

FUNDAÇÃO EDSON QUEIROZ

UNIVERSIDADE DE FORTALEZA

Centro de Ciências Tecnológicas – CCT

Disciplina – Sistemas Autônomos

Lidar

G. A. Davi.

Sumário

Resumo

Introdução

Materiais e métodos

Materiais usados

Baixar programas

Configurar

Ligar o sensor

Table of Contents

[EXECUÇÃO DO ROS E LASER 8](#_Toc524104475)

[... 8](#_Toc524104476)

[Estes dados estão dispostos em vetores e correspondem as distancias calculadas por cada ponto de infravermelho emitidos no ambiente pelo laser HOKUYO. Com o laser funcionando iremos visualizar os dados recebidos utilizando um software que já é instalado nativamente com o ROS. É um software de visualização, compatível com vários tipos de dados referentes a aplicações de robótica e muito utilizado no desenvolvimento dessas aplicações por possibilitar a visualização de forma abstraída de toda aplicação robótica em funcionamento, como também um mapeamento sendo realizado e a rota do robô pelo ambiente. 8](#_Toc524104477)

[Para executar o Rviz execute o seguinte comando: 8](#_Toc524104478)

[Rosrun rviz rviz 9](#_Toc524104479)

[Uma interface gráfica irá ser executada. Para estudar essa interface e aprender de forma completa o manuseio da mesma. Acesse o link: http://wiki.ros.org/rviz/UserGuide 9](#_Toc524104480)

[Na side bar do lado esquerdo, no canto inferior clique no botão “ADD” e procure o tópico /scan na lista, na janela que ira se abrir. Após adicionar o tópico /scan marque a check box na side bar. E você deverá visualizar os dados do laser. 9](#_Toc524104481)

[Retirado de https://github.com/afonsohfontes /Jaguar/blob/master/Relatorio.pdf página 8 e 9. 9](#_Toc524104482)

# Resumo

O intuito desse projeto é comparar a eficácia de algoritmos de localização e mapeamento aplicados ao sensor laser Hokuyo 30LX.

# Introdução

Objetivos:

Comparar algoritmos de localização e mapeamento

# Materiais e Métodos

## A materiais usados:

OS: MacOS High Sierra 10.13.6

PROCESSOR: 2,2 GHz intel core i7

RAM: 16 GB 1600 MHz DDR3

GRAPHICS: Intel Iris Pro 1536 MB

ROM: 200 GB

Para tal, foi adicionado uma partição ao disco de 50 GB com o sistema operacional Ubuntu 18.04, no qual foi instalado os softwares necessários.

Sensor lidar – Hokuyo 30lx

Matlab

## Baixar programas

Baixar o Linux Ubuntu 18.04 https://www.ubuntu.com/download/desktop /thank-you?country=BR&version=18.04.1&architecture=amd64

Preparar o Dual Boot e instalar o Ubuntu https://www.maketecheasier.com/ install-dual-boot-ubuntu-mac/ (Caso encontre problemas com o particionamento do disco, leia resolução de problemas no anexo)

Instalar ROS Melodic Desktop-Full Install http://wiki.ros.org/melodic/Installation /Ubuntu

