

A mis hijos Sebastián Alberto y Daniela Ivonne

Agradecimientos

Deseo agradecer al **Instituto de Investigaciones Eléctricas** (IIE) su apoyo y confianza depositada en mí para llevar a buen término mi programa de Maestría en Inteligencia Artificial. Asimismo, me permito destacar el papel del **Consejo Nacional de Ciencia y Tecnología** (CONACyT) como promotor para la formación de recursos humanos, y al cual le agradezco su participación financiera durante mi desarrollo académico.

Quiero expresar mi inmensa gratitud a la **Universidad Veracruzana** y al **Laboratorio Nacional de Informática Avanzada** (LANIA) por haberme abierto las puertas de la Maestría en Inteligencia Artificial. Gracias al trabajo conjunto de estas instituciones y a su preocupación por ofrecer un programa de excelencia, he tenido la oportunidad de hacer investigación seria en Inteligencia Artificial bajo la tutoría de una gran plantilla de profesores investigadores, nacionales y extranjeros.

Finalmente quisiera referirme al **Dr. José Negrete Martínez** como una de mis más grandes motivaciones en el quehacer científico. No deja de sorprenderme como uno de los exponentes más representativos de la nueva IA, le haya confiado plenamente a uno de sus estudiantes el diseño, implantación y prueba de una arquitectura con ingredientes de la IA clásica. Por lo tanto, solo me resta agradecerle orgullosamente su confianza y manifestarle mi admiración por su obra en esta comunidad.

MAESTRIA EN INTELIGENCIA ARTIFICIAL

UNIVERSIDAD VERACRUZANA - LANIA



Guía de Conductas Reactivas mediante Planes Abstractos en un Problema de Navegación Robótica

Tesis de Maestría

*Autor (MIA UV / LANIA): Alberto Reyes Ballesteros.
Asesor Interno (MIA-UV): José Negrete Martínez.
Revisor Externo (LANIA): Homero Ríos Figueroa
Revisor Externo (University of Iowa): Ronald Vogel
Revisor Externo (Universidad Simón Bolívar): José Ramírez*

Contenido

Introducción	4
Capítulo 1. Análisis del Problema	6
1.1 Navegación Robótica	6
1.2 Breve Reseña de Arquitecturas	7
1.2.1 Sistemas Deliberativos	7
1.2.2 Sistemas Reactivos	8
1.3 Sistemas Híbridos Reactivos / Deliberativos	9
1.3.1 Evidencia Psicológica	9
1.3.2 Evidencia Neurocientífica	10
1.3.3 Unificación	10
Capítulo 2. Desarrollo de la Propuesta	12
2.1 Planes Abstractos como Guías de Conductas	12
2.1.1 Los planes como Programas	12
2.1.2 Los planes como Intenciones	12
2.2 Subsumción	14
2.2.1 Inspiración Biológica	14
2.2.2 La Arquitectura de Brooks	14
2.2.3 Un Ejemplo	16
2.3 Arquitectura Híbrida	17
2.3.1 Configuración General	17
2.3.2 Planes, Planificación y Seguimiento del Plan	19
Capítulo 3. Marco de Trabajo	21
3.1 Estudio del Caso	21
3.1.1 Descripción del Problema	21
3.1.2 Descripción del Agente en Términos PAGE	22
3.2 Preparación del Kit Robótico Rug Warrior (RW)	24
3.2.1 Kit Básico	24
3.2.2 Expansión del Kit	27

3.3	Sistema Perceptual RW	30
3.3.1	Fotoceldas	31
3.3.2	Fototransistores	32
3.3.3	Detectores de Proximidad	33
3.3.4	Sensores Táctiles	34
3.3.5	Micrófono	36
3.3.6	Sonar	37
3.3.7	Propiocepción	38
3.4	Ambiente Artificial AMBART-RW	39
	Capítulo 4. Implantación Técnica	42
4.1	Mecanismo de Solución	42
4.2	Abstracción Jerárquica del Problema	42
4.2.1	Operadores Conductuales	43
4.2.2	Planificador Jerárquico PLANABS	43
4.3	Descomposición de Operadores	45
4.3.1	Redes de Conductas	46
4.3.2	Conductas Primitivas	48
	Capítulo 5. Análisis de Resultados	52
5.1	Resultados Experimentales	52
5.1.1	Proceso de Planificación	52
5.1.2	Ejecución de Conductas	53
5.1.3	Evoluciones del Agente en Forma Global	56
5.2	Trabajo Futuro	57
5.2.1	Mejoramiento del Prototipo	57
5.2.2	Incorporación del Agente en Ambientes Reales	59
5.3	Conclusiones	60
	Créditos	63
	Apéndice	64
	Lista de Figuras	65
	Referencias	67

Introducción

A pesar del avance de los sistemas de control de procesos en la industria, y a los altos niveles de automatización en la vida diaria, aún existen actividades que por su complejidad deben ser realizadas por los humanos. Actividades como la supervisión y el muestreo en plantas de procesos y plataformas submarinas, la detección de explosivos, el manejo de sustancias tóxicas, etc.; todas éstas elevan el factor de riesgo del personal que las realizan. El surgimiento de la Robótica Móvil pretende relajar este tipo de problemas con la incorporación de una característica carente en los sistemas tradicionales, la inteligencia artificial. La evolución e implantación de este concepto, desde el polo simbólico [Newell 61] hasta el de la no representación uniforme, explícita e interna de las capacidades del sistema, el mundo exterior y las metas deseadas [Brooks 91], han destacado deficiencias y virtudes de ambos polos.

Una integración de estas ideas requiere la confrontación de muchos problemas. Es claro ver que cada uno de los polos mencionados conduce a subconjuntos diferentes de dificultades para llevar a cabo la resolución de un mismo problema. Es nuestra tesis que ninguna de estas propuestas es completamente satisfactoria cuando se usa aisladamente, una combinación bien balanceada de estos conceptos promete dotar de autonomía, inteligencia, robustez y flexibilidad a un agente físico para la realización de tareas como las expuestas inicialmente.

El presente trabajo tiene por objeto proponer una arquitectura que incorpore conductas reactivas o automáticas a cierta capacidad de razonamiento abstracto en la resolución de un problema de navegación típico. El diseño de tal arquitectura involucra áreas de interés de la IA, como son: Planificación, Programación de Agentes, y Robótica Móvil cuyos exponentes han sentado las bases de la nueva IA. Entre ellos destacan los trabajos de J. C. Latombe, Nils Nilsson, Scott Benson, Pattie Maes, Rodney A. Brooks, Karl Pfleger, Barbara Hayes-Roth, Kristian Hammond, etc.

Para ilustrar el mecanismo de solución a un problema de navegación empleando la arquitectura *híbrida* propuesta se ha construido un pequeño ambiente artificial que recuerda algunas características del local donde ocurren fugas de vapor en una central nuclear mexicana. Nuestra tesis no sólo pretende la *hibridación* mencionada, también pretende establecer bases sólidas para el diseño de robots inteligentes y autónomos con fines de supervisión y adquisición de datos en ambientes peligrosos o en tareas de navegación monótonas o pesadas.

La incorporación física de un agente con capacidades perceptuales y motrices robustas hace necesaria la participación de otras disciplinas como la ingeniería eléctrica, mecánica, electrónica, y las ciencias de la computación, lo cual permitirá a estudiantes e investigadores de la Maestría en Inteligencia Artificial contar con una herramienta de laboratorio más para asistir la investigación en IA y otras disciplinas relacionadas. Por su parte, la solución satisfactoria a problemas reales similares fomentará el vínculo entre el sector académico con otros dominios como el industrial, productivo, militar, etc.

Este trabajo se organiza de la siguiente manera: La primera sección está dedicada al análisis de la navegación robótica vista como un problema aún no resuelto en IA. En esta sección se hace un breve recuento de las arquitecturas más representativas que han intentado dar solución a la navegación robótica bajo conceptos diferentes. Se mencionan trabajos relacionados con los sistemas reactivos y deliberativos. Asimismo, se ejemplifican algunas propuestas que han intentado combinar ambas escuelas.

La segunda sección inicia el desarrollo de una nueva propuesta que intenta reunir algunos elementos de varios trabajos relacionados con el área e introducir el concepto de los planes como intenciones para relajar el diseño de los agentes inteligentes. En esta sección se describe también la arquitectura más representativa de la escuela reactiva conocida como subsumción, se introduce al concepto de los planes abstractos como guías de conductas y se muestra la configuración de una arquitectura híbrida que conjunta ambas propuestas.

La tercera sección muestra el marco de trabajo a través de una aplicación inspirada en un hecho real, la navegación robótica en ambientes industriales. Aquí se describe la configuración general del agente en términos PAGE [Russell 95]; se ilustra el sobrediseño de un kit robótico comercial para los efectos deseados; se explica cuidadosamente el sistema perceptual del kit; y se muestra el diseño de un pequeño ambiente artificial emulando algunas características de un problema de navegación real.

En la cuarta sección se describe la implantación técnica de la arquitectura propuesta. Aquí se ilustra el algoritmo de un planificador jerárquico empleado como mecanismo para la selección de operadores de alto nivel y parámetros de búsqueda del agente. Por su parte, se explica un método para la descomposición de los operadores conductuales y el papel de la arquitectura de subsumción para el control de conductas en competencia durante el seguimiento del plan.

La sección quinta por su parte está dedicada a evaluar la arquitectura bajo tres aspectos fundamentales: la planificación abstracta, la ejecución de conductas, y las evoluciones del agente en forma global durante la resolución del problema de navegación tipo. En esta misma sección se hace una breve exposición del trabajo futuro para mejorar el prototipo e incorporar al agente en ambientes reales. Finalmente se establecen las conclusiones destacando las ventajas y desventajas de la arquitectura.

Capítulo 1. Análisis del Problema.

1.1 Navegación Robótica

Por naturaleza, los seres vivos están dotados de la habilidad para navegar en su ambiente. Los salmones, por ejemplo, pueden localizar el lugar donde depositan sus huevos a miles de kilómetros de distancia, las palomas pueden hallar su destino aun en días nublados, los abejorros pueden imprimir en el ambiente la trayectoria a una fuente alimenticia en respuesta a un mensaje de otra abeja. Sin embargo, pocos robots móviles pueden alcanzar todos sus objetivos sin necesidad de marcas de referencia artificiales en el ambiente. Por tal motivo, la navegación robótica se ha considerado como un problema aún no resuelto en Inteligencia Artificial [Jones 93].

De acuerdo con [Everett 95] los métodos de navegación artificial actualmente utilizados que más se acercan a una solución se pueden clasificar en tres grandes grupos:

- **Control Reactivo.** Estrategia basada en conductas que acopla información sensorial en tiempo real con acciones motrices, sin la intervención de representaciones simbólicas para el modelado parcial o total del ambiente operativo de un robot. El control reactivo ajusta bien en los ambientes dinámicamente cambiantes.
- **Sistemas Deliberativos.** Involucran el acoplamiento de sensores para la detección de objetos con algún tipo de capacidad absoluta para modelar el mundo. Se emplean distintos paradigmas para la planificación de la trayectoria deseada. La principal ventaja de estos sistemas sobre el control reactivo es que, de existir, se garantiza la determinación de un camino franco a la meta.
- **Enfoques Híbridos Reactivos / Deliberativos.** Combinan ambas metodologías para lograr una operación más robusta en ambientes dinámicos.

Los sistemas reactivos pueden usarse como una forma de navegación restrictiva que ajusta bien en tareas y conductas específicas, el problema es que carecen de la adaptabilidad suficiente para producir conductas generales y versátiles. A su vez, cuentan con elementos de acción y percepción muy variados pero ninguno de cognición, lo que limita a los robots a formas de vida de muy bajo nivel. Gracias al conocimiento y un plan, un robot podría destacar sus conductas y extender sus aplicaciones a dominios más significativos.

Este trabajo considera que el éxito de un sistema robótico móvil descansa en los sistemas híbridos reactivos / deliberativos. Ninguna corriente en forma aislada es superior a otra, cada una resuelve distintas partes de un problema. La idea es que el elemento reactivo este relacionado con la ejecución de un plan, y la deliberativa con la determinación de objetivos y la formulación del plan para alcanzarlos.

1.2 Breve Reseña de Arquitecturas

Con el objeto de mostrar algunos ejemplos representativos de las propuestas que conforman los sistemas híbridos reactivos / deliberativos, a continuación se muestran algunos trabajos realizados en cada área.

1.2.1 Sistemas Deliberativos

Todo el trabajo significativo de Robótica en el marco de los sistemas deliberativos tiene su inicio con Shakey [Nilsson 69], el robot creado por el *Stanford Research Institute* a finales de los 60s. Las versiones iniciales de Shakey se basaban en la construcción de planes completos mediante métodos de demostración de teoremas; los planes eran seguidos por el robot sin verificarlos durante la ejecución, lo que traía como consecuencia una falla casi inevitable de los planes, incluso con pequeñas trayectorias.

Por su parte, la arquitectura deliberativa de Shakey, basada en la demostración de teoremas con conocimiento representado explícitamente, era de pobre desempeño y creaba un efecto de balanceo en el robot al tener que detenerse para la ejecución de las acciones subsiguientes. Este último problema fue resuelto por Shakey delegando las operaciones conocidas (secuencias de acciones) a un conjunto de acciones de nivel intermedio (ILA), las cuales a su vez eran consecuencia de la secuenciación de acciones de bajo nivel. Esta elevación del nivel de abstracción del conjunto de acciones permitió contar con un planificador un poco más eficiente y robusto, el cual estaba basado en STRIPS [Fikes 71].

STRIPS avanzó un poco la arquitectura de Shakey al introducir el concepto de macro operador que no es más que la compilación del resultado del proceso de planificación y tener comportamientos incrementales mejores en el robot en condiciones rutinarias.

El cambio definitivo lo brindó PLANEX [Fikes 72], al incorporar el concepto de precondition para los macro operadores y monitorear el proceso de ejecución. De esta manera, se verificaba el estado actual y de no existir se invocaba a STRIPS para generar otro plan.

En resumen, la arquitectura de Shakey se basaba en un sistema de planificación jerárquica (con distintos niveles de abstracción), un modelo del ambiente y el monitoreo de la ejecución del plan. Algunos autores han heredado parte de estos elementos para construir diversas arquitecturas, y los avances significativos tienen que ver con la precisión de las acciones y el grado de representación del mundo. Tal es el caso de J. C. Latombe, quién plantea tres enfoques básicos de planificación de movimientos robóticos en [Latombe 91]:

- **Navegación métrica**, la cual hace uso de particiones del ambiente mediante celdas o grafos que discretizan el espacio de búsqueda.
- **Navegación por hitos**(*landmarks*), que persigue la determinación de la posición del robot basándose en ciertas ubicaciones predeterminadas y la utilización de sensores y/o mapas topológicos.

- **Enfoques probabilísticos**, los cuales asumen un grado de incertidumbre en la determinación del estado del ambiente la cual se compensa con modelos que complementan a los sensores. Aquí están los enfoques que asumen la accesibilidad parcial del ambiente tales como POMDP (*Partially Observable Markov Decision Problems*).

Como se ha mencionado, estos enfoques se ven a su vez modificados por el grado de conocimiento que el robot tenga del ambiente. Si el conocimiento tiende a ser completo, entonces las técnicas de partición del ambiente en celdas o grafos puede ser adecuado; en el caso extremo en el que no existe conocimiento del ambiente, entonces debe pensarse en algoritmos de los denominados *on-line*, cuyo horizonte de planificación tiende a ser mínimo.

1.2.2 Sistemas Reactivos

Los sistemas reactivos se originaron durante el movimiento cibernético de 1940. Grey Walter [Walter53] desarrolló una tortuga electrónica capaz de moverse en el mundo, evadiendo umbrales ambientales percibidos y atraído por ciertos objetivos. Posteriormente se incluyeron otras metas como una estación de recarga. Cuando decrecía su potencia, la tortuga era atraída a un puerto con el recargador, y una vez adquirida la energía necesaria era repelida por éste. En tal caso, no había las construcciones representacionales de la IA clásica; la percepción controlaba directamente las acciones motrices.

Braitenberg revivió el interés por estas criaturas [Braitenberg 84], y demostró que éstas podían manifestar conductas comparables a las de los animales tales como: cobardía, agresión, amor, exploración y lógica. Sus experimentos mostraron la posibilidad de obtener conductas complejas a partir de transformaciones senso-motoras.

Rosenschein y Kaelbling plantearon una arquitectura basada en el concepto de autómatas situado que eliminaba toda representación explícita y deliberación [Rosenschein 85, 90]. La generación de estos autómatas podía darse por dos métodos básicos: compilación fuera de línea basada en representaciones explícitas, y descomposición del autómata en comportamientos. Ambos métodos tenían la idea de generar un autómata finito conectado a los sensores y actuadores del robot a través de estados.

Rosenschein pudo demostrar que su compilador de autómatas finitos generaba estados internos que correspondían a proposiciones lógicas acerca del ambiente del robot, de esa forma se podía afirmar que el agente poseía cierto conocimiento de su ambiente sin necesidad de una representación explícita de las proposiciones lógicas que lo sustentaban.

Brooks es considerado como el líder más reciente del paradigma puramente reactivo. Su grupo impulsó este enfoque con el desarrollo de la arquitectura de subsumción [Brooks 86]. Brooks rompió con el esquema general de la IA clásica de los 70s y 80s (senso-plan-acto) y puso en entredicho el papel de la representación del conocimiento en toda la IA. La arquitectura de subsumción tiene una inspiración biológica en el sentido conductual, tomando como modelo la semejanza con algunas conductas en los insectos pero sin considerar los mecanismos

biológicos que las producían.

Casi a la par que aparecía la arquitectura de subsumción, otros investigadores se interesaron en incorporar los resultados de los etólogos y neurocientíficos a los sistemas robóticos. Por ejemplo, Arkin explotó estos modelos con el propósito de construir sistemas inteligentes empleando esquemas de interacción en el diseño de sistemas de control robótico reactivos [Arkin 90a].

1.3 Sistemas Híbridos Reactivos / Deliberativos

Antes de mencionar las bases biológicas que sustentan los sistemas híbridos reactivos / deliberativos, cabe mencionar que existe otro tipo de métodos relacionados con la solución del problema de la navegación, a los que algunos autores se refieren como **ingeniería trascendental**. Estos investigadores se mantienen al margen de las soluciones basadas en los sistemas biológicos y consideran que no deben restringir su trabajo a tales métodos. Un ejemplo representativo de la ingeniería trascendental es el desarrollo de un control dinámico para el seguimiento automático de caminos por el grupo Dickmann de Munich, Alemania. Aunque existen argumentos en contra de la búsqueda de soluciones basada en los sistemas biológicos (ej. que los robots usan fundamentalmente distinto hardware, o que la evolución quizá no sea el mejor instructor), se puede ganar mucho soporte técnico comprendiendo la estructura de estos sistemas.

1.3.1 Evidencia Psicológica

Estudios psicológicos [Shiffrin 77] han mostrado la existencia de dos modos distintos de conducta: el intencional (controlado) y el automático (reflejo). Norman y Shallice enumeran en [Norman 86] las características de las tareas que requieren del modo conductual intencional (recursos atencionales deliberados) de la siguiente manera:

- Planificación (toma de decisiones)
- Resolución de conflictos
- Acciones novedosas o mal aprendidas (no reflexivas)
- Acciones peligrosas o difíciles
- Superación de hábitos y tentaciones

Otras acciones motoras normalmente son automáticas y pueden ocurrir sin atención. El modelaje de conductas coordinadas se lleva a cabo mediante un mecanismo de agenda para esquemas motrices múltiples. Los procesos de niveles más altos que requieren de la atención, alteran los umbrales de los esquemas (relajadamente traducidas como conductas básicas) para variar dinámicamente la interacción entre ellos. Este modelo incorpora canales de control verticales y horizontales (en la misma forma que los usa Brooks). Las diversas conductas horizontales son mediadas por canales verticales que interconectan las conductas y les

permiten una modulación dinámica como resultado de los recursos atencionales (planificación, resolución de conflictos, etc.). Los esquemas se disparan mediante eventos perceptuales pero se modulan con procesos atencionales. Esto proporciona un modelo psicológico coherente para la integración de conductas actuales múltiples que están controladas por un procesamiento de más alto nivel.

El modelo Norman-Shallice muestra varias interconexiones entre el control automático y deliberado:

- La selección del esquema es la función principal del control deliberado (los canales verticales proporcionan el mecanismo de selección).
- Los esquemas (tareas conductuales) están en competencia unos con otros.
- Existen experimentos neurocientíficos consistentes con este modelo

Aun si finalmente esta teoría resultara errónea para sentar las bases de la conducta motriz y la planificación psicológica, el modelo ofrece una prueba útil de la posible integración de los sistemas deliberativos y el control reactivo en un robot.

1.3.2 Evidencia Neurocientífica

El control del nivel superior sobre los diversos subsistemas independientes está también bien soportado por la literatura neurocientífica. Hay varias instancias donde la coordinación de la locomoción está determinada por neuronas autorítmicas usadas para formar el patrón de acción motriz. Las neuronas motrices, cuyo disparo se invoca desde el procesamiento neuronal del nivel superior, activan estos patrones conductuales automáticos [Ewert 80].

Los estudios experimentales de Humphrey de la Universidad de Emory han aclarado la interconexión entre el control motriz local de bajo nivel y su modulación desde procesos del nivel superior. Analizando las señales emitidas por el sistema nervioso central, los circuitos neuronales espinales, y la inervación local de los músculos al probar la acción de sujetar en los monos, se pudieron observar distintas funciones del control. El control local, en esencia, manejaba los procesos de actuación mientras el procesamiento del nivel superior proporcionaba las señales para adaptarse a cambios de posición importantes.

