Fundamentos en Robótica

Unidad 3.1

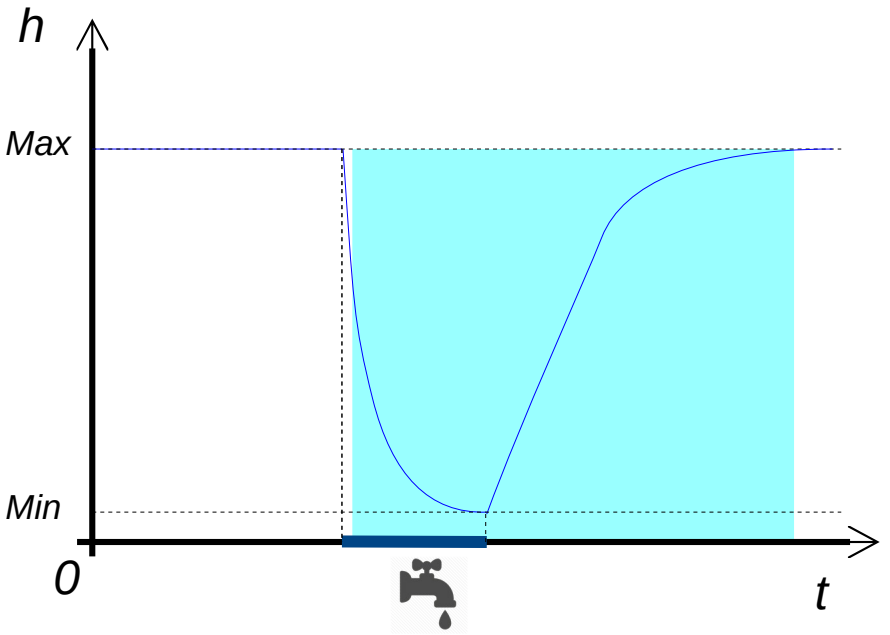
Teoría de Control

mina
network management | artificial intelligence

Temario

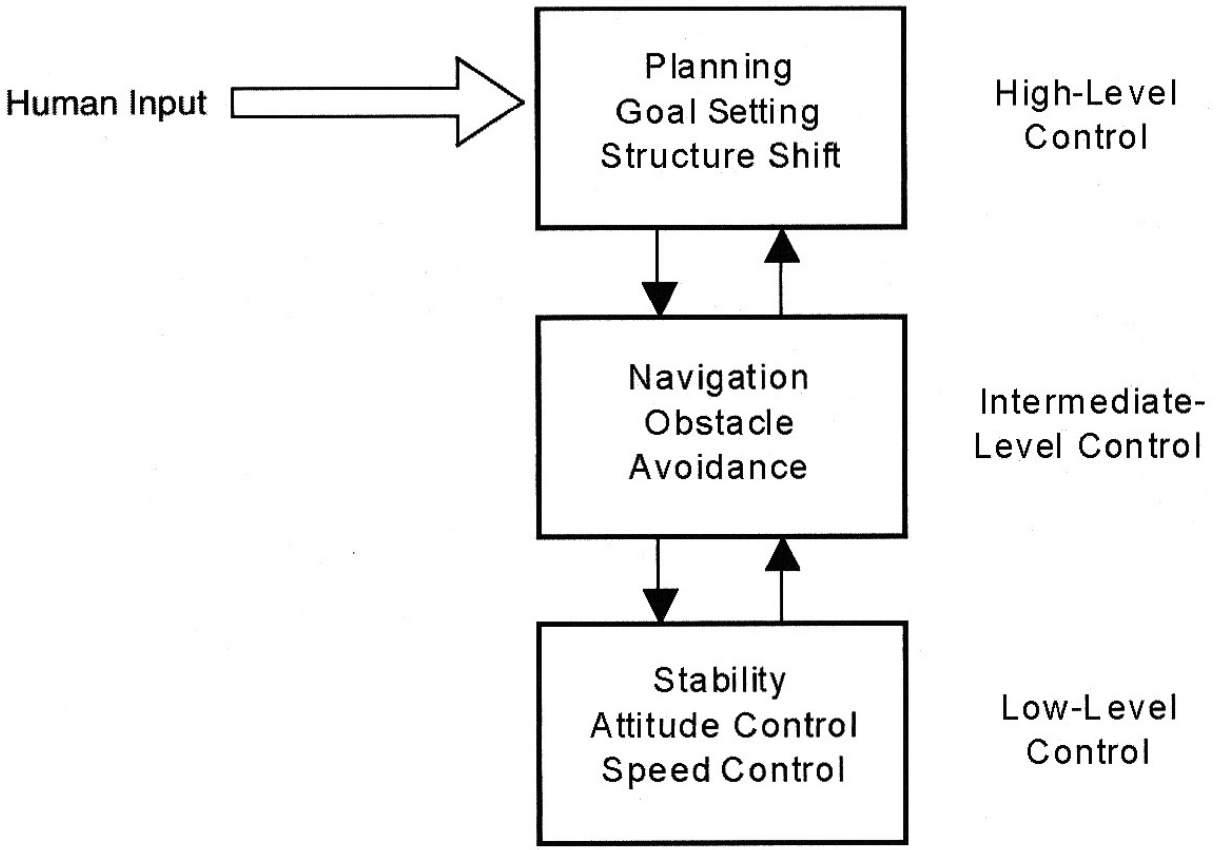
- Introducción al control
- Control lineal y PID
- Control no lineal

Sistemas de control



Sistemas de control

Niveles de control



Modelos del sistema

- Lineal vs no lineal
  Principio de superposición: en un sistema lineal la respuesta a la suma de dos estímulos es la suma de las respuestas a cada uno.

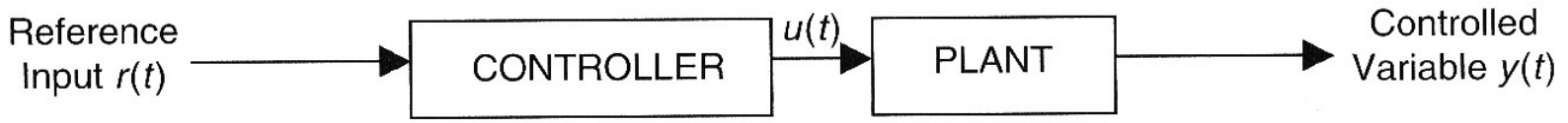
- Determinista vs estocástico

- Contínuo vs discreto
  - $t \in R$
  - $i \in Z$

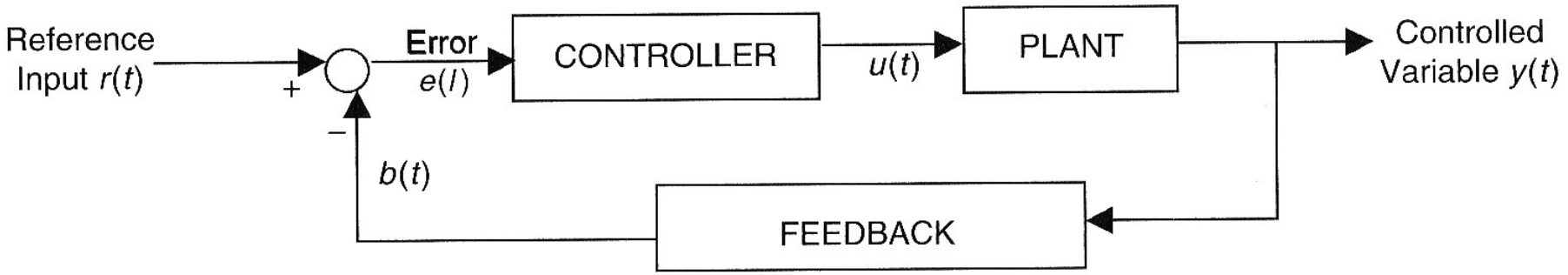
Objetivos del sistema

- Regulación
  - Termostato
  - Control de carga
- Seguimiento (*tracking*)
  - Control de movimientos, trayectorias
- Optimización
  - Búsqueda de soluciones que minimicen uso de recursos

