I+D

Ambiente de experimentación remota de robótica móvil

Remote experimentation environment of mobile robotics

Eric Rodríguez[*]
Édgar Altamirano[**]
Arnulfo Catalán[***]

Fecha de recepción: 25 de abril del 2011
Fecha de aceptación: 16 de junio del 2011

Resumen

La enseñanza de diferentes disciplinas técnicas y científicas requiere de la realización de experimentos para consolidar los conceptos teóricos vistos en las aulas. El entorno tecnológico que ofrece Internet, en conjunto con el ámbito educativo, se han combinado en las instituciones educativas para brindarles a los estudiantes nuevas formas de aprender y acceso a infraestructuras remotas para realizar experimentos. En este documento se presenta la propuesta de un ambiente de experimentación remota para mejorar el aprendizaje de materias experimentales, como es el caso de la robótica móvil. Los experimentos que se pueden realizar con esta herramienta consisten en probar diferentes estrategias para realizar el recorrido de un laberinto desde un punto inicial a uno final con la ayuda de los sensores infrarrojos del robot y mediante un algoritmo de control hecho en Java. El sistema le brinda al usuario información del ambiente remoto por medio de imágenes y datos con los que se puede observar los resultados de la tarea que se va a realizar.

Palabras clave: laboratorio virtual, robótica móvil, educación a distancia.

[*] Unidad Académica de Ingeniería, de la Universidad Autónoma de Guerrero Av. Lázaro Cárdenas S/N, Ciudad Universitaria. Chilpancingo, Guerrero, México. Teléfono: 01 (747) 47 2-79-43. Correo electrónico: erodriguez@uagro.mx

[**] Unidad Académica de Matemáticas, de la Universidad Autónoma de Guerrero Av. Lázaro Cárdenas S/N, Ciudad Universitaria. Chilpancingo, Guerrero, México. Teléfono: 01 (747) 47 2-79-43. Correo electrónico: edgar@altamirano.biz.

[***] Unidad Académica de Ingeniería, de la Universidad Autónoma de Guerrero Av. Lázaro Cárdenas S/N, Ciudad Universitaria. Chilpancingo, Guerrero, México. Teléfono: 01 (747) 47 2-79-43. Correo electrónico: arnol26@gmail.com.

Abstract

The teaching of scientific and technical disciplines required for conducting experiments to consolidate the theoretical conceptsseen in the classroom. The technological environment of the Internet in conjunction with Education, have been combined intoeducational institutions to give students new ways of learningand remote access to infrastructure to conduct experiments. This document presents the proposal of a remote experimentation environment to enhance learning experimentalmaterials, as is the case of mobile robotics. Experiments can be performed with this tool are to test different strategies for the path of a maze from a starting point to an end with the help of the infrared sensors of the robot and control algorithm using a Java based. The system gives the user information travéspor remoteenvironment through images and data that you can see the results of the task being performed.

Key words: virtual laboratory, mobile robotics, distance education.

Introducción

Las prácticas de laboratorio constituyen un elemento importante del proceso de formación de profesionales en diferentes programas educativos, en el cual, unidas a las sesiones de introducción de conceptos teóricos y a la resolución de ejercicios, conforman las distintas etapas por las cuales un estudiante adquiere las competencias necesarias para desempeñarse en el campo profesional. En la actualidad, en la parte correspondiente al trabajo experimental, la incursión de las tecnologías de la información y la comunicación (TIC) han revolucionado este proceso de formación en las instituciones educativas, debido a que muchas de ellas están entrando en una etapa en la que buena parte de las actividades propias de la vida estudiantil se desarrollan a distancia accediendo a páginas web.

La estructuración de la información, mediante la transmisión interactiva de audio, video y datos por medio de Internet han conducido al desarrollo de laboratorios virtuales como entornos de experimentación en los cuales los usuarios pueden interactuar sobre una serie de componentes gráficos, cada uno representando un elemento importante del experimento por realizar. Con la creación de estos entornos de formación virtual, es posible cubrir los problemas de experimentación que se presentan en algunas instituciones educativas, lo que les permite afrontar de manera eficiente la formación de nuevos ingenieros e investigadores. Además, los laboratorios virtuales pueden adoptar procesos de enseñanza aprendizaje a distancia a la medida del alumno –a su tiempo, a su interés específico y a su forma de aprender–, adicionalmente, permiten el acceso a cualquier hora y desde cualquier lugar, mediante el uso de una computadora y acceso a Internet. Por esto, la realización de un laboratorio remoto resulta una aportación importante como herramienta educativa.

