Institut de Robòtica
i Informàtica Industrial

Robot-person accompaniment simulator tutorial: Capabilities

Ely Repiso, Anaís Garrell, Alberto Sanfeliu.

Horizon 2020
European Union funding
for Research & Innovation

AI4EU TERRINet
The European Robotics Research Infrastructure Network

This work has received partial founding from European Union's Horizon 2020 research and innovation program under the projects AI4EU (H2020-ICT-2018-2-825619) and TERRINet (H2020-INFRAIA-2017-1-two-stage-730994)

In addition to being able to accompany people in the environments and with the robots already provided, using the theoretical methods described in Repiso et al. IROS2017, and etc (All references are included in the document of Conceptual view). you can modify many parts of this software to easily adapt it to the needs of your robots and their environments, ans so on.

1. How to change the map

You can change the map of the environment used in the simulation or real system by other of our maps or your own maps.

1.1. Change the map in the Gazebo launch:

1.1.1. If the launch do not need change of robot and people coordinates:

Terminal$ roslaunch iri_ana_gazebo sim_sample_companion_with_person.launch **world_name:=fme_door_open**

Terminal$ roslaunch iri_ana_gazebo sim_sample_companion_with_person.launch **world_name:=fme_door_open_with_obst**

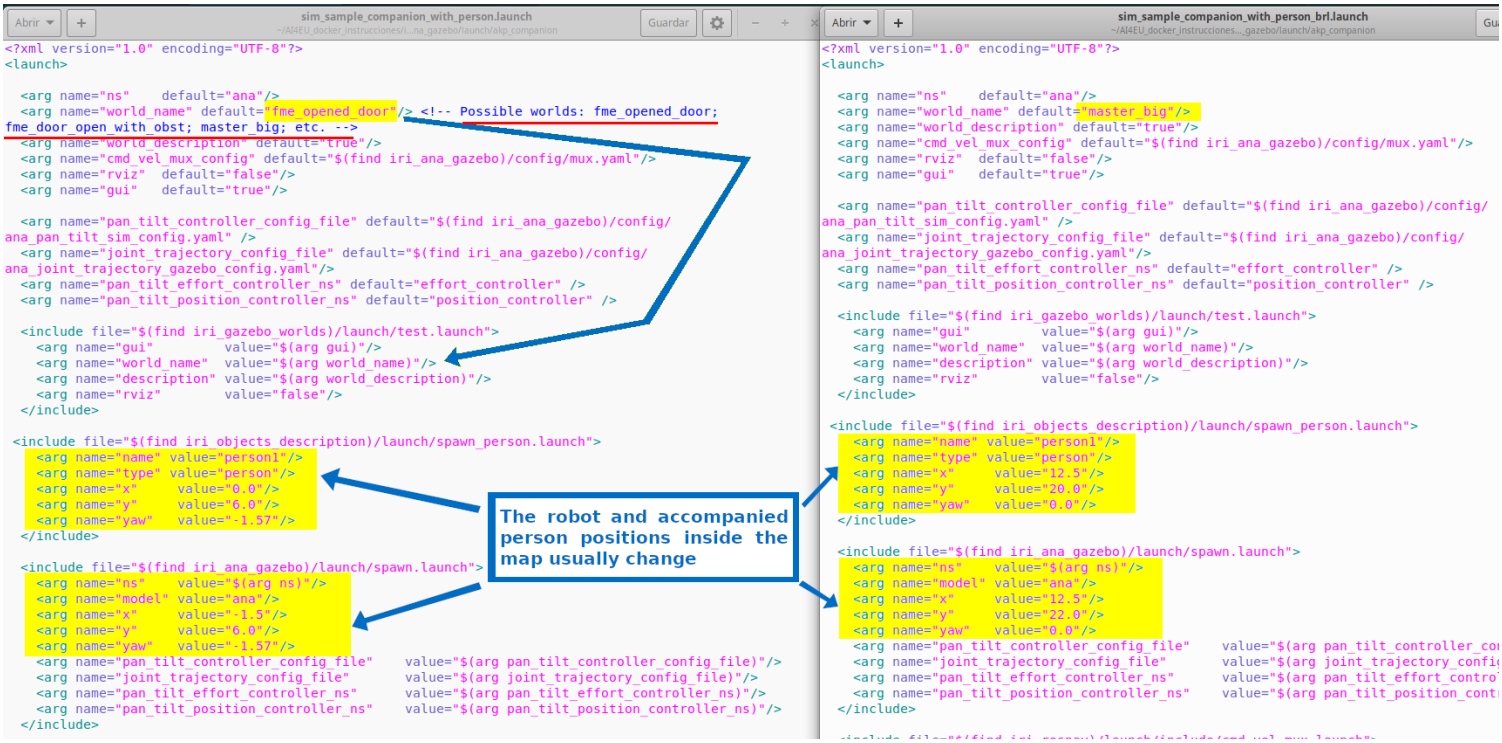
1.1.2. If the launch need change of robot and people coordinates:

Terminal$ roslaunch iri_ana_gazebo **sim_sample_companion_with_person.launch world_name:=fme_door_open**

Terminal$ roslaunch iri_ana_gazebo **sim_sample_companion_with_person_brl.launch world_name:=master_big**

Note: These launchs for Ana-robot are included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/robots/ana/iri_ana_gazebo/launch/akp_companion

Differences inside the launch files:



1.2. Change the map in the launch of the ros-nav, detection and tracking nodes:

1.2.1. If the launch do not need change of robot and people coordinates:

Terminal$ roslaunch iri_robot_assaop tracker_nav_and_detection_for_ASSAOP.launch **map_name:=fme_door_open**

Terminal$ oslaunch iri_robot_assaop tracker_nav_and_detection_for_ASSAOP.launch **map_name:=fme_door_open_with_obst**

1.2.2. If the launch need change of robot and people coordinates:

Terminal$ roslaunch iri_robot_assaop **tracker_nav_and_detection_for_ASSAOP.launch map_name:=fme_door_open**

