

VI Congreso Iberoamericano de Docencia Universitaria



La **opción** por la **interdisciplinariedad**.
El **estudiante** como **protagonista**.

COMUNICACIÓN

4, 5 y 6 de
noviembre
2010 LIMA - PERÚ



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

www.pucp.edu.pe/vicidu

ESTUDIO Y PRÁCTICA DEL CONTROL BORROSO EN UN MUNDO VIRTUAL

Ángel ARROYO¹, Luís USERO², Óscar CALVO³

Escuela Universitaria de Informática^{1,3}, Escuela Politécnica²

Universidad Politécnica de Madrid^{1,3}, Universidad de Alcalá de Henares²

Resumen

Hoy en día, el estudio del Control Borroso se encuentra dentro de los planes de estudio de gran parte de las Ingenierías. Los entornos virtuales tipo Second Life u OpenSim, posibilitan la recreación de distintas actividades formativas (clase magistral, acción tutorial, prácticas). En este artículo presentamos nuestra experiencia de varios años en la utilización de estos espacios como entornos de experimentación en el diseño y desarrollo de sistemas simulados. En nuestro caso, estos sistemas reflejan el comportamiento complejo mediante el trabajo conjunto de un determinado grupo de controladores borrosos.

Palabras Clave: Control Borroso, mundos virtuales, Second Life, metaversos

1. Introducción

El aprendizaje del Control Borroso, habitualmente, se inicia con un estudio teórico de la Lógica Borrosa (Lógica Difusa o *Fuzzy Logic*). Mediante la lógica borrosa podemos manejar matemáticamente conceptos definidos con incertidumbre; es decir, conceptos que son vagos, ambiguos e imprecisos (Juan es alto; hoy hace mucho frío; María es joven...). Los problemas del mundo real están plagados de términos imprecisos y cualquier sistema que pretenda abordar estos problemas deberá incluir la imprecisión en sus razonamientos al igual que lo hacen los humanos. En la lógica borrosa, se usan modelos matemáticos para representar nociones subjetivas, como caliente/tibio/frío, transformándolos en valores concretos que puedan ser manipulados por los ordenadores.

El Control Borroso (*Fuzzy Control*) surge de la aplicación de la teoría de la Lógica Borrosa al campo de los Sistemas de Control. Este tipo de Sistemas de Control es una tecnología emergente aplicada al campo de la automatización industrial que permite realizar tareas de control habitualmente realizadas con Controladores Programables o PLCs. El Control Borroso nos permite diseñar y construir aplicaciones de control representando el conocimiento de los expertos mediante reglas lingüísticas del tipo "IF Distancia = Baja AND Velocidad = Baja THEN Frenada = Alta". Esto nos permite diseñar sistemas de control para una amplia variedad de tareas de control (bucle cerrado o abierto, monovariable o multivariable, para sistemas lineales o no lineales) dentro de una amplia gama de aplicaciones (clasificación, reconocimiento de patrones, procesamiento de imágenes digitales, visión artificial, toma de decisiones en tiempo real, diagnóstico de sistemas, inteligencia artificial en vídeo-juegos, sistemas de control de aparatos de aire acondicionado, vehículos, lavadoras, cámaras, ascensores...) [1][2].

Nuestra propuesta, en la que llevamos trabajando varios años [3][4], consiste en emplear espacios virtuales que cuentan con motor físico y que permiten la simulación de estos procesos sin los costes asociados a ese mismo aprendizaje en el mundo real.

2. Entorno de trabajo

En nuestro caso contamos con una isla en Second Life (Tesis Island) y con varias en OpenSim en las que cada año ponemos en práctica el desarrollo de distintos controladores borrosos por parte de los alumnos.



