

APLICACIÓN DE TÉCNICAS DE INTELIGENCIA ARTIFICIAL PARA LA NAVEGACIÓN DE ROBOTS MÓVILES UTILIZANDO EL SIMULADOR PLAYER/STAGE

Marcelo Marinelli, Horacio Kuna, Hugo Plenc, Fernando Korol,

Departamento de Informática, Facultad de Ciencias Exactas Químicas y Naturales.
Universidad Nacional de Misiones.

marcelomarinelli@fceqyn.unam.edu.ar, hdkuna@unam.edu.ar, hplenc@fceqyn.unam.edu.ar,
fernandokorol@fceqyn.unam.edu.ar

CONTEXTO

Este proyecto es la continuación de línea de investigación del proyecto acreditado 16Q366 “Evaluación de la Efectividad para Evadir Obstáculos de un Sistema Experto de Navegación para Robot Móvil” y el “Programa de Investigación en Computación” de la Facultad de Ciencias Exactas Químicas y Naturales de la Universidad Nacional de Misiones; también vinculado con el Doctorado en Ingeniería de Sistemas y Computación que funciona en la Universidad de Málaga.

RESUMEN

En la robótica móvil hay distintos problemas típicos que se están investigando, por ejemplo el problema de seguir un contorno le permite a un robot realizar una trayectoria en la modalidad denominada “in door”, en donde se pretende que un robot domestico, de oficina o industrial se desplace llevando una carga desde un punto origen hasta un punto destino siguiendo a una pared y evitando los obstáculos que se le interpongan.

Existen distintas técnicas de Inteligencia Artificial que se aplican a estos sistemas de navegación como Redes Neuronales, Lógica Difusa, Neuro-Difusas, etc.

Por medio del simulador y seleccionando la técnica adecuada se pretende implementar un controlador para la navegación autónoma y evaluar los resultados obtenidos.

Palabras clave: Navegación de Robot, Inteligencia Artificial, Simulador de Entorno.

1. INTRODUCCION

1.1 Tipos de Control de Navegación

Los sistemas de navegación autónoma para robots móviles modernos combinan el control reactivo y el deliberativo en modelos de capas [1], [2]. Esta estratificación se diseña de manera que la capa de nivel mas bajo se encargue del control reactivo basado en los datos que recibe de los sensores, estos pueden ser grupos de radares ultrasónicos, infrarrojos o de barrido láser y permiten interactuar con el entorno encargándose de tareas como ser: seguir a un objeto a una determinada distancia o moverse en un entorno complejo siguiendo una trayectoria que mantenga una distancia fija a una pared.

Las arquitecturas híbridas suelen ser de tres capas y sobre la capa reactiva descrita en el párrafo anterior, interactúan las capas ejecutiva y deliberativa. La primera recibe la información de la capa inferior y realiza una planificación de corto plazo pudiéndose encargar de la localización de la posición y del almacenamiento de los mapas de entorno. La capa deliberativa se encarga de la planificación y utiliza distintos modelos para la toma de decisiones a mediano plazo. Pueden existir modelos con mayor número de capas, como por ejemplo una capa adicional para la interfaz de usuario.

Para la capa de control se utilizan, con muy buenos resultados, los controladores difusos combinados con técnicas de generación automática de conjunto de datos [3].

A fin de aumentar la eficiencia de estos controladores se utilizan técnicas de

optimización basadas en algoritmos genéticos [4], [5] y [6]. La metodología empleada en este trabajo combina el aprendizaje automático de la base de conocimientos utilizando una sistema híbrido, en el que se aplica la cooperación entre reglas (COR) complementado con un algoritmo de optimización mediante colonia de hormigas (OCH) [1] y [7]. Esta combinación tiene la característica de generar bases de reglas en forma rápida, con muy buena precisión.

Se utiliza el algoritmo desarrollado en [1] adaptándolo a un simulador compuesto por un servidor de simulación denominado Player (<http://playerstage.sourceforge.net>) bajo licencia GNU que permite el control de un robot que interactúa sobre los dispositivos de navegación y lee las señales provenientes de sus sensores a través de una conexión TCP/IP. Este servidor se conecta a un simulador de robots en dos dimensiones denominado Stage perteneciente al mismo proyecto. Estos dos programas en conjunto son denominados Player/Stage y pueden realizar la simulación del control de un robot sobre escenarios artificiales de manera que permiten desarrollar algoritmos de control de robots móviles en distintos lenguajes y ponerlos a prueba sin disponer de un robot real.

