



Actividad 5.1 (Control de Posición)

Carlos Adrián Delgado Vázquez A01735818

12 de abril del 2025

Introducción

El presente documento tiene como objetivo describir el proceso de implementación de un controlador de posición para un robot móvil diferencial. Se busca que el robot alcance una serie de coordenadas deseadas a partir de una posición inicial fija, usualmente el origen. Posteriormente, se plantea una estrategia para la creación de un controlador auto-sintonizable, capaz de ajustar sus ganancias en función del desempeño del sistema.

Desarrollo del Controlador

Modelo Cinemático

Para el diseño del controlador se parte del modelo cinemático de un robot móvil diferencial, el cual está dado por las siguientes ecuaciones:

$$\begin{cases} \dot{x} = v \cos(\phi) \\ \dot{y} = v \sin(\phi) \\ \dot{\phi} = \omega \end{cases}$$

donde x y y son las coordenadas del robot en el plano, ϕ es el ángulo de orientación respecto al eje horizontal, v representa la velocidad lineal y ω la velocidad angular.

Este modelo es adecuado para representar el comportamiento de plataformas móviles de dos ruedas, donde el movimiento está restringido a una dirección principal determinada por el ángulo ϕ .

Ley de Control

Se emplea un controlador proporcional en espacio cartesiano, basado en la retroalimentación del error de posición. Para transformar el error en coordenadas cartesianas al marco de referencia del robot, se utiliza la pseudoinversa del Jacobiano de orientación, definido como:

$$J(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}$$

A partir de esta matriz, la ley de control implementada es la siguiente:

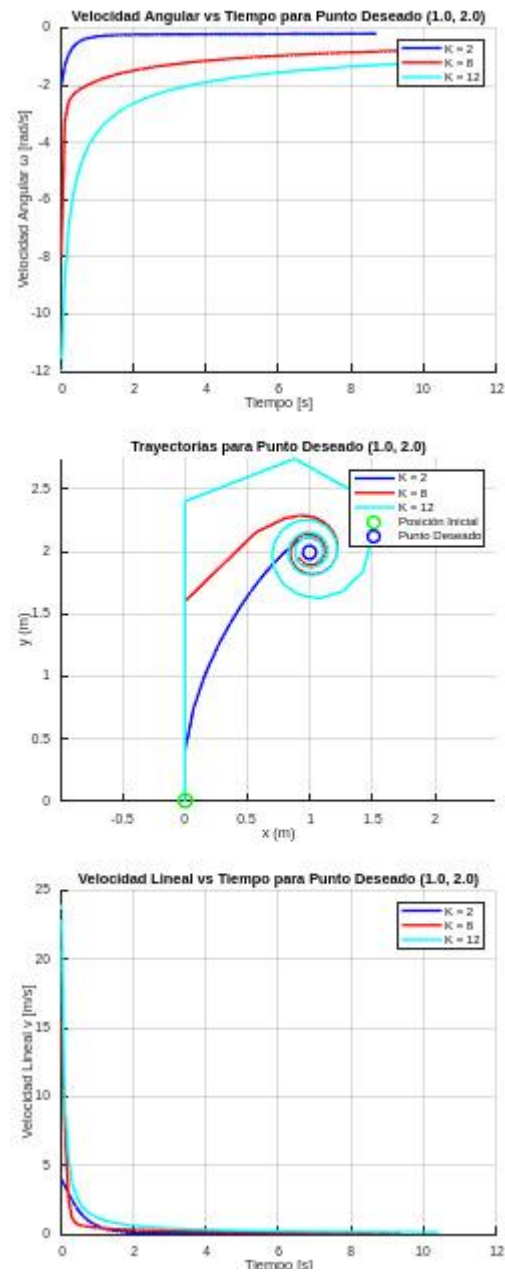
$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J^{-1}(\phi) \cdot K \cdot \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix}$$

donde K es una matriz diagonal de ganancias proporcionales, y (x_d, y_d) representa la posición deseada a alcanzar.

Para evaluar el comportamiento del sistema, se implementa un lazo de simulación que permite al robot alcanzar múltiples puntos deseados partiendo siempre desde el origen. Una vez que el robot llega a un punto con un error menor a un umbral definido (por ejemplo, 0.1 metros), se termina la simulación de esa trayectoria y se inicia una nueva hacia el siguiente punto.

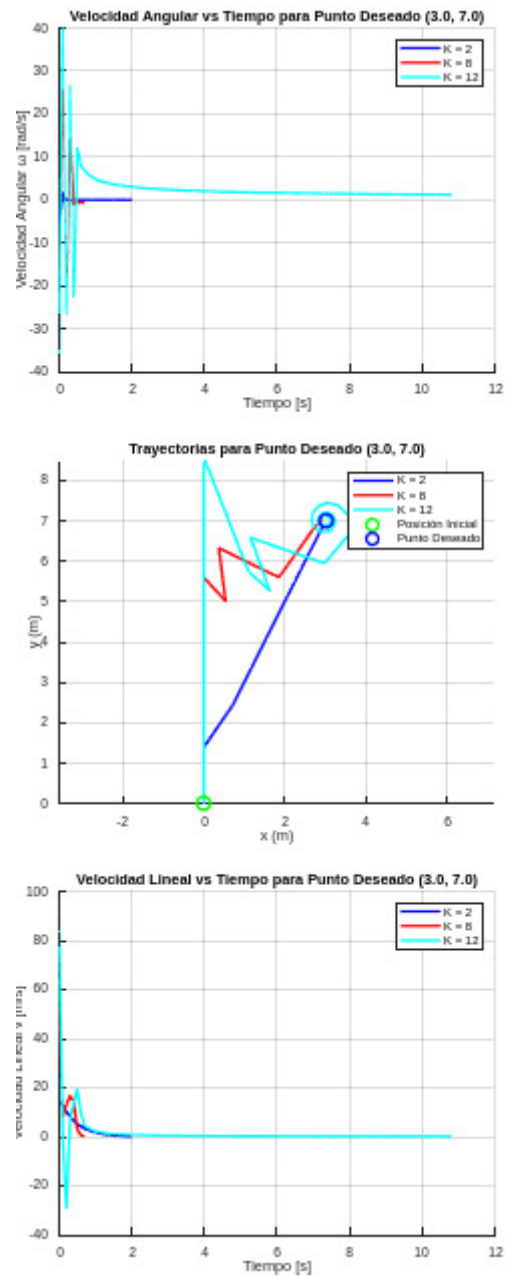
Asimismo, se comparan los resultados de diferentes valores de la matriz de ganancias K , con el fin de observar su efecto en el tiempo de respuesta, precisión y suavidad del movimiento.

