



---

# **Ejercicios 2 y 3**

## **Control posición-tracking**

Robótica

---

Miguel Jorge Arasanz Malo	839385
Andrei Dumbrava Luca	844417
Diego Roldán Urueña	841723

3 de marzo de 2025

# Índice

1. Ejercicio 2.....	3
2. Ejercicio 3.....	6

## 1. Ejercicio 2

Para la realización de este ejercicio se ha seguido la implementación del pseudo-algoritmo de la página 17 del tema 8 de la asignatura. Este pseudo-programa añade todas las funcionalidades para que el robot se desplace hasta una nueva posición y esté ‘mirando’ hacia una dirección proporcionada por el usuario.

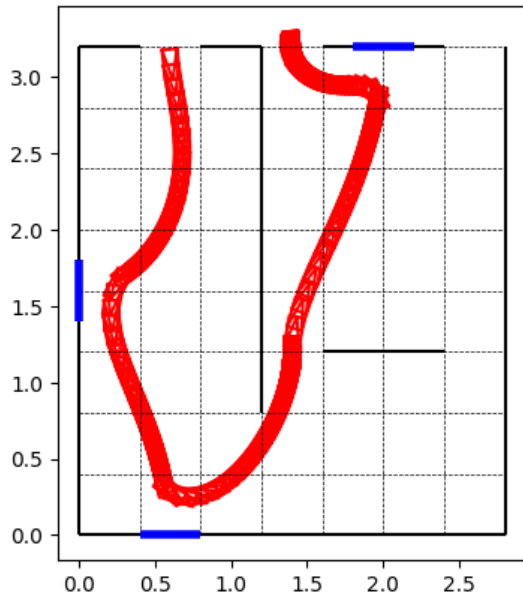
Se ha implementado, esta vez en Python, la función control de pose. Esta función recibe como parámetros una lista con posiciones globales de cuadros y puertas. El robot debe posicionarse a una distancia y ángulo concretos con respecto a los cuadros. Estos parámetros deberán de definirse en función al robot, ¿a que distancia es capaz de observar una imagen? y ¿qué rango de visión tiene la cámara usada? El último parámetro que recibe la función representa la posición global inicial del robot.

Se han definido como parámetros internos de la función, las velocidades lineal y angular máxima que queremos que el robot alcance. El tiempo que el robot avanza en cada simulación también es un parámetro. A parte de estos, también se han definido los parámetros para cumplimentar la matriz K, que se usa para calcular las velocidades lineales y angulares que debe utilizar el robot en cada momento:

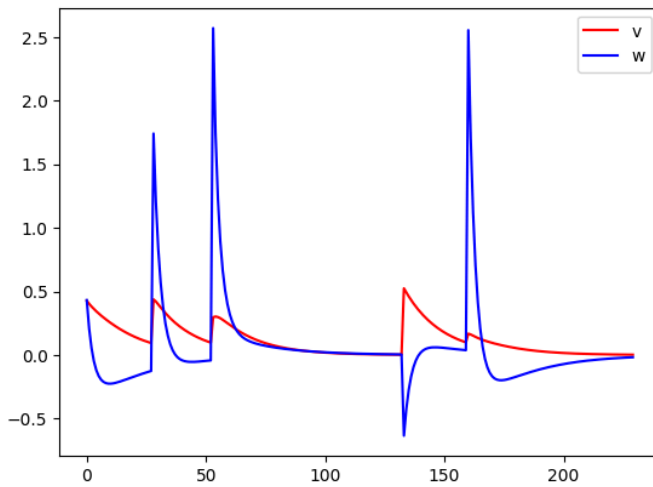
$$K = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{pmatrix} = \begin{pmatrix} k_{\rho} & 0 & 0 \\ 0 & k_{\alpha} & k_{\beta} \end{pmatrix}$$

Como parámetros se han establecido los siguientes. Para la distancia a la que el robot ve un cuadro se ha asignado una baldosa (40mm). Para el ángulo al que el robot ve un cuadro se ha asignado 30 grados. El tiempo entre simulaciones es de 250 milisegundos. Para los parámetros para completar la matriz K se ha utilizado:  $k_{\rho}=0.25$ ,  $k_{\alpha}=1.5$  y  $k_{\beta}=0.8$ . Se ha entendido cual es la función de cada parámetro y se han seleccionado teniendo en cuenta su propósito.

