

# Aprende fundamentos de control construyendo un siguelíneas

I. Alvarado

J.A. Borja

F. Salas

D. Muñoz de la Peña

Dpto. Ingeniería de Sistemas y Automática, Universidad de Sevilla

Avd. de los Descubrimientos, s/n,

Sevilla 41092, ialvarado@us.es, jaborja@us.es, salas@us.es, dmunoz@us.es,

## Resumen

*En este artículo se presenta un robot siguelíneas basado en Arduino de bajo coste que se ha usado como proyecto en la asignatura Control Automático de segundo curso del Grado de Ingeniería de Telecomunicaciones de la Universidad de Sevilla. La idea es que los alumnos puedan aprender electrónica, programación, tratamiento de señales y control por medio de la construcción y control de este robot móvil tanto en las clases de docencia práctica en el laboratorio de la asignatura, como en sus horas de trabajo fuera del horario lectivo.*

**Palabras clave:** Educación, control, robótica móvil.

## 1. Introducción

La enseñanza del control no es, en general, una tarea sencilla. Realizar el control de cualquier sistema real conlleva aplicar conocimientos de distintas disciplinas. Hay que aplicar conocimientos de electrónica, de tratamiento y acondicionamiento de señales, de programación, de control. El control, a su vez, puede requerir el desarrollo de un modelo, la identificación del mismo, también puede requerir el aprendizaje del uso de herramientas de simulación, de programación y de control propiamente dicho.

Por este motivo, para que el alumno se concentre en el estudio de la dinámica del sistema controlado a menudo se simplifican los sistemas; se le da el sistema casi completo al que solo le falta la parte a estudiar o directamente trabaja en simulación. El problema de este enfoque es que, aunque el alumno aprende a controlar un sistema, cuando tiene que hacer el control de un sistema real, no sabe como atacar el problema.

Una pedagogía popular para abordar este problema es el aprendizaje basado en proyectos (ABP), [7] [5]. Esta metodología plantea un conjunto de cuestiones como punto de partida. Mediante la realización de un proyecto se van dan respuestas a dichas cuestiones dando proporcionando al alumno el conjunto de conocimientos requeridos.

El aprendizaje basado en proyecto involucra a los estudiantes en la exploración de preguntas importantes y significativas a través de un proceso de investigación y colaboración. Los estudiantes hacen preguntas, predicciones, diseñan soluciones y aprenden a recopilar y analizar datos, utilizar tecnología, fabricar productos y compartir ideas.

Uno de los problemas que enfrenta PBL en el campo de la automática es la necesidad de disponer de un sistema para analizar y controlar. Una opción es el uso de laboratorios virtuales, ver [3] y sus referencias. Sin embargo, los experimentos físicos tienen un valor pedagógico por sí mismos porque el alumno tiene que resolver numerosos problemas que no están presentes en los entornos virtuales. Otra posible solución son las prácticas de laboratorio, pero como ya comentamos, suelen ser sistemas simplificados donde el alumno solo resuelve el parte del sistema de control.

Hoy en día el precio actual de los componentes electrónicos brinda a los educadores la oportunidad de enseñar control mediante la construcción de plataformas de bajo costo. Por ejemplo, en [4] se presenta un robot autoequilibrado de dos ruedas, en [8] se muestra un ejemplo de una bola y una viga y en [6] se construye un levitador magnético para enseñar a controlar, [2] donde se presenta un robot móvil autoequilibrado.

El coste de algunos de estos dispositivos es tan bajo, que abre la posibilidad de que cada alumno pueda adquirir su propio equipo para trabajar fuera del horario lectivo y fuera del laboratorio, es decir, que cada alumno pueda tener su laboratorio en su propia casa.

Este artículo presenta los primeros resultados de un proyecto desarrollado en la Universidad de Sevilla con el objetivo de diseñar un robot siguelíneas de bajo coste que puede ser construido y controlado por los estudiantes de Control Automático en sus propias casas de forma independiente, siguiendo un enfoque PBL, aprendiendo electrónica, programación, modelado (tipo caja negra), control y procesamiento de señales. La figura 1 muestra el robot resultante.

