

LABORATORIO 2

Rodrigo Aguilera

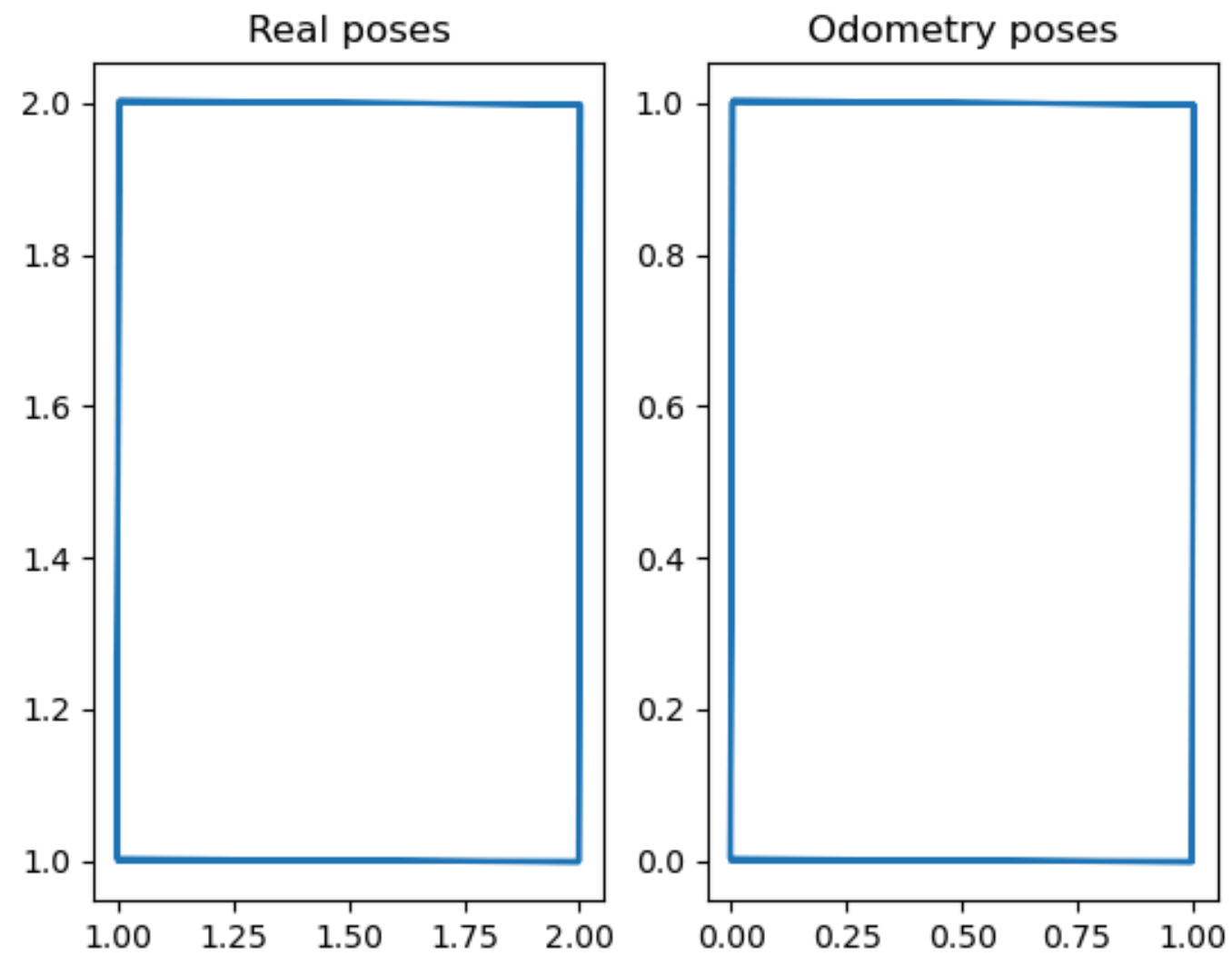
Vicente Urizar

Benjamín Yañez

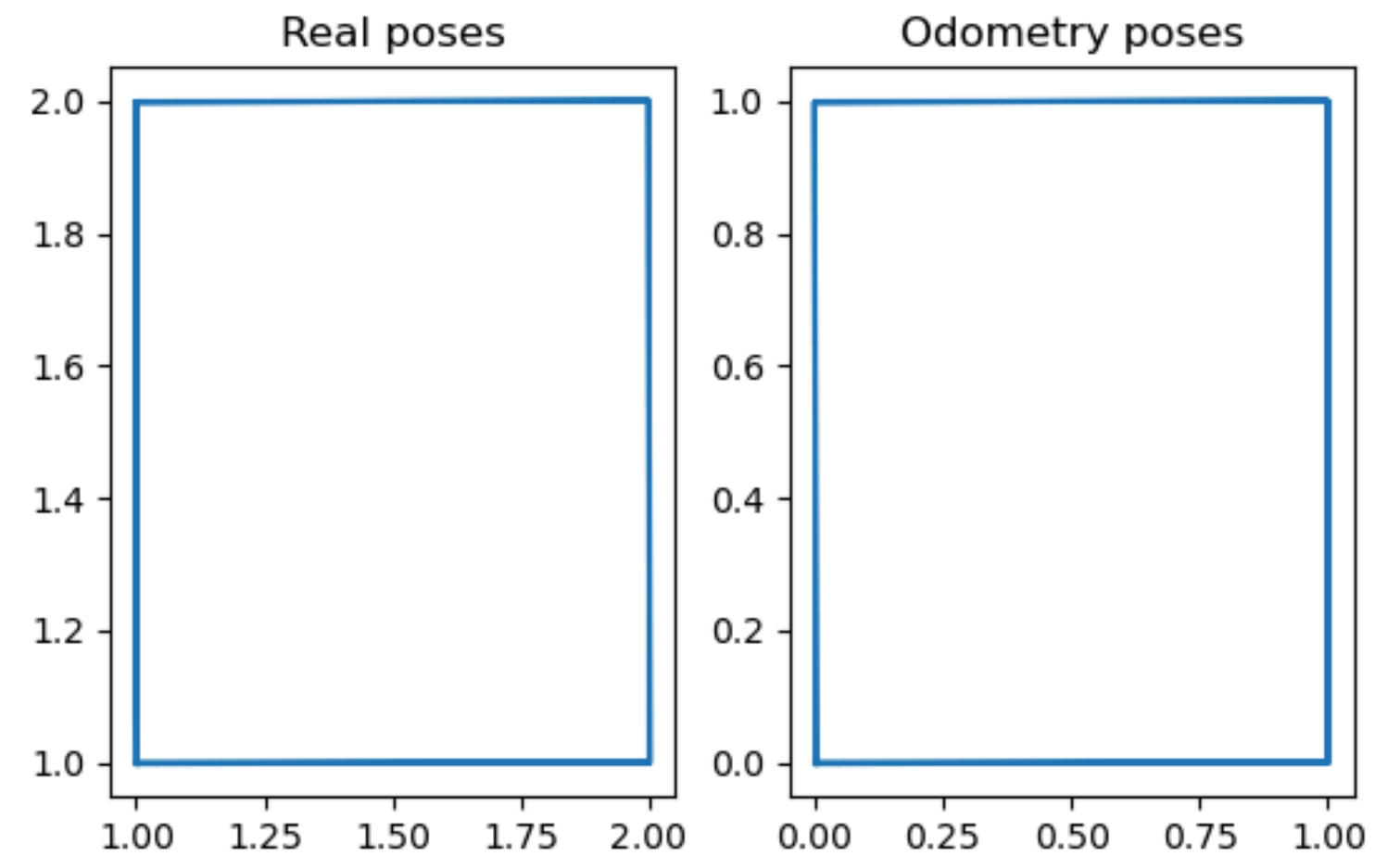
Parte 1

Navegación utilizando control P y PI (Simulador)

