

SIMULADOR DE UN ROBOT MÓVIL COMO COMPLEMENTO DE UN KIT DE ROBOT VELOCISTA.

A. VILLAR y J. PASTOR.

*Departamento de Electrónica. Escuela Politécnica Superior. Universidad de Alcalá.
Alcalá de Henares (Madrid) España.*

El presente artículo presenta el diseño de un simulador de un robot móvil autónomo pensado para complementar un Kit de Robot Velocista [2]. El simulador emula el comportamiento de la tarjeta de control del kit, simulando todas sus librerías y permitiendo que el programa principal que el alumno tiene que hacer para controlar el robot sea el mismo en el simulador que en el sistema final.

Este sistema permite que los alumnos prueben los algoritmos sin necesidad de utilizar el sistema real y puede ser utilizado como complemento a un curso de iniciación a la programación.

1. Introducción

Como se comenta en [1] [2] el diseño de microrrobots es una actividad muy interesante y atractiva que puede ser utilizada por estudiantes universitarios de enseñanzas técnicas para poner en práctica sus conocimientos, y para introducir en la tecnología a estudiantes no universitarios y aficionados.

Además, el diseño de microrrobots tiene como ventaja fundamental el estar relacionado con muchas áreas de conocimiento diferentes: electrónica, sistemas sensoriales, algoritmia, etc. Sin embargo, esto se puede convertir en un inconveniente cuando se plantea la organización de un taller de iniciación al diseño de microrrobots dirigido a personas sin conocimientos técnicos en estas áreas. Lo cual, lleva a la necesidad de poder dividir de un modo claro, los diferentes problemas que un usuario se puede encontrar a la hora de iniciarse en este tipo de disciplinas. De este modo nace el simulador de robot velocistas.

El objetivo principal de la creación de un simulador de robot es facilitar la enseñanza de este tipo de disciplinas a los estudiantes primerizos, y facilitar pruebas más complejas a estudiantes avanzados, dada la facilidad de reajustes del sistema.

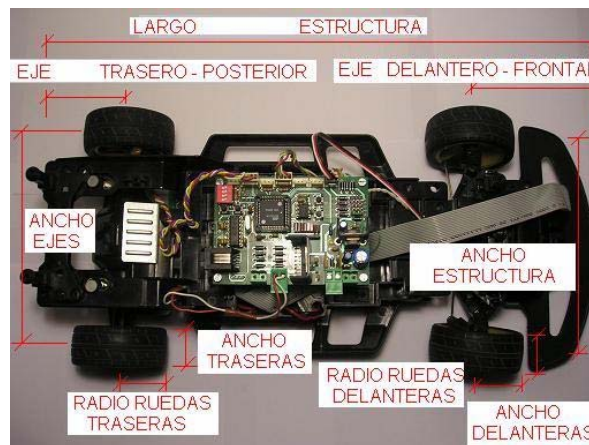


Figura 1. Robot velocista.

El Kit de robot velocista [2] trataba de resolver la problemática de impartir un curso de robótica, en poco tiempo y a personas de toda índole. Para lo cual se pensó en un robot de fácil entendimiento (robot velocista), desarrollado de un modo sencillo (uso de la estructura de un coche teledirigido) (Fig.1).

2. Objetivos

El objetivo inicial es realizar un simulador para complementar el Kit de robot velocista [2] para taller de iniciación a la robótica de modo que sea más sencillo el aprendizaje para los alumnos así como la enseñanza de la parte de programación y algoritmia.

Las principales cualidades buscadas a la hora de crear el simulador fueron su compatibilidad con el Kit de robot velocista [2] de modo que los usuarios una vez creado sus algoritmos y validados en simulación, pudieran probarlos en los robot.

Por lo cual, el simulador desarrollado debía tener las siguientes características principales:

- Simulación de diferentes tipos y modelos de robots velocistas.
- Simulación del comportamiento de la tarjeta ALCAI2C [1].
- Herramienta de manejo sencillo para los usuarios que se inicien en las competiciones de robot.
- Herramienta configurable mediante un entorno gráfico de dos modos: cargando una configuración mediante un archivo o introduciendo manualmente todos los parámetros en el sistema.