Bucles de control

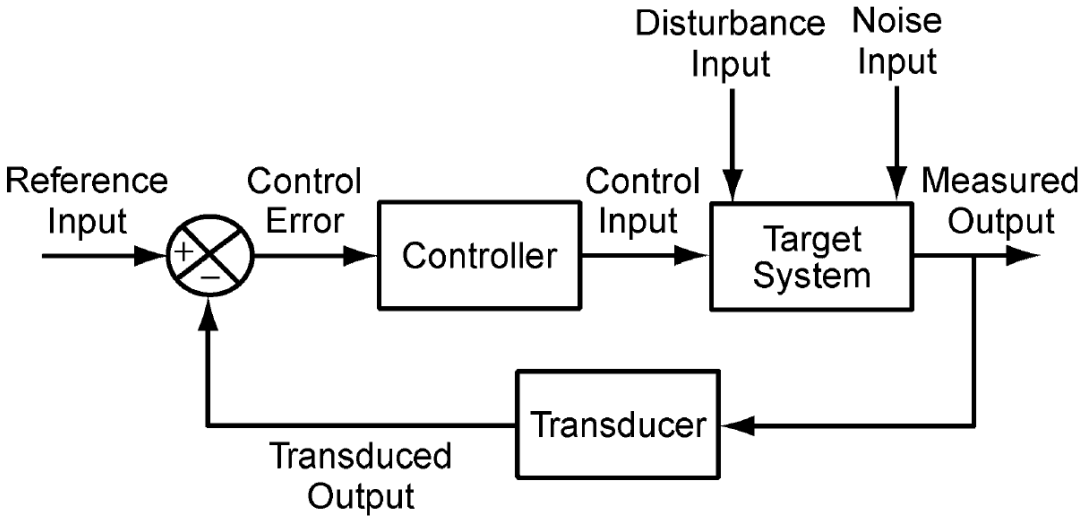


Control de bucle abierto (*feedforward*)



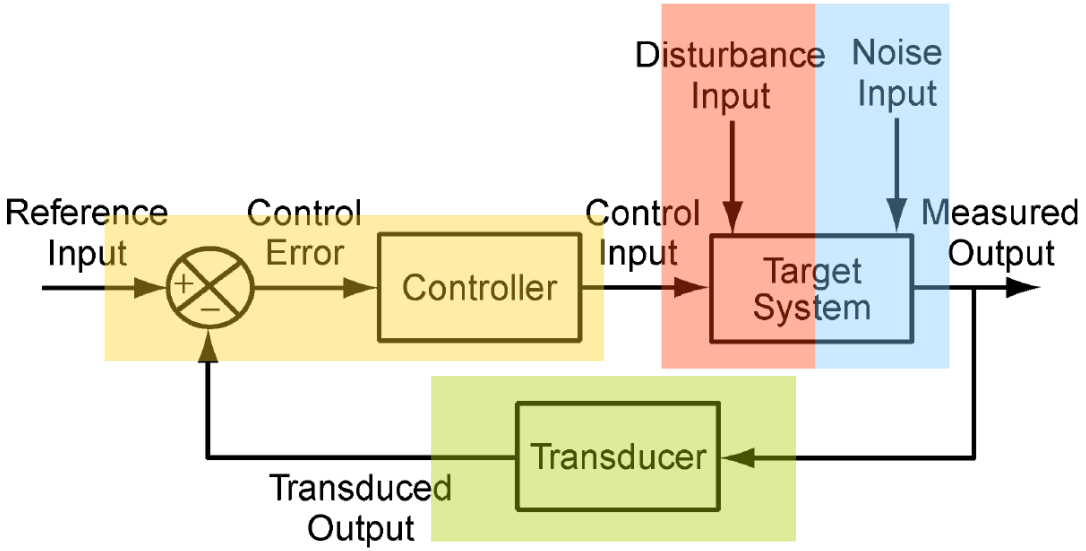
Control de bucle cerrado (*feedback*)

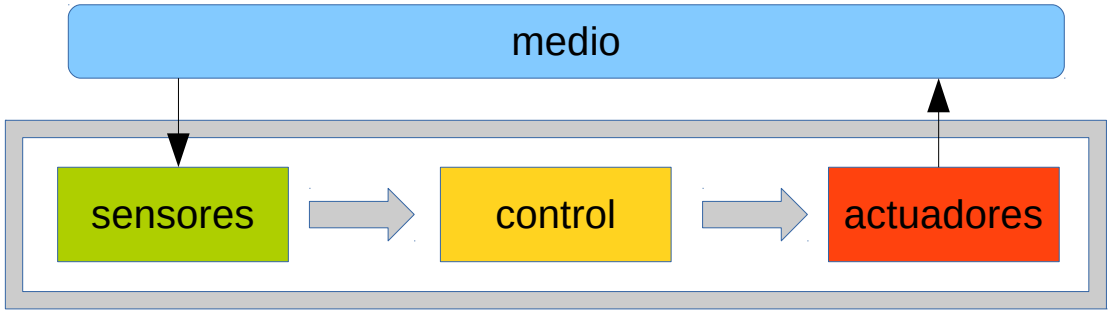
Sistema de control



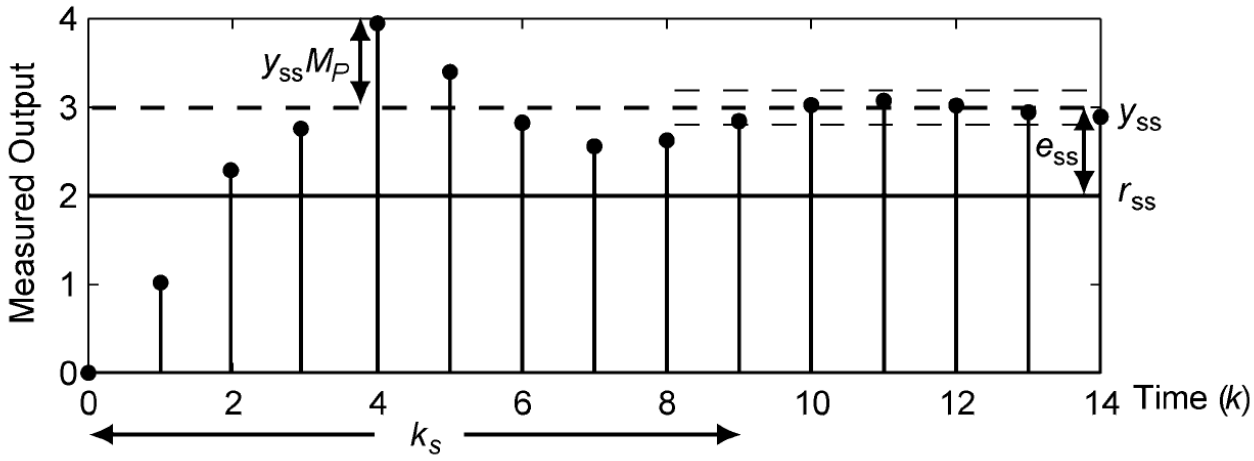
- *Entrada de referencia*: objetivo del sistema
- *Controlador*: computa acciones
- *Entrada de control*: acciones
- *Objetivo (Planta)*: sistema que estamos controlando
- *Salida*: resultados de la actuación
- *Retroalimentación*: observación de los resultados