1.3.3 Unificación

Las arquitecturas híbridas permiten la reconfiguración de los sistemas de control reactivo basados en el conocimiento disponible del mundo, dotándolos de mayor flexibilidad que la de los sistemas puramente reactivos. La reconfiguración dinámica de los sistemas basados en la deliberación (razonamiento sobre modelos del mundo) mejora sustancialmente el diseño de robots de propósito general.

Debe reconocerse que los sistemas robóticos puramente reactivos no son apropiados en todas las aplicaciones robóticas. En situaciones donde el mundo deba modelarse con precisión,

donde haya incertidumbre restringida, y exista cierta garantía de que virtualmente no habrá cambios en el ambiente durante la ejecución, frecuentemente se prefieren los métodos deliberativos, ya que es posible llevar a cabo efectivamente un plan. En el mundo real, donde funcionan los agentes biológicos, estos prerequisites para los planificadores puramente deliberativos simplemente no existen. Si los roboticistas esperan contar con máquinas operando en el mismo ambiente que los seres humanos, se requiere de los métodos de control reactivo. Muchos creen que los sistemas híbridos capaces de incorporar tanto razonamiento deliberativo como ejecución reactiva son necesarios para aprovechar todo el potencial de los sistemas robóticos.

Arkin fué uno de los primeros en defender el uso de ambos sistemas de control como arquitectura de un robot autónomo. Incorporando un planificador tradicional capaz de razonar con un sistema de control reactivo modular y flexible, le fue posible obtener configuraciones robóticas específicas que integraban elementos conductuales y perceptuales, y conocimiento ambiental **apriori** [Arkin 90b]. Tal sistema fue probado en varias aplicaciones, tanto en interiores como en exteriores.

Gat propuso un sistema híbrido de tres niveles llamado Atlantis [Gat 92], el cual incorporaba un deliberador basado en Lisp, un secuenciador que manejaba las fallas del sistema reactivo, y un controlador reactivo. Este sistema fue probado y utilizado exitosamente en los prototipos para la navegación en Marte.

Nilsson presentó también un formalismo híbrido conocido como Programación Teleo-Reactiva [Nilsson 94] para la organización y ejecución de acciones en agentes que operan en ambientes dinámicos. El comportamiento que se pretende con este formalismo es aquel que hace que un agente reaccione rápida y apropiadamente a situaciones típicas, y siempre orientado a la meta. Esta mezcla de control local basado en ciclos cortos percepción-acción con la conducta orientada a metas explica el nombre de Teleo (fines, metas, etc.) - Reactivo.

Capítulo 2. Desarrollo de la Propuesta.

2.1 Planes Abstractos como Guías de Conductas

Un plan puede tomar diferentes formas, dependiendo de la relación deseada con cierta conducta. De acuerdo con la definición de plan podemos establecer dos enfoques principales:

1. Cualquier esquema detallado, programa o método desarrollado previamente para alcanzar un objetivo.
2. Un proyecto o, meta propuesta o tentativa.

Según Agre y Chapman [Agre 90] a estos enfoques se les puede llamar *planes como programas* y *planes como intenciones*, respectivamente.

2.1.1 Los Planes como Programas

Desde este punto de vista, los planes son programas ejecutables compuestos por acciones primitivas. Por lo tanto, la planificación es un tipo de programación automática, y el seguimiento del plan se reduce a la simple ejecución directa de acciones primitivas. Este modelo ha sido usado en diversas formas por los investigadores dedicados a la planificación. Los planes de la IA clásica imponían una estructura secuencial a sus acciones constitutivas [Russell 95]; esto es, un agente ejecutaba secuencialmente sus planes sin verificar el efecto de sus acciones. En otros trabajos [Agre 87], los planes tenían una estructura menos secuencial y más reactiva para garantizar el alcance de objetivos bajo diversas condiciones ambientales. En este último trabajo, los agentes ejecutaban sus planes en un ciclo de percepción- acción muy breve. Otros investigadores combinaron estas técnicas [Drummond 93], [McDermott 92], aunque aún prevalecía la representación de los planes como programas ejecutables. Existen otros modelos de planificación menos investigados como la planificación condicional donde el sensado del ambiente es menos frecuente y más deliberado.

En tareas no muy triviales, la representación de toda las situaciones ambientales relevantes es poco práctica y problemática [Ginsberg 89]. Una vez pasando de acciones nominales en ambientes artificiales a acciones reales en ambientes naturales, la semántica de las acciones individuales puede variar considerablemente [Suchman 87]. Peor aún, estas limitaciones son tan solo las que surgen asumiendo un modelo válido. Los planes como programas, tal como lo ejemplifica la IA clásica, presuponen que un agente tiene capacidades conductuales fijas y una base de conocimiento estático. También asumen que las diversas metas del agente y el criterio de objetivo son estáticos. Estas suposiciones son irrealistas ya que un agente debe adaptar su conducta cuando estos factores cambien.

2.1.2 Los Planes como Intenciones

Desafortunadamente, y a pesar de la fuerte intuición compartida por muchos investigadores, la mayoría coincide en que los planes, la planificación, y el seguimiento de un plan no son más

que un modelo alternativo prescriptivo. Muchas metáforas para los planes desenfatan el control directo de los planes sobre la conducta considerándolos como simples consejeros. Sin embargo, es muy difícil encontrar modelos precisos y representaciones detalladas de los planes para afectar la conducta de un agente real.

Pfleger y Hayes-Roth consideran en [Hayes 96] la posibilidad de utilizar algunas intuiciones acerca de los planes que no son programas en términos de las intenciones de un agente. De acuerdo con esto, un plan sólo guía sin determinar las acciones específicas de un agente. Por esto, el agente no tiene que ejecutar directamente sus planes, sino conductas consistentes con el plan. El uso adecuado de los planes con este enfoque permite a un agente reducir la complejidad para controlar su ejecución y restringir la búsqueda para alcanzar sus metas. Algunos trabajos usan parte de estas ideas para controlar la conducta de un sistema: [Chrisman 91], [Firby 94], [Gat 92], [Johnson 87], [Tommelein 91]. En resumen, la idea de los planes como intenciones consiste en que un plan abstracto describa la tarea a realizar y no el método a aplicar; de tal suerte que este plan sólo guíe o inflencie una conducta reactiva.

Algunas características del modelo surgen observando ciertos aspectos importantes que el diseño del agente debe considerar. Estos aspectos se pueden clasificar en dos categorías:

1. Reducción del uso de conductas estáticas
2. Reutilización del conocimiento disponible

La característica particular de esta propuesta es la posibilidad de reducir el uso de conductas estáticas y de reutilizar el conocimiento disponible. Entre más complejo es un agente es menos deseable el mantenimiento de conductas estáticas. Por lo tanto es muy importante generar o seleccionar adecuadamente un patrón de conductas donde no existan elementos conductuales cuyo disparo este sujeto a situaciones poco probables o imposibles. Hayes-Roth plantea un esquema muy interesante en [Hayes 96] donde las conductas se pueden quitar o poner en respuesta a un evento de la ejecución. El problema con este diseño es que la carga dinámica de software en un agente autónomo requiere un sistema de comunicación en línea entre el nivel cognitivo y el nivel físico que no cualquier plataforma soporta. Sin embargo, esta capacidad mejoraría en forma importante el desempeño de un agente situado en ambientes muy dinámicos.

El conocimiento disponible también debe afectar las conductas potenciales y las propiedades del agente. La idea aquí es reutilizar un plan previamente elaborado cuando las características iniciales del problema coincidan una a una con las especificaciones de algún problema exitosamente resuelto. Esta característica esta inspirada en el razonamiento basado en casos [Hammond 86], aunque sin las etapas de selección y refinamiento del plan. El uso adecuado del repertorio de conductas y el conocimiento resultante de procesos de planificación previos eliminan en muchos casos la necesidad de replanificar, reduciendo automáticamente el tiempo para disponer de un espectro de habilidades.

2.2 Subsumción

Los planes como intenciones ofrecen la posibilidad de usar cualquier mecanismo de disparo y control de conductas. En general, la flexibilidad de la planificación como guía de conductas permite que estas conductas puedan interactuar apropiadamente para la realización de tareas diversas durante el seguimiento del plan.

La idea no es ejecutar cientos de instrucciones para alcanzar un objetivo, sino desarrollar conductas consistentes aun con el plan más abstracto. Subsumción ha demostrado, desde su primera implantación en un agente físico, la capacidad de adaptación al dinamismo del ambiente empleando estrategias que tratan a los sensores como activadores de conductas.

A continuación se exponen las bases conceptuales de subsumción con la intención de proponer una arquitectura híbrida que aproveche las bondades de la planificación como guía de conductas disparadas por sensores o reactivas en la resolución de un problema de navegación.

2.2.1 Inspiración Biológica

Brooks discute que la inteligencia humana se desarrolló a través del tiempo, y que esta inteligencia está constituida por capas aún más primitivas [Brooks 91]. Mientras los animales unicelulares aparecieron hace 3.5 billones de años, las actividades que asociamos con la inteligencia evolucionaron más recientemente; por ejemplo, la arquitectura hace 9000 años, la escritura hace 5000, el conocimiento experto hace algunos cientos de años. Por lo anterior, Brooks afirma que la conducta de solución a problemas, el lenguaje, el conocimiento experto y la razón son todos muy simples en cuanto se manifiesta la esencia del ser y la reacción [Brooks 90]. Esta esencia del ser comprende el movimiento en y la interacción con un ambiente dinámico de modo que se asegure la supervivencia.

Brooks señala en [Brooks 91] que los sistemas biológicos de más alto nivel fueron formados por su unión con los sistemas existentes en respuesta a presiones del ambiente. Esto inspira la construcción de su arquitectura de Subsumción, discutiendo que:

- La inteligencia debe ser reactiva al dinamismo del ambiente
- Un robot móvil debe operar en escalas de tiempo similares a las de los animales y los humanos.
- La inteligencia debe generar conductas robustas con sensores inciertos, bajo ambientes impredecibles, y mundos cambiantes.

2.2.2 La Arquitectura de Brooks

La arquitectura de Subsumción ofrece una manera de combinar el control distribuido en tiempo real con conductas disparadas por sensores. Su unidad de construcción básica es la

Máquina de Estado Finito Aumentada (AFSM). Una ASFM es una máquina de estado finito que incluye además elementos temporizadores y un conjunto finito de registros de estado (Figura 2.1).

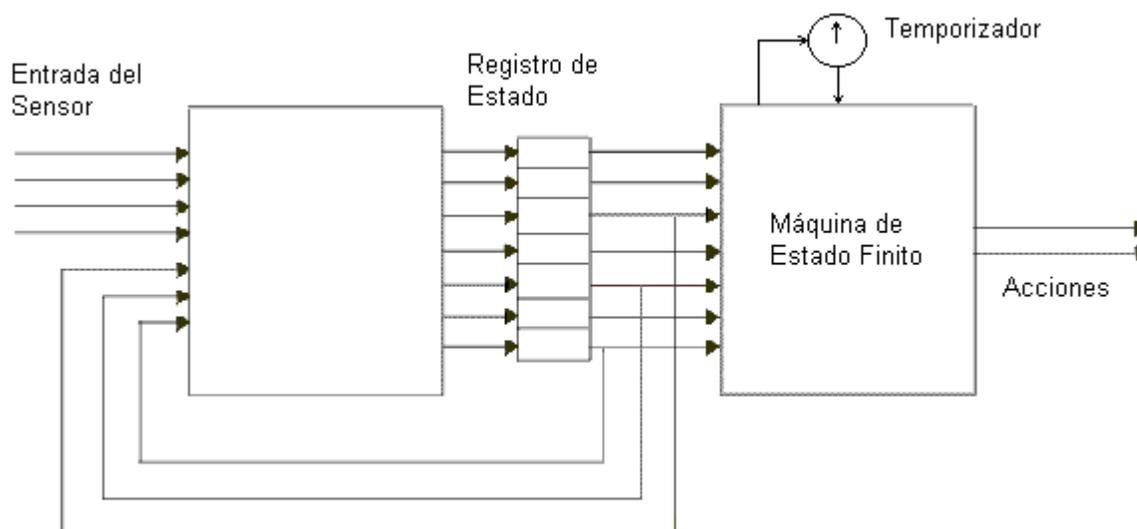


Figura 2.1. Máquina de Estado Finito Aumentada (ASFM).

Las ASFM pueden producir una salida y/o cambiar de estado al recibir un mensaje de entrada o del tiempo transcurrido, para esto los elementos temporizadores se utilizan como cuentas regresivas. Los registros de estado conforman la entrada a la máquina, y pueden reemplazar su contenido al recibir señales de otra ASFM o de algún sensor del robot.

Una ASFM puede suprimir la entrada de otra ASFM conectando su salida al registro de estado de esta otra. En este caso el mensaje en el registro es sustituido por el mensaje supresor. Una ASFM puede inhibir la salida de otra ASFM conectando su salida a la salida de esta otra. En este caso el mensaje original no se pierde, pero queda sin efecto durante un tiempo predeterminado. Estos mecanismos constituyen la esencia de la resolución de conflictos y establecen la prioridad de las conductas que componen un robot. Las ASFM no pueden compartir estados ya que los temporizadores y los mensajes actúan de manera asíncrona. Además, una ASFM no puede leer los registros de otra.

Nótese que en ninguna parte de este esquema existe la idea de que una conducta llame a otra como subrutina. En vez de esto, todas las conductas corren en paralelo, permitiendo en forma temporal que las conductas de más alto nivel supriman a las de niveles inferiores. Sin embargo, cuando las conductas de más alto nivel ya no están activas por una condición de un sensor dado, éstas cesan la supresión de las conductas de niveles inferiores para que éstas últimas tomen el control de las acciones. Así, la arquitectura es inherentemente paralela y los sensores interactúan a través de todas las capas de conductas. No hay una estructura de datos unificada o modelo geométrico del mundo.

Un grupo coordinado de ASFMs constituye una conducta y un conjunto de conductas establece un espectro de habilidades. Un espectro de habilidades constituye una especificación informal de la clase de comportamiento que deseamos que el robot exhiba en el ambiente. Observemos que cada espectro de habilidades de un robot puede hacer algo por sí mismo. La idea central es poder construir espectros de habilidades, de tal forma que el comportamiento pueda mejorarse agregando un nuevo espectro de habilidad a los ya existentes, sin que estos deban ser modificados, es decir, subsumiéndolos.

En la arquitectura de Subsumción, el diseñador debe establecer una a una las relaciones entre las conductas (salidas, entradas, supresión e inhibición) para obtener el comportamiento deseado del robot.

Las metas del robot en estas relaciones de dependencia o prioridades entre conductas están implícitas en la estructura del mismo y pueden ser de muy diversa naturaleza. Los datos de percepción son ilimitados de igual forma. El arbitraje entre conductas se realiza por medio de los mecanismos de inhibición y supresión. Estos mecanismos no contemplan la fusión de comandos, esto es, no es posible que dos conductas se activen de manera simultánea.

Algunas desventajas de esta propuesta están en relación con la escalabilidad de la arquitectura. A medida que la complejidad de un robot aumenta en términos del número de conductas y metas que se deben considerar, resulta muy difícil establecer las relaciones entre las conductas que componen el robot. Por otra parte, las metas del robot no pueden variar en el tiempo que ya se encuentran especificadas en la estructura de este.

2.2.3 Un Ejemplo

Supongamos que se desea implementar una conducta simple de evasión de obstáculos en un robot usando un detector de infrarrojos. El robot deberá avanzar hacia adelante mientras su camino este libre de obstáculos, y cambiar de dirección en cuanto se enfrente a algo o alguien. En tal caso se requieren tres conductas: AVANZAR, VIRAR y PARAR. La conducta de más bajo nivel en este ejemplo es PARAR. Se espera que la conducta VIRAR suprima las del nivel inferior cuando el robot detecte una persona, y AVANZAR suprima a PARAR cuando el camino quede libre. La conexión de estas dos conductas dan como resultado el esquema mostrado en la figura 2.2.

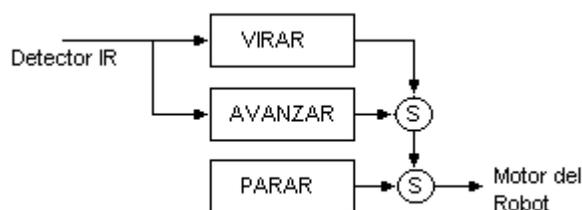


Figura 2.2. Red de Conductas Simple para la Evasión de Obstáculos

Mientras no se detecte algún objeto o persona, el robot se mantendrá avanzando. Una vez que

el sensor IR detecte algo, la conducta VIRAR empezará a enviar señales al motor del robot.

La conducta VIRAR continuará enviando señales al motor hasta que ya no se detecte nada. Es entonces cuando el robot empezará a avanzar al frente, suprimiendo las señales de VIRAR. En esta condición AVANZAR tomará el control, y se mantendrá así hasta que el robot nuevamente detecte una persona.

Este sistema funciona bien así, pero ahora se desea que el robot se detenga cuando se percate que sus baterías se están agotando, dado que con bajo nivel de voltaje la electrónica del robot no responde igual. Por lo tanto ahora se agrega una nueva conducta, llamada ESPERAR, la cual, deberá inhibir la salida de las conductas de avanzar y virar durante el tiempo en que se recarguen sus baterías. Ahora el esquema de conductas queda como sigue:

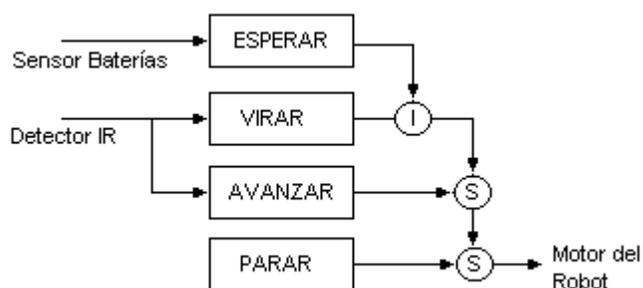


Figura 2.3. Red de Conductas de la figura anterior con un módulo para la supervisión de carga

Si algo estuviera frente al robot, VIRAR enviaría sus comandos de salida al motor; sin embargo, si se llegara a detectar bajo voltaje en las baterías, la conducta ESPERAR es la que, independientemente de la situación, inhibiría el flujo de comandos de VIRAR y AVANZAR. Así, el único mando que recibirá el motor del robot es el de PARAR, y se mantendrá así el tiempo necesario para recargar sus baterías. En cuanto el sensor correspondiente detecte suficiente carga eliminará la inhibición de conductas en los niveles inferiores, y de acuerdo a la situación, el robot avanzará o cambiará de dirección.

2.3 Arquitectura Híbrida

2.3.1 Configuración General

Las habilidades necesarias en un agente para la resolución de un problema de navegación se pueden resumir de la siguiente manera:

1. Adaptabilidad en ambientes dinámicos e impredecibles con acciones influenciadas por las metas (autonomía).
2. Atención a metas múltiples y desarrollo de conductas variadas (inteligencia).

Como se ha visto a lo largo de este capítulo, el modelo de los planes como guía de conductas implícitamente sugiere la hibridación de la escuela deliberativa con la basada en conductas.

Este complemento garantiza de cierto modo la incorporación de las habilidades generales anteriores en un agente para resolver problemas navegacionales típicos.

Así, la arquitectura híbrida propuesta tiene dos niveles de control: **nivel físico** y **nivel cognitivo**. Gracias al primer nivel, el agente será capaz de desarrollar tareas externas como la percepción y acción en el mundo real. Mientras que con el segundo, podrá realizar tareas internas de razonamiento más abstracto como el análisis de la situación actual para establecer las metas y un orden de atención, la planificación de un curso de acción adecuado, la reutilización de planes de control previos, y la resolución de conflictos.

