

Una Aplicación Basada en Redes de Agentes Para Control Autónomo

Daniel A. Ojeda Fernández^a Lizet Liñero Suárez^b Ricardo Ribeiro Gudwin^c
{danof^a, llinero^b, gudwin^c}@dca.fee.unicamp.br

Departamento de Ingeniería de Computación y Automatización Industrial (DCA)
Facultad de Ingeniería Eléctrica y de Computación (FEEC)
Universidad Estatal de Campinas – UNICAMP
Campinas – São Paulo – Brasil

Resumen

Las redes de agentes fueron propuestas e implementadas como una herramienta computacional, que facilita el desarrollo de sistemas inteligentes. En este trabajo, presentamos una aplicación para el problema de control de un vehículo auto-guiado, enfocada como un sistema de control inteligente, y desarrollada usando redes de agentes.

Palabras Clave: Redes de agentes, sistemas inteligentes, control autónomo, sistema de control inteligente.

1. INTRODUCCIÓN

El problema de navegación autónoma es bastante conocido dentro de la comunidad científica; por consiguiente, existen diversos trabajos y enfoques para intentar resolver este problema de la mejor manera [2], [3], [4], [5], [6], [7], [11]. De forma resumida, el problema puede ser enunciado de la siguiente manera: un vehículo autónomo se encuentra dentro de un determinado ambiente, compuesto por obstáculos y metas. El problema consiste en controlar el vehículo, de modo que éste navegue por el ambiente, evitando los obstáculos y alcanzando la(s) meta(s).

Las redes de agentes [9] pueden ser utilizadas, de manera particular, como una herramienta para el modelamiento, análisis, diseño y simulación de sistemas inteligentes. En este trabajo, presentamos un sistema de control inteligente para un vehículo auto-guiado, implementado como una red de agentes. El enfoque adoptado para el desarrollo de esta aplicación, es el mismo seguido por Gudwin [7]. Utilizando un sistema de sensores, el vehículo crea, de forma interactiva, un modelo del ambiente en el que se encuentra. Este modelo es utilizado por el sistema de control inteligente, para determinar las acciones de control que permitan al vehículo navegar por el ambiente y alcanzar la meta, evitando colisionar con los obstáculos.

2. REDES DE AGENTES

Las redes de agentes fueron propuestas por Guerrero [9], como una especialización de las redes de objetos desarrolladas por Gudwin [7]. Inicialmente, las redes de objetos surgieron como una herramienta formal, para la especificación y simulación de sistemas y eventos discretos, dotados de algunas características interesantes tales como aprendizaje y adaptación. Las redes de agentes pueden ser utilizadas en diversos contextos, para el modelaje de sistemas complejos, con el propósito de análisis, así como para la simulación y control de este tipo de sistemas. Particularmente, pueden ser usadas en el análisis, diseño y simulación de sistemas inteligentes. Con este propósito se implementó, dentro de nuestro grupo de investigación, una herramienta computacional llamada ONtoolkit (Object Networks toolkit, que disponibiliza el uso de redes de agentes en las diferentes etapas de desarrollo de un sistema inteligente [9].

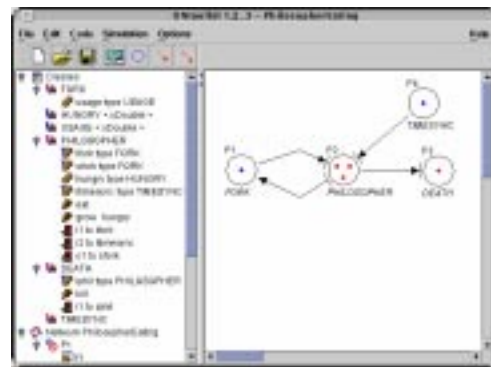


Figura 1. Interface Gráfica de Usuario del ONtoolkit.

Básicamente, el ONtoolkit está formado por tres módulos principales:

- MTON (Multi-Threaded Object Network) engine: constituido por un conjunto de clases, que implementan el formalismo de las redes de agentes y su máquina de simulación.