Sistemas de control



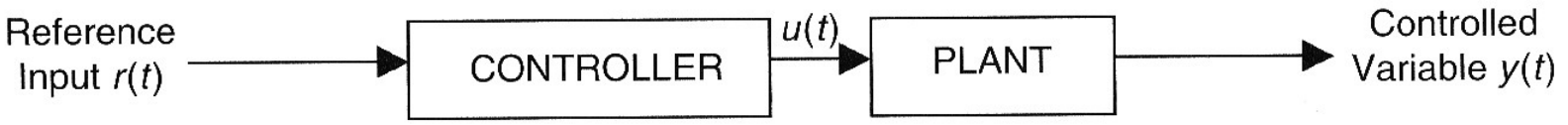


Características



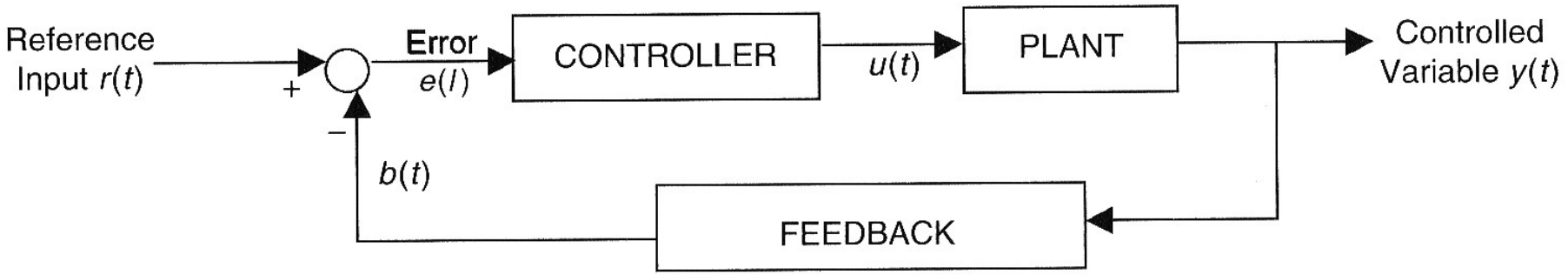
- r_{SS} objetivo del sistema
- y_{SS} valor al que converge el controlador
- e_{SS} error estable
- k_S tiempo de convergencia
- M_P *overshoot* máximo

Bucle abierto



$$u(t) = K_c r(t)$$
$$y(t) = K_p u(t)$$
$$y(t) = K_c K_p r(t) = K r(t)$$

Bucle cerrado



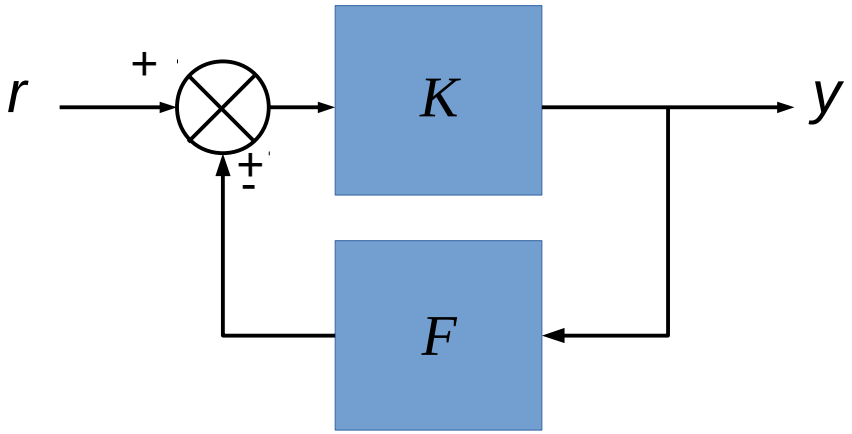
$$u(t) = K_C e(t) = K_C [r(t) - b(t)]$$

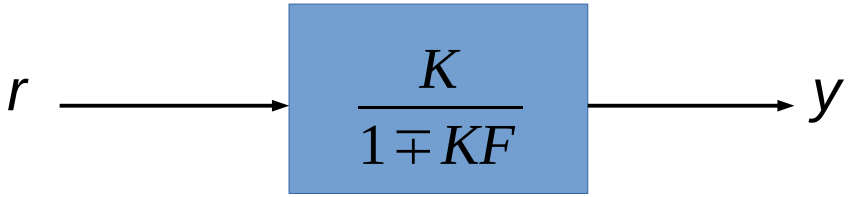
$$b(t) = K_F y(t)$$

$$y(t) = K_P K_C [r(t) - K_F y(t)]$$

$$y(t) = \frac{K_P K_C}{1 + K_P K_C K_F} r(t) \approx (1/K_F) r(t) \quad si\ K_C\ grande$$

