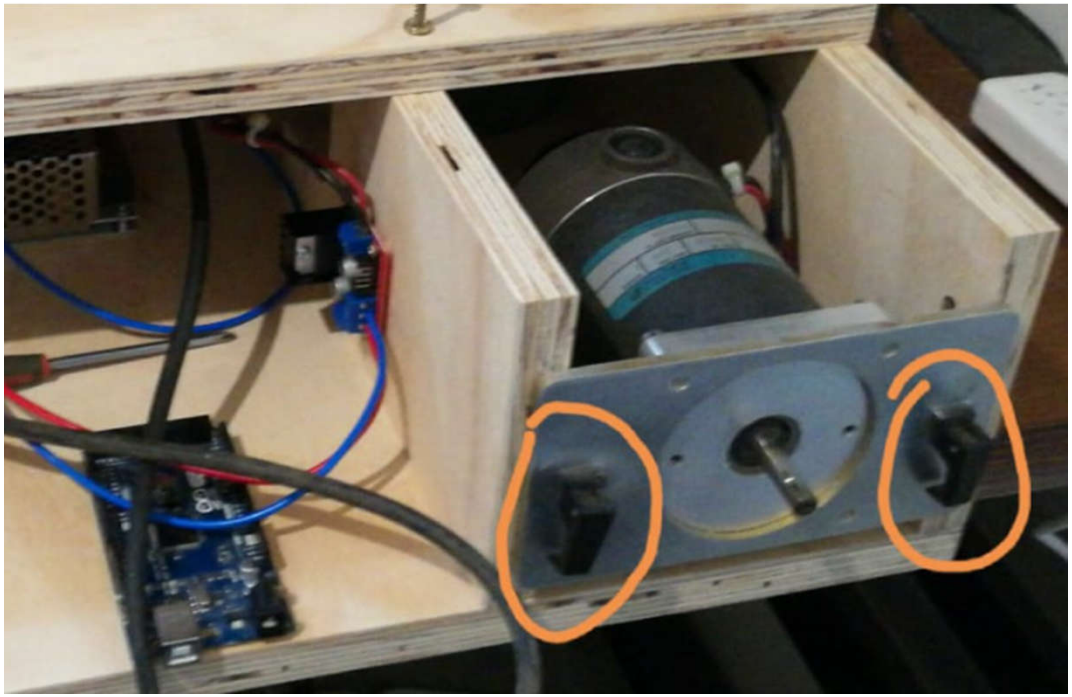
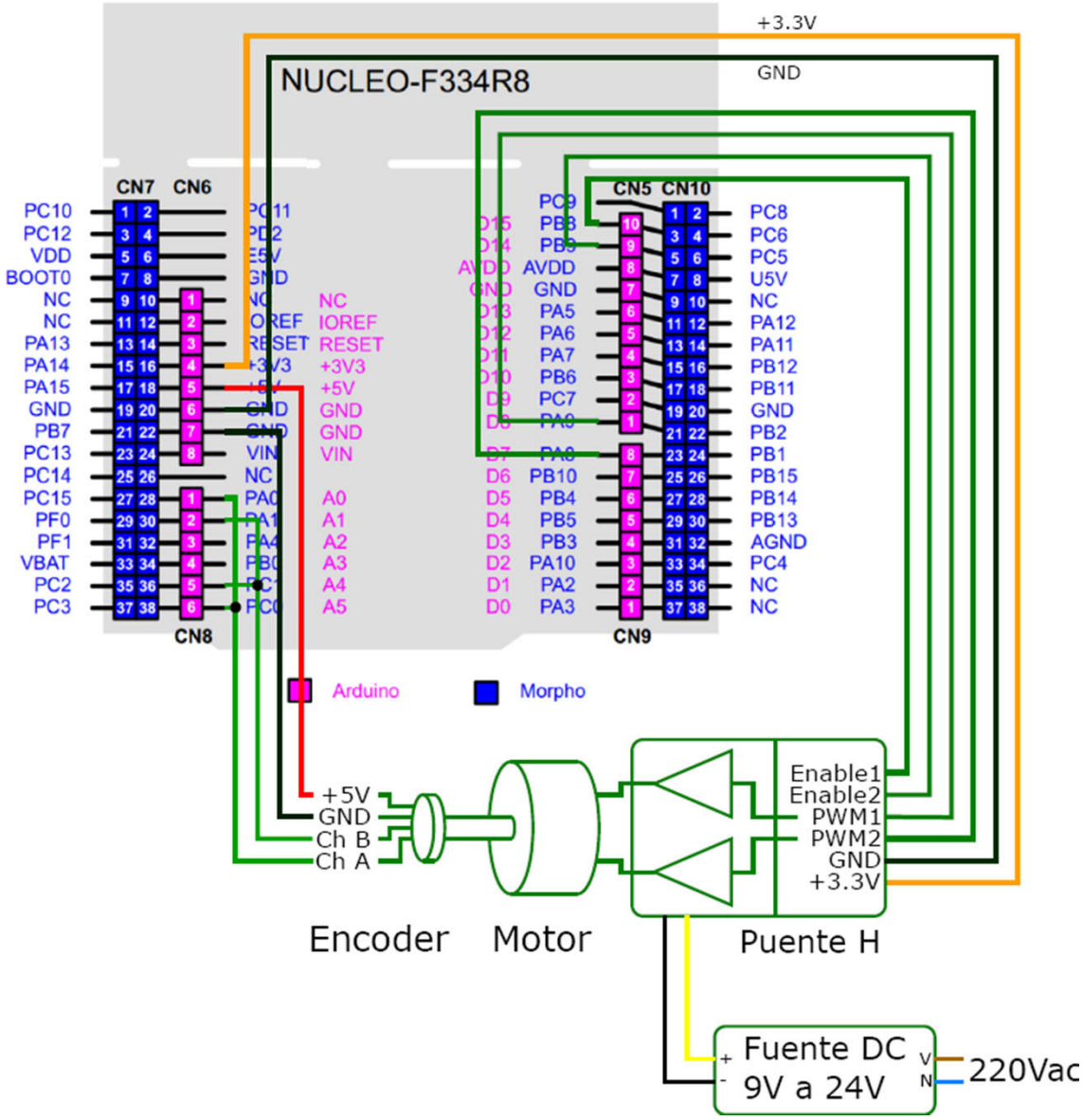
PID con placa Nucleo y Python

