TERCERA PRÁCTICA

SLAM BASADO EN EL FILTRO EXTENDIDO DE KALMAN

MIGUEL IAN GARCÍA POZO

GIERM

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE MÁLAGA

Índice

Fase de Predicción

- Desarrollo

Para obtener las matrices jacobianas correspondientes al Modelo Error de Predicción, debemos calcular varias derivadas parciales.

Primero, obtendremos la matriz jacobiana del modelo de movimiento del vehículo, es decir, la odometría. Para ello, debemos hacer las derivadas parciales con respecto a "x", a "y" y respecto a "θ".

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f_X}{\partial x} & \frac{\partial f_X}{\partial y} & \frac{\partial f_X}{\partial \theta} \\ \frac{\partial f_Y}{\partial x} & \frac{\partial f_Y}{\partial y} & \frac{\partial f_Y}{\partial \theta} \\ \frac{\partial f_\theta}{\partial x} & \frac{\partial f_\theta}{\partial y} & \frac{\partial f_\theta}{\partial \theta} \end{bmatrix}$$

Nuestra función del movimiento para "x", "y" y "θ"es la siguiente, ya que x_v representa las coordenadas y orientación (x,y,θ) del vehículo.

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

Teniendo esto claro ya podemos empezar a derivar:

$$\frac{\partial f_x}{\partial x} = 1 + 0 = 1 \qquad \frac{\partial f_x}{\partial y} = 0 + 0 = 0$$

$$\frac{\partial f_x}{\partial \theta} = 0 - V \cdot dt \cdot sen(G + \Delta\theta) = -V \cdot dt \cdot sen(G + \Delta\theta)$$

$$\frac{\partial f_y}{\partial x} = 0 + 0 = 0 \qquad \frac{\partial f_y}{\partial y} = 1 + 0 = 1$$

$$\frac{\partial f_x}{\partial \theta} = 0 + V \cdot dt \cdot cos(G + \Delta\theta) = V \cdot dt \cdot cos(G + \Delta\theta)$$

$$\frac{\partial f_\theta}{\partial x} = 0 + 0 = 0 \qquad \frac{\partial f_\theta}{\partial y} = 0 + 0 = 0 \qquad \frac{\partial f_\theta}{\partial \theta} = 1 + 0 = 1$$

Habiendo calculado todas las parciales, ya podemos montar la matriz jacobiana de la odometría.

$$G_v = \begin{bmatrix} 1 & 0 & -V \cdot dt \cdot sen\,(G + \Delta\theta) \\ 0 & 1 & V \cdot dt \cdot cos\,(G + \Delta\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Ahora, debemos calcular la matriz jacobiana del modelo de conducción del direccionamiento de Ackerman. Para ello, volveremos a derivar la función anterior, pero con respecto a las variables “V” y “G”.

$$\frac{\partial f_x}{\partial V} = 0 + dt \cdot cos(G + \Delta\theta) = dt \cdot cos(G + \Delta\theta)$$
$$\frac{\partial f_x}{\partial G} = 0 - V \cdot dt \cdot sen(G + \Delta\theta) =- V \cdot dt \cdot sen(G + \Delta\theta)$$

$$\frac{\partial f_y}{\partial V} = 0 + dt \cdot sen(G + \Delta\theta) = dt \cdot sen(G + \Delta\theta)$$
$$\frac{\partial f_y}{\partial G} = 0 + V \cdot dt \cdot cos(G + \Delta\theta) = V \cdot dt \cdot cos(G + \Delta\theta)$$

$$\frac{\partial f_\theta}{\partial V} = 0 + dt \cdot sen(G)/B = dt \cdot sen(G)/B \qquad \frac{\partial f_\theta}{\partial G} = 0 + V \cdot dt \cdot cos(G)/B = V \cdot dt \cdot cos(G)/B$$

Con esto, podemos ahora construir nuestra segunda matriz jacobiana.

$$G_u = \begin{bmatrix} dt \cdot \cos(G + \Delta\theta) & -V \cdot dt \cdot sen\,(G + \Delta\theta) \\ dt \cdot sen\,(G + \Delta\theta) & V \cdot dt \cdot \cos(G + \Delta\theta) \\ dt \cdot sen\,(G)/B & V \cdot dt \cdot \cos(G)/B \end{bmatrix}$$