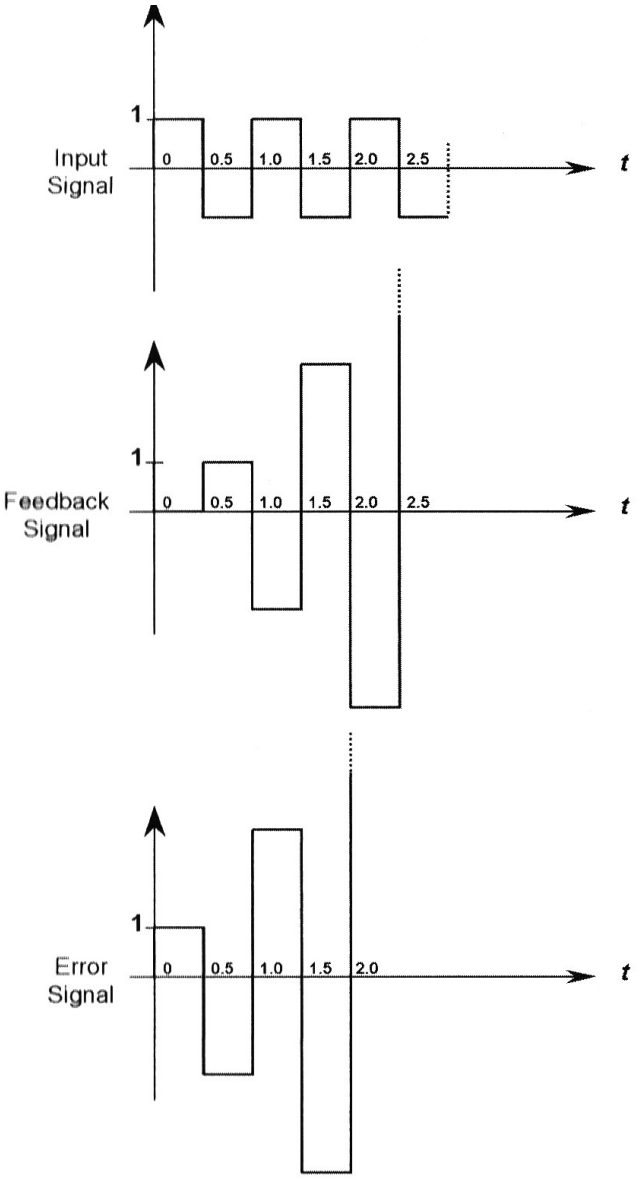
Bucle cerrado



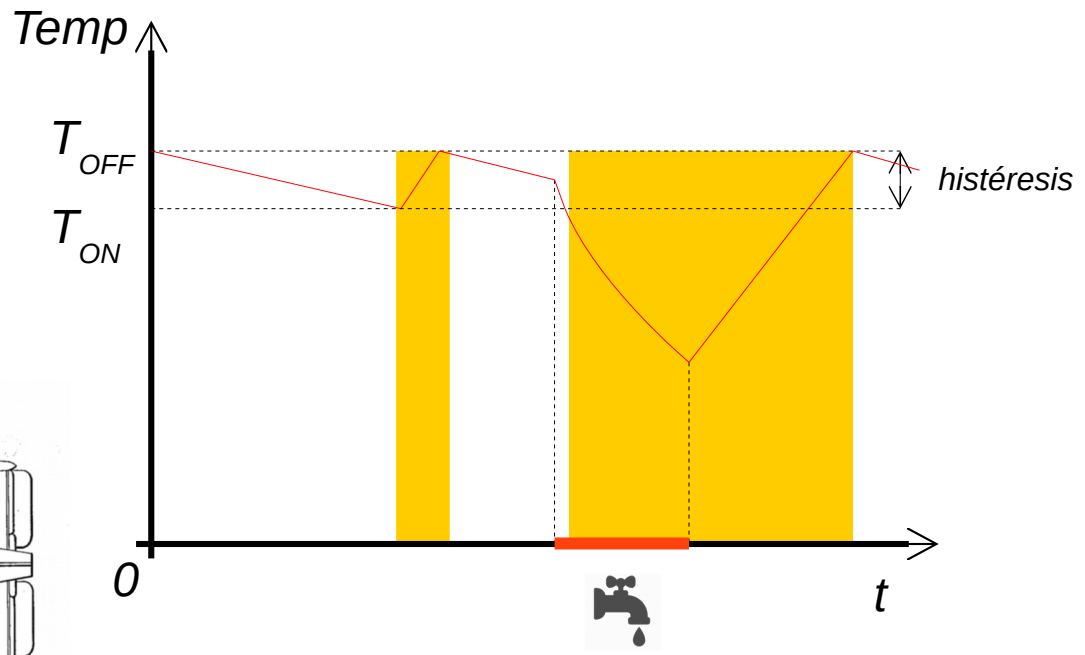


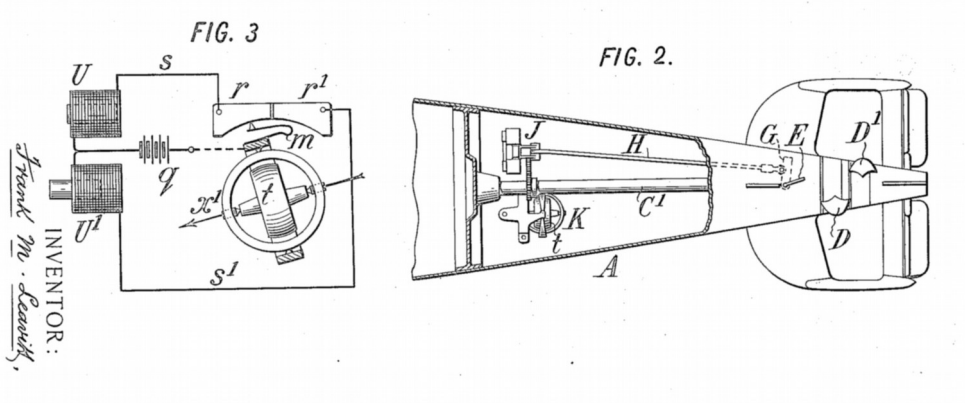
Estabilidad

- Para todo controlador hay una entrada que lo hace inestable (*in the loop*)
- El sistema de control debe ser mucho más rápido que el sistema controlado
- Balance control agresivo – conservador
- La latencia es un problema



Control on-off

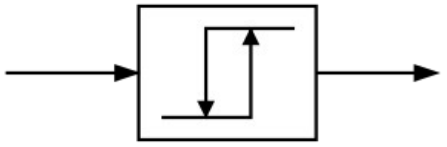




Histéresis

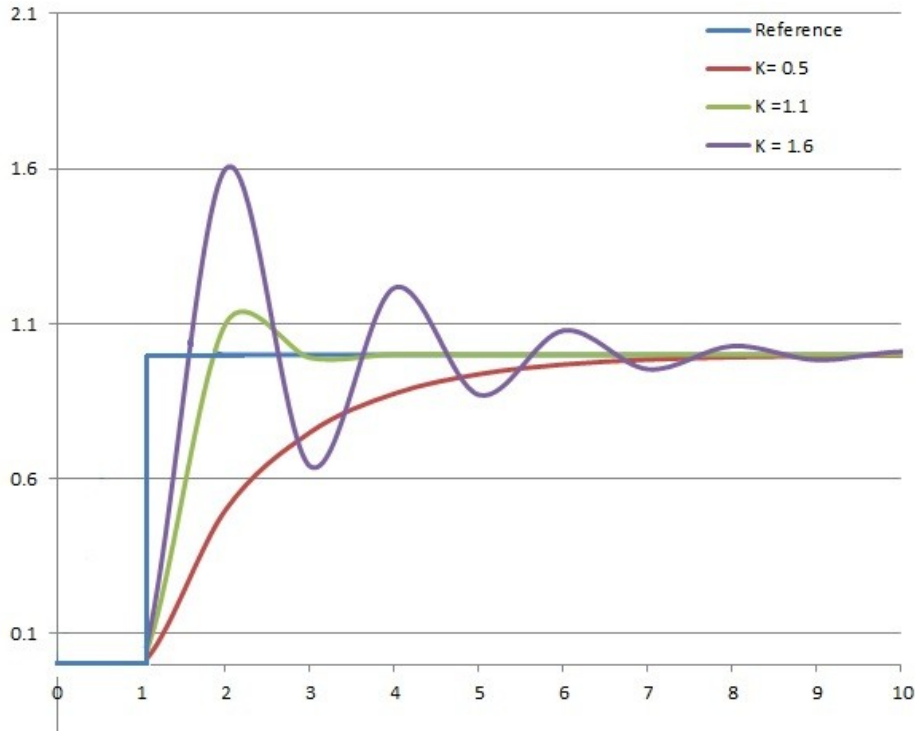
Estado depende de la historia → sigue caminos distintos según de donde viene.

$$u(t)=\begin{cases} K_c \; si \; v<v_{ON} \\ 0 \; si \; v>v_{OFF} \\ u(t-dt) \; si \; no \end{cases} \qquad v_{ON}<v_{OFF}$$



