

# **Sistemas de Control para Robots**

## **PRÁCTICA 1**



Universidad  
de Alcalá

## **Mapeado**

**Pedro Barquín Ayuso**  
**Miguel Ballesteros García**

# 1 MAP

- **Obtener la información del proxy map. Identificando todos los atributos que ofrece el proxy.**

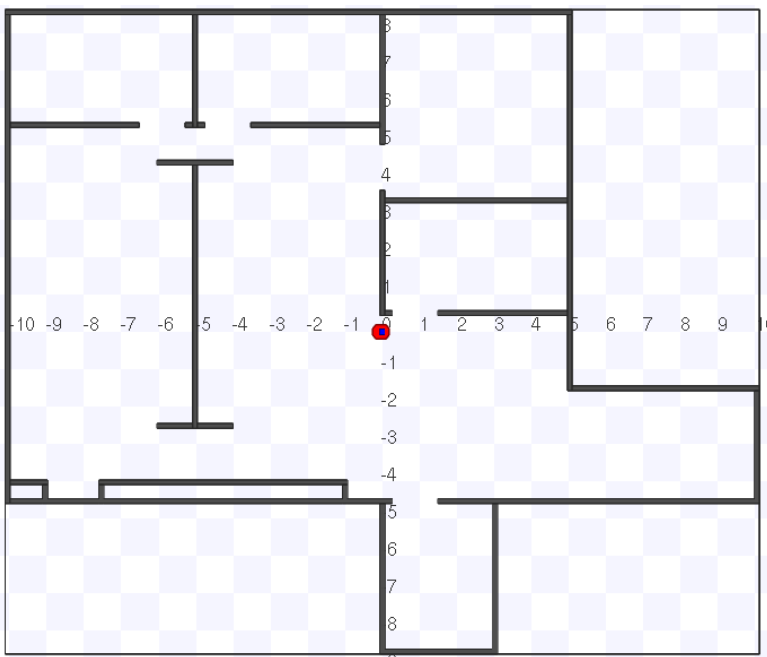
- Info: información del mapa
- Resolution: resolución del mapa en m/celdas.
- Width: anchura del mapa en celdas
- Height: altura del mapa en celdas.
- Origin: coordenada de origen del mapa en metros.
- Cells: ocupación de cada celda (vacía=-1, desconocido=0, ocupada=1)

Nota:

Los demás atributos son vectoriales y no funcionan en nuestro simulador. Por lo que al tratar de obtener su información aparecen siempre como 0 o 0.0