- ON Desktop (Object Network Desktop) : es el “front-end” para el MTON engine. Ofrece al usuario una interface gráfica con facilidades para la creación, edición y simulación de redes de agentes.
- Plug Server: es un punto de comunicación entre la red de agentes, ejecutándose en el MTON engine, y otras entidades externas (sensores, actuadores, otros programas, módulos de visualización, otros dispositivos y recursos disponibles)

Para una explicación mas detallada de lo que son, y de como funcionan las redes de agentes, se remite al lector a la referencia [9]. La figura 1 muestra una vista de la interface gráfica de usuario del ONtoolkit.

3. MODELO DEL VEHÍCULO

El ambiente por donde navega el vehículo es simple, de forma rectangular y dentro de él pueden ser insertados diversos objetos (paredes u obstáculos) estáticos. El vehículo está equipado con mecanismos de sensoreamiento y actuación, que le permiten recoger información y ejecutar acciones sobre el ambiente.

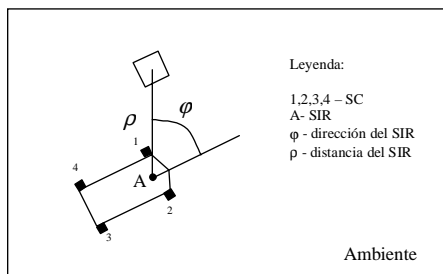


Figura 2. Sensores del vehículo.

Los sensores son de dos tipos (ver figura 2):

- sensor de información remota - (SIR): una simplificación de un mecanismo de visión, para reconocer los objetos del ambiente.
- sensores de contacto - (SC): informan cuando el vehículo entra en contacto con algún objeto del ambiente.

Los actuadores son de dos tipos también:

- actuador de posición de visión: permite posicionar el sensor de información remota, a una distancia y en una dirección determinadas
- actuador de movimiento del vehículo: permiten mover el vehículo, con una velocidad y dirección determinadas.

Las variables de interés, para el control del vehículo, son aquellas que corresponden a su posición dentro del ambiente (x, y, ψ); así como el ángulo de las ruedas en relación al eje longitudinal del vehículo (θ), ver figura 3. Otra variable importante es la velocidad nominal del vehículo (v).

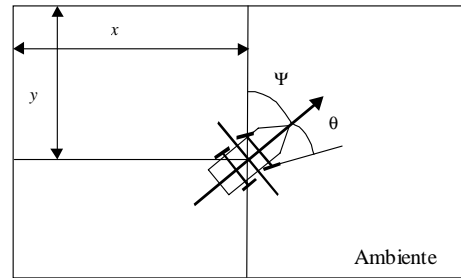


Figura 3. Posición del vehículo dentro del ambiente.

4. SISTEMA DE CONTROL INTELIGENTE

El objetivo de este sistema es generar las acciones de control para que el vehículo navegue por el ambiente y alcance la meta, evitando colisionar con los obstáculos.

La entrada del controlador está dada por las siguientes variables:

x, y, ψ	posición del vehículo
$ax, ay, a\psi$	posición del vehículo en el instante anterior
mx, my	coordenadas de la meta
av	velocidad nominal del vehículo en el instante anterior
$a\theta$	ángulo de las ruedas del vehículo en el instante anterior
$a\phi$	ángulo entre el SIR y el vehículo en el instante anterior
ap	distancia del centro del vehículo al SIR en el instante anterior
c	valores del SC
s	valores del SIR

La salida del controlador está dada por las siguientes variables:

v	velocidad nominal del vehículo a ser aplicada
θ	ángulo de las ruedas a ser aplicado
p	distancia del centro del vehículo al SIR a ser aplicada
ϕ	ángulo entre el SIR y el vehículo a ser aplicado

Según J. Albus, un sistema inteligente está formado por cuatro módulos que se comunican entre si: Percepción Sensorial (PS), Modelamiento del Ambiente (MA), Estimación de Valor (EV) y Generación de Comportamiento (GC), tal como está graficado en la figura 4.

Tomando como referencia esta arquitectura para sistemas inteligentes, referida en [1], nuestro sistema de control inteligente se organiza en siete módulos principales: Módulo de Interface de Entrada (MIE), Módulo de Percepción y Modelamiento del Ambiente (MPMA), Módulo de Generación de Puntos y Arcos (MGPA), Módulo de Generación y Optimización de Trayectorias (MGOT), Módulo de Control Motor

(MCM), Módulo de Control Visual (MCV), Módulo de Interface de Salida (MIS).