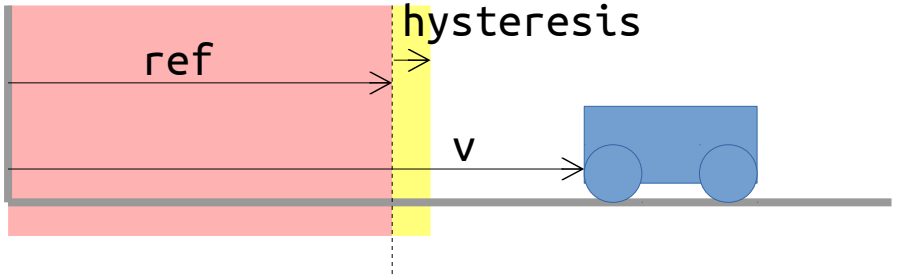
Controlador Proporcional

$$u(t) = K_p e(t)$$



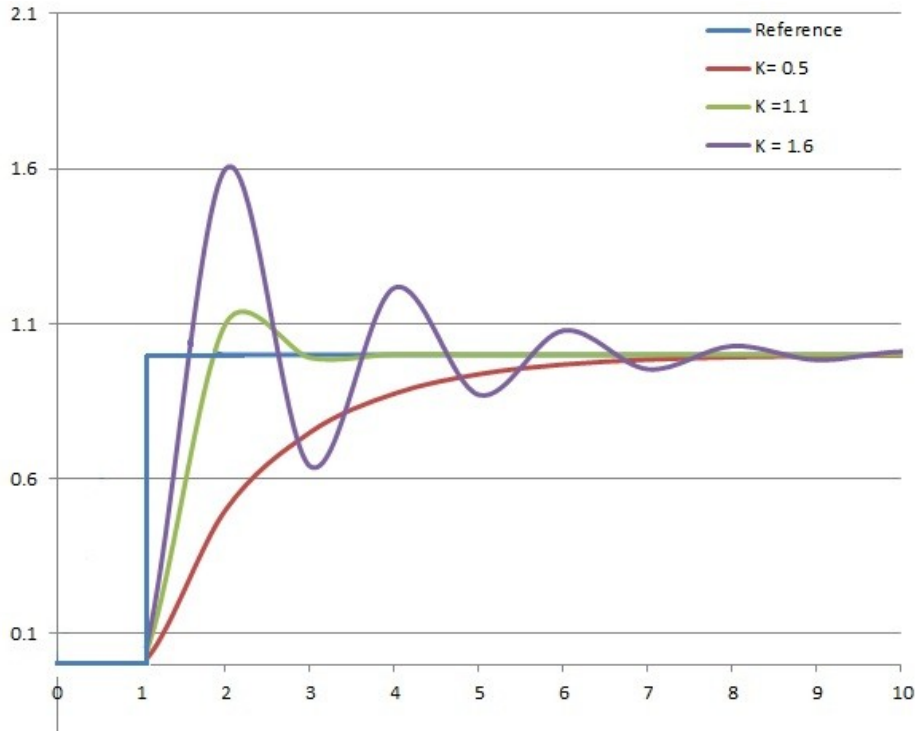
Histéresis

```
function too_close_bad(v, ref)
  return v<ref
end

local now_too_close = false

function too_close(v, ref, hysteresis)
  if now_too_close then
    if v>ref+hysteresis then
      now_out_range = false
    end
  else
    if v<ref then
      now_too_close = true
    end
  end
  return now_too_close
end
```



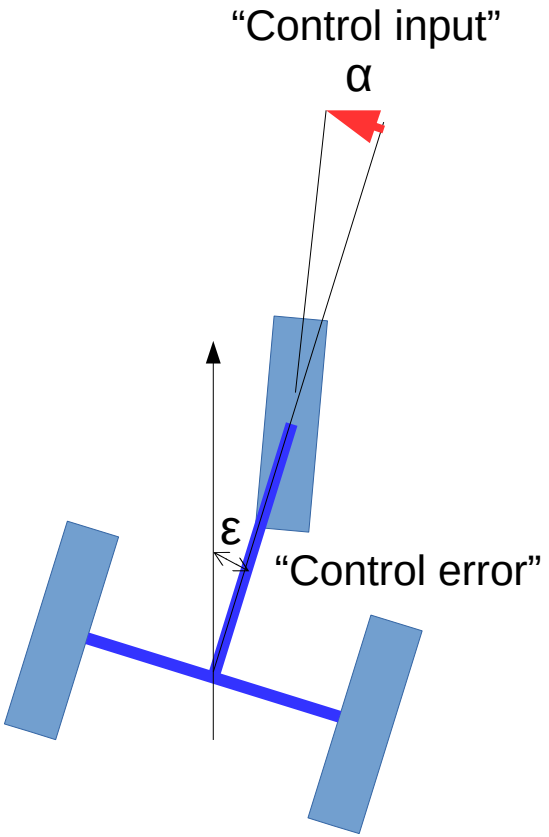
Controlador Proporcional

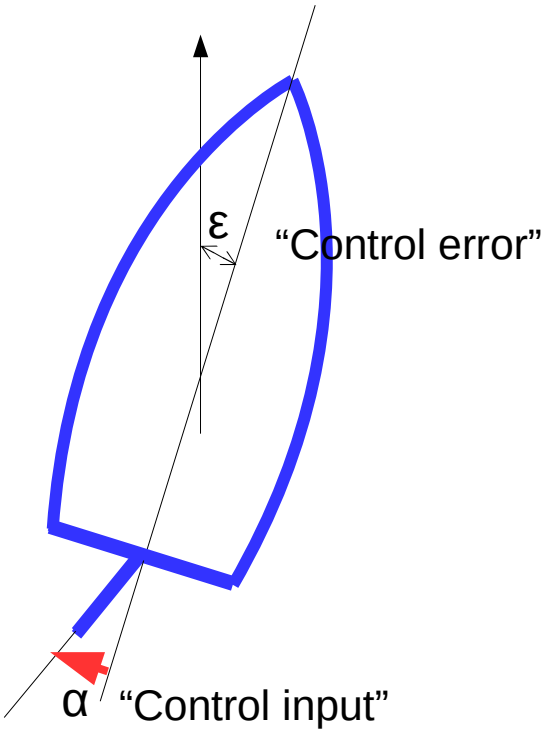
- Similar a un modelo masa-resorte
- K_p controla tiempo de convergencia y oscilaciones