**-¿Qué tipo de representación del entorno utiliza Player/Stage a través de este proxy?**

El driver map utiliza una representación del entorno mediante celdas, partiendo de la imagen base traza un matriz de celdas las cuales corresponde a una parte de cierto tamaño del mapa, en esas celdas se representa el estado de esa parte del mapa mediante -1 0 o 1 lo que quiere decir que el punto este vacío desconocido u ocupado.



```
alumno@ubuntu910: ~/Escritorio/PractiaEjemplo
Archivo Editar Ver Terminal Ayuda
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th -0.087128 stall 0
position2d : x 0.000000 y 0.000000 th -0.188106 stall 0
position2d : x 0.000000 y 0.000000 th -0.262003 stall 0
^C
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$ ./ejemplo1
playerc warning : warning : [Player v.3.0.1] connected on [localhost:66
h sock 3

Informacion del mapa
resolucion : 0.025000
anchura : 809
altura : 689
origen en x : -10.112500
origen en y : -8.612500
vmix : 0.000000
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th -0.086502 stall 0
position2d : x 0.000000 y 0.000000 th -0.186960 stall 0
position2d : x 0.038306 y -0.011788 th -0.298530 stall 0
position2d : x 0.152076 y -0.050657 th -0.343128 stall 0
^C
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$
```

```

driver
(
  name "stage"
  provides ["simulation:0"]
  plugin "stageplugin"
  worldfile "ejemplo1.world"
)

driver
(
  name "stage"
  provides ["graphics2d:0"]
  model "mimapa"
)

driver
(
  name "mapfile"
  provides ["map:0"]
  filename "autolab.png"
  resolution 0.025 #preguntar
)

driver
(
  name "stage"
  provides ["odometry::position2d:0"]
  model "robot1"
)

include "pioneer.inc"
include "map.inc"
include "sick.inc"

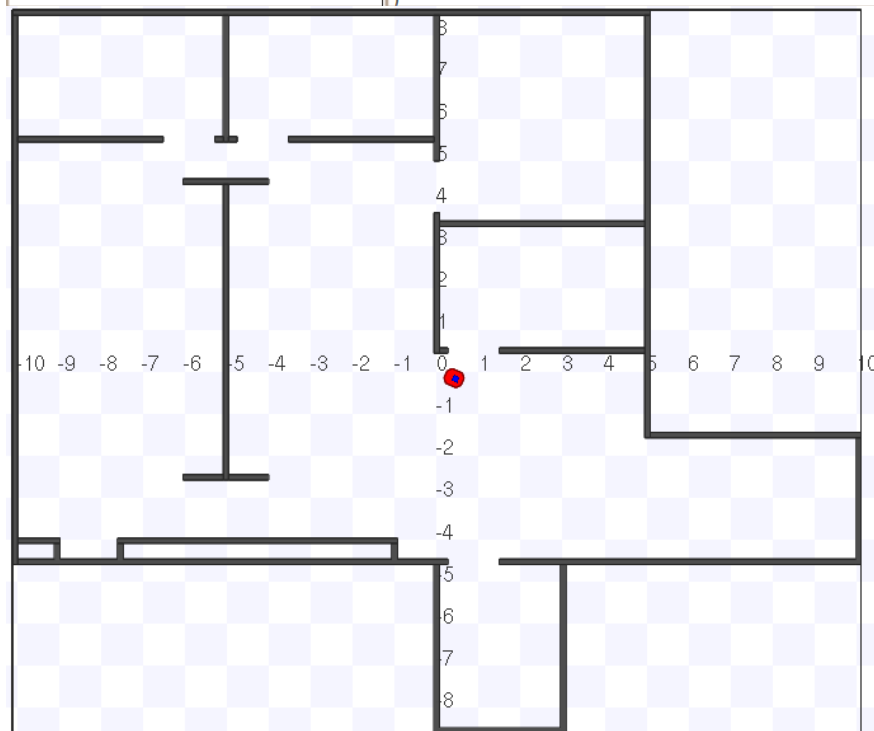
window
(
  size [ 809 689 ] # Tamaño de la ventana de visualización de Stage en píxeles
  center [0.0 0.0] # Coordenadas en metros (mundo) del centro de la ventana
  scale 40 # Relación entre coordenadas en y en metros.

  show_grid 1
  show_data 1
)

floorplan
(
  name "mimapa"
  bitmap "autolab.png"
  size [20.225 17.225 1] # Tamaño en metros del Lo carga centrado en la ventana de Stage
)

pioneer2dx
(
  name "robot1"
  color "red"
  localization "odom" #tipo obtencion posicion
  pose [0.0 0.0 0.0 0.0] #posicion del robot en el mapa
  sicklaser() #añade el laser
)

```



```

alumno@ubuntu910: ~/Escritorio/PractiaEjemplo
Archivo Editar Ver Terminal Ayuda
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th -0.087128 stall 0
position2d : x 0.000000 y 0.000000 th -0.188106 stall 0
position2d : x 0.000000 y 0.000000 th -0.262003 stall 0
^C
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$ ./ejemplo1
playerc warning : warning : [Player v.3.0.1] connected on
h sock 3

Informacion del mapa
resolucion : 0.025000
anchura : 809
altura : 689
origen en x : -10.112500
origen en y : -8.612500
vminx : 0.000000
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th -0.086502 stall 0
position2d : x 0.000000 y 0.000000 th -0.186960 stall 0
position2d : x 0.038306 y -0.011788 th -0.298530 stall 0
position2d : x 0.152076 y -0.050657 th -0.343128 stall 0
^C
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$

```

```

alumno@ubuntu910: ~/Escritorio/PractiaEjemplo
Archivo Editar Ver Terminal Ayuda
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$ ./ejemplo1
playerc warning : warning : [Player v.3.0.1] connected on
65] with sock 3

Informacion del mapa
resolucion : 0.035000
anchura : 809
altura : 689
origen en x : -14.157500
origen en y : -12.057500
vminx : 0.000000
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th -0.046214 stall 0
position2d : x 0.000000 y 0.000000 th -0.159008 stall 0
position2d : x 0.000000 y 0.000000 th -0.216711 stall 0
position2d : x 0.000000 y 0.000000 th -0.283436 stall 0
position2d : x 0.150294 y -0.050532 th -0.345847 stall 0
position2d : x 0.187426 y -0.064423 th -0.357993 stall 0
position2d : x 0.297933 y -0.108394 th -0.388060 stall 0
^C
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$

```

```

driver
(
  name "stage"
  provides ["simulation:0"]
  plugin "stageplugin"
  worldfile "ejemplo1.world"
)

driver
(
  name "stage"
  provides ["graphics2d:0"]
  model "mimapa"
)

driver
(
  name "mapfile"
  provides ["map:0"]
  filename "autolab.png"
  resolution 0.035 #preguntar
)

driver
(
  name "stage"
  provides ["odometry::position2d:0"]
  model "robot1"
)