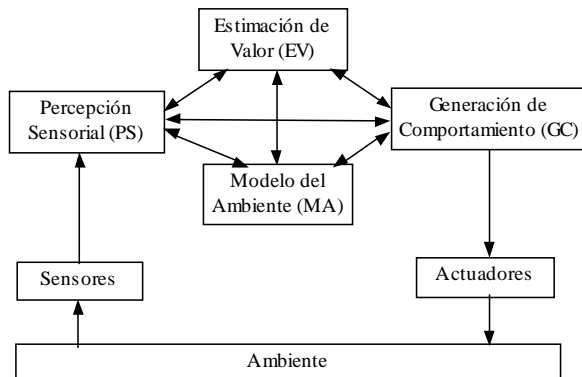


Figura 4. Arquitectura de un sistema inteligente propuesta por J. Albus.

4.1. Módulos del Sistema

4.1.1 Módulo de Interface de Entrada

Este módulo recibe las informaciones provenientes del ambiente, que posteriormente serán usadas para determinar las acciones de control.

4.1.2 Módulo de Percepción y Modelamiento del Ambiente

Este módulo se encarga de procesar las informaciones recibidas por el módulo de interface de entrada, y de generar el modelo del ambiente.

4.1.3 Módulo de Generación de Puntos y Arcos

Usa un algoritmo heurístico para generar los diferentes puntos y arcos, en base a los cuales se construye la trayectoria que el vehículo va a seguir.

4.1.4 Módulo de Generación y Optimización de Trayectorias

Es el encargado de tomar los puntos y arcos generados por el módulo descrito anteriormente, para generar una trayectoria que luego es optimizada y transformada en una acción de control motor.

4.1.5 Módulo de Control Motor

Este módulo calcula las variables que determinan la nueva posición del vehículo dentro del ambiente, es decir, la velocidad nominal (v) y el ángulo de las ruedas (θ).

4.1.6 Módulo de Control Visual

Calcula la nueva posición del sensor de información remota (ρ, ϕ) dentro del ambiente. Su acción es fundamental para el correcto funcionamiento del módulo de percepción y modelamiento del ambiente, y por consiguiente para el resto de los módulos que componen el sistema.

4.1.7 Módulo de Interface de Salida

Integra el control motor y el control visual, dando origen a la acción de control que va directamente para los actuadores del vehículo.

4.2. Coordinación Entre los Módulos del Sistema

Los módulos anteriormente descritos mantienen una coordinación bien definida. Continuamente, cada uno de ellos captura y genera informaciones que van a servir para el trabajo de los módulos restantes.

Inicialmente, se activa el módulo de interface de entrada que recoge todas las informaciones provenientes del ambiente. A continuación se activa el módulo de control visual para la búsqueda y focalización de objetos en el ambiente. Cuando el módulo de control visual termina la focalización, se activa el módulo de percepción y modelamiento del ambiente. Este módulo genera un modelo del objeto focalizado, e intenta integrarlo a los modelos ya existentes. Una vez que concluye la percepción, el sistema de control visual retorna al modo de operación, repitiéndose el ciclo. Esta es la coordinación que existe a nivel de control visual y percepción.

En simultáneo, se ejecuta el control motor, que tiene una coordinación propia. Inicialmente, a partir de la entrada sensorial, y en base al modelo del ambiente, se generan los puntos y arcos, como parte de un plan de movimiento. Aquí termina la actividad del módulo de generación de puntos y arcos. A partir de los puntos y arcos, se genera y optimiza la trayectoria que seguirá el vehículo, dando lugar a un plan de movimiento. Esta etapa es realizada por el módulo de generación y optimización de trayectorias. Se puede observar que estas etapas ocurren de manera secuencial. El módulo de control motor trabaja en simultáneo a los demás módulos. Una vez generado un plan de movimiento (el cual es modificado y/o actualizado constantemente), se genera el control motor con la finalidad de ejecutar dicho plan.

El módulo de interface de salida se ejecuta también de forma independiente. A cada ciclo, recoge la información generada por el control visual y el control motor para generar la acción de control que es enviada a los actuadores.