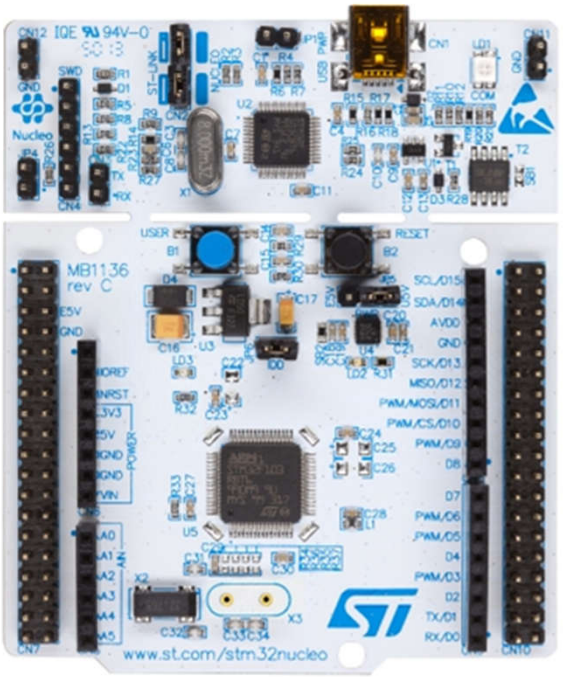
Control de la posición/velocidad angular de un disco, controlando el voltaje del motor.

- Motor DC
- Encoder rotatorio
- Placa Nucleo stm32, símil Arduino



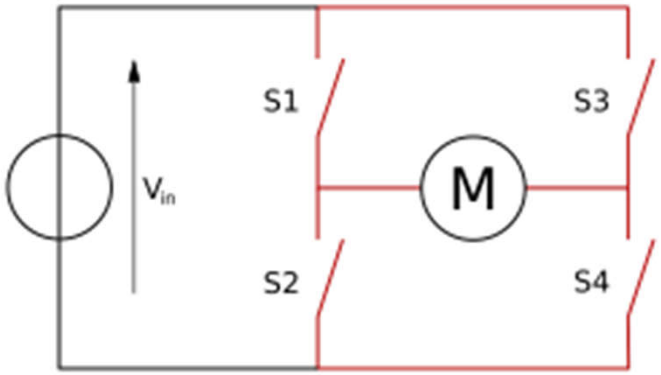
Circuito placa-motor-encoder

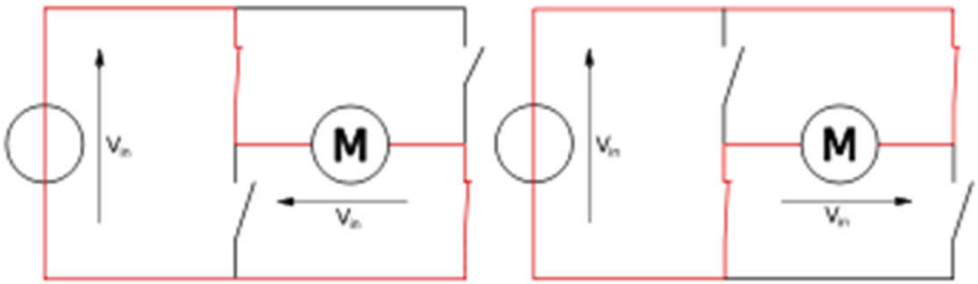




Puente H

Permite hacer andar un motor de corriente continua en ambas direcciones, con una sola fuente de tensión.





