

La odometría estima la posición de un vehículo en función del giro de sus ruedas, de tracción y dirección. La reconstrucción de la trayectoria realizada por el vehículo no considera problemas existentes que provocan errores aleatorios (no sistemáticos) debido a derrapes o deslizamientos, huellas de las ruedas en su contacto con la superficie, baches e imperfecciones en el firme, etc.

Dado que estas hipótesis no se cumplen en la realidad, el cálculo de la posición del vehículo mediante este método es bastante inexacta. Además, pueden existir errores sistemáticos, debido a inexactitudes en el diseño mecánico. Los errores cometidos por la odometría son, además, acumulativos, por lo que el error cometido crece indefinidamente.

Por otro lado, los sistemas de localización, aunque corrigen los errores acumulativos de los sensores internos, poseen una deriva que crece con el tiempo. Para conseguir una estimación de la localización fiable a largo plazo, se desea implementar un SLAM (mapeo y localización simultáneos), donde también es necesario mantener un mapa del entorno con las medidas de un sensor láser de barrido. Sea un vehículo con direccionamiento de *Ackermann* cuyo modelo de conducción viene dado por:

$$u = \begin{bmatrix} V \\ G \end{bmatrix}$$

donde V es la velocidad del vehículo y G el ángulo de dirección. Por tanto, el modelo de movimiento del vehículo (odometría) es:

$$x_v(k) = x_v(k-1) + \begin{bmatrix} V dt \cos(G + \Delta\theta) \\ V dt \sin(G + \Delta\theta) \\ V dt \sin(G)/B \end{bmatrix}$$

donde dt el periodo de tiempo transcurrido entre el instante k y $k-1$, $\Delta\theta$ el incremento de orientación el vehículo y B la distancia entre ruedas.

Se suministra un simulador que implementa un SLAM basado en el Filtro Extendido de Kalman (EKF), que construye un mapa empleando un sensor láser de barrido para detectar puntos clave en el entorno para utilizarlos como referencia. En dicho simulador, entre distintas funciones, las más relevantes para completar el ejercicio son:

- **configfile**. Fichero de configuración de parámetros del problema: parámetros de control, modelos de ruido de control y observación, etc.
- **ekfslam_sim**. Programa principal: simulador de SLAM basado en EKF.
- **predict**. Función que realiza el paso de predicción del EKF basado en el modelo del vehículo.
- **observe_model**. Función que proporciona la observación y la incertidumbre asociada.
- **update_EKF**. Función que realiza el paso de actualización (corrección) del EKF.
- **Archivos *.mat**. Archivos de datos de *Matlab* que contienen datos sobre trayectorias y localización de marcas en el entorno.

Para utilizar el simulador, en primer lugar, es necesario cargar las estructuras de datos que van a contener la trayectoria real que debe realizar el vehículo y la ubicación de las marcas del entorno. Por tanto, en la ventana de comandos de *Matlab* primero se cargan los datos:

```
>> load('nombre_archivo.mat')
```

carga el contenido del fichero de datos en el *workspace*. Se cargan *lm* (lista de marcas del entorno) y *wp* (lista de puntos del camino). Una vez que se tienen los datos disponibles en el espacio de trabajo, se ejecuta el simulador:

```
>> data=ekfloc_sim(lm,wp)
```



Ejecuta el simulador, mostrando en una ventana una animación del vehículo en movimiento a lo largo de la trayectoria, representando la trayectoria estimada y la observación del entorno realizada en cada posición.

Como resultado de la simulación se obtiene una estructura de datos (denominada generalmente *data*) que contiene:

- *data.path*: estimación del camino realizado por el vehículo.
- *data.odom*: odometría del vehículo (predicción).
- *data.true*: trayectoria real del vehículo.
- *data.state(k).x*: vector de estado de la localización del vehículo en el instante *k*.
- *data.state(k).P*: diagonal de la matriz de covarianza de la localización (incertidumbre) en el instante *k*.

Se pide:

1. Completar la fase de predicción (función ***predict***) con las matrices jacobianas correspondientes al modelo de conducción anterior.
2. Calcular el error cuadrático medio en la distancia y ángulo de orientación a lo largo de la trayectoria, según la ecuación:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (true_i - path_i)^2}{n}}$$

3. Realizar simulaciones para distintos valores de incertidumbre en el control del vehículo y de la observación (al menos un cambio en cada uno de ellos). Modificar dichas incertidumbres en el fichero de parámetros (***configfile***). Estudiar su influencia en el resultado calculando el error cuadrático medio en cada caso y comentar las conclusiones.