegación de ROS se modificó al agregar las reglas sociales 1,2,3 de Pandey [172] y agregar dos más. Estas dos reglas se refieren al espacio social entre individuos y robots. El seguimiento de personas se realiza mediante la detección del esqueleto humano a través del sensor RGB-D (Kinect), donde se calculan cuatro orientaciones del cuerpo humano. Después de realizar la detección, se asigna una identificación al usuario.	[37]

Robot	Hardware	Navegación	HRI	Ref.
Eddie 2016,	La plataforma del robot tiene sensores ultrasónicos, un Kinect y lector de etiquetas RFID, usa el microcontrolador multinúcleo Propeller para controlar el movimiento de una base diferencial.	El robot es capaz de detectar y evitar obstáculos, por medio de los sensores. Se implementó un sistema de seguimiento y se proporcionarán archivos de audio almacenados sobre los objetos.	Reconoce comandos de voz como: seguir al usuario, detectar piezas de museo y brindar información sobre las piezas detectadas. El usuario se comunica con el robot usando gestos y comandos de voz. El primer comando es la palabra "Sígueme", esta función depende del método de distancia que determina el Kinect. Una vez que se active la función "Sígueme", el robot podrá seguir al usuario mientras evita los obstáculos que enfrenta durante su recorrido. Además, se proporciona reconocimiento facial y una captura de movimiento en 3D de cuerpo completo. El objetivo de este robot es ofrecer una experiencia agradable y entretenida a los turistas, acompañándolos a donde sea que vayan y ofreciendo información en cada una de las diferentes exhibiciones.	[38]
SPENCER 2016, es un robot asistente en grandes y ajetreados aeropuertos. Este robot informa y guía a los pasajeros.	El robot SPENCER, utiliza una apariencia similar a la humana, interactúa con el usuario a través de una pantalla táctil, también tiene un lector de pases de abordaje, 2 escáneres láser SICK 2D que cubren 360° de rango, 2 RGB frontales y 2 cámaras RGB-D y un sistema de cámara estéreo. En este proyecto se utiliza el Robot Operating System (ROS).	El robot debe navegar en entornos con alto dinamismo, para esto, se implementaron los siguientes algoritmos: el mapa de ocupación de transformaciones de distribución normal (NDT-OM), el algoritmo de fusión NDT-OM, la estructura de datos denominada mapa de transición condicional (CT-Map). CTMap le permite planificar trayectorias que interfieren menos con el flujo de pasajeros. Para el cálculo de las trayectorias, SPENCER calcula una trayectoria global que evade a las personas y evita cambios repentinos en el movimiento. Esto se logra cambiando la velocidad y anticipando futuras colisiones en el entorno dinámico.	Uno de sus objetivos es llevar a los viajeros de los vuelos de conexión desde su puerta de llegada al control de pasaportes. En este trabajo, se presentan nuevos métodos para percibir, aprender y modelar el comportamiento social humano. El comportamiento del robot SPENCER depende del sistema de supervisión, la tarea, el planificador de acciones y el módulo de planificación de movimiento. El sistema de supervisión decide entre los siguientes planes de acción del robot: interactuar con usuarios potenciales antes de guiar, dar información a usuarios guiados y pedirles a otras personas que despejen el camino. Además, debería considerar fallas o peligros para las personas. Se considera el comportamiento de los humanos a través de un Proceso de Decisión de Markov de Observación Mixta (MOMDP) y un Planificador de Colaboración (CP), para que el robot pueda tener un alto nivel de adaptación a una situación dada.	[42], [43]
2016,	Se utilizó una plataforma móvil PIONEER3-DX, y se adicionó con un LRF progresivo (LRF) y un sensor Kinect hacia atrás. El movimiento del robot estaba controlado por potenciales integrados.	El movimiento del robot guía se controla de forma adaptativa. El robot adapta su velocidad y trayectoria al movimiento del usuario mediante la generación de campos de potencial artificial adaptativo. El control de movimiento adaptativo del robot se basa en el método de campo de potencial artificial. Para controlar el movimiento del robot, las fuerzas atractivas y repulsivas se introducen por separado, tanto para los usuarios como para el punto objetivo, y luego se aceptan.	El robot puede predecir la velocidad de movimiento del usuario y se adapta a él. El robot se mantendrá a cierta distancia del robot. Incluso si el usuario desvía la ruta original, el robot volverá a su tarea de guía. Con el sensor Kinect, las personas se detectan por su rostro, color y la información de movimiento. Es posible distinguir un grupo de usuarios del resto mediante la observación en el marco bayesiano. La distancia social a la que el robot debe mantenerse alejado del usuario es de 1.2 m. Se hicieron guías turísticas para 20 personas, de las cuales 16 cooperaron con el robot y las otras 4 se desviaron de la ruta. Las 20 personas fueron llevadas a su destino con éxito.	[45]

