Prática 1 - Robótica Móvel

Lucas Nogueira Nóbrega,
Mateus Antonio da Silva
Universidade Federal da Paraíba (UFPB)
João Pessoa, Brasil
Email: lucasnnobrega@eng.ci.ufpb.br, mateus.antonio@eng.ci.ufpb.br

***Abstract*—O presente relatório descreve as atividades da Prática 1 realizadas na disciplina de Robótica no período letivo 2019.2. Nessa prática alguns exercícios foram realizados para o aprendizado dos alunos, envolvendo uso dos *softwares* Gazebo 3D, ROS e Matlab, a fim de integrá-los e produzir resultados na observação e controle do robô TurtleBot.**

I. Introdução

Esta prática consiste na utilização do *TurtleBot* e *Gazebo* usando a teleoperação com e sem o MATLAB, na criação de obstáculos usando o MATLAB , e na conversão de um mapa em *bitmap* para *OcupancyGrid*. Para integrá-los e realizar as teleoperações, o ROS (*Robot Operating System*) é essencial, pois contém todos os módulos referentes a abstração de hardware, controle E funcionalidades do robô, além da troca de mensagens entre o *Turtle Bot* e MATLAB.

II. Resultados Obtidos

A. Parte 1

Durante a aula foi possível realizar a primeira parte da prática 1 com sucesso pois foi feito o controle do robô *Turtle Bot* com o teclado, usando o *software* já instalado na maquina virtual. Após a configuração das variáveis de ambiente e do *teleop* para controle do robô, foi possível notar um pequeno *delay* na realização das manobras, porém o turtlebot respondeu perfeitamente a todos os comandos enviados(Ver figura 1). As instruções responsáveis para comandar o turtlebot pelo teclado se encontram abaixo, utilizando as próprias dependencias do ROS:

```
$ roslaunch turtlebot_teleop
keyboard_teleop.launch
```

Já as instruções para iniciar o Gazebo pelo terminal são:

```
$ roslaunch turtlebot_gazebo
turtlebot_mw_empty_world.launch
```

B. Parte 2

A teleoperação com o MATLAB se mostrou interessante, pois o controle foi realizado com dois notebooks, mostrando o controle do *Turtle Bot* no Gazebo a distância. Com as configurações de odometria e do LASER *scan* feitas no Matlab, foi possível observar não só a visão do robô através de um gráfico, como sua trajetória baseada nos comandos enviados pelo teclado. A visão do robô é atualizada de acordo com o posicionamento da sua câmera e com isso pode ser

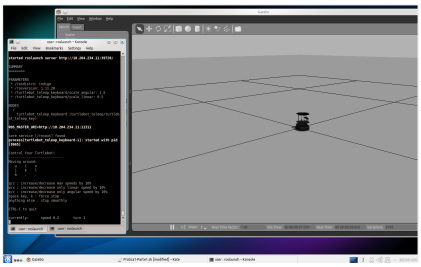

Fig. 1: Controlando Turtle Bot pelo terminal.

controlado mesmo sem a visualização do cenário 3D. Ver figura 2.

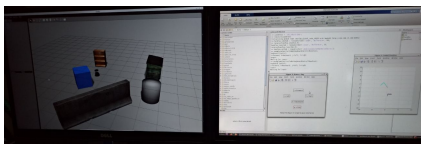

Fig. 2: Integração MatlabxGazebo entre dois computadores

C. Parte 3

Também utilizando dois computadores, foi realizado a criação de obstáculos utilizando a comunicação entre o MATLAB e Gazebo. Estes obstáculos consistiram em muros, esferas , entre outros objetos no qual o robô pode detectar usando os sensores embutidos nele. Na figura 3 pode observar o comportamento da prática.

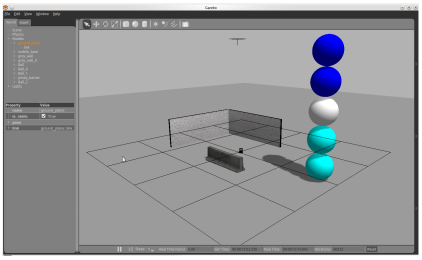

Fig. 3: Gazebo com obstáculos criados pelo MATLAB

D. Parte 4

A criação do mapa foi proveniente da imagem na prática, em que tiramos uma foto em 450 por 360 *pixels* e depois foi salvo em formato *Bitmap*. Esta foto foi processada com as ferramentas do MATLAB para que a proporção entre os *pixels* e metros ficassem na escala correta, por isso foi escolhido que a resolução fosse de 50, como mostra na figura 4. Após esse algorítimo geramos o seguinte *Ocupancy Grid*.

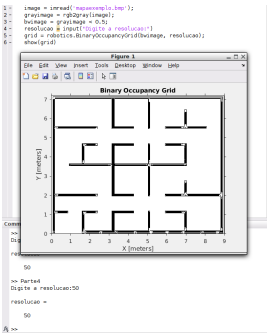

Fig. 4: *Ocupancy Grid*

III. Discussão dos Resultados

A primeira parte da prática aconteceu sem problemas pois todos os comandos para a teleoperação como os para iniciar o Gazebo foram realizados na máquina virtual, que se encontra devidamente configurada para *download* no site do MATLAB.

A segunda parte ofereceu certas dificuldades de conexão do Matlab na máquina *host* com o Gazebo na máquina virtual, porém foi superado através das devidas configurações da placa de rede maquina virtual em modo *bridge* que possibilitou a comunicação da maquina virtual com um outro computador na mesma rede local. Isso nos forneceu uma paralelização das atividades e do processamento de dados, o que tornou a prática mais rápida.

Foi esperado que a terceira parte acontecesse mais rápida já que o processamento computacional era dividido entre duas máquinas. Porém ainda tivemos que esperar para que o primeiro objeto da figura fosse renderizado. Após a primeira renderização os outros objetos eram inseridos na simulação mais rapidamente.

A criação do *Ocupancy Grid* foi realizada com sucesso, considerando que tivemos que usar uma proporção correta para a converter de Bitmap.