Terminal$ roslaunch iri_robot_assaop **tracker_nav_and_detection_for_ASSAOP_brl.launch map_name:=master_big**

Note: These launchs for Ana-robot are included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/iri_navigation/iri_robot_assaop/launch/

Differences inside the launch files:

tracker_nav_and_detection_for_ASSAOP.launch

```
<!-- -->
<launch>

  <arg name="rviz"                  default="true"/>
  <arg name="rviz_file"             default="$(find iri_ana_rosnav)/rviz/ana.rviz"/>
  <arg name="move_base_params"      default="move_base_params.yaml"/>
  <arg name="costmap_common_params" default="common_params.yaml"/>
  <arg name="costmap_local_params"  default="local_params.yaml"/>
  <arg name="costmap_global_params" default="global_params.yaml"/>
  <arg name="map_name"              default="fme"/>
  <arg name="initial_x"             default="-1.5"/>
  <arg name="initial_y"             default="6.0"/>
  <arg name="initial_yaw"           default="-1.57"/>
  <arg name="local_planner"         default="dwa"/>
  <arg name="global_planner"        default="global_planner"/>
  <arg name="output"                default="screen" />
  <arg name="launch_prefix"         default="" />

  <include file="$(find iri_rosnav)/launch/nav.launch">
    <arg name="ns"                  value="ana"/>
    <arg name="path"                value="$(find iri_ana_rosnav)/params"/>
    <arg name="move_base_params"    value="$(arg move_base_params)"/>
    <arg name="costmap_common_params" value="$(arg costmap_common_params)"/>
    <arg name="costmap_local_params"  value="$(arg costmap_local_params)"/>
    <arg name="costmap_global_params" value="$(arg costmap_global_params)"/>
    <arg name="map_frame_id"        value="map"/>
    <arg name="odom_frame_id"       value="ana/odom"/>
    <arg name="base_frame_id"       value="ana/base_footprint"/>
    <arg name="map_topic"           value="/ana/map"/>
    <arg name="map_service"         value="/ana/static_map"/>
    <arg name="odom_topic"          value="/ana/odom"/>
    <arg name="cmd_vel_topic"       value="/ana/navigation/cmd_vel"/>
    <arg name="scan_topic"          value="/ana/sensors/scan"/>
    <arg name="use_map"             value="true"/>
    <arg name="use_map_server"      value="true"/>
    <arg name="map_name"            value="$(arg map_name)"/>
    <arg name="use_amcl"            value="true"/>
    <arg name="amcl_config"         value="$(find iri_ana_rosnav)/params/
amcl_companion_test_ely2021_mapFME.yaml"/>
    <arg name="initial_x"           value="$(arg initial_x)"/>
    <arg name="initial_y"           value="$(arg initial_y)"/>
    <arg name="initial_yaw"         value="$(arg initial_yaw)"/>
    <arg name="use_fake_loc"        value="false"/>
    <arg name="use_gmapping"        value="false"/>
    <arg name="local_planner"       value="$(arg local_planner)"/>
    <arg name="global_planner"      value="$(arg global_planner)"/>
    <arg name="output"              value="$(arg output)" />
    <arg name="launch_prefix"       value="$(arg launch_prefix)" />
  </include>

  <include file="$(find iri_rosnav)/launch/include/pointcloud_to_laserscan.launch">
    <arg name="ns"            value="ana"/>
    <arg name="node_name"     value="pcl_to_laserscan"/>
    <arg name="scan_topic"    value="/ana/sensors/scan"/>
    <arg name="cloud_topic"   value="/ana/sensors/velodyne_points"/>
```

tracker_nav_and_detection_for_ASSAOP_brl.launch

```
<!-- -->
<launch>

  <arg name="rviz"                  default="true"/>
  <arg name="rviz_file"             default="$(find iri_ana_rosnav)/rviz/ana.rviz"/>
  <arg name="move_base_params"      default="move_base_params.yaml"/>
  <arg name="costmap_common_params" default="common_params.yaml"/>
  <arg name="costmap_local_params"  default="local_params.yaml"/>
  <arg name="costmap_global_params" default="global_params.yaml"/>
  <arg name="map_name"              default="master_big"/>
  <arg name="initial_x"             default="20.5"/>
  <arg name="initial_y"             default="39.0"/>
  <arg name="initial_yaw"           default="0.0"/>
  <arg name="local_planner"         default="dwa"/>
  <arg name="global_planner"        default="global_planner"/>
  <arg name="output"                default="screen" />
  <arg name="launch_prefix"         default="" />

  <include file="$(find iri_rosnav)/launch/nav.launch">
    <arg name="ns"                  value="ana"/>
    <arg name="path"                value="$(find iri_ana_rosnav)/params"/>
    <arg name="move_base_params"    value="$(arg move_base_params)"/>
    <arg name="costmap_common_params" value="$(arg costmap_common_params)"/>
    <arg name="costmap_local_params"  value="$(arg costmap_local_params)"/>
    <arg name="costmap_global_params" value="$(arg costmap_global_params)"/>
    <arg name="map_frame_id"        value="map"/>
    <arg name="odom_frame_id"       value="ana/odom"/>
    <arg name="base_frame_id"       value="ana/base_footprint"/>
    <arg name="map_topic"           value="/ana/map"/>
    <arg name="map_service"         value="/ana/static_map"/>
    <arg name="odom_topic"          value="/ana/odom"/>
    <arg name="cmd_vel_topic"       value="/ana/navigation/cmd_vel"/>
    <arg name="scan_topic"          value="/ana/sensors/scan"/>
    <arg name="use_map"             value="true"/>
    <arg name="use_map_server"      value="true"/>
    <arg name="map_name"            value="$(arg map_name)"/>
    <arg name="use_amcl"            value="true"/>
    <arg name="amcl_config"         value="$(find iri_ana_rosnav)/params/
amcl_companion_test_ely2021_mapFME.yaml"/>
    <arg name="initial_x"           value="$(arg initial_x)"/>
    <arg name="initial_y"           value="$(arg initial_y)"/>
    <arg name="initial_yaw"         value="$(arg initial_yaw)"/>
    <arg name="use_fake_loc"        value="false"/>
    <arg name="use_gmapping"        value="false"/>
    <arg name="local_planner"       value="$(arg local_planner)"/>
    <arg name="global_planner"      value="$(arg global_planner)"/>
    <arg name="output"              value="$(arg output)" />
    <arg name="launch_prefix"       value="$(arg launch_prefix)" />
  </include>

  <include file="$(find iri_rosnav)/launch/include/pointcloud_to_laserscan.launch">
    <arg name="ns"            value="ana"/>
    <arg name="node_name"     value="pcl_to_laserscan"/>
    <arg name="scan_topic"    value="/ana/sensors/scan"/>
    <arg name="cloud_topic"   value="/ana/sensors/velodyne_points"/>
```