Robot	Hardware	Navegación	HRI	Ref.
s/n 2013, .	La plataforma móvil consiste en una base de robot Pioneer P3DX, un telémetro láser y una cámara de rango que se encuentra en la parte superior del robot.	Presentan un enfoque de navegación que aprende rutas previamente realizadas por una persona.	En este proyecto, la interacción del robot humano está planificada para el trabajo futuro.	[49]
s/n 2015,			se realizó un estudio sobre la comunicación entre humanos y robots. A través de los ojos humanos que muestran la mirada para la comunicación no verbal. Se ha investigado el diseño estático y los comportamientos dinámicos de las cabezas de los robots para una comunicación adecuada de la mirada con los humanos. Estos hallazgos son muy importantes para los robots de comunicación, como los robots de guías de museos. Esta investigación demostró que el giro de la cabeza del robot con los parpadeos de los ojos da una impresión amistosa mientras que el giro de la cabeza del robot sin parpadeos es adecuado para hacer que las personas desvíen su atención hacia la dirección en la que mira el robot.	[22]
s/n 2007 y 2015,			Algunas investigaciones se centraron en el desarrollo de técnicas de interacción, es decir, no incluyeron la navegación en sus pruebas experimentales.	[23], [24]

Robot	Hardware	Navegación	HRI	Ref.
Tbot 2018, opera en los pasillos de un edificio universitario donde se encuentran las oficinas de algunos docentes de la institución.	Consiste en una base móvil de tipo diferencial <i>iRobot</i> , la cual cuenta con un giroscopio y dos codificadores para el cálculo de la odometría, la velocidad lineal máxima es de 0.5 m/s , la velocidad angular es de $180^\circ/\text{seg}$. Además, el robot tiene dos sensores Microsoft Kinect para la percepción del entorno, estos sensores consisten en una cámara RGB y un emisor/receptor de luz infrarroja para la construcción del entorno en tres dimensiones. La altura del robot es de 120 cm y realiza la comunicación entre las diferentes tareas mediante el Sistema Operativo de Robots (ROS) el cual permite el paso de información entre un proceso y otro de manera paralela.	La actividad de guía de personas inicia con la detección y localización del usuario en un espacio tridimensional. Con esta información se calcula la velocidad del usuario con base en la distancia que recorre en un intervalo de tiempo. El robot cuenta con la información del mapa de su entorno previamente reconocido, el punto inicial y la ubicación de los lugares a los que el usuario puede dirigirse. Con esta información se realiza el cálculo de trayectorias, donde se genera una trayectoria global del punto inicial al punto final realizando un cuadrículado en el mapa y definiendo las celdas libres por las que puede pasar el robot. Además, es necesario conocer la localización del robot dentro del mapa la cual está basada en algoritmos probabilísticos (odometría, telemetría, localización de Monte Carlo aumentada). Con la posición actual del robot y la información del entorno es posible calcular trayectorias locales para la evasión de obstáculos, este cálculo consiste en conocer la velocidad lineal y angular del robot, así como la proximidad de obstáculos, del punto meta y de la trayectoria global. Con esta información, se realiza una simulación de la posición próxima del robot al cabo de un corto periodo de tiempo, para predecir si el robot colisionará. El movimiento del robot depende de la velocidad lineal y la velocidad angular, estas variables se ajusta por medio de un controlador difuso con un lazo interno y otro externo. El lazo interno consiste en un controlador PI+D difuso para adaptar la velocidad lineal del robot a la velocidad estimada del usuario. El lazo externo tiene como entrada la velocidad lineal del robot obtenida en el lazo interno y el error en orientación, el cual se calcula a partir de la trayectoria deseada y la posición actual del robot.	Por medio de normas sociales, una pantalla y con un sintetizador de voz se entabla la comunicación con el usuario. La adaptación de velocidad del robot busca brindar un mejor servicio al usuario, ya que incrementará las probabilidades de que el usuario lo siga y ambos lleguen a su objetivo. El control de las velocidades del robot está basado en normas reglas sociales de interacción entre humanos y robots.	54 , 55 , 56 , 57 , 58