Comunicación entre Python y la placa Nucleo

```
In [2]: ser = serial.Serial(port='COM6', baudrate=115200, bytesize=8, parity='N',
stopbits=1, timeout=0.005, xonxoff=0, rtscts=0)
   ...: ser.close()
   ...: ser.open()
```

Inicialización comunicación serie PC-placa

```
   ...: #reset controlador
   ...: ser.write(bytes('X','utf-8'))
   ...: time.sleep(0.01)
   ...: ser.flushInput()
```

Reset placa, pone en cero posición y voltaje

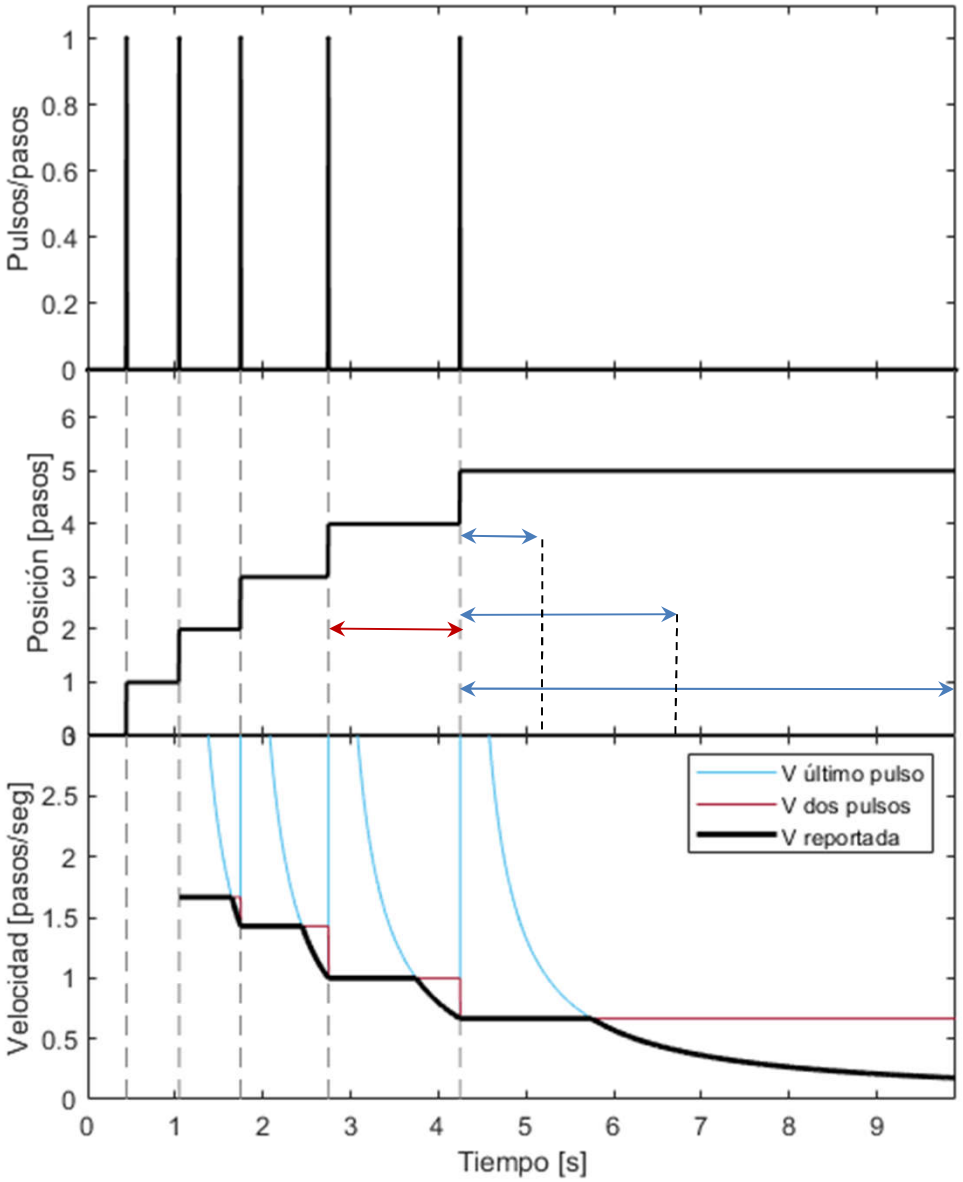
```
   ...: #escribo voltaje, pregunto posicion y velocidad
   ...: str = 'V0\n\r'
   ...: ser.write(bytes(str,'utf-8'))
   ...: time.sleep(0.002)
   ...: s = ser.readline(25)
   ...: print(s)
b'000000000,+0.000000e+00\r\n'
```