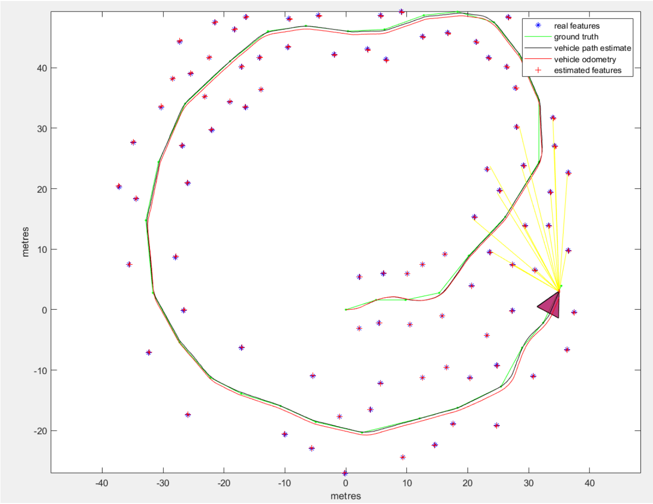
Teniendo ya las dos matrices del Modelo Error de Predicción, podemos introducirlas en el archivo “predict.m” y realizar la simulación.

```
s= sin(g+XX(3)); c= cos(g+XX(3));
vts= v*dt*s; vtc= v*dt*c;

Gv= [1 0 -vts;0 1 vtc;0 0 1];
Gu= [dt*c -vts;dt*s vtc;dt*sin(g)/WB vtc/WB];
```

- Simulación

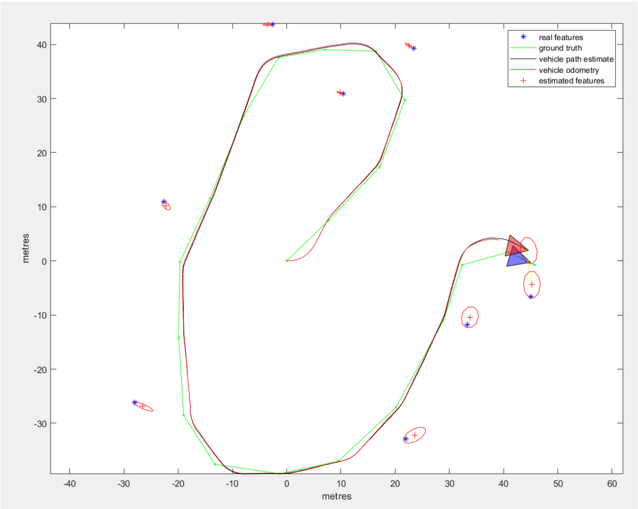
Para dar comienzo a la simulación, debemos cargar primero el mapa ejemplo, y luego ejecutamos el código de la localización SLAM basado en el Filtro Extendido de Kalman.

```
load('ejemplo.mat')
data = ekfslam_sim(lm,wp)
```



Podemos probar con otro mapa ejemplo a ver cómo se resuelve.

```
load('ejemplo3.mat')
data = ekfslam sim(lm,wp)
```



Vemos que, en este segundo caso, la incertidumbre es mucho mayor debido a la falta de marcas con las que el vehículo pueda corregir esos errores que se producen cuánto mayor es el camino recorrido.

Error Cuadrático Medio

- Desarrollo

En este apartado, debemos calcular cuál ha sido el error cuadrático medio, tanto en la distancia como en el ángulo, a lo largo de la trayectoria. Para ello, debemos hacer uso de la fórmula dada en el enunciado para la distancia y para el ángulo.

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

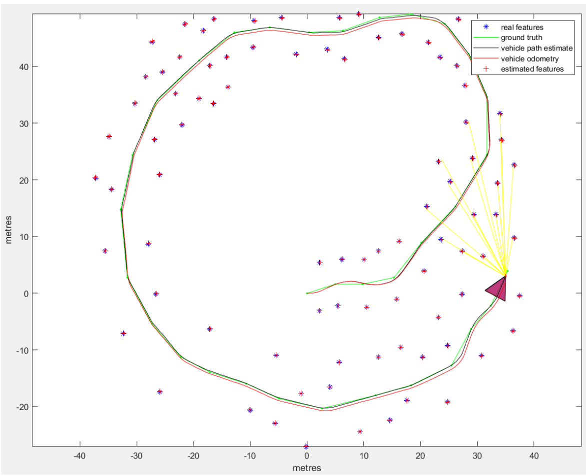
Para obtener los valores “path” y “true”, debemos usar los comandos que nos proporciona el enunciado: "$data.path$" y "$data.true$". El valor “n” es el número de elementos en las matrices de la ruta y el groundtruth. Implementando todo en Matlab, tenemos el siguiente código:

```
function ErrorCuadMedio(ejemplo)

    load(ejemplo) % Carga el mapa con los landmarks

    data = ekfslam_sim(lm,wp); % Realiza la simulación del vehículo

    sumatoriad = 0;
    sumatoriao = 0;
    % Bucle for para hacer el sumatorio
    for i=1:data.i
        % El sumatorio en distancia debemos hacerlo calculando las
        % distancias utilizando pitágoras tanto en la matriz true como en
        % la matriz path
        sumatoriad = sumatoriad + (sqrt(data.true(1,i)^2+data.true(2,i)^2)-sqrt(data.path(1,i)^2+data.path(2,i)^2))^2;
        % El sumatorio en ángulo solo tiene una componente de cada matriz
        sumatoriao = sumatoriao + (data.true(3,i)-data.path(3,i))^2;
    end

    % Dividimos ambos sumatorios entre el número de elementos y le hacemos
    % la raíz cuadrada
    RMSEd = sqrt(sumatoriad/data.i)
    RMSEo = sqrt(sumatoriao/data.i)
end
```

