

## INTERFAZ DE COMUNICACIÓN, VISUALIZACIÓN Y CONTROL DE NAVEGACIÓN DE LA PLATAFORMA MÓVIL P-METIN.

### RESUMEN

Este artículo describe la interfaz de comunicación, visualización y control de navegación, implementada para generar trayectorias en el entorno de trabajo de la plataforma móvil P-METIN. Una de las características principales de dicha interfaz es que en cada trayecto recorrido por la plataforma se tiene información de la lectura de los sensores de abordó. Dicha interfaz se realizó utilizando GUI's de Matlab.

**PALABRAS CLAVES:** P-METIN Plataforma Móvil, para el Estudio de Técnicas Inteligentes de Navegación, Sensores.

### ABSTRACT

*This article describes the communication interface, visualization and navigation control, implemented in order to generate trajectories in the environment of work of the mobile platform P-METIN. One of the main characteristics of this interface is that in each itinerary traveled by the platform, there are information of the reading of the sensors. This interface was carried out using GUI's of Matlab.*

**KEYWORDS:** P-METIN Mobile Platform to the study of intelligent technical of navigation, Sensors.

### CARLOS A. RESTREPO

Estudiante X Semestre  
Ingeniería Eléctrica  
Universidad Tecnológica de Pereira  
cr@utp.edu.co

### LUIS HERNANDO RÍOS G

Ingeniero Electrónico, Ms.C  
Profesor Titular  
Universidad Tecnológica de Pereira  
lhgonza@utp.edu.co

### HERNANDO PARRA

Ingeniero Mecánico; Esp.  
Profesor transitorio  
Universidad Tecnológica de Pereira  
hparra@utp.edu.co

## 1. INTRODUCCIÓN

Son múltiples las aplicaciones de los robots móviles, teniendo en cuenta las diversas funciones que pueden ser encargadas a un robot, tanto en el campo industrial, como en el de experimentación e investigación. En este último los robots móviles son utilizados activamente.

La tendencia actual de los diferentes grupos de investigación en el área de robótica se orienta a la capacitación de un robot móvil para construir un mapa de su entorno de trabajo, así como de la navegación de manera autónoma, evitando obstáculos a través de él.

Una situación más deseable y más desafiante es que el robot móvil tenga la capacidad de generar y mantener su mapa de manera autónoma, mientras logra los objetivos propuestos.

Son muchas las herramientas que se han implementado en el logro de estos objetivos. Una de las estrategias propuestas se basa en el empleo de las técnicas inteligentes y la navegación basada en comportamientos.

El desarrollo de robots móviles capaces de construir su propio mapa y navegar de manera autónoma requiere de estrategias de control inteligente, que puedan manejar la incertidumbre presentada por el entorno de trabajo, al mismo tiempo que se desempeñen en tiempo real con una relativa baja carga computacional.

Los enfoques tradicionales para navegación de robots móviles requieren del uso de algoritmos de planificación con grandes cargas computacionales y modelos del mundo explícitamente predeterminados [1]. El sistema resultante exhibe un comportamiento robusto, pero con un gran gasto computacional y un pobre desempeño en tiempo real.

Adicionalmente, de la necesidad de significativa cantidad de memoria y potencia de procesamiento para almacenar y mantener modelos del mundo, los sistemas de control del robot móvil requieren de un sofisticado sistema de comunicaciones entre módulos que permitan una rápida respuesta ante las exigencias dinámicas del entorno y esto es un problema para los robots móviles cuyos recursos computacionales son llevados a bordo.

En lo que respecta a la navegación de robots móviles, es necesario tener en cuenta que:

-En condiciones iniciales no se conoce la localización del robot móvil en el entorno de trabajo.

-El conocimiento a priori del entorno de trabajo, es en general incompleto, incierto y aproximado, por ejemplo, los mapas típicamente omiten algunos detalles y características temporales y relaciones espaciales entre los objetos que pueden variar desde el momento en que el mapa fue construido. La información métrica puede ser imprecisa e inadecuada.

