

# Estrategias de solución para la prueba del Rastreador, Madrid-Bot 2009

Salustiano Nieva  
Juan Antonio Breña Moral



CONSEJERÍA DE EDUCACIÓN  
**Comunidad de Madrid**

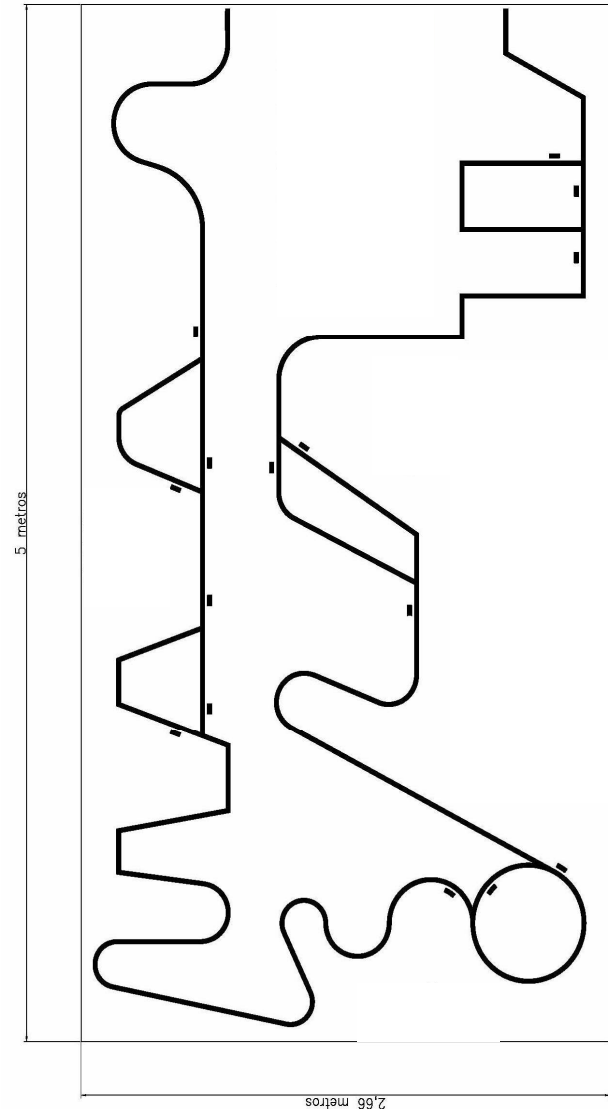


EDUCAMADRID



# Índice

1. Introducción
2. Dudas sobre el robot Sumo
3. Arquitectura del robot
4. Madridbot 2009
5. Arquitectura Subsumption
6. Una estrategia de solución
7. Subsistema de seguimiento de línea
8. Subsistema de detección de bifurcación
9. Pseudo código
10. Sistemas de control automático
11. Recursos
12. Contacto



# Dudas sobre el robot Sumo

Repaso sobre las estrategias de solución vistas en la anterior sesión y otras cuestiones suscitadas por los alumnos.

Videos sobre Sumo Robot:

<http://www.youtube.com/watch?v=qxgcLo8SjYA>

<http://www.youtube.com/watch?v=gIYMAymGzdl>

<http://www.youtube.com/watch?v=KJ7DxdV5IFw>

# Introducción

En esta prueba se valora la habilidad de un micro-robot para recorrer un camino sinuoso previamente establecido en el menor tiempo posible. Se conjugan, por lo tanto, dos aspectos importantes: capacidad de detección y seguimiento del camino y rapidez con que se ejecutan las maniobras.

Mas información sobre la prueba:

<http://www.madridbot.org/Documentos/Documentos%202007/Rastreadores-Propuesta%202007.pdf>

# Introducción

Para conocer mejor este robot, veremos algunos ejemplos de Internet:

<http://www.youtube.com/watch?v=cEV5s7HM1QM>

<http://www.youtube.com/watch?v=mSevpXhwuO8>

<http://www.youtube.com/watch?v=th-G4BdRNXs>

# Arquitectura del robot

## **Modelado de un robot rastreador**

Para desarrollar un robot rastreador, cada alumno deberá plantearse las siguientes preguntas:

¿Qué tipo de sistema locomotor dispondrá un robot rastreador?

¿Cómo interaccionará un robot rastreador con el entorno?

¿El sistema de control, que retos deberá solventar?

# Arquitectura del robot

## **Practica 1: Pensar en la arquitectura del robot**

Cada alumno desarrollara una arquitectura en 5m.

Cada solución, debe pensar en:

1. Sistema de locomoción, si procede en este modelo
2. Sistema de interacción con el entorno

# Arquitectura del robot

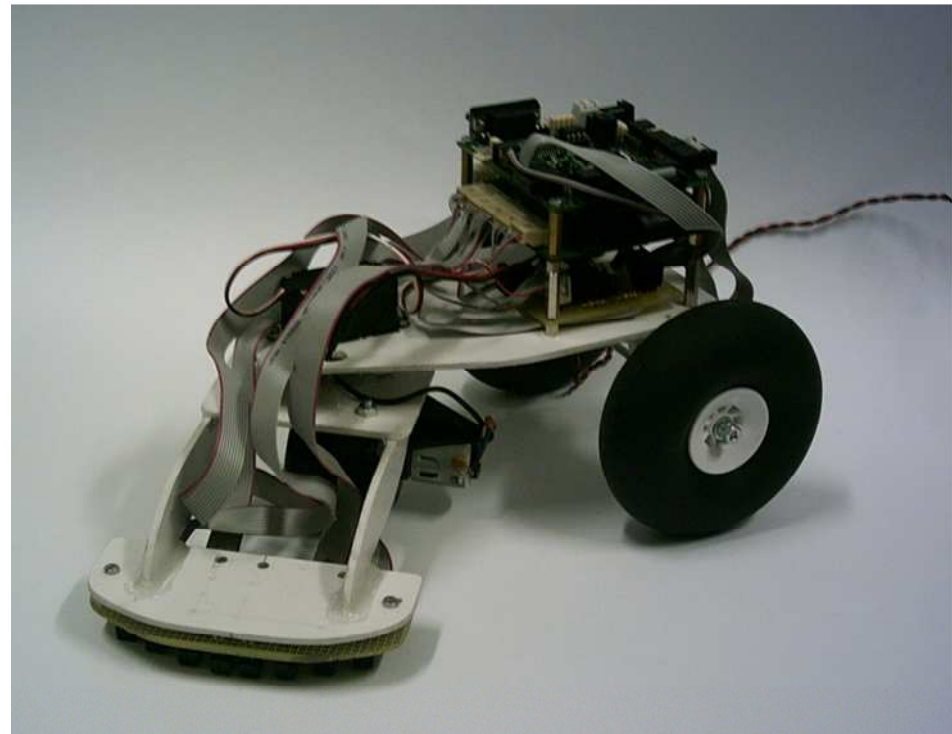
El robot a desarrollar que permitirá superar la prueba del rastreador y constara de la siguiente arquitectura:

## Sistema locomotor

El sistema locomotor del robot constara de 2 motores, los cuales emplean protocolo **PWM** para su control.

