Entorno de Simulación

MINDSTORMS

Introducción

Temática del proyecto

**LaInteligencia Artificial**es un área de las ciencias de la computación, dedicada a estudiar la creación de máquinas inteligentes [2]. El enfoque de agentes inteligentes en esta área ha tenido un constante crecimiento. Un agente se define como todo aquello que puede considerarse que percibe su ambiente mediante sensores y actúa en tal ambiente por medio de efectores [3]. Así un agente en función de la información que recibe de su ambiente y la definición de la meta a alcanzar, debe implementar un mecanismo de toma de decisión que le permita seleccionar la mejor acción, la cual modifica el ambiente de alguna manera.

**La simulación** es una técnica muy poderosa y ampliamente usada en las ciencias para analizar y estudiar sistemas complejos. Simular, es reproducir artificialmente un fenómeno o las relaciones entrada-salida de un sistema. Esto ocurre siempre cuando la operación de un sistema o la experimentación en él son imposibles, costosas, peligrosas o poco prácticas.

Un entorno de simulación es un laboratorio virtual, el cual brinda herramientas para el análisis, comprensión y experimentación de modelos de sistemas reales.

Este proyecto final de carrera de investigación y desarrollo fue pensado para dar soporte a la cátedra Inteligencia Artificial(IA) dictada en la carrera Ingeniería en Sistemas de información (ISI) de la Universidad Tecnológica Nacional, Facultad Regional Santa Fe (FRSF), Argentina, suministrando a los alumnos una nueva herramienta para facilitar el entendimiento de la cátedra, usando para esto una aplicación que permita simular el comportamiento previamente concebido de un agenteinteligente y abordando de esta manera los dos paradigmas previamente descriptos.

Problema a resolver

En el año 2010, se ha incorporado a la cátedra una unidad de LEGO® MINDSTORMS® NXT 2.0 (MINDSTORMS) [1] con el objetivo de que el alumno pueda plasmar en un ambiente real un agente inteligente. De esta manera, al incorporar un programa de agente en un dispositivo MINDSTORMS, se logra que el mismo, se comporte como un agente inteligente en un mundo real y no simulado. La experiencia de los alumnos al trabajar con sensores reales y aspectos físicos reales permite adquirir conocimientos sobre un campo aún no explorado en la enseñanza de inteligencia artificial en nuestra facultad, donde generalmente se trabaja con entornos simulados y donde las condiciones externas están controladas. Esto permite ofrecer una alternativa académica a la enseñanza de agentes inteligentes.

Sin embargo el problema que se plantea para lograr este objetivo de la cátedra es no contar con suficientes unidades MINDSTORMS como para que los alumnos puedan probar sus programas antes de presentar el resultado final a los docentes. En promedio en esta cátedra cursan por año unos 40 alumnos, existiendo aproximadamente unos 15 grupos de alumnos para el desarrollo de trabajos prácticos.

Objetivos y alcance

Ante esta problemática, se considera necesario poder contar con una herramienta de software que a partir de recibir un programa de agente, imite el comportamiento del dispositivo físico controlado por dicho programa. De esta forma, los alumnos podrán probar sus programas de agentes utilizando un simulador del dispositivo físico, y una vez probados y depurados, se podrá ejecutar la versión definitiva en el robot MINDSTORMS. Dado que el robot permite una infinidad de formas posibles de dispositivos físicos (vehículos, humanoides, animaloides, máquinas tales como brazos de robot, etc.) se prevee comenzar en la cátedra con una estructura predefinida de robot, en este caso un vehículo con sensores localizados en distintas posiciones. Se prevee que el tipo y localización del/los sensor/s puede variar.

Bibliografía:

[1] LEGO® MINDSTORMS® NXT 2.0 [ref. de 21 de noviembre 2010]. Disponible en Web: http://mindstorms.lego.com/en-us/Default.aspx

[2] Rethinking Artificial intelligence [ref. de 21 de noviembre 2010]. Disponible en Web: http://web.mit.edu/newsoffice/2009/ai-overview-1207.html

[3] Russell and Norvig (2003). Artificial Intelligence.A modern approach third edition.Prentice Hall.