```

```

include "pioneer.inc"
include "map.inc"
include "sick.inc"

window
(
  size [ 900 900 ] # Tamaño de la ventana de visualiza
  Stage en pixeles
  center [0.0 0.0] # Coordenadas en metros (mundo real
  centro de la ventana
  scale 40 # Relación entre coordenadas en píx
  en metros.

  show_grid 1
  show_data 1
)

floorplan
(
  name "mimapa"
  bitmap "autolab.png"
  size [20.225 17.225 1] # Tamaño en metros del bit
  carga centrado en la ventana de Stage
)

pioneer2dx
(
  name "robot1"
  color "red"
  localization "odom" #tipo obtencion posicion
  pose [0.0 0.0 0.0 0.0] #posicion del robot en el mapa
  sicklaser() #añade el laser
)

```

alumno@ubuntu910: ~/Escritorio/PractiaEjemplo

Archivo Editar Ver Terminal Ayuda

```

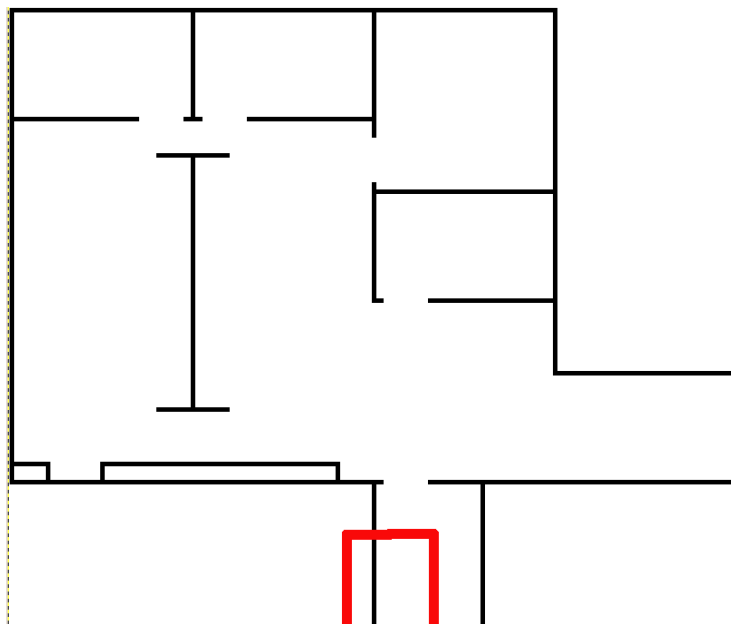
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$ ./ejemplo1
playerc warning : warning : [Player v.3.0.1] connected
65] with sock 3

Informacion del mapa
resolucion : 0.035000
anchura : 809
altura : 689
origen en x : -14.157500
origen en y : -12.057500
vminx : 0.000000
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th 0.000000 stall 0
position2d : x 0.000000 y 0.000000 th -0.046214 stall 0
position2d : x 0.000000 y 0.000000 th -0.159008 stall 0
position2d : x 0.000000 y 0.000000 th -0.216711 stall 0
position2d : x 0.000000 y 0.000000 th -0.283436 stall 0
position2d : x 0.150294 y -0.050532 th -0.345847 stall 0
position2d : x 0.187426 y -0.064423 th -0.357993 stall 0
position2d : x 0.297933 y -0.108394 th -0.388060 stall 0
^C
alumno@ubuntu910:~/Escritorio/PractiaEjemplo$

```

Como vemos en los pares de capturas, al cambiar los parámetros del mapa en el .cfg y en él .world los parámetro que nos devuelven también cambian como se esperaba. En el .cfg configuramos la resolución y en él .world el tamaño del mapa.

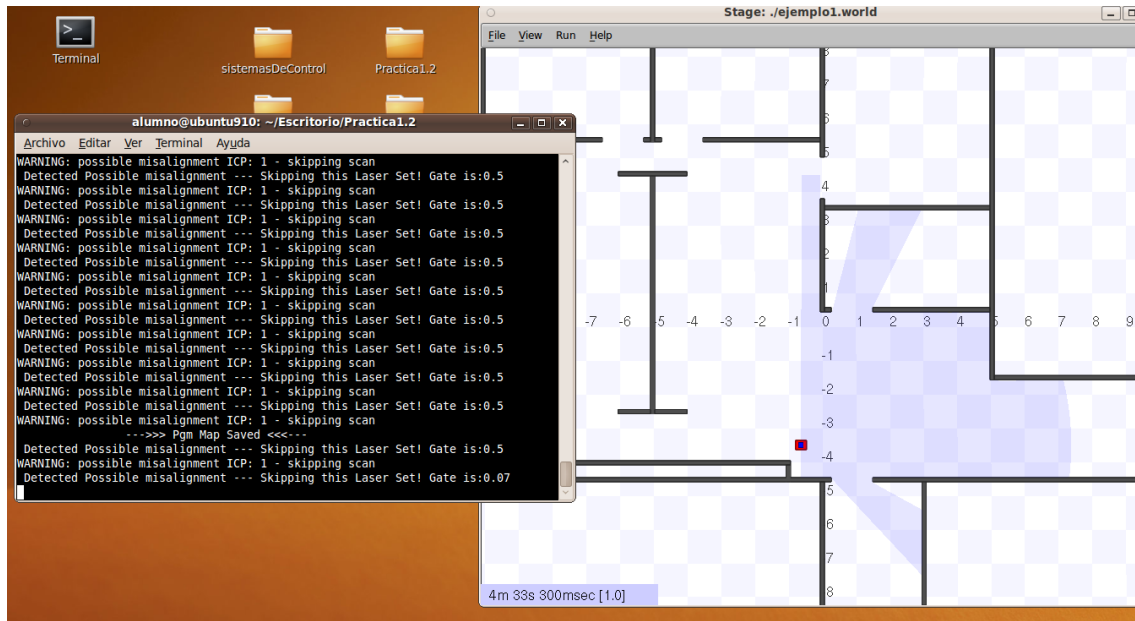
Por otro lado para comprobar el funcionamiento del parámetro cell y ver si efectivamente nos dice la ocupación de cada celda seleccionamos una sección del mapa como por ejemplo la siguiente remarcada en rojo.