Controlador proporcional

Ejemplo: apuntar a una dirección

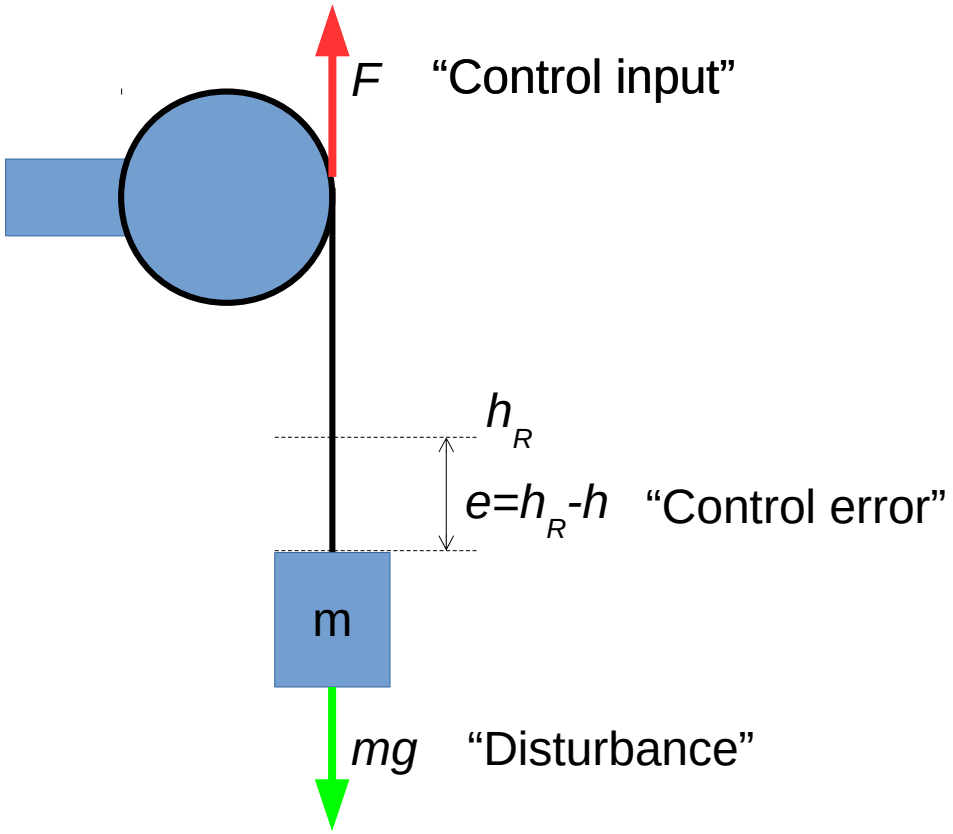




$$\alpha(t) = K_p \, \varepsilon(t)$$

Controlador proporcional

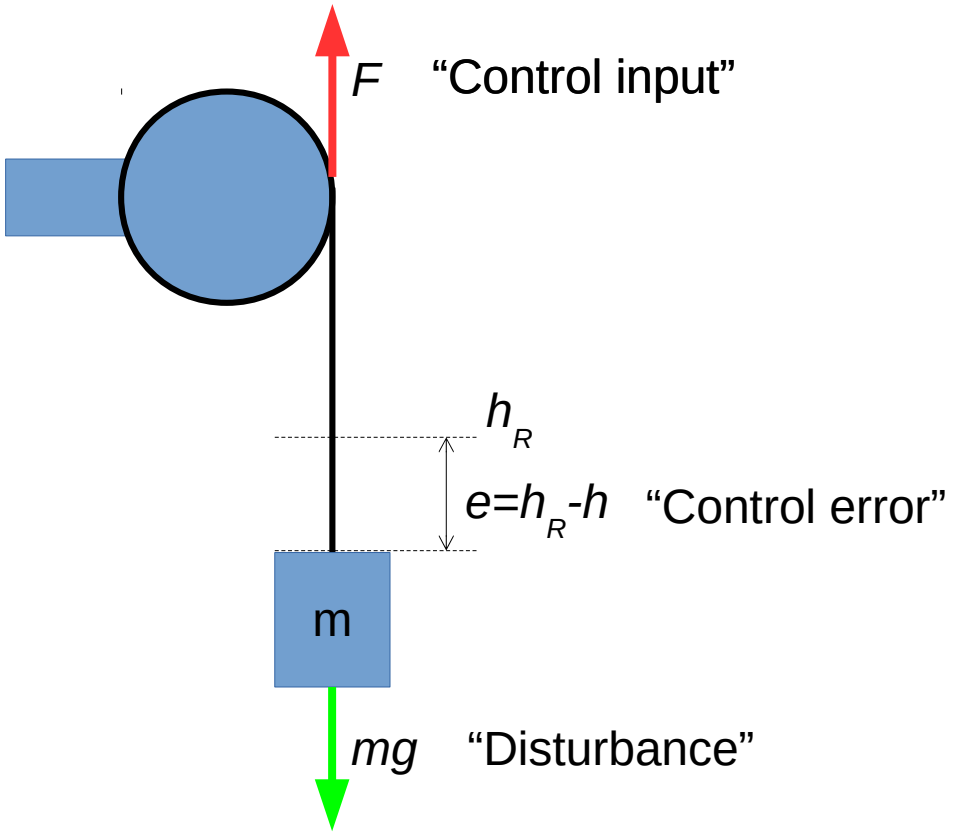
Ejemplo: mantener la carga a una altura



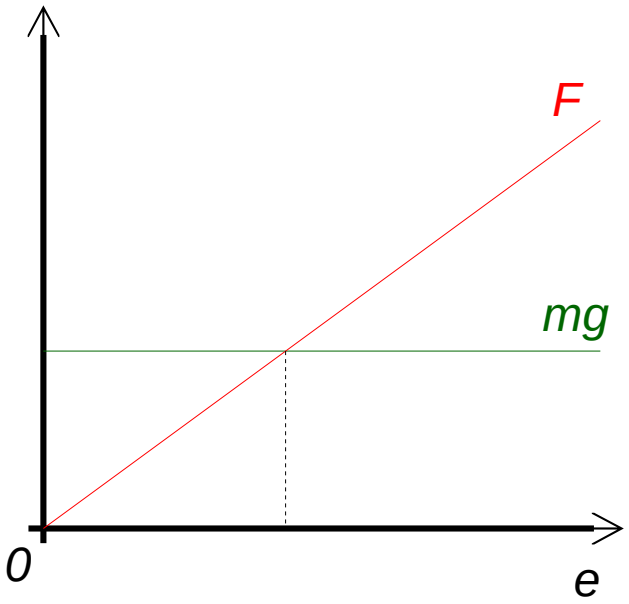
$$F(t) = K_p\, e(t)$$

Controlador proporcional

Ejemplo: mantener la carga a una altura



$$F(t) = K_p\, e(t)$$



Controlador Integral

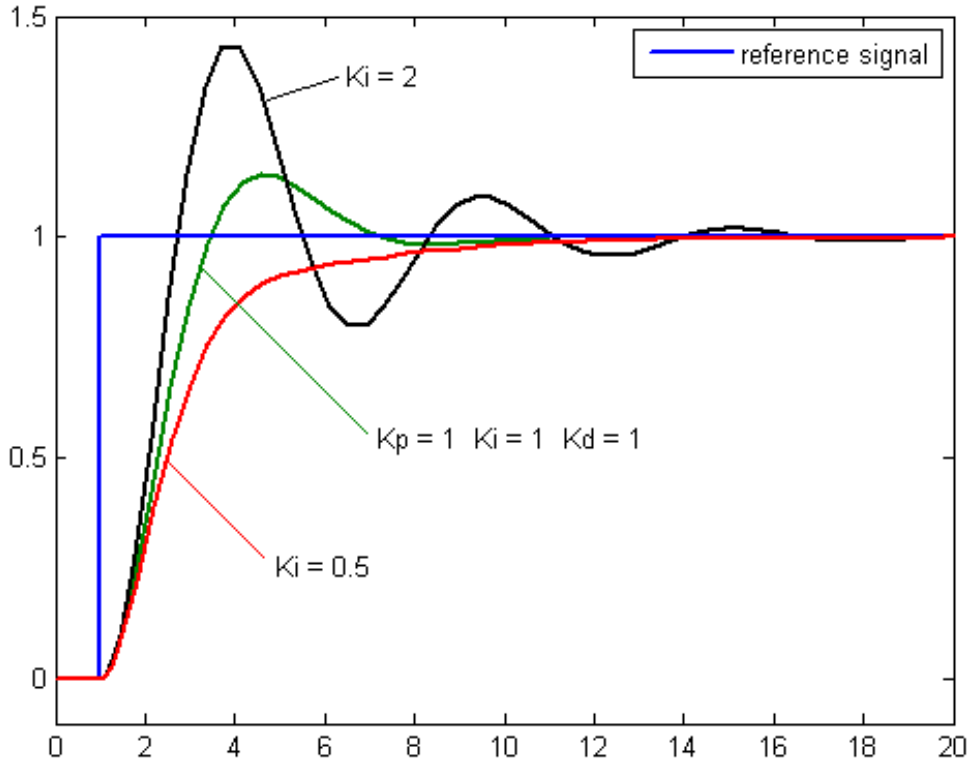
$$u(t) = K_i \int e(t)\,dt$$

- Corrige basado en error acumulado
- Elimina el *steady error*.
- Acelera la convergencia de un controlador P
- Problema de *windup*.
- Controlador PI

$$u(t) = K_p e(t) + K_i \int e(t)\,dt$$

Controlador Integral

$$u(t) = K_i \int e(t)\,dt$$



Controlador Diferencial

$$u(t) = K_d \frac{d}{dt} e(t)$$

- Corrige según tendencia actual
- Acelera respuesta a cambios bruscos
- Modera la sobrecorreción ante cambios lentos
  - Introducido en pilotos automáticos de barcos
- Difícil de usar en sistemas discretos y con ruido
- Controlador PD