[4] LabVIEW [ref. de 21 de noviembre 2010]. Disponible en Web: http://en.wikipedia.org/wiki/LabVIEW; http://www.ni.com/labview/whatis/esa/

[5] SHANON, Robert E. – Simulación de Sistemas. México. Editorial Trillas (1988).

[6] Unidad 1 – Metodología del modelado y simulación (2005) – Cátedra Simulación, UTN-FRSF.

[7] LEJOS [ref. de 21 de noviembre 2010]. Disponible en Web: http://lejos.sourceforge.net/nxj.php

[8] Building the Lego Universe on line [ref. de 21 de noviembre 2010]. Disponible en Web: http://spectrum.ieee.org/consumer-electronics/gaming/building-the-lego-universe-online

[9] Roa, Jorge (2007). FAIA: Framework de Agente que utiliza búsqueda para resolver problemas de IA. Jaiio Jornadas Argentinas de Informática e investigación operativa. Simposio EST de trabajos estudiantiles. ISSN 1666-1095.

[10] Pividori, M., Larrateguy, L. (2008) Comparando agentes basados en objetivos y agentes basados en conocimiento. Jaiio Jornadas Argentinas de Informática e investigación operativa. Simposio EST de trabajos estudiantiles. ISSN: 1850-2776

[11] Lencina, V., Ramirez, C.; Semeniuk, N. Diseño e Implementación de un Agente de Software Mediante la Utilización del Framework FAIA. 4to congreso CNEISI 2010. Santa Fe, Argentina del 18 al 20 de agosto.

[12] Roa, J., Gutierrez, M., Stegmayer, G. (2009). FAIA: Framework para la enseñanza de agentes en IA. IE Comunicaciones: Revista Iberoamericana de Informática Educativa, vol. 1, no. 7 (pp. 1-10).

[13] Roa, Pividori, Gutiérrez, Stegmayer (2010).“Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practices for Design” Capítulo VI: “How to develop intelligent agents in an easy way with FAIA” IGI global. Francisco V. CipollaFicarra, AINCI (International Association of Communication Interactive) and ALAIPO (Latin Association of HCI) ISBN: 978-1-61520-763-3. Pag: 120-140

[14] Gutiérrez, M.; Roa, J.; Pividori, M.; Stegmayer, G.; Vega, J. (2010) FAIA: Un framework para el desarrollo de agentes inteligentes. Capítulo de libro. La tecnología educativa al servicio de la educación tecnológica: Experiencias e investigaciones en la UTN. En prensa.

[15] Santana, W. (2007). IDEM-IA: Un Ambiente de Desarrollo Integrado para el Modelado e Implementación de Agentes de Software. CNEISI 2010 Santa Fe.

Capitulo 2.

Robot Lego NXT



LEGO Mindstorms NXT es un kit de piezas lanzado por LEGO en Julio de 2006, este kit es comercializado en distintas versiones variando en cada una en su cantidad y tipo de piezas. El kit adquirido por la catedra de la facultad está formado por: (i) 612 piezas para armar diferentes formas de robot, (ii) un ladrillo inteligente con un microprocesador ARM7 de 32-bits, cuenta con cuatro puertos para conectar sensores y tres puertos donde conectar actuadores, (iii) tres servo motores, (iv) cuatro sensores, un sensor de luz y color, un sensor ultrasónico y dos sensores de tacto, (v) software para programar el ladrillo inteligente. El hardware tiene un lenguaje de programación residente LabVIEW [5] que es propiedad de NationalInstrument. Este lenguaje es útil en versiones simples del robot pero para obtener comportamiento más sofisticado este lenguaje es muy acotado. El ladrillo inteligente puede ejecutar instrucciones desde archivos cargados en su memoria o bien recibir instrucciones a través de conexiónbluetooth.

**El ladrillo inteligente** que es corazón del kit, y a partir del cual se crean las distintas formas y configuraciones, es alimentado por 6 pilas AA, tiene una pantalla LCD monocromática de 100x60 pixeles, y 4 botones para usar la interfaz de usuario.

