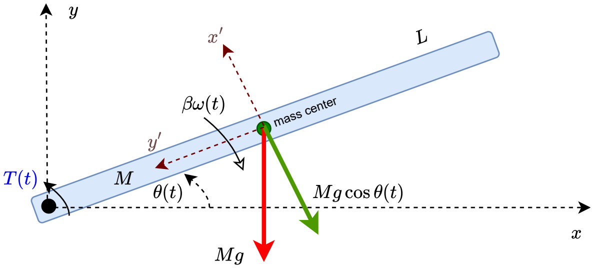
Controlling a Rotating Arm

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
`santoro@dmi.unict.it`

Robotic Systems

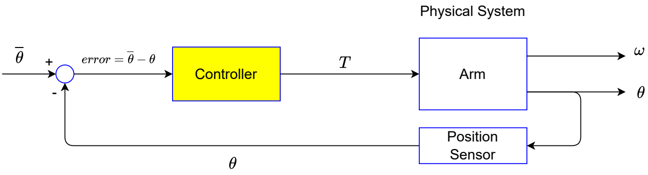


Discretisation

$$\omega(k+1) = (1 - \frac{3}{2}\frac{\Delta T}{ML}\beta)\omega(k) - \frac{3}{2}\frac{\Delta T}{L} g\cos\theta(k) + \frac{3\Delta T}{ML^2}T(k)$$

$$\theta(k+1) = \theta(k) + \omega(k)\Delta T$$

Simple Position Control



Simple Position Control

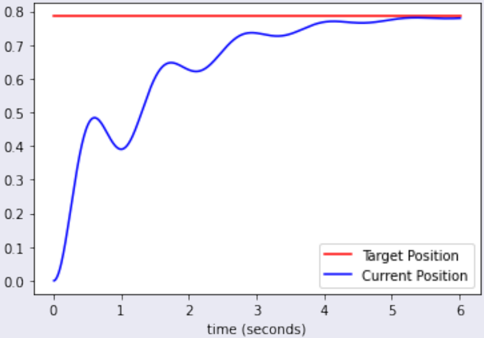
- Let us a consider a “simple” position controller
- For the cart a P Controller has been enough, but...
- in the arm, the presence of gravity requires a “push” (non-zero output) in case of $error = 0$
- A **PI Controller** is mandatory
- However, the natural oscillating behaviour of the system creates some hard problems...

(see examples/simple_control/arm_simple_position_control_PID.ipynb)

Simple Position Control

Simple Position Control

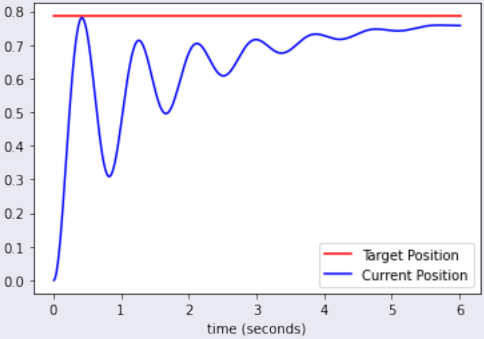
$K_P = 4, K_I = 2$



Control is really slow, and the system oscillates

Simple Position Control

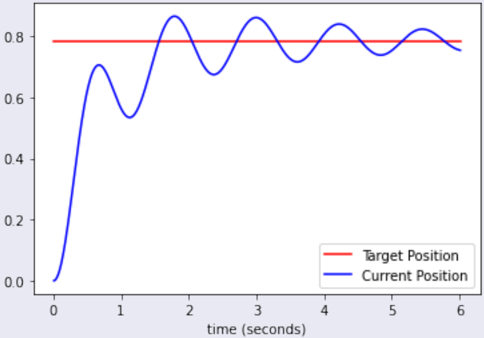
$K_P = 6, K_I = 2$



Incrementing K_P does not help...

Simple Position Control

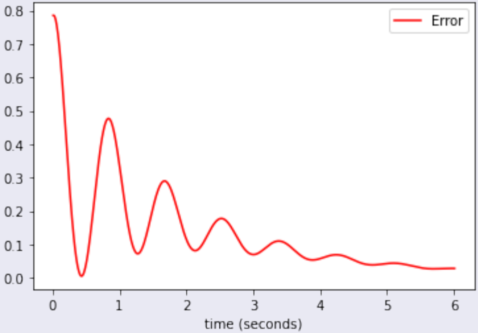
$K_P = 4, K_I = 4$



Also incrementing K_I does not help...