Fija voltaje [-12|12], y pregunta posición y velocidad

posición en pasos (una vuelta 256 o 512 pasos)

velocidad en vueltas por segundo

Cómo mide posición y velocidad



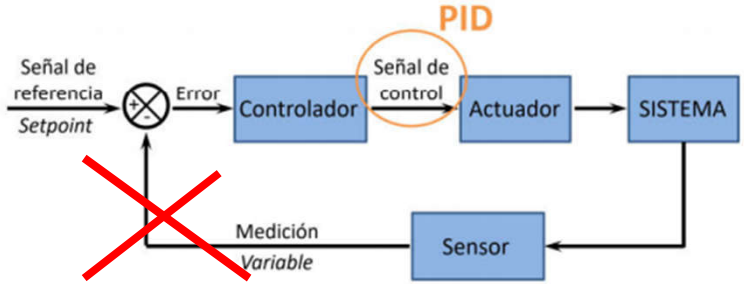
Al girar la rueda, el encoder manda pulsos (negativos si gira para el otro lado)

La placa integra (suma) los pulsos para determinar la posición

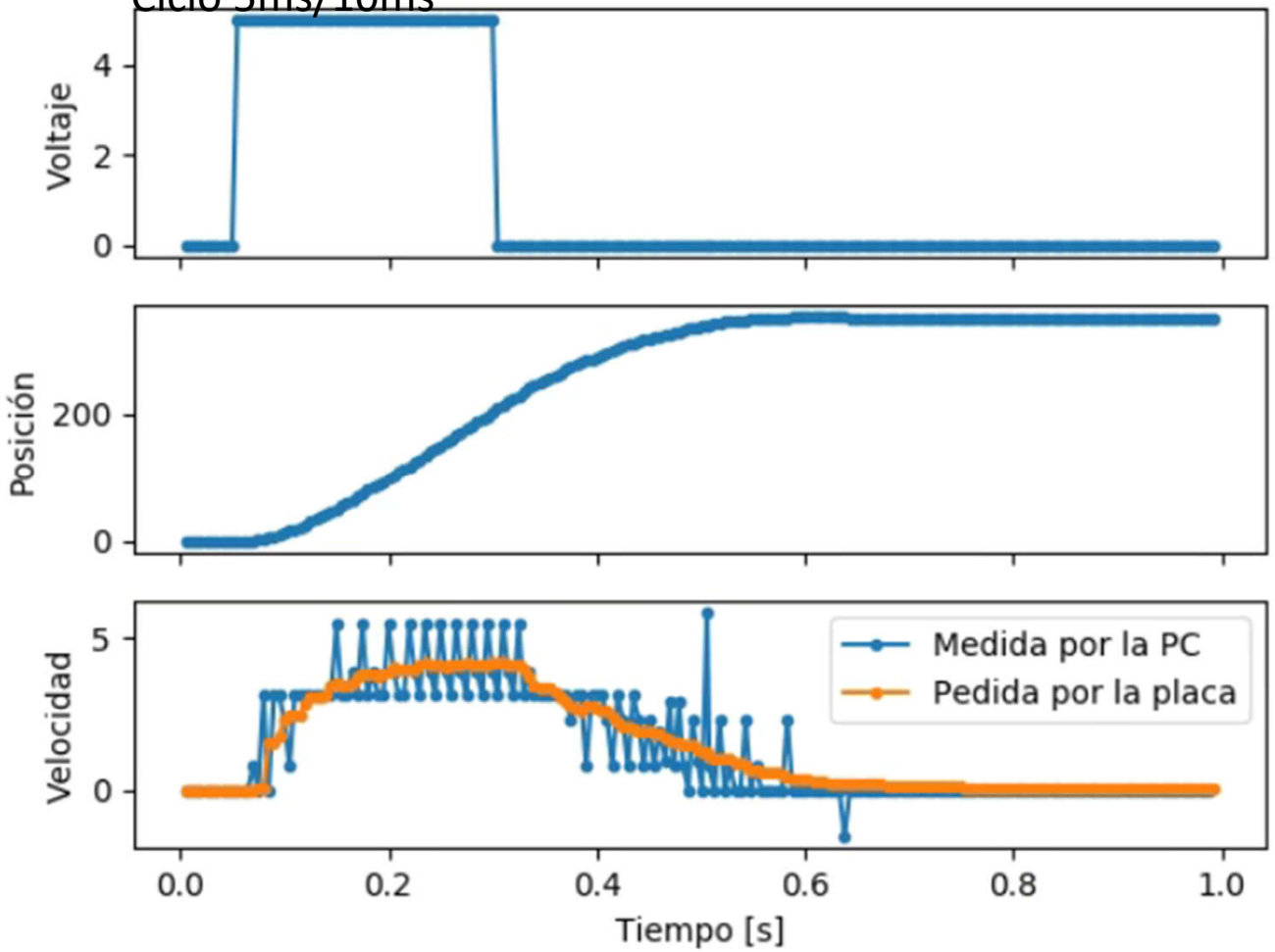
V1 a partir de los últimos dos pulsos
V2 a partir del tiempo desde el último pulso

La placa calcula todo esto y reporta
min(V1,V2)