a) (1, 2)



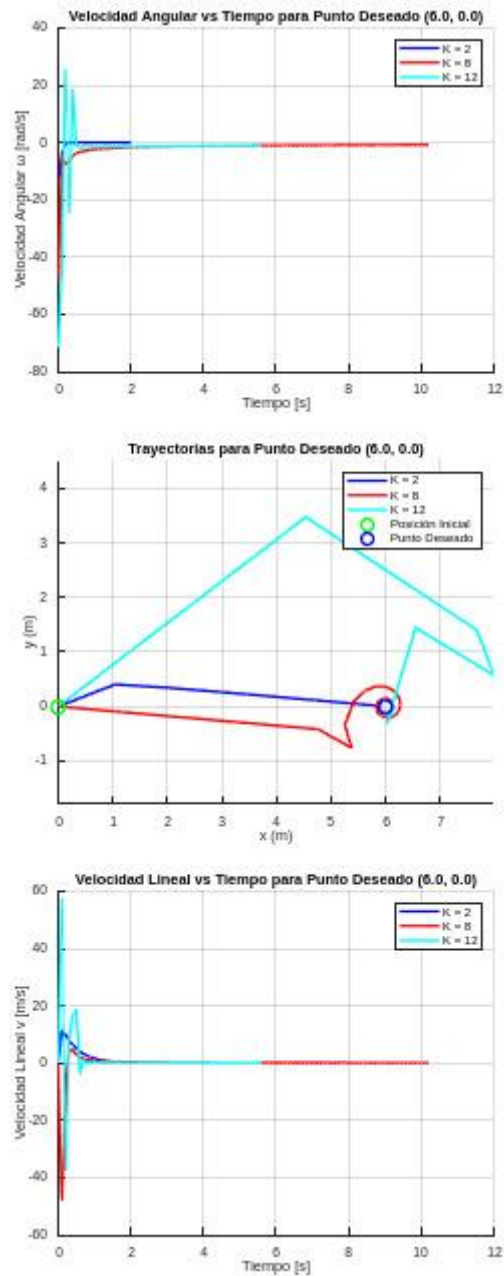
Comparando las 3 ganancias distintas, podemos ver que con una ganancia mayor, tarda demasiado en llegar al punto, por lo que la mejor sería la de $k=2$.

b) (3, 7)



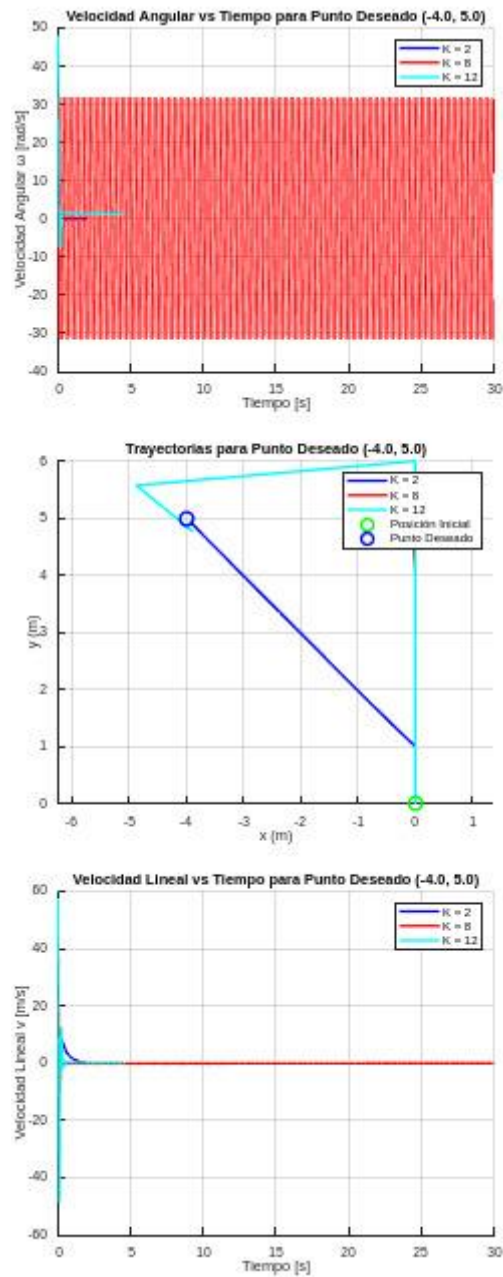
En este caso, podemos ver como la trayectoria más rápida la genera la $k=8$, por lo que podría ser la más óptima.

c) (6, 0)



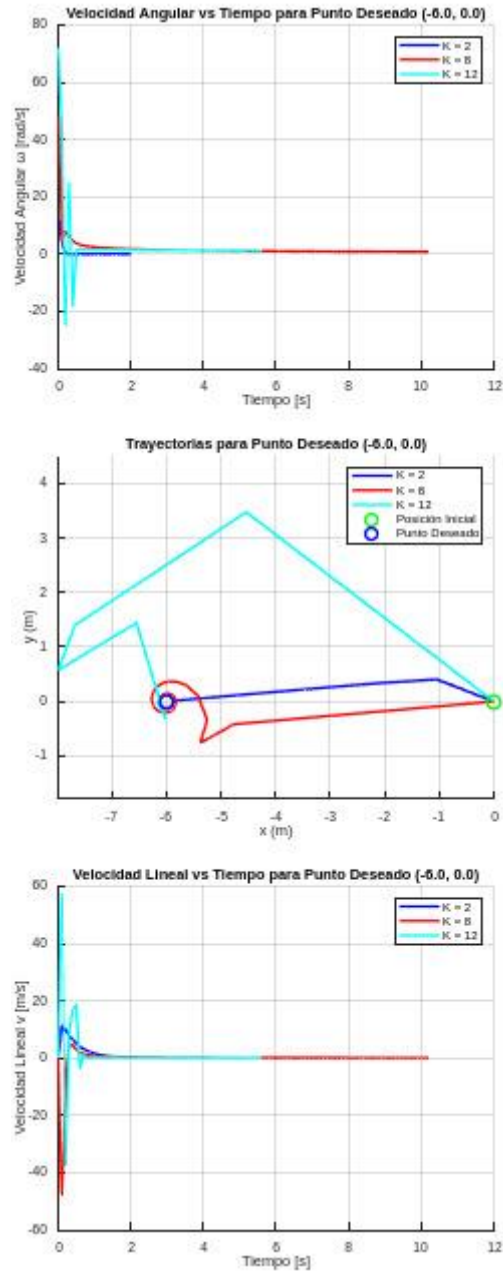
En este caso, no hay duda de que la ganancia de $k=2$ es la mejor, pues es una ruta más directa y rápida para llegar al punto.