Trayectoria controlador P

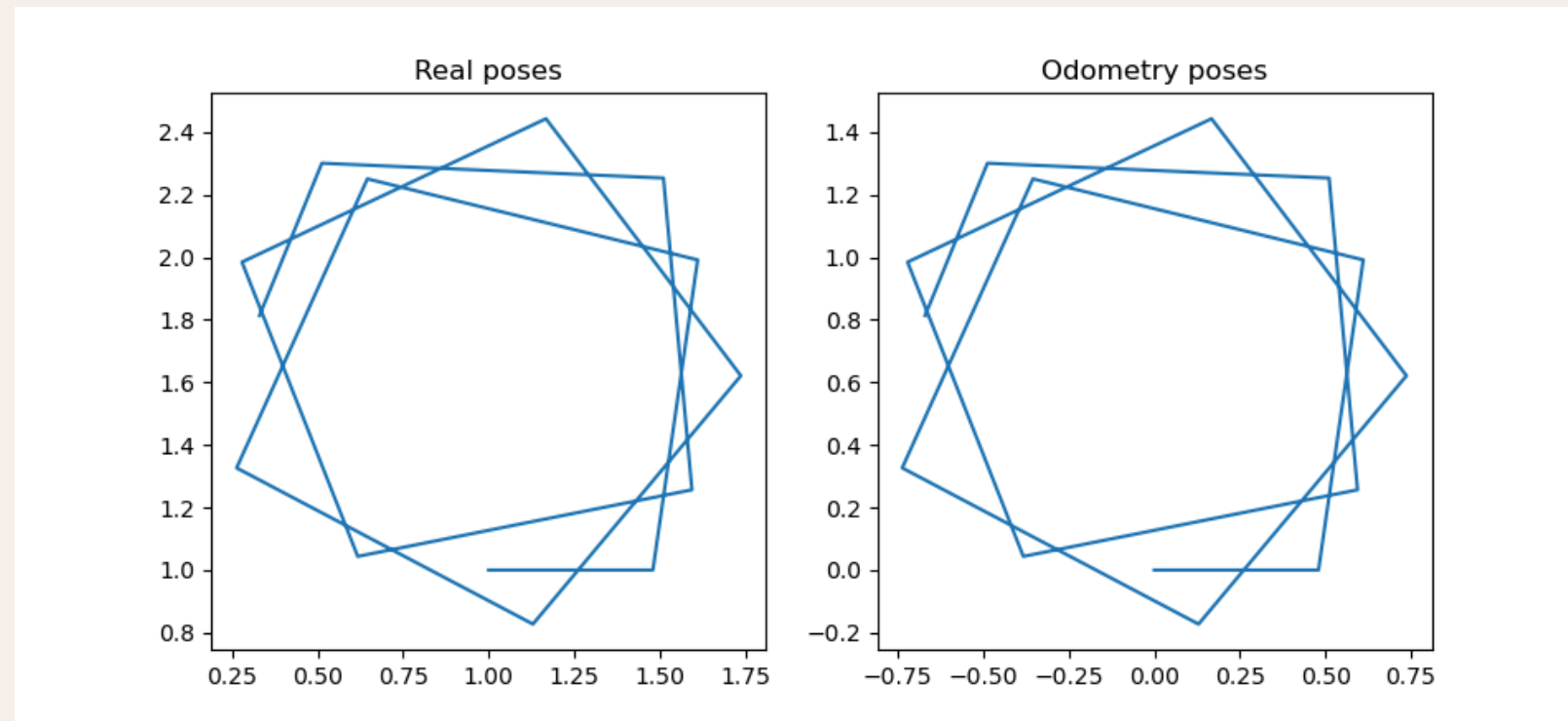


Trayectoria controlador PI

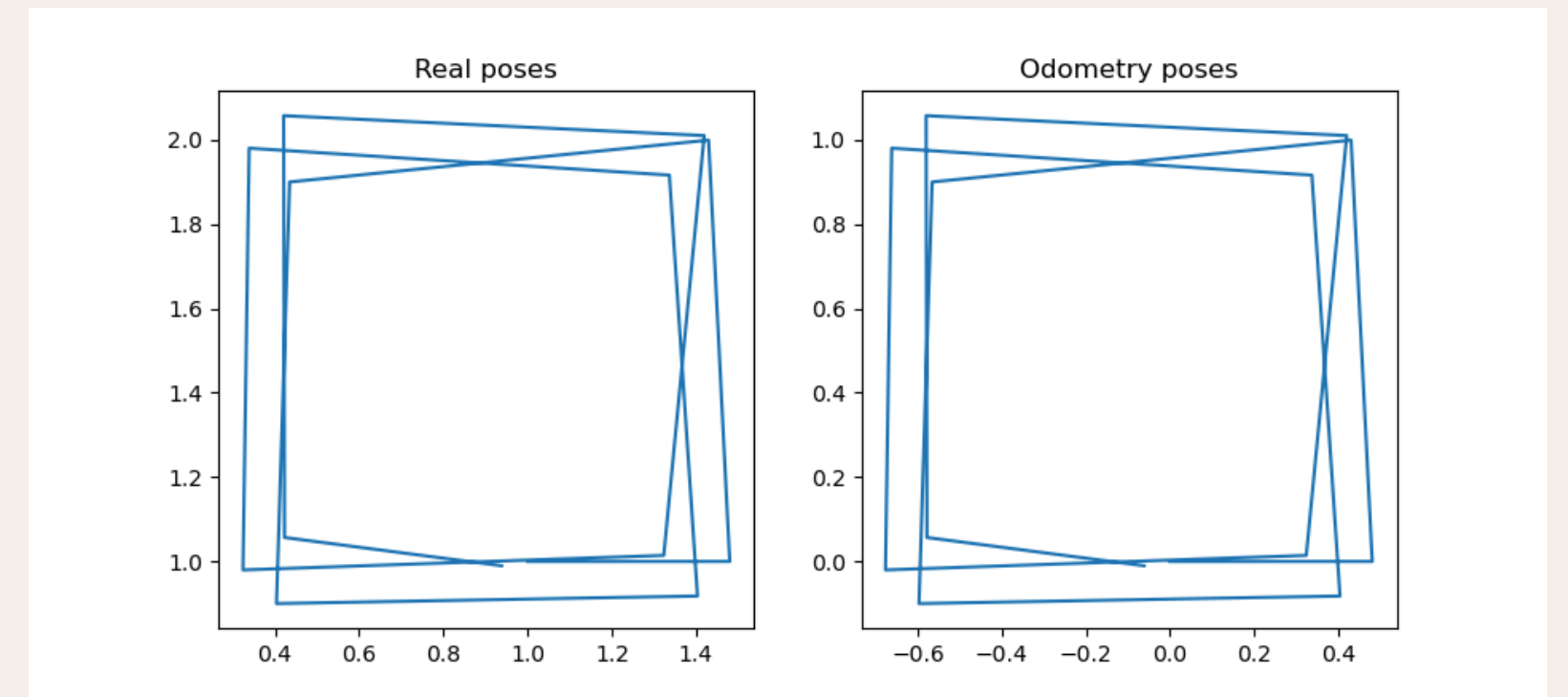