2.3. Change the map for the one person accompaniment launch:

Terminal$ roslaunch iri_robot_assaop gazebo_ASSAOP_OK_ana.launch

Terminal$ roslaunch iri_robot_assaop gazebo_ASSAOP_OK_ana_brl.launch

Note: These launchs for Ana-robot are included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/iri_navigation/iri_robot_assaop/launch/

Differences inside the launch files:

gazebo_ASSAOP_OK_ana.launch

```
<!-- -->
<launch>


   <!-- Ini Navigation Companion Nodes -->
  <!--launch-prefix="xterm -e ddd -args"  -->

 <group ns="$(optenv ROBOT ana)">
 <rosparam command="delete" ns="/ana/move_base" />

   <!--launch-prefix="xterm -e ddd -args"  machine="visio"-->
   <node pkg ="move_base"
         type="move_base"
         name="move_base"
         output="screen">
     <remap from="/ana/move_base_ana/cmd_vel"
            to="/ana/navigation/cmd_vel" />
     <remap from="/ana/move_base_ana/odom"
            to="/ana/odom" />
     <remap from="/ana/front_scan"
            to="/ana/sensors/scan" />
     <remap from="/ana/rear_scan"
            to="/ana/sensors/scan" />
     <remap from="/ana/tracks"
             to="/ana/mht/tracks" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/move_base_params_assaop.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/ana/costmap_common_params.yaml"
command="load" ns="global_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/costmap_common_params.yaml" command="load"
ns="local_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/local_costmap_params.yaml"  command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/ana/global_costmap_params.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/akp_local_planner_params.yaml"
command="load" />
<rosparam file="$(find iri_ana_rosnav)/params/amcl_companion_test_ely2021_mapFME.yaml" command="load" />
 <!--<rosparam file="$(find iri_dabo_rosnav)/params/global_planner/global_planner_params.yaml"
command="load" />
 <rosparam file="$(find iri_dabo_rosnav)/params/mux.yaml" command="load" />-->
     <param name="AkpLocalPlanner/destination_map_path" value="$(find iri_robot_assaop)/maps/
destinaciones_ana_FME_ubuntu18_ros_melodic.txt" />
       <param name="AkpLocalPlanner/results_filename_Zanlungo" value="name_file.txt" />
     <param name="AkpLocalPlanner/robot" value="ana"/>
     <param name="AkpLocalPlanner/person_goal_id"  value="200"/>
     <param name="AkpLocalPlanner/speed_k"  value="1"/>
     <param name="AkpLocalPlanner/person_radi_amp"  value="0.2"/>
     <param name="AkpLocalPlanner/change_treshold_distance_between_steps"  value="0.3"/>
       <!-- for SIM-->
       <param name="AkpLocalPlanner/change_sim"  value="False"/>
       <param name="AkpLocalPlanner/sim_target_per"  value="False"/>
       <param name="AkpLocalPlanner/frame_robot_footprint"  value="ana/base_link"/>
   </node>
 </group>
```

gazebo_ASSAOP_OK_ana_brl.launch

```
<!-- -->
<launch>


   <!-- Ini Navigation Companion Nodes -->
  <!--launch-prefix="xterm -e ddd -args"  -->

 <group ns="$(optenv ROBOT ana)">
 <rosparam command="delete" ns="/ana/move_base" />

   <!--launch-prefix="xterm -e ddd -args"  machine="visio"-->
   <node pkg ="move_base"
         type="move_base"
         name="move_base"
         output="screen">
     <remap from="/ana/move_base_ana/cmd_vel"
            to="/ana/navigation/cmd_vel" />
     <remap from="/ana/move_base_ana/odom"
            to="/ana/odom" />
     <remap from="/ana/front_scan"
            to="/ana/sensors/scan" />
     <remap from="/ana/rear_scan"
            to="/ana/sensors/scan" />
     <remap from="/ana/tracks"
             to="/ana/mht/tracks" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/move_base_params_assaop.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/ana/costmap_common_params.yaml" command="load"
ns="global_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/costmap_common_params.yaml" command="load"
ns="local_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/local_costmap_params.yaml"  command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/ana/global_costmap_params.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/akp_local_planner_params.yaml" command="load" /
>
<rosparam file="$(find iri_ana_rosnav)/params/amcl_companion_test_ely2021_mapFME.yaml" command="load" />
 <!--<rosparam file="$(find iri_dabo_rosnav)/params/global_planner/global_planner_params.yaml"
command="load" />
 <rosparam file="$(find iri_dabo_rosnav)/params/mux.yaml" command="load" />-->
     <param name="AkpLocalPlanner/destination_map_path" value="$(find iri_robot_assaop)/maps/
master_big_destinations_Gazebo_sim.txt" />
       <param name="AkpLocalPlanner/results_filename_Zanlungo" value="name_file.txt" />
     <param name="AkpLocalPlanner/robot" value="ana"/>
     <param name="AkpLocalPlanner/person_goal_id"  value="200"/>
     <param name="AkpLocalPlanner/speed_k"  value="1"/>
     <param name="AkpLocalPlanner/person_radi_amp"  value="0.2"/>
     <param name="AkpLocalPlanner/change_treshold_distance_between_steps"  value="0.3"/>
       <!-- for SIM-->
       <param name="AkpLocalPlanner/change_sim"  value="False"/>
       <param name="AkpLocalPlanner/sim_target_per"  value="False"/>
       <param name="AkpLocalPlanner/frame_robot_footprint"  value="ana/base_link"/>
   </node>
 </group>
```

