Introducción a la Robótica Móvil

Segundo cuatrimestre de 2016

Departamento de Computación - FCEyN - UBA

Práctica - clase 3

Taller Actuación

Problematica

¿Qué cosas se pueden controlar con un PID?

Problematica

¿Qué cosas se pueden controlar con un PID?

- caldera
- aire acondicionado
- posición de una válvula
- nivel de agua de un tanque

¿Qué es lo primero que tenemos que controlar en un robot?

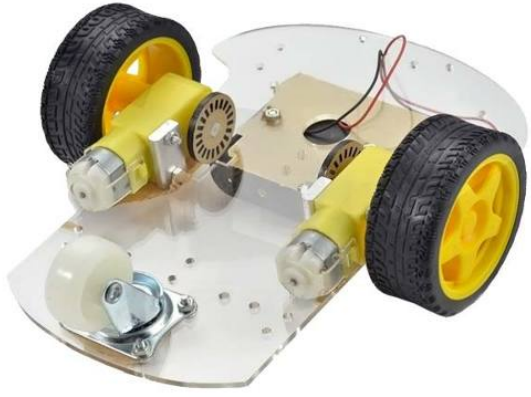
Problematica

¿Qué cosas se pueden controlar con un PID?

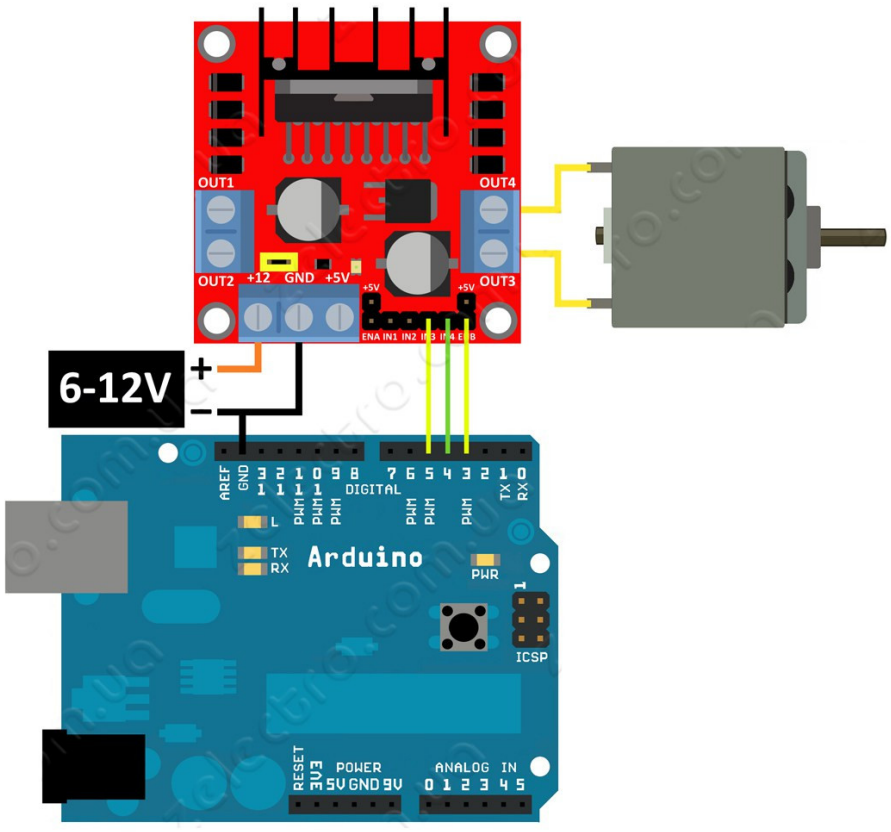
- caldera
- aire acondicionado
- posición de una válvula
- nivel de agua de un tanque

¿Qué es lo primero que tenemos que controlar en un robot?

- los motores del robot



¿Cómo controlamos un motor?



Componentes:

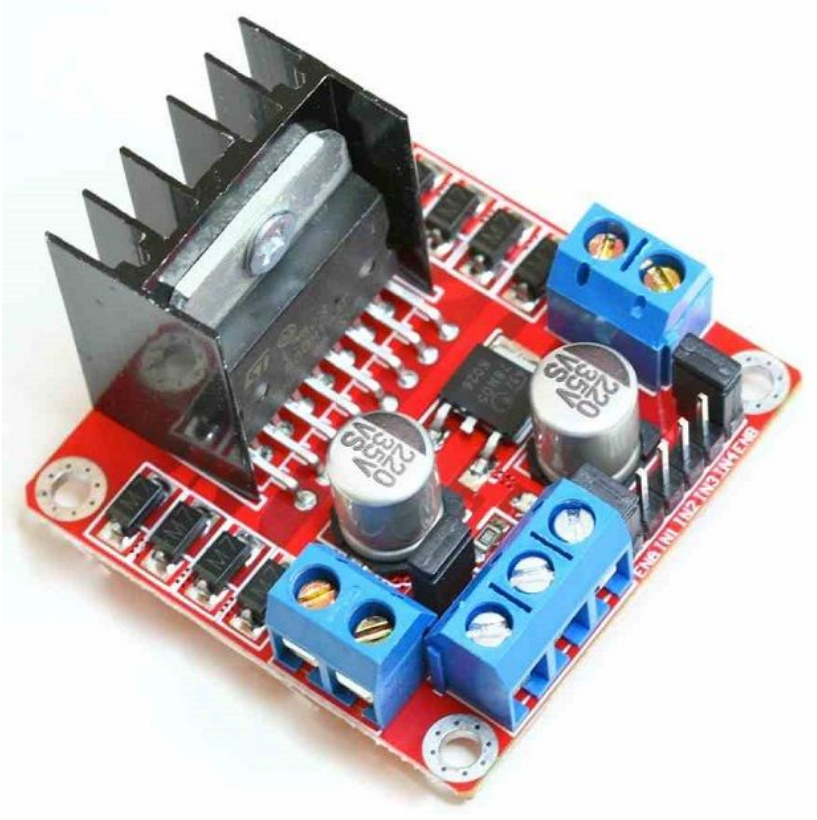
- Arduino
- driver (?)
- motor controlado por `PWM` (?)