Un robot siguelíneas es un robot que intenta se-

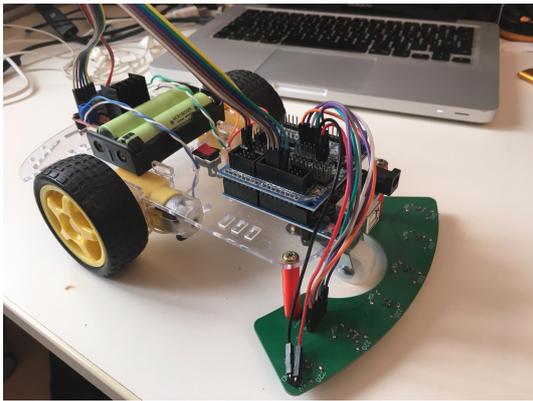


Figura 1: Fotografía del proyecto resultante

guir una línea que describe un camino. La idea es mantener el ángulo entre la línea y el eje del robot igual a cero durante el movimiento, aplicando a los motores, que mueven las ruedas, el voltaje para conseguir dicho objetivo.

Este dispositivo ofrece una serie de retos tecnológicos muy interesantes, además de poder expresar un objetivo de diseño muy directo y claro, ser capaz de diseñar el robot más rápido para un circuito determinado. Esta propiedad es importante, ya que permite que el alumno tenga un resultado claro de su trabajo, e incluso ofrece la oportunidad de realizar una competición entre los diferentes alumnos.

Este artículo se organizará de la siguiente manera; primero, se presenta el prototipo de hardware, a continuación, se presenta un esquema de control diseñado por estudiantes de la Universidad de Sevilla junto con unos resultados obtenidos. Y finalmente el artículo muestra algunas conclusiones y posibles mejoras.

## 2. Hardware

El robot ha sido diseñado de forma que pueda ser construido por alumnos incluyendo las conexiones entre las diferentes partes del mismo. Con objeto de reducir el precio del robot al mínimo se ha comprado un kit compuesto por el chasis, los motores, sus ruedas y el driver de los mismos, un Arduino Uno, un Arduino Uno Sensor Shield, un porta pilas, un interruptor de encendido y una rueda loca que se puede encontrar en internet a un precio muy reducido. A este kit se le ha añadido una placa, diseñada en el departamento, con un array de sensores infrarrojos para detectar la posición de la línea. El precio total del conjunto se puede conseguir entre 20€ y 60€ dependiendo del fabricante (OSOYO, ELEGOO, CEBEKIT entre otros). A continuación se presentan, por separado, las distintas partes del mismo:

### 2.1. Chasis

Placa de metacrilato de 2mm de espesor multi-propósito. En la figura 2 Puede verse el chasis. A media altura a la derecha pueden verse también los soportes de los motores en metacrilato, la rueda loca y el porta-pilas. El kit también incluye los tornillos, aunque no aparezcan en las figuras.



Figura 2: Chasis

### 2.2. Motores

Motores de continua con reductora. En el kit también proporciona las ruedas para los mismos. En la figura 3, pueden verse los motores y las ruedas del kit.



Figura 3: Motores y ruedas

Las únicas soldaduras que este proyecto requiere son los cables que conectan los motores con los drivers, en el lado de los motores. El driver dispone de terminales con tornillos, mediante los cuales se conectan los motores. El resto de las conexiones se pueden hacer mediante cables DuPont con terminales hembra-hembra.

### 2.3. Driver L298N

En la figura 4, puede verse el driver. Los jumpers que en la figura aparecen como *Motor A,B activo* los vamos a quitar y los conectaremos a dos salidas digitales PWM del Arduino (son las que llevan el símbolo ~). En la placa del driver vienen denotados con las etiquetas *ENA* y *ENB* (enable A, enable B, son los pines más cercanos al borde

de la placa). Mediante estas dos señales controlaremos el módulo de la velocidad de los motores.