Lazo abierto

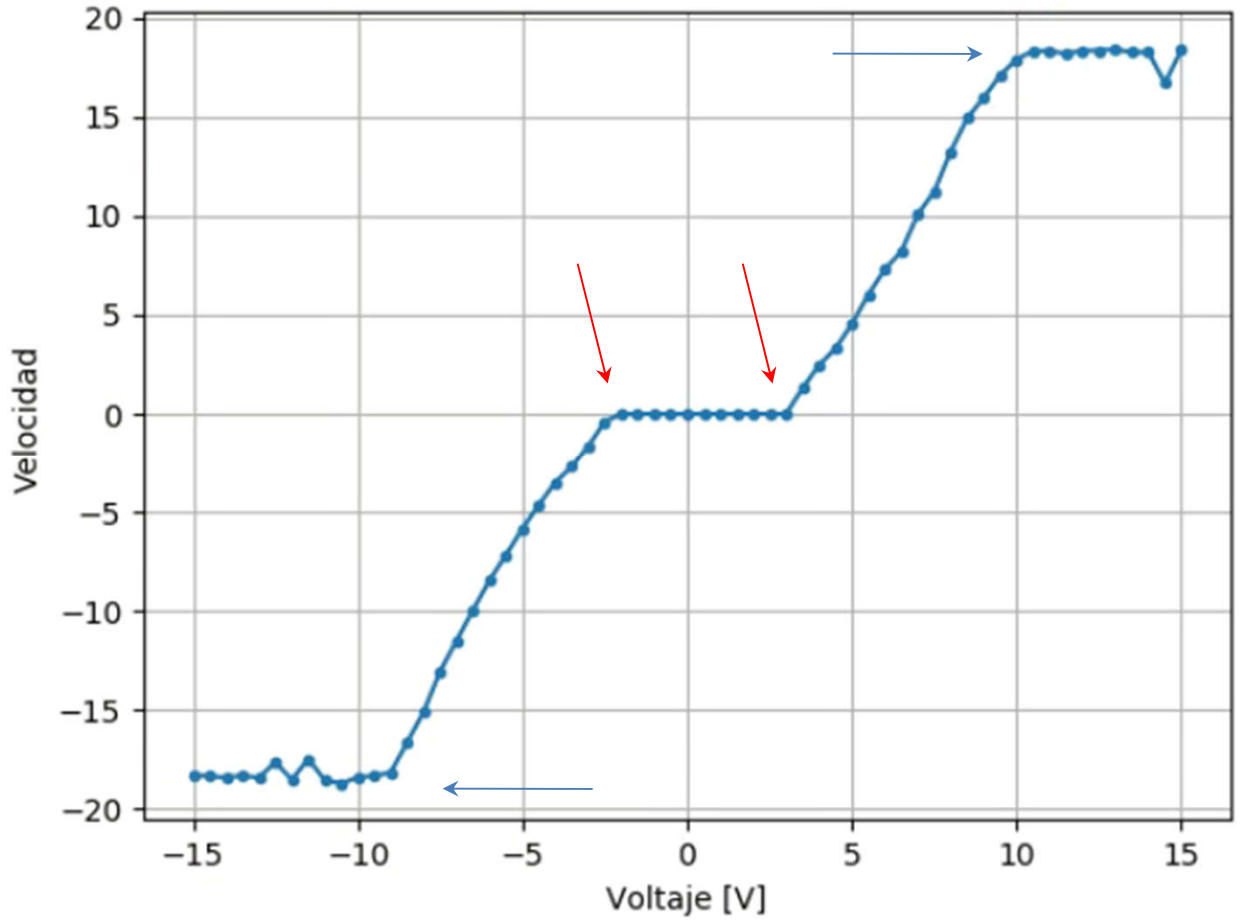


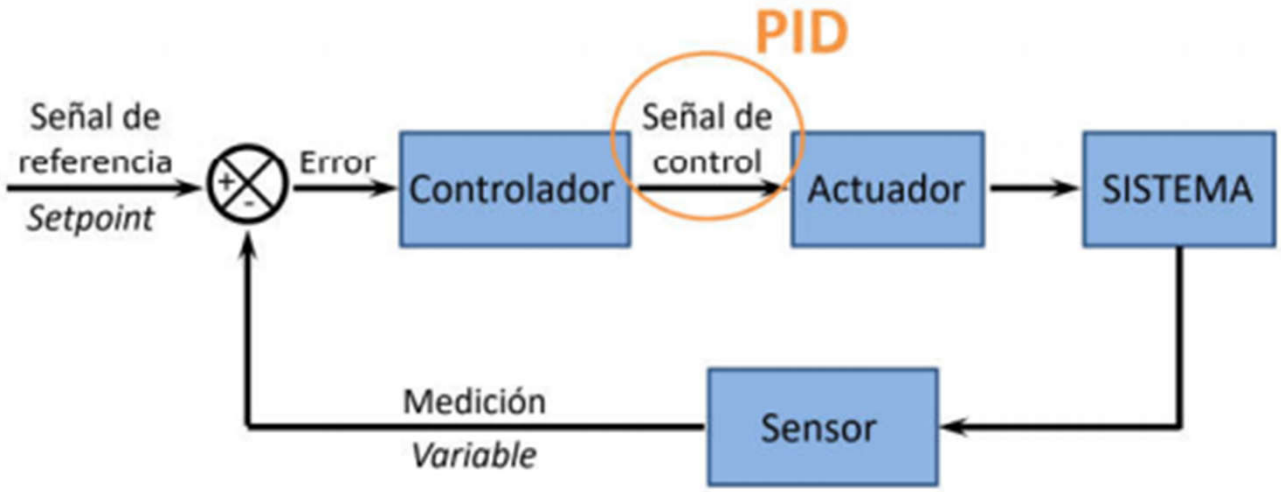
- loop: Pongo voltaje, mido pos y vel
- Ciclo 5ms/10ms



