UNIVERSIDAD DE LA REPÚBLICA URUGUAY

FACULTAD DE INGENIERIA

IMPI INSTITUTO DE INGENIERIA MECANICA Y PRODUCCION INDUSTRIAL

CONTROL

[Bibliografía: *Introduction to robotics: Mechanics and control* - John Craig]

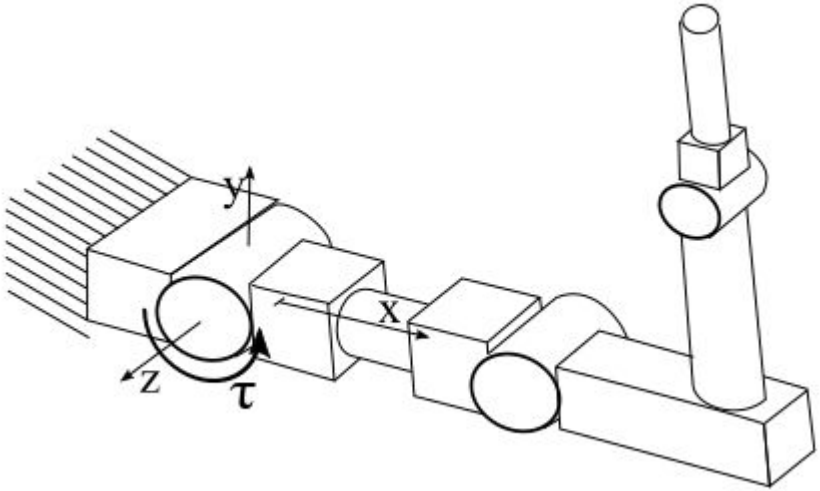
Fundamentos de Robótica Industrial

Versión 2025

Control monoarticular

Consideremos el siguiente ejemplo:

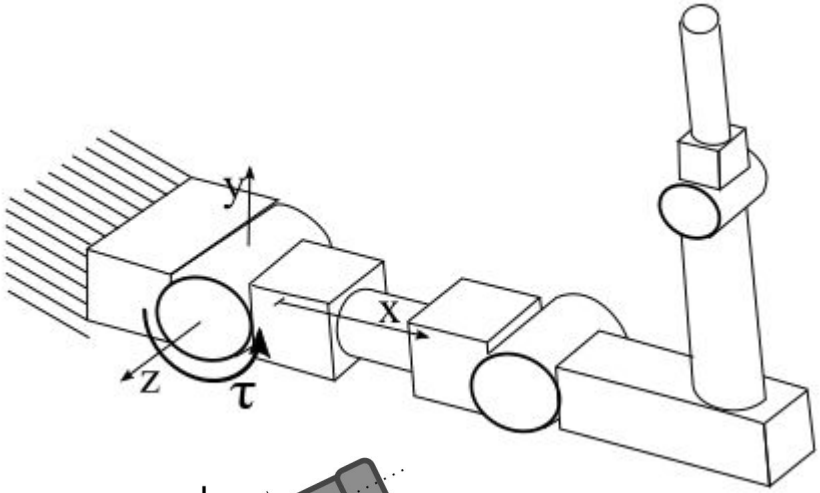
- un robot rígido,
- Con 5 GDL
- Pero todos fijos excepto el primero.



Control monoarticular

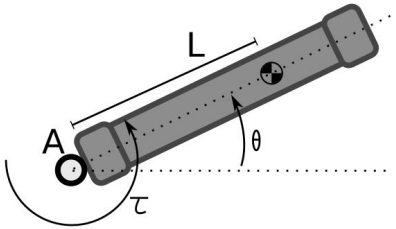
Consideremos el siguiente ejemplo:

- un robot rígido,
- Con 5 GDL
- Pero todos fijos excepto el primero.



Podemos modelarlo como:

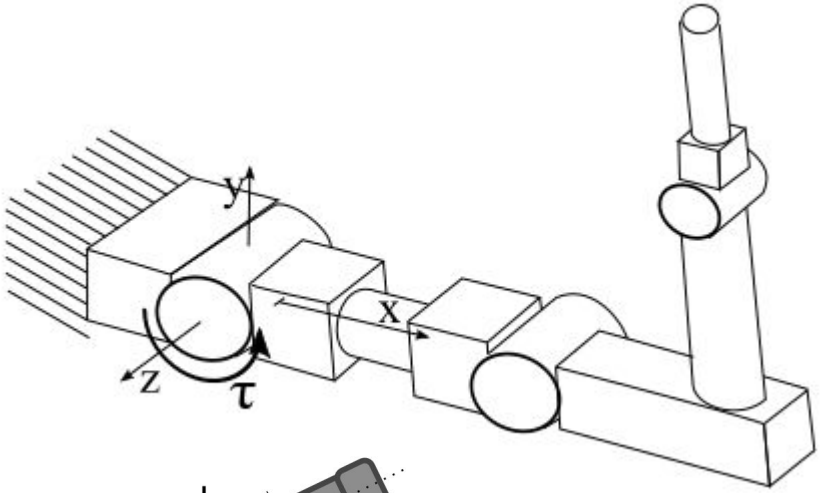
- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**