Es posible observar que existen cuatro secciones del sistema ejecutándose en simultáneo. La primera corresponde al control visual y percepción. La segunda corresponde a la generación de puntos y arcos, y posteriormente la generación del plan, que puede ser considerada también como la generación de un nuevo plan. La tercera corresponde a la generación del control motor, a partir del plan, y la cuarta consiste en la generación de la acción de control y su envío a los actuadores. Cada una de estas secciones, se ejecuta en una secuencia bien definida, lo cual garantiza el correcto funcionamiento del sistema.

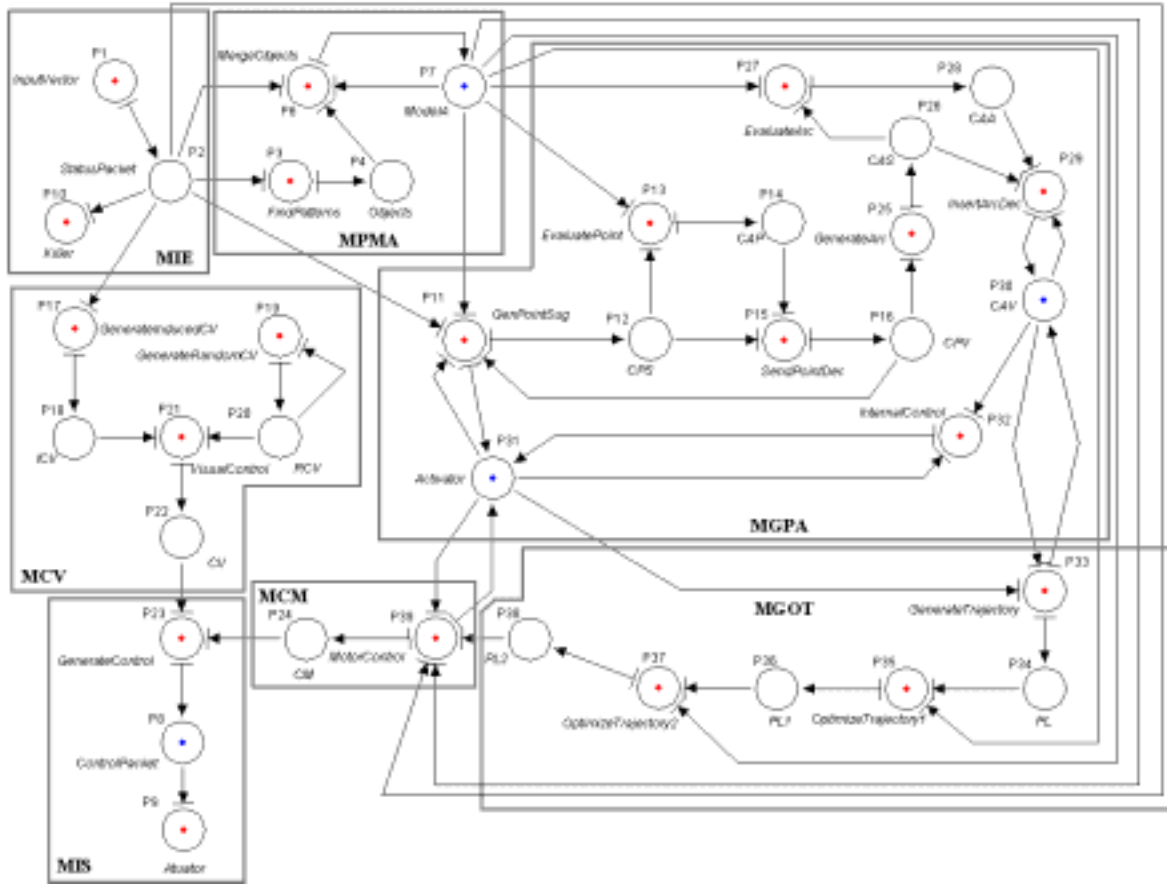


Figura 5. Controlador para el vehículo auto-guiado implementado como una red de agentes.

5. IMPLEMENTACIÓN

El sistema de control inteligente para el vehículo auto-guiado fué implementado como una red de agentes, usando la herramienta computacional ONtoolkit [10].