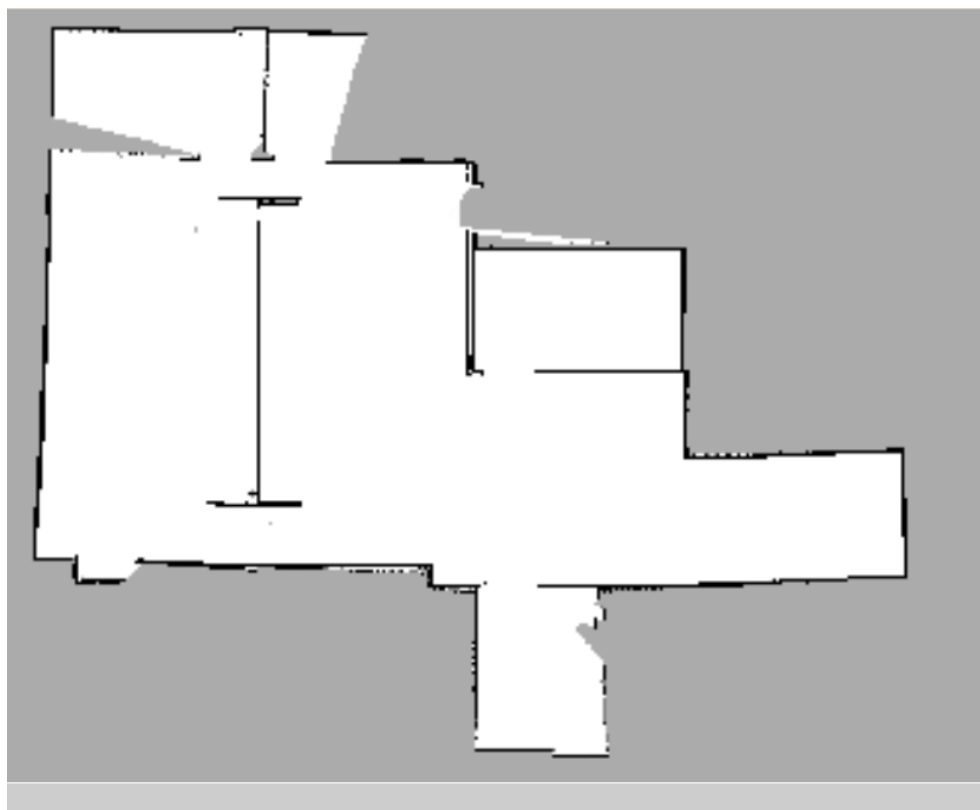
Después llamamos a la función poniendo como parámetros el recuadro marcado el cual nos devuelve del valor de ocupación de cada celda. Para visualizar dicha sección hemos utilizado un bucle en el cual vamos imprimiendo los valores de la celda y mostrándolos por pantalla como podemos ver en la siguiente imagen.