d) (-4, 5)



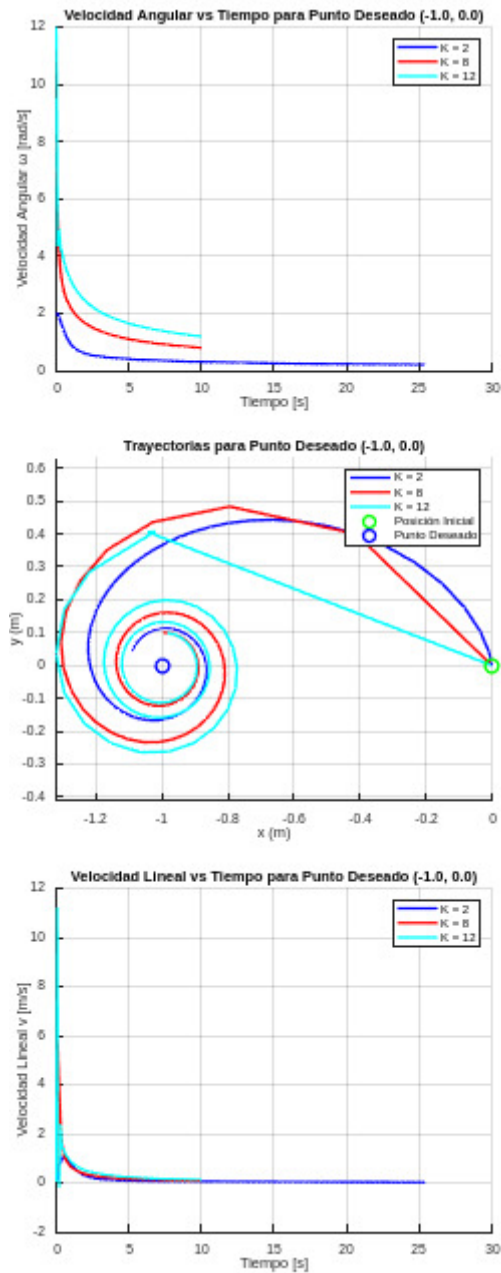
Comparando las 3 ganancias distintas, podemos ver que con una ganancia mayor, tarda demasiado en llegar al punto, o en el caso del $k=8$, ni siquiera llega y entra en bucle, por lo que la mejor sería la de $k=2$.

e) $(-6, 0)$



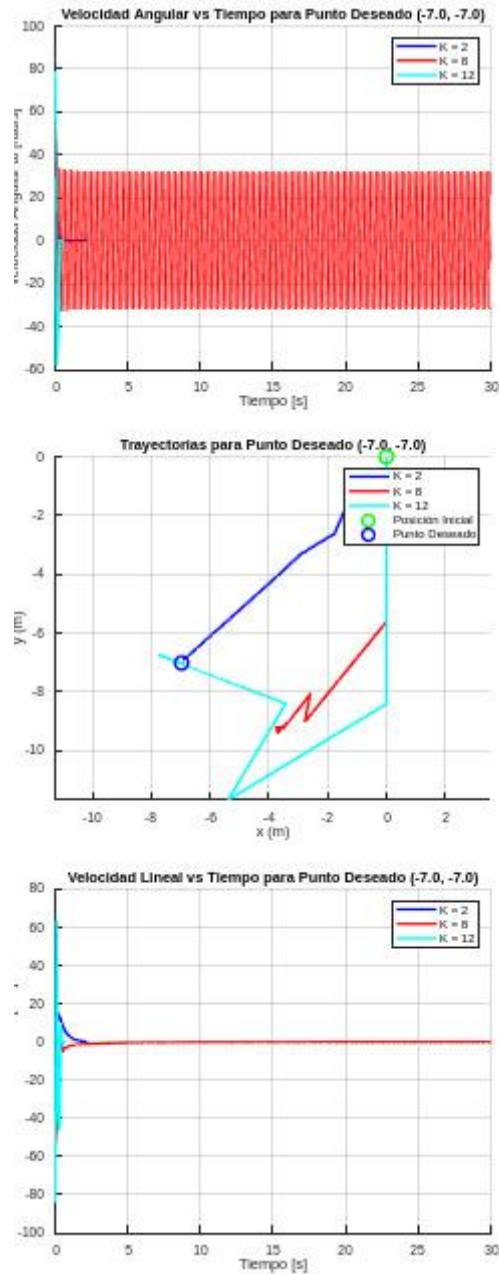
En este caso, no hay duda de que la ganancia de $k=2$ es la mejor, pues es una ruta más directa y rápida para llegar al punto.