3. El simulador

La solución adoptada fue desarrollar un programa en LabWindows [4], entorno de instrumentación virtual desarrollado por National Instruments [5] programable en C, el cual tiene las siguientes funciones:

- Simular las funciones de control del hardware del kit.
- Simular la dinámica de los motores de tracción y dirección.
- Permitir introducir una pista de robot velocista genérica.
- Simular la lectura de los sensores dependiendo de dónde se sitúen en la pista.
- Configurar las medidas del robot y la posición de sensores y motores.

La principal cualidad de este sistema es simular las propiedades más características de los robots velocistas y de la tarjeta de control usada en el kit, ALCAI2C [1]. Es decir, se pueden simular diferentes estructuras de robot velocistas, con diferentes cualidades mecánicas, con diferentes posiciones y número de sensores, pero todas ellas deben estar gobernadas bajo el control de la tarjeta ALCAI2C [1] o bajo el control de un hardware con una librería de funciones compatible con las existentes tanto para el simulador, como para la tarjeta ALCAI2C [1].

Las principales cualidades del simulador desde el punto de vista de usuario son:

- Simulación de dos estructuras mecánicas (diferencial y Ackerman o turismo).
- Posibilidad de seleccionar diferentes medidas del robot así como diferentes posiciones de las ruedas motrices y/o directrices y de los sensores.
- Posibilidad de simular diferentes pistas o circuitos.
- Simulación de lectura de los sensores dependiendo de dónde se sitúen en la pista.
- Simulación de las ecuaciones cinemáticas del robot.
- Simulación de la dinámica mediante la respuesta de los motores.

- Uso de la misma estructura de la librerías de funciones que la tarjeta ALCAI2C [1], por lo cual, el usuario debe crear un archivo control.c, que puede ser ejecutado en el simulador y en la tarjeta hardware.

Cabe destacar que todas estas cualidades son transparentes al usuario, de modo que sólo es necesario comprender qué significan, para hacer un uso eficiente del simulador, sin necesidad de entender cómo el simulador consigue todos los puntos descritos con anterioridad.

Para facilitar el entendimiento de los datos que se piden al usuario, se ha asociado una ayuda a los botones más importantes, la cual se despliega al pulsar sobre los marcadores con el botón izquierdo del ratón.

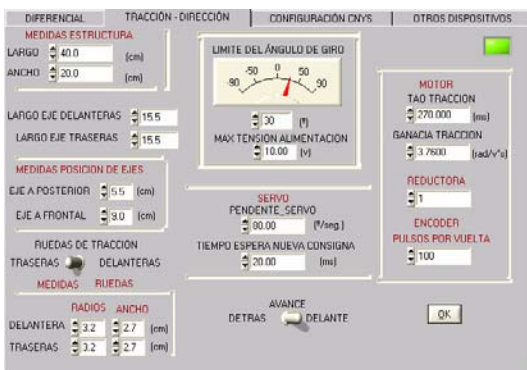


Figura 2. Panel de configuración del robot tipo turismo.

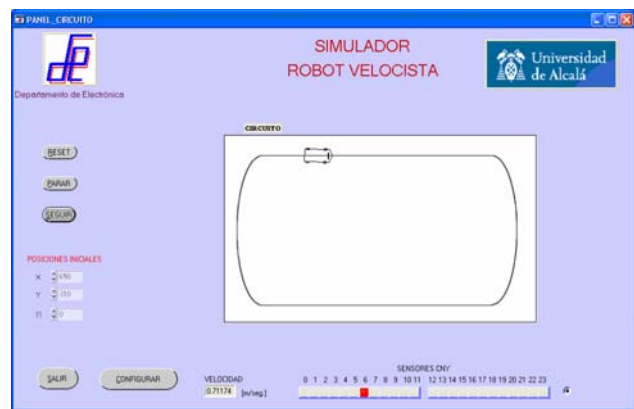


Figura 3. Pantalla del simulador.