## 2 MRICP (on-line)

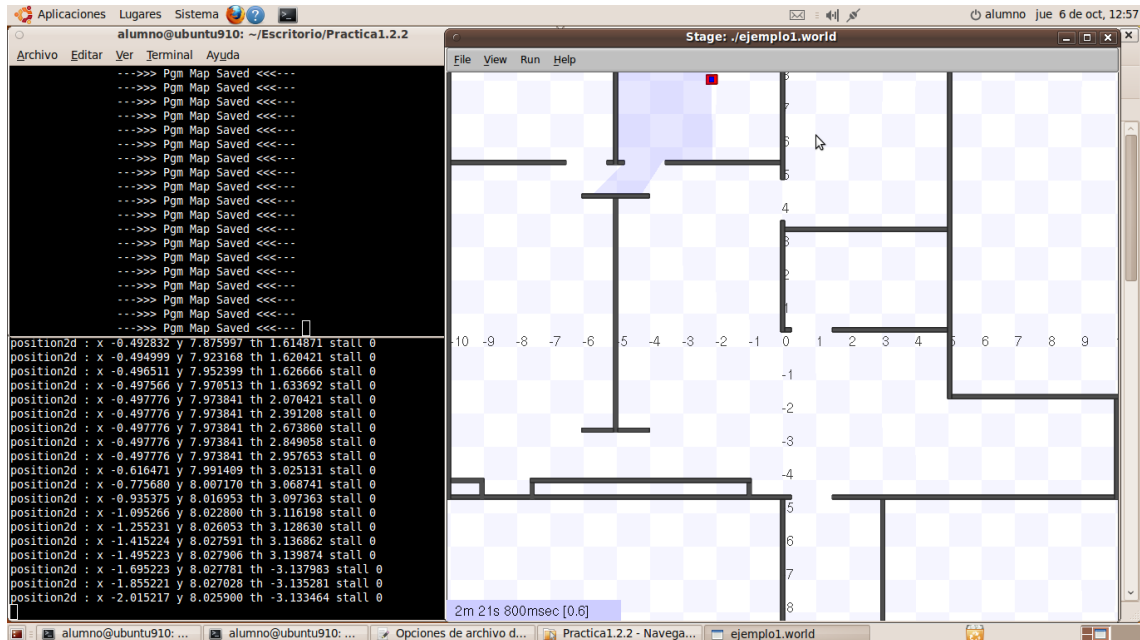
Al probar esta parte de la práctica nos equivocamos al poner el valor de MINR=0.5 y no daba muchísimo error debido a que en el recorrido que realizaba pasaba muy cerca de las paredes y no se cumplía la distancia mínima



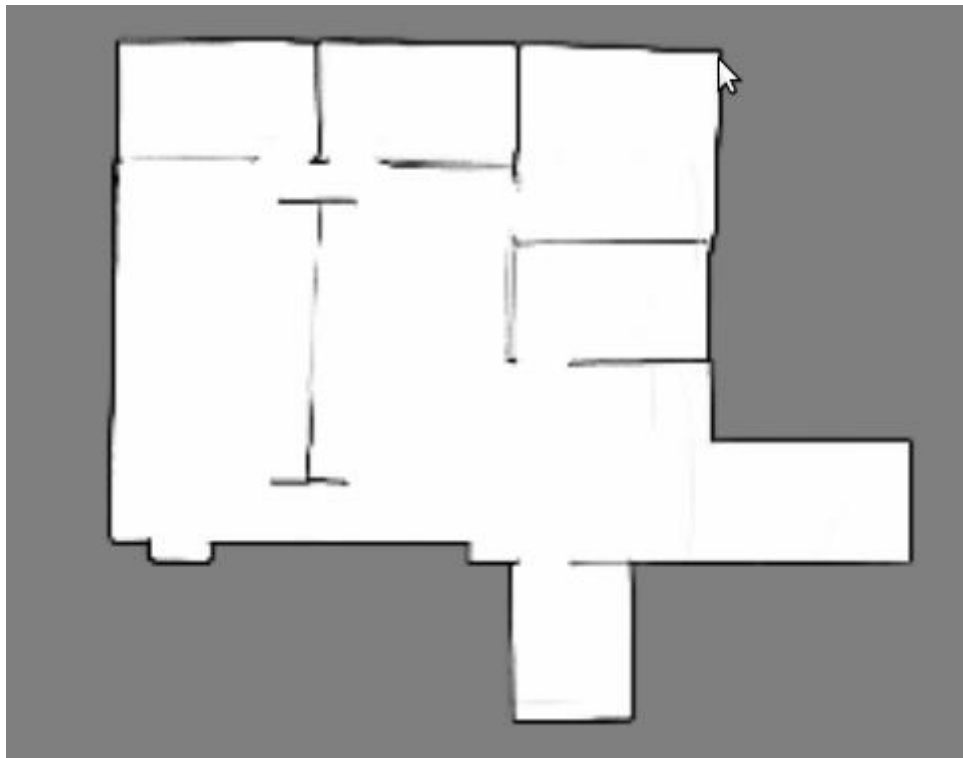
Lo que nos quiere decir es que al dar ese fallo no ha podido capturar la información y como consecuencia no pinta ciertas zonas del mapa en la ocurrió el error.



Al revisarlo vimos que el valor estaba mal lo que hacía que la distancia mínima del láser fuera demasiado grande y al pasar cercanos a las paredes como se efectúa en todo momento en nuestro recorrido nos daba errores y no dibujaba muchas partes del mapa. Una vez puestos los valores buenos desaparecieron los errores y el mapa se dibujaba completamente.