## Sistema sensitivo

El sistema sensitivo del robot esta compuesto por 6 sensores que miden el color

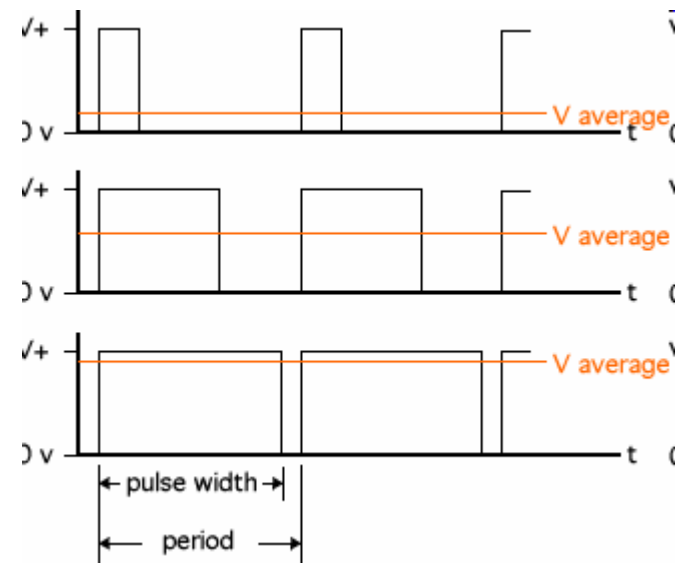
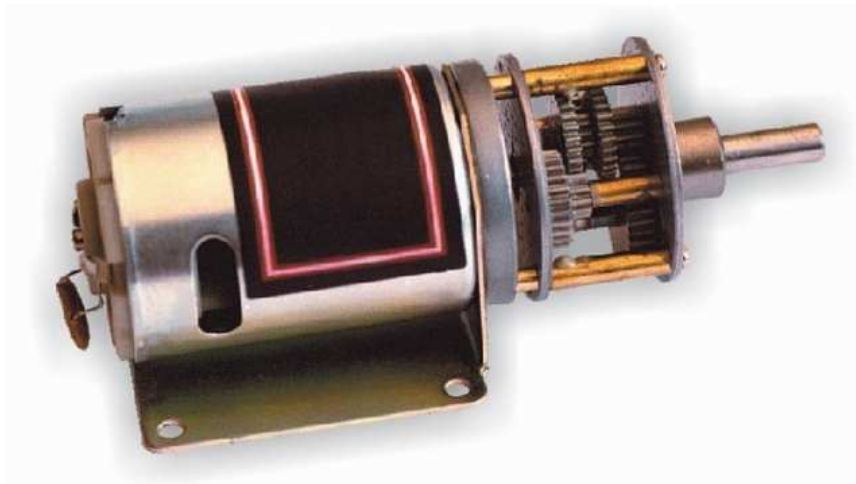




# Arquitectura del robot

## Sistema locomotor

El sistema locomotor estará compuesto por 2 motores de potencia media.



# Arquitectura del robot

## **Sistema locomotor: Objetivos**

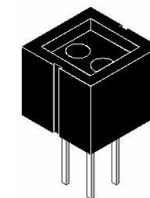
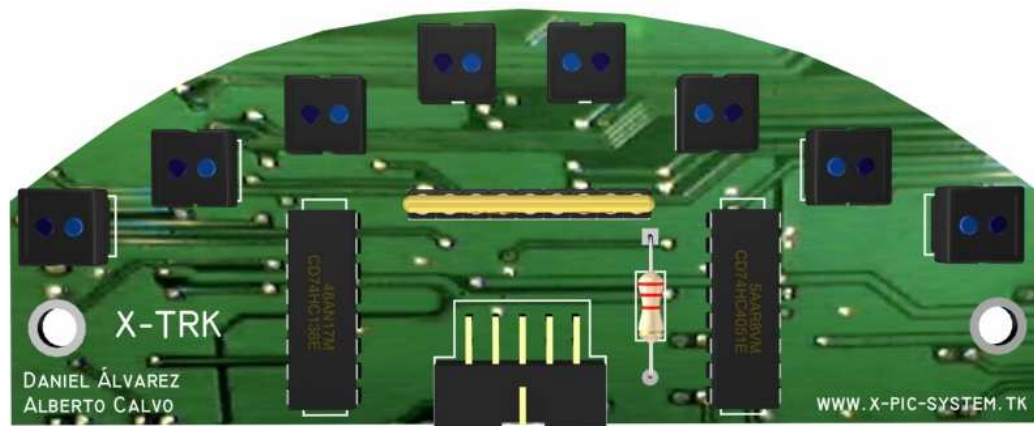
Los objetivos del sistema locomotor son los siguientes:

1. Desarrollo de API para realizar las siguientes operaciones:
  1. Marcha hacia delante
  2. Marcha hacia atrás
  3. Giro a la izquierda de 90°
  4. Giro a la izquierda de X Grados
  5. Giro a la derecha de 90°
  6. Giro a la derecha de X Grados
  7. Stop
  8. Set/Get de velocidad

# Arquitectura del robot

## Sistema sensitivo

El sistema sensitivo del robot esta compuesto por 6 sensores de detección de la reflexión de luz el cual, que debe funcionar como un dispositivo único.