Otra cualidad a resaltar son los dos modos de introducción de datos al sistema, pudiendo configurar manualmente cada una de las variables indicadas en unos paneles gráficos (Fig.2) o introducir mediante un archivo de configuración. Este último modo de trabajo, hace que el sistema sea muy versátil debido a que una vez conseguido parametrizar correctamente el robot velocista, se puede guardar todos esos datos en un archivo, de modo que pueda ser cargado sin necesidad de introducirlos de nuevo cada vez que se desea realizar una nueva simulación. En apartados posteriores se comentan los parámetros que se puede configurar.

4. Descripción general del funcionamiento interno del simulador

El simulador está dividido en tres partes que interactúan de la manera que se observa a continuación.(Fig. 4) :

- Simulación del comportamiento del robot.
- Simulación del control del sistema.
- Representación en la pantalla.

Cada una de las partes del sistema tiene su propio temporizador, esto es debido a que están divididas y son partes independientes. Relacionadas entre si mediante variables o funciones de las librerías, motores, sensores, encoders...

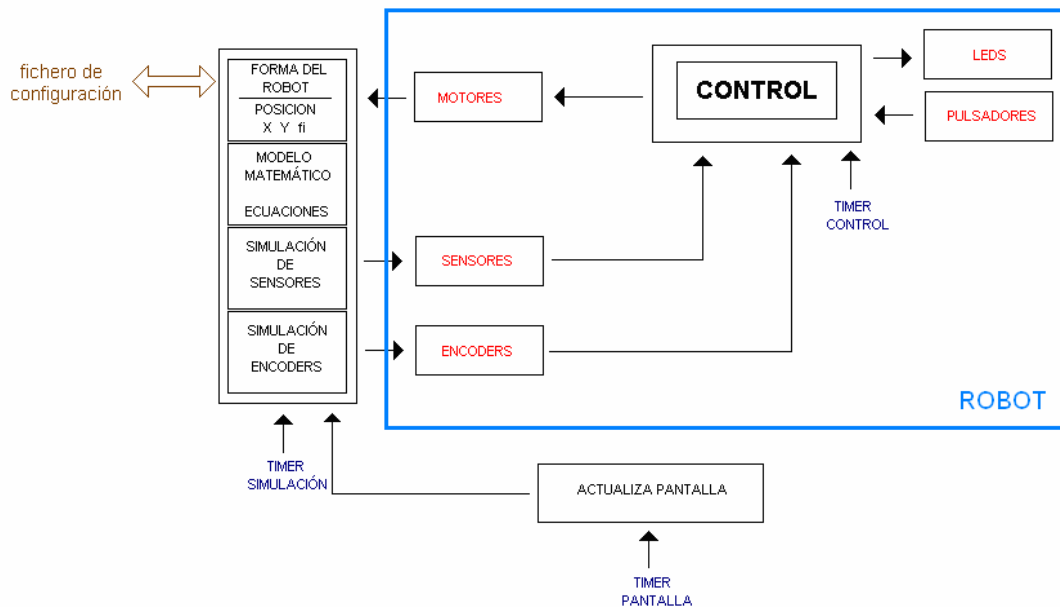


Figura 4. Esquema del método de simulación.

El funcionamiento del simulador es el siguiente:

- Módulo de configuración de los parámetros del simulador.
- Módulo de configuración del conexionado de los diferentes elementos del robot simulado con la placa simulada.
- Núcleo de simulación formado por tres temporizadores (Fig. 4):
 - Temporizador de la simulación del robot con funciones de simulación de la cinemática y dinámica del robot. (Fig. 5).
 - Temporizador del control del sistema con funciones de lectura de sensores, cambio de variables como la tensión de alimentación media que llega a los motores (produciendo cambios de velocidad en estos). (Fig. 5).
 - Temporizador del refresco de la pantalla encargado de la representación de modo gráfico el camino seguido por el robot. (Fig. 5).