Relación no lineal entre voltaje y velocidad

- Saturación
- Voltaje mínimo para moverse
- -Voltaje corregido/linealizado?



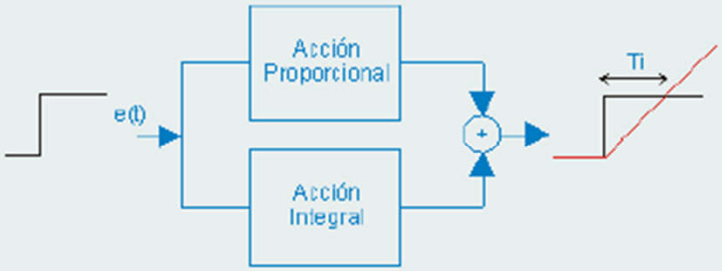


https://github.com/diegoshalom/labosdf/tree/master/software/python/Labo5%202021/PID

Controlador PID. Continuo.

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} \right]$$

$$\begin{cases} K_i = \dfrac{K_p}{T_i} \\ K_d = K_p T_d \end{cases}$$



Constante de tiempo de integración : Ti

Tiempo requerido para que la acción integral iguale el valor de la acción proporcional

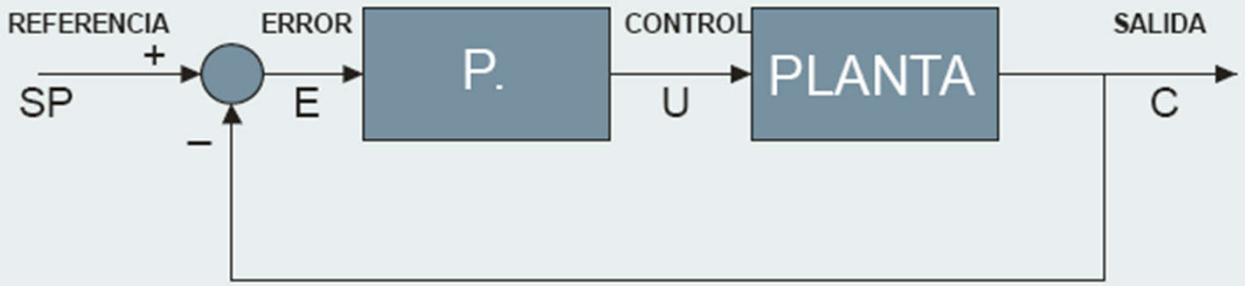
Constante de tiempo de derivación : Td

Tiempo requerido para que la acción derivativa iguale el valor de la acción proporcional

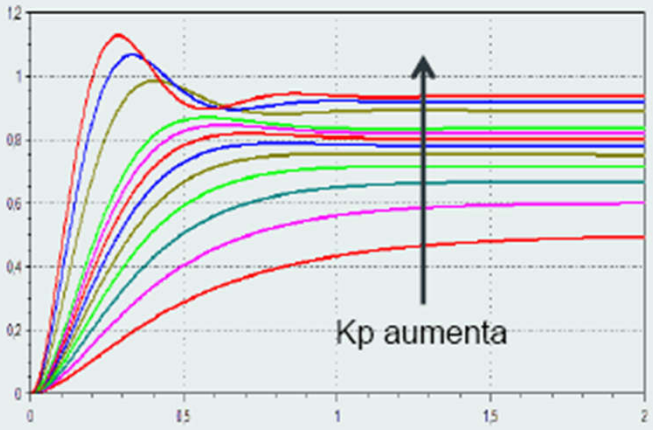
Teoría de Control

Extraído de https://www.academia.edu/34786869/TEOR%C3%8DA_DE_CONTROL

Controlador P.