Tabla 1. Principales ventajas y desventajas de cada tipo de laboratorio

TIPO DE LABORATORIO	VENTAJAS	DESVENTAJAS
Tradicional	Datos reales. Interacción con equipo real. Trabajo colaborativo. Interacción con el instructor	Restricción de tiempo y lugar. Horarios estrictos. Equipo costoso. Requiere supervisión.
Simulado	Ideal para comprender conceptos. No hay restricción de tiempo ni lugar. Es un medio interactivo.	Datos ideales. Falta de colaboración. No hay interacción con un equipo real.
Remoto	Interacción con equipo real. Datos reales. No hay restricción de tiempo ni de lugar. Costo medio.	Presencia virtual. Retardos de tiempo.

Tipos de laboratorios

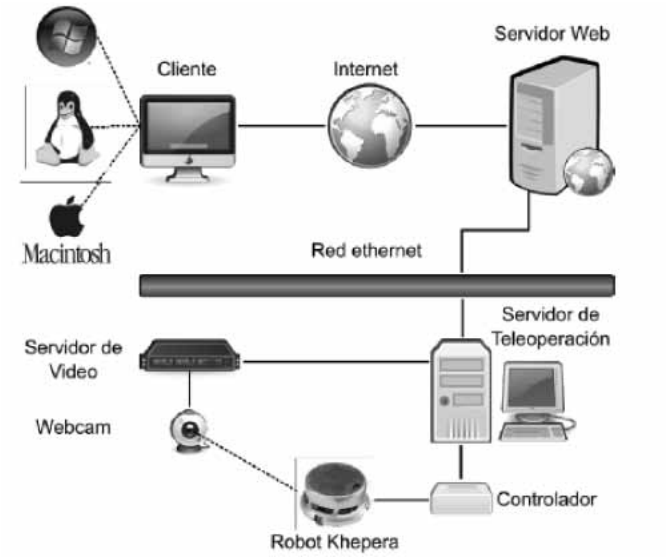
Actualmente, la gran mayoría de las instituciones de educación cuentan con espacios bien definidos para la experimentación de ciertas disciplinas, otras más han aprovechado los recursos tecnológicos que tienen a su alcance para desarrollar otro tipo de laboratorios donde se pueden realizar experimentos, ya sea de manera simulada o remota. En general, existen tres tipos de laboratorios: tradicionales, simulados y remotos. Un laboratorio tradicional ha sido el lugar de trabajo donde los estudiantes y los profesores pueden comprobar mediante la experimentación los conocimientos teóricos adquiridos en clase y así lograr un aprendizaje significativo. La principal ventaja de estos es que el alumno interactúa directamente con el experimento, lo que le permite aplicar las habilidades cognitivas y desarrollar nuevas competencias. Sin embargo, este tipo de laboratorios también presentan ciertos inconvenientes que tienen que ver con la disponibilidad de horario, cantidad de equipo y recurso humano.

Un laboratorio virtual basado en Internet provee acceso a uno o múltiples usuarios para realizar experimentos, ya sea de manera simulada o remota (Moreno, 2001). Un laboratorio virtual simulado es una herramienta de software que recrea el comportamiento de un fenómeno o modelo físico de manera interactiva por medio de la computadora, este tipo de herramientas dependen únicamente de los recursos propios de un servidor como el acceso a bases de datos. Un laboratorio virtual remoto permite operar remotamente

cierto equipo y requiere de un servidor que dé acceso a dichos equipos. La tabla 1 muestra las principales ventajas y desventajas de los tres tipos de laboratorios.

En este documento, se aborda el diseño y desarrollo de un laboratorio remoto de robótica móvil, para su construcción, un modelo de teleoperación ha sido utilizado como se muestra en la figura 1 (Khamis, 2003). Este modelo se basa en un simple protocolo comúnmente usado en computación distribuida "Protocolo de petición/respuesta". Los clientes interactúan con el sistema utilizando un visualizador de Internet para hacer sus peticiones. Estas peticiones son traducidas a simples peticiones HTTP, que son respondidas por el servidor web. El servidor envía estas peticiones convirtiéndolas en peticiones de control de alto nivel que son recibidas por el controlador del robot, el cual, a su vez, las envía al robot como peticiones de bajo nivel para realizar la tarea requerida. La retroalimentación es necesaria para enviar al usuario información sobre el ambiente remoto del robot y las consecuencias de sus comandos.

