

PROGRAMACIÓN DE ROBOTS

Programación de robots móviles PRÁCTICA 6.- EKF para range bearing usando MRPT

El objetivo de la práctica consiste en realizar una localización *range bearing* usando un EKF a partir de las clases proporcionadas por la MRPT.

1.- Análisis del problema

El alumno podrá descargar de la web un fichero comprimido con el código necesario para realizar la práctica (*CmakeLists.txt* y *test.cpp*). El código del fichero .cpp se divide en 5 apartados :

A) Definición de variables

B) Definición de la clase *CRangeBearing* a partir de la clase *CKalmanFilterCapable* ya existente en la MRPT

C) Implantación de la función *TestBayesianTracking*, que realiza las siguientes tareas:

- Creación de la ventana gráfica.
- Creación de un EKF a partir de la clase *CRangeBearing*.
- Simulación del movimiento del vehículo, según este modelo cinemático que el alumno **debe completar**:
 - $x = x + v \cdot \Delta t \cdot (\cos \phi - \sin \phi)$
 - $y = y + v \cdot \Delta t \cdot (\sin \phi + \cos \phi)$
 - $\phi = \phi + w \cdot \Delta t$
 - $v = v + 1 \cdot \Delta t \cdot \cos t$
 - $w = w - 0.1 \cdot \Delta t \cdot \sin t$
- Simulación de las observaciones del sensor con ruido.
- Procesamiento de los datos.
- Impresión de los datos por pantalla y ventana gráfica.

D) Implantación de la función *main*, encarga de llamar a *TestBayesianTracking*.

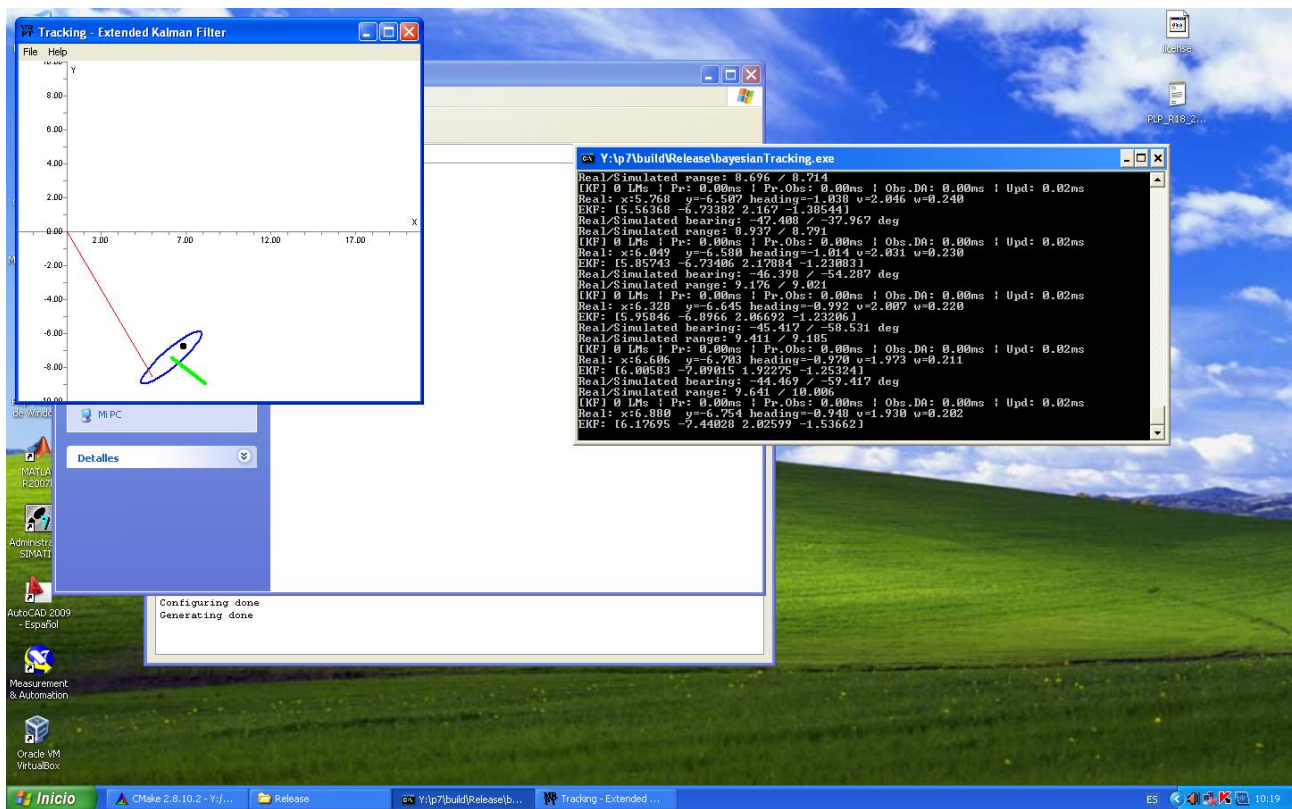
E) Implantación de los métodos de la clase *CRangeBearing*

- El alumno **debe completar** los métodos *OnTransitionModel*, *OnTransitionJacobian*, *OnObservationJacobian*.

