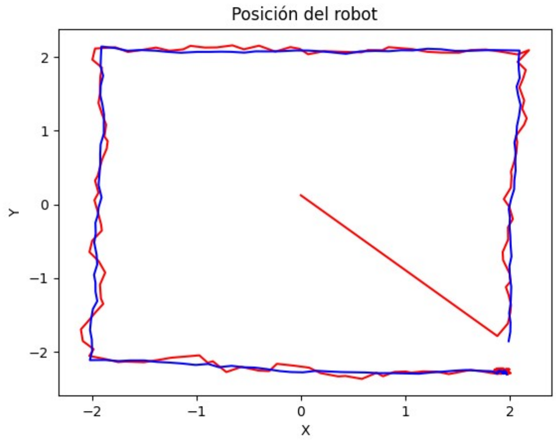
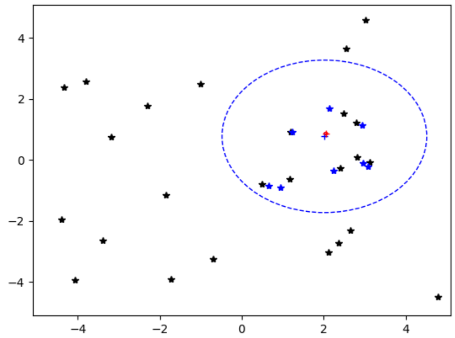
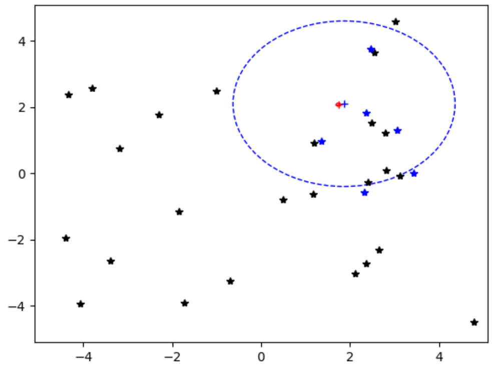
Practica 5: Localización de un Robot con Filtro de Kalman Extendido

A1. Apenas tarda unos pocos pasos en estimar la posición del robot. En la imagen vemos la trayectoria del robot real y la estimación:



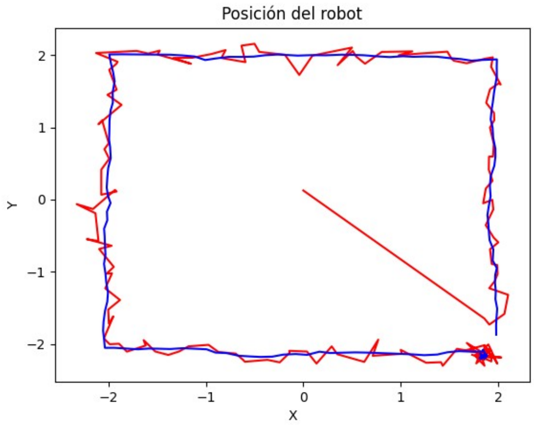
A2. En este caso, a pesar de reducir la covarianza del error de medida sigue tardando bastante poco en determinar la posición correcta del robot. En este caso, al reducir la covarianza, estamos haciendo que la posición que estima tiene una elipse de error más pequeña (debido a reducir la covarianza) por lo que hay veces que el robot real no se encuentra dentro del rango definido por la gaussiana, debido a que esta es muy pequeña y un mínimo error hace que el robot real ya no esté dentro del área estimada (aunque cuando ocurre esto la distancia a la que se encuentra la posición estimada es muy cercana a la posición real).



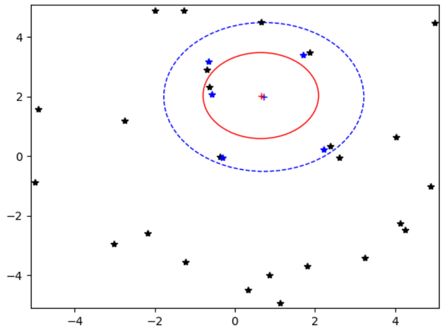


En la primera imagen, la estimación coincide con la posición real del robot y en la segunda no (si la convarianza fuese mayor, la posición real del robot estaría dentro de la gaussiana).