Trayectorias lab pasado

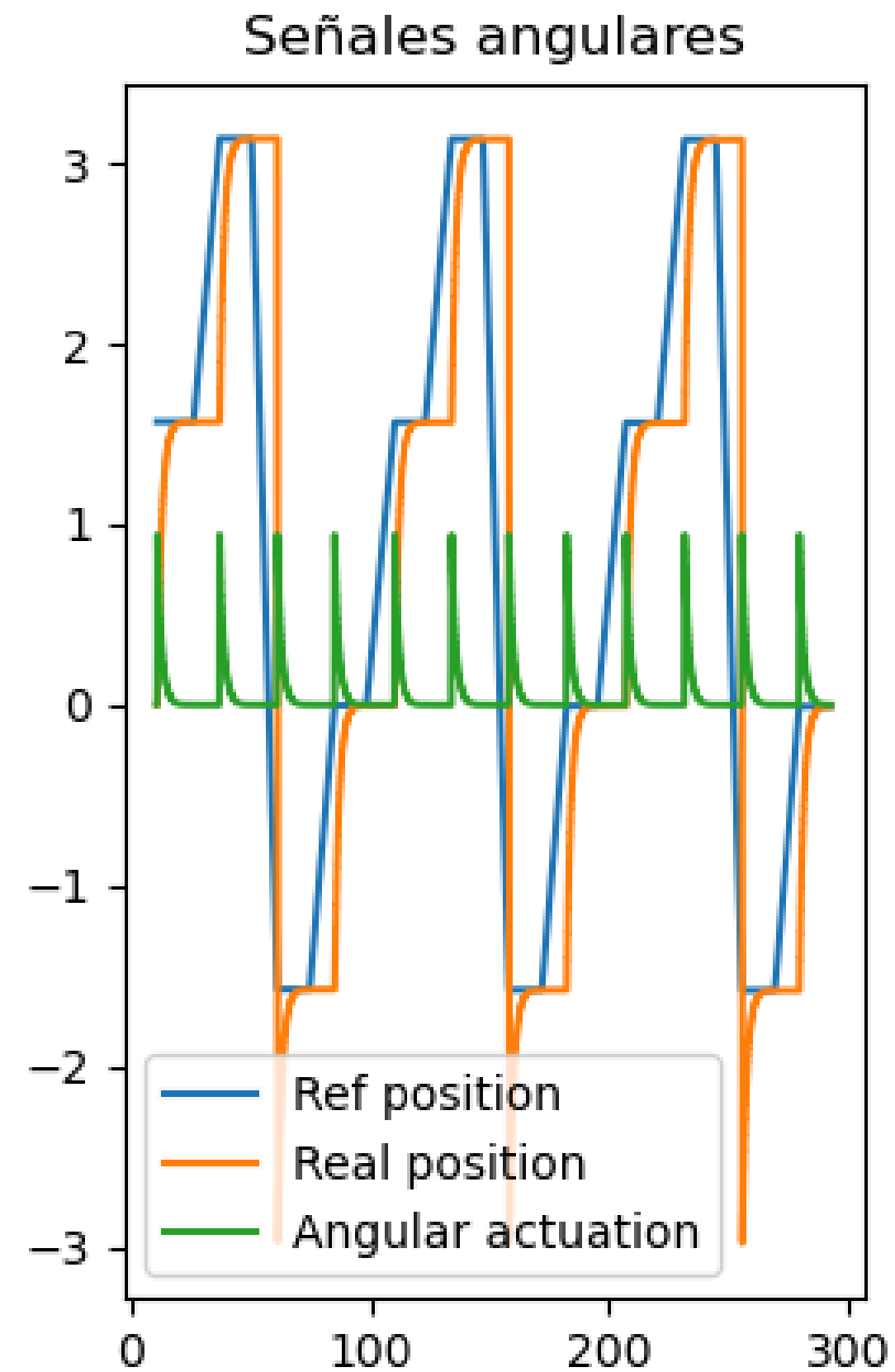
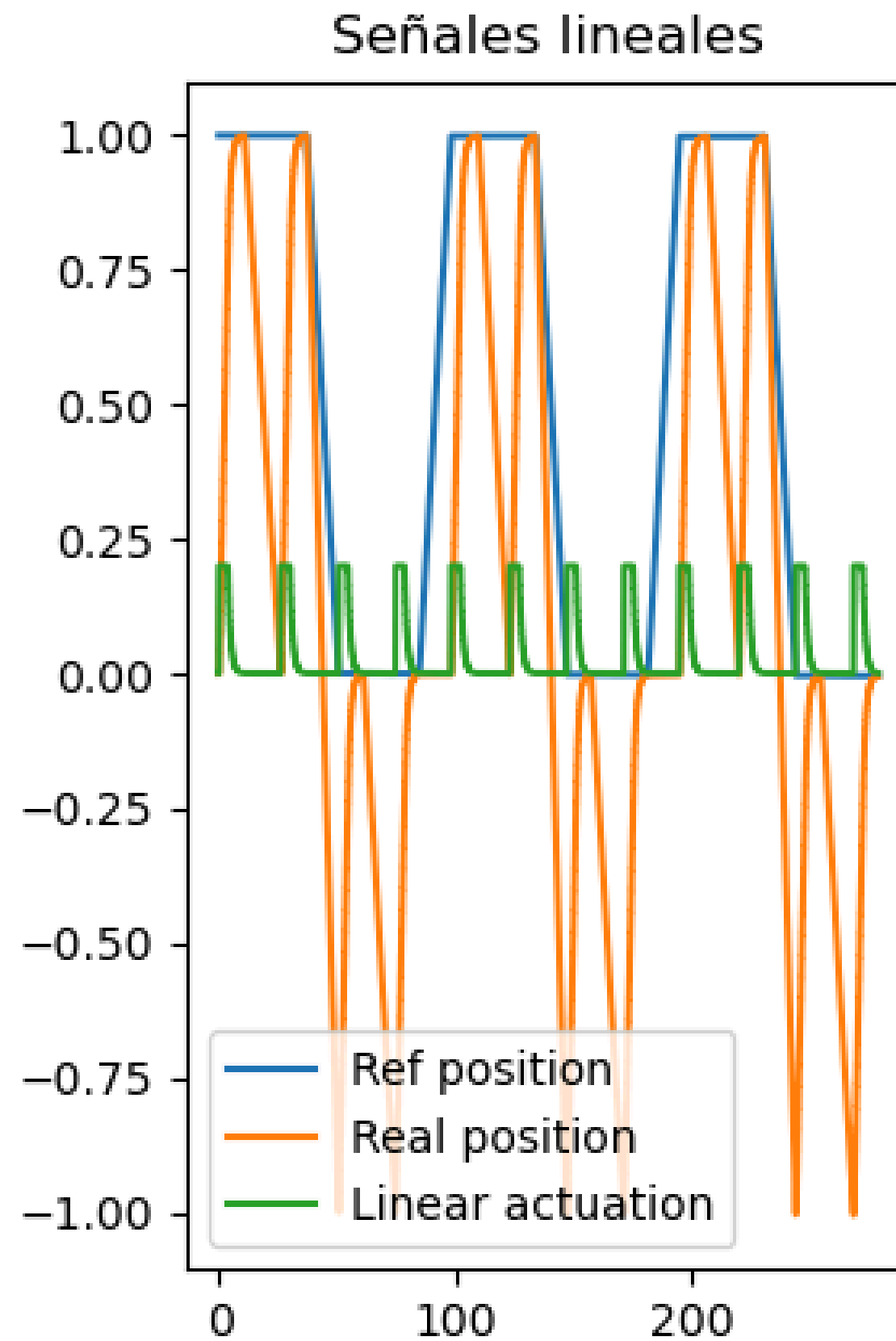
Trayectoria sin corrección