Sensor  
CNY70

Sensor de detección de línea

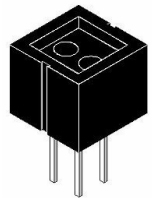
# Arquitectura del robot

## Sensor CNY70

El CNY70 es un sensor de infrarrojos de corto alcance basado en un emisor de luz y un receptor, ambos apuntando en la misma dirección, y cuyo funcionamiento se basa en la capacidad de reflexión del objeto, y la detección del rayo reflejado por el receptor.

Requerimiento del sensor:

El sensor es la necesidad de tener que situarlo muy próximo al objeto para detectar correctamente la reflexión.



Sensor  
CNY70

# Arquitectura del robot

## **Sistema sensitivo : Objetivos**

Los objetivos del sistema sensitivo son los siguientes:

1. Desarrollo de API para realizar las siguientes operaciones:
  1. Detectar posición de línea en GRID de sensores
  2. Detectar señal de bifurcación
  3. Detectar señal de bifurcación falsa

# Arquitectura del robot

## Plataforma de control

La plataforma de control del robot se basara en 2 elementos: Placa controladora con microcontrolador (PIC16F876) y una plataforma de desarrollo

### Placa controladora

La placa electrónica será desarrollada por el IES Antonio Machado

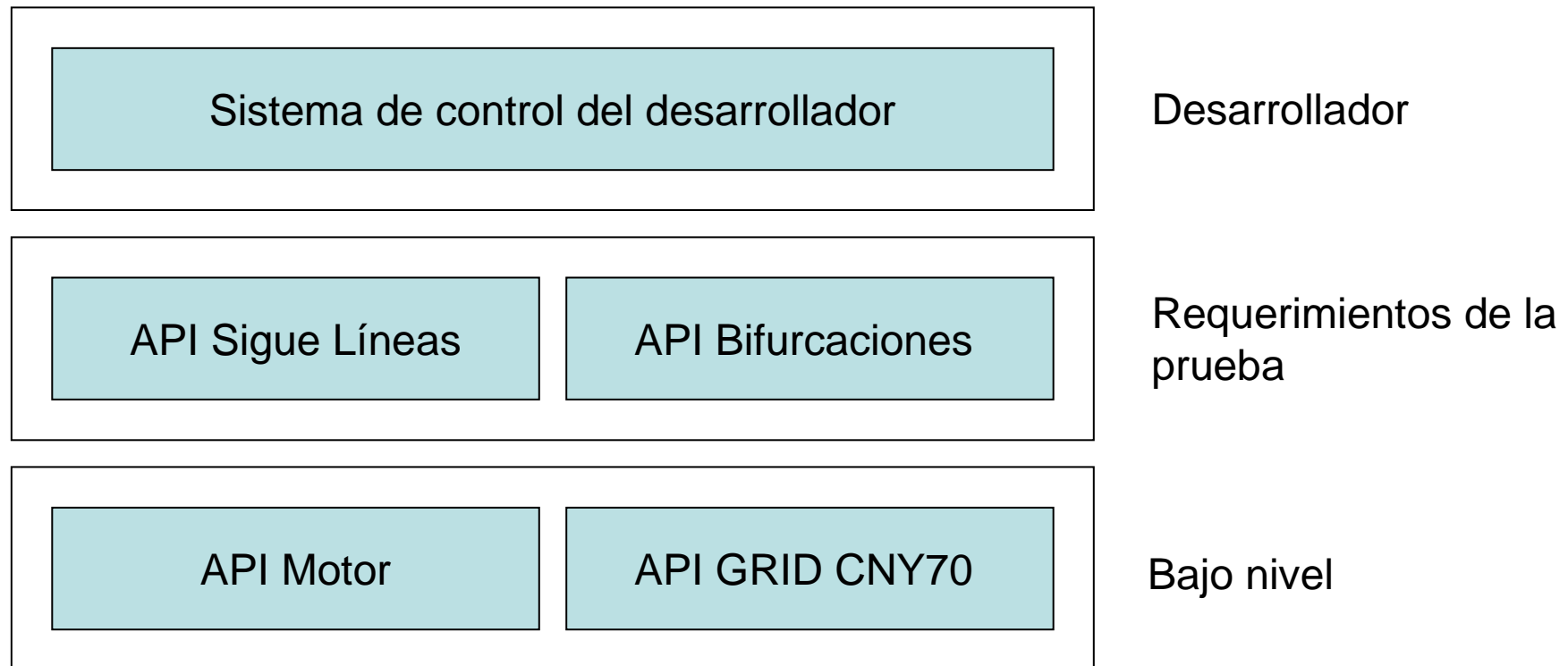
### Plataforma de desarrollo

Para la modelización del robot y su posterior codificación en código para el microcontrolador PIC16F876, se empleara **Proteus**.

# Arquitectura del robot

## Diagrama de subsistemas

Un diagrama de subsistemas que modelara el robot rastreador podría ser el siguiente:

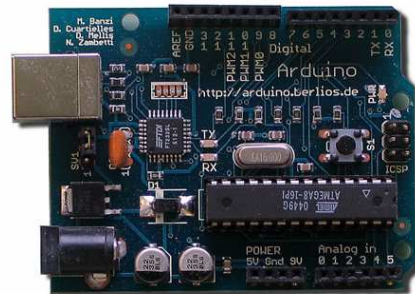


# Arquitectura del robot

## Alternativas

La construcción del desarrollo también podría ser posible a través de las siguientes opciones:

1. Lego Mindstorms NXT
2. Arduino
3. Sun spot
4. VEX





# Madrid-Bot

Madrid-Bot es un concurso de micro robótica organizado por los Centro que imparten las enseñanzas del Ciclo Formativo de Grado Superior de Desarrollo de Productos Electrónicos en la Comunidad Autónoma de Madrid.

Madrid-Bot intenta ser un concurso de micro-robótica y más aun un lugar de encuentro donde los alumnos matriculados en Centros de Educación Secundaria y especialmente los alumnos y alumnas de Bachillerato, Ciclos Formativos de la familia profesional de electrónica, informática, etc.. puedan no sólo competir con sus prototipos sino también compartir sus conocimientos.