Figura 1. Arquitectura básica
de un laboratorio virtual de experimentación remota



Fuente: http://www.ehu.es/ikastorratza/3_alea/laboratorios.pdf

Diseño e implementación del laboratorio

En este proyecto de investigación, el laboratorio remoto se construyó bajo una arquitectura genérica: cliente/servidor para controlar el robot usando el protocolo estándar TCP/IP basado en sockets. Para el desarrollo de la aplicación se utilizó Java, como lenguaje de programación, de lo que se obtuvo como resultado un sistema independiente de la plataforma y orientado a objetos. Esto significa que el usuario puede trabajar con cualquier sistema operativo y ser capaz de acceder a la página del laboratorio para interactuar con el robot. La aplicación se compone principalmente de los siguientes módulos:

- Cliente
- Autentificación
- Interfaz de usuario
- Servidor
- Retroalimentación
- Control del robot

El cliente

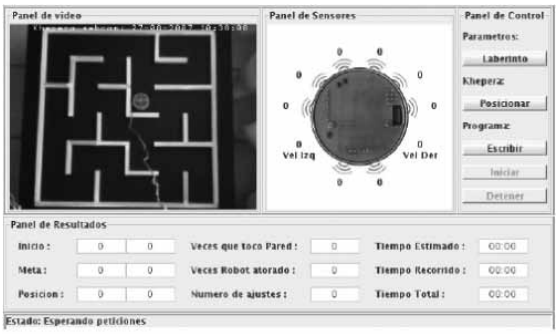
El módulo del cliente se implementó por medio de un Apple de Java,[1] el cual es cargado en una terminal remota mediante un navegador web, esto le permite al usuario interactuar con el robot. Una vez realizada la conexión, el usuario cuenta con un periodo determinado para probar su algoritmo de control y concluir el experimento, esto le da oportunidad a otros usuarios de hacer uso del laboratorio virtual.

Módulo de autentificación

Se decidió utilizar Apache Tomcat para alojar las páginas que dan acceso al laboratorio virtual por lo que esta parte del proyecto se basa más en la configuración que en la programación. Para poder realizar la tarea de autentificación de usuarios válidos y hacer uso del laboratorio virtual fue necesario editar los descriptores XML, tanto del servidor Tomcat, para decidir el método de autentificación que se va a utilizar, como de la aplicación web, para decidir qué recursos serían los protegidos.

Para acceder a la página del laboratorio virtual, se utilizó un mecanismo de autentificación HTTP basada en formularios, implementada por el servidor web, en el que se solicita el nombre de usuario y la contraseña. Si los datos son válidos se pasaría al recurso protegido, si no se mostrará una página de error.

Figura 2. Interfaz de usuario del laboratorio de experimentación remoto



Fuente:http://www.ehu.es/ikastorratza/3_alea/laboratorios.pdf

El panel de usuario

Es la parte del cliente del sistema desarrollado y consta de dieciocho clases. La clase principal es llamada panel usuario que extiende de Applet. En esta clase se hace la instancia de los objetos de los que consta la aplicación. En el método init() del Applet, se añaden el panel de video, el panel de lectura de sensores, el panel de control del robot, el panel de programación, el panel de resultados y el panel de estado como se muestra en la figura 2.

El usuario por medio de un navegador puede acceder a los servicios de la interfaz mediante el protocolo HTTP (protocolo de transferencia de hipertexto) para establecer una conexión con el servidor. El acceso de los usuarios al laboratorio remoto lo hacen por medio de un Applet que provee información gráfica, textual y visual del ambiente remoto en los diferentes paneles que componen la interfaz.

Una parte importante de este proyecto fue proporcionar al usuario retroalimentación visual del ambiente remoto. Para realizar esta

1 Los Applets son aplicaciones Java que se ejecutan dentro de un navegador Web (generalmente, como parte de una página Web) (Eckel, 2002).

tarea, se instaló una webcam sobre el área de trabajo del robot con la que se obtiene una visión global del entorno en el que se moverá el robot, como se muestra en la figura 3.

Para la realización de los experimentos, se utilizó una cámara Quickcam pro for Notebooks de Logitech, ya que está bien soportada en el sistema operativo Linux que se encuentra instalado en el servidor en el que se aloja la aplicación. Los controladores necesarios se pueden obtener de Luc Saillard. Free phillips usb webcam drivers for linux. Las imágenes son adquiridas por la webcam usando video4linux y son recibidas y procesadas por un servidor de video. El servidor de video utilizado es Camserv (J. Travis. Camserv software) que está implementado en lenguaje de programación C y es completamente libre y gratuito. Se desarrolló un applet de Java, como se muestra en la figura 4, para mostrar las imágenes en el panel de video de la interfaz del usuario.

El ambiente del robot Khepera es un laberinto configurable utilizado para probar la inteligencia del robot y funcionalidad del sistema. Las paredes del laberinto son bloques de madera de 2,5 cm de alto por 11,5 cm de largo con lo que se pueden lograr corredores de 10 x 10 cm, espacio suficiente para que el khepera pueda moverse con libertad. Excepto por las paredes externas, todos los bloques internos que forman las paredes son movibles, de forma que se puede cambiar fácilmente el diseño del laberinto.