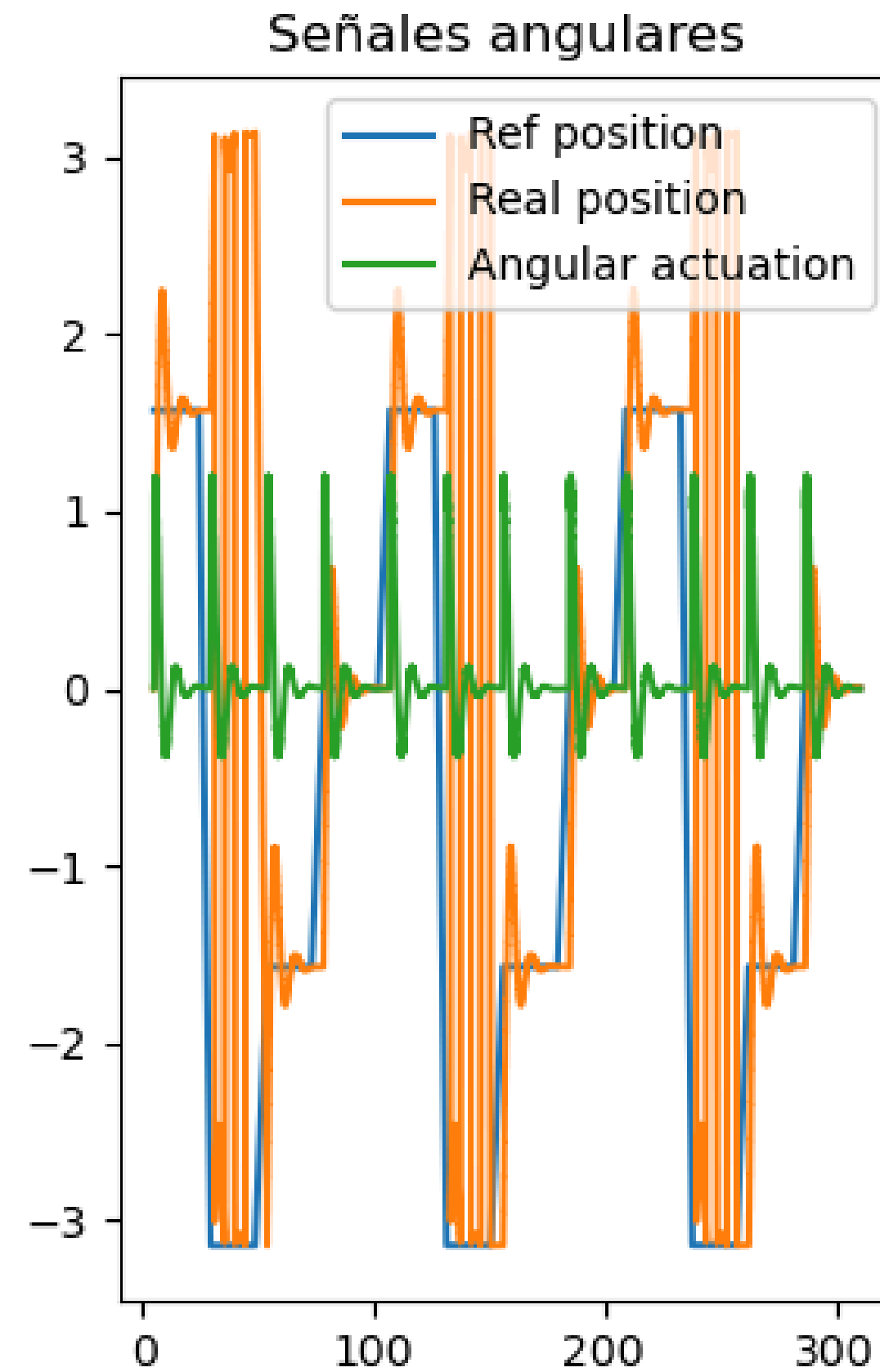
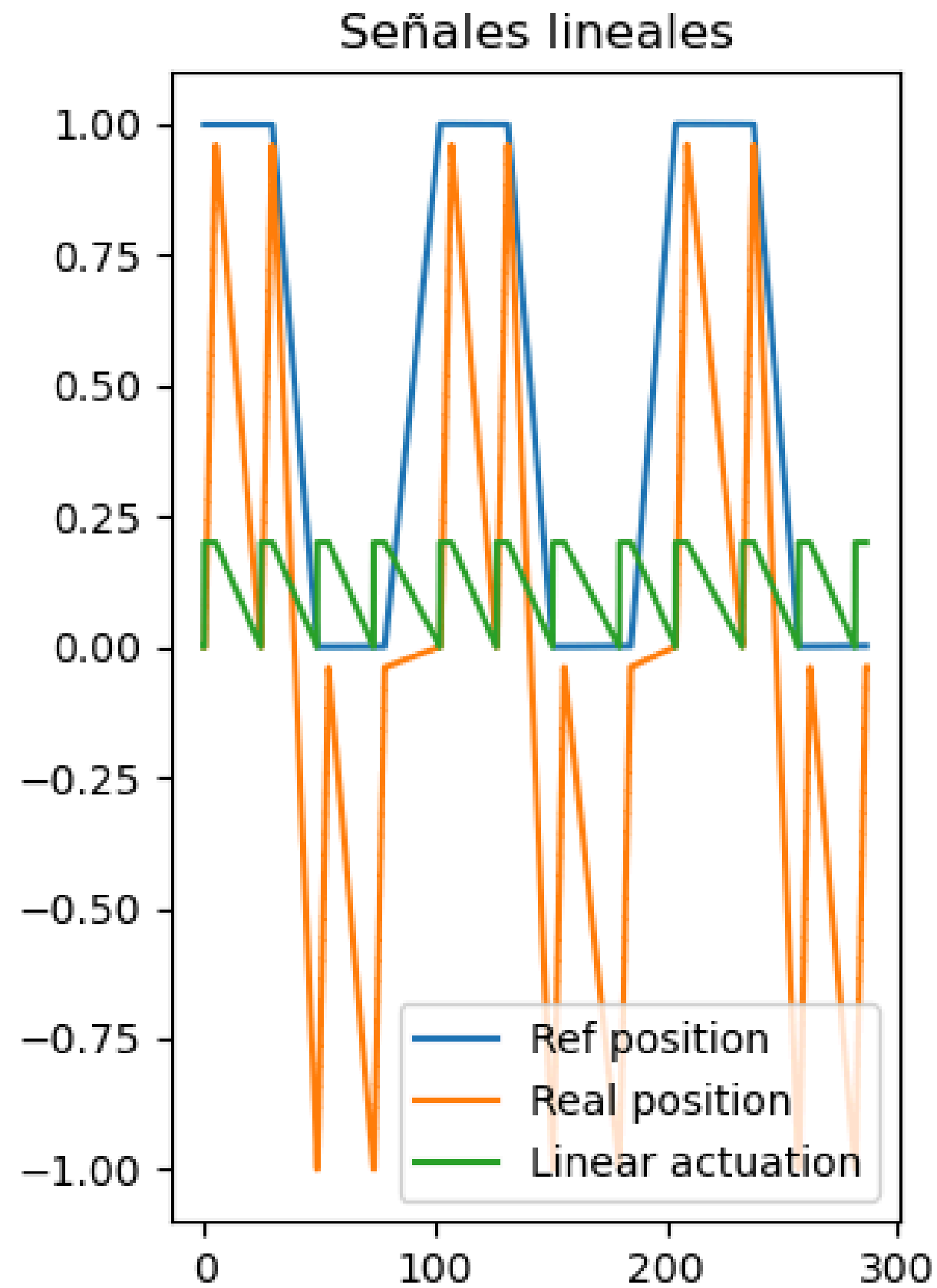
Trayectoria con corrección