Note: In this launch the only change is the txt that includes the static-destinations of the environment, which usually change from one environment to other. These destinations are entrances, exits, work-places, etc (destinations where people usually go inside any environment. How to change these destinations is explained in sec 5 of the current document.

2. Change the robot

You can change the robot used in the simulation or real system by any of our robots or your own robot.

2.1. Change the robot in the Gazebo launch:

Terminal$ roslaunch iri_ana_gazebo sim_sample_companion_with_person_brl.launch world_name:=master_big

Terminal$ roslaunch iri_dabo_gazebo sim_gazebo_dabo_companion.launch

Note: The launch for Ana-robot is included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/robots/ana/iri_ana_gazebo/launch/akp_companion/

And the launch for Dabo-robot is included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/robots/dabo/iri_dabo_gazebo/launch/

Also, both launchs use the same environment, to only see the changes due to change the robot.

Differences inside the launch files:

sim_sample_companion_with_person_brl.launch

```
<?xml version="1.0" encoding="UTF-8"?>
<launch>

 <arg name="ns"    default="ana"/>
 <arg name="world_name" default="master_big"/>
 <arg name="world_description" default="true"/>
 <arg name="cmd_vel_mux_config" default="$(find iri_ana_gazebo)/config/mux.yaml"/>
 <arg name="rviz"  default="false"/>
 <arg name="gui"   default="true"/>
```

Some controllers for the Ana-robot (different for Dabo):

```
 <arg name="pan_tilt_controller_config_file" default="$(find iri_ana_gazebo)/config/
ana_pan_tilt_sim_config.yaml" />
 <arg name="joint_trajectory_config_file" default="$(find iri_ana_gazebo)/config/
ana_joint_trajectory_gazebo_config.yaml"/>
 <arg name="pan_tilt_effort_controller_ns" default="effort_controller" />
 <arg name="pan_tilt_position_controller_ns" default="position_controller" />

 <include file="$(find iri_gazebo_worlds)/launch/test.launch">
   <arg name="gui"         value="$(arg gui)"/>
   <arg name="world_name"  value="$(arg world_name)"/>
   <arg name="description" value="$(arg world_description)"/>
   <arg name="rviz"        value="false"/>
 </include>

<include file="$(find iri_objects_description)/launch/spawn_person.launch">
   <arg name="name" value="person1"/>
   <arg name="type" value="person"/>
   <arg name="x"     value="12.5"/>
   <arg name="y"     value="20.0"/>
   <arg name="yaw"   value="0.0"/>
 </include>
```

To launch the robot model of Ana-robot in gazebo:

```
<include file="$(find iri_ana_gazebo)/launch/spawn.launch">
   <arg name="ns"    value="$(arg ns)"/>
   <arg name="model" value="ana"/>
   <arg name="x"     value="12.5"/>
   <arg name="y"     value="22.0"/>
   <arg name="yaw"   value="0.0"/>
   <arg name="pan_tilt_controller_config_file"    value="$(arg pan_tilt_controller_config_file)"/>
   <arg name="joint_trajectory_config_file"       value="$(arg joint_trajectory_config_file)"/>
   <arg name="pan_tilt_effort_controller_ns"      value="$(arg pan_tilt_effort_controller_ns)"/>
   <arg name="pan_tilt_position_controller_ns"    value="$(arg pan_tilt_position_controller_ns)"/>
 </include>
```

All nodes need the robot to be used to be specified as sensors and other things may be different:

```
<include file="$(find iri_rosnav)/launch/include/cmd_vel_mux.launch">
   <arg name="ns"                value="$(arg ns)"/>
   <arg name="nodelet_manager_name"  value="mux_nodelet_manager"/>
   <arg name="config"                value="$(arg cmd_vel_mux_config)"/>
   <arg name="output"                value="log" />
   <arg name="launch_prefix"         value="" />
 </include>
```

sim_gazebo_dabo_companion.launch

```
<?xml version="1.0" encoding="UTF-8"?>
<launch>

 <arg name="ns"    default="dabo"/>
  <arg name="world_name" default="master_big"/> <!-- other world_name:empty, master_big, fme_opened_door (in:
iri_navigation/iri_gazebo_worlds/models/fme_opened_door)-->
  <arg name="world"      default="$(find iri_gazebo_worlds)/worlds/$(arg world_name).world"/>
  <arg name="world_description" default="true"/>
  <arg name="robot_model" default="dabo_fixed"/>
  <arg name="rviz"  default="true"/>
  <arg name="gui"   default="true"/>
<!-- Ini variables nav-->
  <arg name="move_base_params"        default="move_base_params.yaml"/>
  <arg name="costmap_common_params"   default="common_params.yaml"/>
  <arg name="costmap_local_params"    default="local_params.yaml"/>
  <arg name="costmap_global_params"   default="global_params.yaml"/>
  <arg name="map_name"                default="master_big"/>
  <arg name="initial_x"               default="11.0"/>
  <arg name="initial_y"               default="22.0"/>
  <arg name="initial_yaw"             default="0.0"/>
  <arg name="local_planner"           default="dwa"/>
  <arg name="global_planner"          default="global_planner"/>
  <arg name="output"                  default="screen" />
  <arg name="launch_prefix"           default="" />

  <include file="$(find iri_gazebo_worlds)/launch/test.launch">
    <arg name="gui"         value="$(arg gui)"/>
    <arg name="world_name"  value="$(arg world_name)"/>
    <arg name="description" value="$(arg world_description)"/>
    <arg name="rviz"        value="false"/>
  </include>
```