Driver

¿Qué es un driver?

Los driver permite manejar la corriente que necesitan los motores para moverse, que no puede ser otorgada por un microcontrolador.

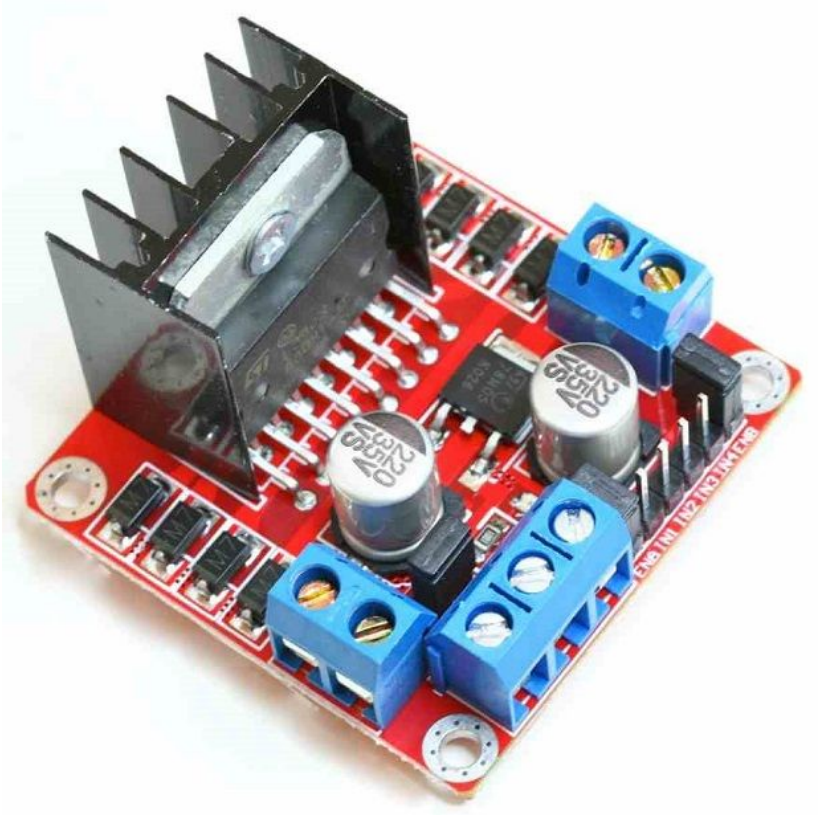
- En la practica utilizaremos el L298
- Puede controlar dos motores CC o un PAP
- Vin = 6V 12V
- Corrienta por canal de salida de hasta 2A



Driver

¿Qué es un driver?
Los driver permite manejar la corriente que necesitan los motores para moverse, que no puede ser otorgada por un microcontrolador.

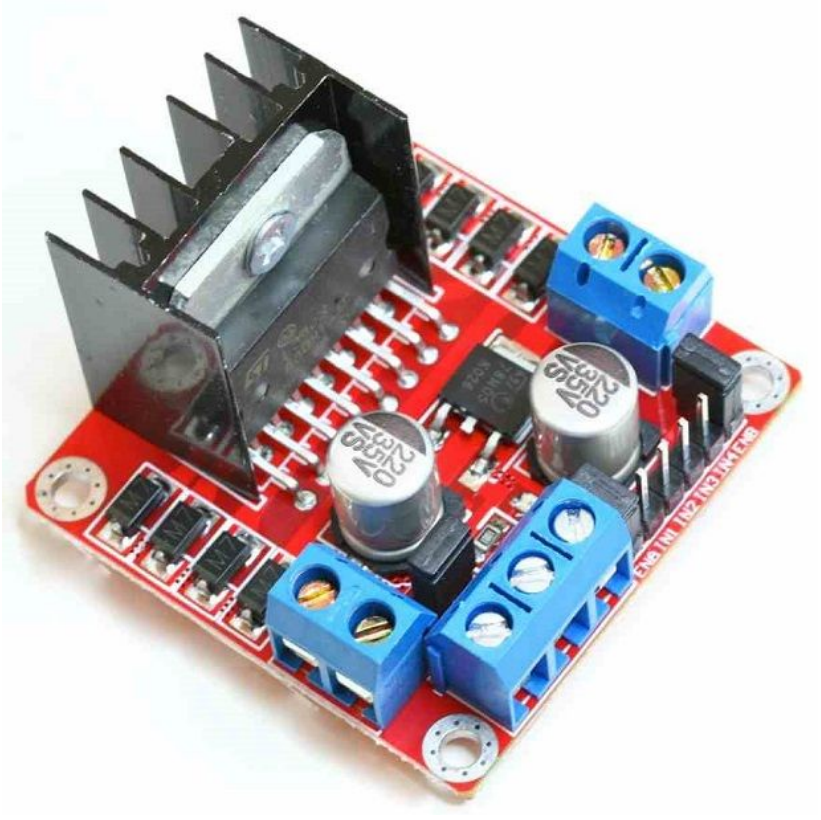
- En la practica utilizaremos el L298
- Puede controlar dos motores CC o un PAP
- Vin = 6V 12V
- Corrienta por canal de salida de hasta 2A