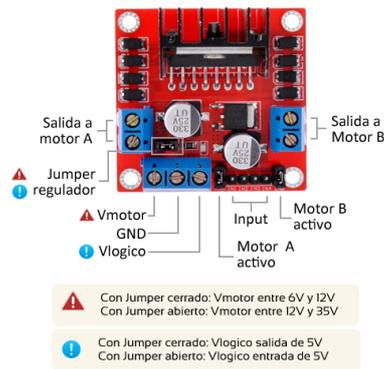


Figura 4: Pines L298N

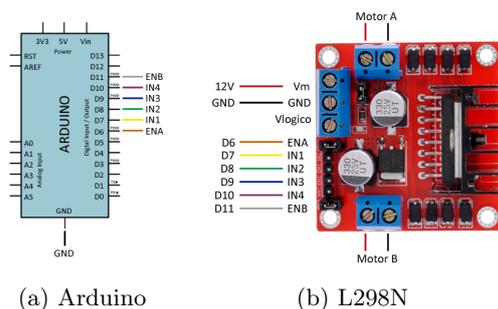
Los 4 pines etiquetados en la imagen como *input* controlarán la dirección de giro de los motores, el freno eléctrico de los mismos o que queden libres. En la placa del driver están etiquetados como *IN1*, *IN2* para controlar el motor A y *IN3* e *IN4* para controlar el motor B de la siguiente manera (véase el cuadro 1).

	Adelante	Atrás	Freno
IN1 (IN3)	HIGH	LOW	LOW
IN2 (IN4)	LOW	HIGH	LOW

Cuadro 1: Configuración dirección de giro de los motores.

Si ponemos el *jumper regulador* podemos alimentar nuestro Arduino a 5V mediante el pin *Vlogico*

En la figura 5 puede verse la conexión con el Arduino:



(a) Arduino (b) L298N  
Figura 5: Conexión L298N - Arduino

**2.4. EL sensor CNY70**

El sensor CNY70, esta compuesto por un diodo led y un foto transistor; es un sensor analógico y nos da un valor de voltaje proporcional a la luz

rebotada. La utilización de un sensor de infrarrojos puede ser de gran utilidad en proyectos en los que haya que hacer distinción entre superficies de color blanco y negro.

El sensor se sitúa a unos milímetros de la superficie a detectar, mientras tanto el diodo interno emite una señal de infrarrojos que absorberá la superficie en el caso de que sea negra o bien rebotará en el caso en el que la superficie sea blanca. Si la señal de infrarrojos rebota, activará la base del fototransistor entrando éste en conducción.

Se montará primero un único sensor para probar como funciona. Realizamos a continuación el montaje de la figura 6 y conectamos la fecha etiquetada como *ARDUINO* a cualquier entrada analógica del Arduino, por ejemplo a la *A0*.

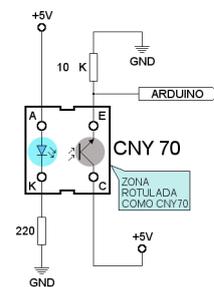


Figura 6: Circuito básico para usar el CNY70

La idea es encontrar un umbral por el cual la lectura del CNY70 se considerará TRUE o 1 cuando encuentre línea y por debajo FALSE o 0 en caso contrario

**2.5. Placa con sensores CNY70**

Con la idea de ser capaces de medir el ángulo que forman la línea y el robot siguelíneas, se crea el siguiente circuito (véase figura 7).

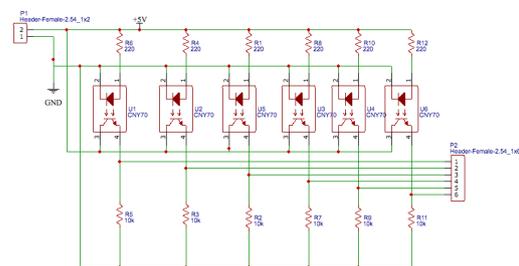


Figura 7: Circuito placa para montar 6 sensores CNY70

Los sensores *cny70* se colocan de modo y forma que estén a  $-25^\circ$ ,  $-15^\circ$ ,  $-5^\circ$ ,  $+5^\circ$ ,  $+15^\circ$  y a  $+25^\circ$  respecto al eje de giro del robot. Véase la parte sombreada en azul de la figura 8. En rojo puede verse el diseño de la placa de metracilato. Se con-

sidera el eje de giro el punto medio de la línea que une los centros de las dos ruedas motrices.

