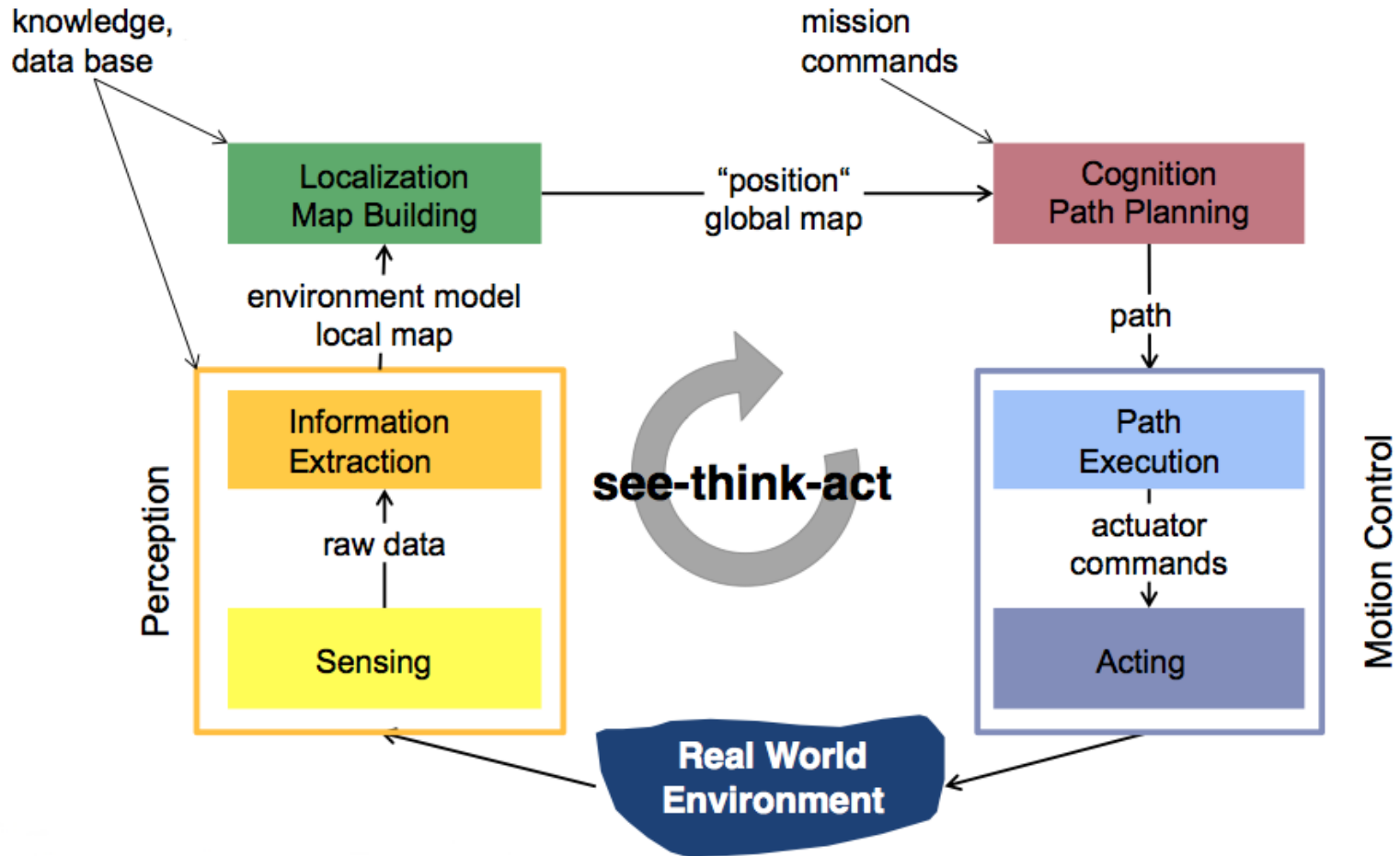


Navegação & Mapeamento

Prof. André Schneider de Oliveira
Prof. João Alberto Fabro

Universidade Tecnológica Federal do Paraná (UTFPR)