El robot Khepera cuenta con un anillo de ocho sensores infrarrojos. El panel de sensores muestra la lectura de estos en formato numérico, mientras se ejecuta el programa de control, su valor se actualiza continuamente mientras el robot se mueve dentro del laberinto y es desplegado cerca de su respectiva posición, como se muestra en la figura5. Estos componentes son capaces de detectar un obstáculo dentro de un rango aproximado de 50 mm. Los valores varían en un rango de 0 a 1024 unidades (Team. Khepera User Manual, Version 1.1, 2002) donde el primer valor indica que ningún objeto ha sido detectado y el segundo indica el máximo valor de proximidad a un objeto, esto significa que el valor de los sensores es inversamente proporcional a la distancia con el objeto.
El panel de sensores despliega en la interfaz del cliente lo que el robot percibe de su entorno mientras se ejecuta el programa de control. En este panel también se muestra la velocidad de las ruedas del robot que varía en el rango de 1 a 9. La velocidad está dada en pulsos/10 m que corresponde a 0,8 cm/seg en la mínima velocidad y 7,2 cm/seg en la máxima velocidad. La figura 6 muestra el diagrama de flujo de las lecturas de los sensores de proximidad y de la velocidad del robot.

Figura 3. Vista global del entorno de trabajo

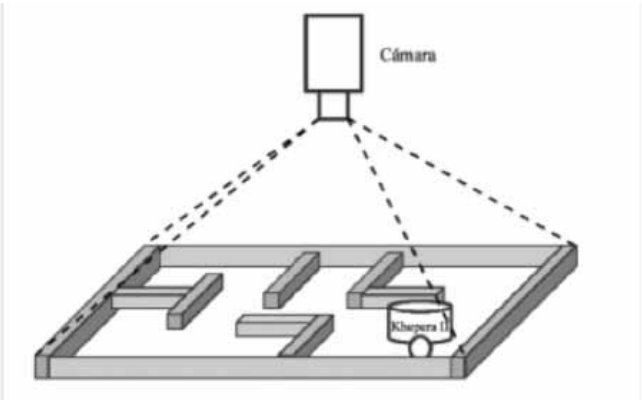
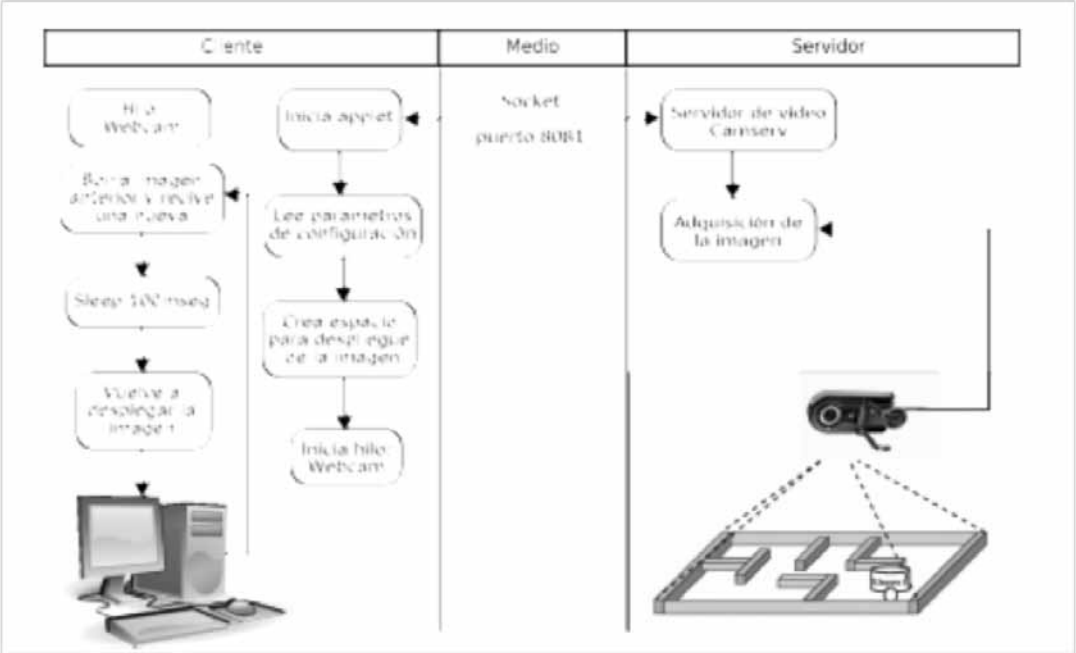


Figura 4. Diagrama de flujo del Applet de video, la imagen es actualizada cada 10 m



El panel de control incluye los botones necesarios para interactuar con la aplicación. Algunos de ellos abren ventanas que le permiten al usuario manipular de manera directa el robot, escribir y enviar un programa para controlar el robot y observar un laberinto virtual, los demás botones le permiten al usuario iniciar o detener el programa en cualquier instante de tiempo.