Arm Position Control

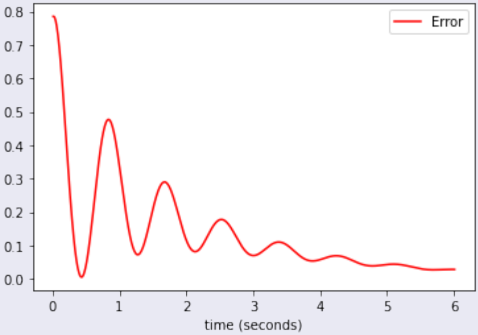
Let's Analyse the trend of the **error**



- When the error **decreases**, the control action is OK
- When the error **increases**, we need more control action

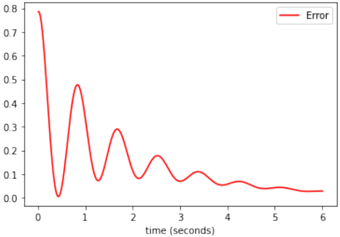
Arm Position Control

Let's **derivate** the **error**



$$\dot{e} = \frac{de(t)}{dt}$$

- When the error **decreases**, $\dot{e} < 0$, the control action is OK
- When the error **increases**, $\dot{e} > 0$, we can increase control action



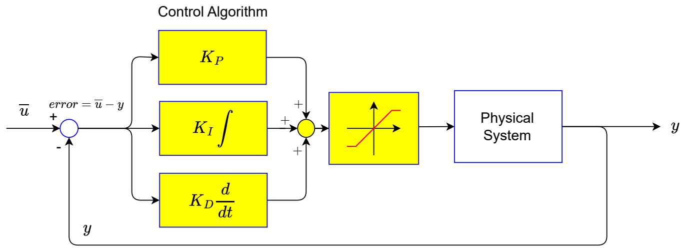
The PID Controller

Let's add a factor **proportional to the derivative of the error**:

$$ControlAction = PI(error) + K_D \dot{e}$$

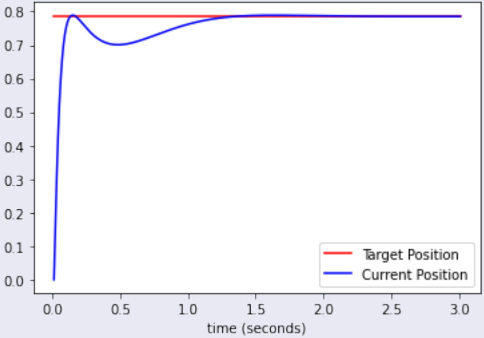
- When the error **decreases**, $\dot{e} < 0$, so the derivative action acts a as "brake" reducing the control action (this is OK, because the trend of the error is, in any case, decreasing)
- When the error **increases**, $\dot{e} > 0$, so the derivative factor **improves** the control action

The PID Contoller



The PID Contoller

$K_P = 10, K_I = 20, K_D = 1.5$

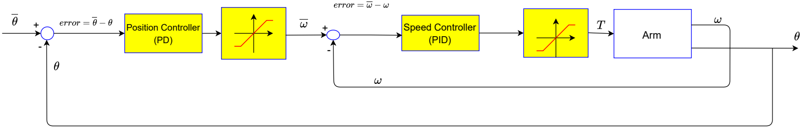


(see examples/simple_control/arm_simple_position_control_PID.ipynb)

- System is fast and no oscillation occurs
- The value of K_I (that is high) is needed due the fact that we must compensate the gravity

Position and Speed Cascading Control

Arm Position Control



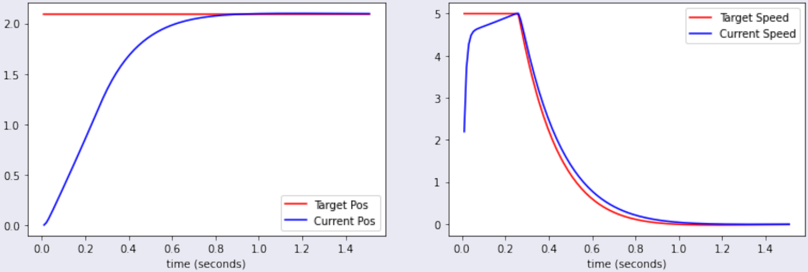
Position and Speed Control

- As in the cart, also in the arm the best solution for position control is the **double loop position + speed**
- In oscillating systems, the position controller is usually a **PD**, while the speed controller is usually a **PID**

(see examples/position_control/arm_position_control.ipynb and godot/arm_no_physics/)

Position and Speed Control

Position: $K_P = 5, Sat = 5\ rad/s$
Speed: $K_P = 5, K_I = 20, Sat = 20\ Nm$



Controlling a Rotating Arm

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it

Robotic Systems