-La información perceptual adquirida en la mayoría de los casos es poco fiable, el limitado rango de los sensores combinado con las características del ambiente (oclusión) y las condiciones adversas de observación induce a ruido y a datos imprecisos y errores en el proceso de interpretación de las medidas.

-Los entornos reales son complejos y la dinámica impredecible, los objetos se pueden mover; otros agentes pueden modificar el ambiente y características que son relativamente estables, pueden cambiar con el tiempo.

-Las acciones de control no son completamente confiables, por ejemplo, las ruedas se pueden deslizar, etc.

Los trabajos presentados en Robótica Móvil para cubrir estas necesidades se orientan a complejos diseños de la mecánica de los robots y sus sensores, a la ingeniería del ambiente, o ambas cosas.

La ingeniería del ambiente o los complejos diseños del robot, incrementan los costos y reducen la autonomía de éste y además no pueden ser aplicados en todos los casos.

Si se quiere construir fáciles y versátiles robots, se tiene que aceptar la idea que dicha plataforma no puede ser demasiado sofisticada y que el ambiente no puede ser modificado.

El principal reto de la robótica móvil es la construcción de robustos programas de control que de manera confiable realicen complejas tareas a pesar de la incertidumbre del ambiente.

## 2. EL RETO DE LA NAVEGACIÓN AUTÓNOMA.

Cualquier aproximación al control de un sistema dinámico necesita utilizar algún conocimiento o modelo del sistema a controlar.

En el caso de un robot este sistema consiste del robot en sí, más el ambiente en el que opera. Desafortunadamente, mientras el modelo del robot como tal puede ser obtenido, la situación es diferente si consideramos a éste dentro de cualquier tipo de ambiente real considerado y sin ningún arreglo previo del entorno.

Los ambientes están caracterizados por la presencia de incertidumbre. La naturaleza de los fenómenos involucrados es tal, que frecuentemente no se está en la capacidad de precisar un modelo o cuantificar este fenómeno. Por ejemplo, si se considera la incertidumbre inducida por la presencia de gente en un ambiente; la gente se mueve alrededor y puede cambiar la posición de los objetos y los enseres del entorno; en la mayoría de los casos, sin embargo, no se puede obtener una distribución probabilística que pueda caracterizar estos eventos. La

interacción del robot con el ambiente causa dificultades similares.

El resultado del movimiento del robot y sus acciones de sensado son influenciadas por un número de condiciones ambientales, las cuales son difícilmente contabilizadas; por ejemplo, el error en el movimiento del robot puede variar como resultado de un piso mojado, la calidad del reconocimiento visual puede ser afectada por un rápido cambio de las condiciones de luz, la certeza de medida de distancia de un sensor ultrasónico es influenciada por la geometría y las propiedades de reflexión de los objetos en el ambiente.

Una estrategia para afrontar esta gran cantidad de incertidumbre es abandonar la idea de obtener un modelo completo del ambiente en la fase de diseño y dotar al robot con la capacidad de construir este modelo por sí mismo y en línea. Esta estrategia es llamada arquitectura Jerárquica. Saffiotti [14].

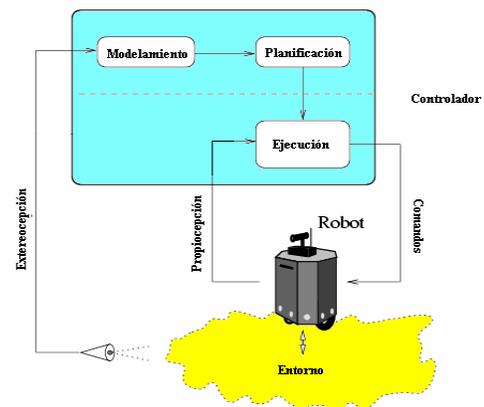


Figura 1. Arquitectura Jerárquica. La capa superior construye un modelo del ambiente y genera un plan de acción. La capa inferior ejecuta ciegamente este plan.