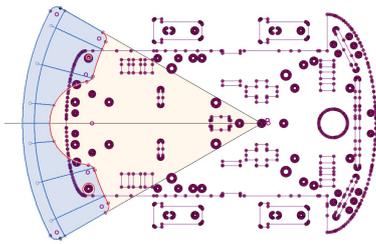


Figura 8: Localización de los sensores CNY70

En la figura 9 puede verse el diseño de la PCB y la placa, ya montada, puede verse en la figura 10

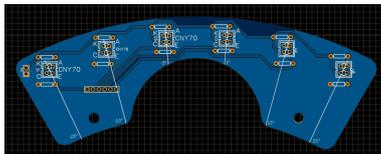


Figura 9: PCB placa sensores cny70

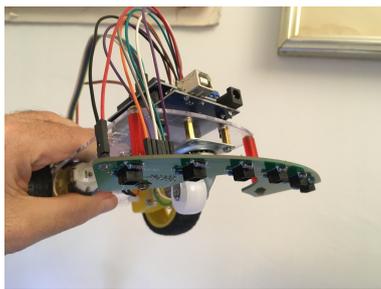


Figura 10: Placa montada

Cada sensor nos proporcionará un 1 si detecta línea y 0 en caso contrario. Almacenaremos la información proporcionada por los sensores ordenados de izquierda a derecha en un vector de enteros *vec\_cny70*. Sea *ang* un entero que contendrá la medida del ángulo y *n* almacenará el número de sensores que valen 1. El ángulo se puede medir mediante el siguiente código:

```

ang=0;
n=0;
for ( i=0; i<6; i++)
{
    ang+=vec_cny70 [ i ] * 10 * ( i + 1 );
    n+=vec_cny70 [ i ];
}
ang=ang/n;
ang=ang - 35;
    
```

Al final de este código *ang* tendrá un valor que irá de  $-25^\circ$  a  $+25^\circ$  en incrementos de  $5^\circ$ . Quedará

pendiente programar que pasa si ningún sensor no detecta línea u otras situaciones anómalas.

El conexionado de esta placa al Arduino es sencillo puesto que la salida de cada uno de los sensores CNY70 va conectado a una entrada analógica del Arduino y los dos pines de alimentación van a 5V y a GND del mismo.

### 2.6. Arduino Uno sensor Shield

El *Arduino sensor shield* viene con el kit del robot. Nos va a proporcionar la posibilidad de hacer todo el circuito del robot sin necesidad de hacer ninguna soldadura (exceptuando los cables de los motores). En la figura 11 se puede ver la descripción de los pines del mismo.

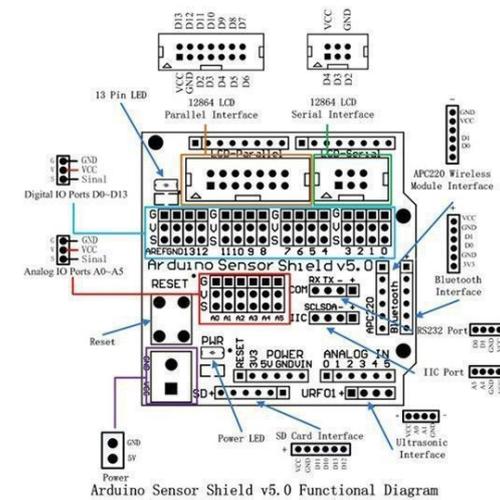


Figura 11: Arduino Sensor Shield V5

### 2.7. Arduino Uno

Arduino Uno es un microcontrolador basado en un ATmega328P. Tiene 14 pines digitales (6 de ellos PWM) y 6 entradas analógicas de 10bits. La velocidad del microcontrolador es  $16MHz$ . La placa contiene toda la electrónica adicional para poder usar los recursos del microcontrolador.

En este proyecto usaremos esta placa porque es la que nos proporciona el kit y porque es muy robusta ante errores en el conexionado, lo que la hace muy apropiada para un alumno que no tiene experiencia trabajando con este tipo de dispositivos.

Vamos a usar el Arduino para leer los 6 sensores CNY70, decidir si están o no detectando línea y construir la medida del ángulo. Por otro lado controlaremos la velocidad y sentido de giro de los motores por medio de dos señales PWM para el módulo y dos pines digitales, por motor, para el sentido. Por último controlaremos el ángulo a

medida que el robot se mueve mediante un PID discretizado que intentará mantener el mismo en un valor próximo a cero para que el robot pueda seguir la línea.