# Madridbot

## Pruebas del concurso

Las pruebas que se realizan en Madridbot, son las siguientes:

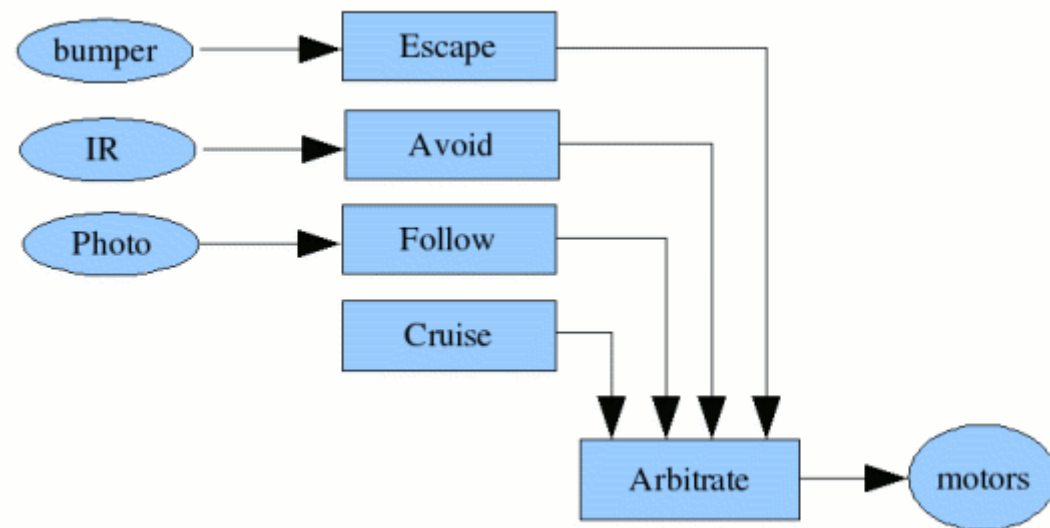
1. Rastreadores
2. Velocistas
3. Laberinto
4. Mini sumo
5. Prueba Libre

# Arquitectura Subsumption

La arquitectura subsumption es una arquitectura de tipo reactiva, desarrollada por Rodney Brooks y se basa en la construcción de capas de comportamiento.

Este robot se podría modelizar a través de 4 comportamientos:

1. Seguir línea
2. Detectar bifurcación



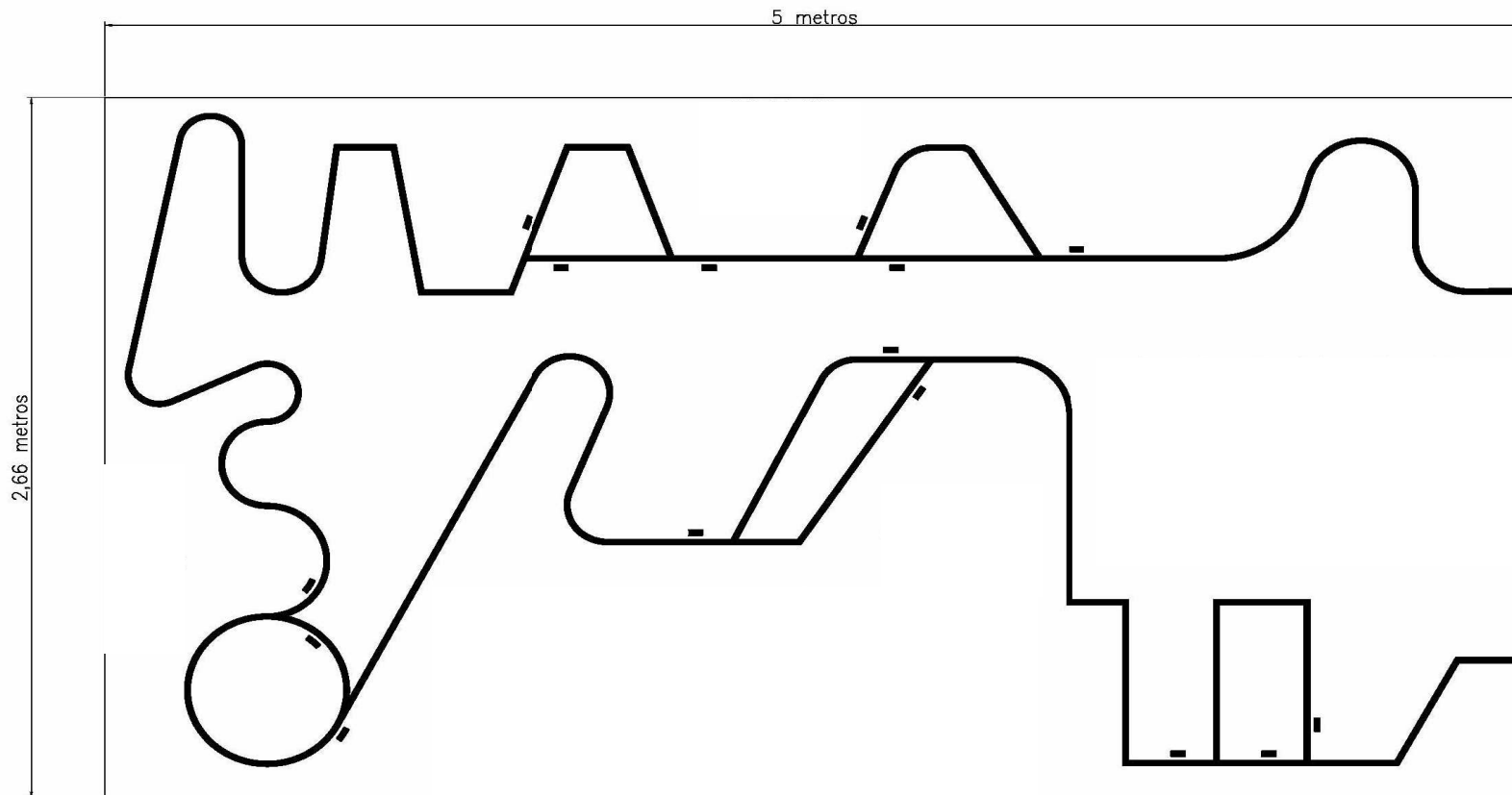
# Una estrategia de solución

La solución se basaría en un sistema que resolviera las siguientes necesidades:

1. ¿Sigo la línea?
2. ¿He detectado una bifurcación?