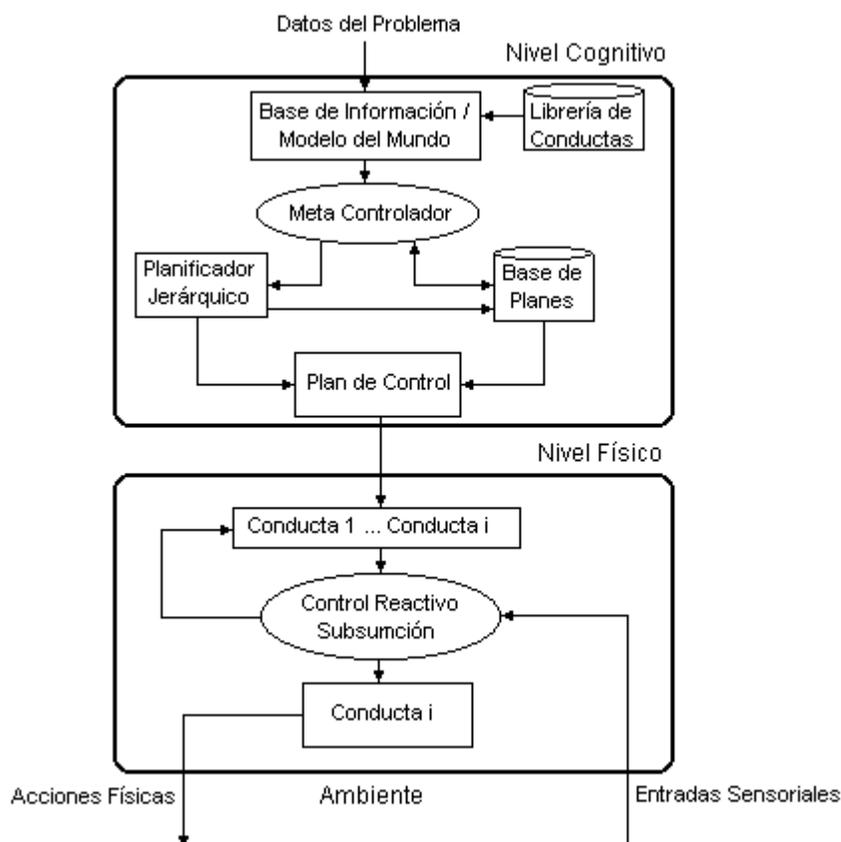


Figura 2.4. Configuración General de la Arquitectura Híbrida Propuesta

Como se muestra en la figura 2.4, sólo el nivel físico tiene contacto con el ambiente exterior. Por su parte, el nivel cognitivo influye la conducta del robot mediante su interacción con el nivel físico. El modelo de control en el nivel físico es la arquitectura de Subsumción. En este nivel se aloja un conjunto de conductas reactivas para diversos propósitos. Estas conductas son procesos ejecutables que pueden activarse mediante algún evento generado internamente, normalmente asociado con la configuración del medio ambiente y los datos del problema. Por su parte, el nivel cognitivo consta de una base de información / modelo del mundo (BI/MM),

un planificador jerárquico [Sacerdoti 74], una base de planes, y un meta controlador. La BI/MM es una base de datos central con información general del problema y acceso a las conductas disponibles para su resolución. El planificador genera planes de control descriptores de las conductas deseables en el robot. La base de planes es un almacén de los planes generados para su futura reutilización que inicialmente está vacío. El meta-controlador decide que hacer de acuerdo con la siguiente regla: "Ejecutar la secuencia de conductas que mejor coincidan con el plan de control actual".

2.3.2 Planes, Planificación y Seguimiento del Plan

La caracterización informal de los planes como descriptores de conductas define operaciones muy simples. Un **plan** es una secuencia de pasos compuestos por tripletas $\langle \text{tarea}, \text{parámetros}, \text{restricciones} \rangle$ donde *tarea* es el nombre de la tarea que el robot realizará (ej. navegar en el ambiente), *parámetros* representa lo necesario para instanciar la tarea indicada (ej. condiciones iniciales y finales de la tarea), *restricciones* es una lista ordenada de propiedades de la ejecución (ej. rapidez, seguridad, etc.). Es muy importante notar que cada uno de estos pasos no se ejecutan directamente, sino sólo describen la actividad deseada. Los argumentos *tarea* y *parámetros* corresponden a las acciones primitivas de los planes como programas igual que los operadores tipo STRIPS [Fikes 71], mientras que las restricciones son necesarias para guiar la selección de una conducta específica con la cual instanciar la parte del plan en ejecución.

La **planificación** es cualquier proceso de decisión cuya salida es una secuencia de pasos particular. Para fines de esta aplicación, los planes pueden ser lineales. Sin embargo, también pueden emplearse representaciones más sofisticadas como la planificación condicional o la planificación reactiva. En principio, cualquier paradigma de planificación de la familia de los planes como programas debe adaptarse al modelo de los planes como intenciones.

El **seguimiento del plan** es la ejecución de las conductas habilitadas que mejor coincidan con los pasos del plan. Esto es, para cada paso del plan en turno el robot debe mostrar una conducta para llevar a cabo la tarea especificada influenciada por los parámetros dados. En casos donde se habilite más de una conducta, el robot escogerá la que mejor satisfaga las restricciones. Por ejemplo, el robot puede tener dos métodos de navegación, uno rápido basado en información de sonar que lo expone a colisiones eventuales, y otro más seguro guiado por una línea guía. Si el agente prefiere la navegación por velocidad escogerá el primero, de otra forma el segundo. Sin embargo, no todas las subconductas subsumidas se habilitarán en cualquier situación. La navegación por velocidad en este caso dependerá más de un conocimiento **apriori** que del procesamiento de datos sensoriales. Si este conocimiento no está disponible, el robot será forzado a escoger el método seguro.

Como se puede observar, el enfoque de los planes como intenciones combinado con un control reactivo ofrece gran adaptabilidad. Quizá lo mejor de todo es que los planes como intenciones puedan estar lo suficientemente desacoplados de las conductas reactivas ejecutables como para que un plan sea capaz de guiar a un agente aprovechando el

conocimiento y los métodos conductuales actuales.

Capítulo 3. Marco de Trabajo.

3.1 Estudio del Caso

3.1.1 Descripción del Problema

Con la idea de plantear un problema de navegación representativo para cumplir con los objetivos de la tesis y poder afirmar que la arquitectura a desarrollar es lo suficientemente robusta, se han elegido como instancia algunas características del problema de detección de fugas de vapor en el Edificio del Turbogenerador de la Central Nucleoeléctrica Laguna Verde. Considero que este problema en particular contiene los elementos necesarios para establecer un marco de solución a problemas navegacionales similares donde un sistema robótico puede ser empleado satisfactoriamente.

Actualmente, una persona se encarga de realizar esta tarea de inspección recorriendo un circuito único delimitado por el espacio libre del edificio y haciendo estación en sitios predeterminados sobre éste. En ocasiones, el inspector sabe con exactitud la localización de las fugas (o puntos de interés) y la *criticidad* de cada zona, con lo cual es capaz de establecer un plan de inspección. Cuando se desconoce la ubicación de los puntos de interés, el inspector dispone de un plan de rutina para iniciar la búsqueda a través del circuito guía. En ambos casos, los puntos de interés son percibidos directamente en forma audiovisual, o con algún instrumento detector portátil más sensible. Por lo tanto, no es necesario planificar todo para alcanzar los puntos de interés sino sólo acotar el espacio de búsqueda donde los sentidos de la persona o algún instrumento detector puedan encontrar las fugas.

Para representar este problema se ha construido un pequeño ambiente de navegación artificial con características similares al real donde un agente físico (robot) intentará caracterizar el mecanismo de solución. Para tal efecto, el agente cuenta con un circuito guía y los sensores necesarios para la detección de objetivos, la evasión de obstáculos y la reacción apropiada al contacto con estos. Además, se han establecido distintos puntos de interés a lo largo del circuito, cada uno de los cuales representa el centro de una zona con objetivos potenciales en donde el agente puede hacer estación e iniciar una búsqueda localizada. Asimismo, cada estación puede tener un valor de prioridad dependiendo del nivel de urgencia de su zona asociada. Esto significa que hay zonas más importantes que otras y por tanto deben atenderse primero. La figura 3.1 muestra una representación abstracta del área de navegación donde se pueden apreciar dos tipos diferentes de objetivos y cuatro estaciones con tres niveles distintos de *urgencia*.

La idea general del mecanismo de solución consiste en establecer los puntos de interés y planificar un orden de atención por estación. En caso de no contar con el conocimiento para localizar las metas ni la prioridad de cada estación, el agente contará con un plan rutinario para iniciar la búsqueda de sus objetivos. En el primer caso, el agente deberá decidir que estación visitar primero en función de la prioridad o nivel de urgencia de la zona y de las metas asociadas, para después intentar localizar por reflejo los puntos de interés bajo cierto

límite de tiempo. Para tal efecto, el agente navegará sobre el circuito guía hasta llegar a la estación correspondiente. Posteriormente, se separará del circuito para iniciar su búsqueda, y una vez alcanzados sus objetivos o superado el tiempo para esta pequeña tarea, deberá buscar el circuito y navegar a la siguiente estación establecida por el plan donde repetirá el mismo proceso. Una vez agotadas las estaciones por visitar, el agente retomará el circuito para escapar de la zona de navegación y así finalizar su trabajo.

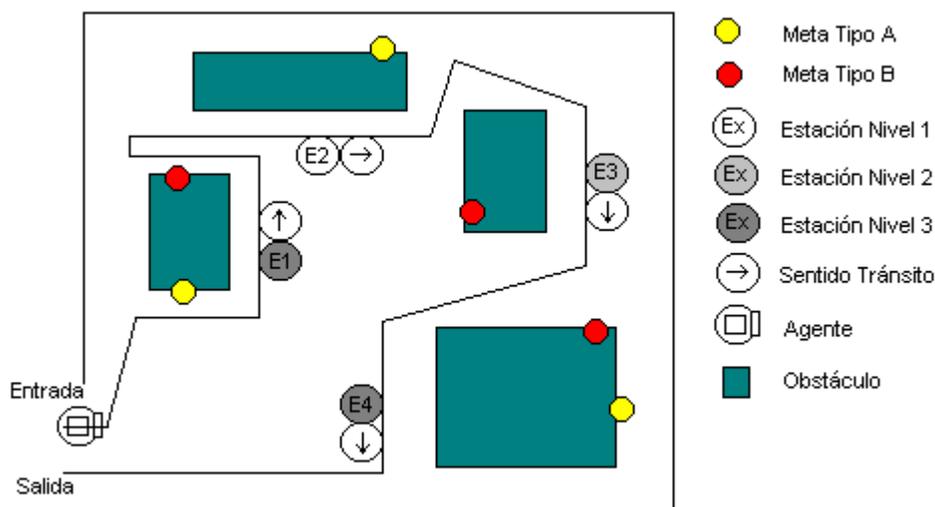


Figura 3.1. Ambiente de Navegación Tipo.

3.1.2 Descripción del Agente en Términos PAGE

Russell y Norvig sugieren en [Russell 95] el modelado de un agente y su situación en el mundo en función de sus capacidades perceptuales y de actuación, de las propiedades del medio ambiente donde éste actúa, y de las metas por alcanzar. Aprovechando este modelo (percepciones-acciones-metas-ambiente o simplemente PAGE por sus siglas en inglés) es posible especificar los elementos disponibles en el agente para la resolución del problema de navegación y las condiciones ambientales típicas de operación.

Así, el conjunto de percepciones se basa en siguiente sistema sensorial:

- Defensa Táctil (*bumpers*) para la detección de contacto.
- Sonar para la medida de distancias y detección de objetos en un rango de 50 a 300 cm.
- Sensores Infrarrojos para la detección de objetos en un rango de 0 a 50 cm.

- Fococeldas para la detección de luz.
- Sensor Seguidor de Línea basado en Infrarrojos (*linetracker*) para la navegación sobre una línea y el reconocimiento de regiones en el área de trabajo.
- Micrófono omnidireccional para la recepción de señales de audio (opcional).

Por su parte, los efectores del agente están compuestos por dos ruedas con motores independientes y un zumbador piezoeléctrico. Mediante estos dispositivos, es posible desarrollar acciones tales como:

- Avanzar linealmente hacia al frente a una velocidad dada
- Retroceder linealmente a una velocidad dada.
- Girar sobre su propio eje con sentido y velocidad dados.
- Formar un arco en sentido horario con velocidad lineal y angular dados.
- Formar un arco en sentido anti-horario con velocidad lineal y angular dados.
- Detenerse
- Emitir un tono con frecuencia y duración dados.

Es importante mencionar que algunas secuencias de acciones pueden dar lugar a conductas típicas. Estas conductas serán descritas en las secciones siguientes.

La meta general del agente es la localización del mayor número de objetivos posibles en el ambiente. Por lo tanto, el alcance de cada objetivo constituye una solución parcial al problema. Físicamente, un objetivo es una fuente de luz de distinta potencia y color con la siguiente disposición:

- Existen objetivos múltiples en el ambiente.
- La activación y cantidad de objetivos (configuración del ambiente) es parametrizable.
- Los objetivos están agrupados por estación de búsqueda.
- Los objetivos son fijos durante la deliberación del agente.
- La deliberación del agente requiere como conocimiento inicial la configuración de los objetivos.
- El mecanismo de atención a objetivos depende parcialmente de la cantidad de estos en el ambiente.

El ambiente artificial puede clasificarse como:

- Parcialmente observable. Los sensores del agente son de alcance limitado.
- Semi-dinámico. La disposición de metas y obstáculos posibles permanece fija durante la ejecución del agente. Sin embargo, es posible variar el estado de activación de una meta durante esta etapa para verificar la capacidad del agente para adaptarse a cambios ambientales.
- Discreto. El estado de las metas puede variar en espacios de tiempo finitos.
- Estocástico. No existe plena confianza en el sistema perceptual.

3.2 Preparación del Kit Robótico Rug Warrior

La primera etapa en cualquier proyecto de robótica móvil es justamente la incorporación física del agente. Entre las diversas opciones consideradas para su adquisición figuraban los robots de *NOMADIC Products* y los distintos modelos de *PIONEER*. El Pioneer AT (*all terrain*) resultó ser la opción más viable tanto técnica como económicamente.

Sin embargo, los trámites de adquisición del equipo condujeron a iniciar el proyecto con los recursos disponibles. Entre estos figuraban dos tarjetas RugWarrior (una de ellas con el kit básico) y una 6.270, ambas basadas en el microprocesador HC11. Finalmente, se decidió usar el kit completo Rug Warrior en virtud de sus características holonómicas y sensoriales, su fácil programación y gran portabilidad.

3.2.1 Kit Básico

Rug Warrior (figura 3.2) es un pequeño robot móvil autónomo desarrollado por el Instituto Tecnológico de Massachusetts (MIT), y diseñado para educadores, investigadores y aficionados interesados en el aprendizaje de la robótica, la investigación de agentes cooperativos, el procesamiento paralelo y en tiempo real, o la simple experimentación con tecnología de punta. Rug Warrior esta basado en el microprocesador MC68HC11A 1FN de Motorola, y esta provisto con la electrónica, sensores, y el hardware mecánico necesarios para una gran variedad de aplicaciones. Asimismo, incluye un poderoso sistema de programación muy fácil de aprender llamado Interactive C, que puede correr en sistemas compatibles con Macintosh , IBM, o bajo UNIX.

El kit robótico básico comprende:

Electrónica:

- Microcontrolador Motorola MC68HC11
- Pantalla alfanumérica de dos líneas de cristal líquido.

- Batería de respaldo para memoria RAM de 32KB
- Puerto Serial RS-232
- Zumbador piezoeléctrico
- Cuatro LEDs indicadores de estado programables
- Sistema de control de motores (puentes H)

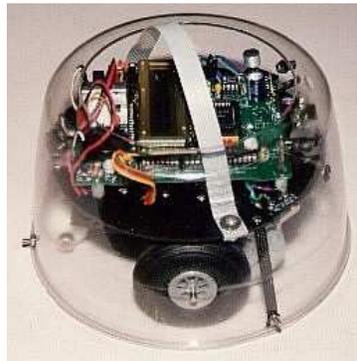


Figura 3.2. Kit Básico Rug Warrior

Mecánica

- 2 motores con engranajes
- 2 ruedas y un *caster*
- Placa chasis
- Cubierta plástica cilíndrica
- Portabaterías
- Accesorios metálicos y plásticos

Sensores

- Detectores de colisiones (*bumpers*)
- 2 Fococeldas
- Detectores de obstáculos infrarrojos con 2 LED's
- Micrófono

- 2 Codificadores (*shaft - encoders*)

Software (PC, Macintosh, Unix)

- Floppy Disk con el compilador "Interactive C" y librerías (del dominio público)
- [Programa de Autoprueba](#)
- [Programa Demo](#)

Documentación

- Guía de Ensamble
- Manual de Interactive C

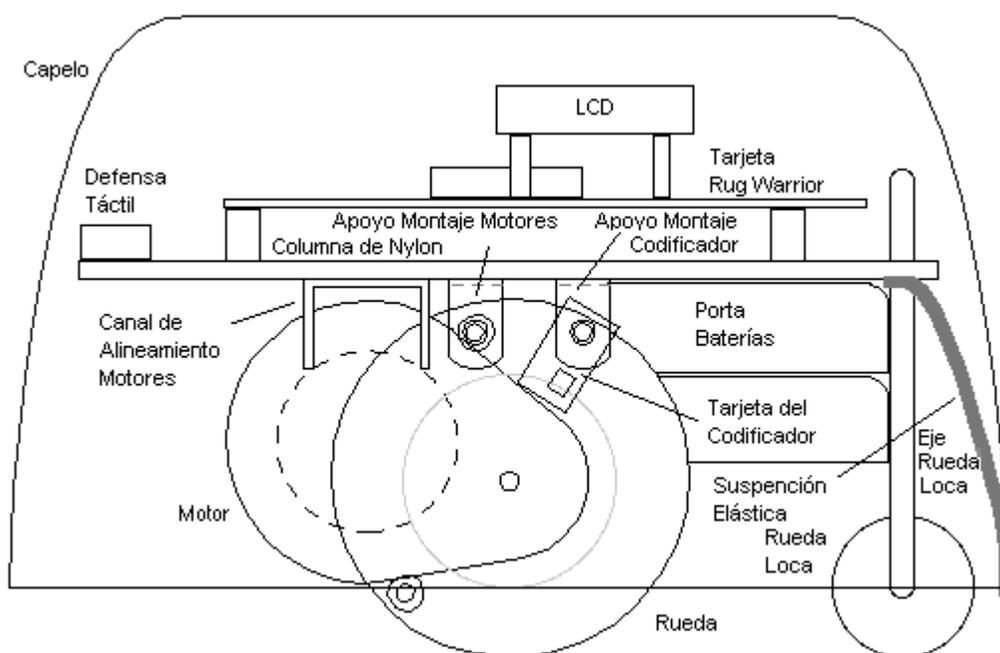


Figura 3.3. Distribución Física de Componentes en el Kit Rug Warrior

La figura 3.4 muestra la integración de los componentes esenciales que constituyen el kit Rug Warrior, así como la interconexión con su microprocesador y la computadora. El suministro de energía del robot está a cargo de una batería de 6V con la cual es posible alimentar los motores de las ruedas y la tarjeta electrónica. El microprocesador o unidad central de proceso (CPU) es el elemento de control y su función es ejecutar instrucciones en forma secuencial. La ejecución de una instrucción produce cambios en su estado, lo cual se refleja como una alteración en el valor de una celda de memoria, el contenido de un registro interno, o el voltaje

en algún puerto. El bus de datos es el canal de comunicaciones entre el CPU y la memoria, gracias al cual es posible la lectura y escritura de datos en 65,536 localidades de memoria diferentes. La unidad de memoria RAM (*random access memory*) externa tiene una capacidad de 32K, lo cual representa exactamente la mitad de la memoria disponible. Debido a que el contenido de la memoria es volátil, se requiere de una batería de respaldo para mantener los programas del robot después de apagar la tarjeta. Los LEDs indicadores son programables por lo que es posible variar su uso a conveniencia. El medio de despliegue es una pantalla de cristal líquido de dos líneas conectada al microprocesador con el cual es posible la impresión de mensajes. Para que los programas puedan escribirse, editarse, o ensamblarse a una forma que el microprocesador pueda entender, la tarjeta del robot contiene una interfaz especial capaz de acondicionar los datos enviados por el puerto RS232 de una computadora. Los sensores y actuadores con que cuenta el robot y su interconexión con el microprocesador se estudiarán en las siguientes secciones. Finalmente, el conector de expansión es un dispositivo con el cual es posible acceder los puertos disponibles del microprocesador para permitir la expansión del kit.

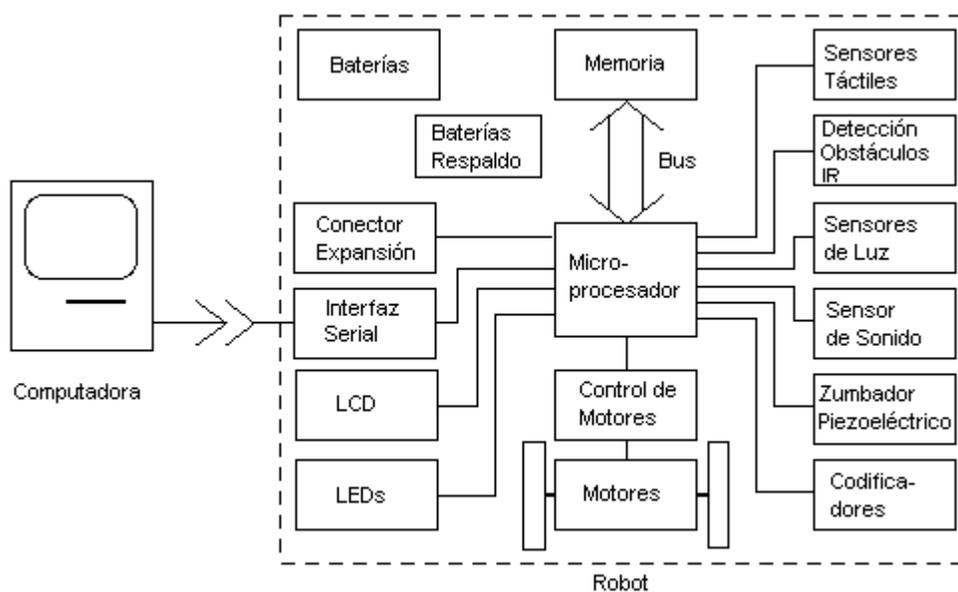


Figura 3.4. Conexión del microprocesador MC68HC11 a sensores, efectores, auxiliares y computadora.

3.2.2 Expansión del Kit

Debido a las necesidades del proyecto y a las restricciones del robot fue necesaria la expansión y adecuación del kit original. Para tal efecto se realizó la ingeniería y el desarrollo de software adicional para mejorar su desempeño eléctrico y ampliar sus capacidades perceptuales y mecánicas. Cabe aclarar que no se descubrió el hilo negro en las mejoras realizadas, sólo se aprovechó el conector de expansión de la tarjeta, se adicionaron sensores, y

en algunos casos se intercambiaron componentes cuando así fue necesario.