Control monoarticular

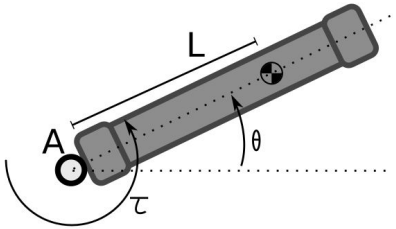
Consideremos el siguiente ejemplo:

- un robot rígido,
- Con 5 GDL
- Pero todos fijos excepto el primero.



Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**

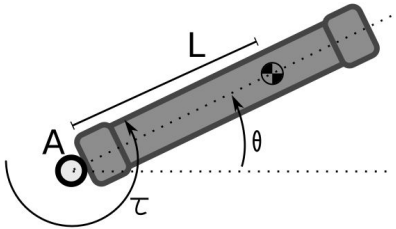


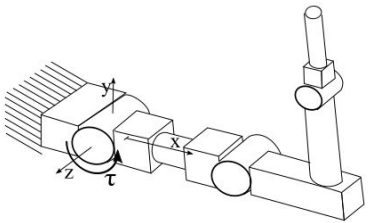
La ley del sistema es: $\tau = ML^2\ddot{\theta} + MgLcos(\theta)$

Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**





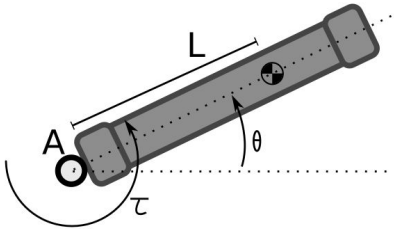
La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

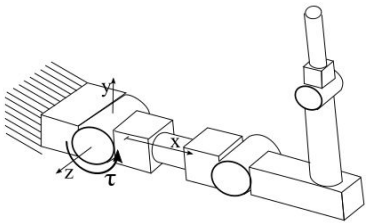
Recordando la ley de control particionada: $\tau = \alpha f' + \beta \longrightarrow \alpha f' + \beta = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**





La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

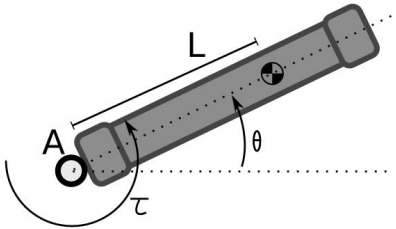
Recordando la ley de control particionada: $\tau = \alpha f' + \beta \longrightarrow \alpha f' + \beta = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

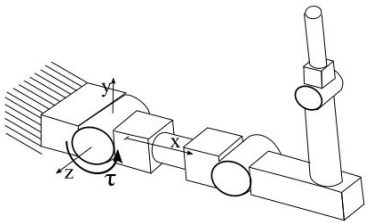
definiendo: $\alpha = ML^2$ **y** $\beta = MgLcos(\theta) + b\dot{\theta}$ $\Rightarrow$ $f' = \ddot{\theta}$

Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**





La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

Recordando la ley de control particionada: $\tau = \alpha f' + \beta \longrightarrow \alpha f' + \beta = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

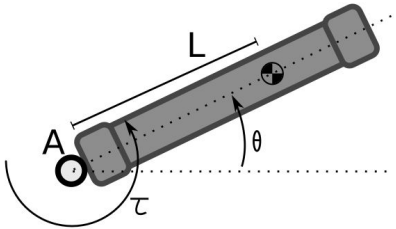
definiendo: $\alpha = ML^2$ y $\beta = MgLcos(\theta) + b\dot{\theta}$ $\Rightarrow$ $f' = \ddot{\theta}$

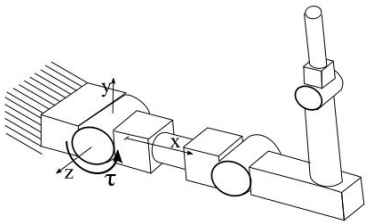
Agregando la ley de control para seguimiento de trayectoria: $f' = \ddot{x}_d + k'_v\dot{e} + k'_p e$ **donde:** $e = x_d - x$

Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**





La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

Recordando la ley de control particionada: $\tau = \alpha f' + \beta \longrightarrow \alpha f' + \beta = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

definiendo: $\alpha = ML^2$ y $\beta = MgLcos(\theta) + b\dot{\theta}$ $\Rightarrow$ $f' = \ddot{\theta}$

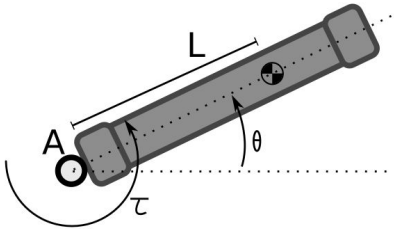
Agregando la ley de control para seguimiento de trayectoria: $f' = \ddot{x}_d + k'_v\dot{e} + k'_p e$ **donde:** $e = x_d - x$

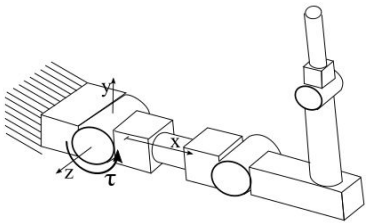
Resultando $\ddot{e} + k'_v\dot{e} + k'_p e = 0$ **donde para que el comportamiento sea críticamente amortiguado:** $k'_v = 2\sqrt{k'_p}$

Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**



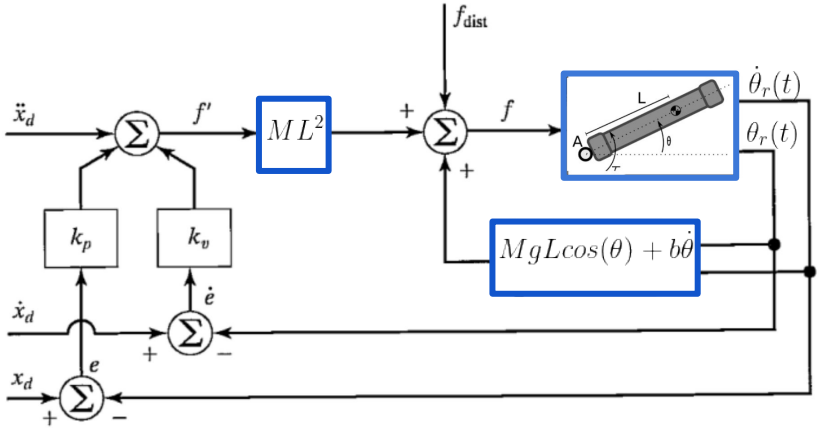


La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

La ley de control: $f' = \ddot{x}_d + k'_v\dot{e} + k'_p e$

con: $e = x_d - x$

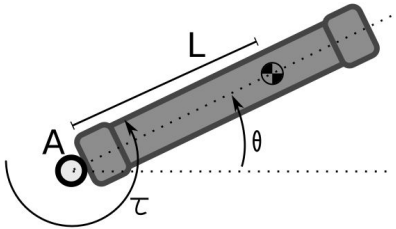
$k'_v = 2\sqrt{k'_p}$

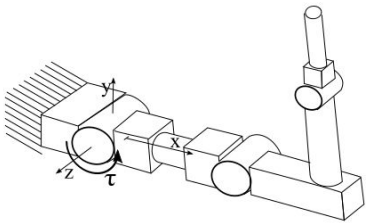


Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**





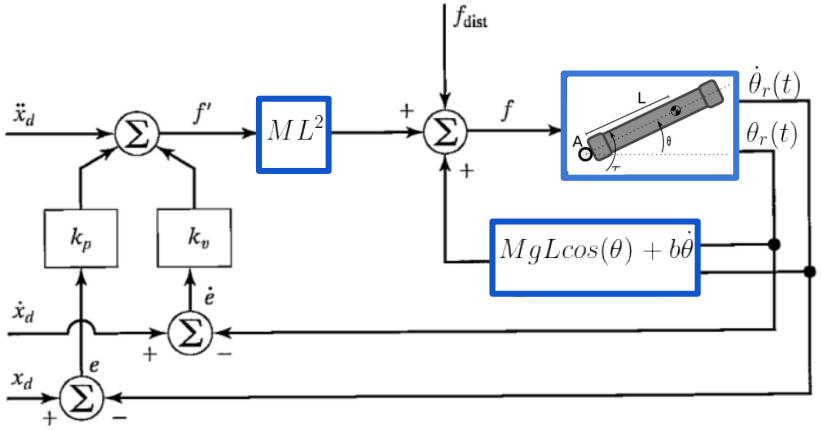
La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