# Una estrategia de solución

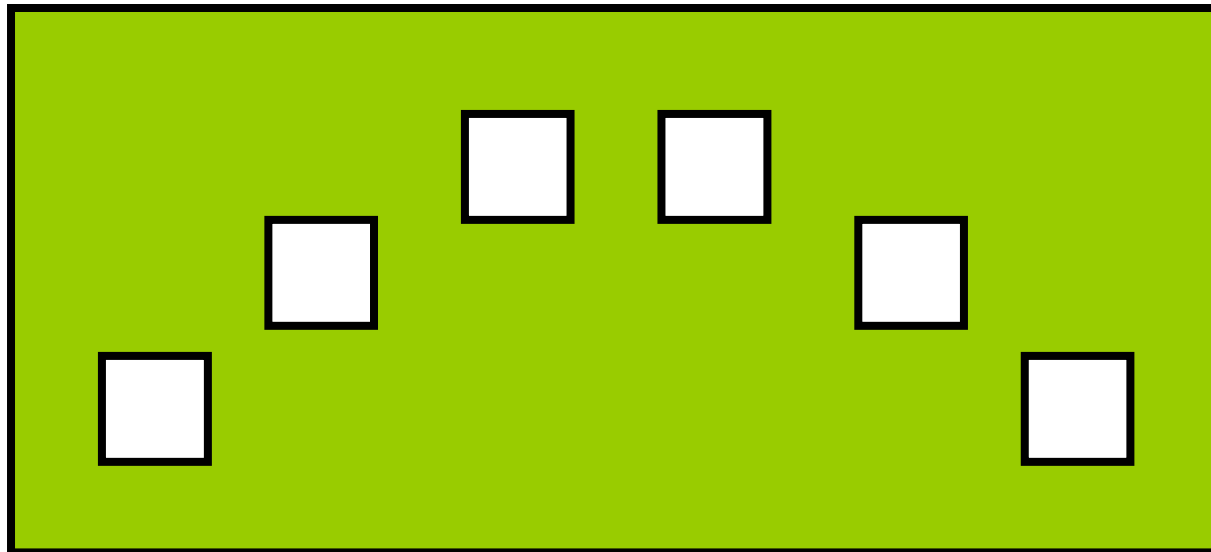
El robot tiene como misión recorrer todo el circuito y detectar las marcas de bifurcación y atender a ellas.



# Una estrategia de solución

Por tanto, el alumno debe dominar como usar el GRID de sensores CNY70 para saber:

1. Hacia donde dirigirse
2. Detectar marcas de bifurcación



# Una estrategia de solución

## **Subsistema de seguimiento de línea**

El subsistema de seguimiento de línea debe velar por mantener al robot en la ruta.

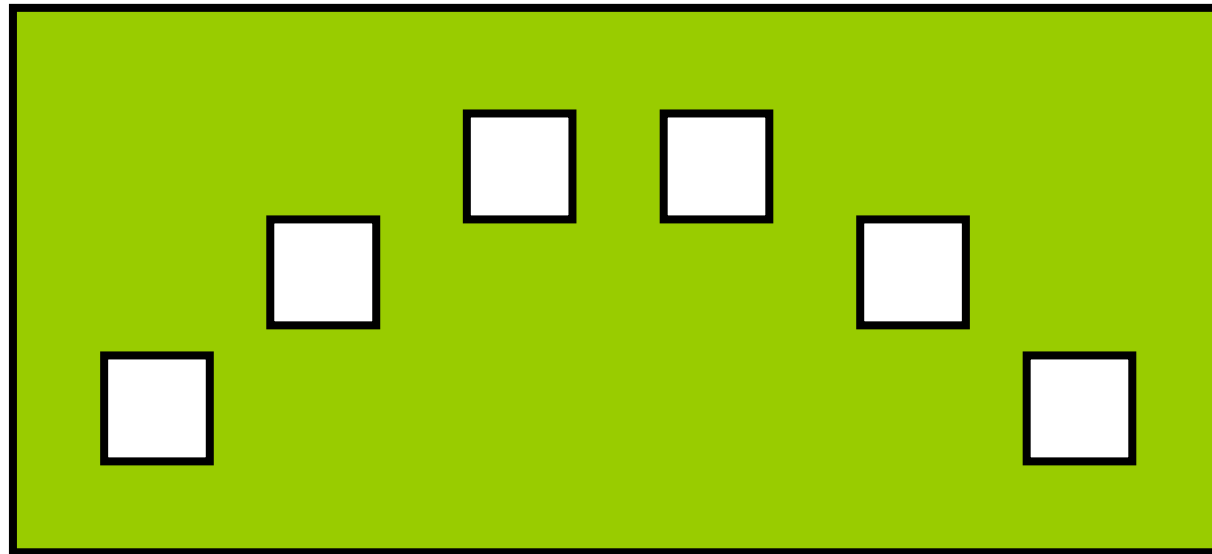


# Una estrategia de solución

## Subsistema de seguimiento de línea

El subsistema deberá manejar 3 casos:

1. Curso correcto
2. Deriva leve
3. Deriva critica

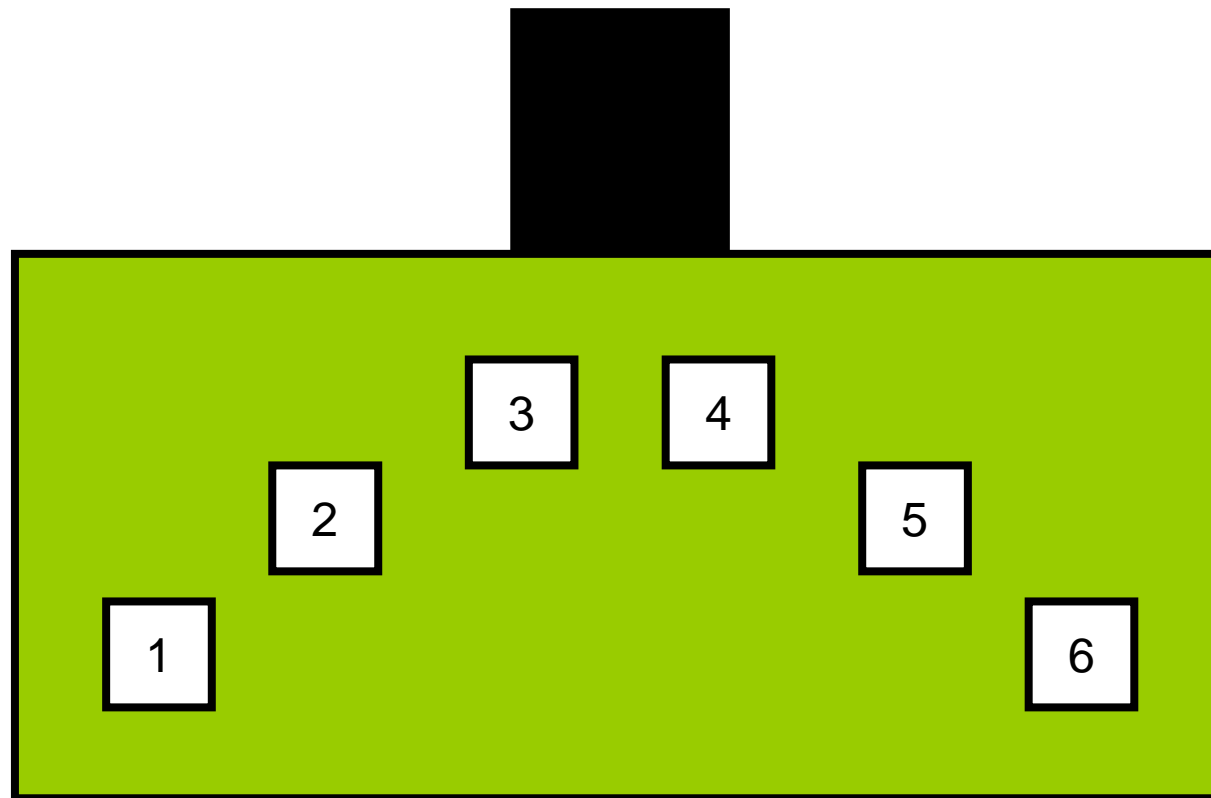




# Una estrategia de solución

## Subsistema de seguimiento de línea: Curso correcto

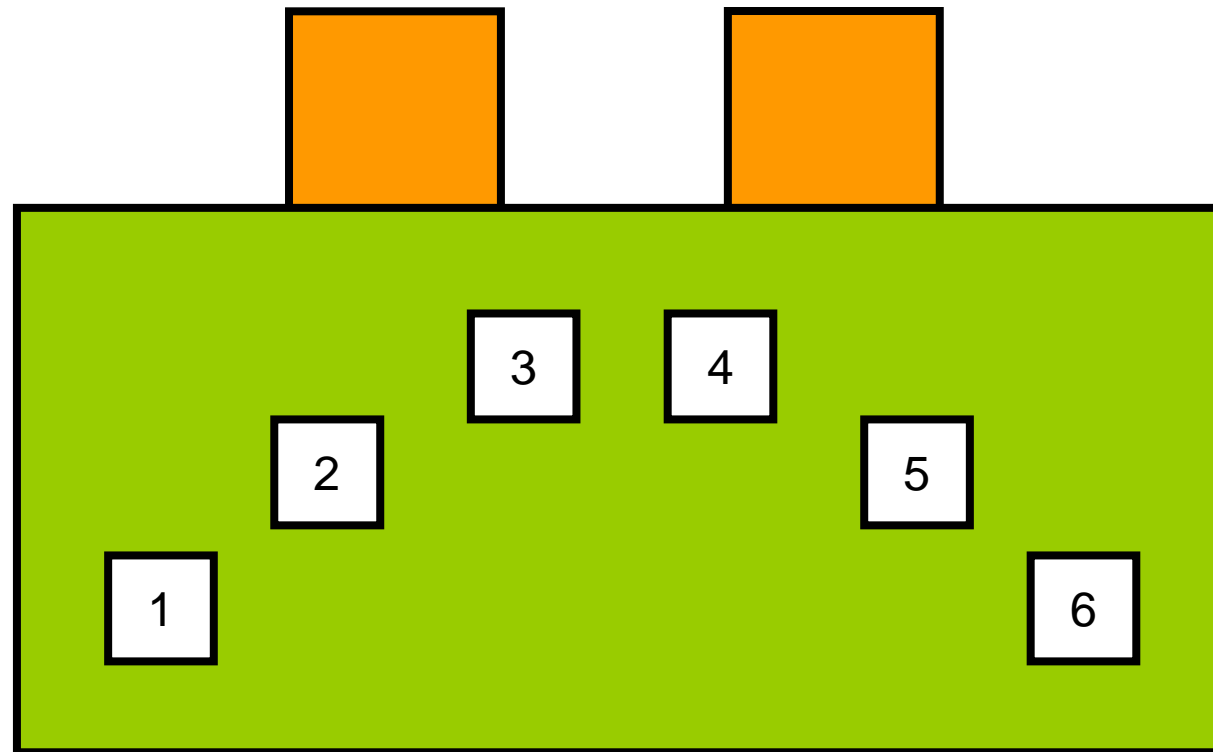
Si el robot, va en curso correcto, no debe realizar ninguna corrección:



# Una estrategia de solución

## Subsistema de seguimiento de línea: Deriva leve

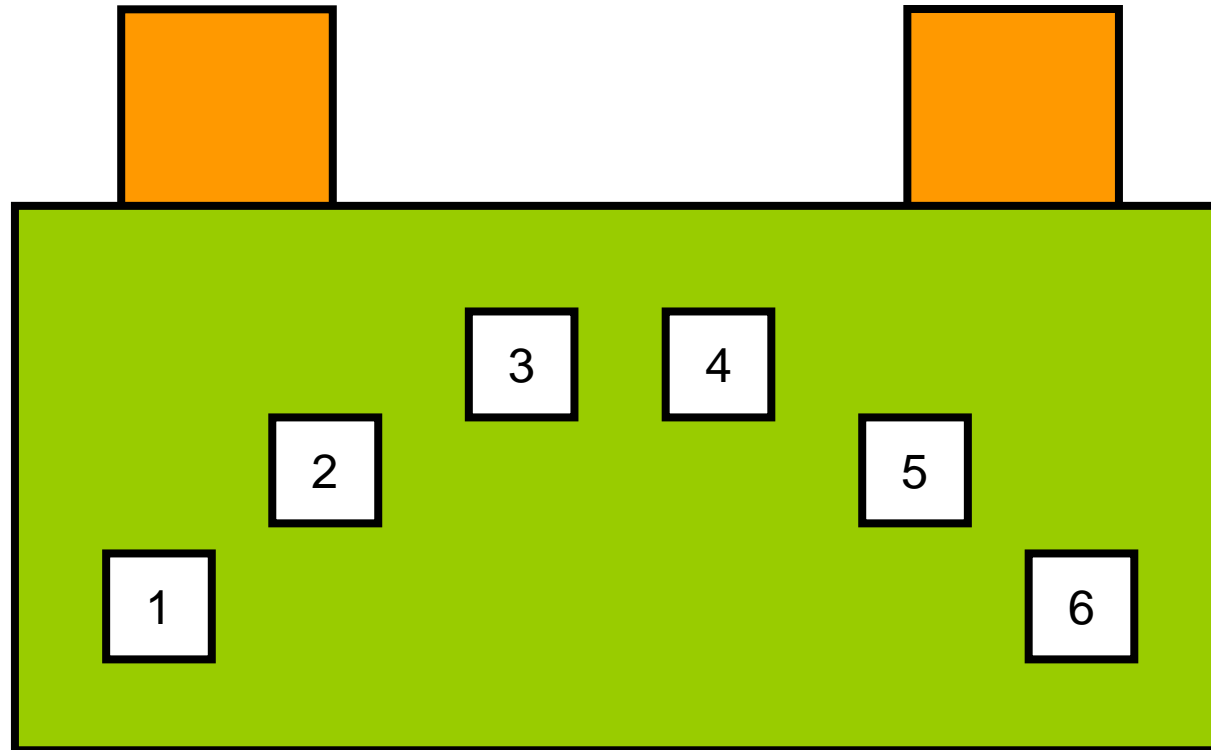
Si el robot se sale del curso de manera leve, debe tomar medidas de corrección.



# Una estrategia de solución

## Subsistema de seguimiento de línea: Deriva crítica

Es posible que en algunos momentos, se produzca una deriva crítica que debe ser corregida para no salirse de la pista.



# Pseudo código

## Seudocodigo rastreadorV1:

```
sigueLineasV1( ) {  
    mientras(true) {  
        status = readCNYSensors();  
        if(status=3 || status=4) {  
            robot.forward( )  
        }else{  
            robot.makeCorrection(status)  
        }  
    }  
}
```

# Una estrategia de solución

## **Premisas**

El subsistema debe tomar en consideración, las siguientes ideas:

1. Una vez iniciado el trayecto, ¿Qué hacer si se pierde?
2. Detectar si hay derivas de navegación y actuar

# Una estrategia de solución

## **Puntos débiles**

Este tipo de solución, no permite tener un comportamiento suave del robot, por tanto es necesario mejorarlo a través de sistemas de control.

# Una estrategia de solución

## Subsistema de detección de señal de bifurcación

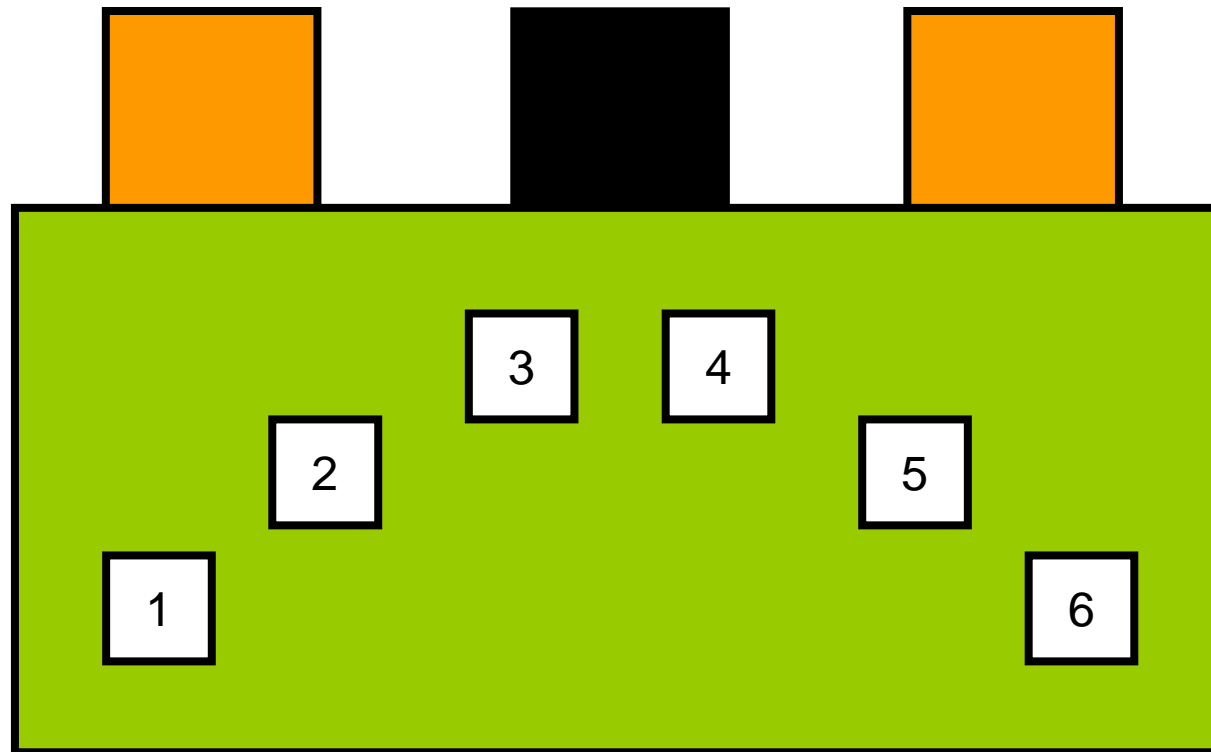
Una vez el robot, es capaz de seguir las líneas que componen el circuito, es necesario mejorar el sistema que usa como entrada el GRID de sensores CNY70



# Una estrategia de solución

## Subsistema de detección de bifurcación

Si el sistema va recto, se activaría el 3 y 4, pero en el camino es posible que se active 1-2 o 5-6.