f) $(-1, 0)$



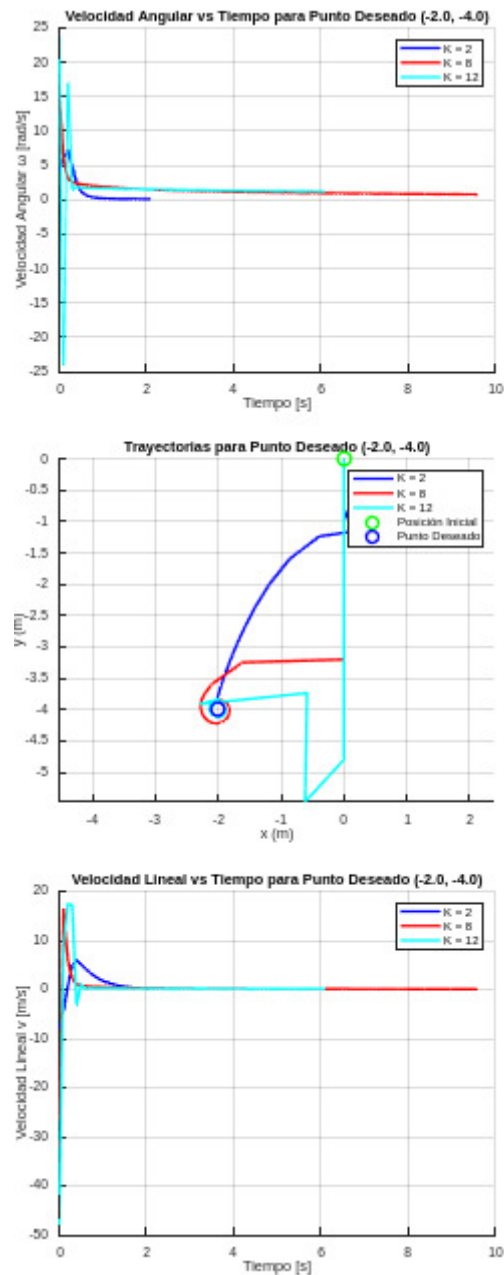
En este caso, notamos que las ganancias más grandes, nos dan rutas más rápidas, y con una ganancia más chica, empieza dar demasiadas vueltas. En este caso elegiremos la de $k=8$, ya que traza una trayectoria más suavizada que con 12, sin embargo, el tiempo de llegada al punto es el mismo.

g) (-7, -7)



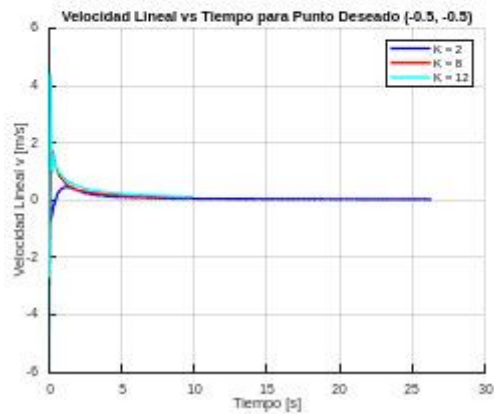
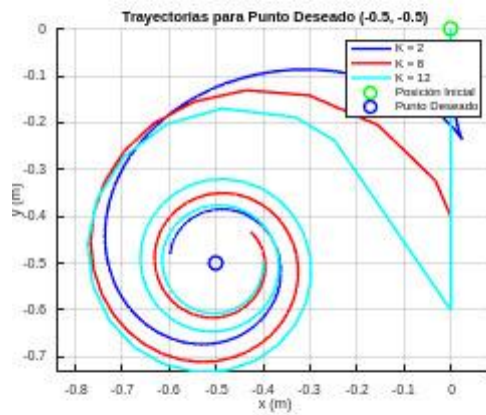
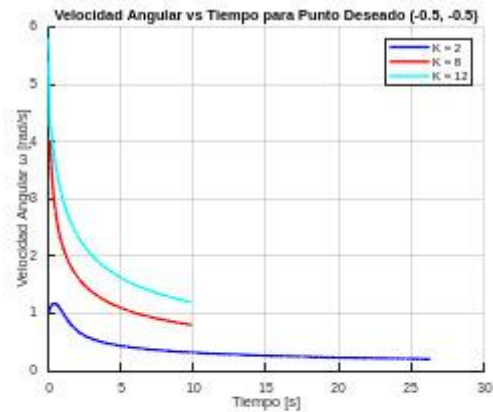
En este caso, no hay duda de que la ganancia de $k=2$ es la mejor, pues es una ruta más directa y rápida para llegar al punto. La de ganancia 12 también llega al punto, sin embargo, tiene cambios más bruscos en la trayectoria.

h) (-2, -4)



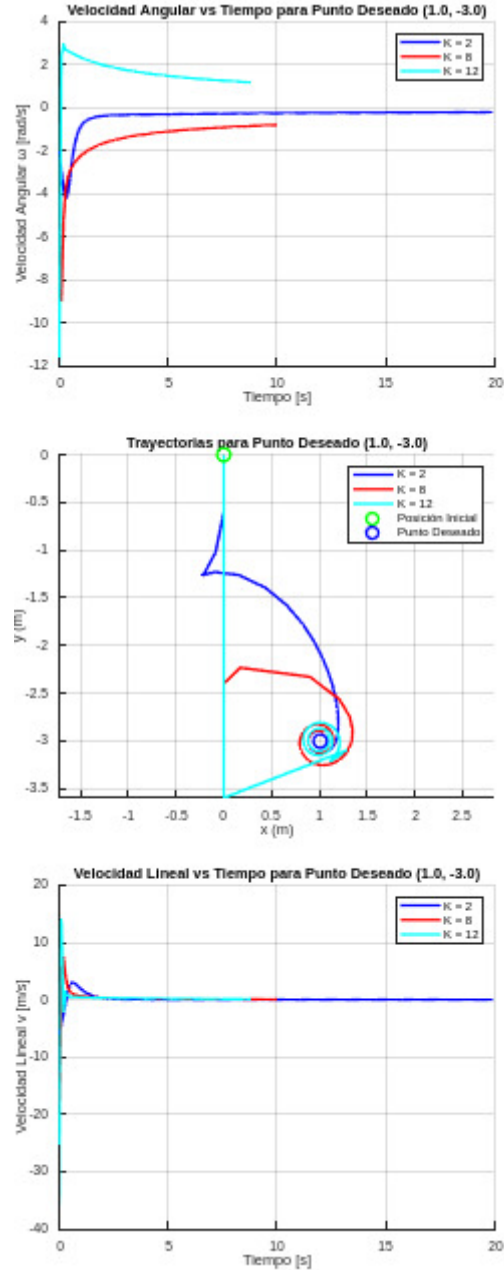
En este caso, es muy parecido al anterior, no hay duda de que la ganancia de $k=2$ es la mejor, pues es una ruta más directa y rápida para llegar al punto. Las otras ganancias llegan al punto, pero dando más vueltas y en más tiempo.

i) (- 0.5, -0.5)