Captura del mapa dibujado mediante mricp:



En el siguiente apartado cambiamos ciertos parámetros para comprobar su utilización mediante la comparación del resultado, pero el único cambio que notamos a priori fue que aparecía el error del principio otra vez.

```
driver
(
    name "mricp"
    provides ["position2d:2" "map:0"]
    requires ["laser:0" "position2d:0"]
    MAXR 7.0
    MINR 0.05
    period 0.7
    map_resolution 0.07
    map_size 20
    interpolate 0
    NIT 15
    gate1 0.6
    gate2 0.05
    player_debug 0
    laser0_ranges [-90 90]
    number_of_lasers 1
    use_max_range 0
    use_odom 0
    start_in 0
    map_saving_period 7
    robot_id 0
    sparse_scans_rate 2
    free_space_prob 0.2
    log 1
    alwayson 1
)
```

#### Parámetros modificados

MAXR- rango máximo del láser 7.8→7.0

MINR- rango mínimo del láser 0.02→0.05

Interpólate- se probó como posible solución al error comentado anteriormente y vimos que falla al ejecutar.

NIT- número de iteraciones para cada exploración 30→15

Use\_odom- sirve para corregir el escaneo del láser mediante la odometría, ponemos a 0.





Aunque el recorrido y el log que capturamos para la siguiente parte los realiza correctamente vuelve a aparecer el error del principio lo cual no le permite dibujar el mapa correctamente. Pero si usamos el log del mismo el recorrido sale correcto.

### 3 PMAPTEST (offline)

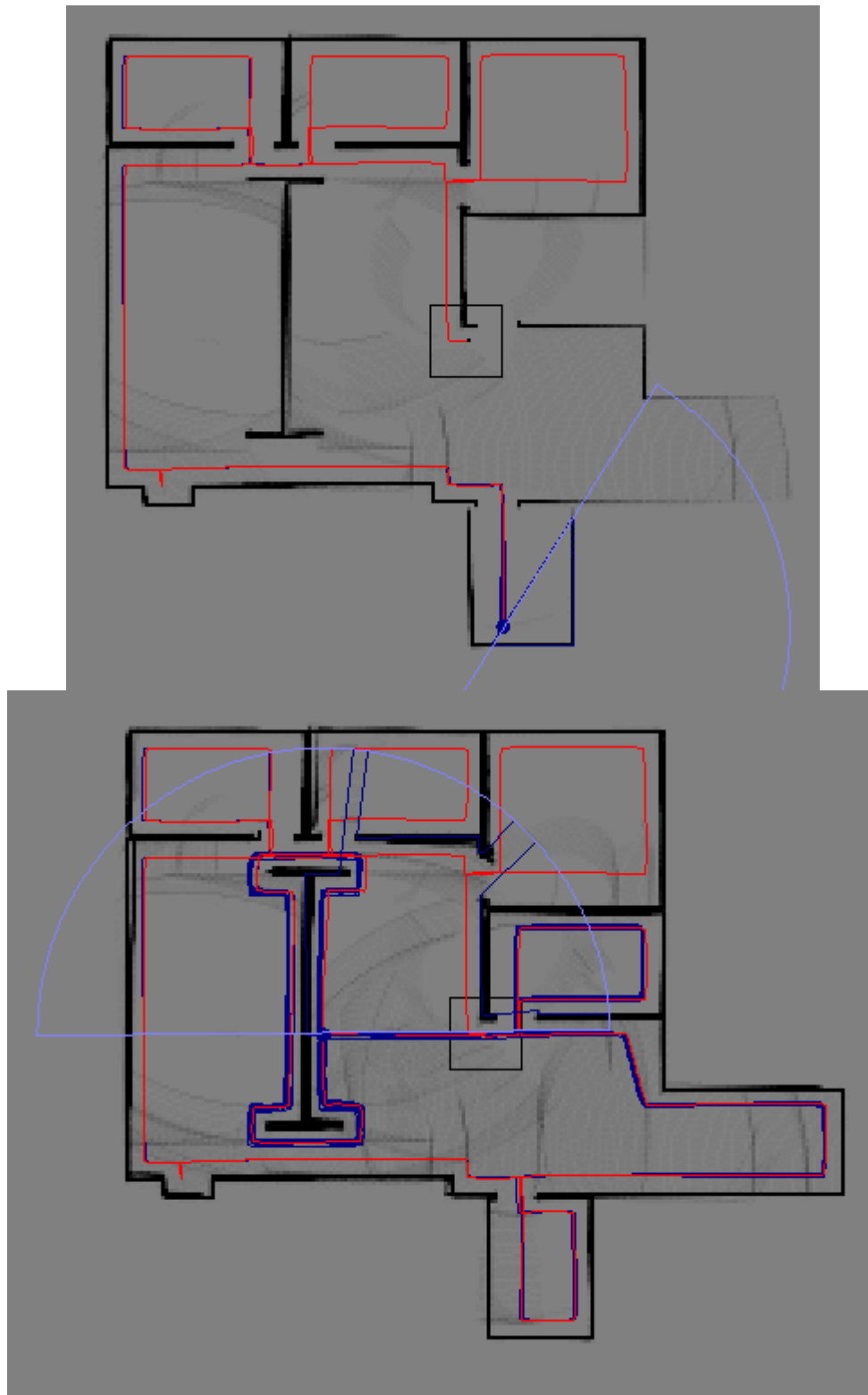
Al realizar un recorrido ya sea mediante el uso de playerjoy, un recorrido prediseñado o en paralelo al usar mrcip, podemos obtener un log de la información que va analizando mediante el siguiente driver

```
driver (
name "writelog"
filename "mydata.log"
requires ["laser:0" "position2d:0"]
provides ["log:0"]
alwayson 1
autorecord 1
)
```