El robot utiliza los sensores exteroceptivos (Cámaras, sensores de ultrasonido, sensores de láser, infrarrojos, etc.), para observar el estado del ambiente. El robot utilizará los sensores propioceptivos (Brújula, encoders en las ruedas) para monitorear las partes de su propia estructura.

Utilizando los sensores propioceptivos, el robot adquiere un modelo del espacio de trabajo en el mismo instante en que la tarea esta siendo realizada. De este modelo, un programa planeador construye un plan que ejecutará la tarea dada en el ambiente dado. Este plan es pasado al programa de control de bajo nivel. Típicamente el procedimiento de ejecución es ciego. El controlador puede utilizar un modelo del robot y monitorear el estado de los efectores del robot (propiocepción), pero esto no permite sensar o modelar el ambiente una vez más.

No es difícil observar las limitaciones de la arquitectura jerárquica para trabajo en tiempo real en ambientes de trabajo. El modelo adquirido por el robot es necesariamente incompleto e inexacto debido a la incertidumbre en la percepción, mas aún el modelo rápidamente se desactualiza por la dinámica del ambiente y ya el plan construido puede ser inadecuado para el ambiente encontrado durante la fase de ejecución. Otro factor preponderante es que los procesos de modelado y planeamiento son computacionalmente complejos y consumen mucho tiempo de cómputo, con lo cual se agranda el problema: Intuitivamente una realimentación (Lazo de realimentación) con el ambiente puede mejorar estas limitaciones, es por ello que aparece el enfoque SMPA "Sense-Model-Plan-Act".

La complejidad del SMPA puede hacer que la respuesta temporal del sistema en ambientes dinámicos tarde incluso segundos.

Actualmente las investigaciones en robótica móvil tienden al desarrollo de numerosas y nuevas arquitecturas que integran Percepción y Acción. La tendencia general se orienta al planeamiento de pequeñas suposiciones sobre el ambiente encontrado en la fase de ejecución y que la ejecución sea sensible al ambiente y se adapte a las contingencias encontradas.

Para lograr esto, los datos de percepción han sido incluidos en la capa de ejecución (Arquitecturas Híbridas).

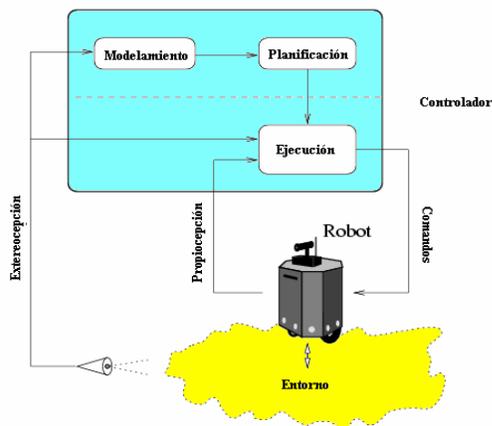


Figura 2. Arquitectura Híbrida. La capa de bajo nivel utiliza la percepción para adaptar dinámicamente el plan de ejecución a las contingencias del ambiente. El módulo de ejecución puede simultáneamente considerar solicitudes del plan de viaje como del ambiente.

La aparente extensión del sistema exteroceptivo al módulo de ejecución tiene 2 consecuencias importantes:

1. La interacción del robot con el ambiente es mucho más ceñida, dado que el ambiente está ahora incluido como lazo cerrado con la capa o módulo de ejecución.
2. La complejidad de la capa de ejecución ha sido incrementada debido a la necesidad de considerar múltiples objetivos: Alcanzar los objetivos tácticos tomados del planeador (Capa o fase de planificación) y reaccionar ante los eventos ambientales detectados por la percepción.

Muchos investigadores han seleccionado cubrir esta complejidad con la estrategia "Divide y Conquistarás", pero esto han avocado a diferentes estilos de división.

Algunos han perseverado dentro de la tradición jerárquica y tienen fuertemente seccionada la capa de ejecución en capas de secuencias y capas de proceso.