Una de las primeras modificaciones realizadas al robot fue la sustitución de su sistema de suministro de energía, el cual estaba compuesto originalmente por una batería comercial de 6VDC. Durante las pruebas realizadas al robot se observó que después de cierto tiempo de operación continua, la tarjeta Rug Warrior se reestablecía (auto-reset). Después de investigar las posibles causas se concluyó que la tarjeta debía operar en un rango de voltaje determinado ($5 \text{ VDC} \pm 10\%$); y una sola fuente alimentando dos cargas (tarjeta y motores) era prácticamente incapaz de mantener el voltaje requerido por la tarjeta. Por lo anterior, se sustituyó la batería original por dos baterías recargables POWER SONIC de 6 V, una para la tarjeta y otra para los motores.

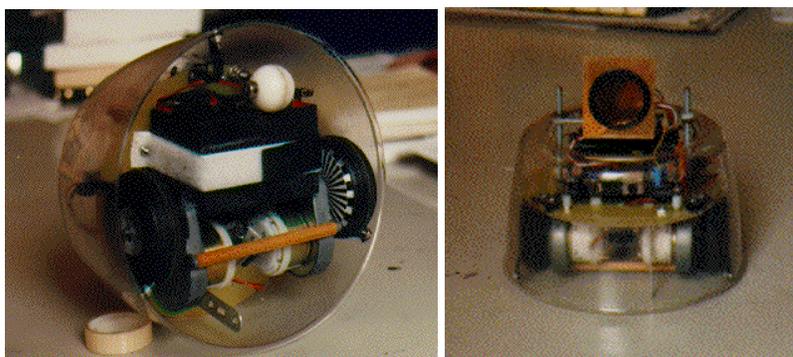


Figura 3.5. Detalle de la Fuentes y Rug Warrior con Sonar

Por su parte, después de probar con varios arreglos para la regulación de voltaje, se optó por colocar un diodo de silicio en serie con la tarjeta con el fin de limitar el voltaje máximo aplicado. Una ventaja implícita al mantener dos circuitos de alimentación independientes fue contar con un suministro selectivo para distintas cargas.

Una de las ampliaciones más importantes realizadas sobre el kit básico fue la incorporación de un sonar Polaroid 6500 cuyas pruebas de operación sólo fueron satisfactorias hasta después de resolver el problema de suministro de energía. Asimismo, se desarrolló el software necesario para su funcionamiento. Los problemas atendidos durante las pruebas se redujeron a falsas conexiones entre módulos del transductor, y a la calibración del sonar para medir distintos rangos de distancia.

Otro sensor que se agregó al robot fue un sensor pyroeléctrico, para el cual ya existían soquets especiales en la tarjeta. El trabajo se redujo a dirigir su campo receptivo mediante un cono de cartulina.

La detección de obstáculos con detectores de infrarrojos es muy susceptible al ruido, por lo tanto es muy importante mantener bien calibrado a 40 KHz el circuito oscilador que produce el haz de infrarrojos. Una modificación determinante para mejorar la detección de obstáculos fue eliminar cualquier fuente de ruido (aun transparente) entre el detector y el objeto. Por lo

tanto, se removió parte de la cubierta plástica del robot justo frente a éste, evitando así cualquier interferencia o efecto óptico indeseable que alterara la buena operación del sensor. Los resultados fueron bastante satisfactorios.

Un sensor más que se agregó al robot fue un seguidor de línea (*linetracker*) adquirido en Mondotronics. Este sensor brinda la posibilidad de navegación a lo largo de una línea pintada sobre el piso. El problema de conectar el *linetracker* a la tarjeta Rug Warrior consistió fundamentalmente en la pobre disponibilidad de puertos de entradas digitales a la tarjeta, así que fue necesario construir un circuito de interfaz capaz de ahorrar los puertos faltantes. La solución al problema consistió en diseñar una especie de convertidor D/A de 3 bits (dado que eran tres pares emisor-receptor de infrarrojos) cuya salida fuera una señal analógica de 0 - 5 VDC que indicara, sin ambigüedades, cual era la combinación de los tres pares emisor-receptor de infrarrojos del sensor activados. Dicha señal podía conectarse a un solo puerto de entradas analógicas con lo cual el problema estaba resuelto. Asimismo, se desarrolló el software necesario para la prueba y uso del sensor.



Figura 3.6. Seguidor de Línea (*linetracker*) y Servo Motor de RC

Finalmente, se agregó al robot un grado más de libertad, para lo cual se acondicionó su cuerpo con soportes y conectores para colocar un servo motor de radio control FUTABA y así accionar un brazo mecánico de aproximadamente 30 cm de longitud. El problema de conectar este componente a la tarjeta Rug Warrior era donde y como generar su señal de control, ya que éste responde a una señal digital de período constante cuyo ancho de pulso determina su posición angular. Aparentemente, sólo existe un puerto del microprocesador capaz de generar dicha señal, pero este ya estaba destinado a los codificadores de las ruedas (*shaft-encoders*). En principio se pensó en compartir el puerto, pero debido a que éste tiene la doble función de entrada-salida digital, no se hizo esperar un comportamiento descontrolado del servo al recibir la señal generada por los codificadores durante su operación simultánea con las ruedas. Una solución poco elegante pero eficiente ha sido el uso secuencial del puerto. Al igual que para los demás aditamentos, se desarrolló el software de autoprueba del servo motor.

Así, los elementos adicionales son:

Electronica

- 2 pilas recargables de 6 VDC c/u
- 1 pila alcalina de 9V

- Convertidor D/A de 3 bits

Sensores

- Sonar Polaroid Serie 6500
- Seguidor de línea infrarrojo (linetracker)
- Sensor Pyroeléctrico

Mecanica

- Servo Motor RC FUTABA
- Soportería para instrumentos y servo
- Claro frontal para detección infrarroja

Software

- [Programa para prueba de sonar](#)
- [Programa para prueba de linetracker](#)
- [Programa para prueba de servo motor](#)

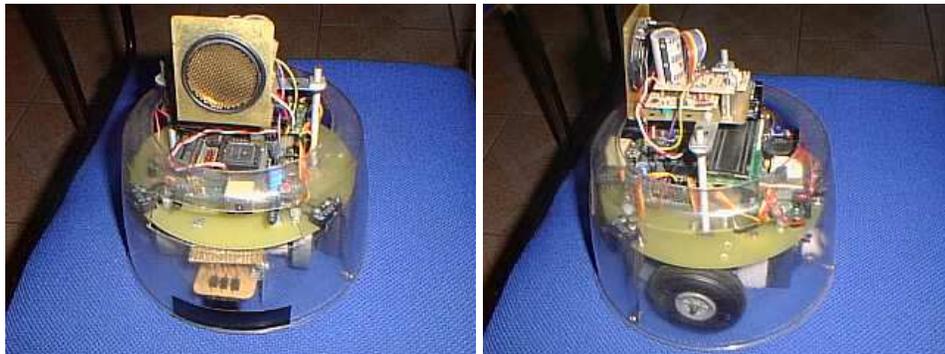


Figura 3.7. Kit Robótico Rug Warrior Extendido

3.3 Sistema Perceptual RW

Por medio de la percepción un robot obtiene información del mundo. Esta información se origina en lo que se conoce como sensores. Desde el punto de vista de la Inteligencia Artificial, un sensor es todo aquello capaz de modificar el estado de cómputo de un agente en respuesta a un cambio en el estado del mundo [Russell 95]. Sin embargo, sensar no es percibir. Eléctricamente, los sensores son sólo transductores que convierten algún fenómeno físico en señales eléctricas que un microprocesador puede leer [Jones 93].

En los humanos, la percepción se lleva a cabo mediante rutinas de sensado, interpretación, cognición y coordinación bastante complejas que muchas veces pasamos por alto. En un robot la percepción está limitada por el ambiente, sus sensores y su programación, por lo que la incorporación de esta característica no es tarea fácil.

Debido a que el mundo real es parcialmente observable, no es posible ver todos los ángulos de una situación ni tampoco recuperar todos los aspectos del estado actual del mundo a partir de los estímulos sensoriales. Aún la parte que sí se puede percibir es considerablemente ambigua. Sin embargo, un robot no necesita saber todo sobre su entorno, basta con adquirir información tal que le indique sin ambigüedad aspectos muy generales del estado del mundo. Hay que recordar que el único conocimiento directo del robot está en función de los parámetros eléctricos sentidos, por lo que la última etapa del proceso perceptual artificial es el desarrollo de software para crear distintos niveles de abstracción y así poder interpretar los datos del sensor como mejor convenga.

Bajo estos términos, un robot puede disponer de una gran diversidad de modalidades sensoriales similares a la vista, el tacto y la audición. Muchas de estas capacidades sensoriales están disponibles en el kit RugWarrior, aunque en esta sección solamente se exponen aquellas que son necesarias para el caso de estudio.

3.3.1 Foceldas

Las fotoceldas son sensores de luz hechas de sulfuro de cadmio (CdS). Estos dispositivos funcionan como simples resistencias variables similares a los potenciómetros, excepto que el cambio en la resistencia se debe a un cambio en el nivel de luz visible.

La interconexión de las fotoceldas en el kit RugWarrior con el MC68HC11 es muy simple (figura 3.8). La resistencia total en el circuito de la fotocelda izquierda es la suma de las resistencias individuales:

$$R_t = R + R_L \dots (3.1)$$

De acuerdo con la ley de Ohm, la corriente a través del circuito es:

$$I = V / R_t \dots (3.2)$$

Para que el convertidor A/D del microprocesador pueda medir un voltaje, parte de la corriente I debe circular por PE1 (Bit 1 del Puerto E). Dado que el MC68HC11 tiene entradas de alta impedancia, y la cantidad de corriente requerida es insignificante comparada con la del resto del circuito, la conexión a PE1 es despreciable. Así, el voltaje presente en PE1 es:

$$V_{PE1} = I R_L \dots (3.3)$$

Si la resistencia en la fotocelda cae conforme aumenta el nivel de luz, el voltaje en PE1 también decrece. Sustituyendo I , tenemos:

$$V_{PE1} = R_L / (R + R_L) * V_{...} \quad (3.4)$$

El convertidor A/D de 8 bits del MC68HC11 asocia el voltaje variable, V_{PE1} , con un valor de 0 a 255. Por analogía, se puede determinar el voltaje en PE0 para la fotocelda derecha.

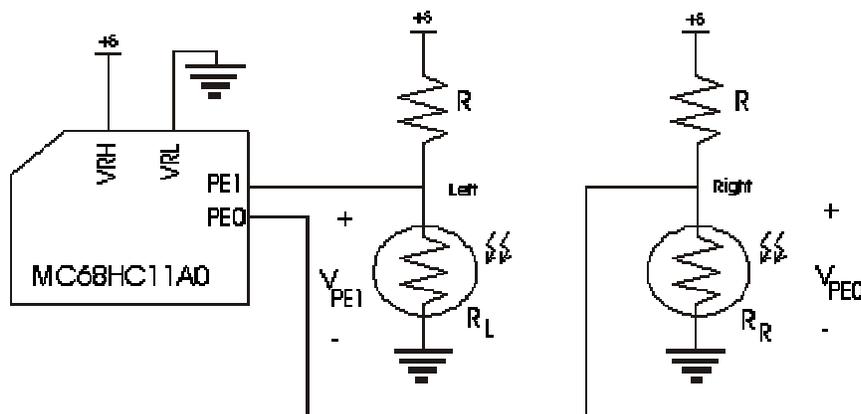


Figura 3.8. Fotoceldas en una configuración de divisor de voltaje conectadas a los bits 1 y 0 del puerto E. El puerto E aquí se usa en modo convertidor A/D.

3.3.2 Fototransistores

Los fototransistores son dispositivos muy sensibles a la luz que responden rápidamente a cambios de intensidad. El empleo de estos dispositivos en esta aplicación es el buscador o seguidor de línea. El seguidor de línea consta de tres pares emisor-receptor (LED infrarrojo-Fototransistor) colocados al frente del kit RugWarrior a 1/8" del piso. La posición del emisor respecto al receptor mantiene un ángulo tal que se asegure la detección de luz reflejada en una superficie clara. Estos dispositivos son sumamente sensibles al ruido por lo que normalmente hacen uso de una cubierta oscura para aislar cualquier incidencia de luz externa que altere la detección. La figura 3.9 muestra el arreglo físico del seguidor de línea.

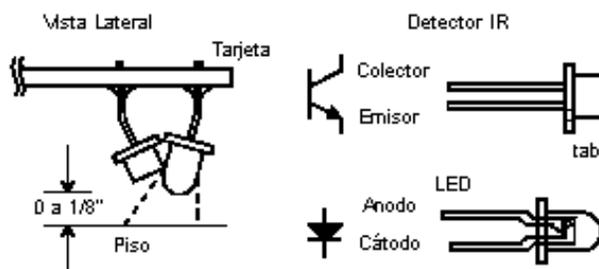


Figura 3.9. Seguidor de Línea

La filosofía de conexión del seguidor de línea con el MC68HC11 se basa en un circuito sumador con un amplificador operacional LM324 en configuración de no inversor. Con esta

modalidad es posible realizar la conversión D/A para alimentar el puerto PE6 del microcontrolador. El arreglo de resistencias conectado a la terminal positiva del amplificador permite diferenciar cualquier combinación de estados de los tres sensores del seguidor de línea. Así, la entrada al PE6 será una señal analógica de 0 – 5 Vdc que el convertidor A/D de 8 bits del puerto E asociará con un valor de 0 – 255. La tabla de entradas-salidas se obtuvo experimentalmente con los siguientes resultados:

Estado Sensores			VPE6	A/D
A	B	C	Voltios	
0	0	0	0.049	2
0	0	1	0.644	32
0	1	0	1.042	53
0	1	1	2.073	105
1	0	0	2.6	132
1	0	1	3.58	182
1	1	0	4.02	205
1	1	1	5.01	255

En la figura 3.10 se ilustra la interconexión del seguidor de línea con el MC68HC11.

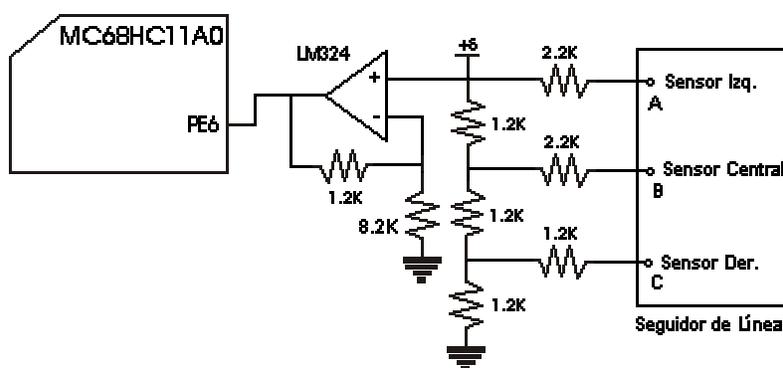


Figura 3.10. Interfaz del Seguidor de Línea con el Microcontrolador

3.3.3 Detectores de Proximidad

Los detectores de cercanía infrarroja o IRs expresan la presencia de "algo" dentro de su cono de detección. Estos detectores sólo son sensibles en un rango inferior al de la luz visible y no a

grandes longitudes de onda. La figura 3.11 muestra un par emisor - detector de infrarrojos. El emisor (arriba) es un LED hecho de arsenato de galio el cual emite energía infrarroja a 880 nm.

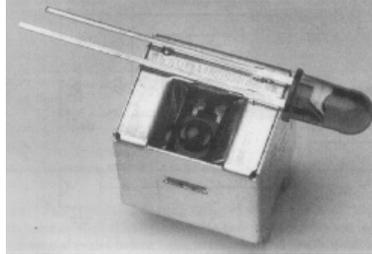


Figura 3.11. LED emisor y detector de infrarrojos.

3.3.4 Sensores Táctiles.

Los sensores táctiles se emplean para la detección de colisiones contra obstrucciones ambientales. Como lo indica su nombre, el proceso de detección involucra un contacto físico directo entre el sensor y el objeto de interés. Existe gran diversidad de tecnologías para llevar a cabo el sensado táctil:

- Cierre de contactos
- Magnético
- Piezoeléctrico
- Capacitivo
- Fotoeléctrico
- Magnetoresistivo
- Piezoresistivo

Además, hay muchas formas distintas de configurar físicamente estas estrategias. Sin embargo, desde la perspectiva de la robótica móvil, las configuraciones pueden clasificarse en tres áreas generales: 1) Antenas Táctiles, 2) Defensas Táctiles, y 3) Arreglos de Superficie Distribuida. El robot Rug Warrior emplea una configuración de defensas táctiles con cierre de contactos.

En su versión más simple, las defensas táctiles están formadas por un arreglo de microinterruptores colocados en la periferia del robot a manera de formar una defensa protectora contra colisiones. Los microinterruptores tienen una placa metálica flexible cargada con resorte con un apoyo fijo en un extremo. El otro extremo de la placa queda restringido a un movimiento de un grado de libertad de modo que cierta presión ejercida sobre la placa

realice la función de cierre de contacto. Estos dispositivos por muy simples que sean son sumamente efectivos.



Figura 3.12. Conjunto sensorial básico de un robot. De izquierda a derecha: micrófono, microinterruptor, sensor de mercurio y fotocelda.

La figura 3.13 muestra dos alternativas para conectar los microinterruptores al MC68HC11. El circuito de la figura 3.13a es muy simple: a cada interruptor le corresponde un pin diferente del puerto E. Cuando el robot entra en contacto con un objeto cierra uno o dos interruptores cambiando el estado del o los bits correspondientes de 0 a 1. Este arreglo tiene la ventaja de ser fácil de entender e implementar, pero usa más de tres líneas de entrada al microcontrolador. Otra forma de obtener la misma funcionalidad pero empleando un solo pin de entrada (figura 3.13b) es usar una red de resistencias para crear diferentes voltajes dependiendo del interruptor que haya cerrado. En este caso, el puerto E debe estar en modo A/D. El software debe leer el pin PE3, hacer una conversión, y luego prender una de ocho banderas. La bandera prendida indica el interruptor o conjunto de interruptores que han cerrado. Esto se determina verificando en cual de los ocho rangos cae el voltaje medido.

El circuito de la figura 3.13.b es un sumador de voltaje cuyo análisis considera que la mayor parte de la corriente fluye desde la fuente de 5 V a través de la resistencia de $2.2\text{ K}\Omega$ y las dos resistencias de $1.2\text{ K}\Omega$ hasta tierra, para formar un circuito con taps a 1, 1/2 y 1/4 del voltaje de alimentación al circuito (dados por los valores de las resistencias). Debido a la configuración descrita y a las tres resistencias de $47\text{ K}\Omega$, el voltaje en PE3 está dado por:

$$PE3 = 1/3 * V * (A+B+C) \dots (3.5)$$

Como el convertidor A/D discretiza el voltaje en un rango de 0 a 255, la relación para PE3 sería:

$$PE3 = 1/3 * 255 * (A+B+C) \dots (3.6)$$

Así, si sólo cierra el interruptor A, el valor en PE3 sería:

$$PE3 = 1/3 * 255 * (0+1/2+1/4) = 64 \dots (3.7)$$

Mientras que si los interruptores cerrados son B y C, el valor en PE3 será 85.

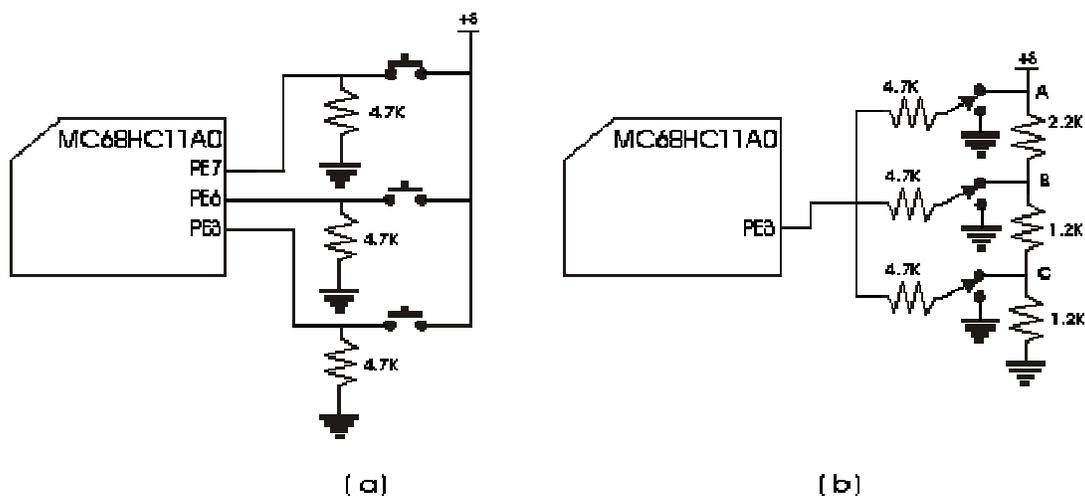


Figura 3.13. Alternativas para la detección táctil. En a) cada interruptor se conecta a un pin separado del puerto E. En b) todos los interruptores se conectan al mismo pin del puerto E.

3.3.5 Micrófono

Los micrófonos son sensores de sonido en el rango audible que permiten la interacción de un robot con su operador. Estos dispositivos (figura 3.12) se pueden conectar fácilmente al microprocesador con sólo amplificar su salida. La figura 3.14 muestra un ejemplo empleando un amplificador operacional LM386 con su salida a un pin A/D del puerto E. La lectura de datos en el puerto se hace por software con rutinas similares a las de los ejemplos anteriores.