El proyecto Player fue desarrollado originalmente en el Robotic Research Lab de la USC (University of South California) pero actualmente es un proyecto de código abierto sustentado por un grupo de desarrolladores [9].

1.2 Técnica para generar conjuntos de datos

Para realizar la generación automática del conjunto de datos se siguen los siguientes pasos:

1. Definición de las variables de entrada y salida dependiendo de la configuración del robot: se determinan las variables de control que son enviadas al robot y los valores de las variables de los sensores que dispone el mismo para utilizar en la percepción del entorno.
2. Definición del universo de discurso: la partición de los conjuntos difusos de todas las variables con sus respectivas precisiones.

3. Definición de una función de puntuación que provee la efectividad del controlador difuso.

4. Definición de la función objetivo para evaluar la calidad global del conjunto de reglas.

5. Modelado matemático del comportamiento del robot para definir las ecuaciones que rigen la dinámica del mismo en las situaciones planteadas.

6. Construcción del conjunto de entrenamiento: a partir de la combinación de los valores que toman las variables, en función de los antecedentes, se obtienen los consecuentes combinando todos los valores posibles que pueden tomar.

1.3 Desarrollo de aplicaciones clientes con Player/Stage

En C++ la conexión entre el programa cliente y el servidor se hace operando con objetos locales denominados proxies. El programa interactúa con los métodos de estos objetos abstrayéndose de la comunicación TCP/IP por la red.

Para diseñar un programa cliente se deben cumplir tres pasos:

1. Realizar la conexión con el simulador a través de un objeto de la clase PlayerClient.
2. Interactuar con los sensores y sistemas de tracción con la clase correspondiente.
3. Establecer un bucle que periódicamente interactúe con los sensores y en función de la telemetría envíe los datos de control correspondientes a los sistemas de tracción.

Inicio de programa

```
#include <libplayerc++/playerc++.h> // se declara la librería
```

```
PlayerClient robot(gHostname, gPort); // realiza la conexión con el robot o simulador
```

```
Position2dProxy pp(&robot, gIndex); // instancia la clase del sistema de tracción
```

```
LáserProxy lp(&robot, gIndex); // instancia la clase correspondiente al barrido láser
```

Inicio de bucle

```
robot.Read(); // realiza la conexión con el robot o simulador
```

d=lp[i] // toma lectura del láser

Rutina de control

pp.SetSpeed(vel_lineal ,vel_angular); // envía datos de control de tracción

fin de bucle

fin de programa

2. LÍNEAS DE INVESTIGACION Y DESARROLLO

Dentro de las líneas de investigación que contienen el “Programa de Investigación en Computación”, este proyecto se enmarca en el área de inteligencia artificial en donde se aplican técnicas de lógica difusa y algoritmos bioinspirados, estas técnicas se utilizarán en trabajos futuros para:

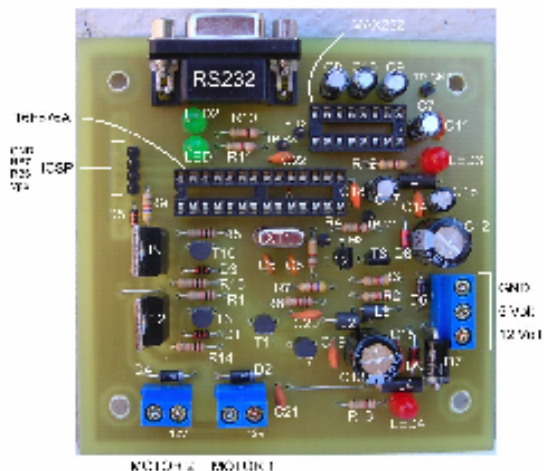
- Controladores difusos aplicados al proceso de elaboración de yerba mate.
- Sistemas de control de navegación para robot utilizando sensores láser.
- Aplicación de controladores difusos para el control de invernaderos con riego artificial.

3. RESULTADOS OBTENIDOS/ESPERADOS

En el Proyecto “Evaluación de la Efectividad para Evadir Obstáculos de un Sistema Experto de Navegación para Robot Móvil” se diseñó una plataforma móvil con un sistema de tracción diferencial compuesto por dos motores de corriente continua.