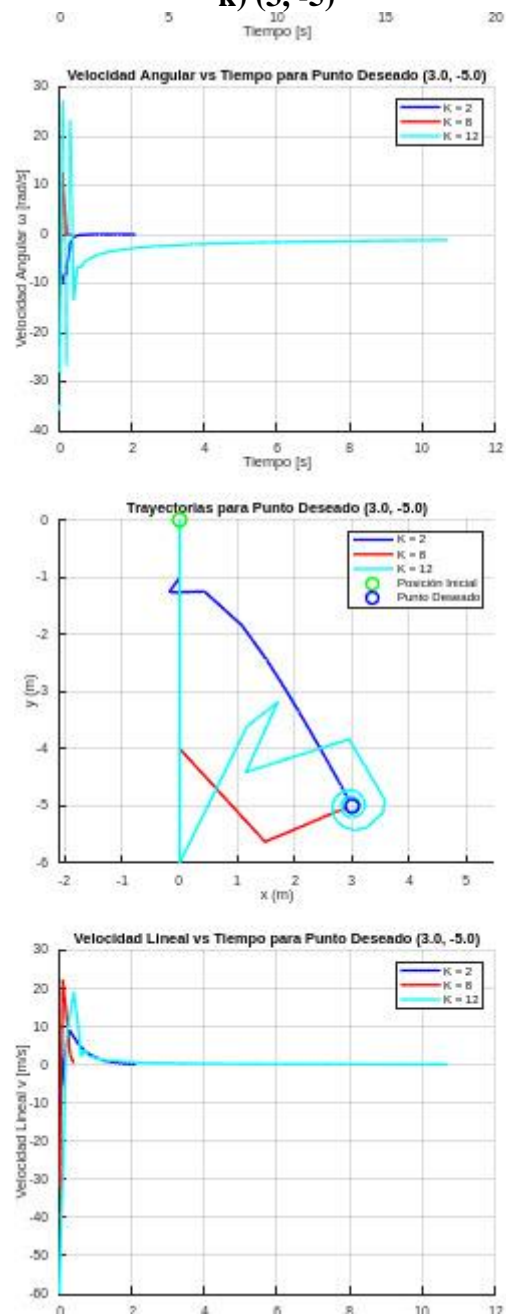
En este caso, notamos que las ganancias más grandes, nos dan rutas más rápidas, y con una ganancia más chica, empieza dar demasiadas vueltas. En este caso elegiremos la de $k=8$, ya que traza una trayectoria más suavizada que con 12, sin embargo, el tiempo de llegada al punto es el mismo.

j) (1, -3)



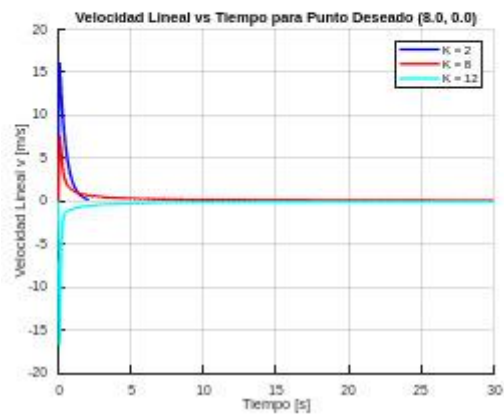
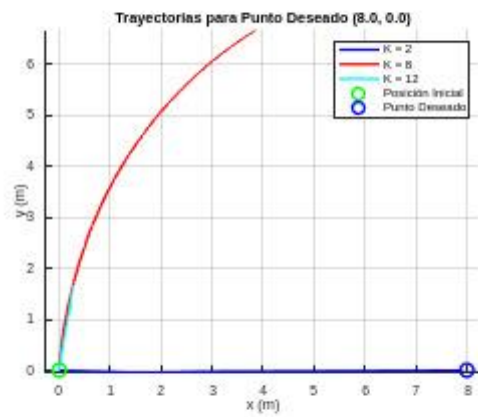
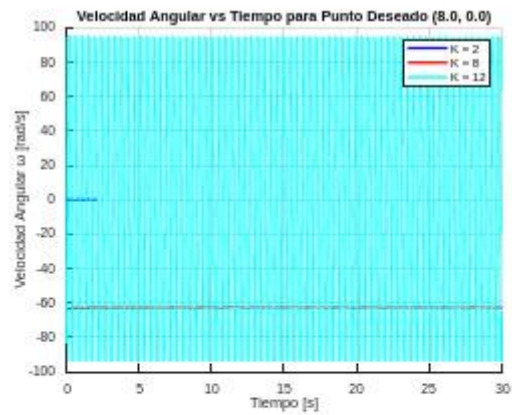
En este caso, notamos que las ganancias más grandes, nos dan rutas más rápidas, y con una ganancia más chica, empieza dar demasiadas vueltas. En este ejemplo, una ganancia de 12, nos garantiza una ruta más rápida que con las de menor ganancia, por la que podríamos elegir esta ganancia.

k) (3, -5)

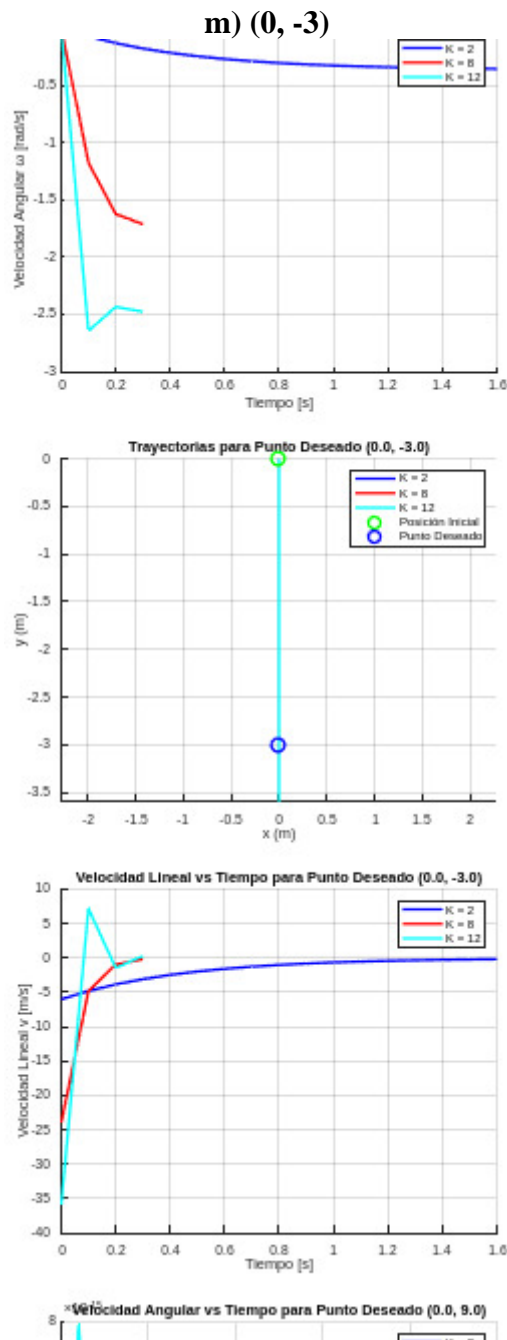


Ahora notamos que la vía más rápida es la de ganancia 8, la cual es óptima, ya que es rápida y no considera tantos cambios bruscos.