### 2.8. Modulo bluetooth HC-06

Si se quieren sacar datos o monitorizar el ensayo, se puede conectar un módulo bluetooth a los conectores dispuestos para ello en el *Arduino Sensor Shield*.

### 3. Software

El esquema de control es el típico problema de regulación en el que el objetivo de control es mantener el ángulo entre el vehículo y la línea igual a cero a pesar de las perturbaciones (que van a venir por los cambios de dirección en la línea a seguir). En la figura 12 vemos los bloques a programar:

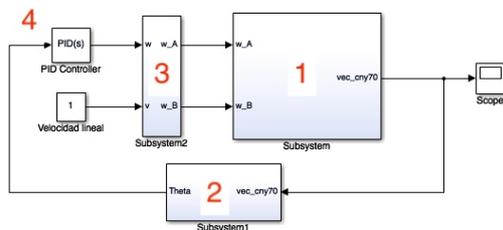


Figura 12: Esquema de control

El bloque 1, recibe como entrada  $\omega_A$  y  $\omega_B$  que son las señales que van a controlar la velocidad de cada motor. Serán dos enteros con signo que irán de -255 a 255 (ya que los controladores de las señales PWM del Arduino son de 8 bits). El módulo de estas señales controlará la señal PWM que se aplican los motores y el signo los pines de dirección. Asimismo este lo que se encarga de leer los 6 sensores CNY70 y los almacenará en un vector de unos y ceros que proporciona al bloque 2.

El bloque 2 construirá el ángulo  $\theta$  (ángulo entre la línea a seguir y el robot) a partir de las medidas del vector como se describe en la sección 2.5. Puede añadirse código adicional para tener en cuenta cuando no se detecta una línea o cuando se llega a un cruce.

El bloque 3 recibe la velocidad angular  $\omega$  del PID y la velocidad lineal  $v$ . Esta segunda magnitud puede ser ajustada como un parámetro más de diseño puesto que cuanto más rápido va el siguelíneas más difícil es sintonizar el controlador. A partir de estas dos magnitudes decidirá las consignas para los dos motores por separado sin más que hacer

$$\omega_A = v + \omega$$

$$\omega_B = v - \omega$$

Se puede tener en cuenta también las posibles saturaciones de las señales de control descritas en el bloque 1.

El bloque 4 es el controlador PID que recibiendo el ángulo entre robot y línea a seguir decidirá la velocidad angular  $\omega$  que ha de adquirir el robot para que el ángulo  $\theta$  sea lo más próxima a 0 .

### 4. Resultados

A continuación se presentan los resultados obtenidos por el siguelíneas siguiendo el circuito exterior de la figura 13

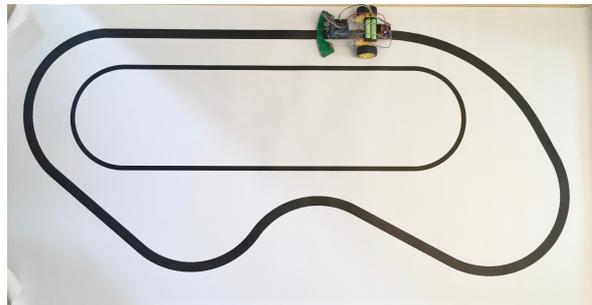


Figura 13: Circuito

Para poder mostrar los resultados se le ha añadido al robot siguelíneas un modulo bluetooth con objeto de mandar el ángulo  $\theta$  y la señal de control  $\omega$  durante un ciclo del mismo. En la figura 14 se pueden ver los resultados. La velocidad lineal en este ensayo es la máxima admisible. Tarda en completar el circuito unos 8 segundos.

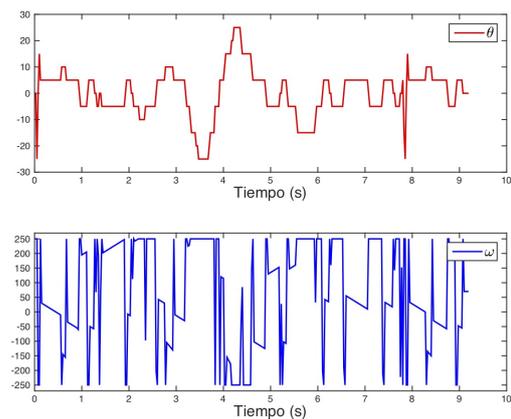


Figura 14: Ensayo

El control utilizado es un PID con un tiempo de muestreo de 25ms. Es un control muy agresivo con un término derivativo muy grande razón por la cual satura mucho.