Para el control de velocidad de los motores se desarrolló un control PWM con un microcontrolador PIC de la empresa Microchip Technology Inc., se diseñó un software que permite entregar cinco valores de velocidad a cada uno de los motores en concordancia con los cinco valores difusos que entregará el controlador difuso.

Las señales de control se interconectan con la CPU principal del sistema por medio de un puerto RS232.



Disposición de Componentes

Fig.1 placa de control PWM

Actualmente se está trabajando en el diseño del controlador difuso para el sistema de navegación.

4. FORMACIÓN DE RECURSOS HUMANOS

En esta línea de investigación se prevén desarrollar dos Tesis de grado de la carrera de Licenciatura en Sistemas de Información y dos Tesis doctorales del Doctorado en Ingeniería de Sistemas y Computación del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga

5. BIBLIOGRAFIA

- [1] M. Mucientes, J. Casillas. “Quick Design of Fuzzy Controllers With Good Interpretability in Mobile Robotics”. IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 15, NO. 4, pp. 636-651, 2007.
- [2] S. Russell, P. Norvig, “Inteligencia Artificial Un Enfoque Moderno”. Madrid, Pearson Educacion S.A. 2004.
- [3] M. Mucientes, R. Iglesias, C. V. Regueiro, A. Bugarini, and S. Barro, “A fuzzy temporal rule-based velocity controller for mobile robotics”. Fuzzy Sets Syst., vol. 134, pp. 83–99, 2003.

- [4] B. del Brio, A. Molina “Redes Neuronales y Sistemas Borrosos”. W. Pedrycz, Ed. Norwell, Alfaomega, pp. 289–306, 2007.
- [5] A. Bonarini, “Evolutionary learning of fuzzy rules: Competition and cooperation”. in *Fuzzy Modelling: Paradigms and Practice*, W. Pedrycz, Ed. Norwell, MA: Kluwer Academic, pp. 265–284, 1996.
- [6] M. Mucientes, D. L. Moreno, C. V. Regueiro, A. Bugarini, and S. Barro, “Design of a fuzzy controller for the wall-following behavior in mobile robotics with evolutionary algorithms”. in *Proc. Int. Conf. Inf. Process. Manage. (Uncertainty Knowledge-Based Syst. IPMU’2004)*, Perugia, Italy, pp. 175–182, 2004.
- [7] J. Casillas, O. Cordon, I. Fernandez de Viana, F. Herrera, “Learning cooperative linguistic fuzzy rules using the best-worst ant system algorithm”. *International Journal of Intelligent Systems* 20:4, pp. 433-452. ISSN 0884-8173. doi:10.1002/int.20074, 2005.
- [8] B. Gerkey, R. Vaughan, K. Stoy, A. Howard, G. Sukhatme, M. Mataric. “Most Valuable Player: A Robot Device Server for Distributed Control”. Robotics Research Labs, University of Southern California Los Angeles, CA 90089-0721, USA.
- [9] Collett, T.H., MacDonald, B.A., Gerkey, B.P.: “Player 2.0: Toward a Practical Robot Programming Framework”. *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005)*, 2005.
- [10] J. Urzelai, J. P. Uribe, and M. Ezkerra, “Fuzzy controller for wallfollowing with a non-holonomous mobile robot”. *Proc. 6th IEEE Int. Conf. Fuzzy Syst. (Fuzz-IEEE’97)*, Barcelona, Spain, pp. 1361–1368, 1997.34
- [11] J. Casillas, O. Cordon, and F. Herrera, “COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 4, pp. 526–537, 2002.
- [12] J. Casillas, O. Cordon, “COR methodology: A simple way to obtain linguistic fuzzy models with good interpretability and accuracy,” in *Accuracy Improvements in Linguistic Fuzzy Modeling*, J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, Eds. Heidelberg, Germany: Springer, 2003.
- [13] M. Dorigo and T. Stutzle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.
- [14] O. Cordon, F. Herrera, I. F. de Viana, and L. Moreno, “A new AGO model integrating evolutionary computation concepts: The best-worst ant system,” in *Proc. 2nd Int. Workshop Ant Algorithms*, Brussels, Belgium, pp. 22–29, 2000.