4.1. Simulación del movimiento del robot

La simulación del robot se realiza mediante la simulación de su comportamiento cinemático y dinámico. Con las ecuaciones del comportamiento cinemático se calculan los cambios de posición respecto a un punto de origen. Estas ecuaciones indican cómo es el movimiento del vehículo simulado, es decir, cómo se comporta el sistema permitiendo simular la trayectoria del vehículo.

Para un vehículo con tracción diferencial los parámetros de la ecuación son:

- Posición inicial en el plano (X, Y, FI) del robot (posiciones respecto al origen de referencia).
- Velocidad lineal de las ruedas.
- Tiempo transcurrido.

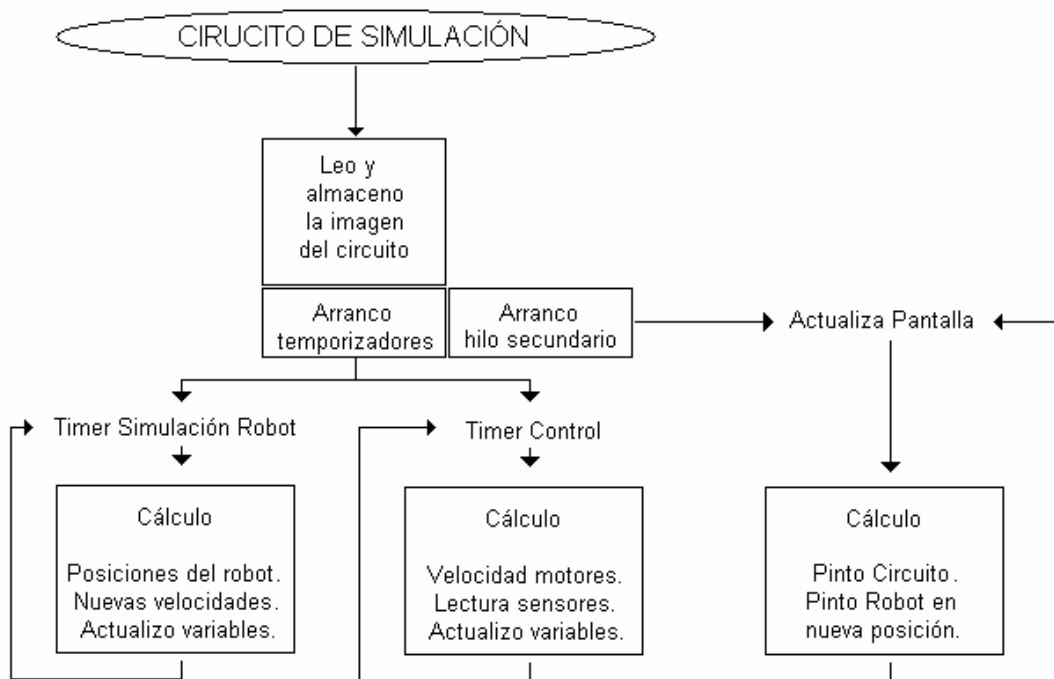


Figura 5. Esquema de funcionamiento del entorno de simulación.

Para un vehículo con tracción y dirección los parámetros de la ecuación son:

- Posición inicial en el plano (X, Y, FI) del robot.
- Velocidad lineal del motor de tracción.
- Ángulo de giro de las ruedas de dirección.
- Tiempo transcurrido.

Con las ecuaciones del comportamiento dinámico lo que se calcula es cómo se producen los cambios de trayectoria, ya que el comportamiento dinámico sólo indica la trayectoria seguida sin tener en cuenta, por ejemplo, la inercia del vehículo. Para simular el comportamiento dinámico del vehículo y dado que hace uso de motores eléctricos para su movilidad, se usa las ecuaciones dinámicas que definen a los motores en las que viene descrita la inercia de los estos. Mediante estas ecuaciones de la dinámica se realiza el cálculo de cómo se producen los cambios de trayectorias del movimiento de los vehículos.