1) (8, 0)

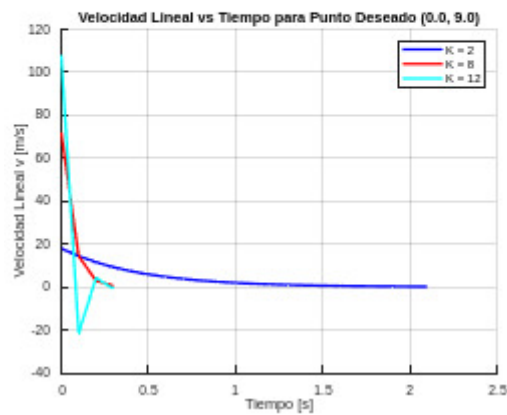
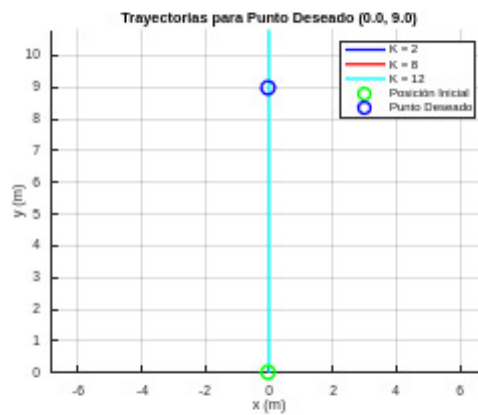
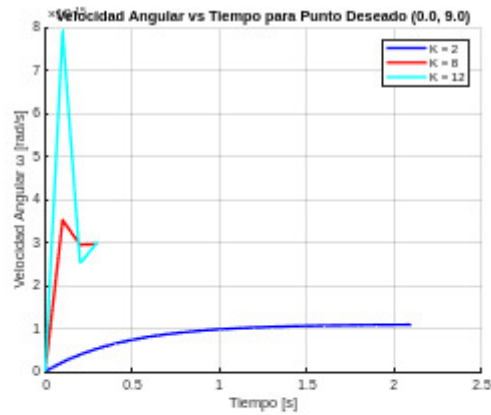


Aquí podemos ver que la única trayectoria que funciona es la de ganancia 2, ya que las otras, ni si quiera llegan al punto de objetivo.



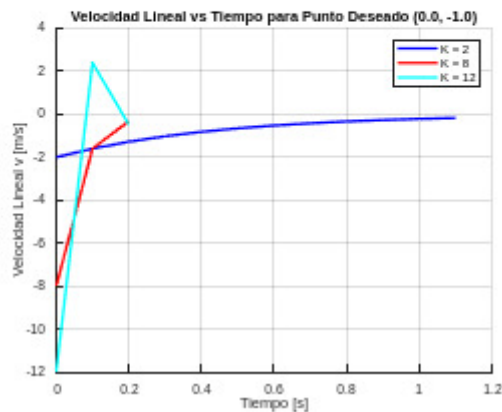
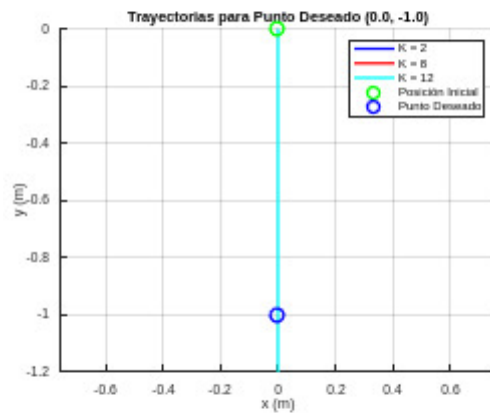
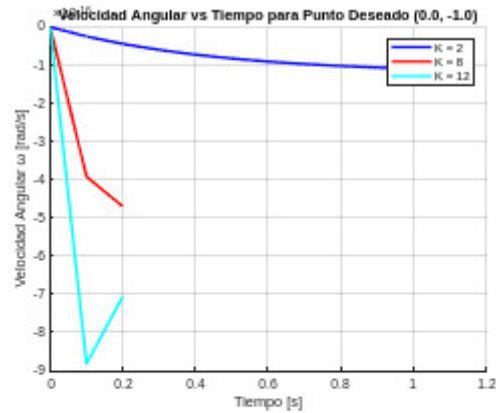
En este, podemos ver que prácticamente llegan al mismo tiempo al punto con las ganancias de 8 y 12, sin embargo, elegiremos la de 8, ya que parece ser que hay un pequeño sobretiro en el de 12 que no se ve en la de 8.

n) (0, 9)