Gráficos señales controlador P



Gráficos señales controlador PI



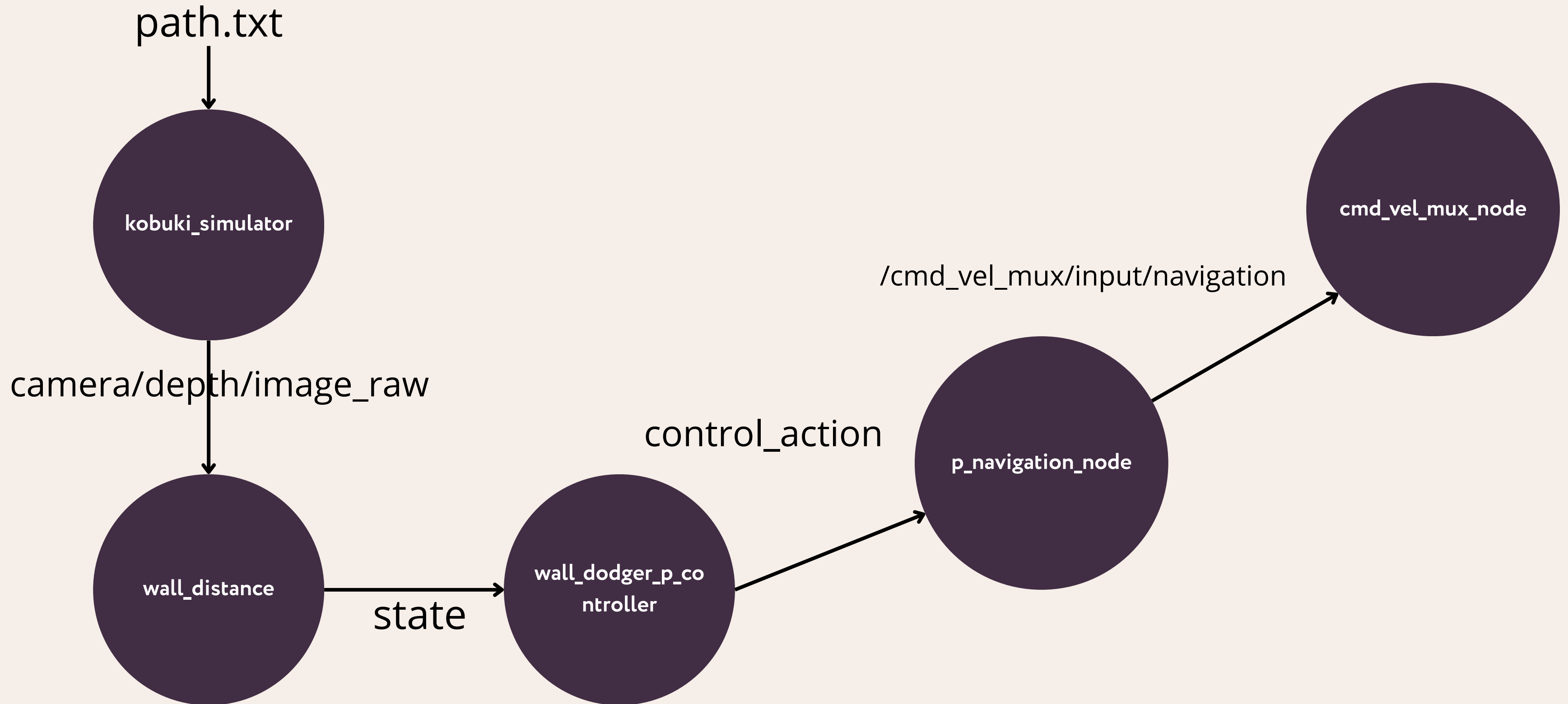
Cálculo del error angular

La odometría es medida de -180° a 180° , por lo que cuando se saca el error cuando el ángulo deseado es 180° empieza a haber un problema, ya que la odometría va a empezar a marcar entre -180° y 180° cambiando de manera brusca el error, la manera en la que se arregló esto fue aplicando la función $\text{atan2}(\sin(\text{error}), \cos(\text{error}))$, que al momento de que el ángulo de odometría pasa a -179° , el error será como -359° o por ahí porque ahora en vez de restarle al ángulo estará sumando, lo que hacemos con aplicar atan2 al error es transformar este ángulo