La figura 5 muestra el Sistema de Control Inteligente como una Red de Agentes.

La secuencia de implementación del controlador puede ser vista gráficamente en la figura 6. El primer paso fué la especificación de la estructura interna de las clases y de la topología de la red (especificación del modelo), ver figura 7 y 8. Luego, fué editado el código específico de cada clase y el código del núcleo de la red (implementación del modelo), ver figura 9. El ONSL (Object Network Specification Language) es un lenguaje definido dentro del ONtoolkit, que provee los formalismos necesarios para la especificación e implementación del modelo.

Para que la red de agentes especificada pueda ser reconocida por el MTON engine, es necesario efectuar un proceso de compilación. En un primer momento, es invocado el ONSLC (Object Network Specification Language Compiler), que genera los archivos .java conteniendo las estructuras y esqueletos de la red. En

seguida, es invocado el compilador Java (javac), generando el código java (.class) a ser ejecutado durante la fase de simulación.

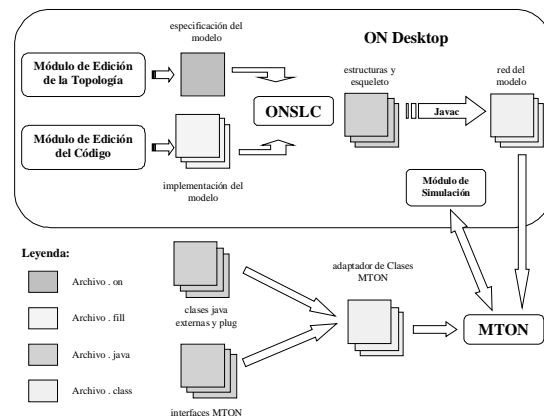


Figura 6. Secuencia de implementación del sistema de control inteligente usando el ONtoolkit

Una de las facilidades que brinda el ONtoolkit, es la posibilidad de usar clases Java externas y plugs. Parte de la dinámica del sistema de control inteligente, para el vehículo auto-guiado, está implementado en clases Java externas, las cuales fueron acopladas y usadas como parte de la red de agentes.

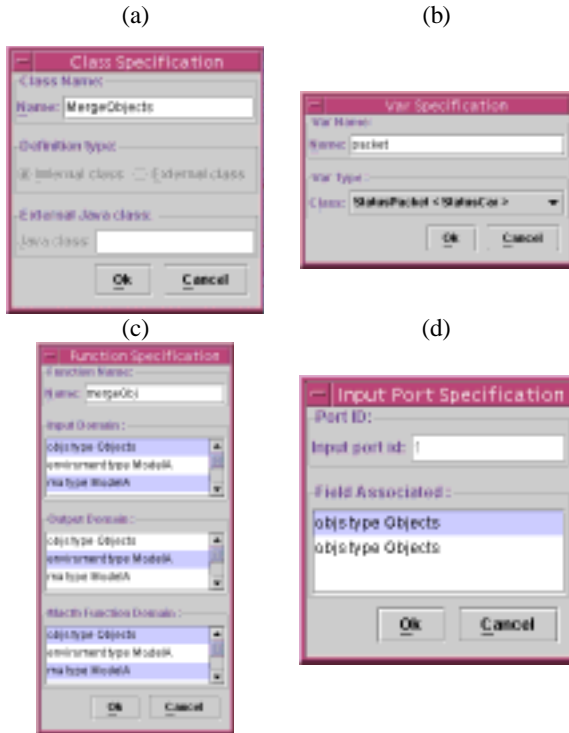


Figura 7. Especificación de la estructura de las clases dentro de la red de agentes. (a) Especificación de una clase. (b) Especificación de una variable. (c) Especificación de una función. (d) Especificación de las puertas de entrada de una clase.