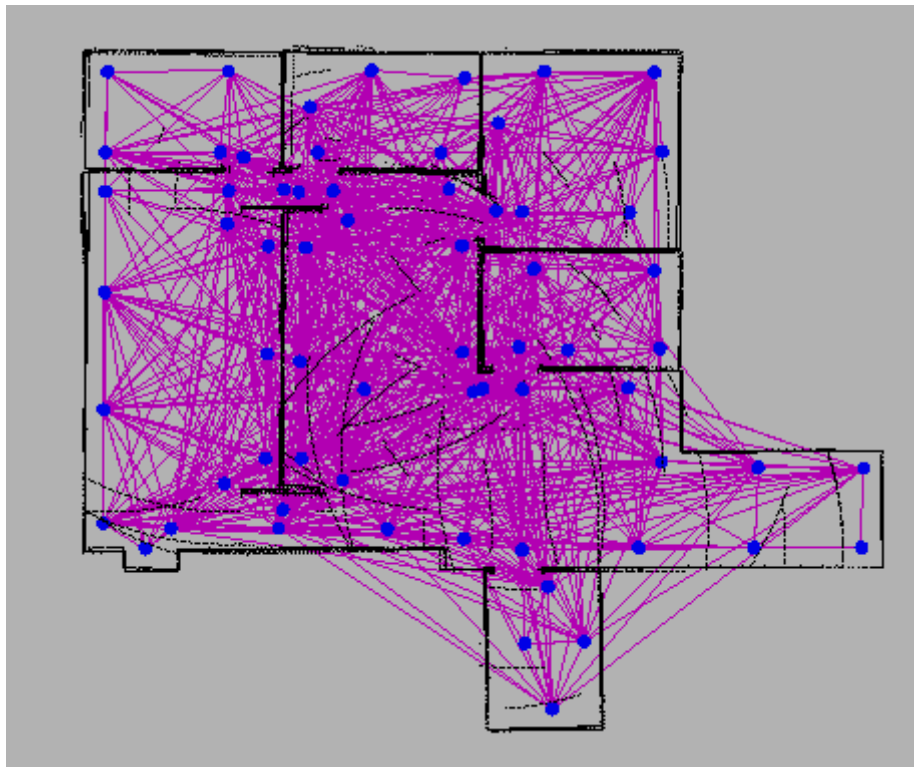
Añadiendo la parte de código del log obtenemos un archivo con el camino recorrido de este tipo.

```
## Player version 3.0.1
## File version 0.3.0
## Format:
## - Messages are newline-separated
## - Common header to each message is:
##   time      host      robot  interface index  type  subtype
##   (double) (uint) (uint) (string) (uint) (uint) (uint)
## - Following the common header is the message payload
0000000000.200 0 6665 laser 00 004 001 +0.000 +0.000 0.000
0.156 0.155
0000000000.400 0 6665 position2d 00 004 001 +00.000 +00.000 +
0.000 +00.330 +00.280
0000000000.400 0 6665 position2d 00 001 001 +00.000 +00.000 +
0.000 +00.000 +00.000 +00.000 0
0000000000.400 0 6665 laser 00 001 001 0001 -1.5708 +1.5708 +
0.00870247 +8.0000 0361 4.480 0 4.480 0 4.481 0 4.482 0 4.483
0 4.484 0 4.486 0 4.488 0 8.000 0 8.000 0 8.000 0 8.000 0
8.000 0 8.000 0 8.000 0 8.000 0 8.000 0 8.000 0 8.000 0
8.000 0 8.000 0 8.000 0 8.000 0 8.000 0 8.000 0 8.000 0
8.000 0 8.000 0 8.000 0 4.755 0 4.724 0 4.738 0 4.753 0
4.768 0 4.783 0 4.799 0 4.815 0 4.832 0 4.849 0 4.867 0
4.885 0 4.904 0 4.923 0 4.943 0 4.964 0 4.984 0 5.006 0
5.028 0 5.051 0 5.074 0 5.098 0 5.122 0 5.147 0 5.173 0
5.199 0 5.227 0 5.254 0 5.283 0 5.312 0 5.342 0 5.372 0
5.404 0 5.436 0 5.469 0 5.503 0 5.538 0 5.573 0 5.610 0
5.647 0 5.685 0 5.724 0 5.765 0 5.806 0 5.848 0 5.892 0
5.936 0 5.982 0 6.028 0 6.076 0 6.126 0 6.176 0 6.228 0
6.281 0 6.336 0 6.389 0 6.445 0 6.501 0 6.560 0 6.618 0
6.681 0 6.743 0 6.810 0 6.874 0 6.945 0 7.039 0 7.109 0
```