Apéndice B

Pseudocódigos

Para el diseño e implementación de los diversos algoritmos utilizados en la tarea de guía de personas, es necesario considerar que deben ejecutarse en tiempo real, es decir, mientras el robot va en movimiento. Por esta razón, el procesamiento de los algoritmos debe ser rápido y eficiente. Los pseudocódigos de los algoritmos desarrollados y adaptados para la aplicación de robot guía se muestran a continuación.

Algoritmo 2 Imagen integral.

- 1: Entrada: Imagen RGB (640x480).
 - 2: Salida: Imagen Integral II .
 - 3: $II(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$.
 - 4: Return II .
-

Algoritmo 3 Filtros Haar.

- 1: Entrada: Imagen II (641x481).
 - 2: Salida: Muestras de características M .
 - 3: Calcula características tipo a, b .
 - 4: **for do** $1 \leq i \leq 640, 1 \leq j \leq 24$.
 - 5: **for dow**, h tal que $i + h - 1 \leq 640, j + 2w - 1 \leq 480$.
 - 6: $S1 = \sum_{i, i+h-1} II(i, j) \times \sum_{j, j+w-1}$.
 - 7: $M = M + 1$.
 - 8: **end for**
 - 9: **end for**
 - 10: Return M .
-

Algoritmo 4 Clasificador Adaboost.

-
- 1: Entrada: muestras M de imágenes $(x_1, y_1), \dots, (x_n, y_n)$ donde $y_i = 0, 1$ para muestras negativas y positivas, respectivamente.
 - 2: Salida: Ventana de la muestra clasificada correctamente V .
 - 3: Inicializa los pesos $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ para $y_i = 0, 1$, respectivamente, donde m y l son los números de muestras negativas y positivas, respectivamente.
 - 4: Para $t = 1, \dots, T$:
 - 5: Normaliza los pesos $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$.
 - 6: Para cada característica, j , entrenar un clasificador h_j el cual está restringido al uso de una sola característica.
 - 7: El error es evaluado respecto a $w_t, \epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
 - 8: Escoge el clasificador, h_t , con el más pequeño ϵ_t .
 - 9: Actualiza los pesos:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}. \quad (\text{B.1})$$

- 10: donde $e_i = 0$ si la muestra x_i es clasificada correctamente, $e_i = 1$ en otro caso, y $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
- 11: El clasificador final es:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{en otro caso} \end{cases} \quad (\text{B.2})$$

- 12: donde $\alpha_t = \log \frac{1}{\beta_t}$.
 - 13: Return V .
-

Algoritmo 5 Validación por color de piel.