A continuación se muestra la trayectoria que ha seguido el robot para salir desde la salida, observar todos los cuadros y finalizar el recorrido en la meta.



Este es el perfil de velocidades lineal y angular al seguir esta trayectoria.



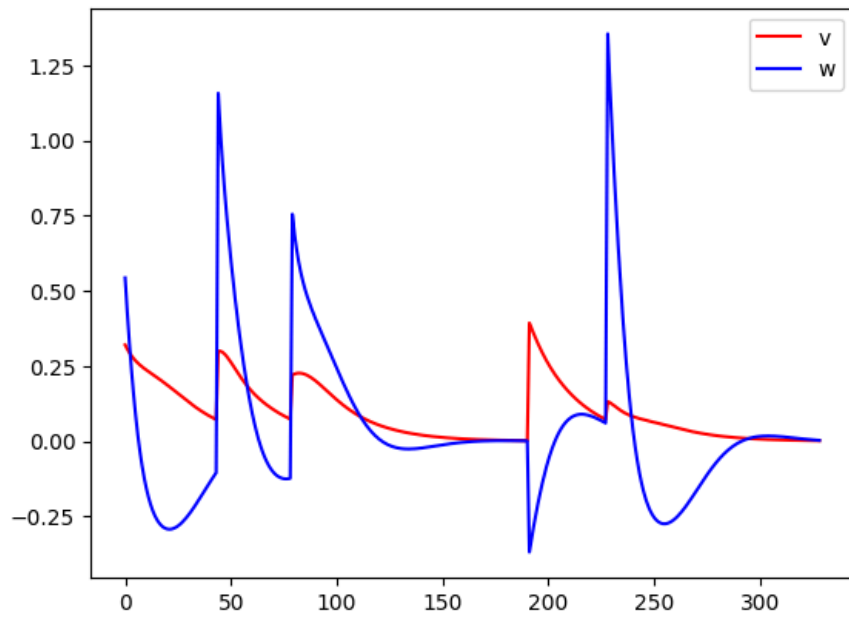
El segundo apartado del ejercicio pide que la velocidad lineal máxima sea 1.5m/s y la velocidad angular máxima sea 2 rad/s, se han aplicado dichas restricciones. Para ello, los parámetros para completar la matriz K se han calculado a partir de las fórmulas en la imagen inferior. Se han asignado los siguientes valores para  $\rho_{\max}$ , para  $\alpha_{\max}$  y para  $\beta_{\max}$  respectivamente 8,  $\pi$  y  $\pi$ .

$$k_{\rho} = v_{\max}/\rho_{\max}$$

$$k_{\alpha} = \omega_{\max}/\alpha_{\max}$$

$$k_{\beta} = \omega_{\max}/\beta_{\max}$$

Este es el perfil actualizado de velocidad lineal y angular.



## 2. Ejercicio 3

Para la realización de este ejercicio se ha seguido el esquema del pseudo-algoritmo incluido en la página 20 del mismo tema. Con este algoritmo se logra visualizar la trayectoria tanto del robot como la del móvil, además de marcar el punto de intercepción del robot sobre el objeto móvil, que en este caso es otro robot.