El botón laberinto del panel de control abre una ventana que muestra gráficamente el diseñó de un laberinto virtual, similar al que se encuentra en el espacio de trabajo remoto del robot, figura 7. En esta ventana se especifican las coordenadas tanto del punto de inicio como de la meta, se calcula la trayectoria óptima que debe seguir el robot entre estos dos puntos y el tiempo estimado para ir de un punto a otro. Los datos del laberinto virtual, diseño y valores son cargados al momento de abrir esta ventana, desde una base de datos que se encuentra al lado del servidor.

El botón de posicionamiento abre una ventana que muestra los botones que le permiten al usuario controlar el robot de manera remota, con la idea de dirigirlo a la posición de inicio que se muestra en la ventana del laberinto virtual. Esto le permite al sistema medir los parámetros de tiempo y control, en comparación con los establecidos en esa ventana. Cada uno de los botones de la ventana de posicionamiento, mostrados en la figura 8, activa un programa del lado del servidor, a fin de enviar los comandos necesarios al robot, para su control, mediante el puerto serial. El programa finaliza automáticamente al concluir la tarea requerida.

La ventana de programación está compuesta por un editor de texto en el cual el usua-

rio puede escribir el conjunto de instrucciones de programación automática, es decir, el algoritmo de control que va a ser enviado y ejecutado por el robot. Se utiliza un área de texto para desplegar los posibles errores que se generen en el proceso de compilación del programa y un botón que tiene la función de enviar el código escrito hacia el servidor. La figura 9 muestra la ventana de programación que incluye el código necesario para que el usuario comience a escribir su programa de control. Una vez que el usuario ha terminado de escribir su programa en el editor de texto, este se envía al servidor por medio del botón "enviar programa", en el que se guarda en un directorio predeterminado con el nombre de "Usuario.java" y se inicia el proceso de compilación.

Figura 5. Lectura de los sensores del robot
Fuente: elaboración propia.

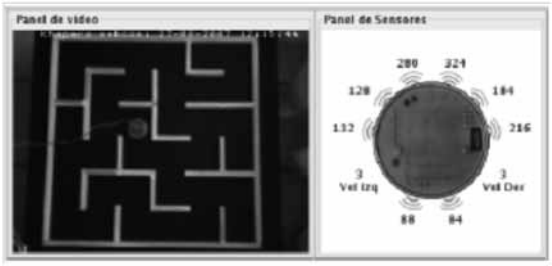
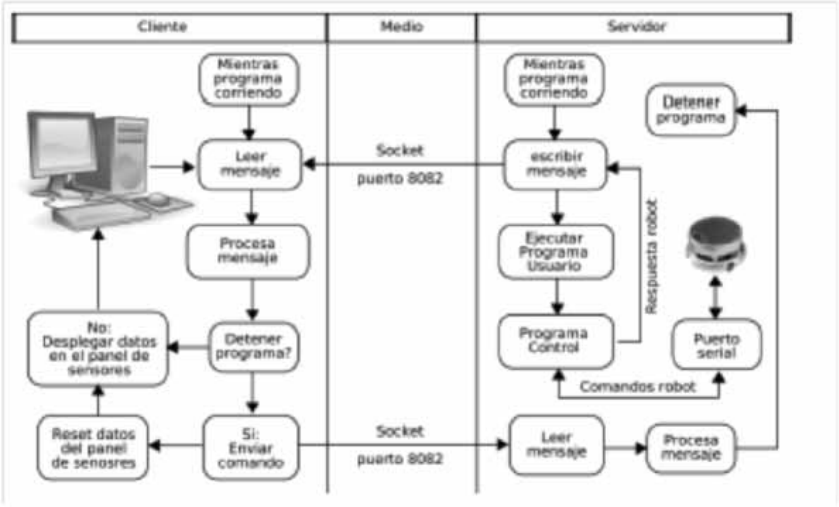


Figura 6. Diagrama de flujo para la lectura de los sensores y actuadores del robot



Fuente: elaboración propia.

Si el programa está libre de errores, la ventana de programación se cierra y se habilita el botón "iniciar programa "en la interfaz del usuario, de lo contrario, el servidor captura y envía al cliente, mediante el socket el posible error de sintaxis que es desplegado en el área de texto que se encuentra en la parte inferior de la ventana de programación, lo que indica el tipo de error y el número de línea en el que se está generando ese error. La figura 10

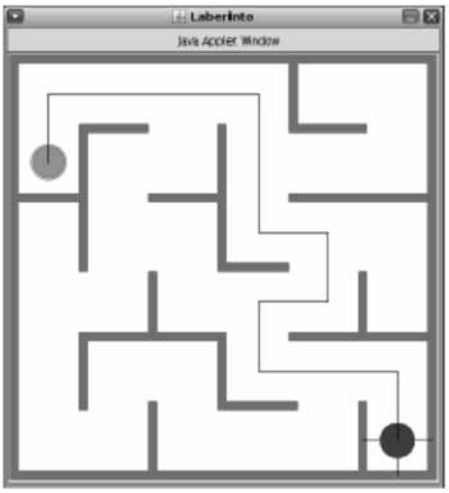
muestra el diagrama de flujo del proceso de envío y la compilación del programa escrito por el usuario.