-
- 1: Entrada: Posición de la ventana con rostro V , imagen RGB (640x480), base de datos de color de piel.
 - 2: Salida: color de piel validada $CP = True, False$.
 - 3: **if** color actual = color base de datos **then**
 - 4: **if** cantidad de color = 40% de los pixeles en V **then**
 - 5: $True$
 - 6: **else** $False$.
 - 7: **end if**
 - 8: **end if**
 - 9: **Return** CP .
-

Algoritmo 6 Reconocimiento de Usuario.

-
- 1: Entrada: imagenRGB, ejemplos de entrenamiento.
 - 2: Salida: base de datos del rostro del usuario.
 - 3: Las imágenes de entrenamiento se almacenan en el directorio de datos.
 - 4: El programa se entrena de los ejemplos de entrenamiento y crea una base de datos de Eigenfaces que se almacena otro archivo.
 - 5: La detección de rostros se realiza utilizando un clasificador de haarcascade.
 - 6: Si existe una base de datos de Eigenfaces, esta se carga. Sino existe, el programa intenta entrenar y crear la base de datos de Eigenfaces.
 - 7: Se valida si el rostro corresponde al usuario.
-

Algoritmo 7 Clasificación por color de vestimenta.

- 1: Entrada: imagen en RGB I (640x480), posición de rostro del algoritmo [4](#).
 - 2: Salida: zona de color Z .
 - 3: Convertir la imagen I de RGB a HSV.
 - 4: Ubicar ventana2 $V2$ debajo de la posición de ventana V .
 - 5: **for** $i = 1 : 256$ **do**
 - 6: $h(i) = 0, s(i) = 0, v(i) = 0$.
 - 7: **end for**
 - 8: **for** $i = 1:640$ **do**
 - 9: **for** $j = 1:480$ **do**
 - 10: $h(k+1) = V(i, j)$.
 - 11: $h(k+1) = h(k+1) + 1$.
 - 12: **end for**
 - 13: **end for**
 - 14: Definir rangos para cada variable: $Hue = \pm 5, Saturation = \pm 50, Value = \pm 45$.
 - 15: Encontrar zona con color predominante y definir centro C .
 - 16: Return C .
-

Algoritmo 8 Segmentación por distancia.

- 1: Entrada: imagen en Depth Id (640x480), centro de color C .
 - 2: Salida: zona segmentada S , localización del usuario L .
 - 3: Evaluar semilla $Seed$ en píxel en posición C con datos de profundidad $Seed = profundidadenI(Cx, Cy)$.
 - 4: Establecer rango $RS = Seed \pm 20$ cm.
 - 5: **for** $doi = (640x480)$
 - 6: **if** vecinos del píxel actual están dentro de RS **then**
 - 7: Agregar a vecindario S .
 - 8: **end if**
 - 9: **end for**
 - 10: Return S, L .
-

Algoritmo 9 Estimación de velocidad.

- 1: Entrada: localización del usuario L .
 - 2: Salida: velocidad estimada del usuario B .
 - 3: $H = \sqrt{Zu^2 + Xu^2}$.
 - 4: $\alpha = \tan^{-1}\left(\frac{Xu}{Zu}\right)$.
 - 5: $PoseX = \cos(\theta + \alpha) + Xr$.
 - 6: $PoseY = \sin(\theta + \alpha) + Yr$.
 - 7: **while** $L=True$ **do**
 - 8: Calcular distancia entre usuario y robot $d = (PoseX, PoseY)_{anterior} - (PoseX, PoseY)_{actual}$.
 - 9: Calcular tiempo transcurrido $t = t_{anterior} - t_{actual}$.
 - 10: $Velu = d/t$.
 - 11: $Velu_{anterior2} = P_0, Velu_{anterior1} = P_1, Velu_{anterior0} = P_2, Velu = P_3$.
 - 12: $B(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, \quad t \in [0, 1]$.
 - 13: **end while**
 - 14: Return B .
-

Algoritmo 10 Algoritmos de Navegación.