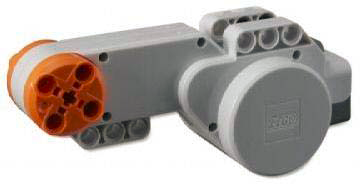


**El sensor de tacto**, tiene dos estados posibles 1 para cuando se encuentra presionado o 0 cuando se encuentra sin presionar, es el sensor mas simple del kit.

**El sensor de ultrasonido**, mide distancias desde el sensor hasta obstáculos que están frente a él. Este sensorpuede configurarse para devolver la medición en centímetros (cm) o pulgadas. Su alcance máximo es de 233 cm con una precisión de +-3 cm. El principio de funcionamiento de este sensor es enviar ondas de ultrasonido y medir el tiempo que toma a

dichas ondas rebotar contra una superficie y regresar hacia el sensor. Este sensor funciona dentro de las especificaciones siempre que las superficies que detecta sean planas.

**El sensor de luz y color,** este sensor detecta el nivel de luz y también incluye un LED para iluminación de objetos. El sensor debe ser calibrado y es capaz de distinguir entre los colores blanco, negro, rojo, verde, amarillo y azul.

**Servo motor**,estos son los actuadores del robot y a partir de los cuales se realiza el movimiento del vehículo.

Con estos elementos es posible implementar un agente que perciba a través de los sensores conectados y que actúe a través de sus actuadores. Este tipo de agente se desenvuelve en un ambiente real.

Capitulo 3.

LeJOS NXJ

Al momento de comenzar con este proyecto la catedra cuenta con el ambiente de desarrollo LeJOS NXJ instalado en el robot en lugar del software grafico NXT-G que viene con el kit NXT de fábrica.

LeJOS NXJ es un ambiente de desarrollo de programación java para las unidades Lego Mindstorms NXT, que contiene una maquina virtual para interpretar código java pre compilado y software adicional para cargar y ejecutar los programas escritos en el lenguaje JAVA.

Para trabajar con LeJOS en el robot, es necesario llevar a cabo una serie de pasos que permiten la compilación, transferencia y ejecución en el robot de los programas. Esta configuración tiene variaciones según el Sistema Operativo (SO) que tenga instalado la computadora que va a ser usada como herramienta complementaria para escribir, compilar, y transferir los programas al robot.

Instalar JAVA en la computadora:

El JDK (Kit de desarrollo de aplicaciones java) es el paquete de software que permite compilar y ejecutar aplicaciones java. LeJOS NXJ ha sido probado con las versionsJDK 1.5, 1.6 y 1.7, y no es compatible con versiones anteriores. Se recomienda el uso de JDK 1.6 o superior.

La descarga JDK se realiza desde el sitio oficial de Oracle [http://www.oracle.com](http://www.oracle.com/).

Instalar el driver Lego NXT USB en la computadora:

Para que la computadora reconozca al robot a través de cable USB, se debe tener instalado un driver USB. Este driver viene junto con el software proporcionado en el kit Lego, y la última versión se encuentra disponible para ser descargada en el sitio oficial [http://mindstorms.lego.com](http://mindstorms.lego.com/).

Existe también la alternativa de transferir programas al robot por bluetooth, para esto se debe contar en la computadora con un modulo bluetoothcapás de comunicarse con otros dispositivos bluetooth.

Instalar leJOS en la computadora:

El paquete leJOS está disponible para su descarga en el sitio oficial de leJOS http://lejos.sourceforge.net. Existen distintas opciones de instalación según el SO.

Instalar leJOS en el Robot NXT:

Para instalar el firmware NXJ en el robot, se necesita encender el robot, conectar el robot a la PC y yejecutar el programa nxjflash o el programa grafico nxjflashg instalados en la computadora en el paso anterior. Este proceso borra todo los archivos del robot e instala leJOS NXJ en el robot.