Estas ecuaciones se ven restringidas debido a la necesidad de discretización, ya que sería imposible el cálculo de las mismas para todos los puntos de la trayectoria que sigue un vehículo. Para discretizar el sistema se hace uso de un temporizador para conseguir el cálculo de los diferentes puntos que generan una trayectoria entre dos puntos inicio y final de la misma. El valor del temporizador es el encargado de hacer que la discretización sea más o menos fiel a la realidad.

El cálculo de este parámetro, *Tiempo del Simulador* se ve afectado por la constante de tiempo menor de los motores y el espacio máximo recorrido en un período del Tiempo de simulador.

El problema del segundo parámetro que afecta al *Tiempo de Simulador* es que depende de cómo sea la trayectoria que se le exige al vehículo, este es un parámetro difícil de medir por lo que se omite su uso. Por lo cual se realiza el cálculo con el primer parámetro que afecta al *Tiempo de Simulador*. Para hacer

que el sistema discreto y el real se aproximen lo más posible se calculan 10 puntos para la constante de tiempo menor del sistema. Siendo entonces el *Tiempo de Ssimulador* 10 veces menor que la constante de tiempo menor de los motores.

Para evitar el problema de desear simular más rápido de lo que se puede, existe un parámetro también configurable que trata sobre la *relación entre el tiempo real* (lo que tardaría un robot real en realizar la operación como la mandada) y *el tiempo simulado* (lo que tarda el robot simulado en realizar la misma operación).

4.2. Simulación del programa de control

El control del robot se realiza mediante la simulación de la placa ALCAI2C [1] la cual está formada por un microcontrolador que controla los sensores, motores, encoders y demás dispositivos que tiene el robot simulado. El control se realiza cada período de control o *Tiempo de Control*, por lo que se trata de un control discreto de una planta seudo continua.

En ninguno de los casos el *Tiempo de Control* será menor que el Tiempo de Simulación dado que esto haría que las variables de posición no hubiesen cambiado y daría lugar a que el control del sistema fuese erróneo. Para realizar el control el usuario dispone de un conjunto de funciones que son equivalentes a las que se usan para la tarjeta ALCAI2C [1]. De modo que el archivo usado para hacer pruebas en el simulador pueda ser trasladado al robot sin necesidad de realizar cambios.

4.3. Monitorización de los cambios y actualización de la pantalla

La actualización de la pantalla se lleva a cabo mediante un hilo de ejecución externa, ya que es una de las partes que más tiempo necesitan en ejecución, siendo ejecutada con menor periodicidad.

La actualización de la pantalla es un sistema independiente al resto de temporizadores, sólo es necesaria en el caso de que se desee observar el comportamiento del sistema por pantalla. Debido a su independencia se hace posible el uso de hilos de ejecución en paralelo, ya que el momento de entrar a ejecutarse no afecta al resultado del movimiento.

5. Simulación de la cinemática y dinámica de un robot

Uno de los principales soportes de un simulador es que su comportamiento sea lo más parecido a la realidad. Debido a este hecho, para poder desarrollar un simulador se debe apoyar en diferentes ecuaciones que expliquen el comportamiento del sistema a simular.

El programa desarrollado puede simular básicamente dos tipos de robot:

- Tracción diferencial (tanque)
- Tracción y dirección (coche).

En este apartado nos centraremos en el robot de tipo turismo, dado que es del cual tenemos prototipo actualmente.

5.1. Robot de tracción y dirección

Este tipo de vehículos se caracteriza por tener dos motores, uno que imprime la velocidad lineal con la que se realizan los movimientos y el segundo que indica en que dirección se va aplicando dicha velocidad.

