

Taller 10: Localización basada en EKF: Predicción

Introducción a la Robótica Móvil

May 9, 2018

1 Introducción

En este taller se **comenzará** a resolver la localización de un robot móvil, dotado de una unidad inercial, *encoders* y un telémetro láser. El método de localización a utilizar estará basado en el Filtro de Kalman Extendido (EKF).

Bajo el esquema de EKF, para la fase de predicción, el robot utilizará la información de velocidad lineal (v) provista por los *encoders* y la de velocidad angular (ω) provista por el giróscopo incluido en la unidad inercial. El modelo de proceso a utilizar es el del vehículo diferencial visto. Para la fase de actualización, el robot utiliza el sensor láser para detectar una serie de postes ubicados en distintos lugares en el ambiente, que serán los *landmarks* del sistema. La detección de dichos postes a partir de un sensado del láser se logra utilizando el módulo de resuelto en el taller anterior.

En este taller nos enfocaremos en la **etapa de predicción del filtro**. Se deberá completar un esqueleto de código, correspondiente al nodo principal de localización. A su vez, este nodo ya resuelve:

- Recepción de los sensados de odometría, unidad inercial y *landmarks* detectados
- Invocación del módulo de EKF para ejecutar el modelo de proceso y el de sensado
- Publicación de la pose estimada (media y covarianza)

A su vez, el módulo EKF encapsula la implementación del filtro en forma genérica, realizando las operaciones necesarias para resolver la fase de predicción y actualización del filtro.

Lo que deberá ser completado por el alumno son los modelos correspondientes que definen al Filtro de Kalman Extendido en el problema particular de la localización como fue descrita.

Para ello, se deberán implementar una serie de métodos que deben resolver el cálculo de los distintos modelos, jacobianos y matrices de covarianza. Tener en cuenta que, debido a que el Filtro de Kalman se puede implementar en forma matricial, todos estos modelos serán siempre matrices o vectores.

1.1 Visualizar la covarianza del estado en RViz

Para visualizar la covarianza del estado actual en **RViz** se utiliza un plugin adicional llamado **rviz_plugin_covariance** el cual se entrega junto con el material para el taller.

Este plugin se compila como cualquier otro paquete **pero para que RViz lo cargue** de manera correcta se debe ejecutar los siguientes comandos **en la raíz del workspace de catkin**:

```
source devel/setup.bash
rviz
```

De esta manera **RViz** detecta la presencia del plugin y lo añade a su interfaz.

Ejercicio 1: Modelo de movimiento (o proceso)

Se desea obtener un primer método de localización puramente predictivo (es decir, sin ejecutar la fase de actualización de EKF).

Deberán trabajar con el archivo fuente **/robmovil_ekf/localizer_ekf.cpp**, donde se encuentran los diferentes métodos que deberán completar con información del modelo. El *framework* de EKF subyacente utilizará entonces el modelo provisto para hacer los cálculos de actualización de estado correspondientes.

NOTA: El primer elemento de los vectores y matrices del *framework* utilizado comienzan en la posición 1

Para correr el experimento deberán utilizar la escena:

/robmovil_ekf/vrep/laser_landmarks.ttt

Utilizar el comando:

```
roslaunch robmovil_ekf robmovil_ekf.launch
```

Y cargar la configuración de **RViz**: **/robmovil_ekf/launch/ekf.rviz**

Se pide:

- Implementar el modelo de movimiento (o proceso) y los jacobianos correspondientes.

Para esto deberán completar (**donde indican los comentarios**) las funciones:

```
void makeBaseA()
void makeA()
void makeBaseW()
void makeProcess()
```

Donde **makeProcess** refiere al modelo de proceso $f(\vec{x}, \vec{u})$, **makeA**/**makeBaseA** al Jacobiano de f respecto de \vec{x} , **makeBaseW** al Jacobiano de f respecto de \vec{w} .

Notar que el ruido del modelo de movimiento (o proceso) se encuentra ya definido en:

```
void makeBaseQ()
```

Donde `makeBaseQ` refiere a la covarianza de \vec{w} .

- Evaluar mediante una simulación en V-Rep el sistema de localización obtenido, comparando la pose estimada con la real (*ground-truth*) en RViz.
- ¿Que sucede si se cambia la fricción del piso? ¿Que tan buena es la estimación obtenida?