de 359° a su equivalente de -180° a 180° , lo que para nuestra suerte nos arregla el problema anterior, ya que el error pasara a ser -1° que coincide con el error real que buscábamos que sería:

$$-180^\circ + 179^\circ = -1^\circ$$

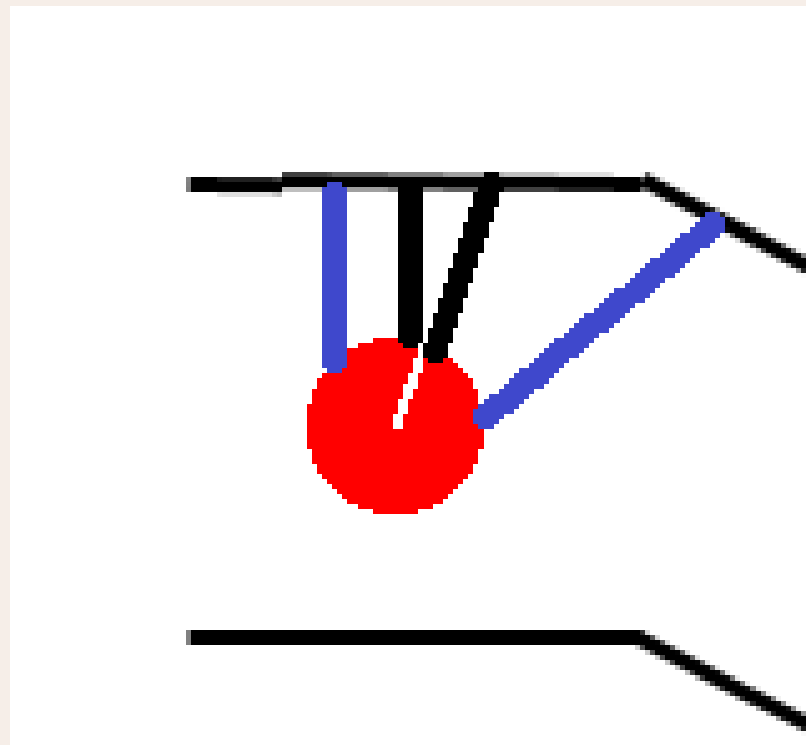
Parte 2

Nodos



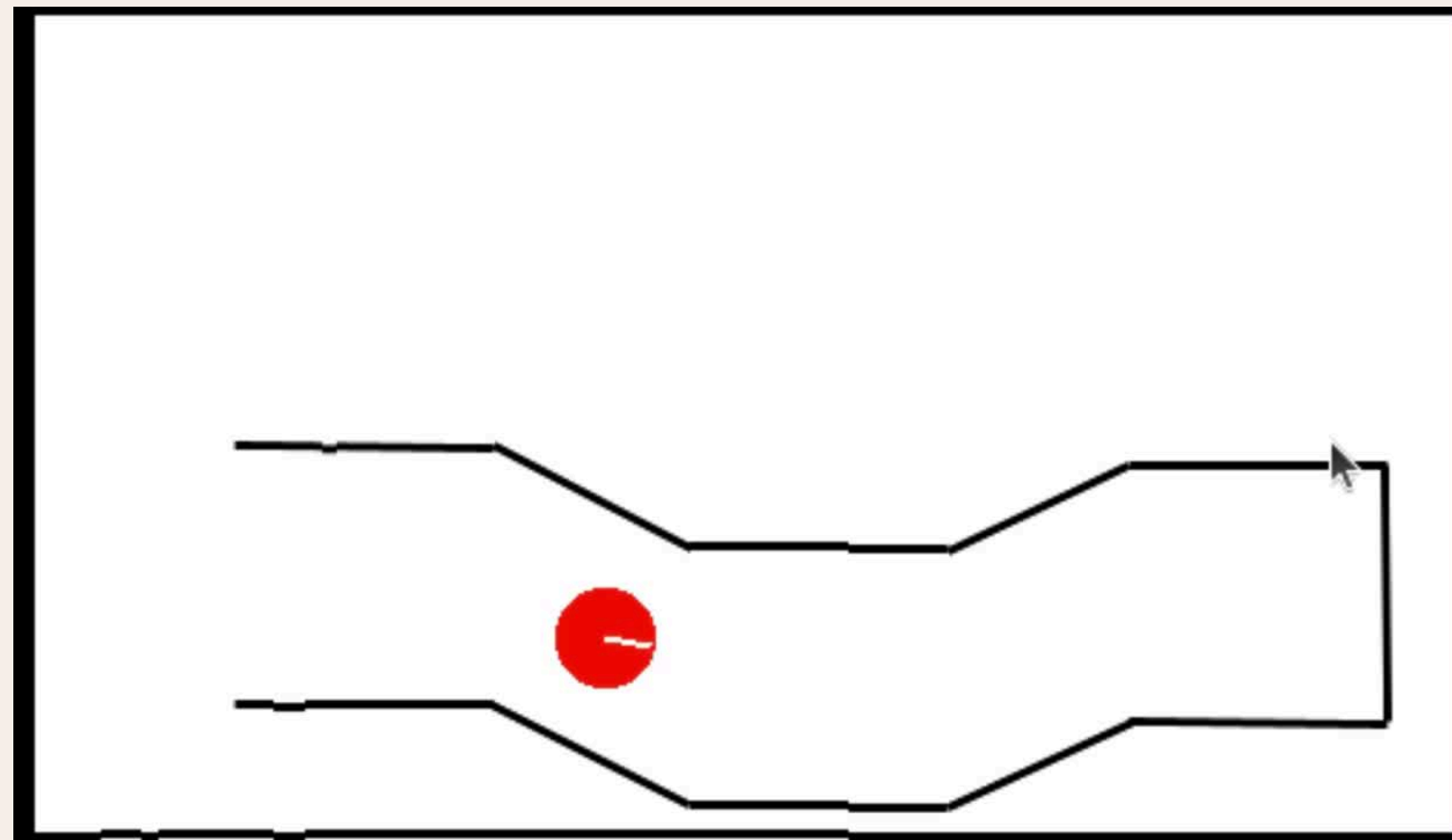
wall_distance

- Suscrito a "camera/depth/image_raw" (kobuki).
- Divide la imagen en tres zonas, pero solo ocupa la izquierda y derecha, no la central.
- Dividir en dos mitades daba peores resultados
- Encuentra el punto más cercano (nanmin) en cada zona
- diferencia = izquierda - derecha. Esta es la variable de control.
- Publica la información en el topic "state".