## 5. Resultados docentes

El año 2020-21 se ha usado este prototipo como experiencia piloto voluntaria para 4 grupos de 3 alumnos de la asignatura Control Automático del grado de Ingeniería en Telecomunicaciones.

La experiencia ha sido positiva, consideran un buen complemento a la asignatura de control. Asimismo han hecho una serie de comentarios o críticas constructivas. La más relevante es que la realización del proyecto consume mucho tiempo por lo que habrá que tenerlo en cuenta en el currículo de la asignatura. Se han hecho 4 ejercicios previos a la realización del robot:

- 1 - Toma de contacto con Arduino.
- 2 - Pulsadores, puerto serie y funciones.
- 3 - Sensores y actuadores del siguelíneas.
- 4 - Control de un ventilador de un PC [1]. Con esta práctica los alumnos aprenden a programar un PID en tiempo discreto, garantizando el tiempo de muestreo mediante interrupciones temporales. Además también tienen que programar un filtro de media para poder generar la medida de la velocidad de giro a partir de la señal que proporciona el encoder integrado en el propio ventilador.

Los alumnos han hecho hincapié en que necesitan más formación previa con Arduino y una última sesión para montar el robot y orientar como programar un PID para el seguimiento de la línea.

Este año que entra van a hacer este proyecto en grupos de 3, todos los alumnos de la asignatura, y se va a realizar una evaluación competitiva con objeto de incentivar un ajuste fino de los parámetros del PID

### Agradecimientos

Los autores de este artículo desean darle las gracias a *Luis Llamas* por todos sus tutoriales sobre Arduino, algunos de los cuales han sido usados para crear este prototipo

### English summary

#### Line following robot as a control subject project

#### Abstract

*This article presents a low-cost Arduino-based line-following robot that has been used as a project in the automatic control subject of the Telecommunications Engineering degree. The idea is that students can learn electronics, programming, signal processing and control through the construction and control of this mobile robot.*

**Keywords:** Education, control, mobile robotics.

## Referencias

- [1] I. Alvarado, J.A. Borja, R. Haes, D. Muñoz de la Peña (2021). Control de velocidad de un ventilador de una CPU. Jornadas de Automática, Castellón, España Septiembre 1-3.
- [2] C. Gonzalez, I. Alvarado, D. Muñoz La Peña (2017). Low cost two-wheels self-balancing robot for control education. 20th IFAC World Congress, Toulouse, France July 9-14.
- [3] Goodwin, G.C., Mediol, A.M., Sher, W., Vlacic, L.B. and Welsh, J.S., (2011). Emulation-based virtual laboratories: a low-cost alternative to physical experiments in control engineering education. IEEE Transactions on Education, 54(1), pp.48-55.
- [4] Hau-Shiue, J. and Kai-Yew, L. (2013). Design and control of a two-wheel self-balancing robot using the arduino microcontroller board. 10th IEEE International Conference on Control and Automation (ICCA), 634-639.
- [5] M. Lehmann , P. Christensen , X. Du & M Thrane (2008) Problem-oriented and project-based learning (POPBL) as an innovative learning strategy for sustainable development in engineering
- [6] Lilienkamp, K.A. and Lundberg, K. (2004). Low-cost magnetic levitation project kits for teaching feedback system design. Proceedings of the American Control Conference.
- [7] Perrenet, J.C., Bouhuijs, P.A.J. and Smits, J.G.M.M., (2000) The suitability of problem-based learning for engineering education: theory and practice. Teaching in higher education, 5(3), pp.345-358.
- [8] Rashied, Z., Hamees, M., Hassan, M.U., Hameed, S., and Khatri, N.A. (2016). Real time implementation of Robust PID controller for stabilization of Ball Balancing Beam. International Journal of Conceptions on Information Technology and Computing. 6-9.



© 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).