To launch the robot model of Dabo-robot in gazebo:

```
  <include file="$(find iri_dabo_gazebo)/launch/spawn.launch">
    <arg name="ns"    value="$(arg ns)"/>
    <arg name="model" value="$(arg robot_model)"/>
    <arg name="x"     value="11.0"/>
    <arg name="y"     value="22.0"/>
    <arg name="yaw"   value="0.0"/>
  </include>

<!-- person1 is the nearest person to the robot. This person will be the accompanied one, if you use exactly
our instructions included in the read-me of the node. -->
  <include file="$(find iri_objects_description)/launch/spawn_person.launch">
    <arg name="name" value="person1"/>
    <arg name="type" value="person"/>
    <arg name="x"     value="11.5"/>  <!-- changes the x component of the initial person1's position inside
the environment (it is in map coordinates).  -->
    <arg name="y"     value="20.0"/> <!-- changes the y component of the initial person1's position inside th
environment (it is in map coordinates)-->
    <arg name="yaw"   value="0.0"/> <!-- changes the initial person1's orientation inside the environment (it
is in map coordinates)-->
  </include>
```

Note: We show the most important changes, but you may enter in both files and compare the differences with these two, to be able to use other robots.

2.2. Change the robot in the launch for the nodes of ros-nav, detection and tracking:

Terminal$ roslaunch iri_robot_assaop tracker_nav_and_detection_for_ASSAOP_brl.launch map_name:=master_big

Terminal$ launch iri_dabo_gazebo sim_gazebo_dabo_companion.launch **// in this case these nodes are launch with gazebo.**

Note: The launch for Ana-robot is included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/iri_navigation/iri_robot_assaop/launch/

And the launch for Dabo-robot is included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/robots/dabo/iri_dabo_gazebo/launch/

Also, both launchs use the same environment, to only see the changes due to change the robot.

Differences inside the launch files:

tracker_nav_and_detection_for_ASSAOP_brl.launch

```
<group ns="$(optenv ROBOT ana)">
  <!-- People Tracking MHT (remap) -->
    <!-- published topics:  /$(env ROBOT)/mht/tracks -->
    <!-- subscribed topics: /$(env ROBOT)/detections -->
    <!-- service clients: -->
    <!-- service servers: -->
    <!-- action clients: -->
    <!-- action servers: -->
  <node pkg ="iri_people_tracking_mht"
        type="iri_people_tracking_mht"
        name="mht"
        output="screen">
    <remap from="/ana/mht/detections"
           to="/ana/lpd/peopleMapFiltered" />
    <remap from="/ana/mht/odom"
           to="/ana/odom" /> <!-- /topic-odometry-of-the-robot -->
    <remap from="/ana/mht/odom_for_medium_velocity"
           to="/ana/odom" /> <!-- /topic-odometry-of-the-robot -->

  <!-- People Tracking MHT parameters for outdoor environments with few false alarms-->
        <param name="~/frame_id"  type="string" value="map" /> <!-- tf of the map for the tracker -->
    <param name="local_tracker"             type="bool"   value="false"/>
    <param name="~/association_threshold"   type="double" value="1.0" />
    <param name="~/confirmation_threshold"  type="double" value="0.9" />
    <param name="~/deletion_threshold"      type="double" value="0.4" />
    <param name="~/laser_probability"       type="double" value="0.9" />
    <param name="~/laser_false_alarm"       type="double" value="0.1" />
    <param name="~/laser_new_track"         type="double" value="0.11" />
    <param name="~/laser_no_detection"      type="double" value="0.97" />
    <param name="~/laser_no_confirmed"      type="double" value="0.02" />
   <param name="~/laser_no_confirmed_iteration" type="double" value="0.01" />
   <param name="~/augment_covariance_track" type="double" value="0.5" />
    <param name="~/mht_velocity_margin" type="double" value="1.0" />
    <param name="~/covariance_no_track" type="double" value="5.0" />
  </node>
```

We use the group ns in Ana-robot, but not in dabo. Then in Dabo are less remaps to change.