Instalar Gazebo <http://gazebosim.org/tutorials?tut=ros_installing>

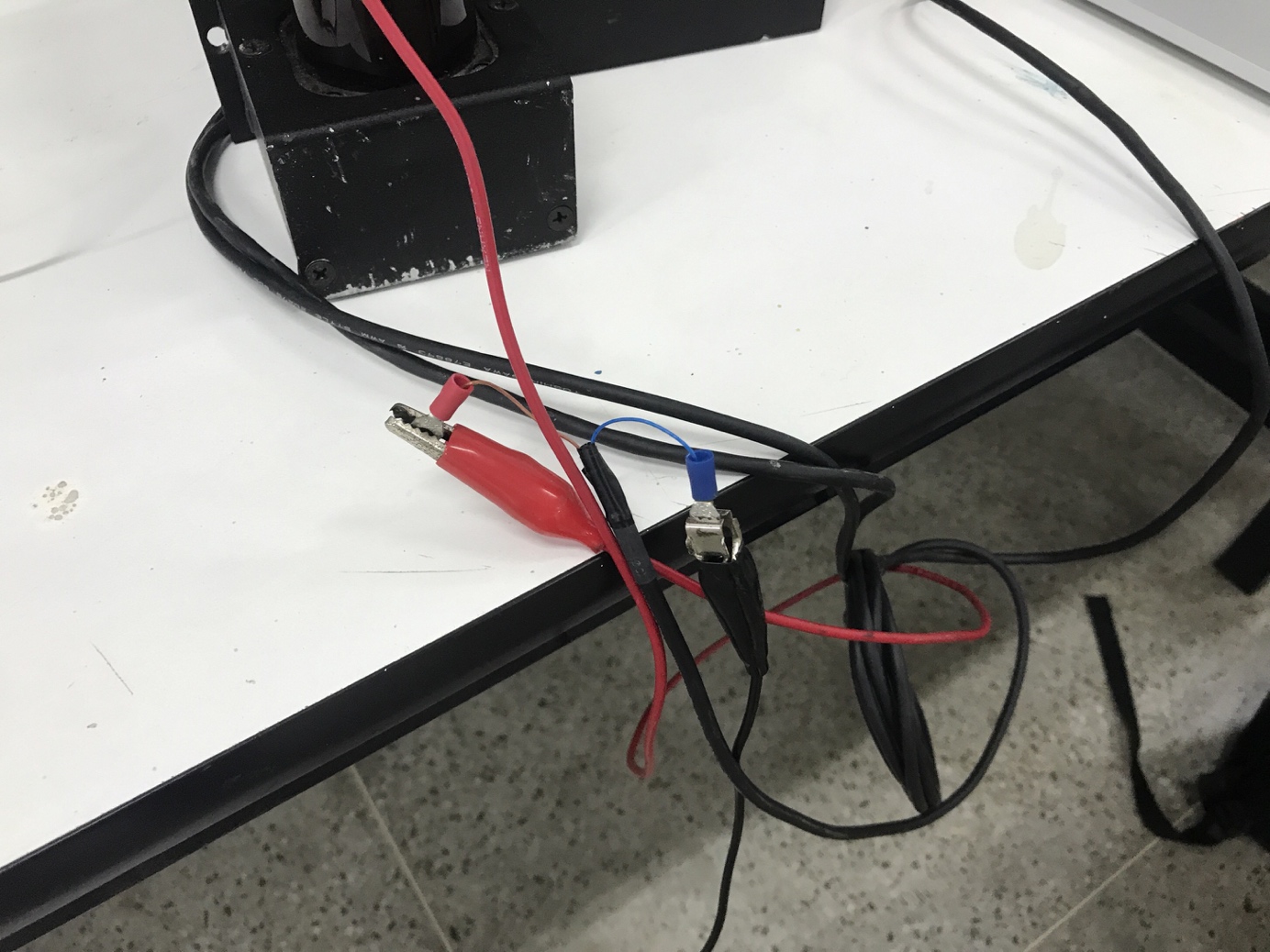
## Ligar o sensor

Para ligar o sensor você deve alimentar o cabo marrom com 12 volts e o cabo azul com 0 volt. O sensor consome 1 ampere e 0.7 de corrente máxima e normal, respectivamente. Após os dois LEDs indicadores acenderem, um vermelho outro verde, conecte o cabo USB do sensor ao computador.

Para saber mais sobre esse aparelho, leia os links abaixo. <https://www.robotshop.com/media/files/pdf/utm-30lx-overview.pdf> <http://www.lara.prd.fr/_media/imara/platforms/hardware/mobilink/lasers/utm-30lx_um.pdf>

Abaixo as imagens de como liguei o sensor:







## Configurar

Configurar workspace: http://wiki.ros.org/catkin/Tutorials/create\_a\_workspace

## Instalar o package:

Baixe o package Laser Hokuyo - Driver Hokuyo Node em <https://github.com/ros-drivers/hokuyo_node> e cole na pasta src do seu workspace.

Abra o terminal de comandos e mude o diretório para o seu workspace

$ cd ∼/home/user/your\_workspace

Execute catkin\_make para configurar e instalar o package

$ catkin\_make

Se houver algum erro na execução do comando anterior, refaça as etapas anteriores.

Caso algum erro seja lançado na execução do comando anterior, é provável que seja necessário instalar antes outro package. Leia em Solução de Problemas.

Agora vamos saber se tudo está funcionando corretamente.

Abra um terminal e inicialize o roscore:

$ roscore

Abra um novo terminal e execute

$ source ~/your\_workspace/devel/setup.bash

$ rosrun hokuyo\_node hokuyo\_node

Caso tenha algum problema aqui, verifique se há correção na sessão Solução de Erros

Obs: você deve executar o comando “source ~/your\_workspace/devel/setup.bash” toda vez que abrir um novo terminal para associar o mesmo com o seu workspace, como discutido aqui: <https://answers.ros.org/question/206876/how-often-do-i-need-to-source-setupbash/>

Se tudo tiver ocorrido bem, o topic /scan já deve ter sido criado para receber os dados do LIDAR. Você pode verificar se os dados realmente estão chegando.

Abra um novo terminal e digite

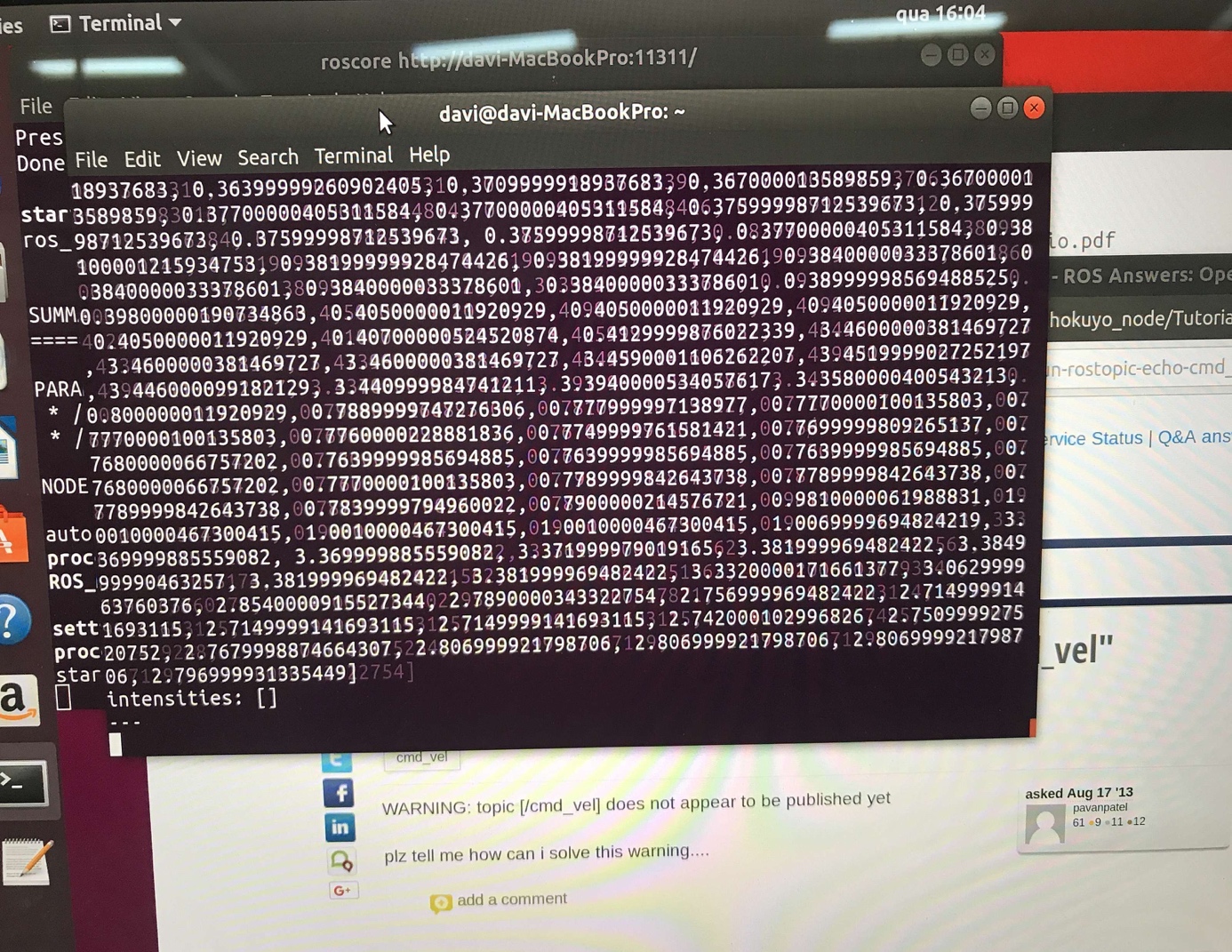
$ rostopic list

Deve aparecer um topic de nome /scan

Digite

$ rostopic echo /scan

A leitura do sensor deve aparecer na tela.