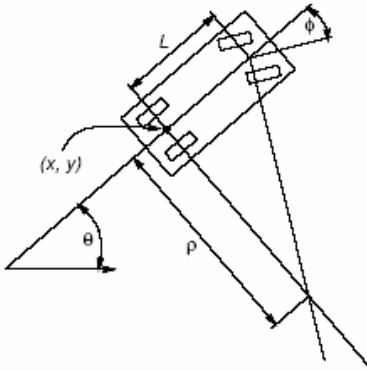
Comportamiento cinemático

Las ecuaciones que rigen el comportamiento de este tipo de vehículo son (Ec. 1) (Ec.2) (Ec.3)[7] y están relacionadas los parámetros presentados en la Fig. 6.

$$\Delta x = (r \cdot 2 \cdot \pi) \cdot T_m \cdot w \cdot \cos(\theta \cdot \pi / 180) \quad (1)$$

$$\Delta y = (r \cdot 2 \cdot \pi) \cdot T_m \cdot w \cdot \sin(\theta \cdot \pi / 180) \quad (2)$$

$$\Delta \theta = (180 \cdot r \cdot \pi \cdot 2 \cdot w / L) \cdot T_m \cdot \tan(\Phi \cdot \pi / 180) \quad (3)$$



- x → Posición x en un plano de coordenadas XY [mm].
- y → Posición y en un plano de coordenadas XY [mm].
- θ → Ángulo que se encuentra girado el vehículo en el plano XY [°].
- r → Radio de las ruedas de tracción [mm].
- w → Velocidad angular de las ruedas de tracción [rad/seg.].
- L → Distancia de separación entre las ruedas delanteras y traseras [mm].
- T_m → Constante de tiempo. (Tiempo de simulación del movimiento) [seg].

Figura 6. Movimiento tracción y dirección.

Comportamiento dinámico

Para simular este tipo de comportamiento se ha realizado mediante la utilización de las ecuaciones de comportamiento dinámico de los motores Ec.4 y Ec.5, ya que estos son de fácil estudio.

Las ecuaciones que caracterizan un motor de continua en tiempo discreto, según las aproximaciones ya desolladas en el apartado anterior en la discretización del movimiento del robot son:

$$\frac{W(s)}{V(s)} = \frac{A'}{1 + \tau' \cdot s} \xrightarrow{\text{DISCRETIZAR}} \frac{W(z)}{V(z)} = \frac{A_{DIS} \cdot (1 + Z^{-1})}{1 + \tau_{DIS} \cdot Z^{-1}} \quad (4)$$

$$W[n] = A_{DIS} \cdot V[n] + A_{DIS} \cdot V[n-1] - \tau_{DIS} \cdot W[n-1] \quad (5)$$

Siendo los parámetros de Ec. 4 y Ec.5.

- $W[n]$ → Velocidad angular en ese instante.
- $W[n-1]$ → Velocidad angular en el instante anterior.
- $V[n]$ → Tensión que se aplica en ese instante al motor.
- $V[n-1]$ → Tensión que se aplica en el instante anterior al motor.
- A_{DIS} → Ganancia discreta directa.
- τ_{DIS} → Tiempo de respuesta.

Cabe destacar que existen dos aproximaciones realizadas en esta ecuación discreta del sistema. Una de ellas es debida a la propia discretización y el método utilizado (transformación bilineal), la segunda es debida a que un motor de continua no se comporta como un sistema de primer orden, sino como uno de segundo por lo que se comete un segundo error al decir que un motor se comporta como la ecuación indicada inicialmente.

6. Parámetros configurables en el simulador

Los parámetros configurables en el simulador son:

- Pista del velocista.
- Tipo de vehículo a simular.(Fig. 7) (Fig.8).
- Dirección de avance del vehículo.
- Forma de la estructura. (Fig. 7) (Fig. 8).
- Características electromecánicas de los motores.
- Posición y tamaño de los sensores CNY70 [6].
- Relación temporal entre los diferentes temporizadores.