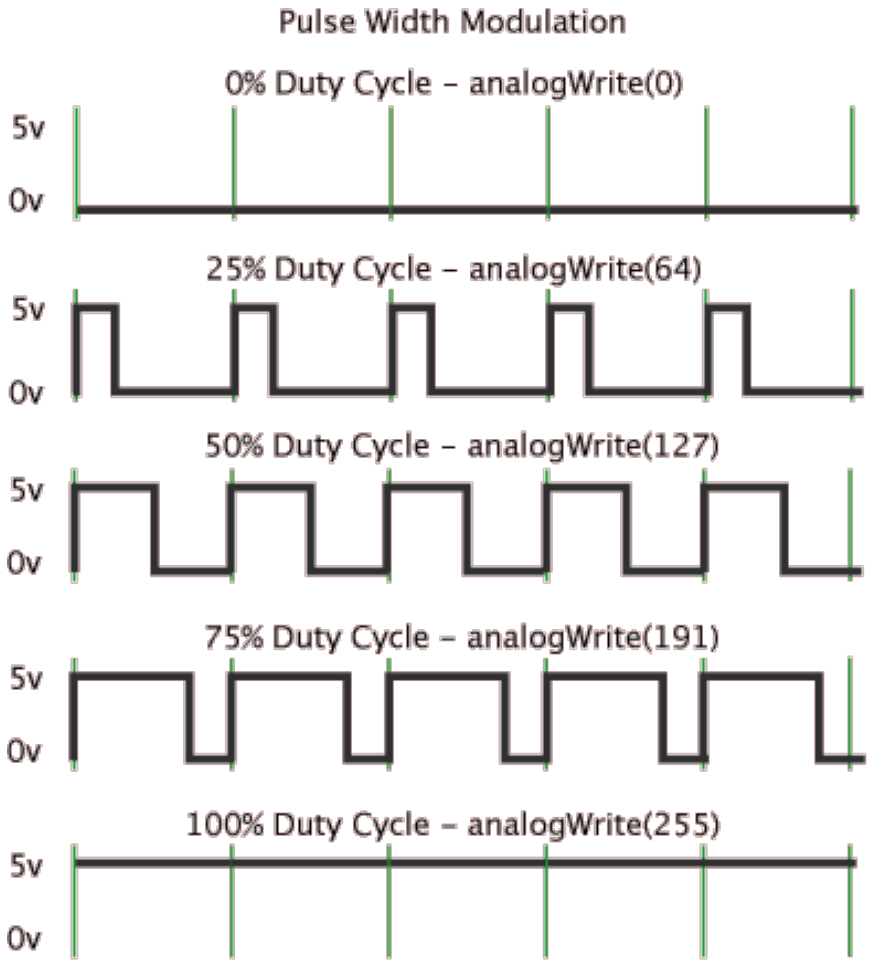
Driver

¿Qué es un driver?
Los driver permite manejar la corriente que necesitan los motores para moverse, que no puede ser otorgada por un microcontrolador.

- En la practica utilizaremos el L298
- Puede controlar dos motores CC o un PAP
- Vin = 6V 12V
- Corrienta por canal de salida de hasta 2A

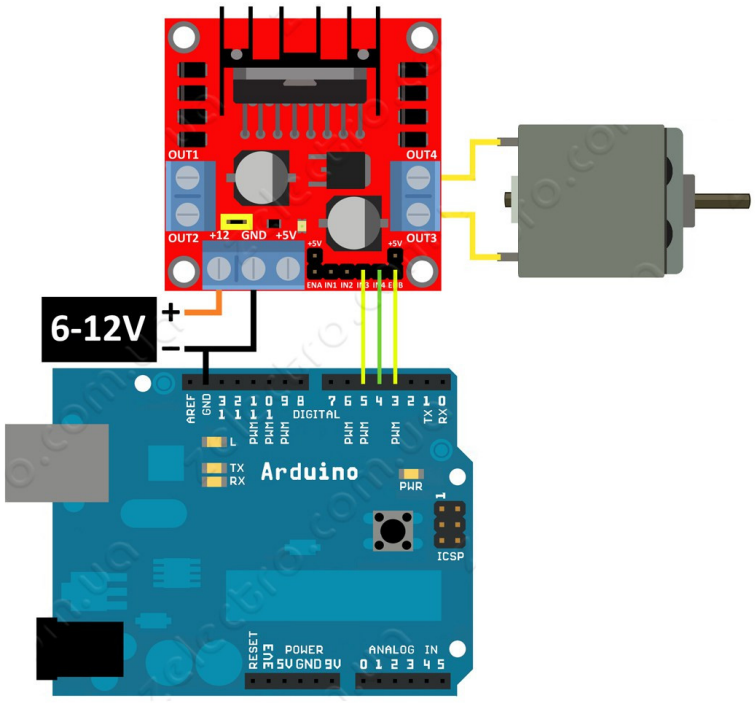


Modulación por Ancho de Pulso



- Los motores los controlamos con PWM ¿Por qué?
- Se utiliza en otras aplicaciones, ¿Ejemplos?
- En Arduino usamos la función `analogWrite()`.