-
- 1: Entrada: posición del robot P_r , posición destino P_d , mapa m , Entorno z_t .
 - 2: Salida: trayectoria local y global Tl, Tg .
 - 3: Cargar parámetros y transformaciones de posición al marco referencial del mapa.
 - 4: Captura posición inicial x_{t-1} .
 - 5: Calcular odometría u_t y próxima posición x_t (Algoritmo [11](#)).
 - 6: Calcular errores del telémetro $p(z_t|P_r, m) = \prod_{k=1}^K p(z_t^k|P_r, m)$.
 - 7: Calcular la probabilidad de la posición (x, y) de los objetos más cercanos dentro del mapa.
 - 8: Por medio de la función MCLA localiza al robot dentro del mapa.
 - 9: Cuadricular mapa y calcular zonas libres en el mapa.
 - 10: Trazar varias trayectorias desde P_r hasta P_d .
 - 11: Evaluar la mejor ruta.
 - 12: Trazar trayectoria local.
 - 13: Return Tl, Tg .
-

Algoritmo 11 Algoritmos de modelo de odometría [113](#)

-
- 1: Entrada: u_t, x_{t-1}
 - 2: $\delta_{rot1} = atan2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$
 - 3: $\delta_{trans} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}$
 - 4: $\delta_{rot2} = \bar{\theta}' - \theta - \delta_{rot1}$
 - 5: $\hat{\delta}_{rot1} = \delta_{rot1} - sample(\alpha_1 \delta_{rot1} + \alpha_2 \delta_{trans})$
 - 6: $\hat{\delta}_{trans} = \delta_{trans} - sample(\alpha_3 \delta_{trans} + \alpha_4 (\delta_{rot1} + \delta_{rot2}))$
 - 7: $\hat{\delta}_{rot2} = \delta_{rot2} - sample(\alpha_1 \delta_{rot2} + \alpha_2 \delta_{trans})$
 - 8: $x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$
 - 9: $y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})$
 - 10: $\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$
 - 11: Return $x_t = (x', y', \theta')^T$
-

Algoritmo 12 Control PI+D Difuso.

- 1: Entrada: velocidad del usuario B .
 - 2: Declarar ganancias $T = 0.15, Kud = T, Kupi = T, K = 0.9, L = 10, Kp = 0.5, Kd = 0.3, Ki = 0.0005$.
 - 3: Calcular entradas $in1 = velB - veloreal, in2 = -in1, in3 = in1a, in4 = veloreal, in5 = veloreala, in6 = out2a, in7 = out3a, vnt = (in1 - in3)/T$.
 - 4: Calcular ecuaciones del controlador.
 - 5: $a = L * (Ki * in1 + Kp * vnt) / (2 * (2 * L - Ki * abs(in1)))$.
 - 6: $b = L * (Ki * in1 + Kp * vnt) / (2 * (2 * L - Kp * abs(vnt)))$.
 - 7: $am = L * (Ki * in1 + Kp * vnt) / (2 * (2 * L + Ki * abs(in1)))$.
 - 8: $bm = L * (Ki * in1 + Kp * vnt) / (2 * (2 * L + Kp * abs(vnt)))$.
 - 9: $c = (1/2) * (L + Kp * vnt)$.
 - 10: $d = (1/2) * (L + Ki * in1)$.
 - 11: $e = (1/2) * (-L + Kp * vnt)$.
 - 12: $f = (1/2) * (-L + Ki * in1)$.
 - 13: $g = 0, h = L, i = -L$.
 - 14: $a1 = L * (K * in2 - Kd * ynt) / (2 * (2 * L - K * abs(in2)))$.
 - 15: $b1 = L * (K * in2 - Kd * ynt) / (2 * (2 * L + K * abs(in2)))$.
 - 16: $c1 = (1/2) * (L - Kd * ynt)$.
 - 17: $d1 = (1/2) * (-L + K * in2)$.
 - 18: $e1 = (1/2) * (-L - Kd * ynt)$.
 - 19: $f1 = (1/2) * (L + K * in2)$.
 - 20: $g1 = 0, h1 = -L, i1 = L$.
 - 21: Calcular salida del controlador pi $upi = in6 + Kupi * (a + am + b + bm + c + d + e + f + g + h + i)$.
 - 22: Calcular salida del controlador d $ud = in7 + Kud * (a1 + b1 + c1 + d1 + e1 + f1 + g1 + h1 + i1)$.
 - 23: Calcular salida del controlador pi+d $upid = upi + ud$.
 - 24: Return upid.
-