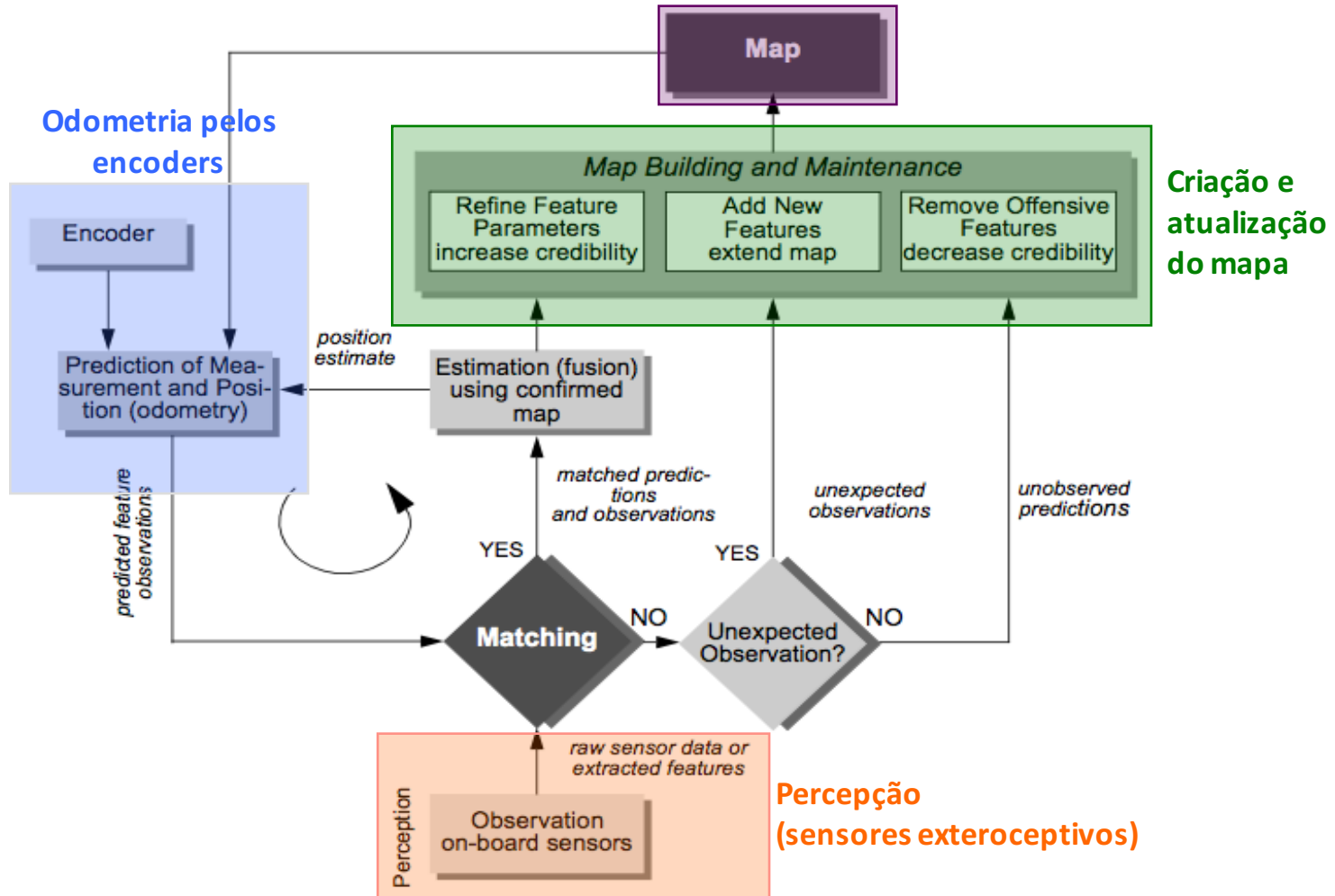
Controle de Robôs Móveis



Navegação & Mapeamento

- SLAM (Simultaneous localization and mapping)
 - Construção e atualização de um mapa de conhecimentos do ambiente durante a navegação
- O mapa é construído com informações da percepção do robô sobre o ambiente (sensores exteroceptivos)
- As informações do mapa permitem ao robô otimizar o planejamento das trajetórias

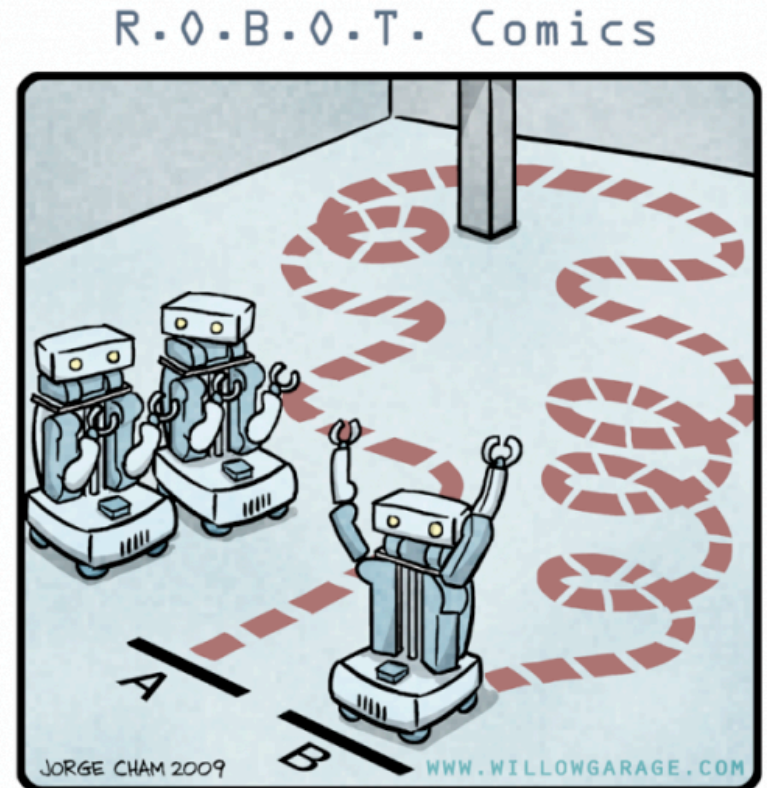
SLAM (Simultaneous localization and mapping)



Navigation Stack - ROS

- Utiliza informações da odometria, sensores e pose (posição e orientação) desejada para determinar as velocidades dos atuadores.
- É uma composição do:
 - Planejamento de trajetória
 - Controle de posição
 - SLAM