Control de un motor: Esquema



Componentes:

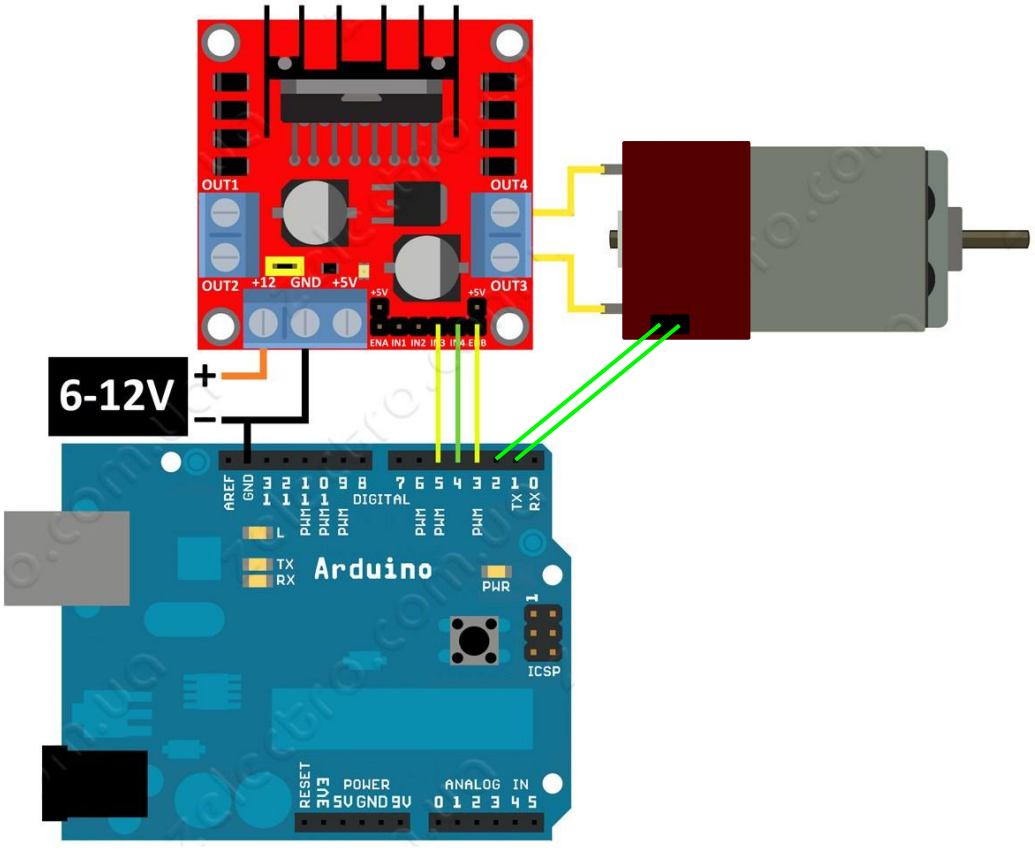
- Arduino
- motor controlado por `PWM`
- driver

¿ Cómo lo controlo?

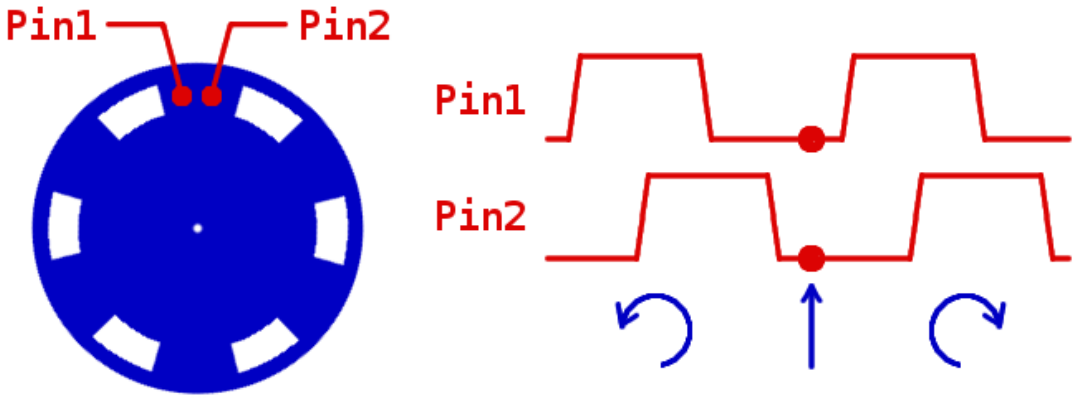
- un *pin* para *enable*
- dos *pines* para `PWM`
- adelante $pwm_1 = 100$ y $pwm_2 = 0$
- atras $pwm_1 = 0$ y $pwm_2 = 100$
- frenado $pwm_1 = 0$ y $pwm_2 = 0$

¿Falta algo?
¿Podemos cerrar el lazo?