sim_gazebo_dabo_companion.launch

```
<!-- DETECTION AND TRACKING -->
<!-- DETECTION AND TRACKING -->
<!-- DETECTION AND TRACKING -->

 <!-- People Tracking MHT (remap) -->
    <!-- published topics:  /$(env ROBOT)/mht/tracks -->
    <!-- subscribed topics: /$(env ROBOT)/detections -->
    <!-- service clients: -->
    <!-- service servers: -->
    <!-- action clients: -->
    <!-- action servers: -->
  <node pkg ="iri_people_tracking_mht"
        type="iri_people_tracking_mht"
        name="mht_2"
        output="screen">
    <remap from="/mht_2/detections"
           to="/lpd_2/peopleMapFiltered" />
    <remap from="/mht_2/odom"
           to="/dabo/odom" /> <!-- /topic-odometry-of-the-robot -->
    <remap from="/mht_2/odom_for_medium_velocity"
           to="/dabo/odom" /> <!-- /topic-odometry-of-the-robot -->

 <!-- People Tracking MHT parameters for outdoor environments with few false alarms-->
        <param name="~/frame_id"  type="string" value="map" /> <!-- tf of the map for the tracker -->
    <param name="local_tracker"             type="bool"   value="false"/>
    <param name="~/association_threshold"   type="double" value="1.0" />
    <param name="~/confirmation_threshold"  type="double" value="0.9" />
    <param name="~/deletion_threshold"      type="double" value="0.4" />
    <param name="~/laser_probability"       type="double" value="0.9" />
    <param name="~/laser_false_alarm"       type="double" value="0.1" />
    <param name="~/laser_new_track"         type="double" value="0.11" />
    <param name="~/laser_no_detection"      type="double" value="0.97" />
    <param name="~/laser_no_confirmed"      type="double" value="0.02" />
               <param name="~/laser_no_confirmed_iteration" type="double" value="0.01" />
               <param name="~/augment_covariance_track" type="double" value="0.5" />
    <param name="~/mht_velocity_margin" type="double" value="1.0" />
    <param name="~/covariance_no_track" type="double" value="5.0" />
  </node>
```

```
  <!-- Laser People map Filter -->
   <!-- published topics:  /$(env ROBOT)/peopleFiltered -->
   <!-- subscribed topics: /$(env ROBOT)/people -->
   <!--                    /$(env ROBOT)map -->
   <!-- service clients: -->
   <!-- service servers: -->
   <!-- action clients: -->
   <!-- action servers: -->
    <node pkg ="iri_laser_people_map_filter"
          type="iri_laser_people_map_filter"
          name="lpmf">

                      <param name="~/filter"   type="bool" value="True" /> <!-- true to enable
iltering when we have a map -->
      <param name="~/neighborRadius"   type="double" value="0.3" />
      <param name="~/markerWidth"      type="double" value="0.7" />
      <param name="~/markerHeight"     type="double" value="1.5" />
      <param name="~/markerR"          type="double" value="1.0" />
      <param name="~/markerG"          type="double" value="1.0" />
      <param name="~/markerB"          type="double" value="0.0" />
      <param name="~/markerA"          type="double" value="0.9" />
      <param name="~/markerLifeTime"   type="double" value="0.1" />

    <remap from="/ana/lpmf/people"
           to="/ana/lpd/people" />
    <remap from="/ana/lpmf/people_out"
           to="/ana/lpd/peopleMapFiltered" />
    <remap from="/ana/lpmf/map"
           to="/ana/map" /> <!-- topic of the map -->

    </node>
</group>
```

```
  <!-- Laser People map Filter -->
   <!-- published topics:  /$(env ROBOT)/peopleFiltered -->
   <!-- subscribed topics: /$(env ROBOT)/people -->
   <!--                    /$(env ROBOT)map -->
   <!-- service clients: -->
   <!-- service servers: -->
   <!-- action clients: -->
   <!-- action servers: -->
    <node pkg ="iri_laser_people_map_filter"
          type="iri_laser_people_map_filter"
          name="laser_people_map_filter">

                      <param name="~/filter"   type="bool" value="True" /> <!-- true to enable
filtering when we have a map -->
      <param name="~/neighborRadius"   type="double" value="0.3" />
      <param name="~/markerWidth"      type="double" value="0.7" />
      <param name="~/markerHeight"     type="double" value="1.5" />
      <param name="~/markerR"          type="double" value="1.0" />
      <param name="~/markerG"          type="double" value="1.0" />
      <param name="~/markerB"          type="double" value="0.0" />
      <param name="~/markerA"          type="double" value="0.9" />
      <param name="~/markerLifeTime"   type="double" value="0.1" />

    <remap from="/laser_people_map_filter/people"
           to="/dabo/people" />
    <remap from="/laser_people_map_filter/people_out"
           to="/lpd_2/peopleMapFiltered" />
    <remap from="/laser_people_map_filter/map"
           to="/dabo/map" /> <!-- topic of the map -->

    </node>
```

```
<!-- Laser People Detector (remap) -->
  <!-- published topics:  /$(env ROBOT)/people -->
  <!-- subscribed topics: /$(env ROBOT)/sensors/front_laser_scan -->
  <!-- service clients: -->
  <!-- service servers: -->
  <!-- action clients: -->
  <!-- action servers: -->
 <node pkg ="iri_laser_people_detection"
       type="iri_laser_people_detection"
       name="lpd"
       output="screen"
       >

     <param name="~/rangeThreshold"     type="double" value="0.0" />
     <param name="~/detectionThreshold" type="double" value="-0.07" />
                    <param name="~/personRadius"        type="double" value="0.7" />
     <!-- load different trained detectors (to switch at rangeThreshold) -->
     <param name="~/boostFilePath"      type="string" value="/usr/local/include/iridrivers/boostData
boostFile.txt"  />
     <param name="~/boostFilePath2"     type="string" value="/usr/local/include/iridrivers/boostData
boostFile2.txt" />
     <!-- transform detections to base_link frame -->
     <param name="~/selectPosesFrame"   type="bool"   value="False" />
     <param name="~/posesFrame"         type="string" value="/scan" />  <!-- frame where you want
the poses -->
     <param name="~/selectScanFrame"    type="bool"   value="False" />
     <param name="~/scanFrame"          type="string" value="/scan" /> <!-- frame of the scan -->
     <!-- visualization -->
     <param name="~/markerWidth"        type="double" value="0.7" />
     <param name="~/markerHeight"       type="double" value="0.4" />
     <param name="~/markerR"            type="double" value="0.0" />
     <param name="~/markerG"            type="double" value="0.0" />
     <param name="~/markerB"            type="double" value="0.5" />
     <param name="~/markerA"            type="double" value="0.5" />
     <!-- filter detections by range/x/y -->
     <param name="~/filterPosesMode"    type="bool"   value="False" />
     <param name="~/filterR"            type="double" value="0.0" />
     <param name="~/filterX"            type="double" value="0.0" />
     <param name="~/filterY"            type="double" value="0.0" />

   <remap from="/ana/lpd/scan"
          to="/ana/sensors/scan" /> <!-- topic: scan del robot -->
   <remap from="/ana/lpd/people"
          to="/ana/lpd/people" />
 </node>
```

```
<!-- Laser People Detector (remap) -->
  <!-- published topics:  /$(env ROBOT)/people -->
  <!-- subscribed topics: /$(env ROBOT)/sensors/front_laser_scan -->
  <!-- service clients: -->
  <!-- service servers: -->
  <!-- action clients: -->
  <!-- action servers: -->
 <node pkg ="iri_laser_people_detection"
       type="iri_laser_people_detection"
       name="lpd_2"
       output="screen"
       >

     <param name="~/rangeThreshold"     type="double" value="0.0" />
     <param name="~/detectionThreshold" type="double" value="-0.07" />
                    <param name="~/personRadius"        type="double" value="0.7" />
     <!-- load different trained detectors (to switch at rangeThreshold) -->
     <param name="~/boostFilePath"      type="string" value="/usr/local/include/iridrivers/boostData
boostFile.txt"  />
     <param name="~/boostFilePath2"     type="string" value="/usr/local/include/iridrivers/boostData
boostFile2.txt" />
     <!-- transform detections to base_link frame -->
     <param name="~/selectPosesFrame"   type="bool"   value="False" />
     <param name="~/posesFrame"         type="string" value="/scan" />  <!-- frame where you want the
poses -->
     <param name="~/selectScanFrame"    type="bool"   value="False" />
     <param name="~/scanFrame"          type="string" value="/scan" /> <!-- frame of the scan -->
     <!-- visualization -->
     <param name="~/markerWidth"        type="double" value="0.7" />
     <param name="~/markerHeight"       type="double" value="0.4" />
     <param name="~/markerR"            type="double" value="0.0" />
     <param name="~/markerG"            type="double" value="0.0" />
     <param name="~/markerB"            type="double" value="0.5" />
     <param name="~/markerA"            type="double" value="0.5" />
     <!-- filter detections by range/x/y -->
     <param name="~/filterPosesMode"    type="bool"   value="False" />
     <param name="~/filterR"            type="double" value="0.0" />
     <param name="~/filterX"            type="double" value="0.0" />
     <param name="~/filterY"            type="double" value="0.0" />

   <remap from="/lpd_2/scan"
          to="/dabo/sensors/front_scan" /> <!-- topic: scan del robot -->
   <remap from="/lpd_2/people"
          to="/dabo/people" />
 </node>
```

```
  <node pkg ="iri_laser_people_detection"
        type="iri_laser_people_detection"
        name="lpd"
```