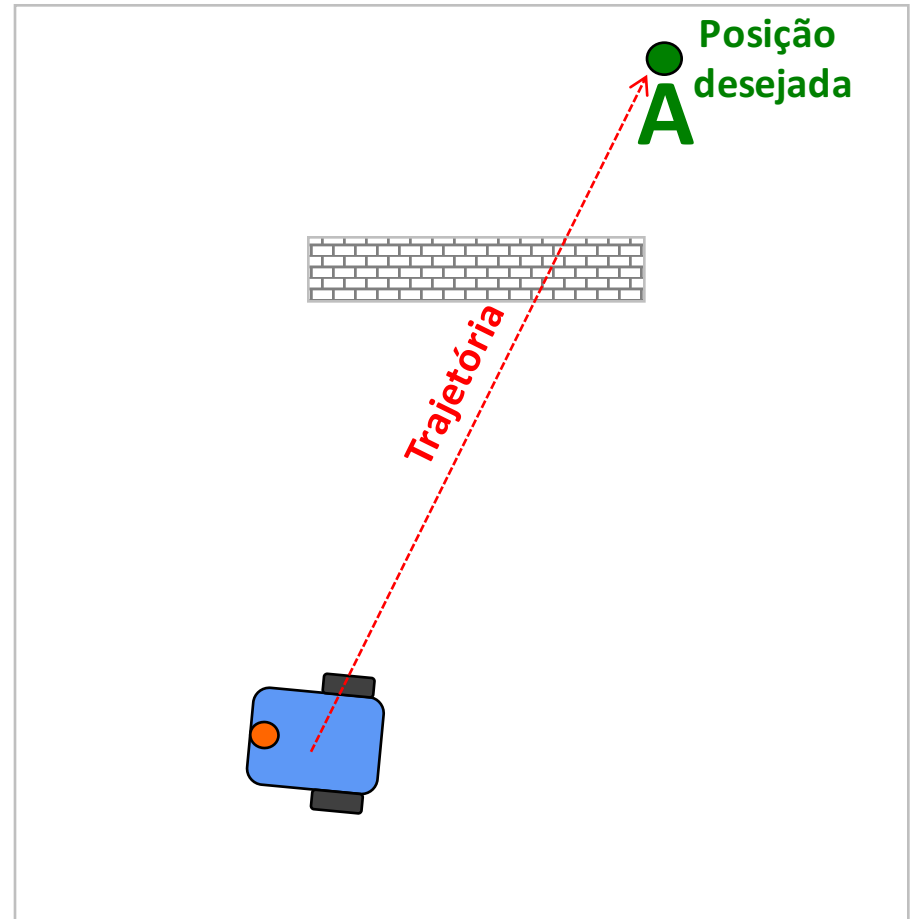
<http://wiki.ros.org/navigation>



"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

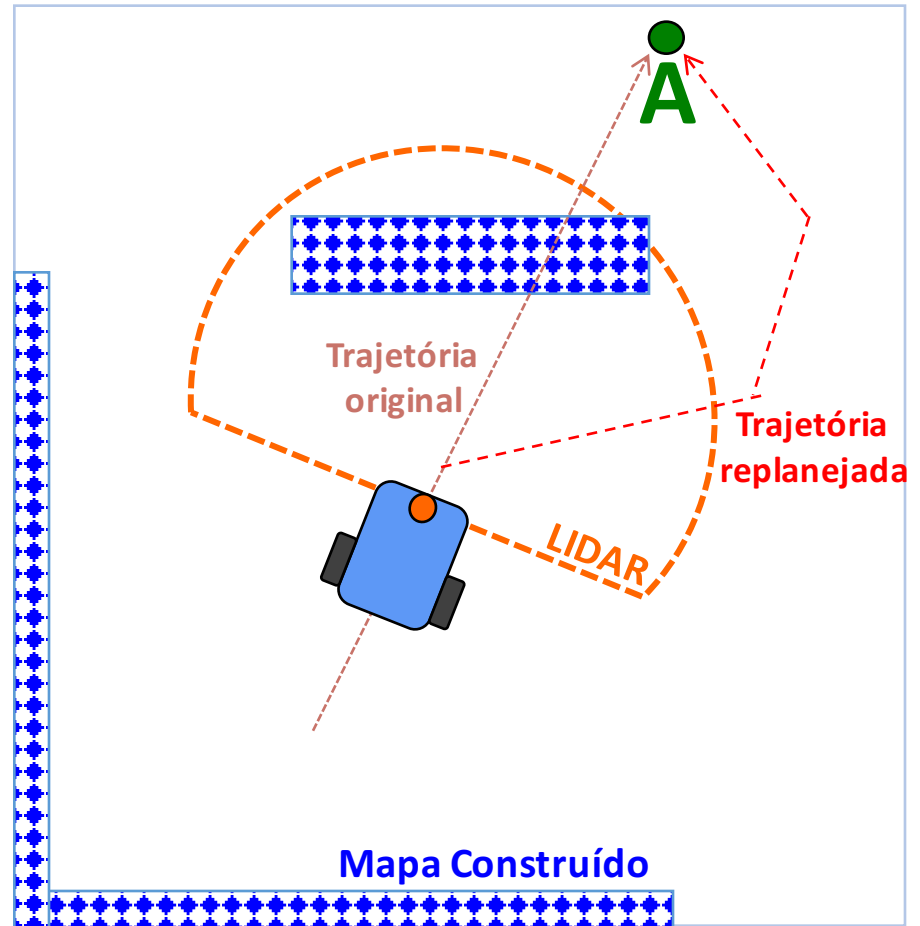
Funcionamento - Navigation Stack

- No início, o conhecimento do robô sobre o ambiente é nulo.
- É atribuída uma posição desejada (GOAL) no ponto A
- É planejada a trajetória da posição do atual até o GOAL.