Control de un motor: Esquema Completo



Encoder + Libreria



Funciones:

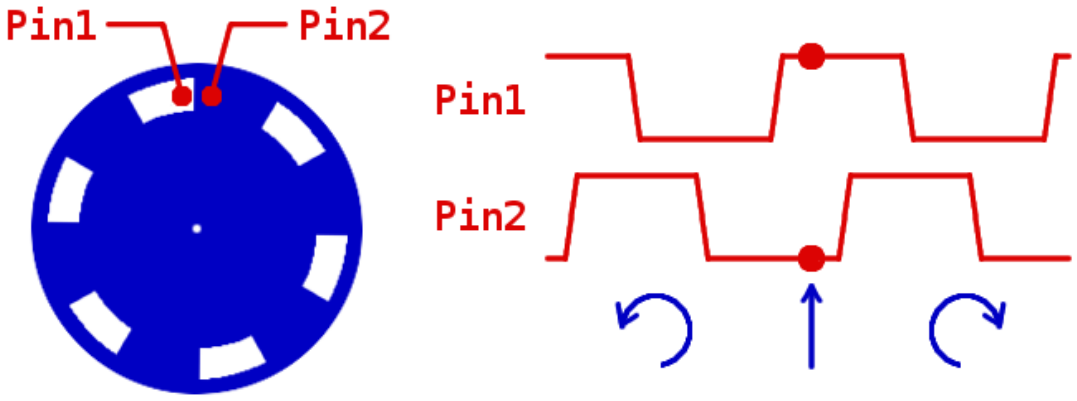
`#include <Encoder.h>`: incluye la librería Encoder.

`Encoder myEnc(pin1, pin2)`: crea un nuevo objeto *Encoder* usando 2 *pins*. (Mayor *performance* si son pines de interrupciones ¿Por qué?)

`myEnc.read()`: Devuelve la posición acumulada, puede ser positiva o negativa.

`myEnc.write(newPosition)`: Cambia el valor de la posición acumulada (reset) al valor *newPosition*.

Encoder + Libreria



Funciones:

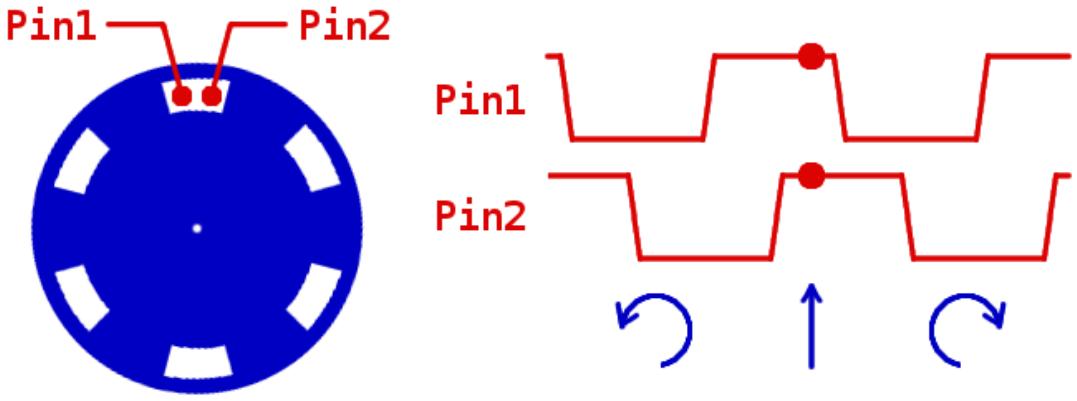
`#include <Encoder.h>`: incluye la librería Encoder.

`Encoder myEnc(pin1, pin2)`: crea un nuevo objeto *Encoder* usando 2 *pins*. (Mayor *performance* si son pines de interrupciones ¿Por qué?)

`myEnc.read()`: Devuelve la posición acumulada, puede ser positiva o negativa.

`myEnc.write(newPosition)`: Cambia el valor de la posición acumulada (reset) al valor *newPosition*.

Encoder + Libreria



Funciones:

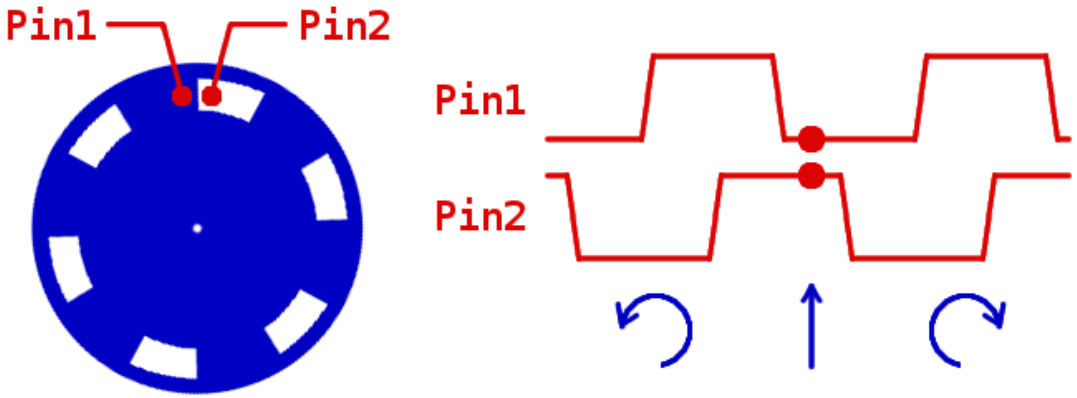
`#include <Encoder.h>`: incluye la librería Encoder.