wall_dodger_p_controller

- Suscrito a "state". Recibe la diferencia.
- Multiplica por $k_p = 1.4$
- Publica la acción de control en "control_action"

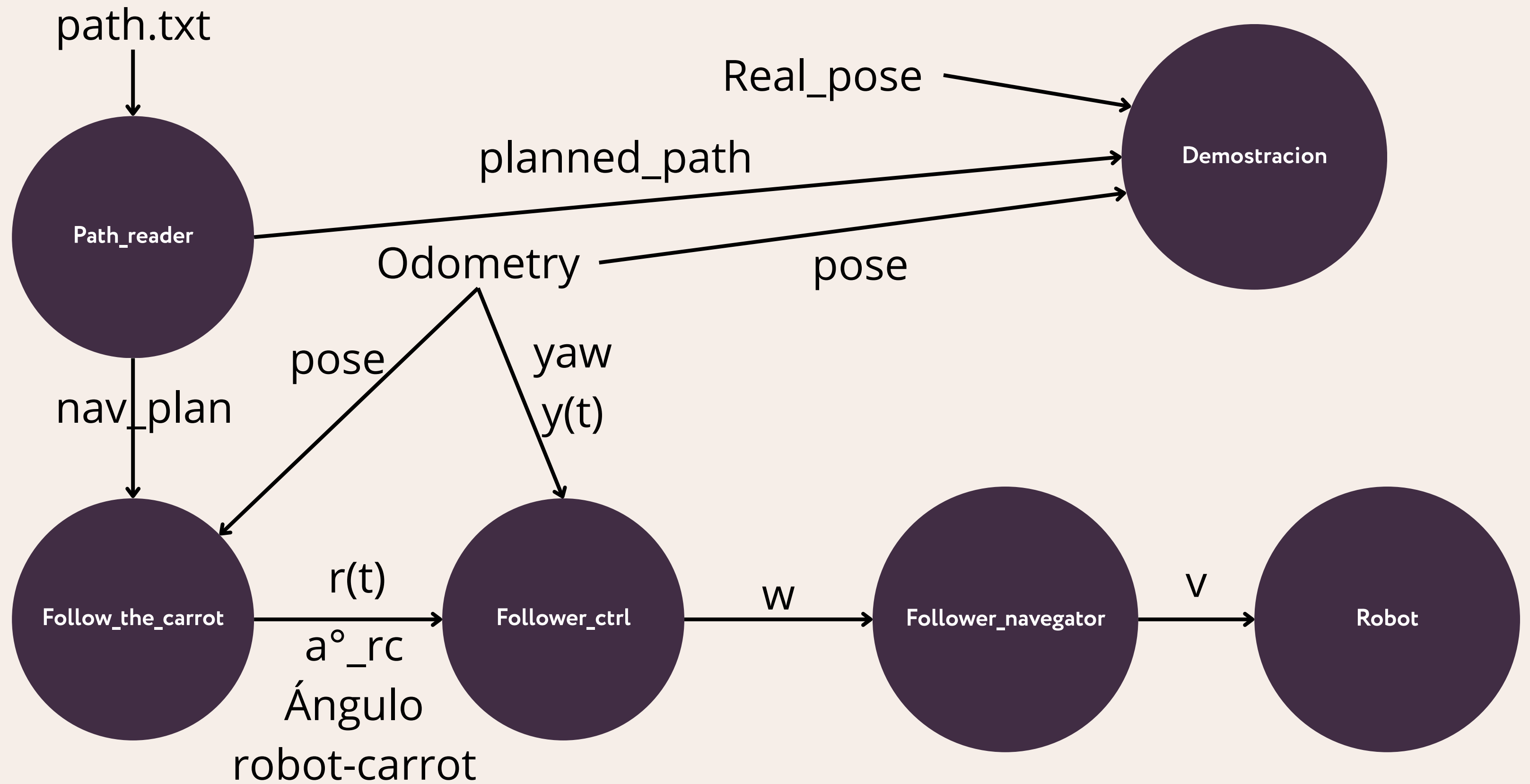


p_navigation_node

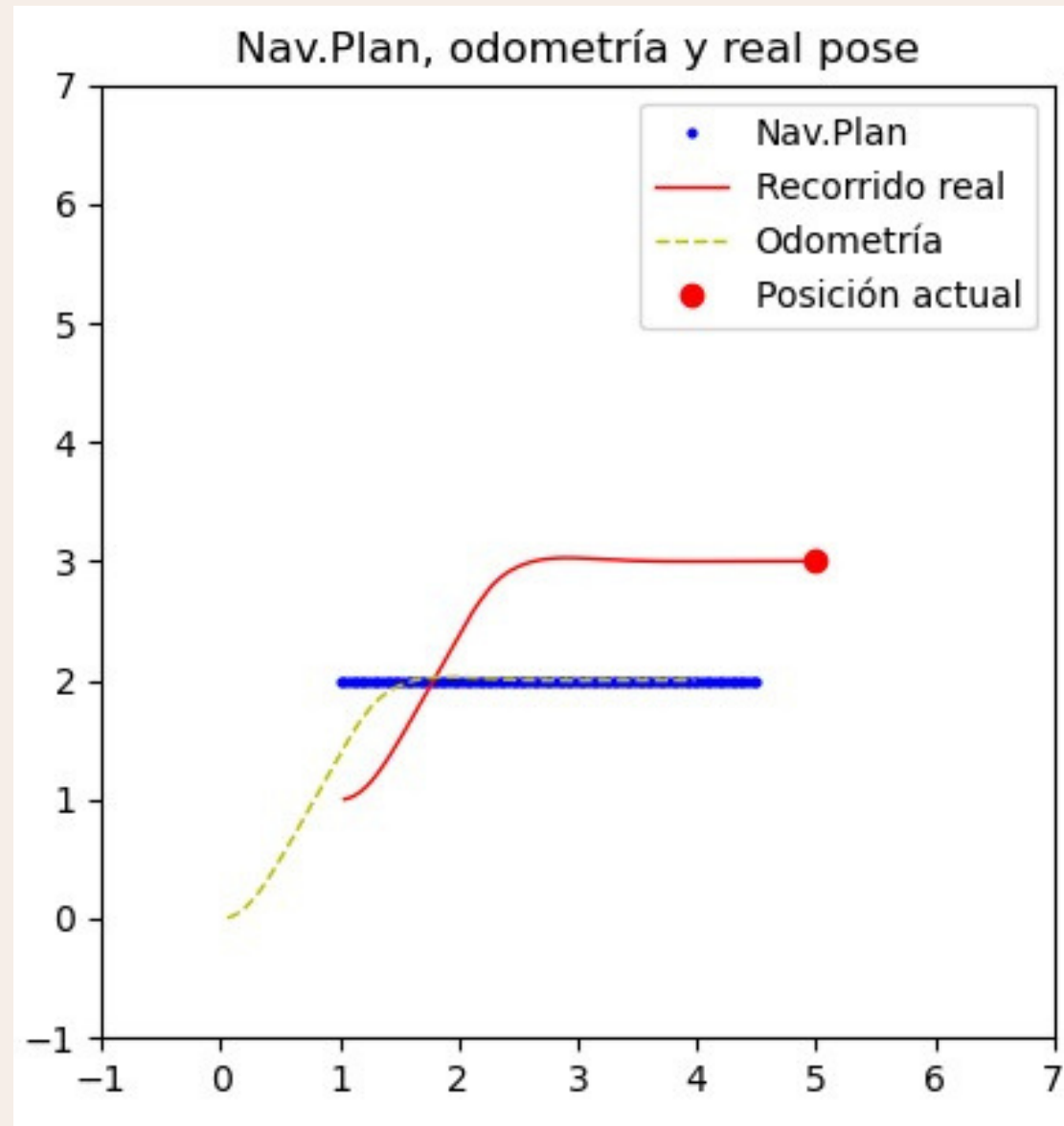
- Suscrito a “control_action”.
- Crea un Twist, con velocidad lineal constante = 0.2m/s
- Velocidad angular inicia en 0, se modifica con la callback
- `self.speed.angular.z = accion.data`, siempre y cuando no sea mayor a 1rad/s (máxima).