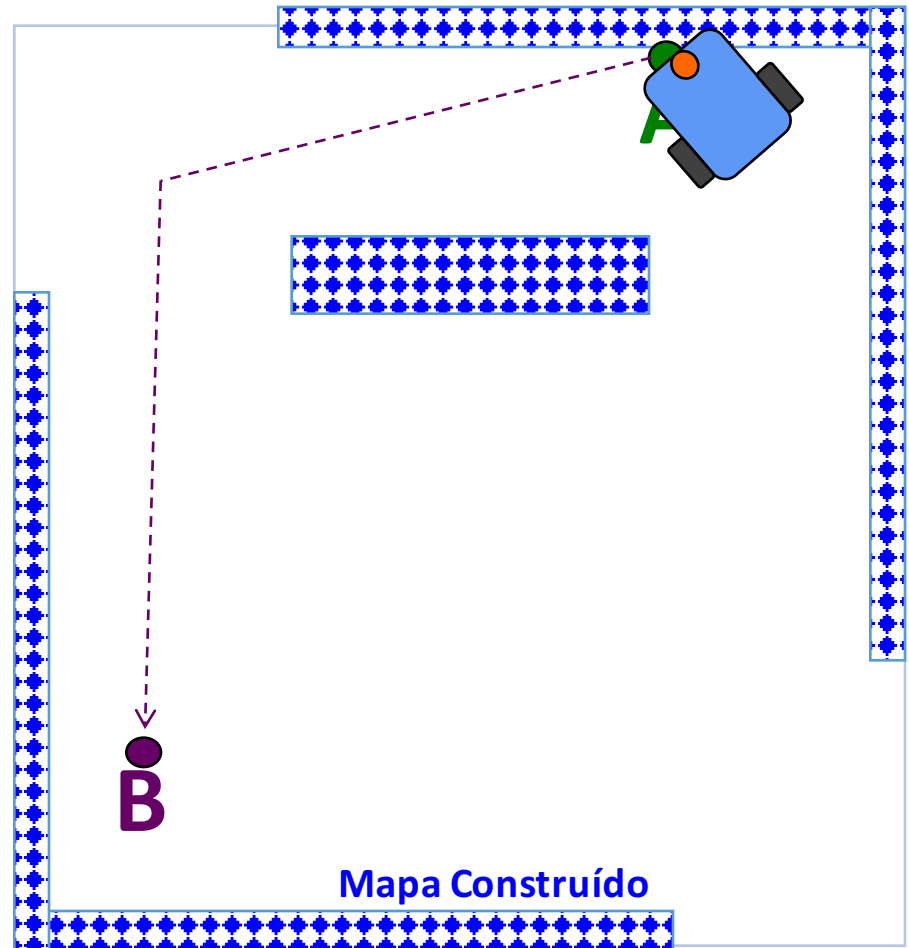
Funcionamento - Navigation Stack

- Durante o seguimento da trajetória o robô constrói o mapa do ambiente com informações dos sensores (percepção)
- Ao detectar um obstáculo na trajetória original a mesma é replanejada

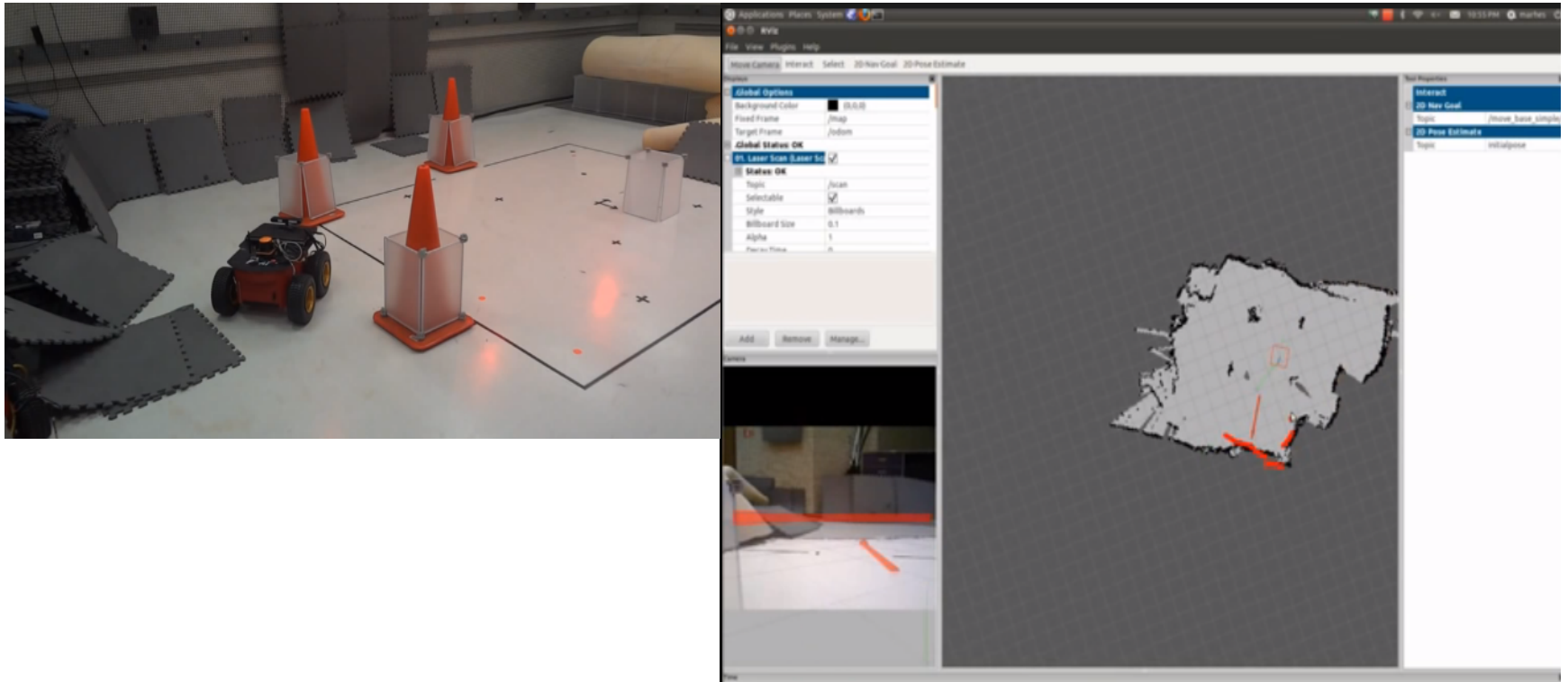


Funcionamento - Navigation Stack

- Ao atribuir o GOAL no ponto B o mapa permite o planejamento da trajetória otimizada (evitando o obstáculo)