$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t)$$

Controlador PID

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{d}{dt} e(t)$$

- Los parámetros *K* se eligen para obtener cierto comportamiento
  - Análisis matemático (Teoría de Control)
  - Simulación
  - Heurísticas y pruebas
- Es común que se omita algún componente: controladores P, PI y PD.

https://www.youtube.com/watch?v=fusr9eTceEo

Controlador PID

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{d}{dt} e(t)$$

Forma equivalente (standard)

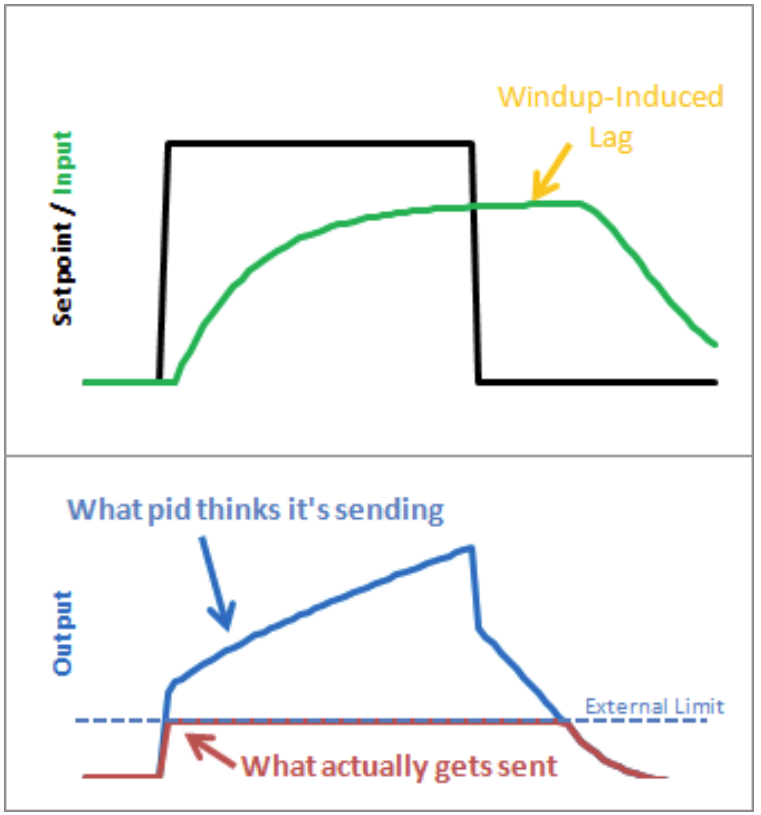
$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{d}{dt} e(t) \right)$$

Es un controlador proporcional (de ganancia K_p) con el error corregido:

- Proyectar tendencia T_d segundos hacia adelante.
- Compensar error acumulado en T_i segundos.

PID: controller windup

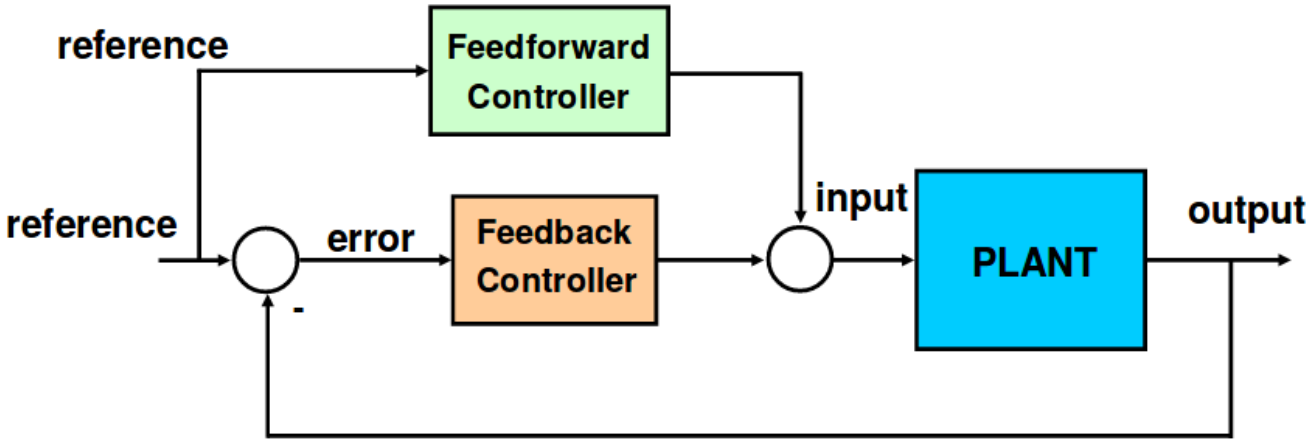
Si por alguna razón el sistema no responde al input, el controlador se puede comportar mal.



http://brettbeauregard.com/blog/2011/04/improving-the-beginner%E2%80%99s-pid-reset-windup/