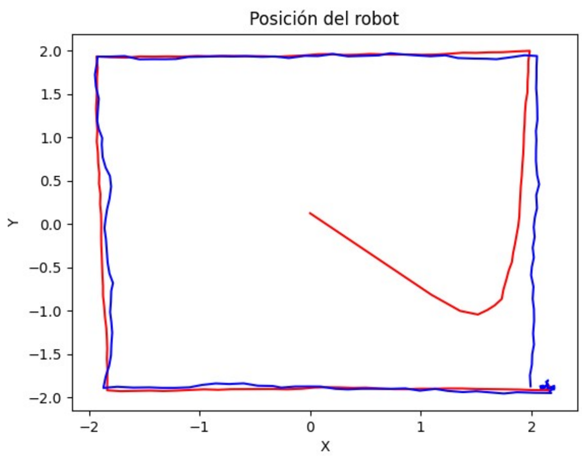
Si nos fijamos en la trayectoria del robot y de la estimación vemos que tiene que hacer muchas correcciones durante la trayectoria (aún así estima la posición al inicio de moverse el robot)



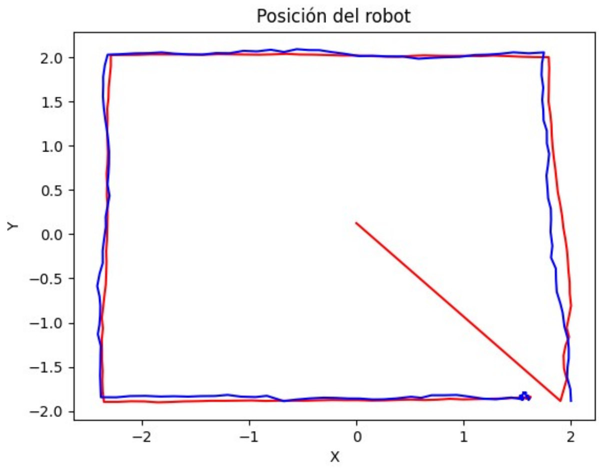
A3. En este caso, llega a estimar correctamente la posición real del robot pero tarda muchos más que antes, esto es debido a que el error de medida del robot es demasiado grande. Vemos que ahora la elipse de covarianza es mucho mayor (fruto de la modificación hecha).



En la siguiente imagen vemos la estimación de la posición frente a la posición real.

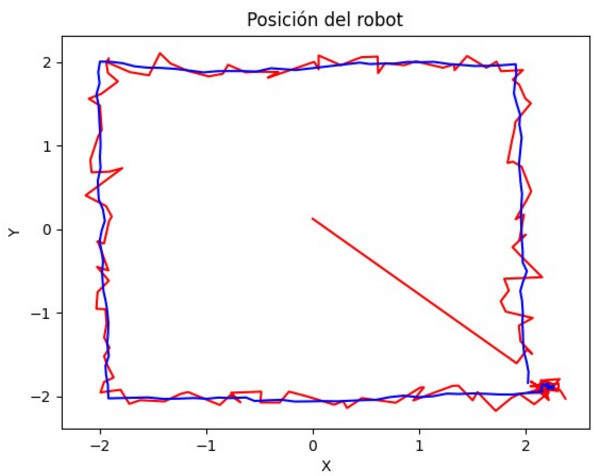


A4. Si multiplicamos el ruido de estado por 0.01 obtenemos el siguiente comportamiento:



Ahora al no cambiar el ruido de medida, la estimación de la posición no hace tantas correciones como en el caso anterior.

Si multiplicamos el ruido de estado por 100 obtenemos la siguiente trayectoria:



Ahora la estimación hace más correciones sobre la posición.

Tras esto podemos ver que cambiando la covarianza del ruído de medida parece que influye más con respecto al ruído de estado. Si comparamos el primer ejemplo (ruído de medida*0.01) con el último (ruído de estado*100) vemos que los comportamientos son bastante parecidos, y lo mismo para ruído de medida*100 y ruído de esto por 0.01, con la diferencia de que en el ejemplo de ruído de medida*100 la estimación tarda más pasos en lograr acercarse a la posición real del robot.

A5. Al reducir el número de landmarks a 5, en la mayor parte del mapa el robot no detectará ninguno, por lo que el algoritmo no podrá actualizar y solo podrá predecir, haciendo que el error de estimación aumente en gran medida, y que cuando no hay landmarks la estimación no se corresponde con la posición real del robot.

A6. En cuanto al tiempo que tarda en estimar la posición, apenas varía con respecto a la implementación cartesiana. Sin embargo, la elipse de covarianza vemos que cambia, en el modelo cartesiana mantiene siempre la misma proporción, sin embargo en el método range and bearing la elipse se orienta hacia los landmarks. Esto es debido a que ltiene una estimación del ángulo en el que se encuentra el landmark mucho más segura con respecto a la distancia que hay entre el landmark y el robot.

A7. Si el robot no detecta ningún landmark no puede actualizar la posición, solo predecir. Sin embargo el error no aumenta tanto como en el modelo cartesiano. En el cartesiano cuando no detectaba ningún landmark en pocos pasos la estimación ya no coincidía con la posición real, sin embargo con range and bearing la estimación sigue manteniendo la posición real dentro de la gaussiana.

Al detectar solo un landmark vemos que aparece lo explicado en la pregunta anterior, la elipse se alinea con este landmark ya que tiene menos error en la estimación del ángulo con respecto al robot que de la distancia del landmark al robot.