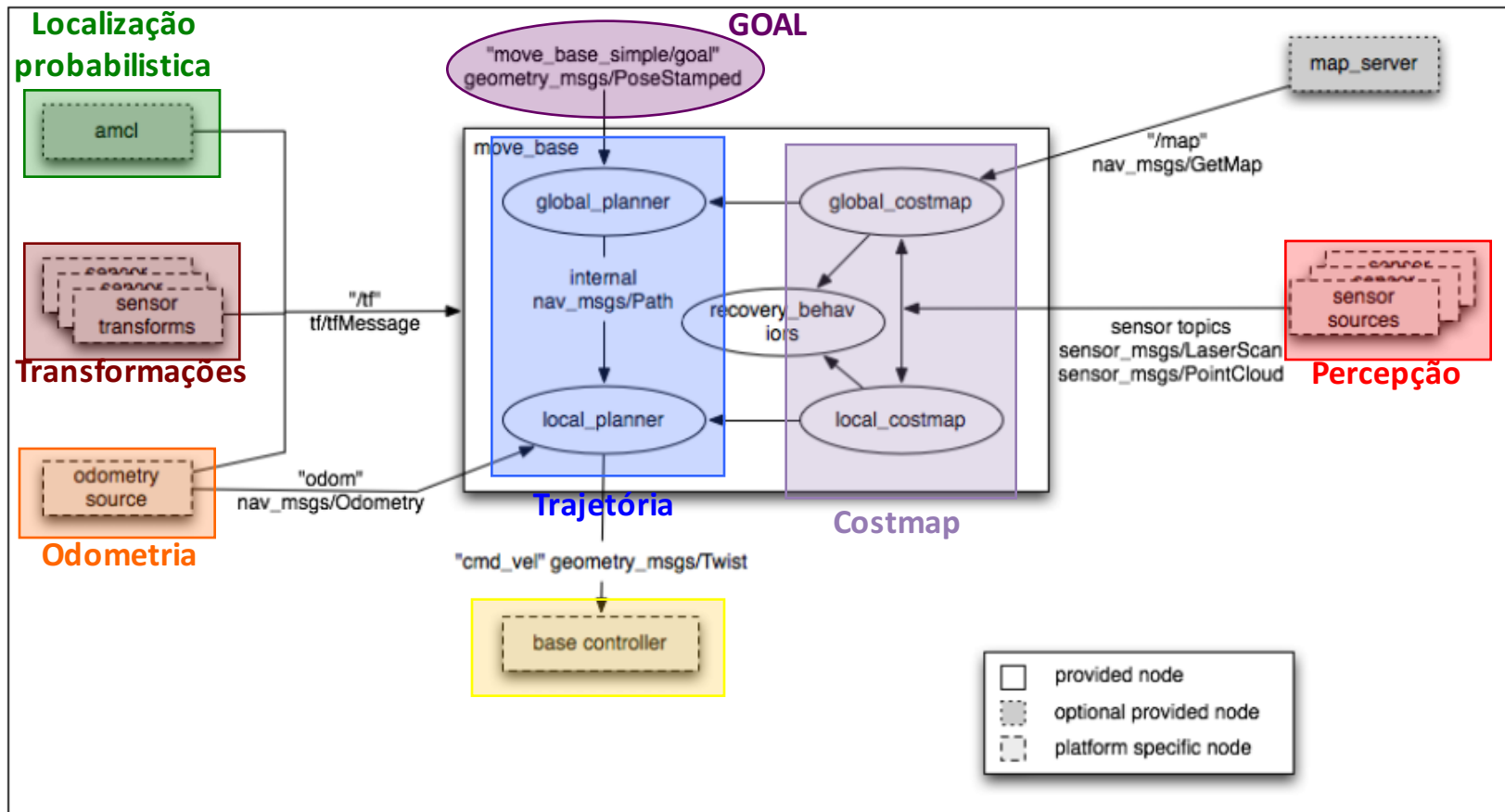


Exemplo - Navigation Stack



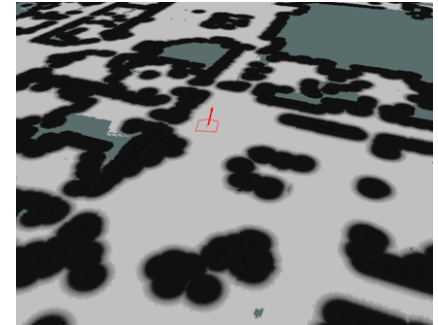
<https://www.youtube.com/watch?v=buWjU6wx8u4>

Navigation Stack

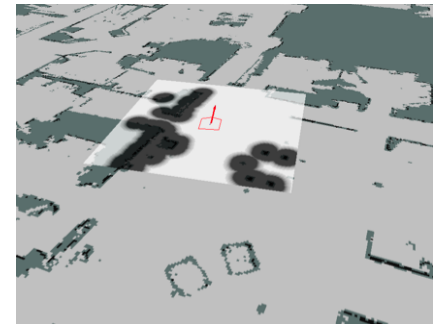


Costmap

- É a estrutura de dados que armazena os locais seguros para o robô em um “grid” de células
- O Costmap normalmente é binário e representa um espaço livre ou ocupado
- O robô pode movimentar-se pelo mapa através de duas formas:
 - A navegação global planeja trajetórias para um GOAL distante (global costmap)
 - A navegação local cria trajetetos para um GOAL próximo e evita obstáculos (local costmap)

