

## INTRODUCTION

In this activity, you are to write a robot simulation program, which will move random positions and will avoid hitting the wall. Remember that in your last tutorial, your robot was hitting the wall and got crashed. Hence to avoid this, we will use laser device mounted in the robot. This will help us in determining the closeness to the wall. Again the movement will be performed randomly. Your task is to print the trace positions both in local (robot) coordinate and global coordinate system and keep moving in the space provided.

## SETTING UP THE FILES

Download *wallavoid.tgz* file from LMS or dropbox in *clab* directory. The *tgz* file is a tar-gunzip compressed file. You need to untar the file by running the following command.

```
tar -xvzf wallavoid.tgz
```

Copy your *move.c* file and *makefile* from the *playerstage* tutorial to *wallavoid* directory. You will have to modify the program to read laser device sensor from the robot in the simulation world. Make sure that you have the following files in the *wallavoid* directory.

```
laser.cfg  
laser.world  
table.png  
pioneer.inc  
map.inc  
sick.inc
```

Use the same *makefile*, as the previous one to compile your program. Run your code on *player-stage* model as follows:

```
player laser.cfg &  
./move
```

Check the movement of the robot in your simulation GUI.

## TASK: ADDING LASER DEVICE

Your *laser.cfg* file provides a laser device to the robot sitting in the simulation world. To read the laser, your program needs to create and subscribe laser device to the client structures. Hence you might have to use the following functions accordingly.

```
playerc_laser_create &  
playerc_laser_subscribe
```

The *playerc\_laser\_t* structure has two distance members: *min\_left* and *min\_right* which can be used to find the obstacles. Also remember that laser takes around few milliseconds, to read the data. On knowing the obstacles your program is supposed to move away and continue moving in the environment. Note that your program should avoid hitting the wall as compared to previous tutorial.

## SUBMISSION

Make sure you are in the *wallavoid* directory.

Then run the command:

```
submit clab mr wallavoid <your-iiitb.org-email-address>
```

Submit your code. Note that your final exam will have a question on player-stage program.