A partir de este momento ya se está en condiciones de compilar y cargar programas a través de los comandos nxjc nombre\_del\_programa.java para compilar y nxjnombre\_del\_programa para cargar el programa en el robot.

Entorno de desarrollo integrado (IDE):

Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de integrateddevelopmentenvironment), es un programa informático compuesto por un conjunto de herramientas de programación.

El uso IDEs está altamente divulgado entre la comunidad de desarrolladores ya que integra distintas herramientas para la escritura, organización, depuración y ejecución programas escritos en diferentes lenguajes de programación.

En caso elegimos Eclipse, este es un IDE de código abierto ampliamente usado y con soporte para distintos lenguajes de programación, entre los cuales se encuentra JAVA y el cual cuenta con un plug-in para leJOS, el cual luego de ser configurado permite compilar y enviar los programas al robot.

Capitulo 4.

Configuración del robot

Para la realización de este proyecto, se eligió la configuración del robot mostrada en la figura 1. Esta configuración usa un sistema de movimiento diferencial.

Dicha configuración se corresponde con un vehículo explorador, que cuenta con dos ruedas tipo oruga y la ubicación de hasta 4 sensores de los ya descriptos (tacto, luz y ultrasonido) que puede ser modificada de acuerdo a lo que el vehículo requiere para manejarse dentro del ambiente planteado.

Cada rueda tipo oruga es manejada por un motor, los dos motores utilizados se corresponden a los puertos de salida 1 y 2, el primero para la rueda izquierda y el segundo para la rueda derecha del vehículo.

En el caso de los sensores, dependiendo el ejemplo que se esté simulando, se usan 1 o mas sensores. Existen 4 posiciones posibles para agregar un sensor en el vehículo, estas son al frente del vehículo izquierda, central y derecha dichas posiciones se corresponden con los puertos de entrada 1, 2 y 3 respectivamente, y una posición central en la parte trasera del vehículo que se corresponde con el puerto de entrada 4.

Ambientes creados

Para este proyecto final de carrera se han generado los siguientes ambientes, los cuales fueron probados tanto en el simulador como en el robot real, dando un resultado satisfactorio la similitud entre ambos resultados.

Line follower:

Este ambiente es una grilla o espacio abierto con una pista dibujada en su interior de color negro. En la configuración del robot se utiliza el sensor de luz y el objetivo del vehículo es conducirse sobre la pista sin penderla.

Labyrinth:

Este ambiente es un laberinto donde el objetivo de robot es partir desde un punto inicial y sortear el laberinto hasta llegar a la salida del mismo. Para este ejemplo se utilizan el sensor de ultrasonido y un sensor de tacto.

Island:

Este ambiente es una grilla cerrada y tiene una isla en el medio de la misma. El objetivo del robot es moverse dentro de esta grilla sin chocar con ningún obstáculo. El robot usa el sensor de ultrasonido para lograr este objetivo.

Capitulo 5.

Implementación del vehículo explorador

Programas similares:

Al momento de comenzar con este proyecto, se investigo en la web la existencia de programas que cumplan con los requisitos solicitados por la catedra. Si bien es una necesidad extendida sobre todo para distintas catedras donde usan dicho robot para la enseñanza, y existen distintos foros y sitios especializados en robots Lego donde se plantea la necesidad de un simulador, hasta este momento no existen herramientas capaces de simular el comportamiento total de un robot lego en sus distintas configuraciones y utilizando el mismo código java que se usa en robots que corren bajo leJOS.

En esta búsqueda encontramos una librería llamada JGameGrid, esta librería da soporte para la creación de juegos simples en 2 dimensiones dentro de una matriz de pixels. De está forma usando la librería se permite fácilmente instanciar una matriz personalizada y agregar los llamados actores que en nuestro caso son el robot y los obstáculos.Dentro del sitio también existe una librería llamada NxtJLib la cual implementa un simulador del robot Lego nxt teniendo una configuración de un vehículo parecida a la propuesta en este proyecto, y pudiendo agregar y utilizar distintos sensores pero bastante acotada respecto a las funcionalidades de leJOS que implementa, y planteando interfaces de movimiento en lugar de implementar las clases leJOS.