- Simulación

Ahora solo debemos ejecutar la función dándole un mapa ejemplo para simular el comportamiento del vehículo y obtener los Errores Cuadrático Medios en distancia y en ángulo.

```
ErrorCuadMedio('ejemplo.mat')
```

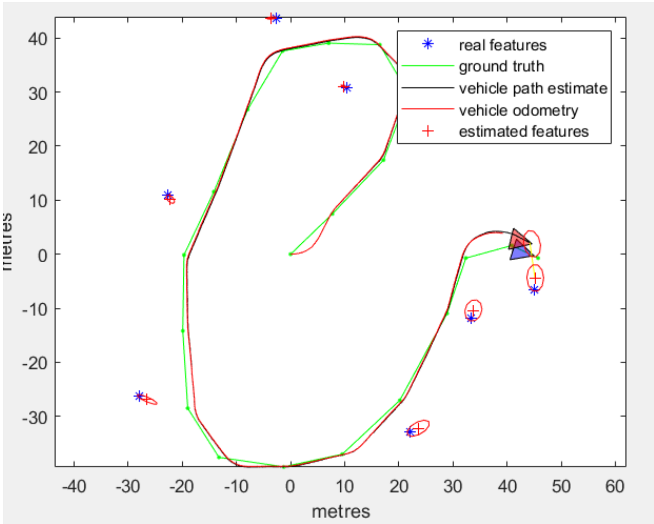


Esta simulación nos da como resultado los siguientes Errores Cuadrático Medios:

```
RMSEd =
    0.0693
RMSEo =
    0.3302
```

Para poder hacer una comparación, vamos a probar usando otro mapa ejemplo.

```
ErrorCuadMedio('ejemplo3.mat')
```



El resultado del ejemplo 3 es:

```
RMSEd =
    0.3017
RMSEo =
    0.3006
```

Podemos ver que el Error Cuadrático Medio en distancia en este ejemplo es cinco veces mayor que el anterior, debido a la falta de “landmarks” que ayuden al vehículo a reducir la incertidumbre.

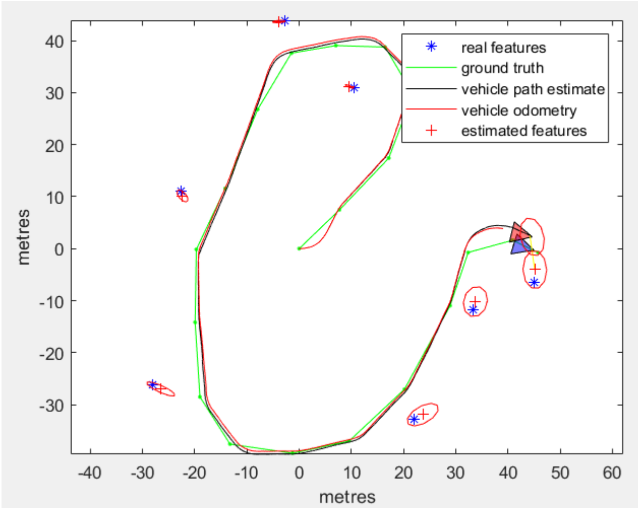
Distintos valores de Incertidumbre

- Pruebas

Vamos a probar subiendo el ruido en la velocidad de 0. 3 a 0. 8 m/s. Veamos qué ocurre:

```
% control noises
sigmaV= 0.8; % m/s
sigmaG= (3.0*pi/180); % radians
Q= [sigmaV^2 0; 0 sigmaG^2];
```

```
ErrorCuadMedio('ejemplo3.mat')
```