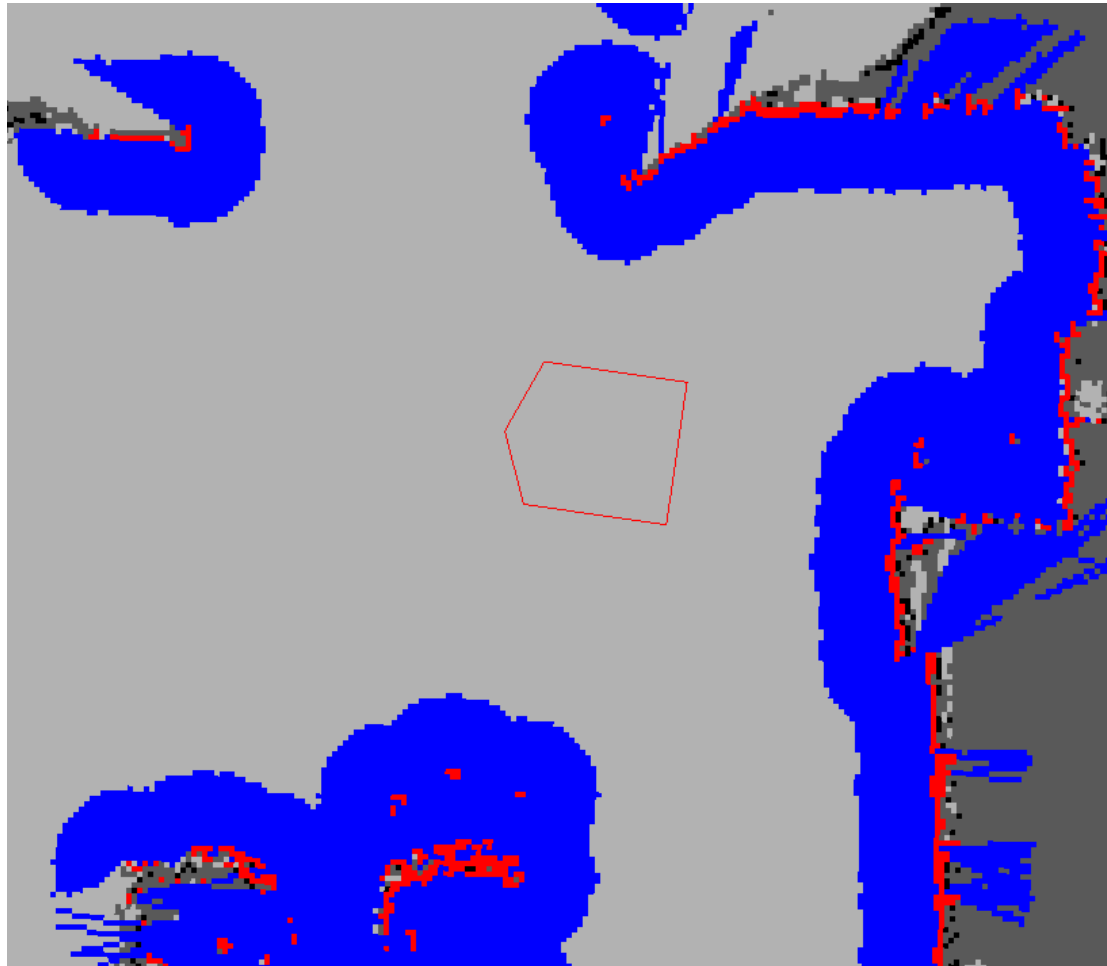


Global Costmap



Local Costmap

Exemplo Costmap



Configuração do Navigation Stack

1. `costmap_common_params.yaml` – **configurações gerais do Costmap**
2. `global_costmap_params.yaml` - **configurações do Costmap global**
3. `local_costmap_params.yaml` – **configurações do Costmap local**
4. `base_local_planner_params.yaml` - **parâmetros do planejamento da trajetória**
5. `move_base.launch` – **lançador do Navigation Stack**

Costmap

Configurações Padrões

costmap_common_params.yaml

```
obstacle_range: 2.5          # distância máxima em que o obstáculo será
                              # inserido no Costmap (m)

raytrace_range: 3.0          # distância máxima de observação dos sensores (m)

footprint: [[x0, y0], [x1, y1], ... [xn, yn]]
                              # formato 2D do robô (vista superior)

inflation_radius: 0.55       # raio (m) para determinação dos custos de
                              # navegação no mapa

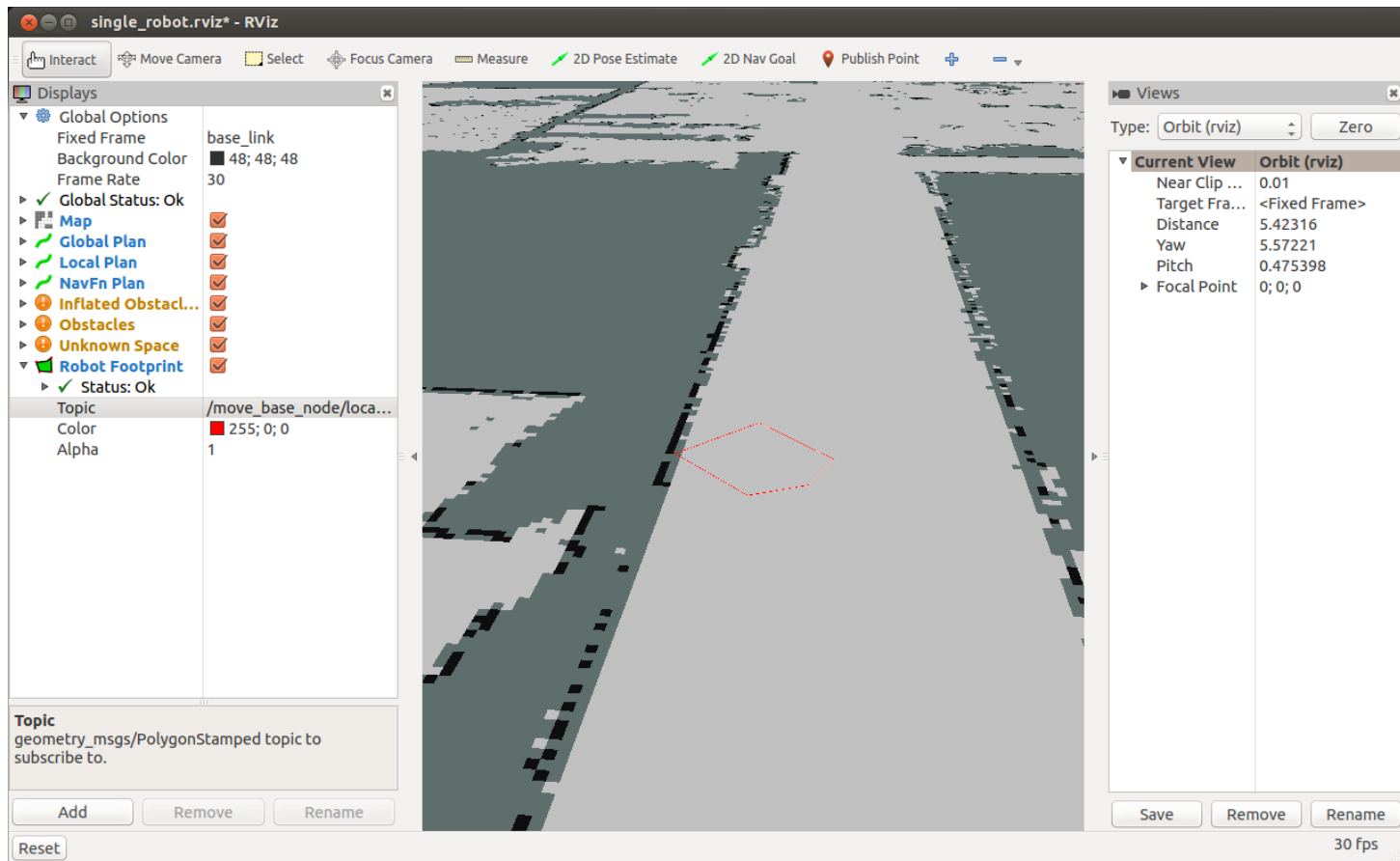
transform_tolerance: 0.3     # tolerância de atraso para as transformações

observation_sources: laser_scan_sensor point_cloud_sensor
                              #sensores de percepção

laser_scan_sensor: {sensor_frame: frame_name, data_type: LaserScan, topic: topic_name, marking: true,
clearing: true}
                              # exemplo da percepção por sensor LIDAR

point_cloud_sensor: {sensor_frame: frame_name, data_type: PointCloud, topic: topic_name, marking: true,
clearing: true}
                              # exemplo da percepção por sensor Kinect
```

