1. Simulador del Robot

Robot	
MapaReal	BinaryOccupancyMap
Mapa	BinaryOccupancyMap
PosReal	1x3 Vector
Posicion	1x3 Vector
InitPos	1x3 Vector
figMapaReal	Figure
figMapa	Figure
figSensor	Figure
Sensor	rangeSensor
TimeStep	Double
TotalTime	Double
GraphicsOn	Boolean
ruidoOn	Boolean
ruiLidar	Double
ruiOdo	Double
RobotGen	
escanearAlrededores	
mostrarScan	
unirEscaneo	
unirEscaneoOld	
plotCircleGrid	
moverRobot	
moverRobotA	
moverRobotA2	
updatePlotRobot	
checkCollided	
ExploredData	
DistanceMoved	

El código que se presenta se basa en OOP (Object Oriented Programing). El objeto principal es Robot, que esta compuesto por:

- ***MapaReal*** (binary Occupancy Map): Almacena el entorno real en el que se encuentra el robot.
- ***Mapa*** (binary Occupancy Map): Almacena el mapa que va construyendo el robot a través de sus sensores.
- ***PosReal*** (vector 1x3): Almacena las coordenadas reales del robot respecto al world. Las unidades son metros y radianes.
- ***Posicion*** (vector 1x3): Almacena las coordenadas del robot relativas a la ***InitPos***. Si se ha habilitado ruido no se garantiza que la ***Posicion*** sea igual a las de ***PosReal***. Las unidades son metros y radianes.
- ***InitPos*** (vector 1x3): Almacena las coordenadas relativas al mundo iniciales del robot. Las unidades son metros y radianes.
- ***figMapaReal*** (figure): Puntero a la imagen del ***MapaReal*** para actualizar la misma. Muestra el ***MapaReal*** y la ***PosReal*** del robot.
- ***figMapa*** (figure): Puntero a la imagen del ***Mapa*** para actualizar la misma. Muestra el ***Mapa*** y la ***Posicion*** del robot.
- ***figSensor*** (figure): Puntero a la imagen del escáner para actualizar la misma. Muestra la información del escáner.
- ***Sensor*** (rangeSensor): Objeto para la simulación del sensor lidar.
- ***TimeStep*** (Double): Frecuencia con la que se realizaría el bucle de control del robot.
- ***TotalTime*** (Double): Contador del tiempo que ha estado operando el robot.
- ***GraphicsOn*** (Boolean): Variable que habilita la aparición de figuras durante la ejecución de los algoritmos.
- ***ruidoOn*** (Boolean): Variable que habilita el ruido en la ejecución de los movimientos.
- ***ruiLidar*** (Double): Variable que regula el máximo error relativo de las medidas del sensor. Sus unidades van en porcentaje (ej: 50 = 50% de error relativo máximo).

- ***ruiOdo*** (Double): Variable que regula el máximo error relativo de las medidas del sensor. Sus unidades van en porcentaje (ej: 50 = 50% de error relativo máximo).

Para utilizar la clase existen los siguientes metodos:

- ***RobotGen***: In (***map,pos,rangosensor,tp,Res,gOn,rOn,rLidar,rOdo***) Out [***Robot***]

  Constructor del robot.

  - ***map***: matriz del mapa de ocupación binario. Su resolución de este mapa ha de ser en metros.
  - ***pos***: vector con la posición inicial del robot. Las unidades son metros y radianes. Ej: [3, 3, pi/2].
  - ***rangosensor***: vector de 1x2 con la distancia mínima y máxima que tiene el sensor para escanear. Las unidades son metros. Ej: [0, 10].
  - ***tp***: double que marca el ***TimeStep*** del robot. La unidad es el segundo. Ej: (0.07)
  - ***Res***: double que marca la resolución del ***Mapa*** a generar por el robot. La unidad es m^-1. Ej: (10).
  - ***gOn***: booleano que marca ***GraphicsOn***.
  - ***rOn***: booleano que marca ***RuidoOn***. Si está en false no se requerirán las siguientes variables.
    - ***rLidar***: double que marca ***ruiLidar***.
    - ***rOdo***: double que marca ***ruiOdo***.
- ***escanearAlrededores***: In(***Robot***) Out [***distancia, angulo***]

  Función que devuelve la información obtenida por el lidar.
- ***mostrarScan***: In (***Robot, distancia, angulo, force***). Out(***Robot***)

  Función que muestra en la figura ***figSesor*** la información obtenida por el sensor Lidar. ***distancia*** y ***angulo*** son las variables generadas por ***escanearAlrededores***. ***force*** es un booleano que fuerza los graficos en caso de que ***GraphicsOn*** sea false.
- ***unirEscaneo:*** In (***Robot, distancia, angulo***). Out(***Robot***)

Función que actualiza el ***Mapa*** del robot con la información del Lidar. ***distancia*** y ***angulo*** son las variables generadas por ***escanearAlrededores***.

- ***unirEscaneoOld:*** In (***Robot, distancia, angulo***). Out(***Robot***)

  Función que actualiza el ***Mapa*** del robot con la información del Lidar. ***distancia*** y ***angulo*** son las variables generadas por ***escanearAlrededores***. Versión antigua donde el mapa constantemente cambiaba.

- ***plotCircleGrid:*** In (***inner,outer,sidespoly,indivisions***)

  Función de apoyo a ***mostrarScan*** donde se genera unos ejes esféricos.

  - ***inner:*** marca el radio interior de la rejilla.
  - ***outer:*** marca el radio exterior de la rejilla.
  - ***sidespoly:*** marca el número de lados con los que se aproxima el círculo.
  - ***indivisions:*** marca cuantos circulos se muestran en la rejilla.

- ***moverRobot:*** In (***Robot, velL, velA***). Out(***Robot***)

  Función que realiza el movimiento del robot con la velocidad lineal (***velL***) y la angular (***velA***) que se le pasa.

- ***moverRobotA:*** In (***Robot, Path***). Out(***Robot, Path***)

  Función que implementa un controlador ***purePursuit*** que navega el robot siguiendo el ***Path*** marcado***.***

  - ***Path:*** vector de nx2 compuesto por n puntos del mapa. Ej: [0,0; 0,1; 1,1]

- ***moverRobotA2:*** In (***Robot, Path***). Out(***Robot***)

  Función que implementa un controlador más robusto que en ***moverRobotA*** que navega el robot siguiendo el ***Path*** marcado***.***

  - ***Path:*** vector de nx2 compuesto por n puntos del mapa. Ej: [0,0; 0,1; 1,1]

- ***updatePlotRobot:*** In (***Robot)*** Out (***Robot)***

  Función que muestra el la ***Posicion*** del robot en el ***Mapa*** y la ***PosReal*** en el ***MapaReal.***

- ***checkCollided:*** In (***Robot***) Out (***Collided***)

  Función que devuelve si el robot se encuentra en una posición inválida.

- ***ExploredData:*** In (***Robot)*** Out (***mark)***

Función que devuelve el porcentaje de paredes detectado en el mapa. Esta función sirve para evaluar los algoritmos.

- ***DistanceMoved:*** In (***Robot)*** Out (***dist)***

  Función que devuelve la distancia recorrida por el robot hasta el momento. Esta función sirve para evaluar los algoritmos.

La simulación del robot implica una carga computacional muy avanzada, debido a las constantes actualizaciones gráficas, por lo que solo se recomienda habilitar GraphicsOn si se quiere ver o comprobar el correcto funcionamiento de los algoritmos.