`Encoder myEnc(pin1, pin2)`: crea un nuevo objeto *Encoder* usando 2 *pins*. (Mayor *performance* si son pines de interrupciones ¿Por qué?)

`myEnc.read()`: Devuelve la posición acumulada, puede ser positiva o negativa.

`myEnc.write(newPosition)`: Cambia el valor de la posición acumulada (reset) al valor *newPosition*.

Encoder + Libreria



Funciones:

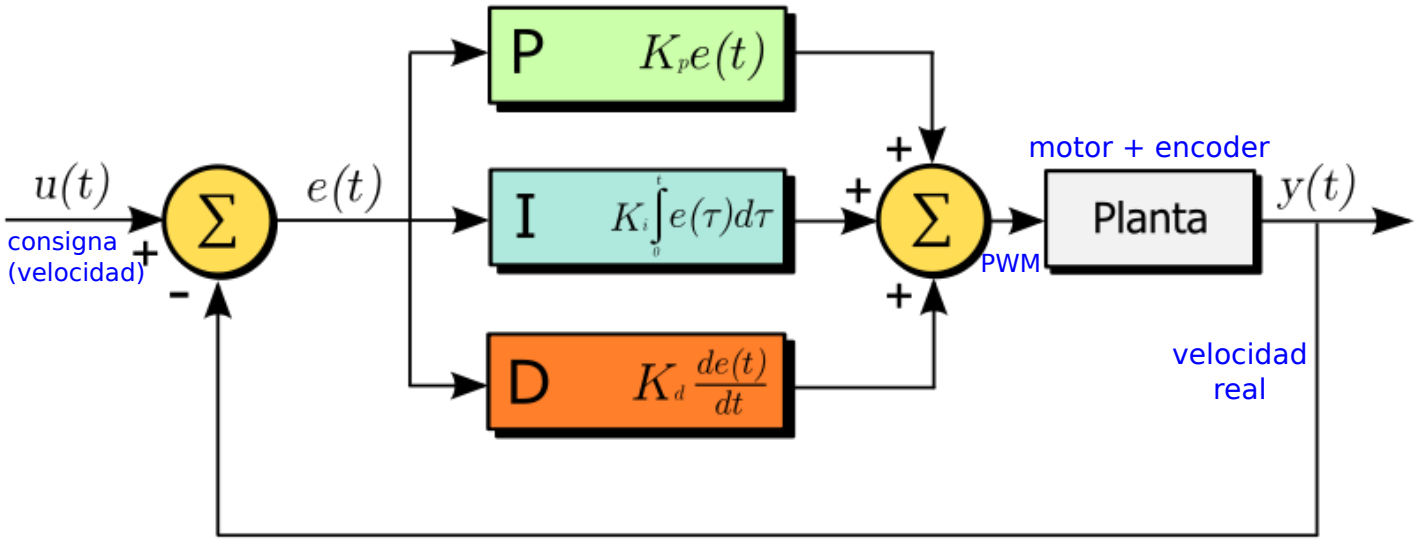
`#include <Encoder.h>`: incluye la librería Encoder.

`Encoder myEnc(pin1, pin2)`: crea un nuevo objeto *Encoder* usando 2 *pins*. (Mayor *performance* si son pines de interrupciones ¿Por qué?)

`myEnc.read()`: Devuelve la posición acumulada, puede ser positiva o negativa.

`myEnc.write(newPosition)`: Cambia el valor de la posición acumulada (reset) al valor *newPosition*.