En el panel de resultados, se muestran los parámetros iniciales definidos en el panel de dibujo y los resultados obtenidos al ejecutarse el programa de control escrito por el usuario. Los primeros se refieren a las coordenadas de la posición inicial a partir del cual se van a medir los parámetros de tiempo y control, las coordenadas de la meta y el tiempo estimado para ir de un punto a otro. Los demás resultados son calculados mientras el robot ejecuta los comandos enviados por el programa del usuario mientras se mueve en su ambiente de trabajo.

Los valores inicio y meta están relacionados con las coordenadas del punto de partida del robot dentro del laberinto y las coordenadas del punto que debe encontrar mientras se está ejecutando el algoritmo de control. El robot cumple con el objetivo del experimento cuando los sensores infrarrojos detectan una fuente de iluminación ubicada en la misma posición que el punto meta. Los dos primeros valores de la segunda columna, las veces que el robot tocó pared y el número de veces que el robot se atoró se determinan mientras el algoritmo de control mueve el robot dentro de su entorno, el primero de ellos estima el número de veces que el robot choca contra cualquiera de las paredes del laberinto basándose en la información obtenida por los sensores de proximidad. Se considera que el robot tocó una de las paredes cuando cualquiera de sus sensores alcanza el máximo valor de proximidad a un objeto.

Para estimar el segundo valor de la columna, la cantidad de veces que el robot se atoró, se tomó en cuenta el tiempo que el robot se detiene por más de 5 segundos, mientras el

Figura 7. Ventana del laberinto virtual



Fuente: elaboración propia

Figura 8. Botones de posicionamiento del robot

programa de control continuó ejecutándose; si esto sucede, una rutina del sistema mueve el robot para continuar con el algoritmo de control. El tercer valor, número de ajustes, se refiere al número de correcciones que el usuario ha hecho a su programa, este valor se incrementa cada vez que el programa de control del usuario es detenido y reenviado al servidor.

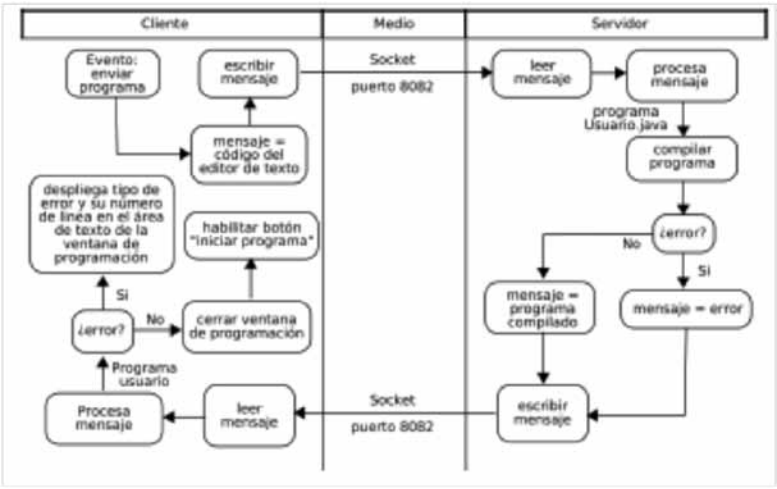
En la tercera columna se muestra el valor del tiempo estimado que el robot debe tardar en realizar la tarea de ir de la posición

de inicio a la meta a una velocidad promedio. El segundo valor, tiempo de recorrido, mide el tiempo que el robot está haciendo para encontrar la meta mientras se ejecuta el programa de control. El tercer valor se refiere al tiempo total acumulado que requirió el usuario en realizar el experimento. La medición de estos parámetros se relacionan con los conceptos de sensores, control y programación de robots, mediante el uso de técnicas de inteligencia artificial que un usuario del laboratorio virtual debe conocer para que su aprendizaje sea significativo.