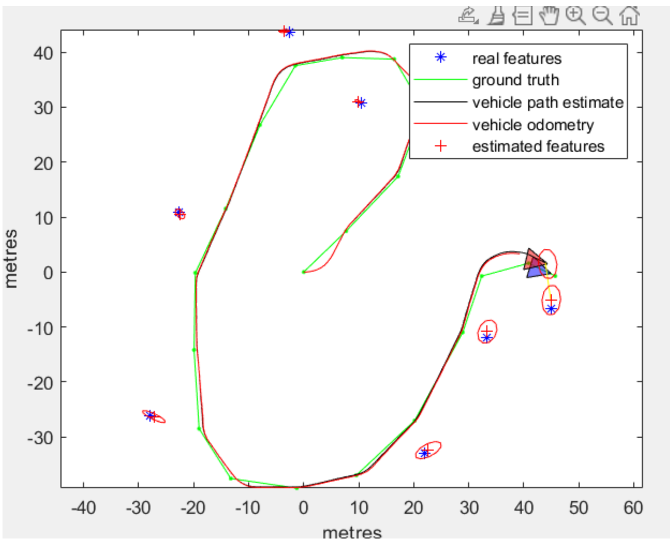

```
RMSEd =
    0.2587
RMSEo =
    0.3400
```

Viendo los resultados nos damos cuenta que el error en la distancia ha disminuido mientras que en la orientación ha aumentado.

Probemos ahora modificando el ruido en la observación de 0.1 a $0.5\ m$, devolviendo el valor de la velocidad a su valor original.

```
% observation noises
sigmaR= 0.5; % metres
sigmaB= (1.0*pi/180); % radian
R= [sigmaR^2 0; 0 sigmaB^2];
```

```
ErrorCuadMedio('ejemplo3.mat')
```

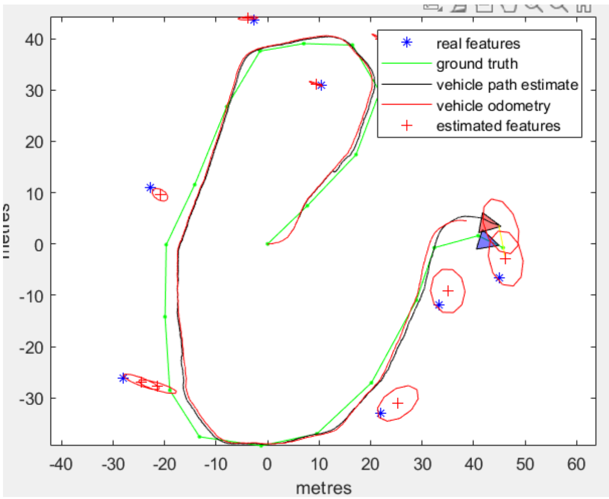



```
RMSEd =
    0.2726
RMSEo =
    0.2779
```

Vemos que en este caso ambos errores se han visto reducidos.

Si ahora probamos modificando el error en la orientación, obtenemos:

```
% control noises
sigmaV= 0.3; % m/s
sigmaG= (10.0*pi/180); % radians
Q= [sigmaV^2 0; 0 sigmaG^2];
```

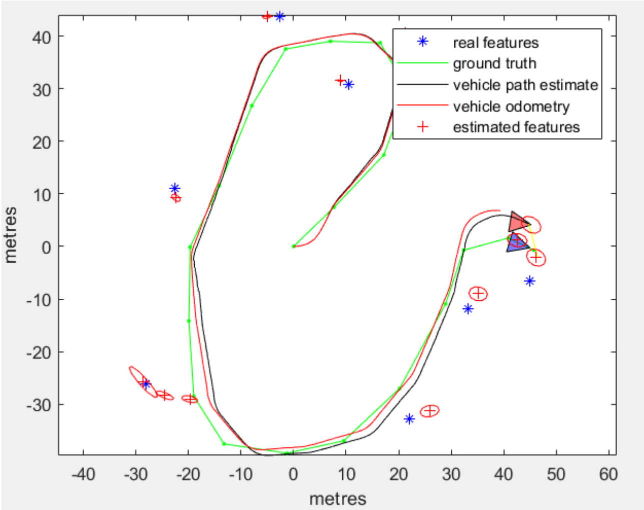



```
RMSEd =
      0.9613
RMSEo =
      0.4104
```

En este caso, es muy destacable el gran aumento del error en distancia.

El último caso que nos queda por probar es el error en la observación del ángulo. Veamos qué ocurre.

```
% observation noises
sigmaR= 0.1; % metres
sigmaB= (10.0*pi/180); % radians
R= [sigmaR^2 0; 0 sigmaB^2];
```




```
RMSEd =
      0.6044
RMSEo =
      0.3279
```

En este caso vemos que el error en distancia es el doble que en el caso original, aunque el error en ángulo se mantiene parecido. También podemos ver en el mapa la mala detección de los “landmarks”.

- Conclusión

La conclusión que podemos sacar de estas pruebas, es que el cambio de la incertidumbre en metros no afecta en gran medida al error en la localización, mientras que la incertidumbre en la orientación o el ángulo aumenta mucho el Error Cuadrático Medio en distancia. Por otro lado, hemos observado que en ninguno de los casos el error en el ángulo ha variado mucho, esto nos quiere decir que este error no se ve muy influido por los ruidos en el movimiento y la observación.