Execute o ROS e o Lidar (retirado de https://github.com/afonsohfontes /Jaguar/blob/master/Relatorio.pdf em 28 de agosto de 2018)

*EXECUÇÃO DO ROS E LASER*

*...*

*Estes dados estão dispostos em vetores e correspondem as distancias calculadas por cada ponto de infravermelho emitidos no ambiente pelo laser HOKUYO.  
Com o laser funcionando iremos visualizar os dados recebidos utilizando um software que já é instalado nativamente com o ROS. É um software de visualização, compatível com vários tipos de dados referentes a aplicações de robótica e muito utilizado no desenvolvimento dessas aplicações por possibilitar a visualização de forma abstraída de toda aplicação robótica em funcionamento, como também um mapeamento sendo realizado e a rota do robô pelo ambiente.*

*Para executar o Rviz execute o seguinte comando:*

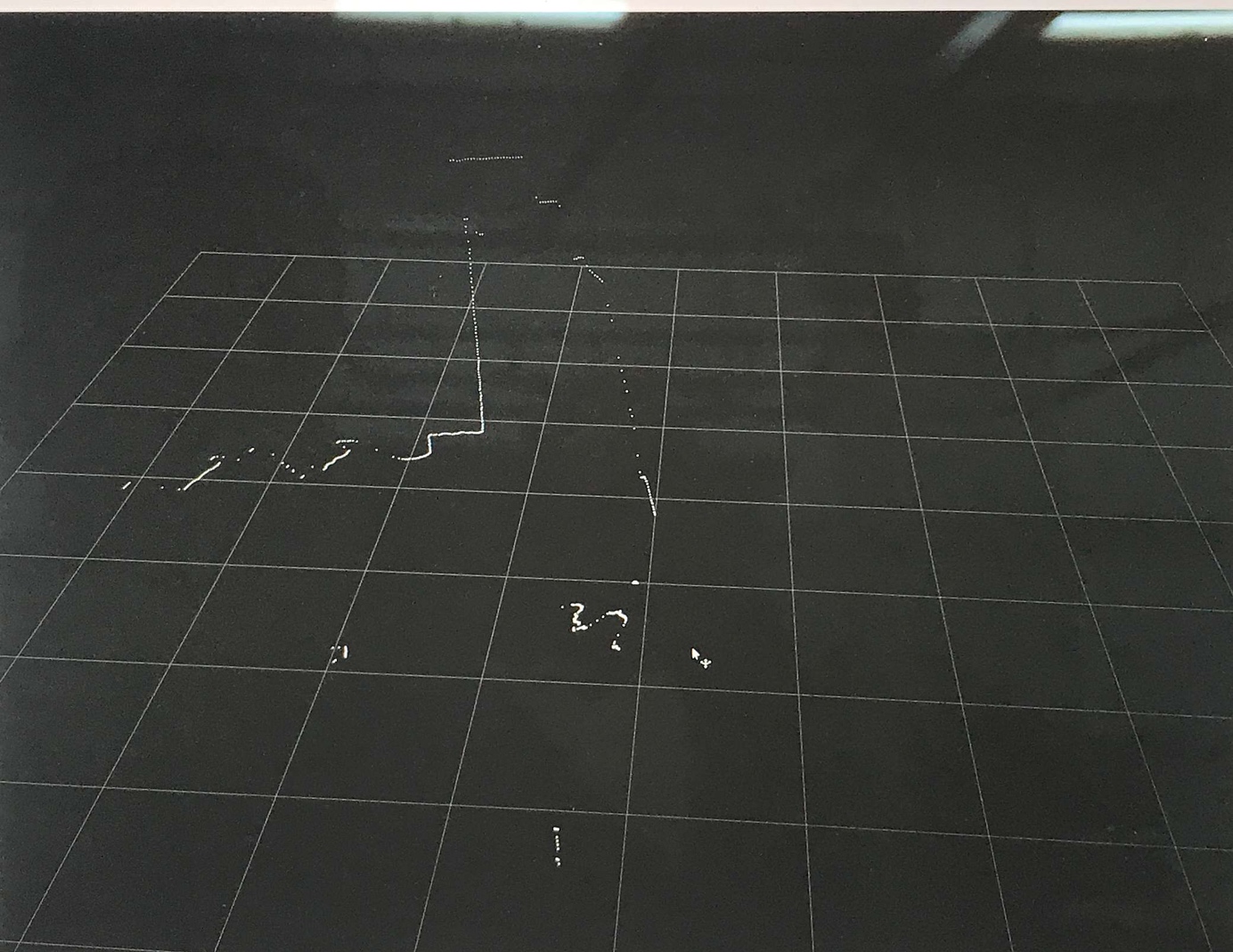
*Rosrun rviz rviz*

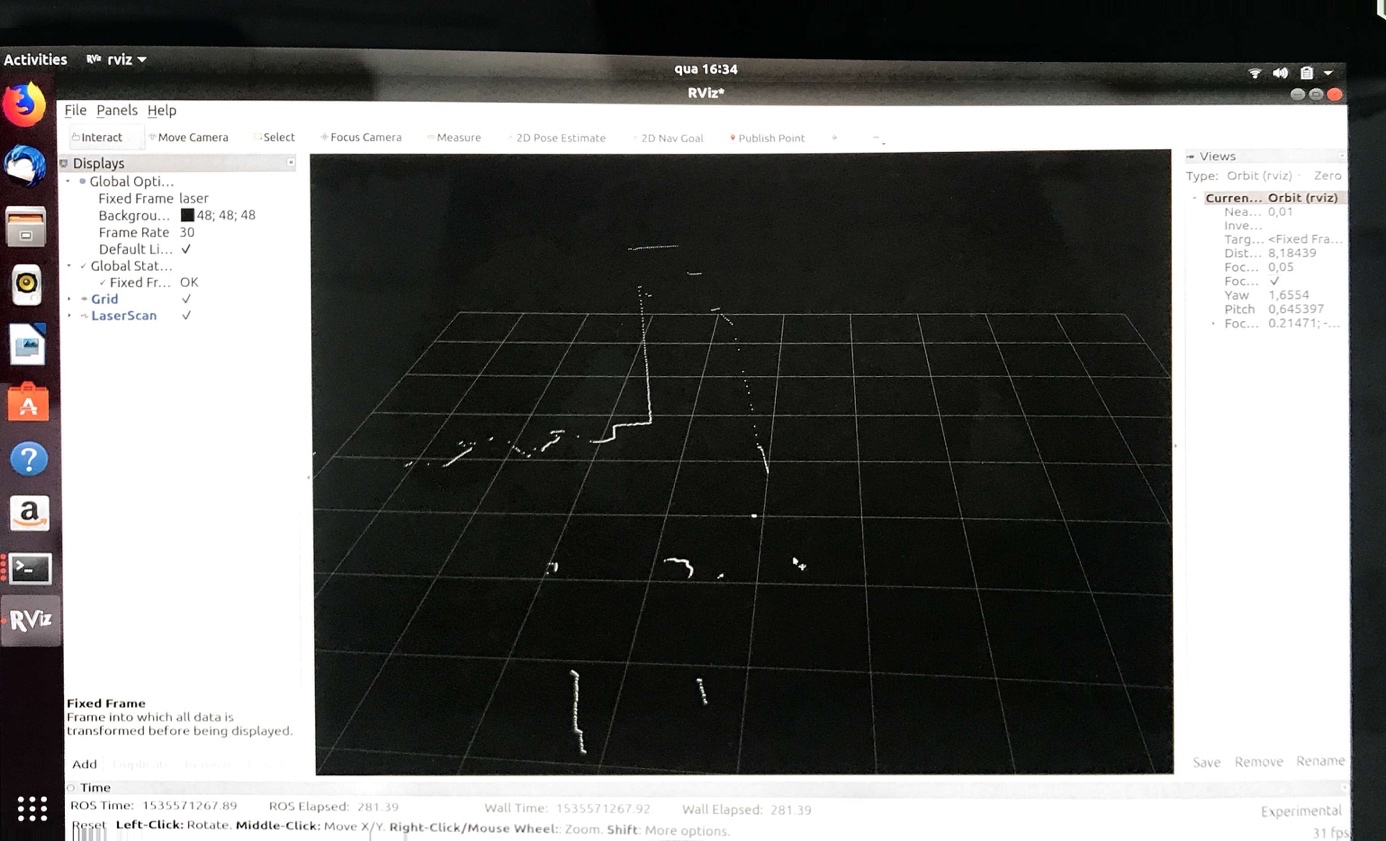
*Uma interface gráfica irá ser executada. Para estudar essa interface e aprender de forma completa o manuseio da mesma. Acesse o link: http://wiki.ros.org/rviz/UserGuide*

*Na side bar do lado esquerdo, no canto inferior clique no botão “ADD” e procure o tópico /scan na lista, na janela que ira se abrir. Após adicionar o tópico /scan marque a check box na side bar. E você deverá visualizar os dados do laser.*