Uno de los problemas con los micrófonos es la necesidad del muestreo continuo de la señal de audio. Por ejemplo, si un robot trata de detectar una palmada o un silbido, debe vigilar la señal del micrófono con la frecuencia suficiente para no perder el evento. La señal producida por una palmada puede durar alrededor de un milisegundo, lo cual implica que el microprocesador cheque la salida del micrófono con esta frecuencia. En casos como este, se prefiere el uso de microprocesadores exclusivos para el monitoreo del micrófono.

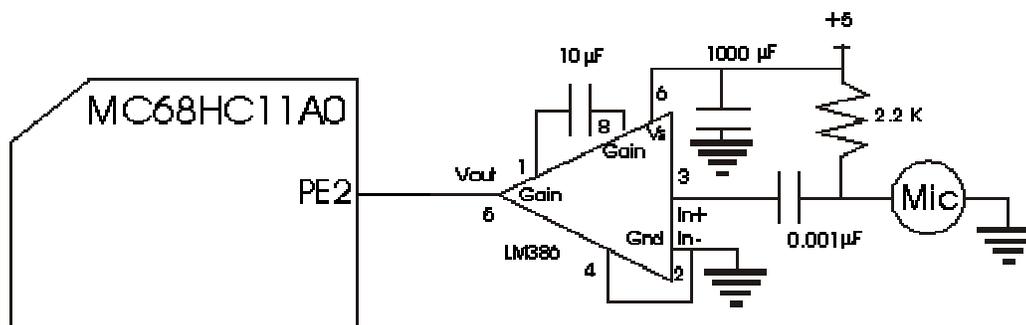


Figura 3.14. Circuito para micrófono empleando un amplificador operacional.

Otro problema es la inducción de ruido producido por los propios motores del robot, por lo

que se recomienda aislar los micrófonos a fin de evitar confundir la interpretación de la señal.

3.3.6 Sonar

El término sonar proviene de *Sound NAVigation and Ranging* (navegación y telemetría por sonido). El sonar proporciona información muy útil sobre objetos que estén próximos a un robot y frecuentemente se le utiliza en casos de emergencia para evitar rápidamente una colisión. A veces se le emplea para el trazo de mapas en áreas grandes. En este último caso, se dispone de un conjunto de sensores en torno al perímetro del robot, cada uno apuntando en una dirección distinta. Idealmente, el sensor puede medir la distancia que lo separa del obstáculo más próximo en la dirección en la que apunta. La figura 3.15 muestra un sonar polaroid con su tarjeta de interfaz al microprocesador.

El sonar se basa en la medición del tiempo que tarda un impulso sonoro producido por el sensor en llegar a un objeto y en ser reflejado por éste. El impulso por lo general es de 50 KHz, más de dos veces el límite superior de los humanos que es de 20 KHz. El sonido con esta frecuencia tiene una longitud de onda de 7 mm. Un sensor típico produce un haz cónico con una apertura de 10° o más. Los objetos muy lisos reflejan el sonido como si fueran espejos perfectos. El sonido sólo se recibe en aquellos fragmentos de superficie que se encuentran muy perpendiculares al haz. Los objetos con superficies planas y aristas afiladas reflejan muy poco sonido en la mayoría de las direcciones.

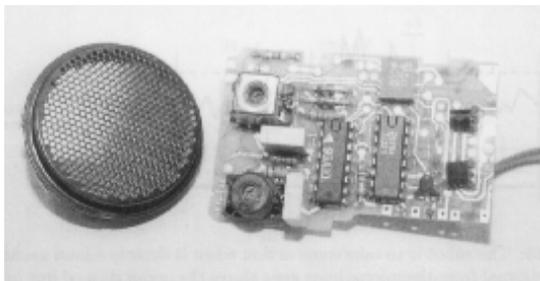


Figura 3.15. Sonar Polaroid y su tarjeta de interfaz.

Debido a la fácil inclusión de ruido en las mediciones con sonar, se sugiere construir un modelo probabilista del sensor y luego utilizar una actualización bayesiana para integrar la información obtenida durante la navegación de un robot. Esto podría facilitar la construcción de mapas de exactitud razonable, y la eliminación de imágenes fantasmas.

La tarjeta de un sonar maneja tres señales de control digital que interactúan con el microprocesador (figura 3.16). Una de ellas, VSW, es la señal que recibe del microprocesador (desde PA4). Existe cierto retraso entre el envío del impulso de ultrasonido y el momento en que en realidad se emite, por lo cual hay una señal de retorno (XLG) al PA1 del microprocesador, que indica el inicio real del impulso. En este momento, el microprocesador inicia la cuenta de tiempo. El circuito analógico del sonar crea un pulso de 300 V para que el

transductor envíe la señal de ultrasonido y lo pone en modo receptor. El tiempo de retorno es registrado y enviado al PA2 del microprocesador desde el pin FLG del sonar. Con esta información, el microprocesador puede calcular la duración en tiempo del viaje de ida y vuelta del impulso, y considerando que la velocidad del sonido es de 300 m/s puede saber la distancia recorrida. La mitad de la distancia recorrida es la distancia al objeto.

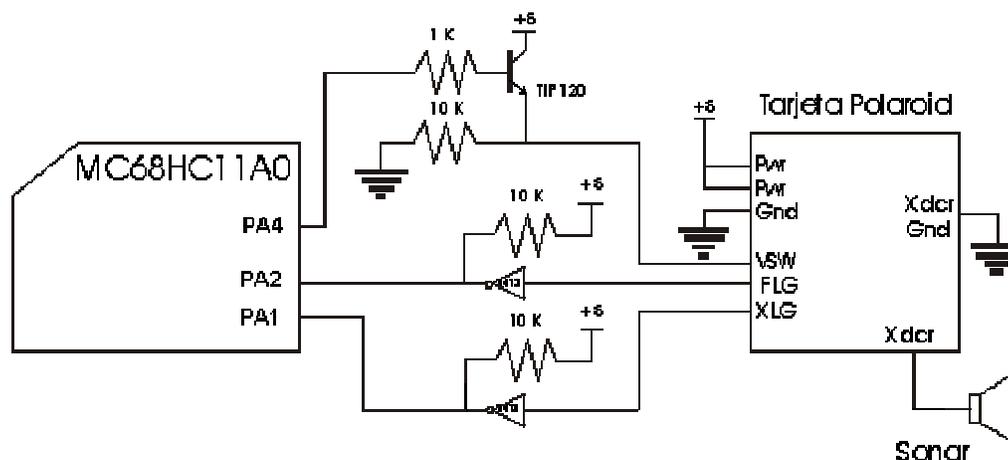


Figura 3.16. Interfaz de un sonar polaroid con el MC68HC11

3.3.7 Propiocepción

Al igual que los seres humanos, los robots disponen de un sentido propioceptivo, que les permite saber cual es el estado interno que guardan respecto al mundo, esto es, localización de sus articulaciones, posición, orientación de sí mismo, estado de carga, etcétera.

Para medir cambios de posición, Rug Warrior se vale de la odometría, utilizando sensores que miden el giro de sus ruedas (o en el caso de los motores de pasos, que giran cierto ángulo por paso, miden la cantidad de ellos). Estos sensores se conocen como codificadores incrementales, y la señal que entregan es un tren de pulsos.

Típicamente un codificador incremental (*shaft - encoder*) está montado sobre la parte externa del eje de un motor, y la señal que entrega corresponde a una orientación en particular de la misma. Cada vez que la flecha gira cierto ángulo, el estado de su salida cambia de un alto a un bajo, o viceversa. Por lo tanto, la proporción de generación de pulsos corresponde a la proporción de giro de la flecha. La generación de pulsos se realiza mediante un fotorelector, el cual dirige un haz de luz infrarroja sobre un disco con segmentos claros y oscuros. Los 64 segmentos claros y oscuros están dispuestos en forma radial sobre el disco con lo cual es posible conmutar entre los estados de reflejo y no reflejo para producir un tren de pulsos en la salida de un fototransistor.

Para leer los datos del codificador incremental se aprovecha el hardware para el conteo de tiempo conectado al puerto A del microprocesador MC68HC11. Los 8 pines de este puerto tienen varios registros para la captura de entradas (*input capture registers*) y la comparación

de salidas (*output compares registers*), los cuales son capaces de marcar el tiempo en el que sucede un evento sobre esos pines o iniciar eventos con tiempos programados. La tarjeta del Rug Warrior utiliza los pines PA7 y PA0 para aceptar la entrada de los codificadores izquierdo y derecho respectivamente. La función de acumulación de pulsos asociada al PA7 efectúa la cuenta producida por el codificador izquierdo vía software, mientras que el codificador derecho (conectado al PA0) usa su función de captura de eventos para contar los pulsos.

El problema con estos sensores es que el desplazamiento producido por la movilidad aumenta el error de posición de la rueda conforme avanza el robot, el cual puede llegar a varias unidades porcentuales en relación con la distancia recorrida.

Mediante el sensado del nivel de voltaje en sus baterías, un robot puede determinar cuando es necesario visitar la estación de recarga o escapar de un área de trabajo antes de quedarse sin energía. El diseño de los indicadores de nivel de batería se basa en circuitos comparadores de voltaje. Este tipo de sensado no ha sido implementado en el RugWarrior; sin embargo se propone como trabajo futuro.

3.4 Ambiente Artificial AMBART-RW

El diseño físico del ambiente artificial consta de 16 secciones rectangulares de 30x100 cm cuyas paredes pueden ser opacas o de malla metálica. Cada sección esta provista con bisagras de perno removible en ambos lados para obtener diferentes configuraciones (rectangular o poligonal). El conjunto de secciones rectangulares unidas por las bisagras conforman los limites de un área máxima de navegación de 16 m² (400 veces el área que ocupa el agente).



Figura 3.17. Ambiente Artificial AMBART-RW empleando 8 secciones rectangulares

La disposición física de los obstáculos puede variar en cantidad como en posición, siempre y cuando el área ocupada por estos no exceda el 25% del área de navegación. Los obstáculos están compuestos por bloques de cartón o madera, preferentemente de color claro, de modo que permanezcan inmóviles y que puedan soportar otros objetos como lámparas y bocinas.

El circuito guía es una línea continua marcada sobre un piso blanco mediante cinta adhesiva de color negro. Los arcos para cambios de dirección del agente deben tener radios de curvatura de 10 cm como mínimo para evitar que el robot pierda el circuito por efectos inerciales. La razón fundamental de usar un circuito guía es dirigir la navegación del robot por zonas seguras, evitar su atascamiento, y proporcionar un medio de ingreso al y escape del área de navegación.

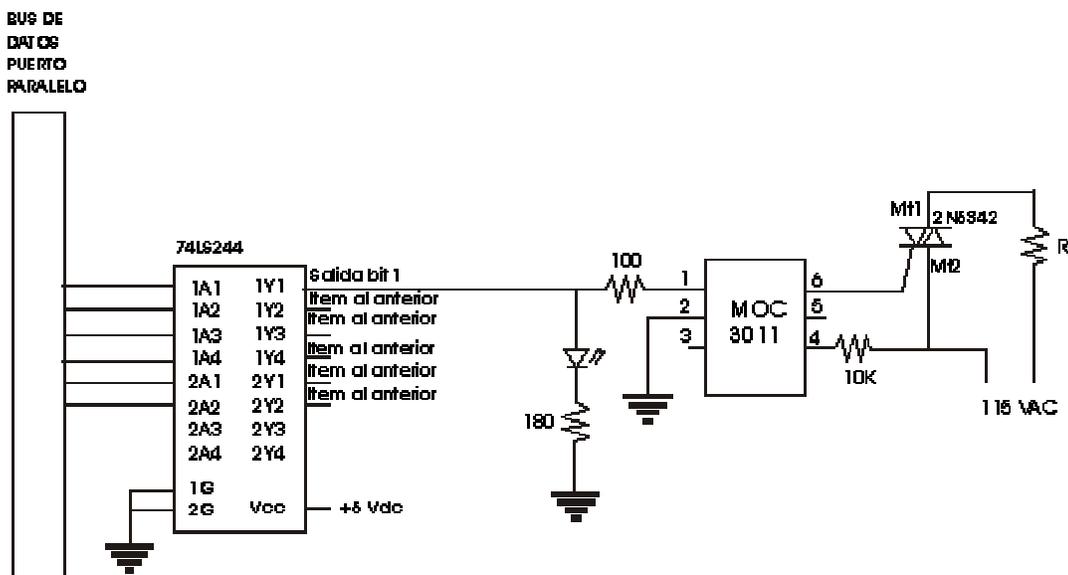


Figura 3.18. Interfaz para el puerto paralelo mostrando acoplamiento óptico

El control de activación de metas esta compuesto por una interfaz gráfica con el usuario (GUI), una interfaz de hardware, cables y conectores eléctricos mediante los cuales el usuario puede energizar o desenergizar alguna fuente de luz para lograr diferentes configuraciones del ambiente.

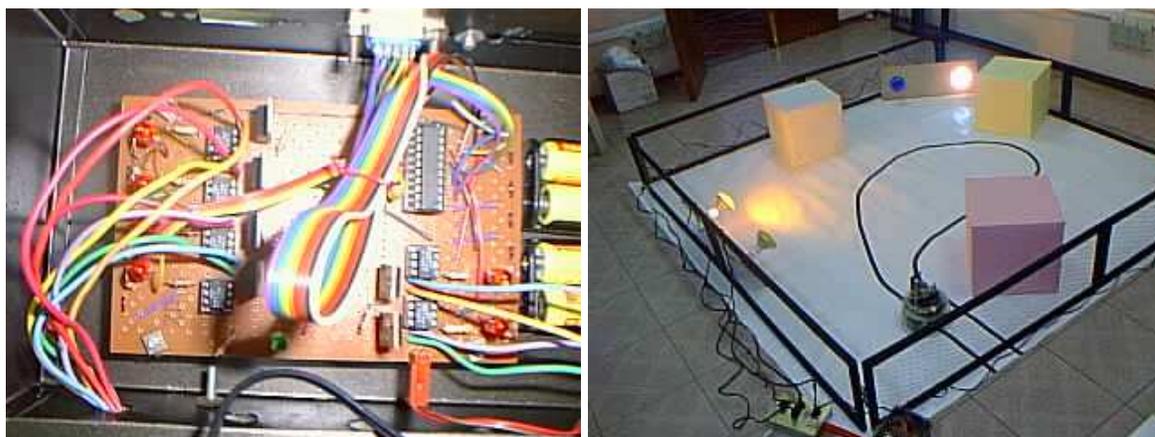


Figura 3.19. Interfaz de Hardware y Ambiente Artificial Iluminado

La interfaz de hardware recibe señales a través del puerto paralelo de la computadora para realizar un acoplamiento óptico entre un circuito de 6 VDC y otro de 115 VAC. El primer circuito hace las funciones de cierre de contactos y el segundo de energización de 6 lámparas de 75 W para conformar los objetivos del robot. La tarjeta consta de optoacopladores MOC3011, triacs 2N6342, leds, un buffer 74LS244, resistencias de 100, 180 y 10 K Ω , una fuente de CD, conectores DB-25 y DB-9 y es capaz de manipular una máximo de 6 bits (número de objetivos).

Capítulo 4. Implantación Técnica.

4.1 Mecanismo de Solución

Los planes de muy bajo nivel implicarían varios cientos de pasos para resolver el problema de navegación expuesto en la sección 3.1.1. El espacio de planes de esta magnitud sería tan grande que aun las técnicas de planificación más sofisticadas probablemente no llegarían a una solución en una cantidad razonable de tiempo. Por su parte, los planes con niveles de abstracción bastante elevados producirían soluciones muy lejanas al tipo de instrucciones que se pueden alimentar directamente a los efectores de un agente.

Este dilema ha sido resuelto utilizando un planificador jerárquico para la selección de operadores de muy alto nivel abstracto, un método para la descomposición de estos operadores, y la arquitectura de Brooks como mecanismo de arbitraje entre las conductas que un agente puede mostrar. En términos de un plan como guía de conductas el operador de más alto nivel es un operador conductual que puede descomponerse en un grupo de habilidades (pasos) para formar el plan que implementa el operador. En caso de existir tareas de búsqueda, los parámetros producto de la planificación se pueden almacenar en un archivo de planes y recuperarse cuando sea necesario. El plan estará completo cuando todos los pasos sean conductas primitivas.

Por lo general, la diferencia entre lo que es primitivo y lo que no es depende del agente encargado de ejecutar el plan. Desde el punto de vista del planificador, la conducta IR_A_ESTACION puede considerarse como primitiva, ya que todo lo que tiene que hacer es indicar al agente que realice esta tarea de navegación. En cambio, el agente considerará que IR_A_ESTACION no es primitiva, mientras que SEGUIR_LINEA y EVADIR_OBSTACULOS si lo son.

Para que la idea de la planificación como guía de conductas funcione se necesitan dos condiciones:

- Descomponer el problema en niveles de abstracción empleando operadores que no sean primitivos en el nivel cognitivo.
- Establecer la relación adecuada de conductas en el nivel físico para aceptar el reemplazo de operadores que no sean primitivos por su respectiva descomposición.

4.2 Abstracción Jerárquica del Problema

Para llevar a cabo la descomposición del problema se utiliza un planificador jerárquico con dos niveles de abstracción llamado PLANABS. En su primer nivel PLANABS formula un plan global con requisitos de tiempo poco exigentes estableciendo la jerarquía de atención a las estaciones que el agente puede efectuar en función de su prioridad. En el segundo nivel reduce el alcance del plan y el tiempo de resolución, por lo tanto hace uso de conocimiento

más detallado, como son los objetivos asociados a cada estación. Este proceso de planificación da como resultado el operador conductual que permitirá al robot alcanzar la mayor parte de sus objetivos, los parámetros de búsqueda (número de objetivos totales, secuencia de estaciones por visitar, número de objetivos por estación, y sentidos de navegación entre estaciones), y el conjunto de restricciones (velocidad del robot y tiempos máximos de búsqueda). Toda esta información se almacena en una base de planes como ejemplos para su uso futuro.

4.2.1 Operadores Conductuales

Antes de explicar el planificador jerárquico, se necesitan conocer los operadores del nivel abstracto más alto disponibles en esta aplicación. Estos operadores describen en términos muy generales tareas de navegación complejas que el agente puede ejecutar. Cuando la naturaleza del problema así lo requiera, los operadores conductuales pueden venir acompañados de un conjunto de parámetros de búsqueda y otro de restricciones. Estos operadores son: INTEGRAL, RUTINA y NULA.

INTEGRAL es un operador que describe tareas de navegación que implican la búsqueda localizada de objetivos. Por lo tanto, requiere parámetros de búsqueda y restricciones muy específicas. Su nombre se debe a que integra una gran variedad de conductas de niveles muy primitivos para alcanzar sus objetivos y, para respetar el orden de visita a cada estación. Normalmente se usa cuando existen diversos grados de urgencia en cada estación posible del agente donde se supone hay objetivos activos.

RUTINA describe tareas de inspección muy generales donde no es necesario alcanzar el objetivo sino sólo apuntarlo desde el circuito de navegación. No necesita parámetros especiales ya que el orden de visita por estación siempre es el mismo. Este operador debe su nombre a que el comportamiento general que manifiesta se relaciona con la supervisión rutinaria por zonas que el personal de las plantas industriales normalmente realiza para la detección anticipada de problemas. RUTINA se usa cuando las estaciones del robot tienen la misma prioridad y, existe al menos un objetivo activo en toda el área de trabajo.

NULA no manifiesta acciones de navegación. Está íntimamente relacionado con problemas sin solución o, cuando no tiene ningún sentido la participación del robot. Este operador es seleccionado en casos donde no hay metas por alcanzar. NULA se manifiesta en el mundo con un mensaje sonoro indicando el fin de la tarea.

4.2.2 Planificador Jerárquico PLANABS

En esta sección se describirán las funciones principales del planificador PLANABS para seleccionar el operador conductual, los parámetros de búsqueda y las restricciones que mejor resuelvan el problema en turno.

La primera función de PLANABS es coleccionar todo el conocimiento disponible sobre el caso actual, tal como la prioridad de cada estación y el estado de los objetivos (BI/MM). Existen

dos casos generales donde no es necesario establecer parámetros de búsqueda y por lo tanto el proceso se reduce a la selección del operador conductual. El primer caso se presenta cuando no hay metas por alcanzar. Esto significa que no hay nada que hacer y por lo tanto no es necesaria la participación del agente. Así, el operador seleccionado es NULA. El segundo caso general considera la existencia de metas en el ambiente y, que no hay estaciones más importantes que otras (todas las estaciones tienen la misma prioridad). En tal situación PLANABS selecciona el operador RUTINA para indicar un recorrido de supervisión sobre todo el circuito de navegación.

Fuera de estos dos casos generales el operador seleccionado es INTEGRAL, y el resto de la planificación se concreta al establecimiento de los parámetros de búsqueda. Para lo cual, hay dos alternativas de solución:

- Extracción de los parámetros de búsqueda de una base de planes (BP).
- Generación de estos parámetros mediante descomposición del problema en niveles de abstracción.

Para llevar a cabo la extracción de parámetros de la BP se toma el conocimiento disponible para formar un vector de entrada. Este vector es comparado con cada ejemplo de la base de planes y, de existir, se extrae el vector de salida (parámetros) correspondiente al ejemplo. Los ejemplos de la BP tienen la forma *entrada - salida*. El lado izquierdo de esta pareja corresponde a la lista de tuplas <prioridad, edo_obj1, edo_obj2>, mientras que el lado derecho contiene tuplas <no_estación, edo_obj1, edo_obj2, sentido_viaje>. En caso de que la búsqueda del conjunto de parámetros fracase, el planificador procede a generarlos.