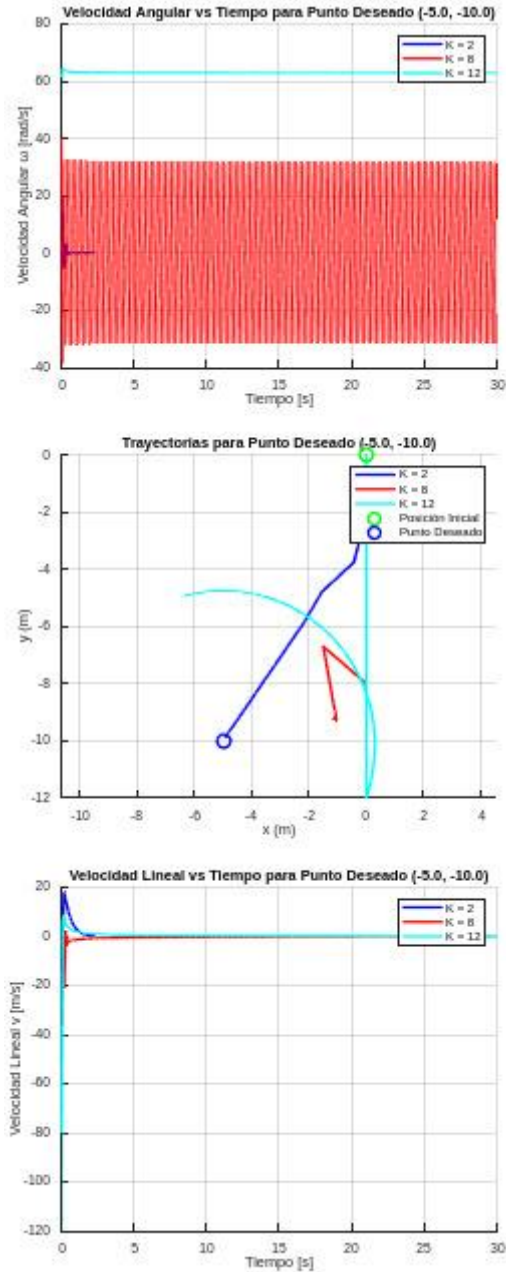
Pasa lo mismo que el anterior, podemos ver que prácticamente llegan al mismo tiempo al punto con las ganancias de 8 y 12, sin embargo, elegiremos la de 8, ya que parece ser que hay un pequeño sobretiro en el de 12 que no se ve en la de 8.

$\hat{n}) (0, -1)$



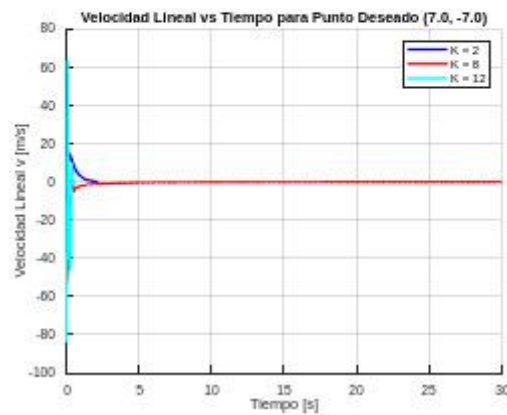
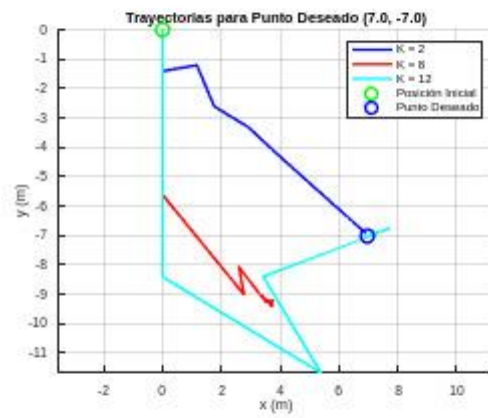
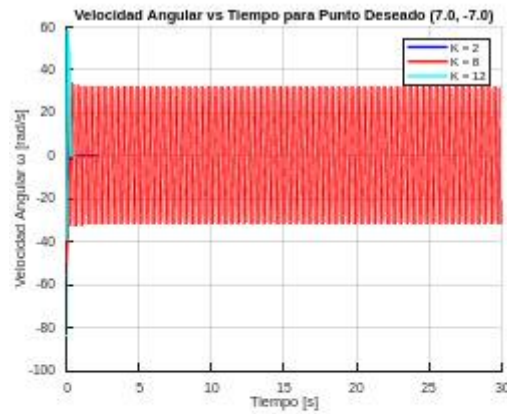
Pasa lo mismo que el anterior, podemos ver que prácticamente llegan al mismo tiempo al punto con las ganancias de 8 y 12, sin embargo, elegiremos la de 8, ya que parece ser que hay un pequeño sobretiro en el de 12 que no se ve en la de 8.

o) (-5, -10)



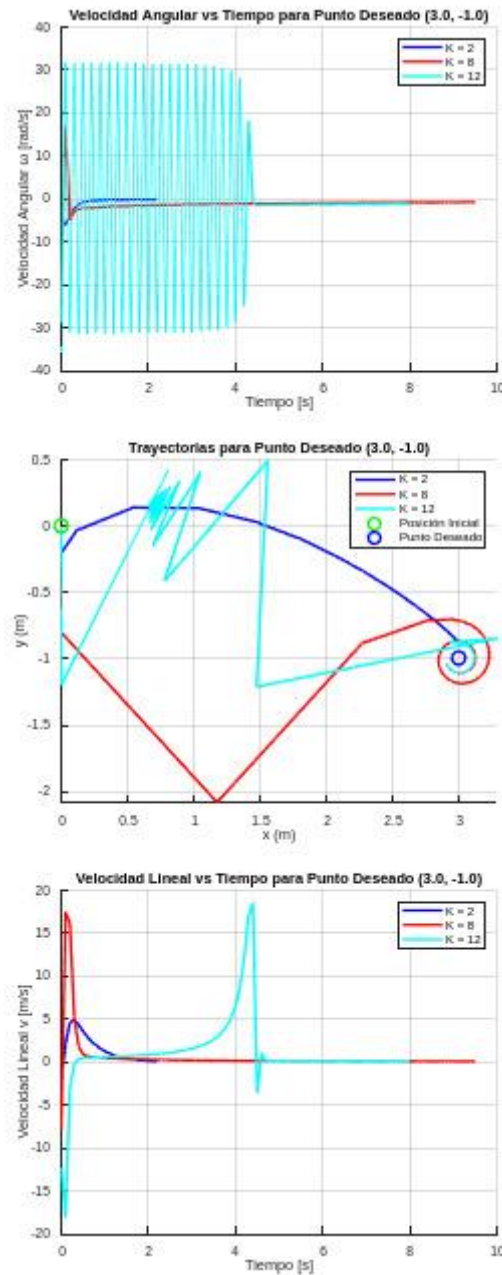
En este caso es definitivo la ganancia de 2, ya que las otras ganancias no encuentran el punto en el tiempo de simulación y la ganancia chica la encuentra de manera rápida y eficiente.

p) (7, -7)