Al ejecutar el pmaptest con el log anterior mediante la interfaz gráfica podemos apreciar en las imágenes posteriores cómo va realizando el mismo recorrido que hemos realizado en la otra prueba y como se va dibujando el mapa en función de las medidas obtenidas por el láser que están en el log.



En la siguiente imagen se puede apreciar la intersección entre los nodos según el algoritmo que implementa el pmaptest internamente. Como curiosidad de la implementación pmaptest.



La siguiente captura es el resultado de dibujar el mapa con pmaptest usando el log anterior.



Repitiendo el proceso con los parámetros modificados obtenemos un mapa similar al anterior exceptuando algunas medidas que tienen menos resolución.



## **Conclusiones**

¿Cuál es la técnica más fácil de implementar y configurar?

Pmaptest ya que no es necesario configurar casi nada, solo hemos de contar con un archivo log en el cual se encuentre el mapa ya recorrido y mediante una sola instrucción podemos obtener el mapeado.

¿Con cuál se obtiene mayor fidelidad del entorno?

Sin contar la técnica map, la que representa con mayor fidelidad el entorno es la online mcrip ya que también es la que más parámetros de configuración posee, lo que nos permite ajustar mas la resolución del mapeado.

¿Cuáles son las ventajas e inconvenientes de cada una de las técnicas?

Mcrip Ventajas: Se puede obtener mayor precisión

Se puede ejecutar online y offline

Mcrip Inconvenientes: Es más complejo de configurar

PmapTest Ventajas: Es más simple

Resultados bastante buenos

PmapTest Inconvenientes: Se necesita tener un log del recorrido

Si el recorrido .log es malo obtendremos malos resultados

## **Robot Real**

Como parte opcional a la práctica se nos permito probar con el robot real, para ello el profesor nos suministró un robot equipado con un láser y con los drivers ya configurados de tal manera que solo era necesario lanzar el programa y navegar con el mediante playerjoy.

También se nos informó de que se ha de tener en cuenta que el driver del láser instalado en el robot devuelve un "range" pero el robot solo admite "laser", por lo que es necesario un driver para convertir de uno a otro ya que, aunque sean distintos al final son lo mismo.

Una vez obtenido el log del recorrido realizado con el robot pasamos a obtener el mapa con las técnicas implementadas en la práctica.

### **MRICP**

Para este caso solo hemos de modificar el archivo .cfg con los siguientes parámetros para poder obtener la información del log y representarla.

```
Driver(  
    Name "readlog"  
    Filename "mydata.log"  
    Provides ["position2d:0" "laser:0" "log:0"]  
)
```

```
Driver(  
    Name "mricp"  
    Provides ["position2d:2" "map:0"]  
    Requires ["position2d:0" "laser:0"]  
    MAXR 7.8  
    MINR 0.02  
    Period 0.7  
    Map_resolution 0.07  
    Map_size 20  
    Interpolated 0  
    NIT 15  
    Gate1 0.6  
    Gate2 0.05  
    Player_debug 0  
    Laser0_ranges [-90 90]  
    Number_of_lasers 1  
    Use_max_range 0  
    Use_odom 1  
    Start_in 0  
    Map_saving_period 7  
    Robot_id 0  
    Sparse_scans_rate 2  
    Free_space_prob 0.2  
    Log 1  
    Alwayson1  
)
```

Con dicho driver solo hemos de lanzar la instrucción `player archive.cfg` y el solo va trazando el mapa según va obteniendo los datos del log mediante el driver `readlog`.

### PMAPTEST

Para este apartado solo hemos de lanzar el comando `pmaptest archivo.log` y el programa empezara a trazar el mapa a través de los log registrados.

### NOTA:

Durante la ejecución del recorrido se desconectó o fallo algo por lo que solo se recorrió el primer metro, por ello no se han subido capturas por no resultar relevantes.