La generación de parámetros por descomposición del problema tiene dos niveles de abstracción. El primer nivel es el más global y sólo considera las prioridades de las estaciones (no los objetivos asociados a cada una de éstas). En este nivel se conforma una lista de tuplas <prioridad, edo_obj1, edo_obj2> ordenada por prioridad y por acceso geográfico; esto significa que de encontrarse con dos estaciones con la misma prioridad de atención, la función de ordenamiento dará preferencia a la estación más próxima a la ubicación del robot. El segundo nivel de abstracción opera en forma más localizada, y por tanto considera la existencia de objetivos cercanos a cada estación. Gracias a este nivel es posible descartar estaciones sin metas activas, lo cual optimizará el orden y cantidad de estaciones por visitar. El conjunto de parámetros de búsqueda en este nivel está representado por una lista de tuplas <prioridad, no_estación, edo_obj1, edo_obj2, sentido_viaje>. Finalmente, el vector de entrada obtenido en la etapa anterior se concatena con los parámetros de búsqueda y se agrega como ejemplo a la BP. La figura 4.1 muestra el funcionamiento del planificador mediante un diagrama de flujo.



Figura 4.1. Diagrama de Flujo del Planificador PLANABS.

4.3 Descomposición de Operadores

Para incorporar la descomposición del operador primero es necesario dividir el conjunto de operadores en primitivos y no primitivos. Segundo, se emplea un método de descomposición jerárquica. Y finalmente, se usa la arquitectura de subsumción para dar seguimiento al plan intencional.

En nuestro dominio y desde el punto de vista del planificador, los operadores primitivos son las redes de conductas IR_A_ESTACION, ALCANZAR_OBJETIVOS, BUSCAR_CIRCUITO y ESCAPAR. ENFRENTAR_OBJETIVOS es un caso de conducta primitiva aun desde el punto de vista del agente. Los operadores no primitivos son: INTEGRAL, RUTINA y NULA.

Puede considerarse que un método de descomposición es semejante a una definición macro correspondiente a un operador. La idea es garantizar la posibilidad de reemplazar un operador no primitivo por su descomposición, al tiempo que todo lo demás encaja perfectamente. La figuras 4.2 muestran la descomposición de los operadores INTEGRAL y RUTINA en un plan

que manifiesta la intención del agente por resolver la tarea. NULA no implanta ninguna descomposición.

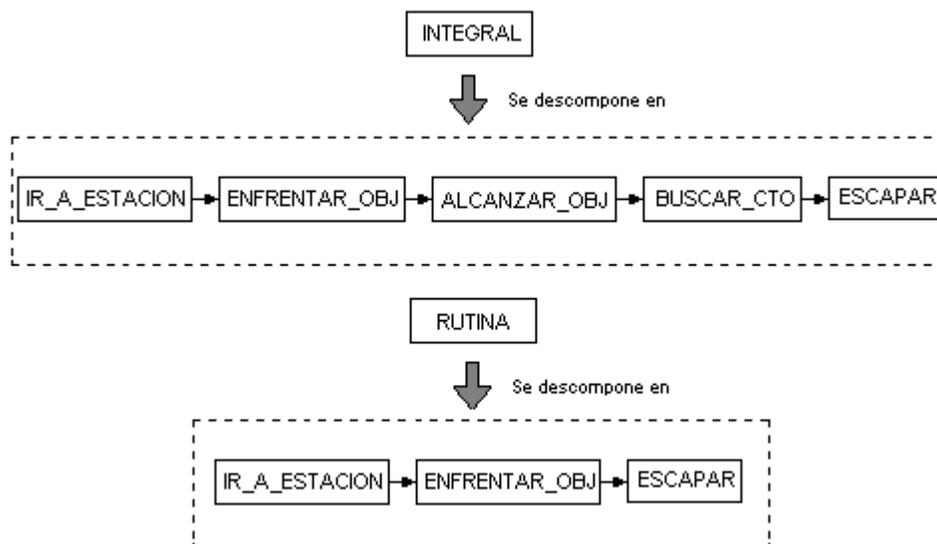


Figura 4.2. Descomposición de los Operadores INTEGRAL y RUTINA.

4.3.1 Redes de Conductas

Brooks emplea las conductas orientadas a tareas (espectro de habilidades) como nivel primario en la descomposición de tareas. Hay que recordar que el concepto de subsumción opera a través de un conjunto de conductas estructuradas en capas con distintos niveles de competencia. Esta sección describe brevemente la estructura de las habilidades disponibles en esta aplicación, incluyendo las conductas primitivas y mecanismos de supresión e inhibición que las conforman.

IR_A_ESTACION. La finalidad de este operador es permitir la translación del robot desde una estación origen a una estación destino. Su conducta base es SEGUIR_LINEA y su sistema sensorial se compone de un seguidor de línea y, un codificador incremental para medir desplazamiento. La distancia entre estaciones es un valor constante que depende del tamaño del circuito de navegación. Por su parte, el valor absoluto de la diferencia entre el origen y el destino se usa como factor de avance. Así, el desplazamiento viene dado por:

$$S=D*FA \dots (4.1)$$

donde:

- S desplazamiento en cuentas digitales,
- D distancia entre estaciones en cuentas digitales,
- FA Factor de avance $\text{abs}(\text{origen} - \text{destino})$

Cabe aclarar que las estaciones no son marcas topológicas sensibles sino zonas que dependen del desplazamiento del robot. Además, no se tiene contemplada la evasión de obstáculos ni la reacción al contacto, por lo que es muy importante que el circuito de navegación permanezca libre de objetos.

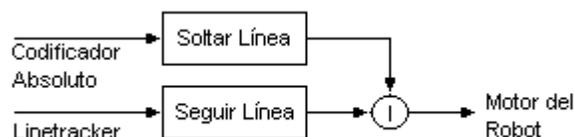


Figura 4.3. Red de Conductas para el Operador IR_A_ESTACION.

De acuerdo con la red de conductas de la figura 4.3, el motor del robot ejecutará las instrucciones de la conducta `SEGUIR_LINEA` mientras el codificador no determine que el robot ya está en la estación destino. Una vez alcanzado su destino, se activará la conducta `SOLTAR_LINEA` enviando a los efectores comandos para girar 90° y avanzar linealmente unos cuantos cm de modo que el robot deje la línea y establezca una zona por defecto.

ALCANZAR_OBJETIVOS. El objetivo de este operador es permitir que el robot alcance un número determinado de fuentes luminosas dentro de un tiempo máximo de búsqueda, evadiendo obstáculos, y registrando cambios de zona en el área de navegación. Las conductas primitivas que lo integran son: `REACCION_AL_CONTACTO`, `EVADIR_OBSTACULOS_2`, `BUSCAR_LUZ` y `SEGUIR_LUZ`. Su conjunto sensorial está compuesto por: defensas táctiles, detectores de infrarrojos, fotoceldas, y el seguidor de línea o *linetracker* como detector de cruces por la línea guía.

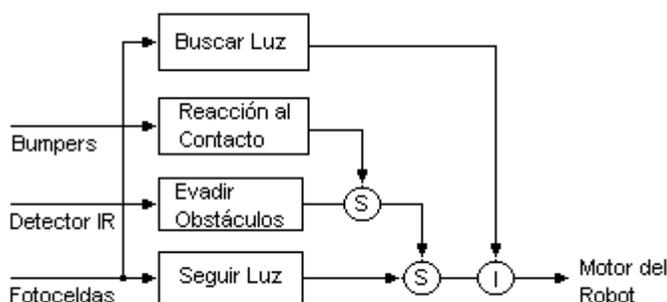


Figura 4.4. Red de Conductas para el Operador ALCANZAR_OBJETIVOS

Durante su ejecución, la conducta `BUSCAR_LUZ` hará que el robot enfrente su objetivo. Después, el robot navegará bajo el mando de la conducta `SEGUIR_LUZ` con el siguiente esquema de activación e inhibición de conductas: Si durante la navegación guiada por luz el robot detecta objetos en su camino, `SEGUIR_LUZ` quedará suprimida por la conducta `EVADIR_OBSTACULOS_2` hasta no detectar objetos al frente. En caso de presentarse una colisión la conducta `REACCION_AL_CONTACTO` suprimirá las dos conductas anteriores

hasta que el robot haya escapado del problema. Cuando el nivel promedio de luz captado por las fotoceldas rebase cierto umbral, es decir que la fuente de luz esté lo suficientemente cerca, el robot se detendrá, girará media vuelta tratando de enfrenar el camino de regreso al circuito, y volverá a detenerse.

BUSCAR_CIRCUITO. Este operador permite al robot buscar el circuito de navegación y, una vez localizado, orientarse correctamente hacia la siguiente estación de visita. Al igual que en otras rutinas de búsqueda, el robot debe evadir obstáculos en su camino y evitar atascamientos. Además, debe contar con alguna heurística para localizar el circuito. Las conductas básicas que lo conforman son: REACCION_AL_CONTACTO, EVADIR_OBSTACULOS_2 y BUSCAR_LINEA. Su sistema sensorial se basa en: defensas táctiles, sensor de línea y detectores de IR.

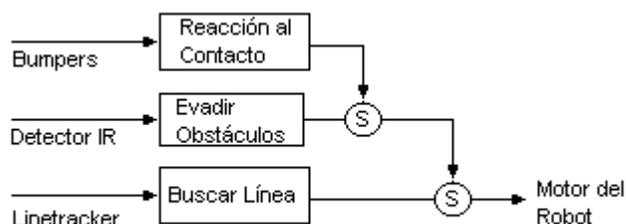


Figura 4.5. Red de Conductas para el Operador BUSCAR_CIRCUITO.

De acuerdo con el esquema de la figura 4.5, la conducta BUSCAR_LINEA mantendrá enviando sus comandos a los efectores del robot a menos que se presente algún obstáculo o atascamiento durante la búsqueda. De lo contrario, se activarán las conductas EVADIR_OBSTACULOS_2 o REACCION_AL_CONTACTO manteniendo suprimida la conducta BUSCAR_LINEA. En esta aplicación, no se incluye una función heurística para localizar el circuito por lo que el comportamiento es simplemente marchar hacia adelante.

ESCAPAR es un operador que esencialmente funciona igual que IR_A_ESTACION. La única diferencia es que su condición de parada considera la existencia virtual de una estación adicional y, por lo tanto, la ecuación 4.1 aplica satisfactoriamente. Lo anterior dará el efecto de abandonar el circuito de navegación una vez terminadas las tareas de supervisión o búsqueda.

4.3.2 Conductas Primitivas

Las siguientes conductas constituyen los elementos básicos de la arquitectura y su función es enviar instrucciones a los efectores del robot. Estos elementos permiten la ejecución de rutinas simples para navegar sobre una línea guía, detectar y alcanzar objetivos, evadir objetos y escapar de colisiones.

EVADIR_OBSTACULOS_1. Esta conducta basa sus acciones en el uso de un sonar para la medida de distancias y la detección de objetos en un rango de 40 a 350 cm. Cuando el sonar

detecta objetos a distancias seguras (>100 cm), el robot puede moverse al frente; por su parte si el objeto se localiza entre 50 y 100 cm, el robot formará arco hacia su derecha con diferentes radios. Una medida preventiva del robot para evitar una colisión es retroceder en presencia de obstáculos a distancias menores de 50 cm.

EVADIR_OBSTACULOS_2. Esta rutina también tiene por finalidad la evasión de obstáculos, pero sus acciones se basan en la detección infrarroja cercana. Para esto, el robot cuenta con dos LEDs emisores de infrarrojos en su parte frontal, con los cuales no sólo es capaz de detectar objetos en un rango máximo de 50 cm, sino de saber la posición relativa de éstos respecto a sí mismo. Esto permite que la variedad de acciones posibles sea más rica. Por ejemplo, si no hay objetos al frente la acción es avanzar; si se detecta un objeto a la izquierda o a la derecha el robot forma un arco en sentido angular opuesto; cuando los dos LEDs IR están bloqueados (objeto al frente) la acción inmediata es girar sobre su eje.

REACCION_AL_CONTACTO. La percepción de objetos en el ambiente no siempre es confiable y por lo tanto no hay que descartar la probabilidad de contacto con ellos. Mediante esta conducta el robot puede escapar de una colisión. Para tal efecto, Rug Warrior utiliza dos *bumpers* frontales y uno posterior separados simétricamente por un ángulo de 60° (figura 4.6). Además, está dotado de una cubierta plástica con la cual se puede efectuar el cierre de interruptores en forma individual o por pares. Así, el robot puede tener una buena idea de que sección de su cuerpo ha chocado. Las combinaciones posibles de contacto son: golpe al frente (cierran sus dos interruptores frontales), golpe frontal izquierdo (interruptor izquierdo), golpe frontal derecho (interruptor derecho), golpe lateral derecho (interruptores derecho y posterior), golpe lateral izquierdo (interruptores izquierdo y posterior), y golpe posterior (interruptor posterior). En esta rutina, el número de choques consecutivos se utiliza como factor de giro del robot para salir más rápido del atascamiento o colisión. La secuencia de acciones para escapar del problema inicia con un pequeño movimiento en sentido contrario a la dirección del golpe, y después un movimiento de rotación contrario a la posición del interruptor cerrado (o bumper activo).

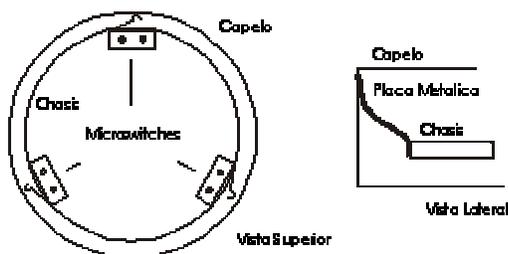


Figura 4.6. Arreglo de Bumpers en el Robot Rug Warrior.

SEGUIR_LUZ. En esta conducta, el robot utiliza un par de fotoceldas ubicadas en su parte frontal para guiar su navegación hacia fuentes luminosas. Esto es, cuando Rug Warrior verifica mayor luminosidad en su sensor derecho (comparado con el izquierdo) la acción es formar un arco hacia la derecha; si ambas fotoceldas miden la misma intensidad de luz el movimiento es al frente; y cuando la fotocelda izquierda sea la que capte mas luz, el robot

formará un arco a la izquierda.

BUSCAR_LUZ (ENFRENTAR_OBJETIVOS). La finalidad de esta conducta es ubicar el robot de frente a una fuente luminosa. Para esto, el robot gira un par de vueltas (720°) sobre su eje determinando, mediante sus fotoceldas, el valor del punto más luminoso en su entorno. Posteriormente, inicia nuevamente con giros hasta que sus sensores capten como valor medio el valor registrado durante la etapa de reconocimiento. Una vez detectada la fuente de luz, se detendrá, y ajustará su posición hasta que las dos fotoceldas adquieran el mismo valor, situándolo así de frente a la fuente de luz.

SEGUIR_LINEA. Esta conducta tiene por objeto efectuar tareas de navegación sobre un circuito guía libre de obstáculos. Para esto, el robot cuenta con un detector de líneas compuesto por tres sensores infrarrojos. En apariencia, este aditamento dota del sentido de olfato al agente, aunque en realidad lo que se efectúa es un proceso de visión muy rudimentario. Esta conducta se dispara cuando al menos uno de los tres sensores detectan la línea sobre el piso. Si los sensores indican la pérdida gradual de la línea, el mecanismo de control efectuará una corrección proporcional al error para devolver al robot al circuito de navegación. Para compensar la pérdida total de la línea por eventos inesperados como inercia o colisiones, el robot es capaz de generar un arco de búsqueda oscilante hasta volverla a capturar. Este tipo de navegación descansa en rutinas de control proporcional con retroalimentación de la teoría de control clásico.

BUSCAR_LINEA. La finalidad de esta conducta es hacer que el robot retorne al circuito guía tras reconocer el cruce por la línea. Esta es la conducta más compleja del conjunto, ya que además de retomar el circuito el robot debe orientarse en el sentido correcto. Para reconocer el cruce por la línea el robot usa su *linetracker*, mientras que para identificar zonas de navegación emplea una variante de un método global de planificación de trayectorias conocido como descomposición de celdas [Latombe 91].

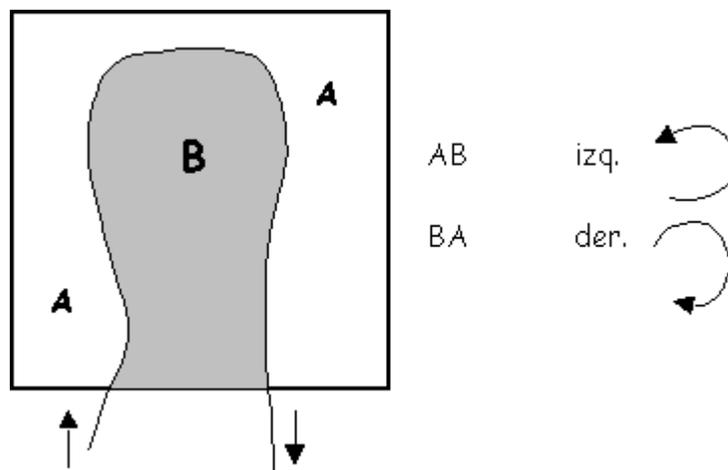


Figura 4.7. Zonas de Navegación de acuerdo con [Latombe 91].

En nuestro dominio, el espacio de navegación se descompuso en dos grandes regiones que el robot puede reconocer. Tal como se muestra en la figura 4.7, la zona A corresponde a la región externa y la zona B a la región interna al circuito. Para orientar al robot en el sentido de viaje correcto al registrar el cambio de área, sus posibles acciones son girar en el sentido antihorario (A B) y horario (BA).

Capítulo 5. Análisis de Resultados

5.1 Resultados Experimentales

Esta sección tiene por objeto evaluar el desempeño de la arquitectura híbrida propuesta considerando la habilidad del agente para resolver el problema de navegación tipo. Durante esta etapa de evaluación se utilizó una computadora personal con procesador pentium MMX, el kit robótico Rug Warrior extendido, y el ambiente artificial AMBART-RW con 8 secciones rectangulares (16 m²). El tipo de pruebas realizadas está relacionado con tres aspectos generales:

1. El proceso de planificación (nivel cognitivo)
2. La ejecución de conductas (nivel físico)
3. Las evoluciones del agente en forma global (ambos niveles)

El proceso de planificación contempla la selección del operador conductual y la obtención de los parámetros de búsqueda. La ejecución de conductas o seguimiento del plan comprende aspectos tales como: la navegación guiada, la conducta orientada a metas, y la capacidad de búsqueda. Y finalmente, las evoluciones del agente en forma global tienen que ver con el grado de efectividad para resolver distintos problemas de navegación.

5.1.1 Proceso de Planificación

La eficiencia de planificación puede expresarse en términos del tiempo de cómputo requerido para establecer el operador conductual y los parámetros de búsqueda. Dado que la complejidad computacional para seleccionar los operadores en esta aplicación puede considerarse despreciable (ver diagrama de flujo de la figura 4.1), las pruebas se concentraron en evaluar la obtención de parámetros de búsqueda por:

1. razonamiento abstracto, y por
2. extracción de la base de planes (BP)

Para evaluar la generación de parámetros por razonamiento abstracto, se midió el tiempo de cómputo invertido en éste proceso incrementando el número de objetivos en el ambiente hasta alcanzar un máximo de 6 (total de objetivos posibles en el problema). En cada prueba, se obtuvo un valor promedio del tiempo después de probar con la misma disposición de las metas ya que cada corrida arrojaba tiempos de cómputo ligeramente diferentes. La figura 5.1a muestra como conforme crece la complejidad del problema también crece la complejidad computacional para resolverlo. Sin embargo, en el peor de los casos, el tiempo de ejecución apenas rebasó los 2 segundos, por lo que puede considerarse satisfactorio tomando en cuenta que este proceso se lleva a cabo solamente una vez y que el plan queda almacenado como ejemplo para uso futuro.

Para evaluar la extracción de parámetros de la BP, se introdujeron al sistema algunos datos de configuración del ambiente tales como metas y prioridades de casos resueltos por procesos de planificación anteriores. Las pruebas consistieron en medir el tiempo invertido en la localización de parámetros recorriendo una BP con 445 registros. Los ejemplos seleccionados correspondieron a cuatro casos ubicados en diferentes posiciones a lo largo de la base. Al igual que en la prueba anterior, se obtuvo un valor promedio del tiempo después de evaluar la misma configuración del ambiente con la intención de mostrar resultados más representativos. La figura 5.1b muestra un crecimiento exponencial del tiempo de ejecución conforme avanzaban los registros de la BP. Aun así, pudo comprobarse que en todos los casos el tiempo invertido en la obtención de parámetros a través de la BP fue menor que el tiempo necesario para generar estos parámetros a través de la descomposición del problema en abstracciones.