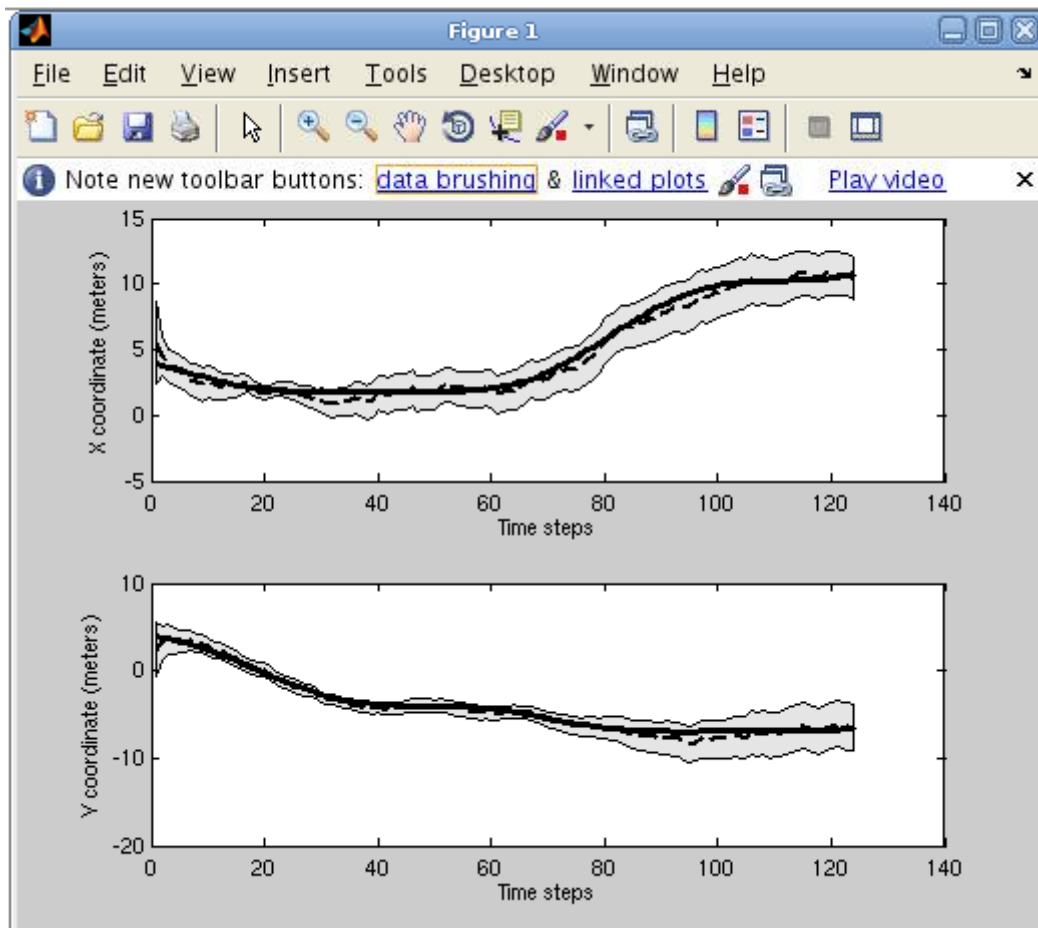
El alumno tiene a su disposición la dirección www.mrpt.org/KalmanFilters con información que le ayudará a resolver la práctica.

2.- Compilar y ejecutar el código

Una vez comprendido el código y completados los apartados del fichero .cpp que sean necesarios compilar y ejecutar el código. Debe obtenerse lo siguiente:



Finalmente, usar el fichero *matlab plot_GT_vs_EKF.m* incluido en el .zip para representar los datos almacenados en el fichero *log_GT_vs_EKF.txt* generado tras una ejecución y analizarlo. La representación obtenida debe tener un aspecto semejante a este:



3.- Responder a las siguientes preguntas

- ¿Cómo se define el estado del robot? ¿Cuál es el estado inicial?
- ¿Cómo se definen las observaciones?
- ¿Qué representan las variables x_{kk} y p_{kk} ?
- ¿Cuáles son las funciones de la clase *CRangeBearing* asociadas a la etapa de predicción?
- ¿En qué posición se sitúa la ventana gráfica donde se muestran los resultados? ¿Cuál es el rango de los ejes visibles?
- ¿Por qué sabemos que estamos usando un EKF y no un KF?
- ¿Qué método EKF, de los que ofrece la MRPT, se ha escogido?
- ¿Qué hace la función *runOneKalmanIteration()*?
- ¿Cómo se obtienen los dos jacobianos? Calcúlalos