Otros parten de la arquitectura jerárquica y descomponen la capa de ejecución en pequeñas unidades de Decisión-Acción o comportamientos (Arquitectura Híbrida Basada en Comportamientos) [1,6].

Un comportamiento es una pequeña unidad de control hecha para alcanzar un objetivo simple en un restringido conjunto de situaciones. Los robots autónomos necesitan para poder realizar tareas complejas, de la activación y cooperación de un número de comportamientos. Cada comportamiento implementa en detalle un control seguro para una subtarea específica como seguir una trayectoria, evitar objetos sensados o cruzar puertas.

Las arquitecturas híbridas no resuelven el problema de navegación autónoma, pero ellas proporcionan una estructura conveniente en la cual los diferentes subproblemas pueden ser repartidos o integrados.

Desde el punto de vista experimental en la navegación basada en sensores, se trata de definir características técnicas como los tipos de sensores utilizados, cómo sensar el ambiente de trabajo, qué clase de estructura de control se debe proporcionar para el control del movimiento de la plataforma móvil.

La plataforma móvil diseñada está equipada con un anillo de sensores de ultrasonido, el cual le proporciona información de distancia. Esta información es implementada en algoritmos de navegación que permiten encontrar los espacios libres y mover la plataforma móvil a través de estos espacios hasta alcanzar su objetivo o meta. La posición objetivo se da con relación a la posición de inicio de la plataforma móvil. La plataforma móvil parará en la posición objetivo por medida realizada con su sistema odométrico.

La figura 3 muestra un escenario típico en el cual se desplaza la plataforma móvil.

**NAVEGACIÓN DE ROBOTS MÓVILES EN AMBIENTES DINÁMICOS**

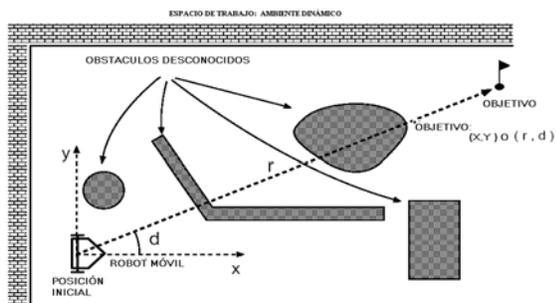


Figura 3. Escenario Típico donde se desplaza una plataforma móvil.

**3. INTERFAZ DE COMUNICACIÓN, VISUALIZACIÓN Y CONTROL DE LA NAVEGACIÓN DE LA PLATAFORMA MÓVIL.**

En la siguiente figura se muestra una foto de la plataforma.

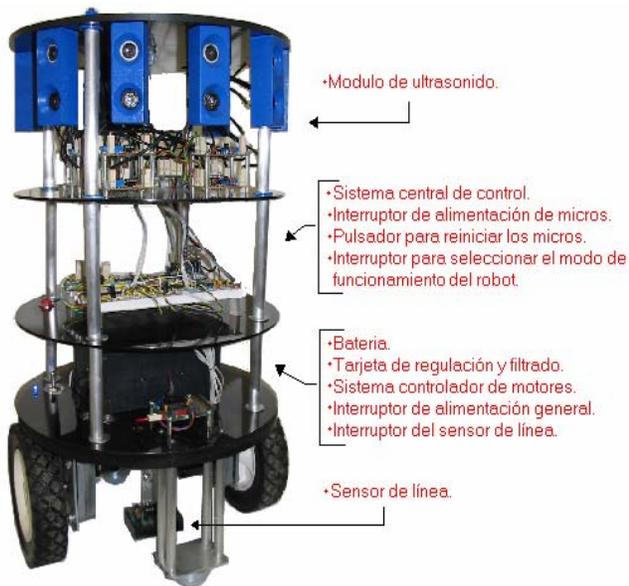


Figura 4. Plataforma Móvil.

Dicha plataforma se comunica a través de comunicación BLUETOOTH con un PC donde la interfaz de comunicación, visualización y control se hace en MATLAB.