Retirado de https://github.com/afonsohfontes /Jaguar/blob/master/Relatorio.pdf página 8 e 9.

No rviz deverá ser parecido como na imagem abaixo (no formato da sua sala).





Baixe os packages slam\_gmapping, open\_slamgmapping, gmapping nos links abaixo: <https://github.com/ros-perception/openslam_gmapping> <https://github.com/ros-perception/slam_gmapping>

Obs: você deve refazer os mesmos passos feitos para a instalação do driver do sensor. Para cada package adicionado na pasta src, você terá que executar um catkin\_make.

Abra o terminal e execute

$ cd ∼/home/user/your\_workspace

Cole primeiro o package slam\_gmapping em src no seu workspace, em seguida execute

$ catkin\_make

Cole o package open\_slamgmapping em src e execute

$ catkin\_make

Por fim, cole gmapping em src e execute

$ catkin\_make

Se tudo tive ocorrido bem, o package gmapping estará pronto para uso.

## Gmapping

Encerre todas as execuções que envolvam o ROS. Para tal, recomendo que reinicie o computador. Em seguida:

Abra um novo terminal e execute:

$ roscore

Abra um segundo terminal:

$ source ~/your\_workspace/devel/setup.bash

$ rosrun hokuyo\_node hokuyo\_node

Abra um terceiro terminal:

$ rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 laser odom 100

Abra um quarto terminal:

$ rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 odom map 100

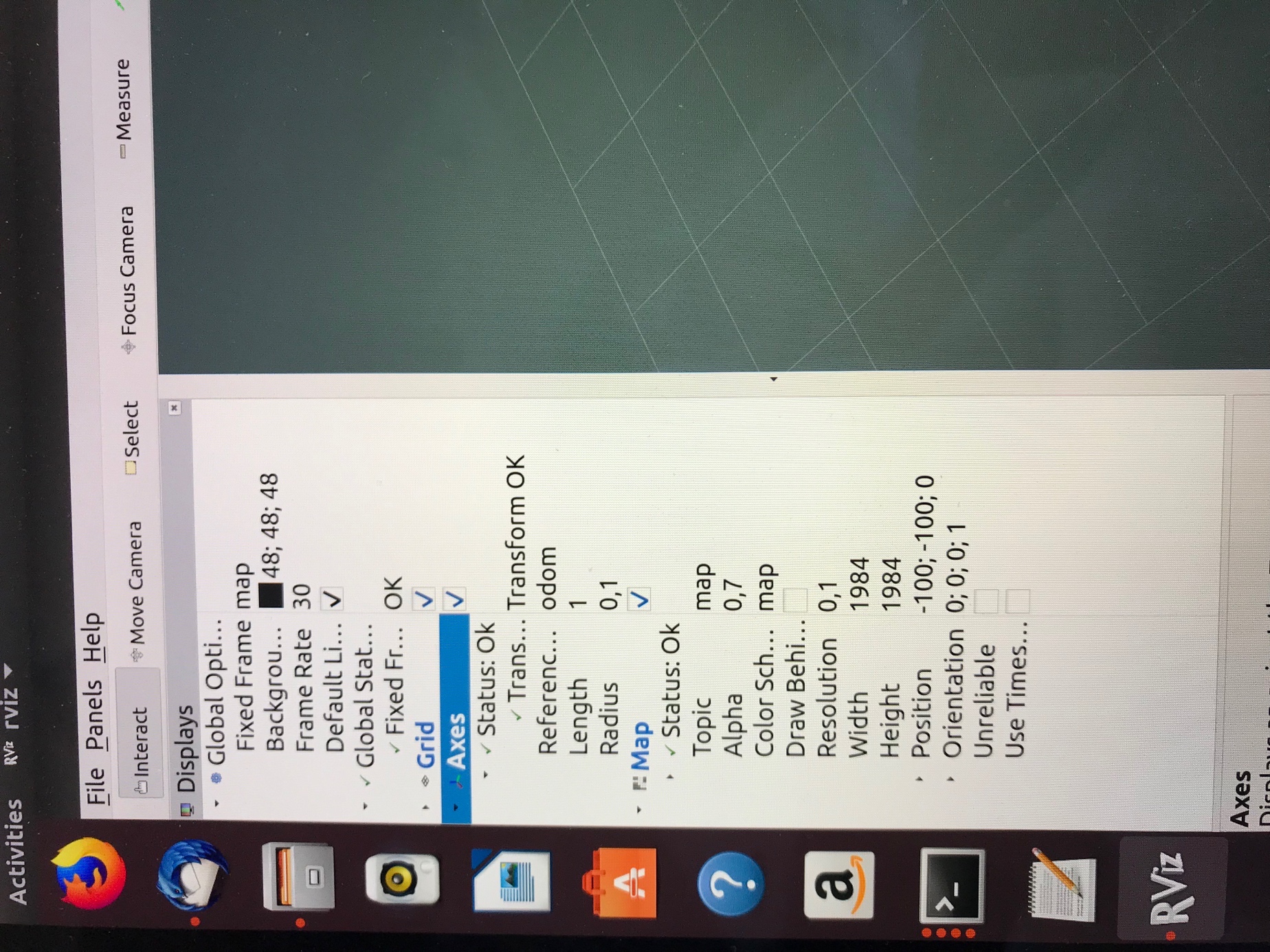
Execute em um sexto terminal:

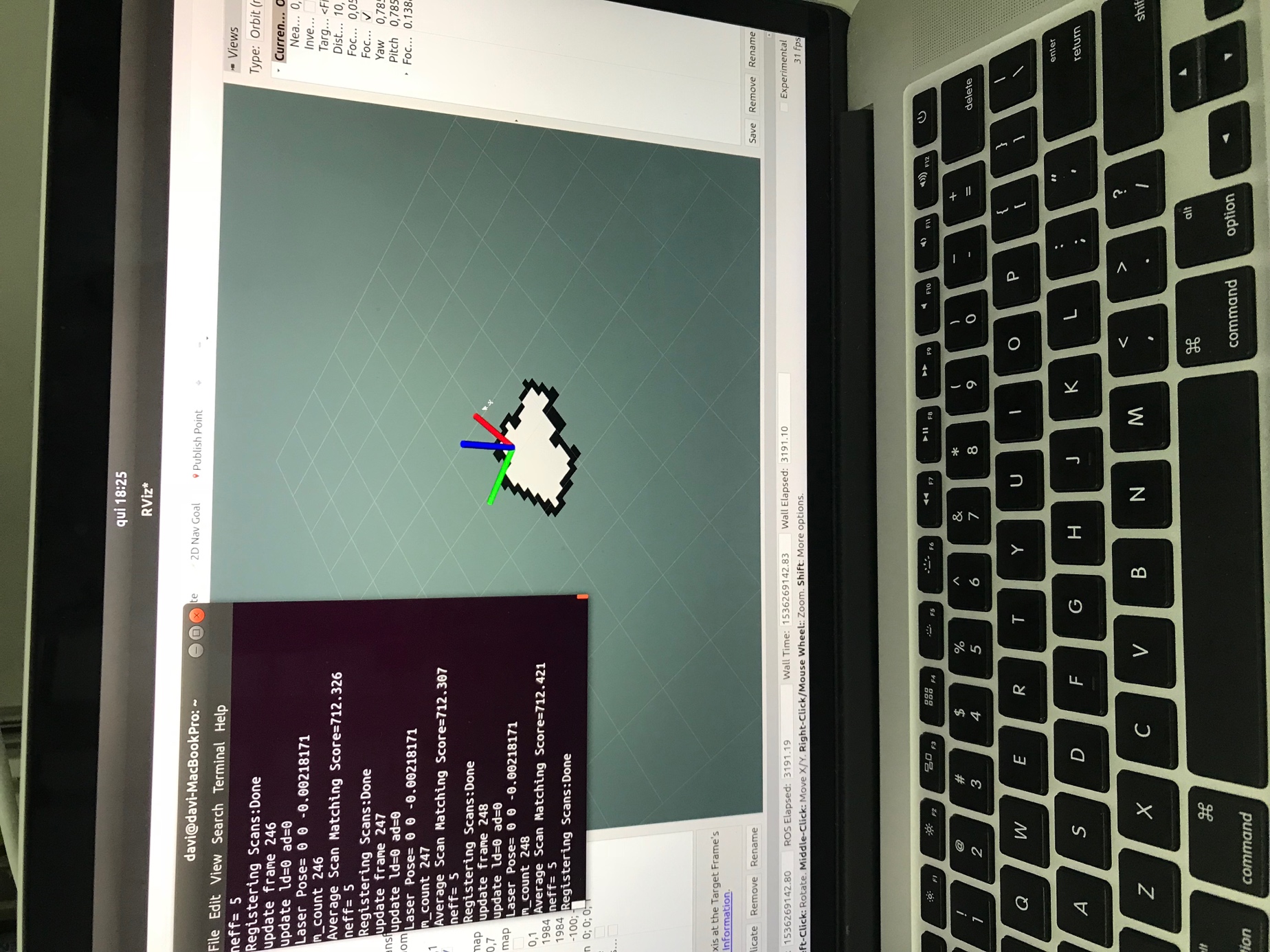
$ rosrun gmapping slam\_gmapping scan:=scan \_linearUpdate:=0.0 \_angularUpdate:=0.0 \_map\_update\_interval:=0.0 \_particles:=3 \_delta:=0.1 \_base\_frame:=laser

(Para mais informações sobre o commando acima, leia Para mais informações: http://wiki.ros.org/tf)

Execute em um sétmo terminal:

$ rviz rviz

No rviz, clique no botão ADD, no canto inferior esquerdo, e adicione Map e Axes. Em seguida, ajuste o topic do Map para /map, o topic do Axes para /odom e o Fixed Frame do Global Options para map. Após as configurações deverá parecer como o da foto.