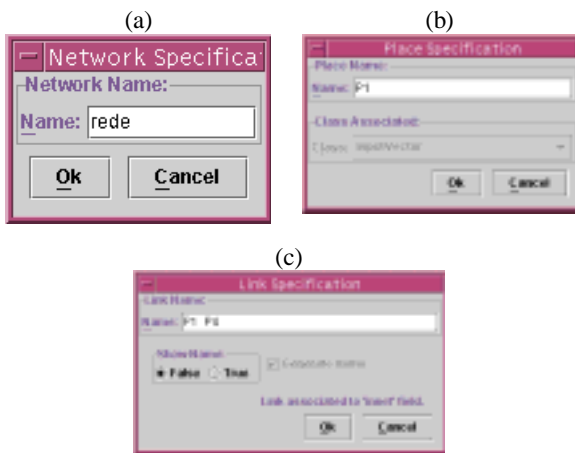


Figura 8. Especificación de la topología de la red de agentes. (a) Especificación de la red. (b) Especificación de los lugares dentro de la red. (c) Especificación de links entre lugares.

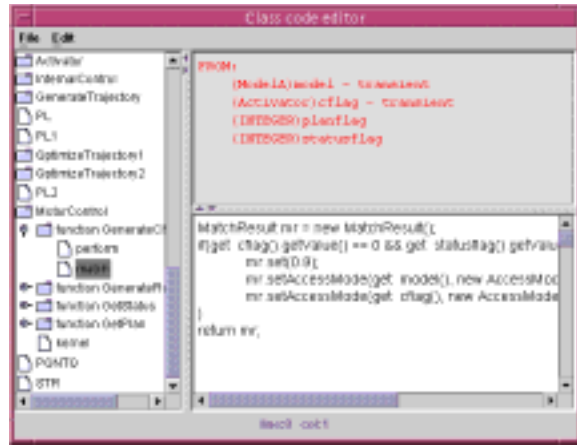


Figura 9. Editando el código de la red de agentes.

El código ejecutable (java.class) y las classes externas debidamente compiladas, son utilizadas por el MTON engine (máquina de simulación de la red de agentes) para efectuar la simulación de la red, ver figura 10. Como fué especificado anteriormente, las classes Java externas, pueden ser utilizadas como parte de las redes de agentes, siempre que implementen las interfaces que proporcionan los mecanismos de seguridad obligatorios para el MTON engine. Todo el proceso de especificación, implementación, compilación y simulación de la red de agentes es efectuado mediante la interacción con el ON Desktop.

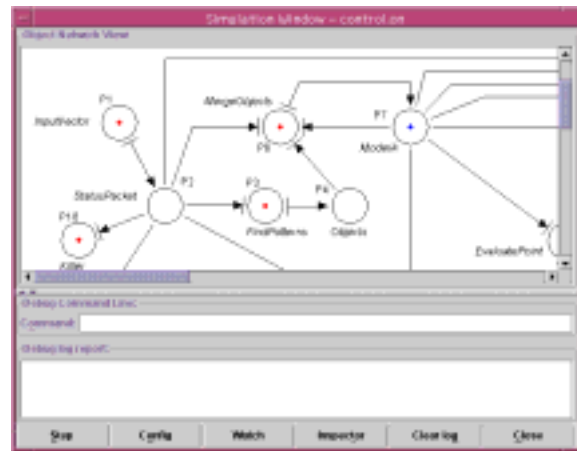


Figura 10. Modo de simulación de la red de agentes.

6. RESULTADOS

Como parte de esta aplicación fué construido un plug, como un módulo de visualización para los resultados de las simulaciones efectuadas.

Fueron utilizados un conjunto de situaciones y ambientes diferentes para verificar los resultados que se obtienen usando el sistema de control inteligente presentado anteriormente. En la mayoría de los casos, el vehículo

auto-guiado consigue llegar satisfactoriamente a la meta evitando los obstáculos. Algunos de estos resultados pueden ser apreciados gráficamente en las figuras 11 y 12.

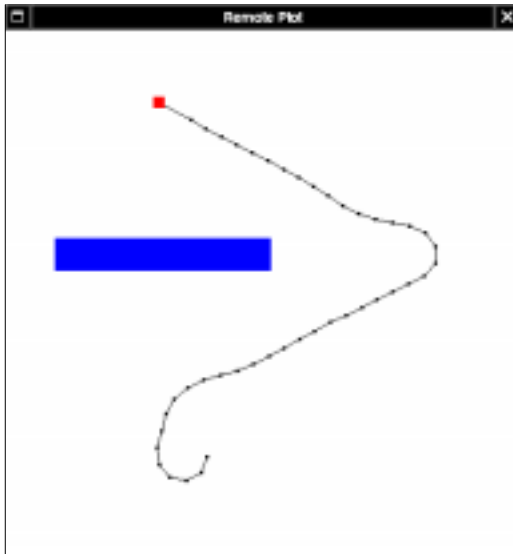


Figura 11. Simulación de la navegación del vehículo auto-guiado – Variante A.