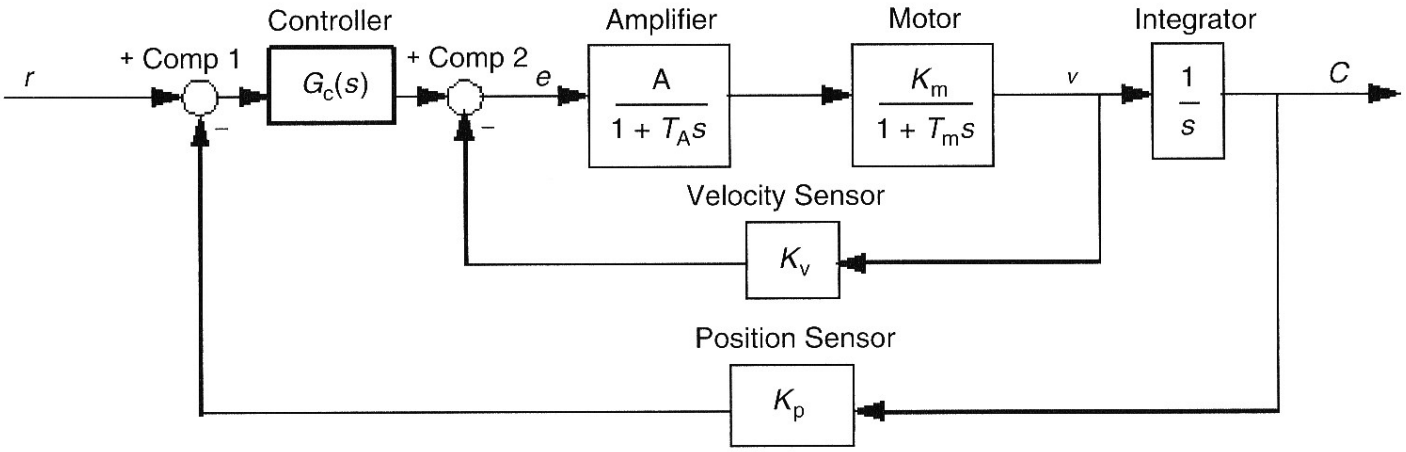
Composición de Controladores

Puedo combinar varios controladores distintos para aprovechar fortaleza de cada uno. Por ejemplo:



Composición de Controladores

La salida de un controlador puede ser el valor de referencia de otro controlador. Ejemplo, servo de posición:



Composición de Controladores

La salida de un controlador puede ser el valor de referencia de otro controlador. Ejemplo, sincronizar dos motores:

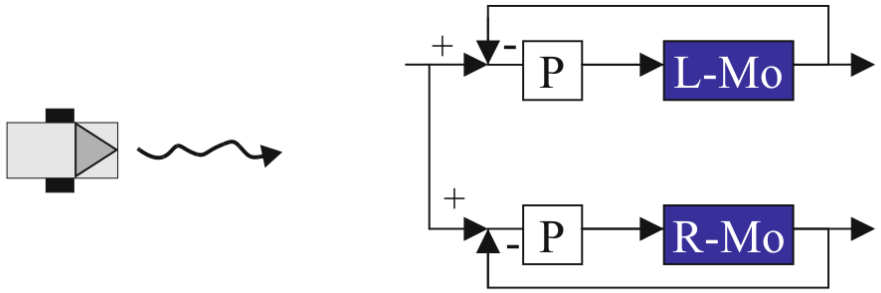


Figure 5.14: Driving straight – first try

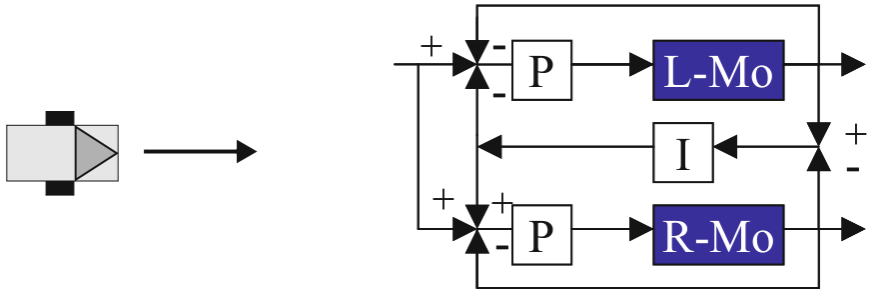


Figure 5.15: Driving straight – second try

Composición de Controladores

La salida de un controlador puede ser el valor de referencia de otro controlador. Ejemplo, sincronizar dos motores:

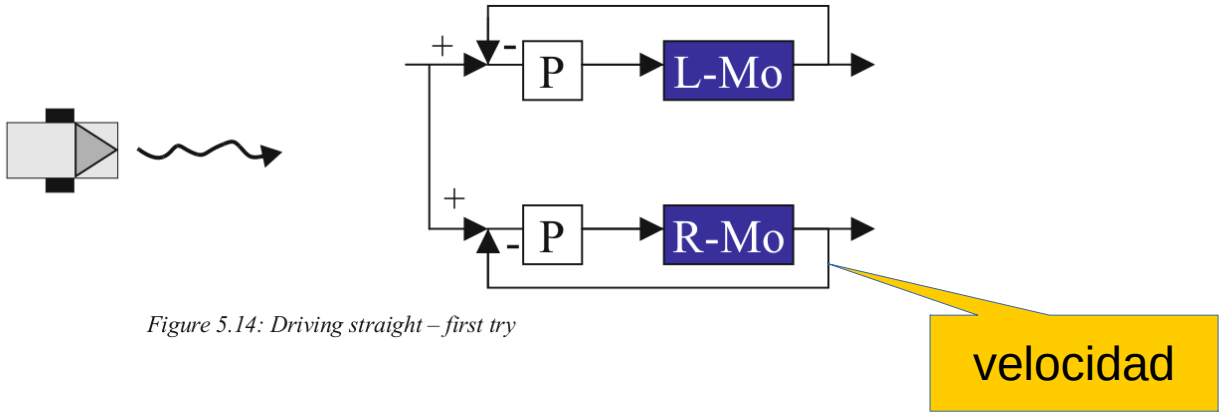


Figure 5.14: Driving straight – first try

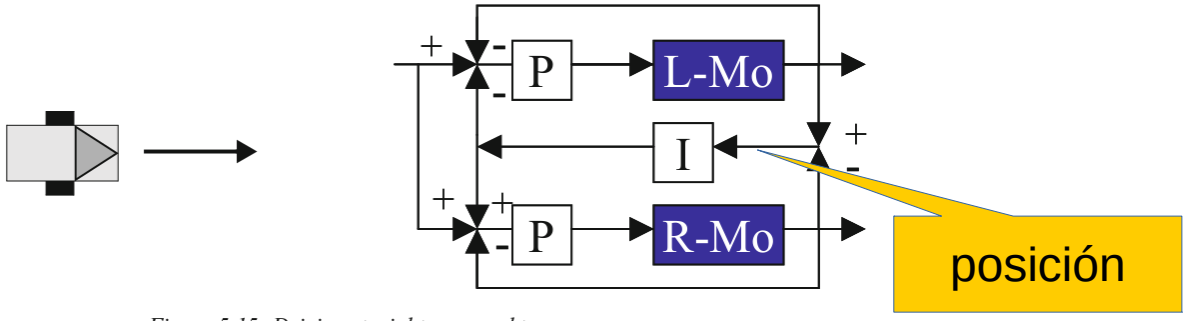


Figure 5.15: Driving straight – second try

Composición de Controladores

Ejemplo: auto autónomo

- Acelerador y frenos
  - Mantener velocidad, distancia... (seleccionar cambios?)
- Volante
  - Dirección asistida
  - Mantenerse en la senda
- Los anteriores, controlados por *Mantenerse en el tráfico*
  - Reaccionar a otros autos, peatones
  - Respetar señales
  - Dirigirse al destino
  - ...

Fin

Referencias

- George A. Bekey, Autonomous Robots, MIT Press - 2005
- Hellerstein J.L., Diao Y., et al, Feedback Control of Computing Systems, Wiley - 2004