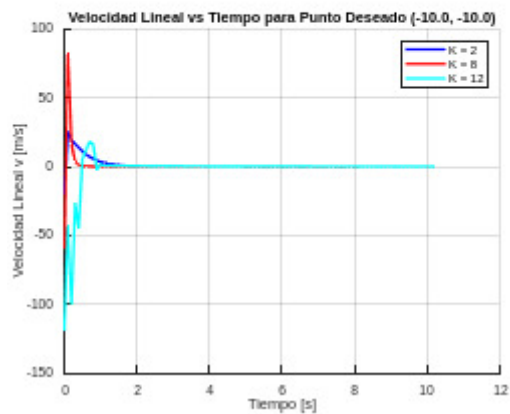
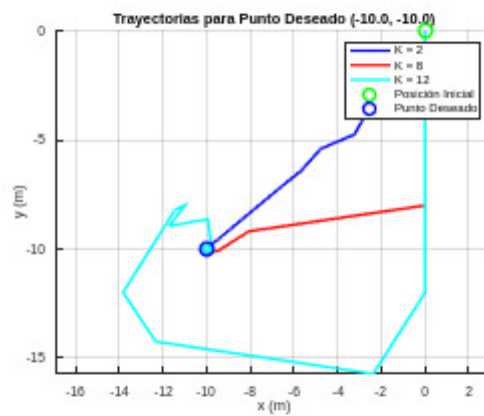
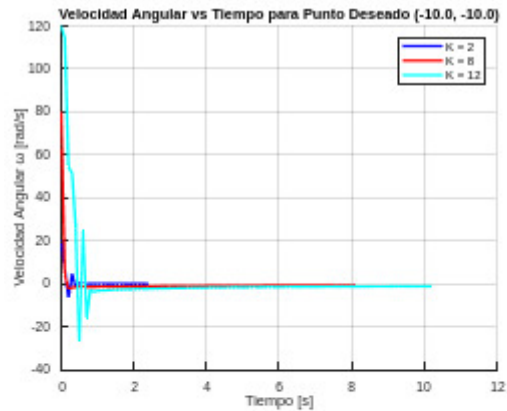
Podemos decir, que el más eficiente en este caso es la ganancia de 2, ya que, a pesar, de que la ganancia de 12 lo logra más rápido, tiene un sobretiro excesivo de hasta 60 rad/s, lo cual haría imposible reaccionar el robot.

q) (3, -1)



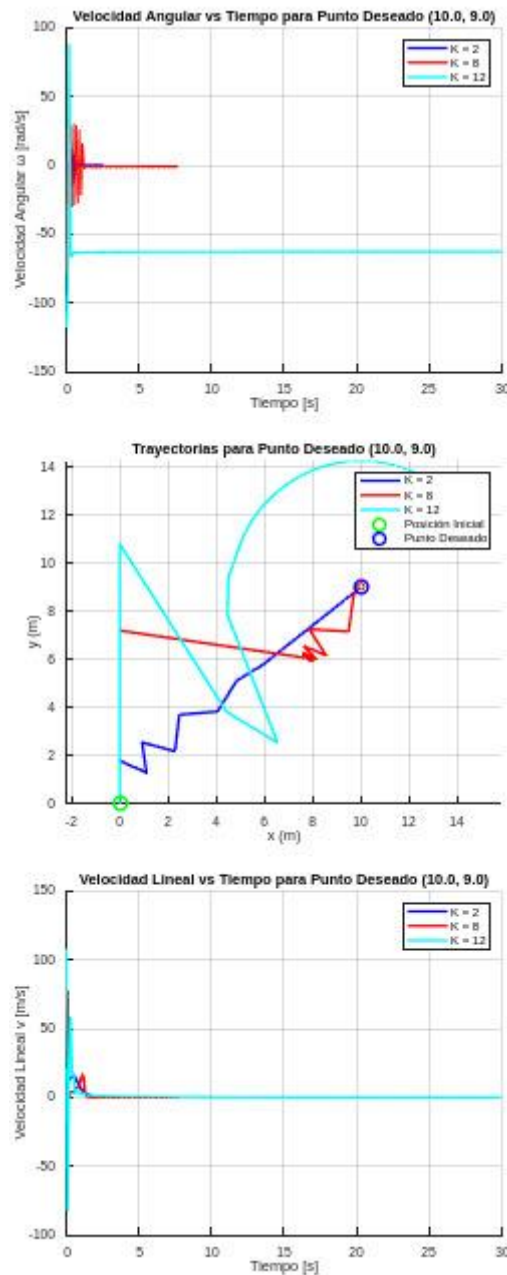
En este caso, las 3 ganancias encuentran su ruta, sin embargo, la que lo hace más rápido y eficiente es la de ganancia 2.

r) (- 10, -10)



Pasa lo mismo que en el anterior, las 3 ganancias encuentran su ruta, sin embargo, la que lo hace más rápido y eficiente es la de ganancia 2.

s) (10, 9)



Y finalmente, ocurre lo mismo que los otros 2, las 3 ganancias encuentran su ruta, sin embargo, la que lo hace más rápido y eficiente es la de ganancia 2.

Estrategia para Controlador Auto-Sintonizable

Para ello, se considera una estrategia basada en la evaluación continua del error de posición y su derivada. Una opción es implementar una lógica de adaptación en la cual, si el error no disminuye con suficiente rapidez o si aparecen oscilaciones, el sistema incrementa automáticamente la ganancia proporcional. Alternativamente, si el error converge de forma

eficiente pero hay picos de velocidad angular, puede optarse por reducir ligeramente la ganancia para suavizar la trayectoria. Esto creo que lo podemos alcanzar con un controlador PID, que analice el error pasado, presente y futuro y nos ayude a calcular las ganancias de la manera más eficiente.

Conclusión

El controlador propuesto permitió al robot móvil alcanzar múltiples coordenadas deseadas de forma exitosa. Pude darme cuenta que no siempre las mismas ganancias servían, pero en la mayoría de los casos, las ganancias pequeñas son las que encuentran mejores rutas, aunque en ciertas ocasiones, las ganancias “medias” o “no tan grandes”, son muy eficientes para hacer un trabajo más rápido. Considero que el controlador con el auto-tune de las ganancias, sería una gran implementación, para que el robot siempre encuentre la ruta más eficiente y no dependa del punto a donde llegue solamente.