La interfaz realizada permite la Generación de trayectorias y la simulación de estas a través de un entorno virtual donde P-METIN navega y grafica la trayectoria realizada. La siguiente figura muestra el panel frontal de la interfaz realizada.

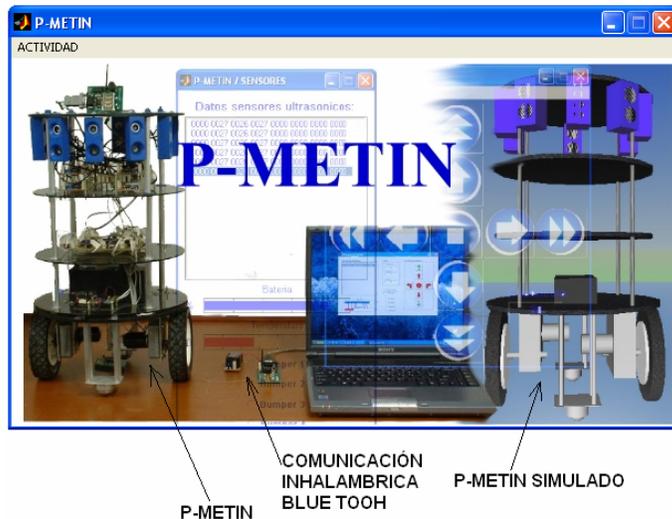


Figura 5. Panel Frontal.

Este panel da la opción en el menú ACTIVIDAD de :

- PLATAFORMA MÓVIL.
- ENTORNO VIRTUAL.
- AMBOS EN PARALELO.



Figura 6. Menú de Actividades.

**3.1. Funciones del menú plataforma móvil**

A su vez el menú PLATAFORMA MÓVIL, tiene unos Submenús de:

- ÓRDENES DE CONTROL
- SEGUIR RUTA
- DEAMBULAR

Cuando se activa el submenú ÓRDENES DE CONTROL entonces aparece el siguiente panel y se dan órdenes directas a P-METIN, el cual las ejecuta de manera instantánea.

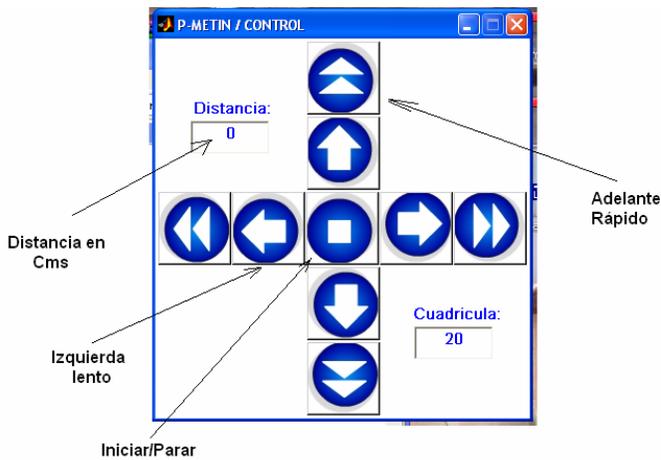


Figura 7. Control de Movimiento.

El submenú SEGUIR RUTA Permite generar una tabla con una ruta predefinida por el usuario de acuerdo a el entorno de trabajo, dicha tabla luego la ejecuta P-METIN.



Figura 8. Diseñar Ruta.

El submenú DEAMBULAR, le permite a P-METIN deambular por su entorno leyendo sensores y evitando obstáculos (por ahora sólo lee los frontales), deambula libremente sin rumbo fijo, evita obstáculos, hasta que una orden de PARAR detiene la plataforma, una vez la plataforma se detiene, grafica la trayectoria ejecutada durante su deambular. La siguiente figura muestra el panel DEAMBULAR y la grafica de trayectoria obtenida.