# Una estrategia de solución

## **Subsistema de detección de bifurcación**

La clave esta en dotar al robot de una cierta memoria que permita detectar si el robot ha detectado una falsa señal de bifurcación o no. Una sencilla forma de hacerlo seria a través de un contador de detecciones que seria parametrizable.

# Pseudo código

## Seudo código rastreadorV2:

```
rastreadorV2( ){  
    int contadorSeñal = 0;  
    mientras(true){  
        readCNYSensors( );  
        detectarSeñal( );  
        if(contadorSeñal > umbralSeñal){  
            bifurcacionDetectada;  
        }  
        seguirLinea( );  
    }  
}
```

# Sistemas de control automático

Los sistemas de control automático mas empleados son los siguientes:

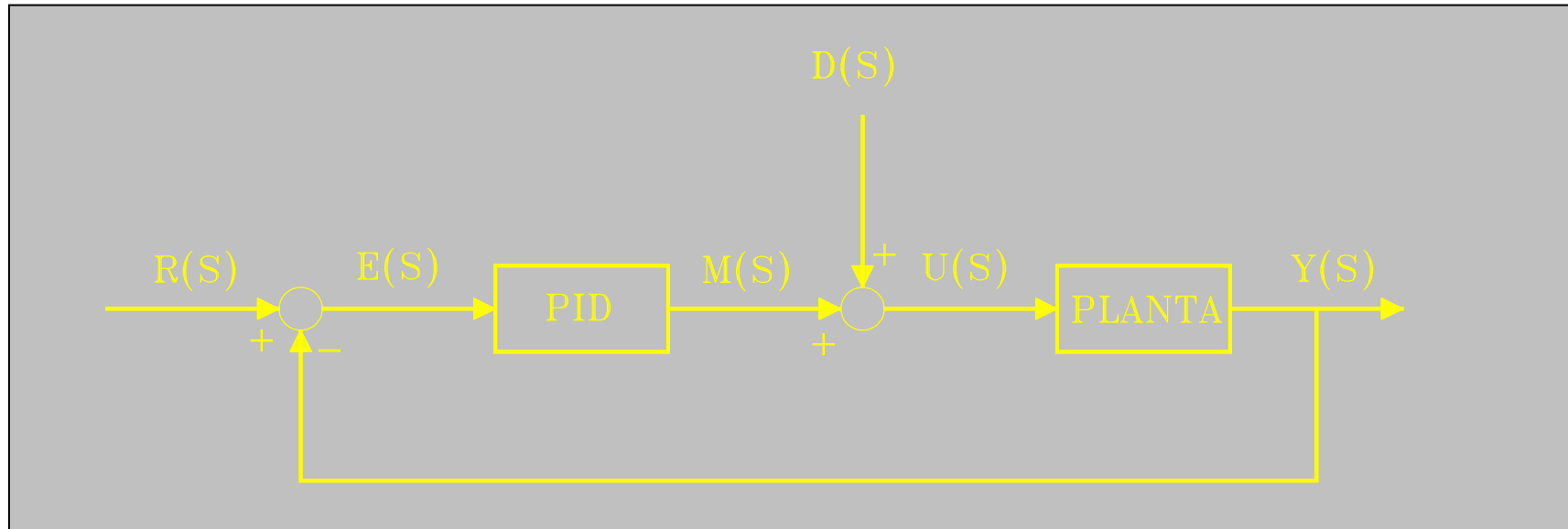
- Control proporcional
- Control integral
- Control derivativo

$$m(t) = K_c e(t)$$

$$m(t) = \frac{K_c}{T_i} \int e(t) dt$$

$$m(t) = T_d \frac{de(t)}{dt}$$

# Sistemas de control automático



donde:

$R(s)$  : Es la señal de referencia o punto de ajuste (Set Point)

$E(s)$  : Es la señal de error

$M(s)$  : Es la señal de salida del controlador

$D(s)$  : Es la señal de perturbación

$U(s)$  : Es la señal o variable de control (variable manipulada)

$Y(s)$  : Es la señal de salida (variable controlada)

# Sistemas de control automático

## **Sistema proporcional**

El dispositivo corrector final no es forzado a tomar una de dos posiciones disponibles. En lugar de esto, tiene un rango continuo de posiciones posibles. La posición exacta que toma es proporcional a la señal de error. En otras palabras, la salida de bloque controlador es proporcional a su entrada.

# Sistemas de control automático

## **Sistema proporcional + Integral**

Este controlador es la suma de una acción proporcional y una integral. Se ha visto que la acción proporcional nos acerca al valor deseado, y la acción integral nos lleva exactamente al valor deseado. Entonces para que combinar ambas acciones, y no sólo usar una acción integral?

# Sistemas de control automático

## **Sistema proporcional + Integral + Derivativo**

Aún cuando el control proporcional-integral es adecuado para la mayoría de las situaciones de control, no es adecuado para todas las situaciones. Hay algunos procesos que presentan problemas de control muy difíciles que no pueden manejarse con un control PI.

# Recursos

## Documentación adicional

Eventos en España:

<http://www.madridbot.org/index.htm>

<http://complubot.educa.madrid.org/inicio.php?seccion=principal>

<http://www.eis.uva.es/amuva/robolid/>

Sumobot:

<http://www.webdearde.com/modules.php?name=Forums&file=viewtopic&printertopic=1&t=1378&start=0&postdays=0&postorder=asc&vote=viewresult&popup=1>

<http://gro.usal.es/robots/diego/diegorbt.pdf>

<http://www.foroselectronica.es/f56/robot-rastreador-slayer-644.html>

<http://www.amazon.com/Robot-Sumo-Official-Pete-Miles/dp/007222617X>

<http://gedex.web.id/index.php?s=computing>

Control PID:

<http://www.aiolosrd.com/modules.php?name=Forums&file=viewtopic&printertopic=1&t=121&start=0&postdays=0&postorder=asc&vote=viewresult&popup=1&sid=95b183615ba733f76b4c7083c6717956>



# Contacto

## Dudas ruegos y preguntas

Para poneros en contacto conmigo:

Mail: [bren@juanantonio.info](mailto:bren@juanantonio.info)

Skype: [juanantonioBM](https://www.skype.com/people/juanantonioBM)