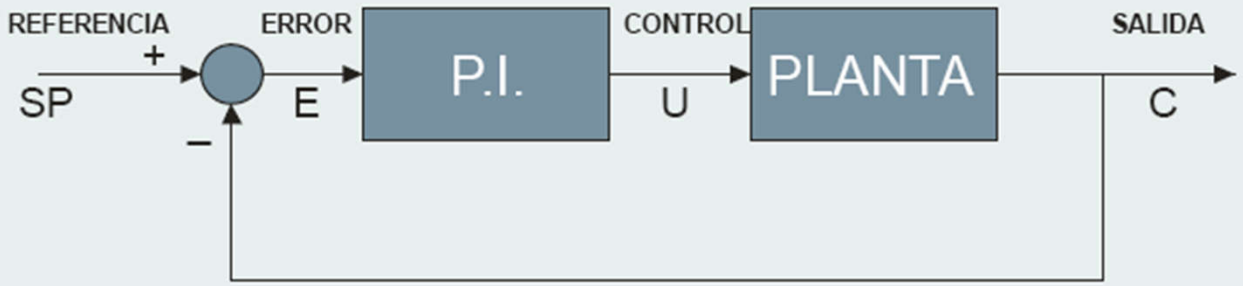


Respuesta a una entrada escalón unitario

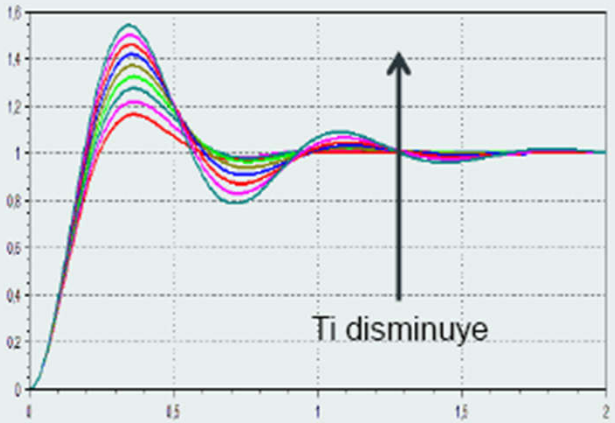


- El error en régimen permanente disminuye
- La velocidad de respuesta aumenta
- El sobrepico aumenta

Controlador P.I.

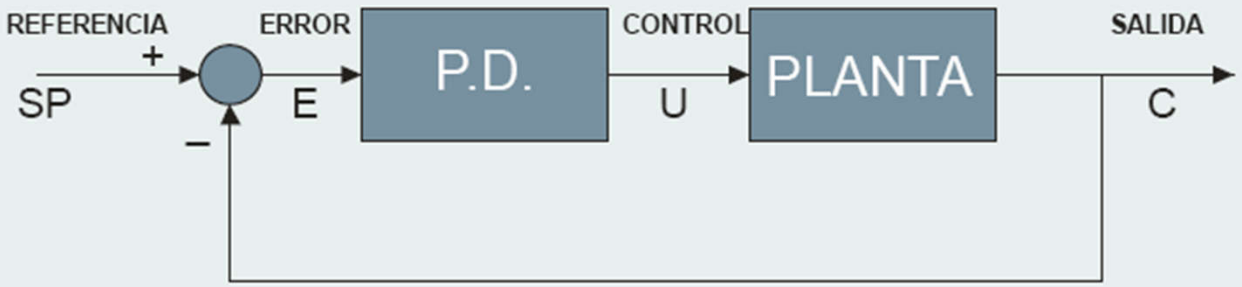


Respuesta a una entrada escalón unitario

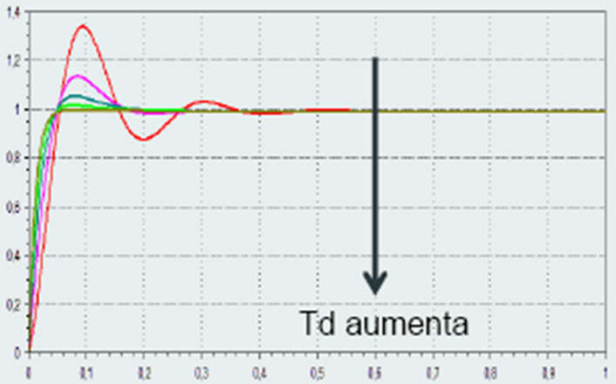


- El error en régimen permanente se elimina
- La estabilidad empeora
- El sobrepico aumenta

Controlador P.D.



Respuesta a una entrada escalón unitario



- La estabilidad mejora
- El sobrepico disminuye
- La velocidad de respuesta aumenta