Sistema de movimiento del robot:

El sistema de movimiento que usa el robot para moverse se llama diferencial, este usa dos ruedas para propulsión y control de dirección a partir de la diferencia de velocidad de cada rueda motora.

Los movimientos posibles mediante este sistema de movimientos son, movimiento rectilineo manteniendo las velocidades de los motores igualadas o bien rotaciones alrededor del punto llamado Centro de Rotación Instantáneo (ICR por sus singles en ingles “Instant Centre of Rotation”). El ICR está ubicado sobre el eje imaginario que coincide con el eje de ruedas del robot , y su posición varia en base a la diferencia de las velocidades y sentido de movimiento de ambos motores en el mismo instante.

Esto se puede probar a través de la figura …



La velocidad angular es una medida de la velocidad de rotación. Se define como el ángulo girado por una unidad de tiempo y se designa mediante la letra griega ω (omega). Su unidad en el Sistema Internacional es el radián por segundo (rad/s).

En cada instante, se debe cumplir que la rueda izquierda y derecha sigan un trayecto que se mueve alrededor del ICR a la misma velocidad angular *ω*.

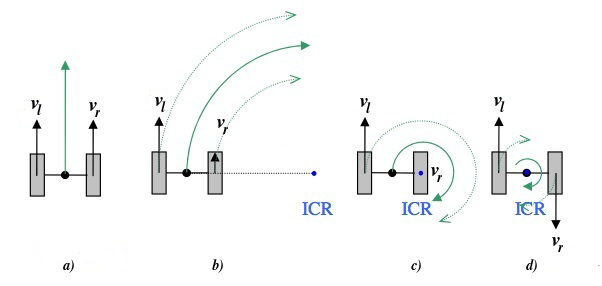
De ésta manera, guiándonos por el esquema de la figura …, podemos escribir:

*ω*(*ρ*+w*/*2) = *vr*

*ω*(*ρ-*w*/*2) = *vl*

donde w es la distancia del eje que une las dos ruedas, vr y vl son las velocidades de las ruedas derecha e izquierda, respectivamente, y ρ(ro)es la distancia desde el ICR al punto medio entre las ruedas. En cualquier instante, se puede resolver para ρ, ω y v:

De aquí se deduce que, si *vl* = *vr* , entonces el radio *ρ* es infinito y el robot se mueve en línea recta. Si *vl* = -v*r* , entonces el radio es cero y el robot rota sobre su eje de rotación vertical. Para otros valores de *vl* y *vr* , el robot no se mueve en línea recta, sino que sigue una trayectoria curva alrededor de un punto a distancia *ρ* del centro del robot, cambiando tanto la posición como la orientación del robot. La figura … ilustra las distintas combinaciones de velocidades y sus resultados en el movimiento del robot.



*Figura … a) vl* =v*r >* 0, b) v*l > vr >* 0, c) *vl >* 0,*vr* = 0, d) *vl* = -*vr*

Cinemática del robot:

La cinemática (del griego κινεω, kineo, movimiento) es una rama de la física que estudia las leyes del movimiento (cambios de posición) de los cuerpos, sin tomar en cuenta las causas (fuerzas) que lo producen y se limita, esencialmente, al estudio de la trayectoria en función del tiempo [wikipedia].

Para poder simular el movimiento del robot se necesita un modelo matemático que muestre como a partir de las entradas, en nuestro caso un punto inicial (x0,y0), una dirección inicial dada por el angulo inicial θ0 y velocidades de cada rueda vr y vl, el robot se mueve por el plano xy.

El modelo matematico que se adapta a nuestro caso es el siguiente:

θ= θ0 + C \*(vr - vl)

x = x0 + ((vr - vl)/2) \* *cos*(θ)

y = y0 + ((vr - vl)/2) \* *sen*(θ)

donde:

C= *radio\_de\_ruedas*/disntacia\_entre\_ruedas

Capitulo 6.

Conclusiones

Fuentes.

<http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT>.