Costmap – Footprint



Costmap

Configurações Globais

global_costmap_params.yaml

```
global_costmap:  
  # Configura as referências globais para o Costmap  
  global_frame: /map  
  robot_base_frame: base_link  
  
  # Configura a frequência de atualização e publicação do Costmap  
  update_frequency: 5.0  
  publish_frequency: 0.0  
  
  # Define que o mapa será criado  
  static_map: false
```


Configuração Base Local Planner

base_local_planner_params.yaml

```
TrajectoryPlannerROS:          # parâmetros do planejamento da trajetória
  max_vel_x: 0.45              # velocidade máxima (X)
  min_vel_x: 0.1               # velocidade mínima (X)
  max_rotational_vel: 1.0      # velocidade angular máxima
  min_in_place_rotational_vel: 0.4 # velocidade minima de orientação (Z)
  acc_lim_th: 3.2              # limite de aceleração angular (Z)
  acc_lim_x: 2.5               # limite de aceleração linear (X)
  acc_lim_y: 2.5               # limite de aceleração linear (X)
  holonomic_robot: false       # robô é holonômico?
```

Criando o pacote para o navigation

1. CRIAR O WORKSPACE ROS

http://wiki.ros.org/pt_BR/ROS/Tutorials/InstallingandConfiguringROSEnvironment

2. CRIAR UM PACOTE NO ROS

http://wiki.ros.org/pt_BR/ROS/Tutorials/catkin/CreatingPackage

Laucher para o Navigation

move_base.launch

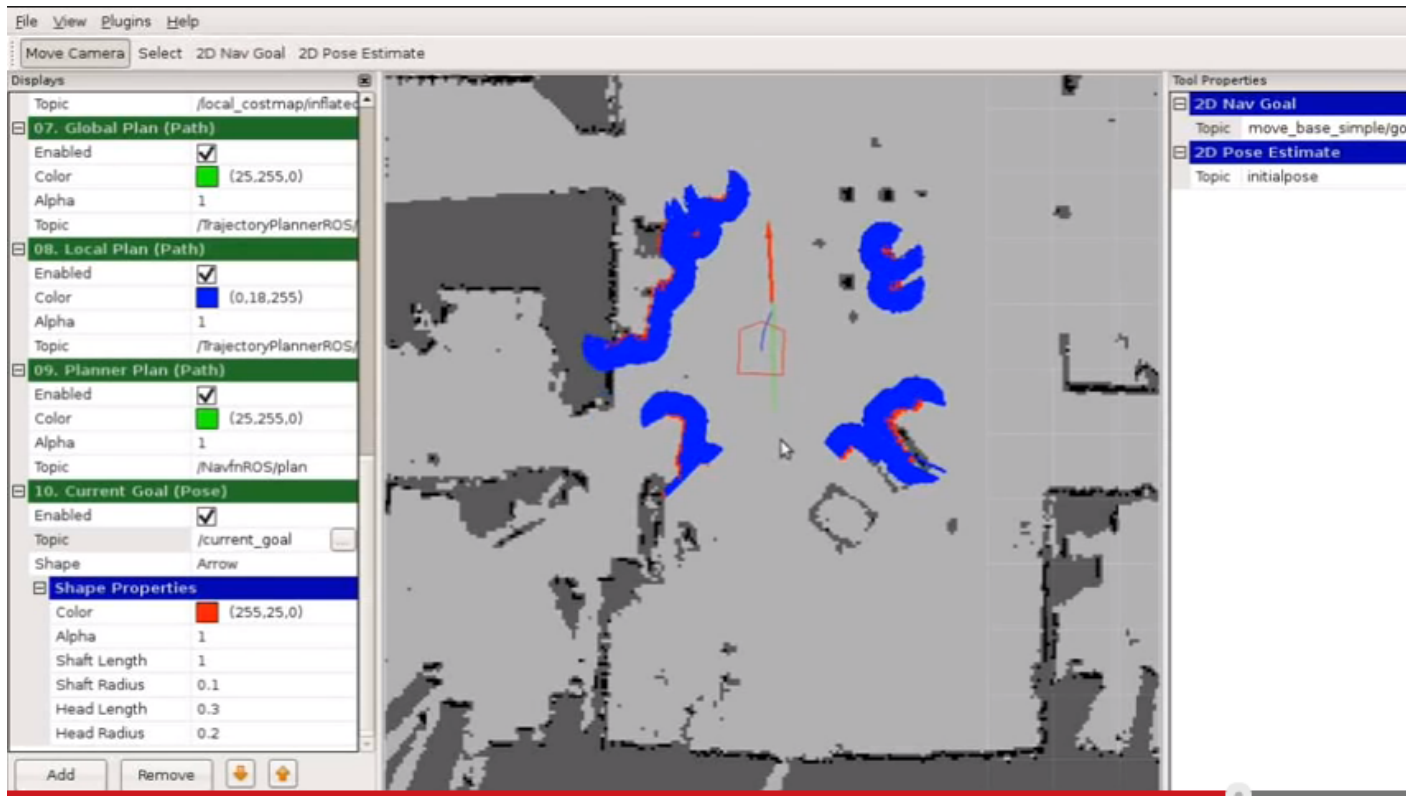
```
<launch>
  <master auto="start"/>

  <node pkg="odom" type="odom" respawn="false" name="odom" output="screen">
  </node>

  <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
  <rosparam file="$(find my_robot_name_2dnav)/costmap_common_params.yaml" command="load"
  ns="global_costmap" />
  <rosparam file="$(find my_robot_name_2dnav)/costmap_common_params.yaml" command="load"
  ns="local_costmap" />
  <rosparam file="$(find my_robot_name_2dnav)/local_costmap_params.yaml" command="load" />
  <rosparam file="$(find my_robot_name_2dnav)/global_costmap_params.yaml" command="load" />
  <rosparam file="$(find my_robot_name_2dnav)/base_local_planner_params.yaml" command="load" />
  </node>

</launch>
```