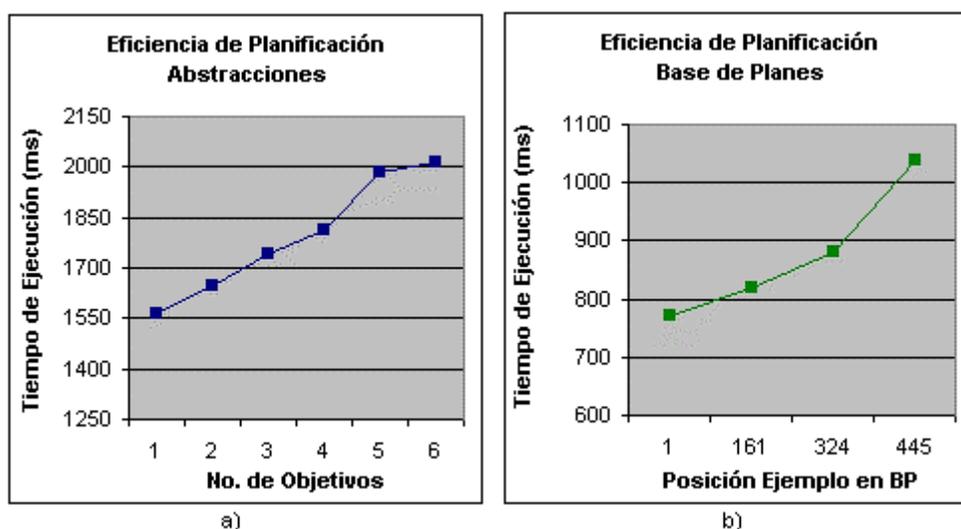


Figura 5.1. Características de Eficiencia durante la Etapa de Planificación.

5.1.2 Ejecución de Conductas

El cálculo del rendimiento en el nivel físico puede ser engañoso ya que es difícil evaluar el comportamiento en forma integral. Esto es, en algunas ocasiones ciertas conductas funcionan bien y en otras no. Por esto, se decidió primero evaluar cada una de las habilidades que componen un operador en forma desacoplada mostrando el rendimiento promedio.

La evaluación de cada red de conductas o habilidades del agente contempló el uso de tres funciones de rendimiento para la navegación reactiva. La primera considera que la navegación sobre la línea guía debe ser fluida, y que la localización de estaciones sea lo más precisa posible. Esto implica que, durante la navegación del robot, la pérdida de la línea sea mínima y el tiempo de navegación sobre ésta sea máxima. Así, podemos establecer como función de rendimiento para la navegación guiada la siguiente ecuación:

$$\eta_{cl} = T_L / (T_L + T_{NL}) \dots (5.1)$$

donde:

- η_{c1} es el rendimiento para la navegación guiada por la línea
- T_L es el tiempo de navegación sobre la línea
- T_{NL} es el tiempo invertido en ajustarse a la línea

La segunda función determina el rendimiento para la conducta orientada a metas considerando la capacidad de robot para fijar su atención en un objetivo. Cada vez que un objetivo esté fuera del campo receptivo de robot, el rendimiento será mínimo, y máximo cuando todas sus acciones están comandadas por la meta. La ecuación 5.2 muestra esta función para el operador INTEGRAL, y la ecuación 5.3 para el operador RUTINA:

$$\eta_{c2} = T_{SL} / (T_{SL} + (T_{EO} + T_{RC})) \dots (5.2)$$

ó

$$\eta_{c2} = T_{EA} / (T_{EA} + T_{LO}) \dots (5.3)$$

donde:

- η_{c2} es el rendimiento para la conducta orientada a metas
- T_{SL} es el tiempo de operación de la conducta "Seguir Luz"
- T_{EO} es el tiempo invertido en evadir obstáculos
- T_{RC} es el tiempo invertido para escapar de colisiones
- T_{EA} es el tiempo de evaluación del ambiente para la búsqueda de objetivos
- T_{LO} es el tiempo invertido en la localización del objetivo

La última relación de rendimiento está dedicada a evaluar la búsqueda del circuito de navegación. Esta función considera que el rendimiento será máximo mientras el robot detecte el camino libre de obstáculos, y mínimo cuando se disparen las conductas de evasión de obstáculos y reacción al contacto. De acuerdo con esto, la relación se puede expresar como sigue:

$$\eta_{c3} = T_A / (T_A + (T_{EO} + T_{RC})) \dots (5.4)$$

donde:

- η_{c3} es el rendimiento durante la búsqueda del circuito de navegación
- T_A es el tiempo de ejecución de la acción AVANZAR al no detectar obstáculos

Por lo tanto, si deseamos evaluar el rendimiento promedio de la arquitectura la relación es la siguiente:

$$\eta_{\text{promedio}} = (\eta_{c1} + \eta_{c2} + \eta_{c3}) / 3 \dots (5.5)$$

Las pruebas para la evaluación de habilidades consistieron en la realización de 10 corridas usando a INTEGRAL y 10 corridas a RUTINA como operadores conductuales. Durante la mitad de las pruebas efectuadas a cada operador, se mantuvo el ambiente libre de obstáculos, mientras que durante la segunda mitad se agregaron tres objetos iguales en diferentes posiciones del área de navegación.

La figura 5.2a muestra el rendimiento para el operador INTEGRAL después de 5 pruebas bajo condiciones similares. En esta gráfica se puede apreciar la facilidad del robot para realizar las tareas de navegación en un ambiente libre de obstáculos, mientras que también puede apreciarse la caída en el rendimiento cuando éstos fueron adicionados. Asimismo puede observarse que, aun en condiciones ambientales similares, la imperfección de los sensores dio lugar a comportamientos no lineales; es decir, el robot mostró acciones diferentes para realizar la misma tarea.

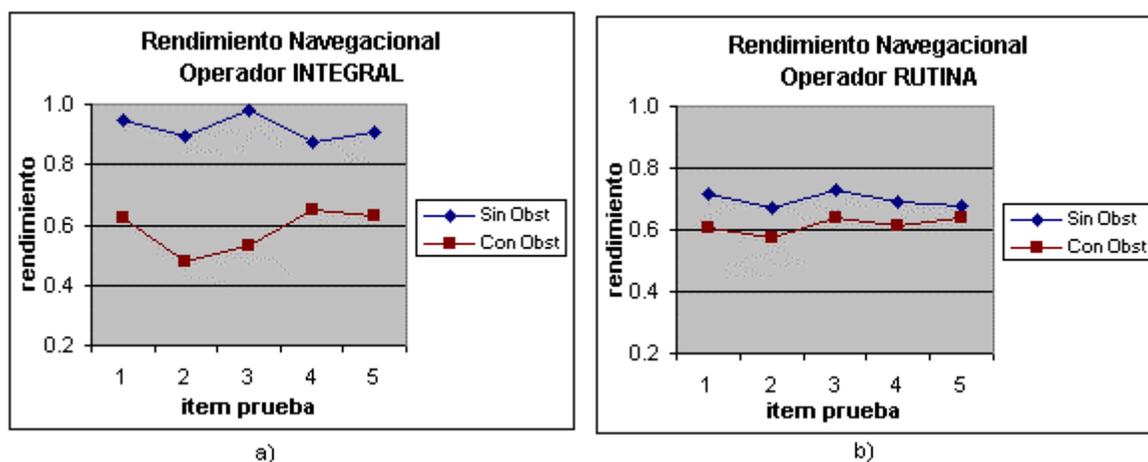


Figura 5.2. Gráfica de Rendimiento Navegacional.

Durante las pruebas realizadas al operador RUTINA, los obstáculos sólo se utilizaron para alterar la percepción de los objetivos y así destacar la capacidad del robot para enfrentarlos. Hay que recordar que este operador no requiere la navegación fuera del circuito ya que sólo emula actividades de supervisión muy generales. Como se puede ver en la figura 5.2b, el rendimiento del operador RUTINA tiende a ser más lineal ya que no implica el disparo de conductas para la evasión de objetos. Por otro lado, el rendimiento disminuye ligeramente ante la interposición de obstáculos entre las estaciones y los objetivos.

Si comparamos las figuras 5.2a y 5.2b, podemos notar que el rendimiento promedio de los operadores INTEGRAL y RUTINA en presencia de obstáculos es muy similar, lo cual indica

que el grado de dificultad en ambos casos es comparable aun cuando las rutinas de sentido sean diferentes.

En términos generales se puede apreciar que después de practicar diversas pruebas bajo condiciones ambientales similares el robot manifestó comportamientos de adaptación diferentes para realizar la misma tarea, comprobando que los sentidos de un robot perciben cambios en el mundo que los humanos ignoran y viceversa.

5.1.3 Evoluciones del Agente en Forma Global

La medida de que tan bien el robot efectuó sus tareas no es función directa del rendimiento ya que en algunos casos pudo mostrar un comportamiento óptimo sin garantizar que hizo lo correcto. Esta sección pretende establecer esta diferencia mediante la evaluación de las evoluciones del agente en forma global. Los rubros que constituyen esta prueba son: la forma en que el agente se orienta al retomar el circuito, su habilidad para llegar a la estación correcta, que tan efectiva es su separación del circuito de navegación, la facilidad para reconocer sus objetivos, su habilidad para abandonar el área de navegación, que tan bien reacciona al contacto, que tan fiel es el seguimiento del plan, cual es su capacidad de orientación regional, que tan fluida es su navegación sobre el circuito, o que tan buena es su habilidad para enfrentar objetivos desde la línea guía.

El criterio de evaluación en este rubro consistió en asignar una calificación en una escala de 0 a 100% indicando si la evolución fue EXCELENTE, BUENA, REGULAR o MALA. La figura 5.3 muestra los resultados promedio de la evaluación de habilidades empleando los operadores conductuales INTEGRAL y RUTINA después de 10 pruebas. Nótese que algunos aspectos no están contenidos en ambas evaluaciones.

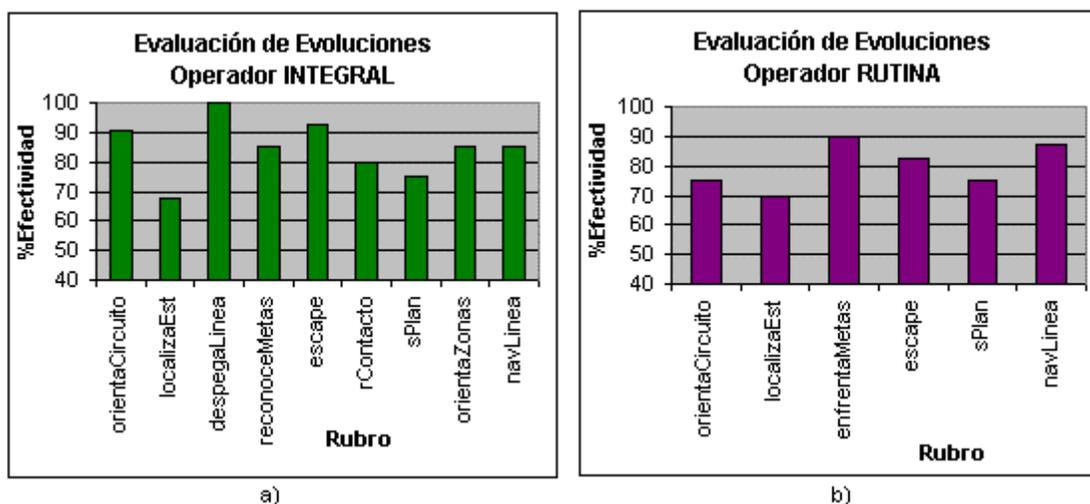


Figura 5.3. Evaluación de Evoluciones

Si analizamos los resultados de la tarea INTEGRAL en la figura 5.3a podemos observar que

los rubros mejor calificados fueron la capacidad de orientación al retomar el circuito, la forma de separarse de la línea, el reconocimiento de metas, y la conducta de escape. Esto significa que los sensores empleados durante estas evoluciones cumplieron satisfactoriamente con su objetivo en la mayoría de los casos. En lo que respecta a RUTINA, puede apreciarse en la figura 5.3b que la navegación sobre el circuito guía y la capacidad para enfrentar metas fueron buenas aunque no excelentes. Esto tiene que ver con el mecanismo de solución del problema ya que en este caso el robot depende en gran parte de sus fotosensores para la detección de objetivos y del seguidor de línea para guiar su navegación. En ambos casos, la localización de estaciones se torna problemática ya que los codificadores incrementales (*shaft encoders*) utilizados eventualmente erraron y por consecuencia afectaron el seguimiento correcto del plan.

Cabe mencionar que estas gráficas muestran la evaluación de las evoluciones en forma global desde el punto de vista de las habilidades conductuales y los resultados no representan totalmente la capacidad para resolver el problema de navegación. Sin embargo, es posible asegurar que aun en los peores casos, el agente localizó objetivos, pudo escapar adecuadamente del área de navegación, y pudo sobrevivir aun después de errar un plan. Por el contrario, en los mejores casos, el agente logró alcanzar todas sus metas gracias a un buen seguimiento y flexibilidad del plan.

5.2 Trabajo Futuro

Existe aún mucho trabajo por hacer en esta área por lo cual se ha dividido el trabajo futuro en:

- Mejoramiento del Prototipo
- Incorporación del Agente en Ambientes Reales

5.2.1 Mejoramiento del Prototipo.

En este rubro surge la necesidad de la retroalimentación al sistema desde el ambiente artificial en tiempo real, dado que con esto será posible reorganizar las metas y permitir acciones congruentes con la situación. Efectivamente, el elemento reactivo permite gran adaptabilidad al medio. Sin embargo, la incorporación de la retroalimentación en tiempo real emulará con mayor realidad la combinación de conductas automáticas con intencionales. Para tal efecto es necesario diseñar un sistema de adquisición de datos basado en sensores de posición, cuyas salidas deberán conectarse a una computadora para su procesamiento. Esta mejora requiere también que el planificador esté conectado todo el tiempo con el microprocesador HC11 del Rug Warrior a fin de llevar a cabo la acción de replanificación en tiempo real. La solución a este problema requiere de un buen manejo de comunicaciones mediante radio modems para evitar conexiones físicas entre la computadora y el microprocesador del robot. Más aun, un sistema de intercomunicación inalámbrico permitiría combinar la teleoperación con arquitecturas híbridas en robots móviles para ampliar su campo de acción.

Otro trabajo relacionado con el prototipo consiste en incorporar algún mecanismo para el

registro de objetivos. Este es un problema relacionado con la posibilidad de clasificar metas para identificar cuales ya han sido atendidas y cuales están pendientes. Hasta el momento, el robot sólo sabe que ya alcanzó una meta, pero no sabe si ya ha sido atendida previamente. La única heurística de búsqueda la proporciona el planificador, el cual informa al robot que estaciones tienen objetivos asociados.

La mayor parte del trabajo futuro está íntimamente ligado con el sistema sensorial. El primer gran problema es el ruido ambiental ocasionado por exceso de luz. Por lo tanto, se propone la realización de un estudio del medio ambiente real para la elaboración de un sistema de filtraje óptico. Esto permitirá contender mejor con cambios ambientales y evitará tomar decisiones erróneas ocasionadas por la detección de fantasmas. Una solución no bien documentada aquí puede ser el uso de una red neuronal entrenada en ambientes típicos para tolerar ciertos umbrales de ruido.

El segundo gran problema sensorial tiene que ver con la incorporación de una función heurística para orientar correctamente al robot durante la búsqueda del circuito de navegación. Como se ha venido mencionando a lo largo de este trabajo, hasta el momento, el robot supone que después de localizar un objetivo el circuito de navegación se encuentra exactamente atrás de él, por lo que simplemente gira media vuelta e intenta navegar en línea recta para eventualmente toparse con el circuito. En algunos trabajos similares se han usado faros de infrarrojos para forzar la navegación hacia una dirección. Este trabajo considera la posibilidad de realizar este tipo de tareas mediante razonamientos geométricos basados en representaciones del ambiente donde un sistema de visión o una brújula pueda facilitar la orientación correcta del robot hacia el circuito.

Un trabajo relacionado con la resolución del problema anterior sugiere también el uso de un anillo de sonares para la formación de mapas topológicos, el inconveniente de un sistema basado en ultrasonido es la reflexión de los pulsos hacia zonas fuera del campo receptivo del robot. Estos sistemas hasta ahora funcionan bien en áreas abiertas. Sin embargo, en espacios cerrados con alta densidad de objetos de diferente geometría éstos no han sido muy exitosos. La solución puede ser una caracterización probabilística de los datos de sonar mediante aproximaciones bayesianas o a través de un análisis de Fourier.

Un trabajo que puede resultar de mucha utilidad para fomentar la investigación en el área de visión computarizada y para dotar de un sistema perceptual completo es la incorporación de una o dos cámaras digitales inalámbricas sobre el cuerpo del Rug Warrior dirigidas por servomecanismos. Gracias a un sistema de visión, el robot sería capaz de formar mapas topológicos empleando detección de bordes en dos y tres dimensiones. Empleando la correlación, podría clasificar objetos por su forma y determinar si se trata de una meta o de un obstáculo. Facilitaría la adquisición de datos con fines de planificación de trayectorias, e incluso podría seguir caminos empleando el análisis de flujo óptico. Por su parte, podría transmitir información con perspectivas diferentes a las de los humanos y de carácter a veces holonómicamente inaccesibles para los mismos. Los servomecanismos facilitarían el diseño de sistemas *pan & tilt* personalizados, y la ausencia de conexiones físicas permitiría una fácil

navegación en el medio ambiente.

Dadas las características de diseño del robot y algunas de las mejoras propuestas también es posible la construcción de un laboratorio de robótica basada en multimedia. La primera piedra para la creación de este sistema ya está puesta, esto es, parte del sistema robótico motivo de este trabajo ha sido implantado en Java con la intención de ser utilizado a través de la red. El uso de una cámara extra mostrando las evoluciones del agente en forma masiva puede asistir a investigadores y estudiantes de los agentes inteligentes en el análisis y mejoramiento de las arquitecturas híbridas. ¡Sólo falta decidirse y hacerlo!.

5.2.2 Incorporación del Agente en Ambientes Reales.

Esta es justamente la parte más interesante del trabajo. El prototipo mostrado está inspirado en la necesidad de realizar tareas peligrosas para los seres humanos. Lo anterior implica dotar a los agentes con habilidades diversas considerando el tipo de ambiente en el cual estarán inmersos. La incorporación de un agente físico en ambientes reales sugiere la selección apropiada del kit robótico o el diseño **ad hoc** de éste. Su fabricación deberá estar sujeta a la normativa internacional en lo que a materiales y componentes mecánicos y electrónicos se refiere. Algunas asociaciones que rigen estos diseños son la IEEE, la ANSI e ISA, así como normas locales dependientes de la aplicación (como los NUREGS para aplicaciones nucleares).

Cabe recordar que este trabajo surgió de un macroproyecto para la detección de fugas de vapor en centrales de procesos con ambientes peligrosos. Por lo tanto, esto es un subproyecto que en su momento puede ser integrado para emplearlo como proyecto de demostración en aplicaciones específicas. Las otras partes del macroproyecto comprenden el diseño de un sensor inteligente para la detección de fugas de vapor, y el alcance de objetivos y adquisición de datos. Además contempla la representación detallada del ambiente real en una maqueta y la simulación de las fugas. Esto último incluye la construcción de pasos a desnivel, pasillos simulados con mampáras, zonas de difícil acceso, una red de tubería plástica presurizada por un pequeño generador de vapor, válvulas de control, boquillas y conexiones cubriendo la mayor parte del área de navegación. Lo importante es la demostración de que el robot pueda superar problemas ambientales típicos de la zona de inspección real, y pueda localizar fluidamente sus objetivos.

Otro tipo de trabajos independientes de la aplicación es la utilización del concepto de los planes como intenciones en las arquitecturas híbridas reactivas-deliberativas en sistemas multiagentes, ya sean agentes de software o robots móviles, quizá el rendimiento puede sorprender. Por su parte, queda abierta la posibilidad de manipular la información de la base de planes con fines de generalización y optimización de la misma. Por ejemplo, después de almacenar una cantidad finita de ejemplos (> 300), es posible entrenar con éstos un perceptrón basado en el algoritmo de retropropagación para aprender a generar planes.

5.3 Conclusiones

Este trabajo presentó una arquitectura que retoma algunas ideas de la IA clásica bajo concepciones modernas de la planificación que conjugadas con los nuevos avances en el área de control inteligente prometen dotar de mayor robustez el diseño de robots autónomos orientados a metas en la resolución de toda una familia de problemas navegacionales. Esta arquitectura muestra una manera sofisticada y madura de seleccionar apropiadamente el operador conductual que mejor resuelva la tarea de navegación en turno. Además permite al agente adaptarse al medio ambiente y alcanzar sus objetivos sin comprometerse con el seguimiento perfecto de un plan. Aunque algunos investigadores han presentado ideas similares de planificación y control de conductas, creemos que este estudio es una demostración reciente de la bondad de la combinación de estas habilidades en agentes físicos inteligentes.

En este trabajo se pueden identificar dos formas importantes de abstracción: taxonómicas y constructivas. Las taxonómicas tienen que ver con el concepto de los componentes del plan (pasos) como conductas ejecutables. Mientras que las constructivas se refieren a la forma que adoptan estos componentes. Entre más abstracto sea un plan, siempre y cuando guíe efectivamente una conducta, el agente puede crearlo más fácil y rápidamente, lo puede almacenar en forma más compacta, y con mayor frecuencia el agente puede reusarlo. El problema es que los planes muy abstractos requieren un esfuerzo computacional mayor para proceder debido al incremento en la flexibilidad. Este incremento en el costo puede minimizarse distribuyendo la instanciación de muchas opciones específicas a través de los diferentes niveles de abstracción que componen un agente jerárquico.

Pocos se han manifestado en contra de los planes como programas tan elocuentemente como Agre y Chapman (1990). Sin embargo, este trabajo está inspirado en la una de las obras más recientes de Barbara Hayes-Roth y Karl Pflieger (1996). Su modelo es específico y operacional, además se muestra a favor de los planes como guía de conductas lo cual permite la combinación conveniente de diversas arquitecturas para la resolución del mismo problema. En general, la obra de Hayes-Roth es el primer trabajo que reconoce la diferencia entre la abstracción de los elementos constitutivos de un plan (ej. pasos del plan) y la forma en que estos elementos están estructurados para formarlos (ej. planes lineales, planes condicionales), lo cual permite mayor adaptación durante el tiempo de ejecución de un agente. Además de la habilidad para adaptarse a cambios ambientales, esta arquitectura tiene la característica de adaptarse a cambios en las habilidades conductuales o conocimiento propio del agente, y de manipular el conjunto de metas o el criterio de objetivo de acuerdo al problema.