... Other laser detector. Dabo needs 2, Ana only needs 1.

The ros-navigation for Ana is different than for Dabo, we need to launch more nodes:

```
<include file="$(find iri_rosnav)/launch/nav.launch">
   <arg name="ns"                  value="ana"/>
   <arg name="path"                value="$(find iri_ana_rosnav)/params"/>
   <arg name="move_base_params"    value="$(arg move_base_params)"/>
   <arg name="costmap_common_params" value="$(arg costmap_common_params)"/>
   <arg name="costmap_local_params"  value="$(arg costmap_local_params)"/>
   <arg name="costmap_global_params" value="$(arg costmap_global_params)"/>
   <arg name="map_frame_id"        value="map"/>
   <arg name="odom_frame_id"       value="ana/odom"/>
   <arg name="base_frame_id"       value="ana/base_footprint"/>
   <arg name="map_topic"           value="/ana/map"/>
   <arg name="map_service"         value="/ana/static_map"/>
   <arg name="odom_topic"          value="/ana/odom"/>
   <arg name="cmd_vel_topic"       value="/ana/navigation/cmd_vel"/>
   <arg name="scan_topic"          value="/ana/sensors/scan"/>
   <arg name="use_map"             value="true"/>
   <arg name="use_map_server"      value="true"/>
   <arg name="map_name"            value="$(arg map_name)"/>
   <arg name="use_amcl"            value="true"/>
   <arg name="amcl_config"         value="$(find iri_ana_rosnav)/params/
amcl_companion_test_ely2021_mapFME.yaml"/>
   <arg name="initial_x"           value="$(arg initial_x)"/>
   <arg name="initial_y"           value="$(arg initial_y)"/>
   <arg name="initial_yaw"         value="$(arg initial_yaw)"/>
   <arg name="use_fake_loc"        value="false"/>
   <arg name="use_gmapping"        value="false"/>
   <arg name="local_planner"       value="$(arg local_planner)"/>
   <arg name="global_planner"      value="$(arg global_planner)"/>
   <arg name="output"              value="$(arg output)" />
   <arg name="launch_prefix"       value="$(arg launch_prefix)" />
 </include>

 <include file="$(find iri_rosnav)/launch/include/pointcloud_to_laserscan.launch">
   <arg name="ns"             value="ana"/>
   <arg name="node_name"      value="pcl_to_laserscan"/>
   <arg name="scan_topic"     value="/ana/sensors/scan"/>
   <arg name="cloud_topic"    value="/ana/sensors/velodyne_points"/>
   <arg name="config_file"    value="$(find iri_ana_rosnav)/params/pointcloud_to_laserscan.yaml" />
   <arg name="output"         value="$(arg output)" />
   <arg name="launch_prefix"  value="$(arg launch_prefix)" />
 </include>
```

Ros-nav for Dabo:

```
<!-- INI nav nodes _companion-->
 <include file="$(find iri_dabo_rosnav)/launch/nav_map_companion.launch">
  <arg name="map_name"              value="master_big" />
  <arg name="initial_x"             value="19.0"/>
   <arg name="initial_y"             value="39.0"/>
   <arg name="initial_yaw"           value="0.0"/>
  </include>
```

Important: You need to adapt the navigation, detection and tracking to the sensors and characteristics that each robot has.

Note: In these launchs change all the remaps for each robot and there are differences due to the different robot shape and sensors.

2.3. Change the robot in the launch for the one person accompaniment:

Terminal$ roslaunch iri_robot_assaop gazebo_ASSAOP_OK_ana_brl.launch

Terminal$ roslaunch iri_robot_assaop gazebo_ASSAOP_BRL_OK.launch

Note: The launch for Ana-robot and Dabo-robot are included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/iri_navigation/iri_robot_assaop/launch/

Also, both launchs use the same environment, to only see the changes due to change the robot.