**NOTA:** Es importante resaltar que en cada uno de los SUBMENUS ejecutados siempre se esta generando una tabla con la lectura de todos los sensores ultrasónicos de ABORDO, es decir que en cada trayectoria se esta leyendo la información de distancia del entorno que circunda a la plataforma móvil

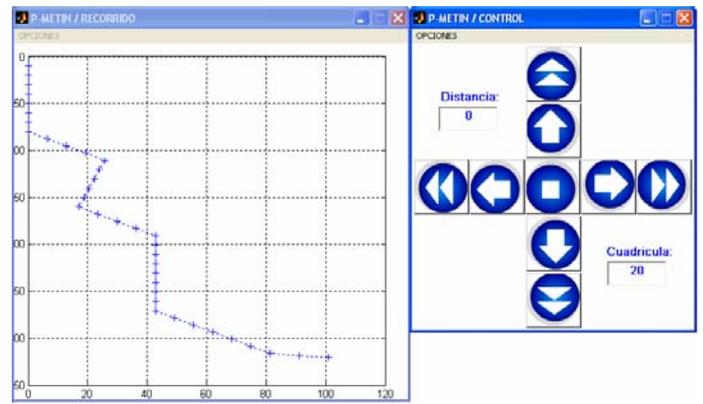


Figura 9. Deambular.

### 3.2 Funciones del menú entorno virtual

El menú ENTORNO VIRTUAL tiene los SUBMENÚS ÓRDENES de CONTROL y SEGUIR RUTA

En este menú se realizó un dibujo animado de la plataforma utilizando el toolbox V-Real Builder 2.0 de Matlab junto con la interfaz gráfica de MATLAB (GUI's) y se activan 2 submenús como son :

- ORDENES de CONTROL y
- SEGUIR RUTA

El submenú ORDENES DE CONTROL permite al programa de simulación activar trayectorias las cuales las ejecuta la animación, y a la vez las grafica en un plano en 2-D.

La siguiente figura muestra los paneles de dicho submenú

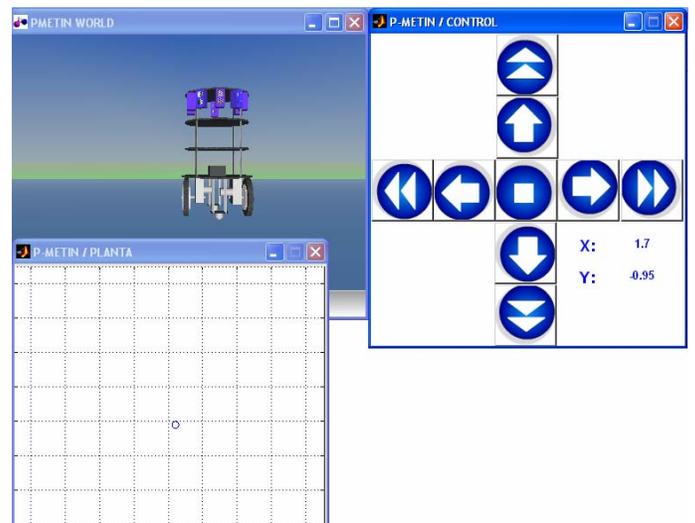


Figura 8. Entorno Virtual.

El submenú SEGUIR RUTA, le permite a la simulación realizar una trayectoria predefinida por el usuario a la vez que grafica dicha trayectoria en un plano 2 -D.

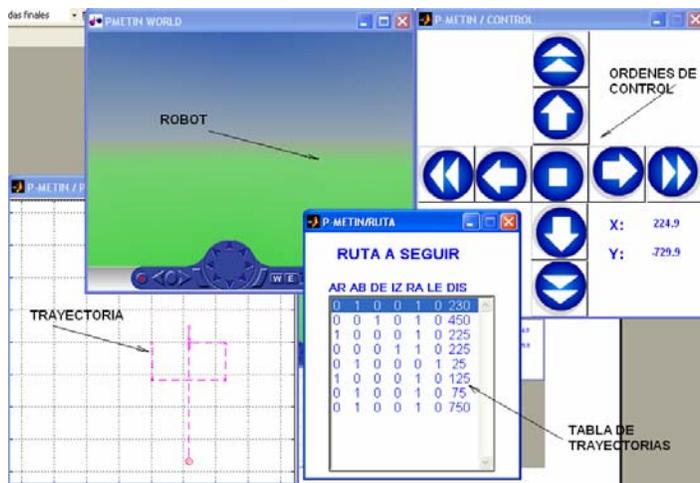


Figura 9. Seguir ruta.