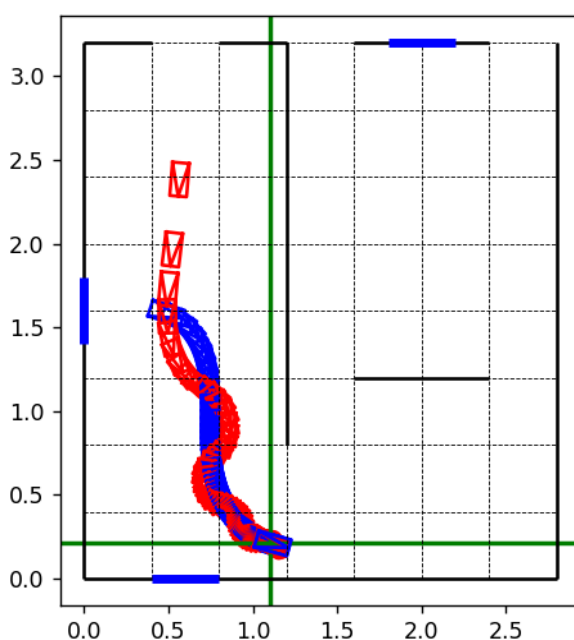
Desde Python se ha implementado una función denominada *tracking* que mediante dos vectores con los parámetros de inicialización necesarios para realizar el movimiento de ambos objetos simula estos movimientos hasta que cumplen la condición de estar a una distancia menor a 0.03 cm y 2 grados, pedida por el enunciado. El algoritmo en sí emplea lo realizado en el ejercicio anterior para simular todo el movimiento necesario definido entre varios puntos, incluyendo estas comprobaciones para lograr cumplir el objetivo de que el robot alcance al móvil.

Por lo que, de la misma manera, las fórmulas son prácticamente idénticas al ejercicio anterior, y se definen los parámetros de la misma manera, empleando  $k_p$ ,  $k_a$  y  $k_b$

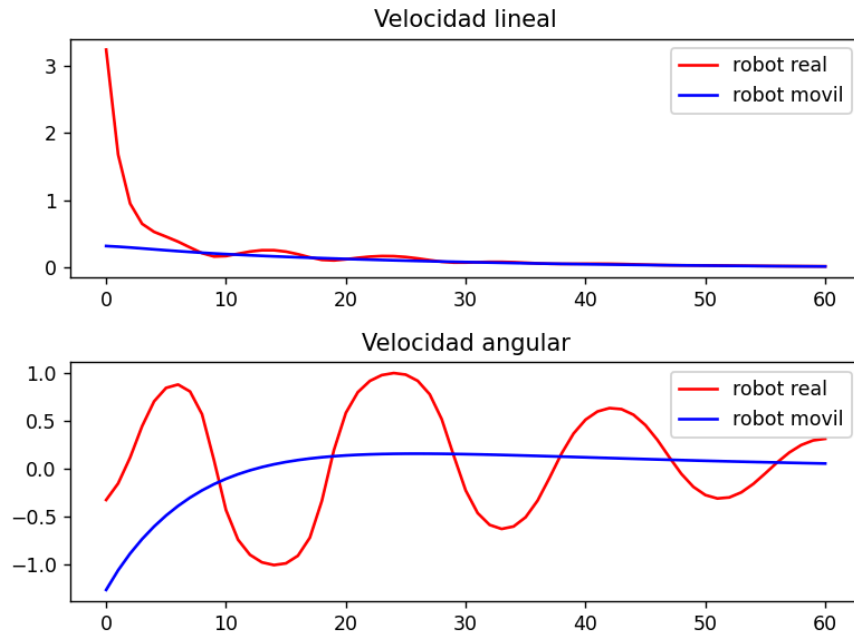
Concretamente, para el primer apartado, donde se pide que el robot alcance al móvil antes de P1, tenemos estas trayectorias, definidas gracias a los parámetros:

Robot:  $k_p=2$ ,  $k_a=0.8$  y  $k_b=0.2$

Móvil:  $k_p=0.2$ ,  $k_a=0.8$  y  $k_b=0.4$



Además de tener representados los diferentes perfiles de velocidad.