Se ha discutido que el modelo de los planes como intenciones ofrece ventajas sobre el modelo tradicional de los planes como programas ejecutables. En casos donde cada conducta se diseña con mucho cuidado, puede parecer redundante contar con dos métodos diferentes para hacer lo mismo. La verdad es que en sistemas muy puntuales con tareas definidas muy concretamente que se realizan en situaciones ambientales muy específicas, efectivamente no se aprecia diferencia. Sin embargo, el modelo de los planes como intenciones se vuelve más

útil conforme los agentes empiezan a ser más complejos y sofisticados, conforme sus tareas comienzan a diversificarse, y conforme el rango de situaciones con el cual tengan que contender sea más variado.

La versatilidad que ofrece la arquitectura Hayes-Roth para utilizar cualquier método de disparo de conductas durante la ejecución ha permitido comprobar la eficiencia y robustez de los sistemas híbridos reactivos-deliberativos. En general, la importancia del proceso de planificación radica en acotar significativamente el espacio de búsqueda para reducir el tiempo de ejecución del agente durante el seguimiento del plan.

La arquitectura de subsumción destaca la posibilidad de que los sensores puedan interactuar a través de todas las capas de conductas. Además no existe ningún compromiso con una estructura de datos unificada o modelo geométrico del mundo, por lo que aun cuando no sea posible mostrar un buen seguimiento del plan el agente siempre termina en la meta. La arquitectura de Brooks es modular y por lo tanto es posible aumentar espectros de habilidades de prioridad variable de acuerdo a las necesidades del problema para mejorar el comportamiento global del agente. Subsumción no permite que dos conductas se activen de manera simultánea gracias a su mecanismo de arbitraje, por tanto en este nivel el problema de la resolución de conflictos es mínimo.

A pesar de que el elemento reactivo presenta desventajas de escalabilidad limitada, el concepto de los planes como intenciones proporciona robustez al agente en relación a su complejidad. Hay que recordar que aunque las metas están implícitas en la estructura del elemento reactivo y no puedan variar respecto al tiempo, el uso de los planes como intenciones en línea permite el redireccionamiento conductual del agente ante situaciones inesperadas. Como se puede notar, los elementos reactivo y deliberativo en la arquitectura propuesta trabajan juntos para cubrir mutuamente sus deficiencias.

Respecto a la aplicación cabe aclarar que el kit robótico utilizado en este trabajo no resuelve problemas de navegación reales. Rug Warrior es un kit diseñado con fines didácticos y experimentales, por lo tanto carece de los sensores y características físicas para usarse en problemas del mundo real. También cabe recordar que sólo se utilizó el problema de navegación en ambientes peligrosos como modelo para evaluar la arquitectura. El tipo de tareas que este robot puede realizar está relacionado con la navegación en ambientes artificiales y la detección de objetivos, por lo que el ambiente debe estar preparado adecuadamente para que el artefacto pueda mostrar sus habilidades.

Considero que la incorporación de agentes inteligentes físicos en actividades complejas, peligrosas o monótonas en diversos dominios como el industrial, militar, espacial, o simplemente en la vida diaria, con sus limitaciones, es completamente factible desde el punto de vista técnico y científico. Sin embargo, es necesario pensar en los beneficios en términos del costo económico debido a que el diseño de robots móviles requiere de inversiones importantes de tiempo, dinero y del personal interdisciplinario adecuado. Lo interesante es que el diseño de estos agentes a nivel prototipo ya lo estamos realizando con éxito en algunas

universidades mexicanas permitiendo abrir al sector productivo una amplia gama de aplicaciones.

Créditos

Es muy importante destacar la colaboración de todas aquellos investigadores y estudiantes que participaron en forma entusiasta, desinteresada, directa e indirecta para llevar a buen término el desarrollo del trabajo presentado. Entre éstos quisiera mencionar a los estudiantes del curso de Control y Sensores Robóticos: Juan Manuel Gutiérrez, Odín Peredo, Pedro Pedrero, Roberto Cruz, Gildardo Medina y Alejandro Rico, con quienes iniciamos el ensamble del kit robótico RugWarrior y practicamos las primeras pruebas operativas.

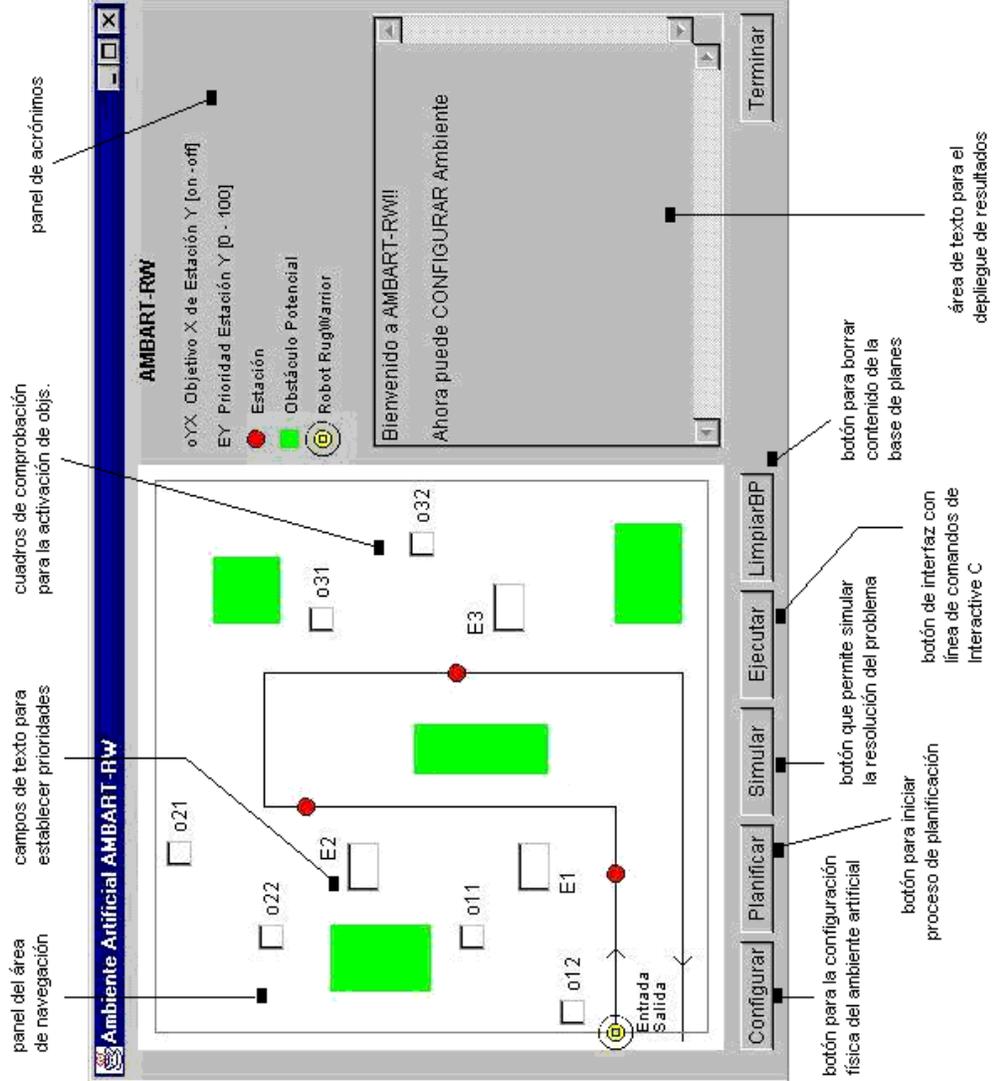
También deseo mencionar la participación de los estudiantes del curso de programación de agentes: Ana Silvia Agüera, Alejandro Rico, Sonia Mestizo y José Luis Llaguno. Este grupo es el responsable de la versión preeliminar simulada de un problema de navegación orientado a la industria empleando agentes inteligentes. Cabe señalar que durante la presentación del proyecto "Simulación de un Problema de Navegación en un Ambiente Industrial Típico", se asumió el compromiso de implantar a nivel de hardware el diseño de un agente capaz de resolver problemas de navegación similares.

Quisiera recordar que la idea original de este proyecto surgió de la inquietud de diseñar un robot detector de fugas de vapor para una central generadora de electricidad. Sin embargo, pese a los nuestros esfuerzos por motivar el uso de robots autónomos en la industria eléctrica, la falta del soporte económico suficiente nos hizo desistir, y orientar nuestro trabajo a un nivel puramente experimental.

Deseo destacar también la colaboración directa de la Maestra Angélica García Vega y la Dra. Amal El-Fallah por sus comentarios y sugerencias en el área de robótica móvil y planificación respectivamente; del Dr. José Ramírez por sus consejos para el mejoramiento de la calidad del presente trabajo y por su aportación durante el curso de programación de agentes; del Dr. Ronald Vogel por su apoyo en la adquisición de equipo y material bibliográfico, y por su dirección siempre entusiasta e incansable en el área de control y sensores; de Ana Silvia Agüera por el apoyo técnico en el área de programación con el lenguaje Java y en el diseño de Interfaces; de Roberto Cruz por su participación directa en la expansión de kit Rug Warrior y asesoría en el área de electrónica.

A manera de resumen, puede apreciarse la colaboración directa o indirecta en este trabajo de prácticamente toda la generación 96-98 de alumnos de la Maestría en Inteligencia Artificial, así como de sus profesores en el área de robótica, agentes inteligentes, y planificación.

Componentes Básicos de la Interfaz Gráfica de Usuario



Lista de Figuras

- 2.1 Máquina de Estado Finito Aumentada (ASFM)
- 2.2 Red de Conductas Simple para la Evasión de Obstáculos
- 2.3 Red de Conductas con un módulo para la supervisión de carga
- 2.4 Configuración General de la Arquitectura Híbrida
- 3.1 Ambiente de Navegación Tipo.
- 3.2 Kit Básico Rug Warrior
- 3.3 Distribución Física de Componentes en el Kit Rug Warrior
- 3.4. Conexión del Microprocesador MC68HC11 con Sensores, Efectores, Auxiliares y Computadora.
- 3.5. Detalle de la Fuentes y Rug Warrior con Sonar
- 3.6. Linetracker y Servo Motor de RC
- 3.7. Kit Robótico Rug Warrior Extendido
- 3.8. Fococeldas en una configuración de divisor de voltaje conectadas a los bits 1 y 0 del puerto E.
- 3.9. Seguidor de Línea
- 3.10. Interfaz del Seguidor de Línea con el Microcontrolador
- 3.11. LED emisor y detector de infrarrojos.
- 3.12. Conjunto sensorial básico de un robot.
- 3.13. Alternativas para la detección táctil.
- 3.14. Circuito para micrófono empleando un amplificador operacional.
- 3.15. Sonar Polaroid y su tarjeta de interfaz.
- 3.16. Interfaz de un sonar polaroid con el MC68HC11
- 3.17. Ambiente Artificial AMBART-RW empleando 8 secciones rectangulares

- 3.18. Interfaz para el puerto paralelo mostrando acoplamiento óptico
- 3.19. Interfaz de Hardware y Ambiente Artificial Iluminado
- 4.1. Diagrama de Flujo del Planificador PLANABS.
- 4.2. Descomposición de los Operadores INTEGRAL y RUTINA.
- 4.3. Red de Conductas para el Operador IR_A_ESTACION.
- 4.4. Red de Conductas para el Operador ALCANZAR_OBJETIVOS
- 4.5. Red de Conductas para el Operador BUSCAR_CIRCUITO.
- 4.6. Arreglo de Bumpers en el Robot Rug Warrior.
- 4.7. Zonas de Navegación de acuerdo con [Latombe 91].
- 5.1. Características de Rendimiento durante la Etapa de Planificación
- 5.2. Gráfica de Rendimiento Navegacional
- 5.3. Evaluación de Evoluciones

Referencias

- [Agre 87] Agre P. E., Chapman D. **Pengi: An Implementation of a Theory of Activity**. In Proceedings of the National Conference on Artificial Intelligence, San Matao, CA, 1987.
- [Agre 90] Agre P. E., Chapman D. **What are Plans for?** Robotics and Autonomous Systems, 6: 17-34, 1990.
- [Agentes 97] Aguera A. S., Llaguno J. L., Mestizo S. L., Reyes A., Rico A. **Proyecto Final del Curso de Programación de Agentes**. Maestría en Inteligencia Artificial, UV/LANIA, Xalapa, México, 1997.
- [Alami 95] Alami R., Robert F., Ingrand F., Suzuki S. **Multi-robot Cooperation through Incremental Plan-Merging**, LAAS / CNRS, IEEE International Conference on Robotics and Automation, Nagoya Japan, 1995.
- [Arkin 90a] Arkin R. C. **The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study**. IEEE Transactions on Systems, Man and Cybernetics, 20(6):1245-1257. 1990.
- [Arkin 90b] Arkin R. C. **Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation**. Robotics and Autonomous Systems, 6:105-122. 1990.
- [Arkin 89] Arkin R. C. **Towards the Unification of Navigational Planning and Reactive Control**, Working Notes of The AAAI Spring Symposium on Robot Navigation, Stanford University, 1989.
- [Arkin 93] Arkin R. C. **Reactive Robotics Systems**, College of Computing, Georgia Intitute of Technology, Atlanta Georgia 1993.
- [Arkin 92] Arkin R. C. **Behavior-Based Robot Navigation for Extended Domains**, Adaptive Behavior, Vol 1, No. 2, MIT, Cambridge, MA, pp 201-225, 1992.
- [Braitenberg 84] Braitenberg V., **Vehicules: Experiments in Synthetic Psychology**, MIT press, Cambridge 1984.
- [Beaufre 95] Beaufre B. , Zeghloul S. **Navigation Method for a Mobile Robot using a Fuzzy Based Method: Simulation an Experimental aspects**. Robotics and Automation, Vol. 10, issue 3, 1995.
- [Brooks 86] Brooks R. A. **A Robust Layered Control System for a Mobile Robot**. IEEE Journal of Robotics and Automation, RA-2, MIT April 1986.

- [Brooks 90] Brooks R. A. **The Behavior Language: User Guide**. Artificial Intelligence Memo 1126, Massachusetts Institute of Technology, 1990.
- [Brooks 91] Brooks R. A., [Intelligence without Reason](#). In *The Artificial Life route to Artificial Intelligence, building embodied, situated agents*. Lawrence Erlbaum, 1991.
- [Brooks 98] Brooks R. A., Breazeal C. Irie R., Kemp C. C., Marjanovi´c M., Scassellati B., Williamson M. M., [Alternative Essences of Intelligence](#), MIT Artificial Intelligence Lab, Cambridge, MA. AAAI, 1998.
- [Chrisman 91] Chrisman L. and Simmons R. **Sensible Planning: Focusing Perceptual Attention**. In *Proceedings of the National Conference on Artificial Intelligence*, San Matao, CA, 1991.
- [Drummond 93] Drummond M, Swanson K., Bresina J. and Levison R. **Reaction-First Search**. In *Proceedings of the National Conference on Artificial Intelligence*, San Matao, CA, 1993.
- [Everett 95] Everett H. R. **Sensors for Mobile Robots: Theory and Application**. Naval Command, Control and Ocean Surveillance Center San Diego, California. A K Peters, Ltd. Wellesley, Massachusetts, 1995.
- [Ewert 80] Ewert J. O. **Neuroethology**. Springer - Verlag, 1980.
- [Fikes 71] Fikes R. E., Nilsson N. **Strips: A New Approach to the Application of Theorem Proving to Problem Solving**. *Artificial Intelligence*, 2: 189-208, 1971.
- [Fikes 72] Fikes R. E., Hart P.E. Nilsson N. **Learning and Executing Generalized Robot Plans**, *Artificial Intelligence*, 3: 251-288, 1972.
- [Firby 93] Firby J., Simmons R. **Mobile Robots II: Architectures for Reactions and Deliberation**. Tutorial MP2: Eleventh National Conference on AI, 1993.
- [Firby 94] Firby R. J. **Task Networks for Controlling Continuous Processes**. In *Proceedings of the Second International Conference on AI Planning Systems*, 1994.
- [Fu -] Fu Daniel D., Hammond Kristian J. and Swain Michael J. **Navigation for Everyday Life**. Department of Computer Science, University of Chicago.
- [Gat 92] Gat E. **Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots**. *Proceedings AAAI-92*, pp. 809-815. 1992.
- [Ginsberg 89] Ginsberg M. L. **Universal Planning: An (almost) Universally Bad Idea**. *IA Magazine*, pages 40-44, 1989.

- [Guerra 97] Guerra H. Alejandro. **Agentes Autónomos y Selección de Acción: Una perspectiva basada en Redes de Comportamiento.** Tesis de Maestría, Maestría en Inteligencia Artificial UV/LANIA, 1997.
- [Hammond 86] Hammond K. J. **CHEF: A Model of Case-based Planning,** In AAAI Proceedings, pp 261-271, 1986.
- [Hayes 96] Hayes-Roth B. and Pflieger K., [Plans Abstractly Describe Intended Behavior](#), Computer Science Department, Stanford University, 1996.
- [Johnson 87] Johnson M. V., Hayes Roth B. **Integrating Diverse Reasoning Methods in the BBL Blackboard Control Architecture.** In Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge, 1987.
- [Jones 93] Jones J. L, Flynn A. M. [Mobile Robots, Inspiration to Implementation.](#) A. K. Peters Wellesley, Massachusetts, 1993.
- [Kaelbling 90] Kaelbling L.P. and Rosenschein S.J. **Acting and Planning in Embedded Systems.** Robotics and autonomous systems, 6(1-2): 35-48, 1990.
- [Latombe 91] Latombe J. C. **Robot Motion Planning.** Kluger Academic Publishers, 1991.
- [Maes 89] Maes Pattie, [How to Do the Right Thing](#), AI Laboratory Vrije Universiteit Brussel and AI Laboratoy MIT, 1989.
- [McDermott 92] McDermott D. **Robot Planning.** IA Magazine, 1992.
- [Newell 61] Newell, A., Simon H. **GPS, A Program that Simulates Thought,** H. Billing, ed., 'Lernende Automaten', Munich Germany, 1961.
- [Nilsson 94] Nilsson N. [Teleo-Reactive Programs for Agent Control.](#) JAIR, 1994.
- [Nilsson 69] Nilsson N. **A Mobile Automaton: An Application of Artificial Intelligence Techniques.** Proceedings of the International Joint Conference on Artificial Intelligence, 1969.
- [Nilsson 84] Nilsson N. **Shakey the Robot.** Technical report, 323: SRI International, 1984.
- [Nilsson 97] Nilsson N and Benson S., [Reacting, Planning and Learning in an Autonomous Agent](#), Robotics Lab, Stanford University, 1997.
- [Nilsson 95] Nilsson N. **Eye on the Price,** AAAI. 0738-4602, 1995.
- [Norman 86] Norman D. and Shallice T. **Attention to Action: Willed and Automatic**

Control of Behavior. Consciousness and self-regulation: Advances in Research Theory, eds Davidson Vol 4, pp. 1-18, 1986.

- [Ramírez 97a] Ramírez J. **Teleo-reactive Neural Networks.** In lecture notes in Computers Sciences 1240, 1997.
- [Ramírez 97b] Ramírez J., Geffner Héctor, [Analysis of Control Programs using Action Models.](#) Departamento de Computacion, Universidad Simon Bolivar, Venezuela, 1997.
- [Rossenblatt 89] Rossenblatt H. J, Payton D. **A Fine-Grained Alternative to the Subsumtion Architecture for Mobile Robot Control.** Proceedings of the International Joint Conference of Neural Networks, IJCNN, 1989.
- [Rosenschein 85] Rosenschein S.J. **Formal Theories of Knowledge in AI and Robotics.** New Generation Computing, 3(4):345-357, 1985.
- [Robert 96] Robert Frédéric. **Coopération Multi-Robots par Insertion Incrémentale de Plans.** Doctorat thèse. Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, France 1996.
- [Russell 95] Russell S., Norvig P. [Artificial Intelligence a Modern Approach.](#) Prentice Hall, 1995.
- [Sacerdoti 74] Sacerdoti E. D. **Planning in a Hierarchy of Abstraction Spaces,** Artificial Intelligence 5: 115-135, 1974.
- [Shiffrin 77] Shiffrin R., Schneider W. **Controlled and Automatic Human Information Processing.** Psychological Review, 84, pp 127-190, 1977.
- [Shoham 93] Shoham Y., **Agent Oriented Programming.** Technical report, MIT Ai Lab, 1993.
- [Sitharama 91] Sitharama, Iyengar, Elfes A. **Autonomous Mobile Robots.** 1991.
- [Suchman 87] Suchman L. **Plans and Situated Actions.** Cambridge University Press, 1987.
- [Tommelein 91] Tommelein I. D. Levitt R. E. and Hayes-Roth B. **SightPlan Experiments: Alternate Strategies for Site Layout Design.** ASCE Journal of Computing in Civil Engineering, 5: 42-63, 1991.
- [Walter 53] Walter W. G., **The Living Brain,** W. W. Norton, New York 1953.
- [Zhang 97] Zhang N. and Liu Wenju, [A Model Approximation Squeume for Planning in Partially Observable Stochastic Domains,](#) JAIR, 1997.