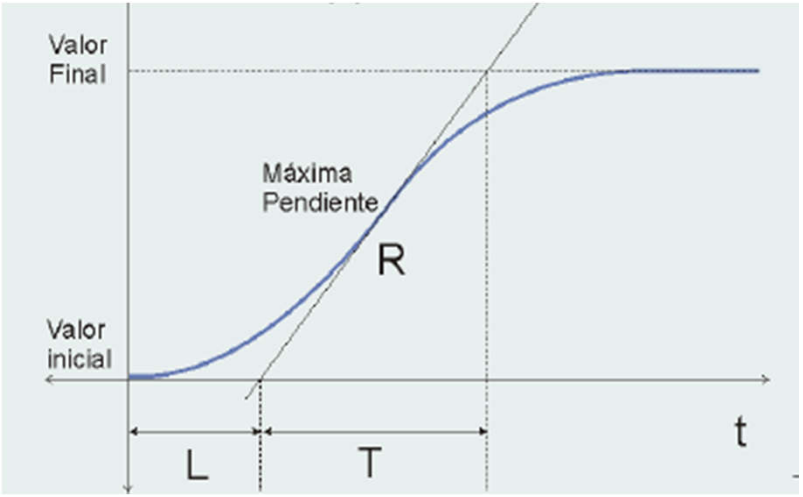
Métodos de sintonización

ZIEGLER-NICHOLS a lazo abierto

Con L y R , se obtienen los parámetros del controlador PID utilizando la tabla

TIPO DE CONTROLADOR	Kp	Ti	Td
P	$\frac{1}{R \cdot L}$		
PI	$\frac{0,9}{R \cdot L}$	$3L$	
PID	$\frac{1,2}{R \cdot L}$	$2L$	$0,5L$

Respuesta a una entrada escalón

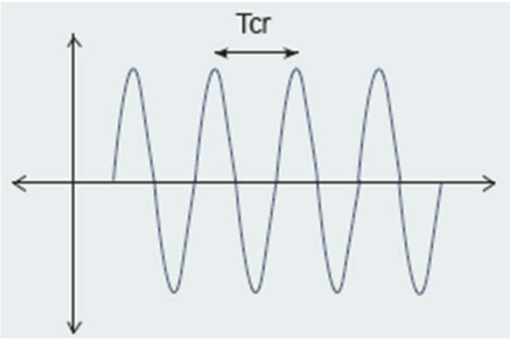


Métodos de sintonización

ZIEGLER-NICHOLS a lazo cerrado

Con Kcr y Tcr , se obtienen los parámetros del controlador PID utilizando la tabla 2.

TIPO DE CONTROLADOR	Kp	Ti	Td
P	$0,5Kcr$		
PI	$0,45Kcr$	$\frac{Tcr}{1,2}$	
PID	$0,6Kcr$	$\frac{Tcr}{2}$	$\frac{Tcr}{8}$



-Sólo término P prendido

-El mínimo K_p que da oscilaciones es el K_{cr}

Actividades para hoy

1- Estandarizar la expresión de PID

$$u(t) = K_p\left[e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{de(t)}{dt}\right]$$

-Incluir Δt

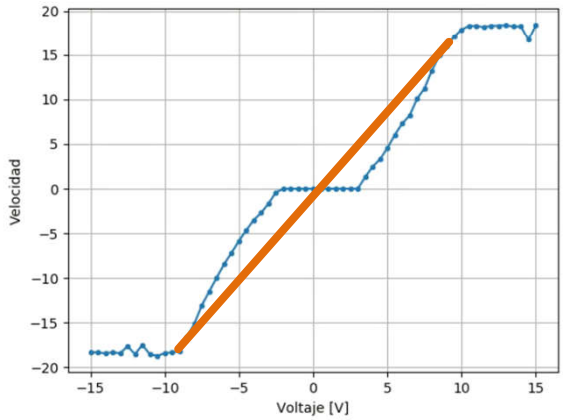
-Mirar/medir/arreglar unidades de velocidad, tal $\frac{dp}{dt} = v$

2- Arreglar término derivativo

$$e(t) = SP(t) - PV(t)$$

$$\begin{aligned}
\frac{de}{dt} &\approx \frac{e(t+\Delta t) - e(t)}{\Delta t} \\
&\approx \frac{SP(t+\Delta t) - PV(t+\Delta t) - SP(t) + PV(t)}{\Delta t} \\
&\approx \frac{SP(t+\Delta t) - SP(t)}{\Delta t} - \frac{PV(t+\Delta t) - PV(t)}{\Delta t} \\
&\approx \frac{d(SP)}{dt} - \frac{d(PV)}{dt} \\
&\approx \frac{d(SP)}{dt} - vel(t)
\end{aligned}$$

3- Implementar voltaje corregido



Algunas posibles actividades

- Posición vs tiempo para:
  - Controlador P para varios valores de Kp
  - Controlador PI para varios valores de Ki (Kp fijo)
  - Controlador PD para varios valores de Kd (Kp fijo)

- Implementar Zielger-Nichols a lazo abierto, o a lazo cerrado.

- Setpoint **fijo** ±[100; 300; 1000; 3000; 10000].
  - 1- Que llegue rápido, aunque haya un poco de oscilaciones.
  - 2- Que llegue exactamente a SP, no importa cuán rápido, sin overshoot ni oscilaciones.

- Setpoint **variable linealmente**, SP(t)=vel*t, con vel=[10; 100; 1000; 10000]
  - ¿Controlando posición o controlando velocidad?
  - (Ojo término D)

- Setpoint **variable sinusoidal**, SP(t)=A*sin(w*t),
  A=[100;1000;10000],
  w = 2*pi*[0.02; 0.1; 0.3; 1; 3]
  - (Ojo término D)

Entregar un documento con las figuras de las condiciones estudiadas, indicando los parámetros PID utilizados en cada una. Pueden escribir una breve explicación de cada una (no más de 2 renglones).