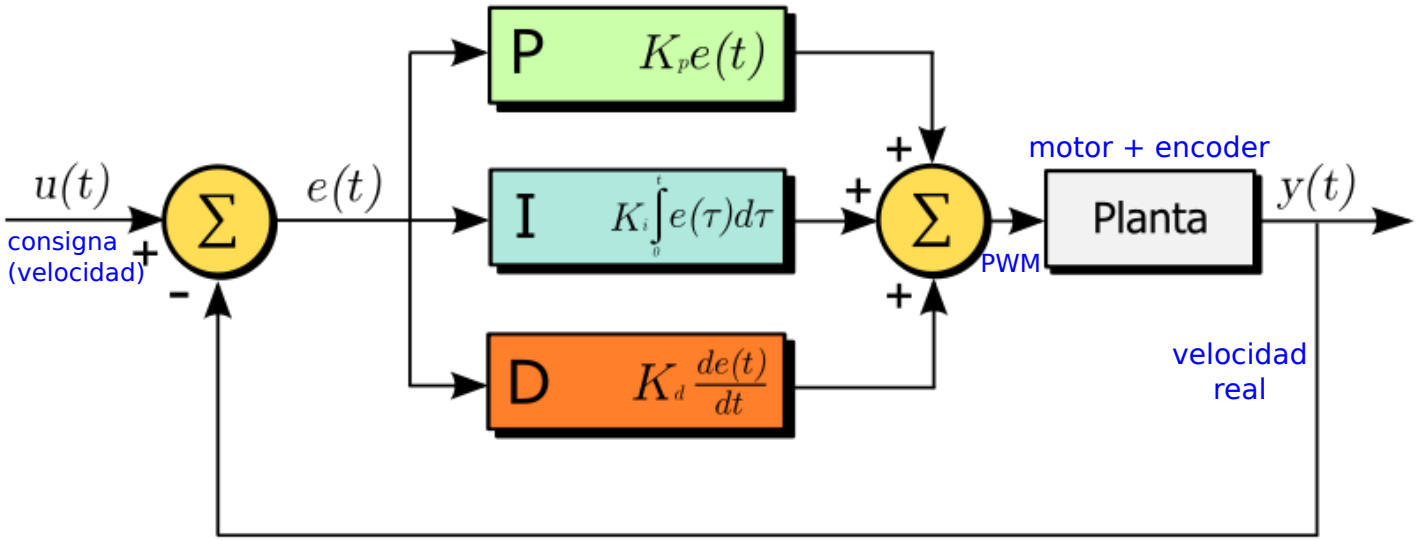
Juntando Todo



Pasos a seguir:

- Calculo el error actual para P.
- Para calcular el I acumulo el error . $I+=e(t)$ (¿satura?)
- Calculo la diferencia entre el error actual y el error anterior
  tip: $e(t)-e(t-1)=(y(t)-u(t))-(y(t-1)-u(t))=y(t)-y(t-1)$.
- Calculo un valor de `PWM` sumando estos valores multiplicados por sus constantes respectivamente.

Juntando Todo

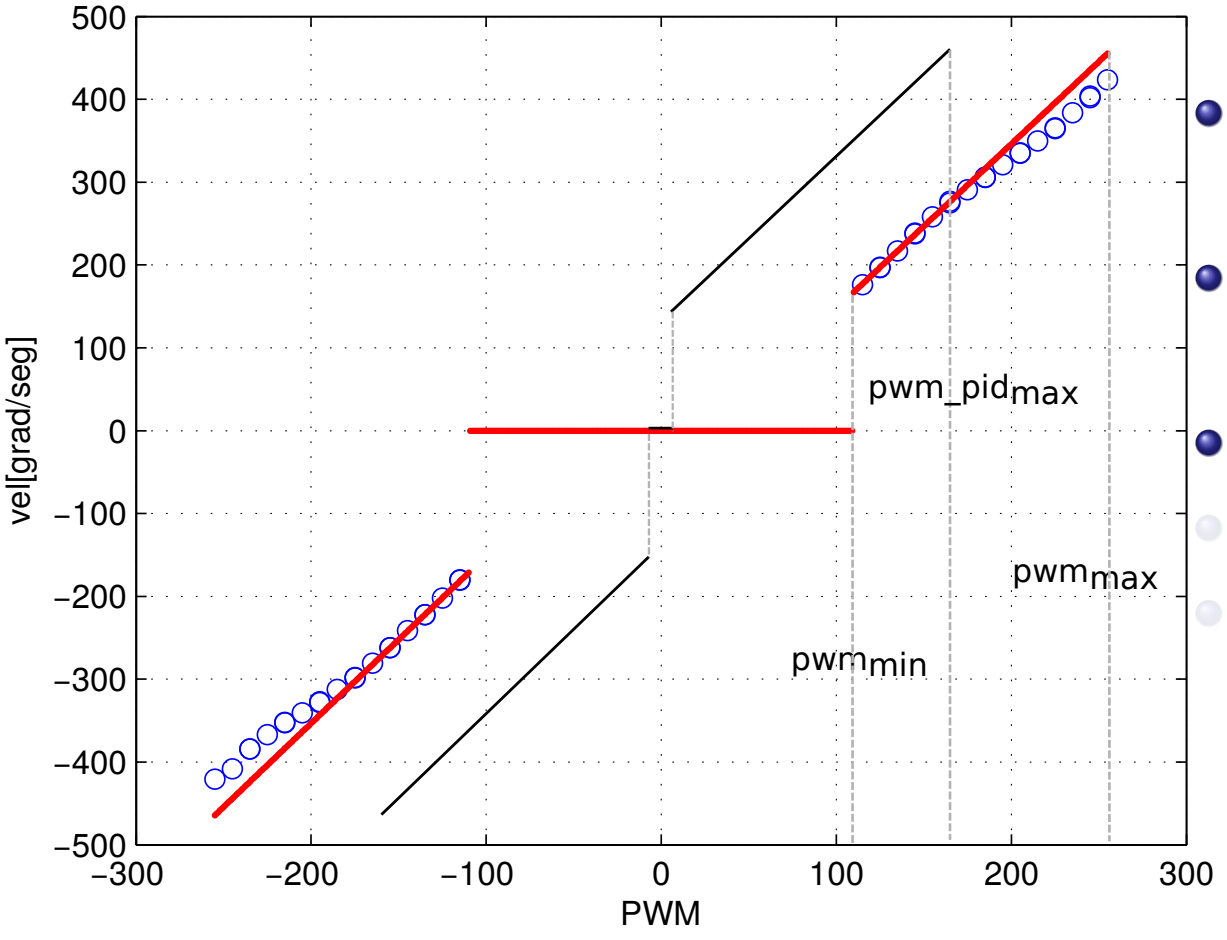


Pasos a seguir:

- Calculo el error actual para P.
- Para calcular el I acumulo el error . $I+=e(t)$ (¿satura?)
- Calculo la diferencia entre el error actual y el error anterior
  tip: $e(t)-e(t-1)=(y(t)-u(t))-(y(t-1)-u(t))=y(t)-y(t-1)$.
- Calculo un valor de `PWM` sumando estos valores multiplicados por sus constantes respectivamente.