## MRPT - Navegation

Baixar MRPT – Navegation que pode ser encontrado aqui: <https://github.com/mrpt-ros-pkg/mrpt_navigation>, colar a pasta no src do seu workspace e, por fim, executar & catkin\_make, da mesma forma feita para instalar o package Hokuyo\_node.

<http://wiki.ros.org/mrpt_localization>

Para o erro

para adicioanr o map

Usage: static\_transform\_publisher x y z qx qy qz qw frame\_id child\_frame\_id period(milliseconds)

rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 0.0 map laser 100

Desligou o pc e não sabe como reabrir tudo?? Siga o passo a passo abaixo:

1. Novo terminal
2. $ roscore (ativar o roscore)
3. Novo terminal
4. $ cd <seu\_workspace> (exemplo: $ cd catkin\_ws)
5. $ source devel/setup.bash (Por garantia, faça isso em todo novo terminal)
6. $ rosrun hokuyo\_node hokuyo\_node (iniciar o sensor)
7. Novo terminal
8. $ source devel/setup.bash (Por garantia, faça isso em todo novo terminal)
9. rostopic echo /scan (Verifica se você realmente está recebendo os dados do sensor)
10. $ rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 0.0 map laser 100 (se n fizer isso, ao abrir o rviz, aparecera um erro ao mapa. entao devemos adicionar nosso mapa laser, algo no fixed frame. Usage: static\_transform\_publisher x y z qx qy qz qw frame\_id child\_frame\_id period(milliseconds) ) Para mais informações: http://wiki.ros.org/tf
11. $ roslaunch mrpt\_localization demo.launch (iniciar a demo)

1 de setembro de 2018: Encontrei um erro no “map” da demo que ainda não consegui resolver, fazendo com que a mesma não opere como o esperado. Espero resolve-lo em breve.

A demo pode ser encontrada em: https://github.com/mrpt-ros-pkg/mrpt\_navigation/blob/master/mrpt\_localization/launch/demo.launch  
  
https://answers.ros.org/question/209276/rviz-slam-rplidar-problem/

<https://answers.ros.org/question/288945/rviz-errortransform-senderunknown_publisher-for-frame-x-frame-x-does-not-exist/>

<http://wiki.ros.org/mrpt_localization>

http://wiki.ros.org/mrpt\_navigation/Tutorials/Installing

Instalar o ros (core, para visualizar os dados, e transmitir via red)

Ler o Sensor

Monitorar e aplicar o slam a partir de um outro computador (com matlab e visualizar, aplicar o slam, no sensor dado)

Implementar via ros algoritmo de comparação entre dois algoritmos

Bibliografia

Evangelista, D.M. e Silva, R.F. e Pinto, I.R.S e AZEVEDO, E.A. Recuperação de funcionamento do Dr Robot Jaguar, Pages 5-9 Agust 2018, Pages 1342-1352. Disponível em: https://github.com/afonsohfontes/Jaguar/blob/master/Relatorio.pdf

# Solução de Problemas

## Debug

Para debugar alguma execução no ROS, abra um novo terminal e digite

$ rosrun rqt\_logger\_level rqt\_logger\_level

Obs: nao debugar o laser! o programa vai travar e será necessário reiniciar o pc

## Particionamento do disco:

Ao tentar fazer o dual boot, não conseguia adicionar uma partição ao disco. Mesmo tendo bastante espaço para o mesmo, o sistema acusade de não haver espaço suficiente. O problema foi gerado pelo Time Machine Snapshot, responsável pelo backup da máquina. Esse problema foi solucionado com a remoção dos Snapshots.

A solução encontra-se em: https://www.macworld.com/article/3260635/macs/how-to-delete-time-machine-snapshots-on-your-mac.html

Não consegui instalar no próprio mac.

## Instalação do Hokuyo Node:

Caso seja lançado uma mensagem de erro, durante a execução do catkin\_make, dizendo que faltou arquivos ou packages, você deve adicionar outros packages na pasta src do seu workspace (a mesma pasta do Hokuyo\_node)

Para solucionar, vc deve fazer:

1. Retirar todos os packages da pasta src
2. Adicionar o package driver\_common na pasta src, este você pode encontrar em: https://github.com/ros-drivers/driver\_common.git
3. Executar $ catkin\_make no terminal no diretório do seu workspace
4. Adicionar o package Hokuyo\_node na pasta src
5. Executar catkin\_make no terminal no diretório do seu workspace

https://answers.ros.org/question/240235/how-to-install-packages-from-github/

A idéia, como pode-se notar, é executar o catkin\_make toda vez que um novo pacote for adicionado ao workspace. Além disso, devemos também adicionar somente um pacote por vez, para evitar problemas. Outros pacotes que podem ser requistados:

<https://github.com/ros-drivers/urg_c>

<https://github.com/ros-drivers/urg_node>

## Erro ao executar Hokuyo\_node (failed to open port – No authorized):

### Caso haja problema ao executar

$ rosrun hokuyo\_node hokuyo\_node

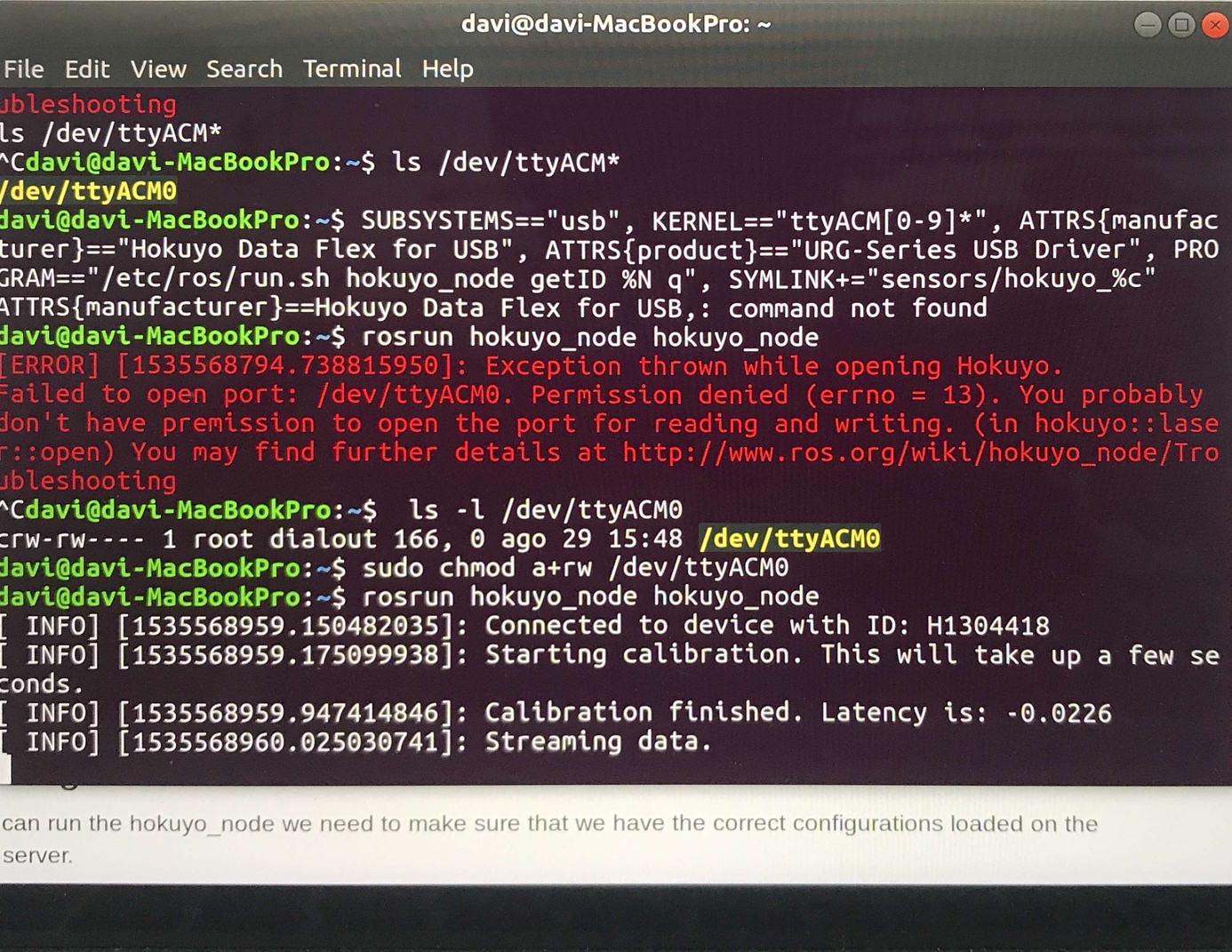
Por não ter autorização para acessar a porta usb, você deve ir para a raiz do sistema:

$ cd

e execute:

$ sudo chmod a+rw /dev/ttyACM0

Como na imagem abaixo.



### Caso o erro for:

[ERROR] [1536087827.457595928]: Exception thrown while opening Hokuyo.

Failed to open port: /dev/ttyACM0. No such file or directory (errno = 2). The requested port does not exist. Is the hokuyo connected? Was the port name misspelled? (in hokuyo::laser::open) You may find further details at http://www.ros.org/wiki/hokuyo\_node/Troubleshooting

Verifique se o sensor está ligado, e o cabo conectado corretamente ao computador.

Remova o cabo USB por alguns minutos e conecte-o novamente. Se o erro insistir, reinicie o computador.

Instalar o hokoyo\_node:

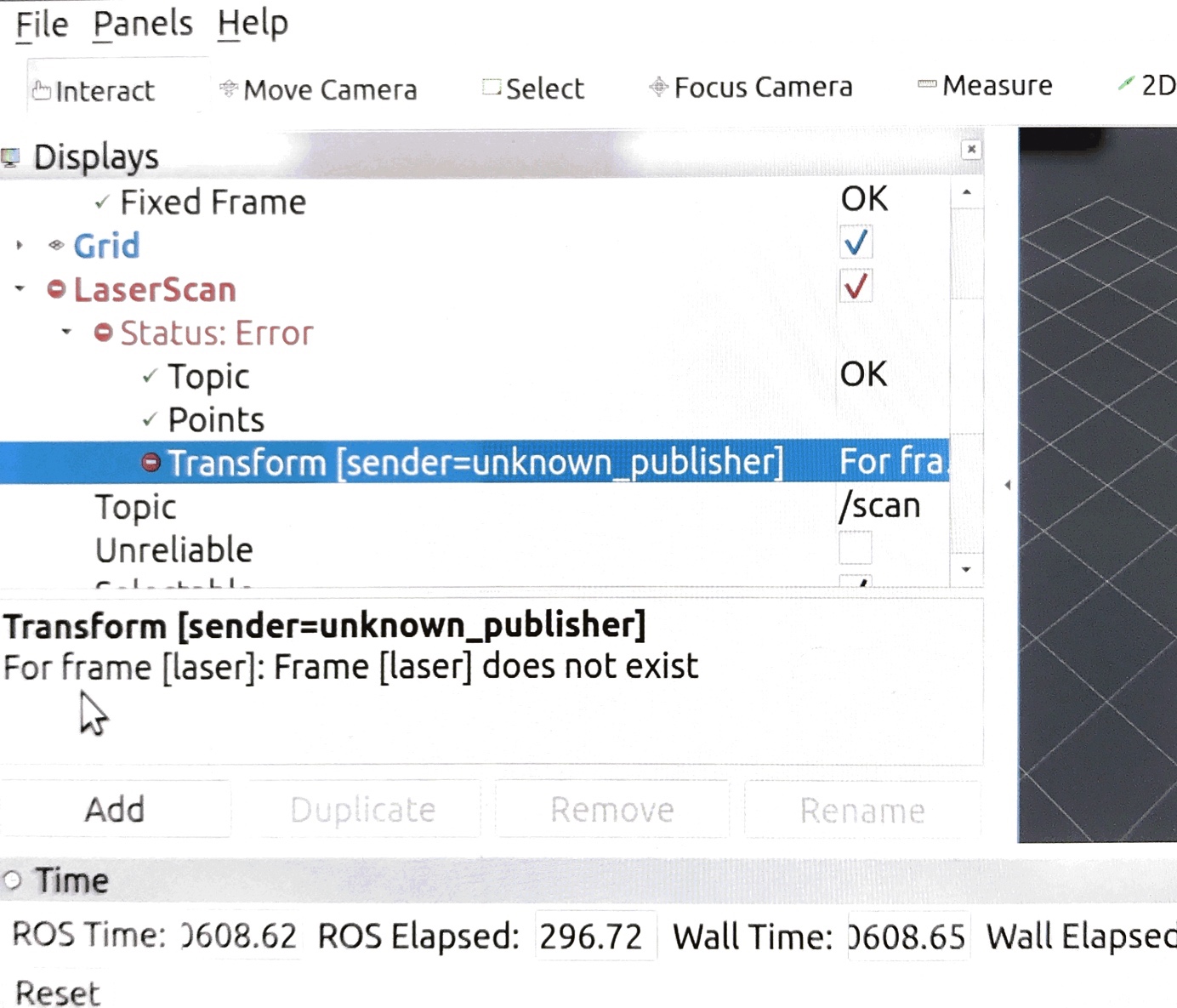
add o driver\_common

catkin\_mak

dps add hokuyo

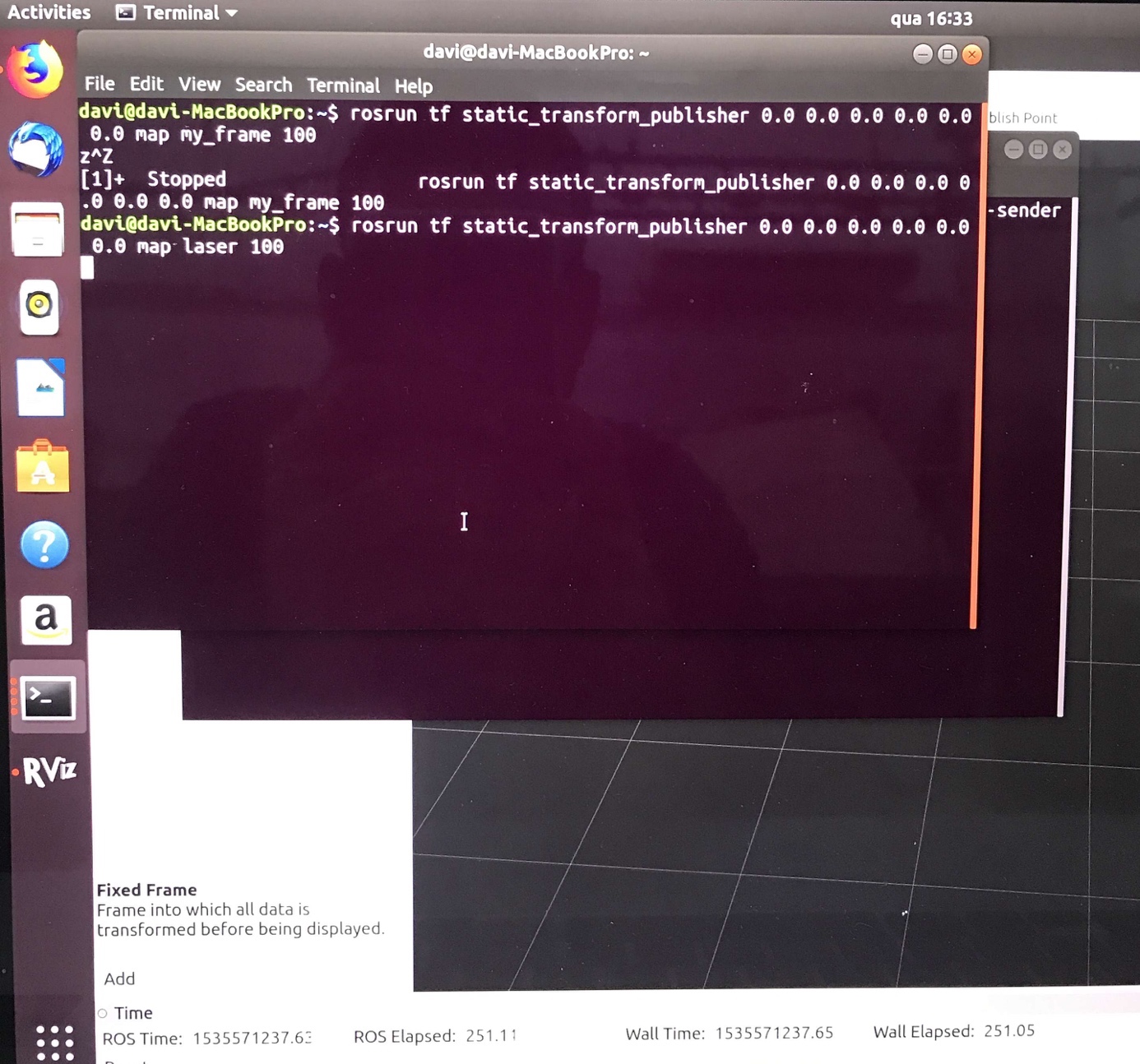
catkin\_make

## Erro ao testar o laser no rviz transform [sender=uknown\_puhlisher]



Abra um novo terminal e execute:

$ rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 0.0 map laser 100

O erro deverá desaparecer.

se algo n funcionar mais, refazer o workspace

## Problemas quanto aos frames do gmapping

Se voce executar em um terminal:

$ rosrun tf view\_frames

um arquivo como na imagem abaixo deverá ser gerado 

Se não estiver nesse formato, encerre todas as execuções com o “static\_transform\_publisher”, abra um novo terminal e execute:

$ rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 laser odom 100

Abra mais um terminal e execute:

$ rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 odom map 100

Pronto! Ao executar

$ rosrun tf view\_frames

A imagem acima deverá ser gerada.

## Comados interessantes ao usar o terminal:

para mostrar todas a funcoes associadas ao nome escrito no terminal, nome + tab (x3

) + y

Cancelar uma execução: control + c (x2)

top