Algoritmo 13 Control Difuso Sugeno.

```

1: Entrada: velocidad lineal de referencia  $upid$ , posición robot  $Pr$ , trayectoria deseada  $Td$ .
2: Definir variables  $Na = -1, N1 = -0.55, N2 = -0.25, Z = 0, P2 = 0.25, P1 = 0.55, Pa = 1$ .
3:  $orientation = atan((Yd - Yr)/(Xd - Xr))$ .
4:  $vusuario = velocidad(Xr, Yr)$ .
5:  $errorRad = orientation - Yawr$ .
6: Se calculan los pesos.
7:  $w1 = minimo(Mng, Mn)$ .
8:  $w2 = minimo(Mnp, Ma)$ .
9:  $w3 = minimo(Mnp, Mb)$ .
10:  $w4 = minimo(Mnp, Mn)$ .
11:  $w5 = minimo(Mz, Ma)$ .
12:  $w6 = minimo(Mz, Mb)$ .
13:  $w7 = minimo(Mz, Mn)$ .
14:  $w8 = minimo(Mpp, Ma)$ .
15:  $w9 = minimo(Mpp, Mb)$ .
16:  $w10 = minimo(Mpp, Mn)$ .
17:  $w11 = minimo(Mpg, Mn)$ .
18:  $w12 = minimo(Mpg, Ma)$ .
19:  $w13 = minimo(Mpg, Mb)$ .
20:  $w14 = minimo(Mng, Ma)$ .
21:  $w15 = minimo(Mng, Mb)$ .
22:  $sum = w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 + w10 + w11 + w12 + w13 + w14 + w15$ .
23: if  $sum > 0$  then
24:    $u0 = (w1 * Z + w2 * N1 + w3 * Z + w4 * N2 + w5 * Z + w6 * Z + w7 * Z + w8 * P1 + w9 * P2 + w10 * Z + w11 * Z + w12 * Pa + w13 * P1 + w14 * Na + w15 * N1) / sum;$ 
25: else  $u0 = 0$ .
26: end if
27: Return  $u0$ .

```

Algoritmo 14 Búsqueda de usuario.

```

1: Entrada: ubicación por color del usuario  $C$ , localización  $L$ , reconocimiento de usuario  $R$ .
2: Salida: usuario localizado  $B$ .
3: while  $t < 5min$  and  $R = False$  do
4:   while  $C = False$  do
5:     Velocidad angular  $w = 0.2$ .
6:   end while
7:   if  $L > 0.6$  then
8:     Velocidad lineal  $v = -0.2$ .
9:   end if
10: end while

```

Algoritmo 15 Supervisor.

```
1: Cargar mapa y localizarse dentro de él.
2: Iniciar algoritmos 2, 3, 4, 5.
3: Iniciar HRI.
4: if Detecta rostro = True then
5:     Esperar instrucción del usuario para iniciar recorrido.
6:     if Recibe instrucción= True then
7:         Iniciar algoritmos de localización y reconocimiento de usuario (6, 7, 8).
8:         while Usuario dentro de espacio HRI=True and Destino =False do
9:             Ejecutar algoritmos para la navegación (9,10, 12, 13).
10:        end while
11:        Ejecutar algoritmos de búsqueda (10, 14).
12:    end if
13: end if
```