Fig. 1 TESIS: Tierra para la Experimentación en Sistemas Inteligentes Simulados

Como se puede observar en la anterior imagen, nuestra isla cuenta con distintos espacios que posibilitan la recreación de distintas actividades formativas (clase magistral, acción tutorial, prácticas sobre sistemas simulados...). En la parte inferior izquierda se puede ver el anfiteatro en el que se imparten clases teóricas. En el edificio azul de la parte superior derecha tenemos el edificio en el que se encuentran los despachos virtuales de los profesores y algunas zonas comunes para todos los usuarios del espacio virtual. En la parte central de la isla se encuentra el edificio dedicado a exposiciones en el que se muestran los mejores trabajos realizados por los alumnos así como algunos tutoriales de consulta útiles para la realización de las prácticas dentro de Second Life. En la parte inferior derecha se encuentran los edificios de los patrocinadores que nos ayudan económicamente a mantener esta novedosa experiencia y, en la parte superior izquierda se encuentra la zona de trabajo para los alumnos en la que se ubican los distintos elementos necesarios para la realización de cada una de las prácticas que llevan a cabo (en esta imagen aparece un campo de fútbol ya que en el momento de tomar la fotografía se está preparando una competición virtual en la que los alumnos tienen que diseñar agentes autónomos que puedan jugar al fútbol en equipo).

3. Metodología docente

Como ya hemos comentado, contamos con un espacio virtual en el que los alumnos pueden recibir clases, acción tutorial y realizar las prácticas preestablecidas. En nuestro caso, los alumnos que utilizan este espacio virtual son los de las asignaturas de Sistemas Inteligentes y Agentes Inteligentes. Ambas asignaturas se imparten en los grados de Ingeniería de Software e Ingeniería de los Computadores de la Universidad Politécnica de Madrid.

En la presentación de la asignatura se les da de alta en TESIS y se les proporcionan las herramientas necesarias para utilizar el espacio virtual. Las clases teóricas sobre Control Borroso se llevan a cabo en clases presenciales en aula y las clases prácticas se imparten en el anfiteatro virtual que se encuentra en TESIS. Durante las clases prácticas los alumnos pueden probar sus diseños dentro de la isla, comprobar su funcionamiento y consultar dudas a los profesores que allí se encuentren.

Con el objetivo de hacer más ameno el aprendizaje de estas técnicas, habitualmente hacemos que los controladores borrosos a diseñar y probar por los alumnos tengan que competir con otros agentes autónomos desarrollados por los profesores. Por ejemplo, durante el curso 2008-2009, los alumnos tenían que diseñar un conjunto de controladores borrosos que definieran el comportamiento de peces predadores. Estos peces tenían que estar formados por un objeto o grupo de objetos con la capacidad de desplazamiento por el entorno acuático de la isla sorteando los obstáculos estáticos o móviles que pudieran encontrarse en su camino. Durante su desplazamiento tenían que capturar el mayor número posible de unidades de alimento (presas diseñadas por los profesores). Para hacerse con las piezas de alimento los objetos creados por los alumnos tenían que contactar físicamente con la pieza. Este contacto sumaba tantos puntos como unidades de alimento tenía la pieza (mayor número de unidades cuanto mayor era la dificultad del contacto).



Fig. 2 Competición predador-presa en TESIS

4. Ejemplo de Controlador Borroso

Actualmente, Second Life se emplea como entorno educativo por parte de un gran número de instituciones para llevar a cabo diferentes acciones formativas [5][6]. En las prácticas realizadas en nuestra universidad, los alumnos tienen que diseñar controladores borrosos que, en conjunto, simulen el comportamiento complejo requerido (en este caso son peces virtuales pero estamos realizando una nueva experiencia con jugadores virtuales de fútbol).