Figura 9. Interfaz de la ventana de programación

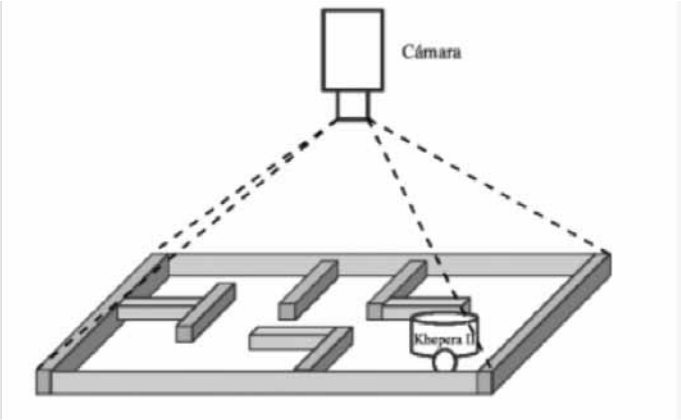
Fuente: elaboración propia

Figura 10. Diagrama de flujo del proceso de compilación del programa de control



Fuente: elaboración propia.

Figura 11. Panel de resultados del experimento



Fuente: elaboración propia.

El servidor

El servidor se implementa como una aplicación que administra la conexión con el cliente mediante el uso de TCP (Transport Control Protocol) basada en Sockets y está formada por dieciséis clases, entre las cuales se encuentra la clase servidor que está corriendo en espera de una conexión. Las peticiones hechas por los usuarios son recibidas, procesadas y ejecutadas por esta aplicación, enviando la respuesta correspondiente al cliente para desplegarla en la interfaz de usuario.

Del lado del servidor se compila y se ejecuta el programa Controlador enviado por el usuario. El código le indica al robot, mediante el puerto serial de la computadora, cómo moverse dentro de su entorno basándose en la información que recibe de sus sensores. Para la realización de esta etapa, se desarrolló un módulo de control que es utilizado para ejecutar el programa realizado por el usuario y la comunicación con el robot.
Para la realización del módulo de control, se utilizaron dos hilos, el primero de ellos es un hilo de control encargado de actualizar constantemente la información de los sensores del robot y enviar los comandos de operación a los actuadores por medio del puerto serial. El segundo hilo es responsable de enviar las lecturas de los sensores y actuadores al cliente por medio del socket. La clase Proceso es la responsable de iniciar y finalizar los hilos de procesamiento involucrados y de terminar el ciclo de vida de los objetos que componen el sistema.

El robot Khepera, utilizado en este proyecto, puede trabajar de dos formas diferentes, de manera autónoma (el programa de control se baja al procesador del robot) o por medio de una conexión al puerto serial de una computadora que es donde se ejecuta el programa de control. Para la implementación de este proyecto se utilizó la segunda opción.
En este modo el robot Khepera utiliza un protocolo de comunicación, basado en comandos y respuestas en código ASCII. Los comandos enviados al robot inician con una letra mayúscula, seguida por el parámetro requerido y finaliza con un retorno de carro representado por el carácter {. La respuesta del robot inicia con la letra minúscula correspondiente al comando enviado, seguida de los datos por transmitir y un retorno de ca-

rro. Por ejemplo, el comando para leer los sensores de proximidad del robot Khepera es el siguiente: N{. La respuesta del robot es regresada como una cadena de caracteres que debe ser procesada para determinar los valores numéricos de cada uno de los sensores de proximidad: n,0,59, 1023, 1023, 78, 0, 0 ,0{.
Dado que este tipo de comunicación entre el robot y la computadora implica abrir una conexión con el puerto serial de la computadora, enviar el comando, recibir una cadena de caracteres como respuesta y procesar esta respuesta implica varias líneas de código, se desarrolló una clase llamada Control Class que facilita la tarea de programación a los usuarios.

En esta clase se encapsulan los métodos para controlar el robot de una manera sencilla. Estos métodos están relacionados con la obtención del valor de proximidad de los sensores infrarrojos, la especificación de la velocidad de las ruedas del robot, así como el control de giros.

El constructor de la clase Control crea una instancia de la clase COM Serial que provee una interface con el puerto serial de la computadora, por medio de la cual el robot se comunica con el algoritmo de control. Para el desarrollo de este proyecto, se utilizó el puerto de comunicación serial estándar RS232 a una velocidad de 19200 baudios, 8 bits de datos, 1 bit de inicio, 2 bits de alto, sin paridad, así como se especifica en el manual de usuario del robot Khepera (Team. Khepera User Manual, Version 1.1, 2002). Para esto, se utilizó el API (Application Programming Interface) de comunicación RXTX (Trent Jarvi, s.f.), que es una librería que provee comunicación con el puerto serial y paralelo para el paquete de desarrollo de Java (JDK por sus siglas en inglés).

Conclusiones

Este laboratorio de experimentación remota virtual de robótica móvil se diseñó como una herramienta educativa complementaria a la materia de robótica móvil para la realización de experimentos con un robot real, relacionados con aspectos de sensores, control y programación. Este laboratorio les facilita a los usuarios interactuar con un robot que se encuentra localizado en un ambiente remoto enfrentándolos a diversas complejidades que involucra un ambiente no simulado.