DC motor dead zone

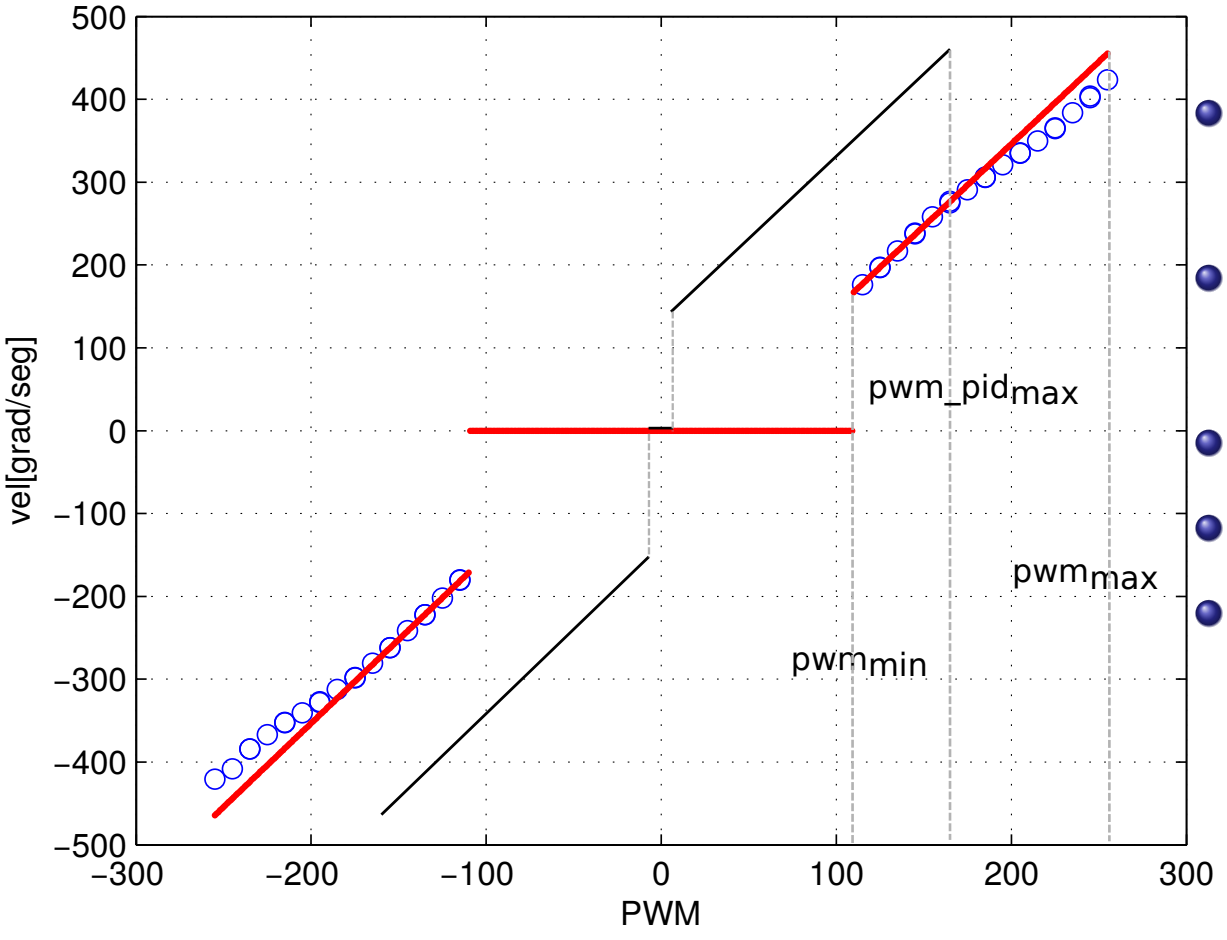
Funcionamiento de un motor real



- Los motores tienen una "zona muerta"
- ¿Afecta esto el control del motor con un PID?¿Por qué?
- ¿Cómo lo solucionamos?
- $pwm\_pid_{max} = pwm_{max} - pwm_{min}$
- $pwm\_pid_{out} = pwm\_pid_{out} + |pwm_{min}|$

DC motor dead zone

Funcionamiento de un motor real



- Los motores tienen una "zona muerta"
- ¿Afecta esto el control del motor con un PID?¿Por qué?
- ¿Cómo lo solucionamos?
- $pwm\_pid_{max} = pwm_{max} - pwm_{min}$
- $pwm\_pid_{out} = pwm\_pid_{out} + |\ pwm_{min}\ |$

Ejercicios

Basta de clases, quiero jugaaaar!

Bajar consigna desde la página!