Para facilitar el diseño y desarrollo de los controladores borrosos, los alumnos emplean una herramienta de código abierto en java conocida como jFuzzyLogic [7]. Esta herramienta permite emplear el estándar FCL (Fuzzy Control Language) en el que cada controlador borroso se configura con un bloque en el que es preciso identificar las variables de entrada y salida así como las reglas de inferencia y los métodos de fuzzificación y defuzzificación de los valores de dichas variables. A continuación se muestra un ejemplo de controlador borroso definido en FCL.

```
Terminal — java — 95x48
FUNCTION_BLOCK ControladorVelocidad

VAR_INPUT
    numObst : REAL;
    velocidad : REAL;
END_VAR

VAR_OUTPUT
    accion : REAL;
END_VAR

FUZZIFY numObst
    TERM media := (2.0, 0.0) (6.0, 1.0) (9.0, 1.0) (13.0, 0.0) ;
    TERM muchos := (8.0, 0.0) (20.0, 1.0) ;
    TERM pocos := (0.0, 1.0) (7.0, 0.0) ;
END_FUZZIFY

FUZZIFY velocidad
    TERM alta := (2.5, 0.0) (5.0, 1.0) ;
    TERM baja := (0.0, 1.0) (2.5, 0.0) ;
    TERM media := (1.0, 0.0) (2.0, 1.0) (3.0, 1.0) (4.0, 0.0) ;
END_FUZZIFY

DEFUZZIFY accion
    TERM acelerar := (0.0, 0.0) (5.0, 1.0) ;
    TERM frenar := (-5.0, 1.0) (0.0, 0.0) ;
    TERM mantener := (-5.0, 0.0) (-1.0, 1.0) (1.0, 1.0) (5.0, 0.0) ;
    METHOD : COG;
    DEFAULT := 0.0;
    RANGE := (-5.0 .. 5.0);
END_DEFUZZIFY

RULEBLOCK Rules
    ACT : MIN;
    ACCU : MAX;
    AND : MIN;
    RULE 1 : IF (velocidad IS baja) AND (numObst IS pocos) THEN accion IS acelerar;
    RULE 2 : IF (velocidad IS baja) AND (numObst IS media) THEN accion IS acelerar;
    RULE 3 : IF (velocidad IS baja) AND (numObst IS muchos) THEN accion IS mantener;
    RULE 4 : IF (velocidad IS media) AND (numObst IS pocos) THEN accion IS acelerar;
    RULE 5 : IF (velocidad IS media) AND (numObst IS media) THEN accion IS mantener;
    RULE 6 : IF (velocidad IS media) AND (numObst IS muchos) THEN accion IS frenar;
    RULE 7 : IF (velocidad IS alta) AND (numObst IS pocos) THEN accion IS mantener;
    RULE 8 : IF (velocidad IS alta) AND (numObst IS media) THEN accion IS frenar;
    RULE 9 : IF (velocidad IS alta) AND (numObst IS muchos) THEN accion IS frenar;
END_RULEBLOCK

END_FUNCTION_BLOCK
```

Fig. 3 Ejemplo de controlador de velocidad en FCL

Este controlador se emplea para ajustar la velocidad de los objetos en función de la velocidad que llevan en ese momento y del número de obstáculos que se encuentran en su camino. Por lo tanto, está compuesto por dos variables de entrada (numObst y velocidad) y por una variable de salida que frena o acelera el objeto. El conocimiento del controlador se encuentra reflejado en los diferentes conjuntos borrosos que definen las etiquetas lingüísticas empleadas y en el conjunto de reglas del controlador.

En la siguiente figura mostramos el conjunto de todas las variables de entrada y de salida así como un ejemplo de la salida obtenida por este controlador para una velocidad de entrada de 3.2 metros por segundo y un total de 11 obstáculos detectados por el objeto en la dirección de su movimiento.