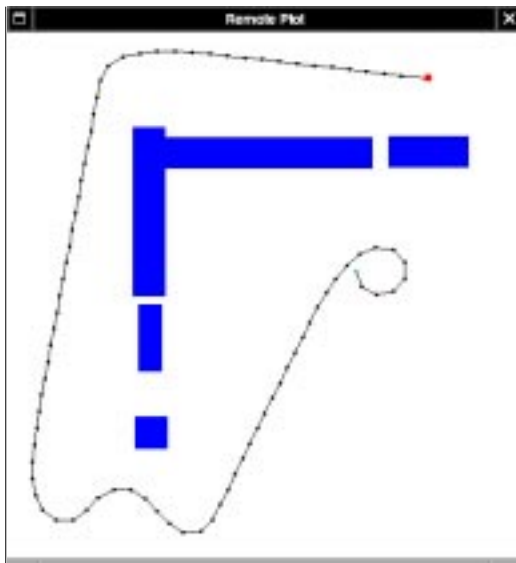


Figura 12. Simulación de la navegación del vehículo auto-guiado – Variante B.

7. CONCLUSIÓN

En el presente trabajo fué presentado un sistema de control inteligente para un vehículo auto-guiado. Dicho sistema de control fué implementado como una red de agentes, usando la herramienta computacional ONtoolkit

(Object Networks toolkit). Los resultados de las simulaciones muestran, de manera general, una solución satisfactoria para este tipo de problemas de control autónomo.

Bibliografía

- [1] Albus, J.S. "Outline for a Theory of Intelligence" - IEEE Transactions on Systems, Man and Cybernetics, vol. 21, n. 3, May/June 1991.
- [2] Beom, H.R.; Cho, H.S. "A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning", IEEE Transactions on Systems, Man and Cybernetics, vol. 25, n. 3, March 1995.
- [3] Brooks, R.A. "Intelligence Without Reason" Proceedings of the Twelfth International Conference on Artificial Intelligence, Vol. 1, pp. 569-595, 1991.
- [4] Chen, C.X.; Trivedi, M.M. "Task Planning and Action Coordination in Integrated Sensor-Based Robots", IEEE Transactions on Systems, Man and Cybernetics, vol. 25, n.4, April 1995.
- [5] De Oliveira, M; Figueiredo, M.; Gomide, F. "A Neurofuzzy Approach to Autonomous Control", Proceedings of the International Conference on Fuzzy Logic, Neural Nets and Soft Computing, pp. 597-598, August 1994, Iizuka - Japan.
- [6] Fan, K.C.; Lui, P.C. "Solving Find Path Problem in Mapped Environments Using Modified A* Algorithm" - IEEE Transactions on Systems, Man and Cybernetics - vol. 24, n. 9, September 1994.
- [7] Gudwin, R.R. "Contribuições ao Estudo Matemático de Sistemas Inteligentes", Tese de Doutorado, DCA-FEE-UNICAMP, Maio 1996.
- [8] Gudwin, R.R; Gomide, F. "Object Networks – A Modelling Tool", Proceedings of FUZZ-IEEE98, WCCIS98 – IEEE World Congress on Computational Intelligence, May 1998.
- [9] Guerrero, J.A.S. "Rede de Agentes: Uma Ferramenta Para o Projeto de Sistemas Inteligentes", Tese de Mestrado, DCA-FEE-UNICAMP, Fevereiro 2000.
- [10] Guerrero, J.A.S; Gomes, A.S.R; Gudwin, R.R. "A Computational Tool to Model Intelligent Systems", Anais do 4º SBAI – Simpósio Brasileiro de Automação Inteligente, Setembro 1999.
- [11] Taylor, C.J.; Kriegman, D. "Vision-Based Motion Planning and Exploration algorithms for Mobile Robots", IEEE Transactions on Robotics and Automation, Vol. 14, No 3, June 1998.