- Publica Twist en “/cmd_vel_mux/input/navigation”

Parte 3

Nodos



Path_line



El error cuadrático medio fue de:

$$\text{ECM}_x = 0.7473$$

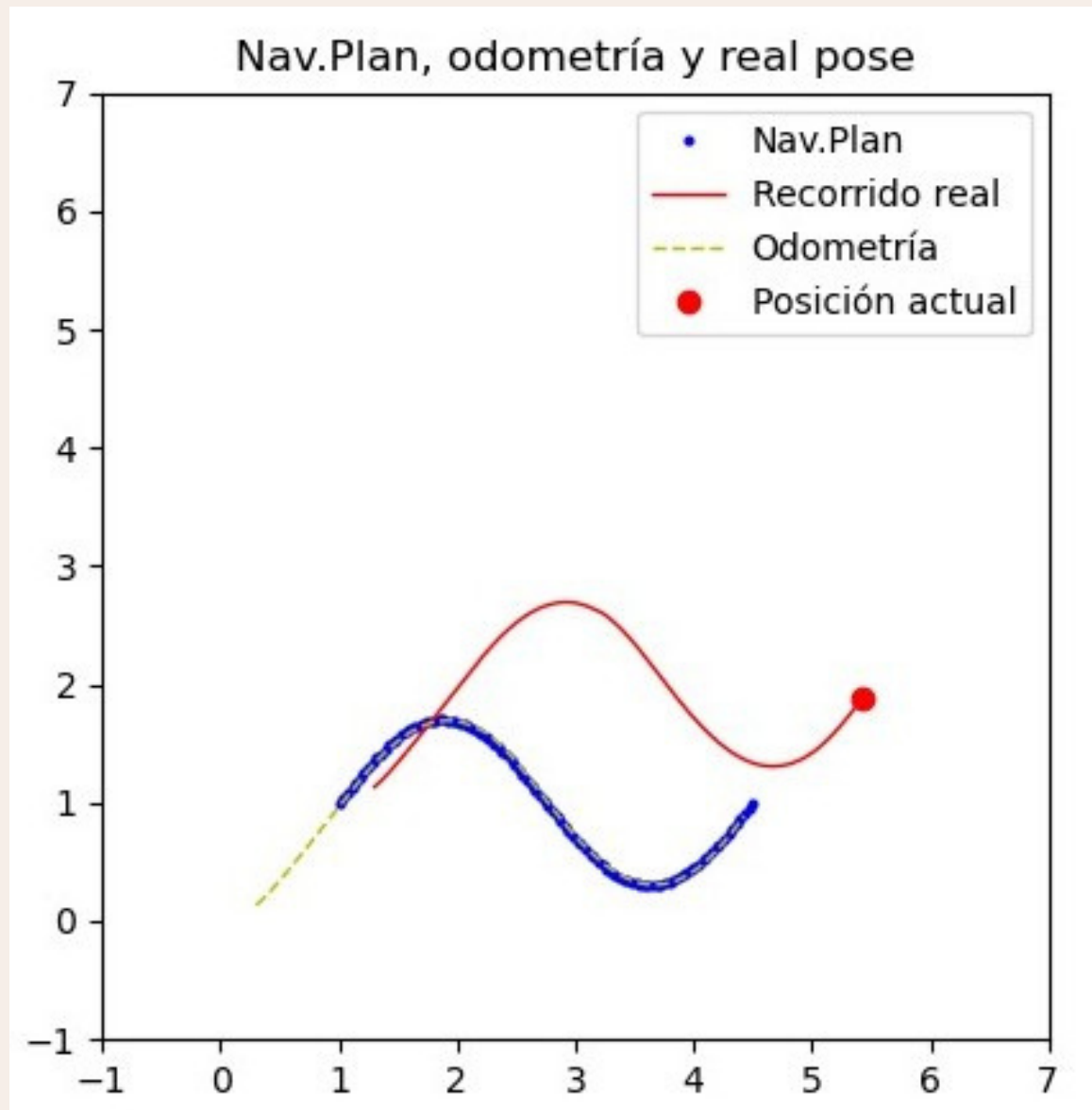
$$\text{ECM}_y = 0.7391$$

Path_sin

El error cuadrático medio fue de:

$$\text{ECM}_x = 0.3659$$

$$\text{ECM}_y = 1.5508$$



Path_sqrt

El error cuadrático medio fue de:

$$\text{ECM}_x = 0.5127$$

$$\text{ECM}_y = 0.1214$$