La ley de control: $f' = \ddot{x}_d + k'_v\dot{e} + k'_p e$

con: $e = x_d - x$

$k'_v = 2\sqrt{k'_p}$

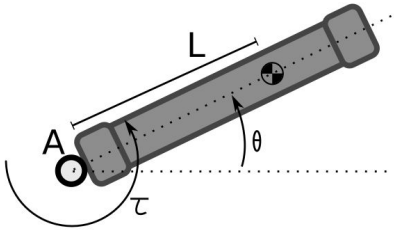
¿Cómo resuelvo la ecuación del sistema?

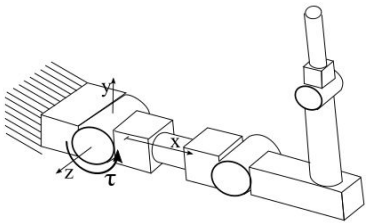


Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**





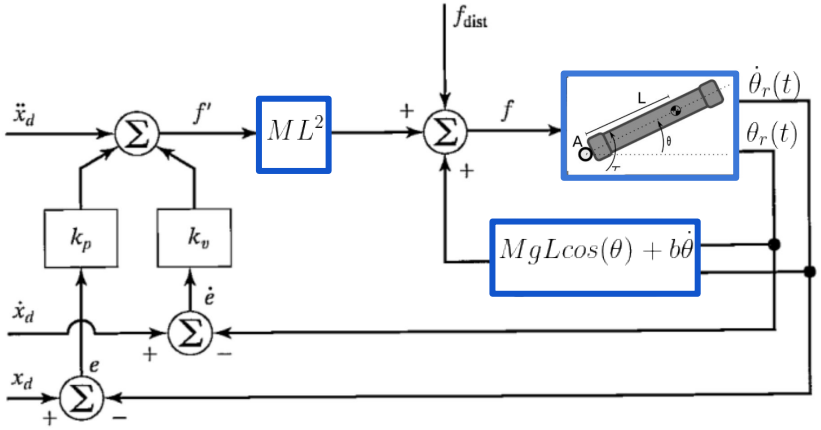
La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

La ley de control: $f' = \ddot{x}_d + k'_v\dot{e} + k'_p e$

con: $e = x_d - x$

$k'_v = 2\sqrt{k'_p}$

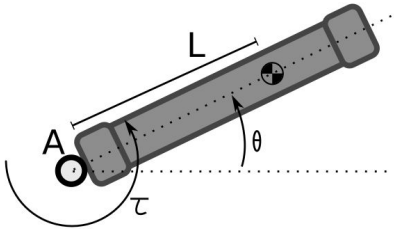
Para alta frecuencia se puede linealizar…

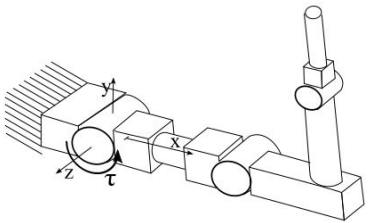


Control monoarticular

Podemos modelarlo como:

- un robot rígido,
- articulado en **A**,
- peso **m** uniformemente distribuido,
- distancia al centro de masa **L**,
- par articular **τ**





La ley del sistema es: $\tau = MgLcos(\theta) + b\dot{\theta} + ML^2\ddot{\theta}$

Reescribiendo $\ddot{\theta} + \frac{b}{ML^2}\dot{\theta} + \frac{g}{L}cos(\theta) - \frac{\tau}{ML^2} = 0$

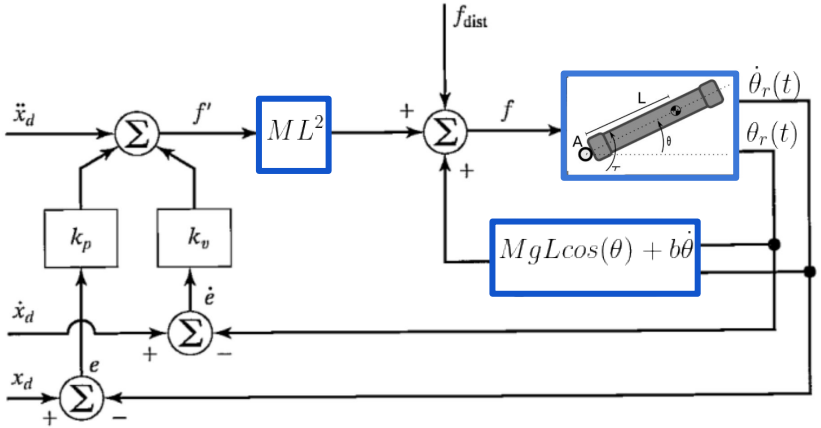
Definiendo:

$$\dot{\theta} = \frac{\theta_t - \theta_{t-1}}{\Delta t}$$

$$\ddot{\theta} = \frac{\theta_t - 2\theta_{t-1} + \theta_{t-2}}{\Delta t^2}$$

Obtengo:

$$\theta_t = \frac{\theta_{t-1}(MgLsen(\theta_{t-1}) - b/\Delta t - 2ML^2/\Delta t^2) + \theta_{t-2}(ML^2/\Delta t^2) - \tau_{t-1} + MgLcos(\theta_{t-1})}{sen(\theta_{t-1}) - b/\Delta t - ML^2/\Delta t^2}$$



Control monoarticular - En loop

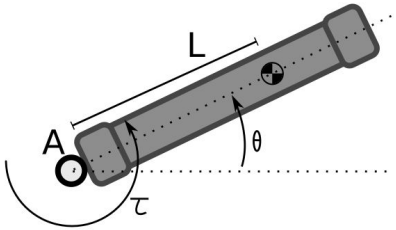
Tengo:

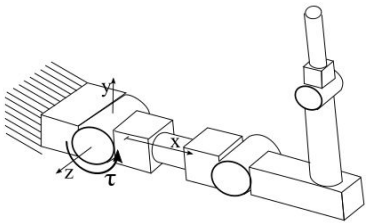
```
M = …
L = …
alpha = M.L**2
tray. deseada = f(t)
vel. deseada ← df(t)
acel. deseada ← ddf(t)
```

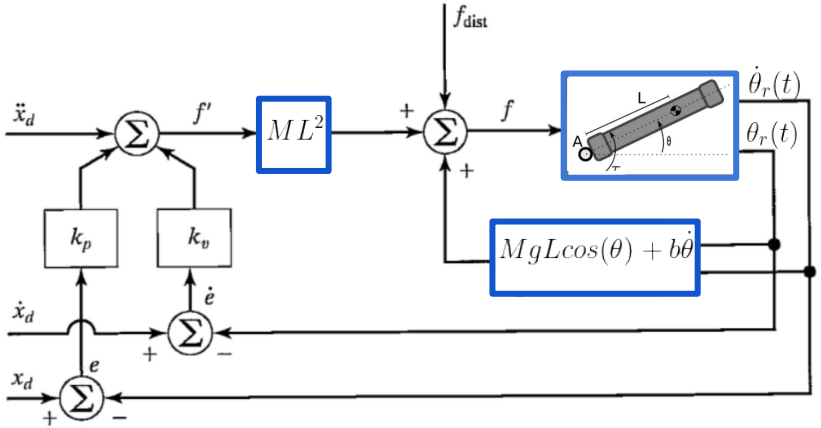
Defino:

```
tiempo = 5          # segundos
sample_rate = 100 # Hz
pasos = tiempo*sample_rate
k_p = …
k_v = …
# Inicializo en t=0
θ_d_vector ← f(0)
dθ_d_vector ← df(0)
ddθ_d_vector ← ddf(0)
θ_r_vector = θ_d_vector
dθ_r_vector = dθ_d_vector
ddθ_r_vector = ddθ_d_vector
e = [0.]
de = [0.]
beta = M.g.L.cos(θ_d(0))
f_p = ddf(0)
tau_vector = alpha*f_p + beta
```

```
# Loop para t > 0
for i in range[1:pasos]
        t = i/sample_rate
        θ_d = f(t)
        dθ_d ← df(t)
        ddθ_d ← ddf(t)
        tau_old = tau_vector[end]
        θ_r_t = Sistema(tau_old)
        # opción 1 - sale del sist.
        dθ_r_t = Sistema(tau_old)
        # opción 2 -dif. fin.
        dθ_r_t =(θ_r_t - θ_r_vec
        e_t = θ_d -  θ_r_t
        de_t = dθ_d -  dθ_r_t
        f_p_t = ddθ_d + k_v*de_t
        f_t = alpha*f_p_t + beta
        # Guardo
        θ_d_vector.append(θ_d)
        …
        θ_r_vector.append(θ_r_t)
        …
        e.append(e_t)
        …
        tau_vector.append(f_t)
```







FIN!