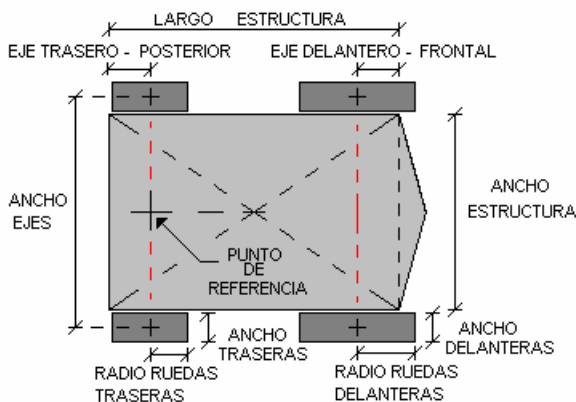


Figura 7. Estructura tipo turismo

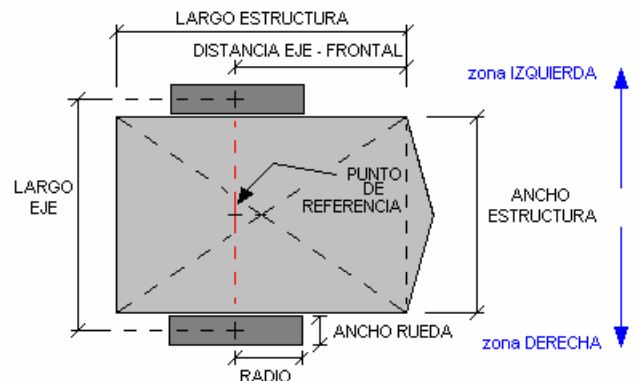


Figura 8. Estructura tipo diferencial.

7. Resultados

Las pruebas llevadas a cabo con el simulador demostraron tener un comportamiento muy similar al del robot real a bajas velocidades encontrándose más diferencias a velocidades medias debido a los factores que no están teniendo en cuenta en la simulación como es el derrape de las ruedas, el efecto de la fuerza centrífuga en el robot y la variación del coeficiente de rozamiento de las ruedas de dirección con la velocidad, la suposición de linealidad de los motores, etc. En las Fig.9 y Fig.10 se puede observar el comportamiento del simulador comparado con el del robot real en una respuesta a un escalón.

En cuanto a la utilización del simulador, si bien en el momento de escribir el artículo la última versión del simulador no se había usado en ningún curso, sí se utilizó una versión preliminar del simulador en un curso de iniciación a la robótica en alumnos de segundo Bachillerato. El simulador sirvió para que los alumnos pusieran en práctica los fundamentos de programación en C recién aprendidos sin necesidad de tener montados los robots reales totalmente funcionales. El simulador también sirvió para que los alumnos depuraran sus propios programas de control en un entorno profesional de programación potente y sencillo de manejar como es LabWindows CVI [4].

Los alumnos se llevaron a casa el simulador y dedicaron su tiempo libre a mejorar los algoritmos fuera del laboratorio.