Configurando Rviz para o Navigation Stack



<https://www.youtube.com/watch?v=0CsSok3QgZk>

\$ rosrn rviz rviz

Principais problemas do Navigation Stack

- Falta de transformação dos sensores/motores até o robô
- Falta de transformação entre o robô e o ambiente

- Para visualizar as transformações (TF):

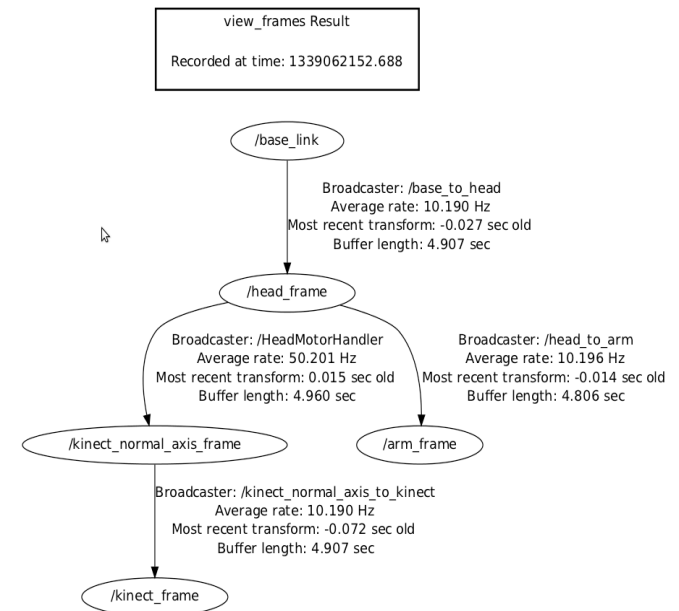
```
$ rosrun tf view_frames  
$ evince frames.pdf
```

- Diagnóstico das transformações:

```
$ roswtf
```

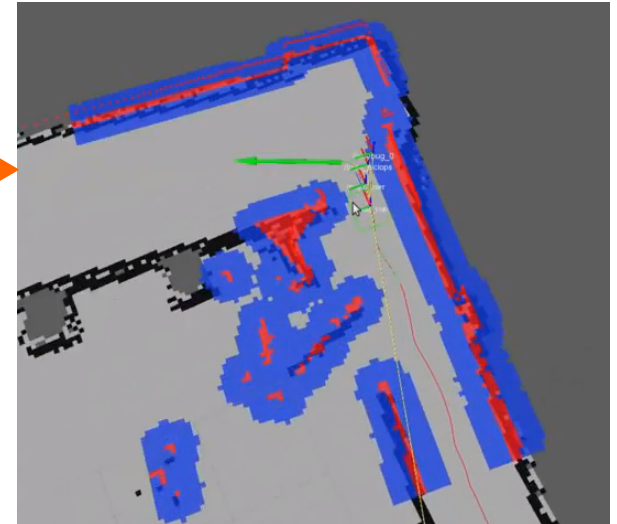
- Exemplo Publisher TF no V-Rep:

```
simExtROS_enablePublisher('tf',1,simros_strmcmd_get_transform,robotHandle,-1,") #TF Robô → World  
simExtROS_enablePublisher('tf',1,simros_strmcmd_get_transform,kinectHandle,robotHandle,") #TF Kinect → Robô
```



Tarefa – SLAM

1. Configurar a navegação e o mapeamento em ambientes desconhecidos através do Navigation Stack
2. Criar um ambiente com paredes e obstáculos no V-Rep
3. Visualizar a navegação e a percepção no Rviz



Referências

- SIEGWART, Roland; NOURBAKHS, Illah Reza; SCARAMUZZA, Davide. **Introduction to autonomous mobile robots**. MIT press, 2011, (**Capítulo 4 – Perception; Capítulo 5.8 Autonomous Map Building**)
- ROS Navigation Stack, **Setup and Configuration of the Navigation Stack on a Robot**

<http://wiki.ros.org/navigation/Tutorials/RobotSetup>

Referências Adicionais

Navigation Stack

- **[SLAM and Navigation with Pioneer 3-AT](https://sites.google.com/site/slamnavigation/)**

<https://sites.google.com/site/slamnavigation/>

- **[Dr. Rainer Hessmer - Using the ROS Navigation Stack](http://www.hessmer.org/blog/2011/04/24/using-the-ros-navigation-stack/)**

<http://www.hessmer.org/blog/2011/04/24/using-the-ros-navigation-stack/>

[http://www.hessmer.org/blog/2012/02/11/ardros-transform-between-base link-and-the-kinect-sensor/](http://www.hessmer.org/blog/2012/02/11/ardros-transform-between-base-link-and-the-kinect-sensor/)