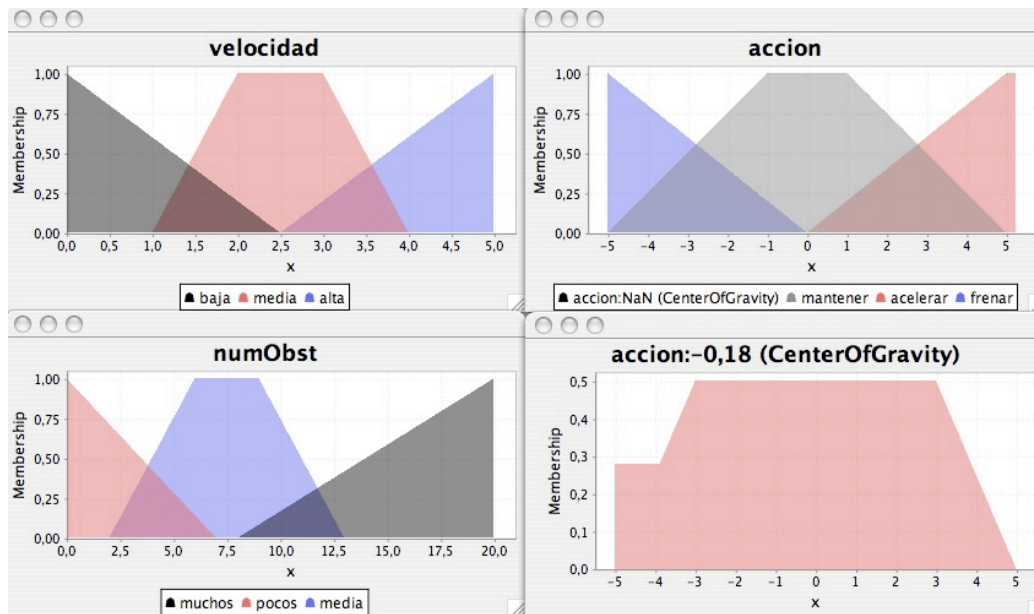


Fig. 4 Variables de entrada y salida del controlador

5. Conclusiones

En Second Life y en otros metaversos [8] poblados por múltiples objetos estáticos y por agentes autónomos con la capacidad de trasladarse por el espacio virtual y de interactuar con su entorno podemos simular sistemas complejos. De forma similar a lo que representa el mundo real para la robótica móvil, estos mundos virtuales, con sus ventajas e inconvenientes, nos sirven como campo de pruebas para un gran número de investigaciones en las distintas áreas de la Inteligencia Artificial.

Una vez que los alumnos cuentan con las bases teóricas necesarias para la comprensión del Control Borroso y, tras una amplia descripción de la multitud de campos en los que puede aplicarse, el alumno puede realizar un conjunto de prácticas dentro del entorno virtual dejando a un lado la costosa experimentación en entornos reales.

Gracias a las herramientas empleadas, descritas anteriormente, los alumnos consiguen una retroalimentación muy positiva sobre el funcionamiento de los controladores borrosos diseñados. Además, también se obtienen otros resultados muy interesantes: (a) los alumnos pueden probar en tiempo real el funcionamiento de sus controladores tantas veces como sea preciso, (b) el entorno social en el que realizan sus prácticas les permite colaborar en el desarrollo, (c) los profesores pueden interactuar en el mismo medio con los alumnos y comprobar *in situ* el correcto funcionamiento de los controladores y, (d) los sistemas desarrollados pueden ser todo lo complejos que se necesite contemplando la interacción en tiempo real de un conjunto de sistemas de control.

6. Referencias

- [1] ROSS, T. *Foundations of Fuzzy Control*. Wiley. 2007.
- [2] KECCMAN, V. *Learning and Soft Computing. Support Vector Machines, Neural Networks and Fuzzy Logic Models*. The MIT Press. 2001.
- [3] ARROYO, A.; SERRADILLA, F.; CALVO, O. (2009) Multimodal Agents in Second Life and the New Agents of Virtual 3D Environments. *Methods and Models in Artificial and Natural Computation*. Lecture Notes in Computer Science (LNCS). Vol. 5601. pp. 506-516. Springer Berlin - Heidelberg.
- [4] ARROYO, A.; SERRADILLA, F.; CALVO, O. (2010) Modeling Spatial-Temporal Context Information in Virtual Worlds. *Methods and Models in Trends in Applied Intelligent Systems*. Lecture Notes in Computer Science (LNCS). Vol. 6096. pp. 437-447. Springer - Heidelberg.
- [5] CHANG, L. "Second Life Learning Community: a peer based approach to involving more Faculty members in Second Life". *Proceedings of the Second Life Education Workshop at the Second Life Community Convention*, pp. 6-10, San Francisco, USA. August 20, 2006.
- [6] KEMP, J; LIVINGSTONE, D. "Putting a Second Life Metaverse skin on learning management systems". *Proceedings of the Second Life Education Workshop at the Second Life Community Convention*, pp. 13-18, San Francisco, USA. August 20, 2006.
- [7] CINGOLANI, P. (2005) *jFuzzyLogic*: Open Source Fuzzy Logic library and FCL language implementation. Last version: 2010. [<http://jfuzzylogic.sourceforge.net/html/index.html>]
- [8] STEPHENSON, N. *Snow Crash*. Bantam Books, 1992.