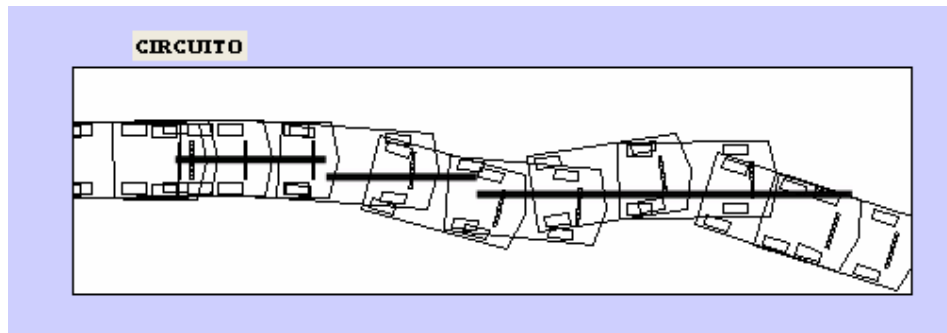


Figura 9. Simulación de la respuesta del vehículo a una pista en escalón

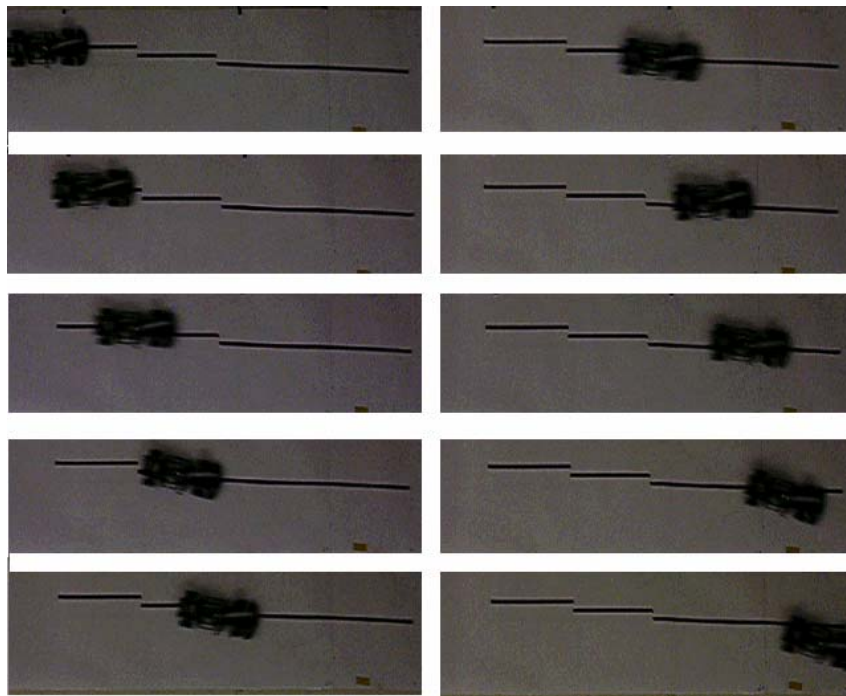


Figura 10. Respuesta real del vehículo a una pista en escalón.

8. Conclusiones

El simulador diseñado sirve de complemento al Kit de robot velocista presentado en [2] pero también se puede utilizar para iniciar a estudiantes en la programación en C haciéndoles que programen el movimiento de un robot y el apagado y encendido de leds.

Disponer de un simulador permite que los alumnos puedan trabajar en casa, fuera del laboratorio, sin necesidad de utilizar un robot real para poner a punto los algoritmos de control. Una vez probados estos algoritmos deben ser validados en la plataforma real y adaptados a las condiciones reales de funcionamiento que el simulador no contempla.

Referencias

- [1] J. Pastor Mendoza, S. Cobreces Álvarez, M. García Pérez, F. Méndez Rebollo, D. Peinador García-Aranda, D. Pizarro Pérez, P. Yagüe Molinero “*KIT de microrrobot velocista para talleres de introducción a los microrrobots*”. CD-ROM de Actas del I Seminario y Concurso Nacional de Microrrobots HISPABOT 2003(ISBN 84-8138-560-3), Servicio de Publicaciones de la Universidad de Alcalá, Abril de 2003.
- [2] J. Pastor Mendoza, D. Alonso García, J.M. Gómez Sánchez, E. Moraleja García y J. Baliñas Santos. *TARJETA ALCAI2C. Sistema de control de un microrrobot basado en el microcontrolador 89C51I2C. Manual de referencia V2.0.* (2004)
- [3] A. Villar y J. Pastor. *Entorno de simulación y control de un robot velocista. TFC.* Escuela Politécnica Superior. Universidad de Alcalá(2005).
- [4] LabWindows. CVI 7.0. <http://sine.ni.com/nips/cds/view/p/lang/en/nid/11104>
- [5] National Instruments <http://www.ni.com/>
- [6] Sensores de infrarojos CNY70 <http://www.datasheetarchive.com/>
- [7] Ecuaciones del movimiento de un robot con tracción diferencial y tracción dirección <http://gamma.cs.unc.edu/COMP290-58/PAPERS/Lavalle-Planning.pdf>