El sistema le brinda al usuario información del ambiente remoto mediante imágenes y datos con los que se pueda observar los resultados de la tarea por realizar. Los experimentos que se pueden realizar con esta herramienta consisten en probar diferentes estrategias de control para salir de un laberinto con la ayuda de los sensores infrarrojos del robot y mediante un algoritmo de control hecho en Java.

La ventaja de utilizar Java como lenguaje de programación es que el laboratorio virtual puede ser ejecutado en cualquier plataforma para la que exista una implementación de la máquina virtual Java. En la actualidad, existe multitud de implementaciones de dicha máquina virtual, lo que nos permite que la aplicación pueda ser considerada como multiplataforma, permitiéndose la ejecución, por ejemplo, tanto en plataformas Windows como en entornos Linux. Sin embargo, cabe aclarar que los usuarios del laboratorio deben tener conocimientos sólidos de este lenguaje para la programación de los algoritmos de control, lo que implica una limitante.
El hecho de que el programa de control enviado por el usuario se ejecute en el servidor permite que el tamaño del algoritmo de control no esté restringido por el tamaño de la memoria y del procesador instalado en el

robot. Además, el hecho de que se utilice un lenguaje de alto nivel permite escribir y depurar con mayor facilidad el programa de control.

Las principales aportaciones de este trabajo son el desarrollo de un ambiente de operación remota para un robot Khepera que debe resolver el problema de un laberinto, la medición de los parámetros de tiempo y control de la tarea realizada que proporcionan una idea del desempeño del algoritmo, la obtención de la trayectoria óptima y el tiempo de recorrido entre el punto de inicio y la meta, la flexibilidad para trabajar con diferentes escenarios y la facilidad para incrementar de manera muy simple el número y el tipo de experimentos a realizar.

Trabajo futuro

Existen diferentes líneas de trabajo de gran interés para la continuación de la presente investigación, entre las que cabe mencionar las siguientes:

Realizar pruebas con los usuarios finales del sistema, utilizando diferentes algoritmos de control para comprobar la aceptación y la funcionalidad de la herramienta.

Integrar el laboratorio remoto a un curso de robótica para realizar experimentación con un robot móvil que ayude a lograr un entendimiento más profundo de la materia.

Integración del laboratorio virtual remoto a un tutor inteligente para proporcionarles a los estudiantes retroalimentación inmediata sobre el desempeño del experimento, mediante la evaluación y diagnóstico de este tutor, contribuyendo así al mejoramiento del proceso de aprendizaje del estudiante.

Proporcionar retroalimentación visual de forma gráfica, de manera que se puedan mover al mismo tiempo el robot real en el ambiente remoto y el robot simulado en el laberinto virtual.

Integrar otros parámetros de evaluación que ayuden a mejorar la evaluación del desempeño de experimentos como el número de veces que el robot ha pasado por un mismo lugar y la estimación de la trayectoria seguida por el robot con la idea de compararla con la trayectoria óptima.

Incluir en la página del laboratorio remoto un enlace de ayudas que le permitan al estudiante mejorar la eficiencia y el desempeño del experimento.

Integración de un simulador del laboratorio virtual remoto en el que los estudiantes puedan probar sus algoritmos de control antes de que sean enviados al robot real con la idea de hacer un uso eficiente de la herramienta.

Este laboratorio remoto puede servir como base para el desarrollo de diferentes experimentos de robótica móvil, equipando al robot con accesorios y diferente tipo de sensores que le ayuden a obtener un mejor conocimiento de su entorno desarrollando así tareas más complejas.

Referencias

Eckel, B. (2002). *Piensa en Java* (2ª ed.). Ciudad: Prentice Hall.

Khamis, A. M. (2003). *Interacción remota con robots móviles basada en Internet*. PhD thesis, Universidad Carlos III de Madrid.

K-Team (2002, mar.). Khepera User Manual, Version 1.1. Preverenges Switzerland: KTeam, S.A., Ch de Vassuet, CP 111 1028.

Moreno, J. (2001). *Un nuevo enfoque metodológico para la enseñanza a distancia de asignaturas experimentales: Análisis, diseño y desarrollo de un laboratorio virtual remoto*

para el estudio de la automática a través de internet. PhD thesis, Universidad Nacional de Educación a Distancia (UNED).

Saillard, L. *Free phillips usb webcam drivers for linux*. Recuperado en mayo del 2006 de: http://www.saillard.org/linux/pwc.

Travis, J. *Camserv software*. Recuperado en mayo del 2006 de: http://cserv.sourceforge.net.

Trent Jarvi (2007). Api de comunicación para el puerto serial hecho en Java. Recuperado en mayo del 2006 de: http://users.frii.com/jarvi/rxtx/license.html.