#### 4. CONCLUSIONES

Se implementó una interfaz de comunicación, visualización y control de navegación, de la plataforma móvil (P-METIN) implementada.

Dicha interfaz permite el control de la navegación de P-METIN, así como la lectura de los sensores de abordo en cada trayecto recorrido, esta información es de vital importancia pues se puede utilizar para levantar mapas de entorno tanto locales como globales y generar estrategias de navegación.

Dicha interfaz permite generar trayectorias y controlar el movimiento de la plataforma, tanto real como en simulación.

#### 5. BIBLIOGRAFÍA

- [1] R. C. Arkin. Behavior-Based Robotics. MIT Press, Cambridge, MA, 1998.
- [2] J. W. Baxter and J. R. Bumby. Fuzzy logic guidance and obstacle avoidance algorithms for autonomous vehicle control. In Proc. of the Int. Workshop on Intelligent Autonomous Vehicles, pages 41-52, Southampton, UK, 1993.
- [3] H. Berenji, Y-Y. Chen, C-C. Lee, J-S. Jang, and S. Murugesan. A hierarchical approach to designing approximate reasoning-based controllers for dynamic physical systems. In Proc. of the Conf. on Uncertainty in Artif. Intell., pages 362-369, Cambridge, MA, 1990.
- [4] H. R. Berenji. The unique strength of fuzzy logic control. IEEE Expert, page 9, August 1994. Response to Elkan's "The paradoxical success of fuzzy logic", same issue.

[5] T. Bilgiç and I. B. Türksen. Measurement of membership functions: theoretical and empirical work. In H.J. Zimmermann, editor, Practical Applications of Fuzzy Technologies, volume 6 of Handbooks of Fuzzy Sets. Kluwer Academic, Norwell, MA, 1999.

[6] R. A. Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, RA-2(1):14-23, 1986.

[7] M. Drummond. Situated control rules. In Proc. of the Int. Conf. on Knowledge Representation and Reasoning, pages 103-113, 1989.

[8] E. Fabrizi, G. Oriolo, and G. Ulivi. Accurate map building via fusion of laser and ultrasonic range measures. In D. Driankov and A. Saffiotti, eds, Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, 2000, pages 257-280.

[9] J. Gasos. Integrating linguistic descriptions and sensor observations for the navigation of autonomous robots. In D. Driankov and A. Saffiotti, eds, Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, 2000, pages 313-340.

[10] E. Gat. Three-layer architectures. In R.P. Bonasso D. Kortenkamp and R. Murphy, editors, Artificial intelligence and mobile robots, chapter 8, pages 195-210. MIT Press, Cambridge, MA, 1998.

[11] R. Ghanea-Hercock and D. P. Barnes. An evolved fuzzy reactive control system for cooperating autonomous robots. In Proc. of the Int. Conf. on Simulation and Adaptive Behavior (SAB), Cap Cod, 1996.

[12] J. Godjevac and N. Steele. Neuro fuzzy control for basic mobile robot behaviours. In D. Driankov and A. Saffiotti, eds, Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, 2000, pages 98-118.

[13] S. G. Goodridge and M. G. Kay. Multi-layered fuzzy behavior fusion for reactive control of autonomous robots. In D. Driankov and A. Saffiotti, eds, Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, 2000, pages 179-204.

[14] D. Diankov and A.Saffiotti, eds. Fuzzy Logic Techniques for Autonomous vehicle Navigation. Springer-Physica Verlag, DE, 2001. Pages 3-24.

[15] H. Kiendl and J. J. Rüger. Stability analysis of fuzzy control systems using facets functions. Fuzzy Sets and Systems, 70:275-285, 1995.