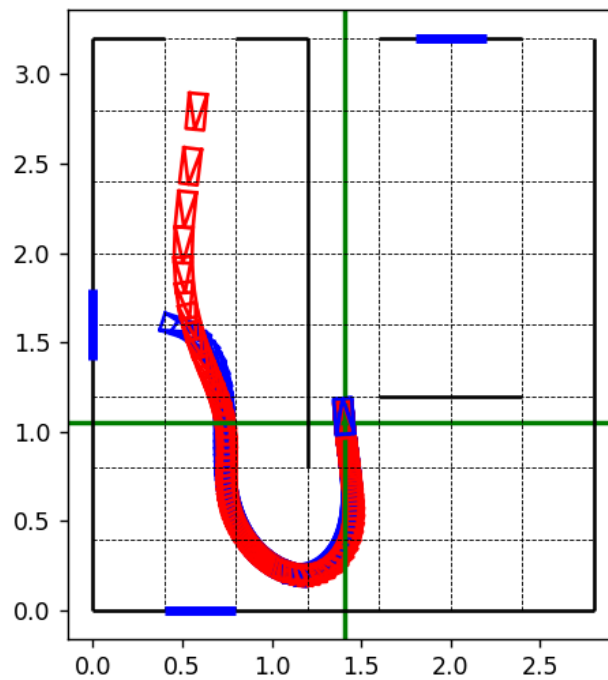


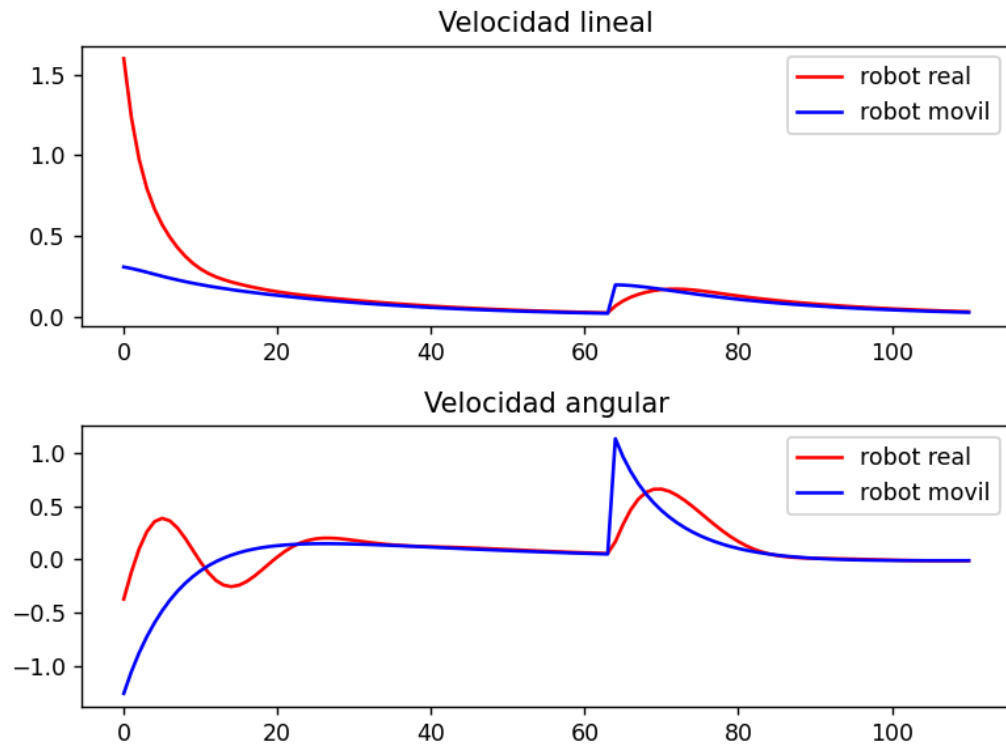
Para el segundo apartado se pide que el robot alcance al otro a partir de la trayectoria entre P1 Y P2, esto se logra con los parámetros a estos valores:

Robot:  $k_p=0.99$ ,  $k_a=1.4$  y  $k_b=0.2$

Móvil:  $k_p=0.19$ ,  $k_a=0.8$  y  $k_b=0.4$

Mientras, la trayectoria definida y los perfiles de velocidad son los siguientes:





\*Los parámetros calculados están optimizados para lograr el objetivo marcado, pero existe la posibilidad de optimizarlos de mejor manera para lograr resultados más visibles.