Differences inside the launch files:

gazebo_ASSAOP_OK_ana_brl.launch

```
<!-- -->
<launch>


   <!-- Ini Navigation Companion Nodes -->
  <!--launch-prefix="xterm -e ddd -args"  -->

 <group ns="$(optenv ROBOT ana)">
 <rosparam command="delete" ns="/ana/move_base" />

   <!--launch-prefix="xterm -e ddd -args"  machine="visio"-->
   <node pkg ="move_base"
         type="move_base"
         name="move_base"
         output="screen">
     <remap from="/ana/move_base_ana/cmd_vel"
            to="/ana/navigation/cmd_vel" />
     <remap from="/ana/move_base_ana/odom"
            to="/ana/odom" />
     <remap from="/ana/front_scan"
            to="/ana/sensors/scan" />
     <remap from="/ana/rear_scan"
            to="/ana/sensors/scan" />
     <remap from="/ana/tracks"
             to="/ana/mht/tracks" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/move_base_params_assaop.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/ana/costmap_common_params.yaml" command="load"
ns="global_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/costmap_common_params.yaml" command="load"
ns="local_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/local_costmap_params.yaml"  command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/ana/global_costmap_params.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana2/akp_local_planner_params.yaml" command="load" /
>
<rosparam file="$(find iri_ana_rosnav)/params/amcl_companion_test_ely2021_mapFME.yaml" command="load" />
 <!--<rosparam file="$(find iri_dabo_rosnav)/params/global_planner/global_planner_params.yaml"
command="load" />
 <rosparam file="$(find iri_dabo_rosnav)/params/mux.yaml" command="load" />-->
     <param name="AkpLocalPlanner/destination_map_path" value="$(find iri_robot_assaop)/maps/
master_big_destinations_Gazebo_sim.txt" />
       <param name="AkpLocalPlanner/results_filename_Zanlungo" value="name_file.txt" />
     <param name="AkpLocalPlanner/robot" value="ana"/>
     <param name="AkpLocalPlanner/person_goal_id"  value="200"/>
     <param name="AkpLocalPlanner/speed_k"  value="1"/>
     <param name="AkpLocalPlanner/person_radi_amp"  value="0.2"/>
     <param name="AkpLocalPlanner/change_treshold_distance_between_steps"  value="0.3"/>
       <!-- for SIM-->
       <param name="AkpLocalPlanner/change_sim"  value="False"/>
       <param name="AkpLocalPlanner/sim_target_per"  value="False"/>
       <param name="AkpLocalPlanner/frame_robot_footprint"  value="ana/base_link"/>
   </node>
 </group>
```

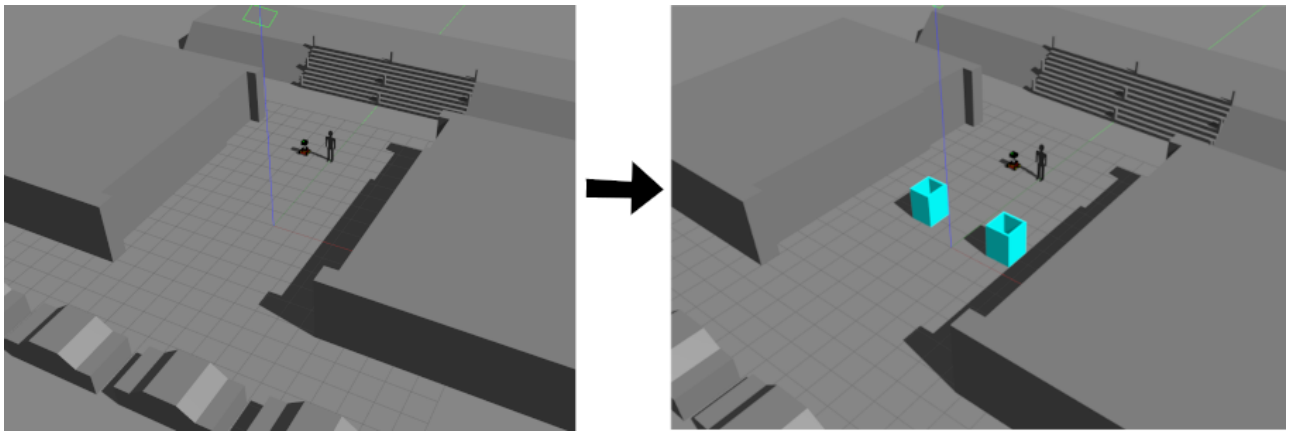
gazebo_ASSAOP_BRL_OK.launch

```
<!-- -->
<launch>


   <!-- Ini Navigation Companion Nodes -->
  <!--launch-prefix="xterm -e ddd -args"  -->

 <group ns="dabo">
  <rosparam command="delete" ns="/dabo/move_base" />

   <node pkg ="move_base"
         type="move_base"
         name="move_base"
         output="screen">
     <remap from="/dabo/move_base/cmd_vel"
            to="/dabo/navigation/cmd_vel" />
     <remap from="/dabo/move_base/odom"
            to="/dabo/odom" />
     <remap from="/dabo/front_scan"
            to="/dabo/sensors/front_scan" />
     <remap from="/dabo/rear_scan"
            to="/dabo/sensors/rear_scan" />
     <remap from="/dabo/tracks"
             to="/mht_2/tracks" />
     <rosparam file="$(find iri_robot_assaop)/config/dabo/move_base_params.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana/costmap_common_params.yaml" command="load"
ns="global_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/dabo/costmap_common_params.yaml" command="load"
ns="local_costmap" />
     <rosparam file="$(find iri_robot_assaop)/config/dabo/local_costmap_params.yaml"
command="load" />
     <rosparam file="$(find iri_robot_assaop)/config/ana/global_costmap_params.yaml" command="load" /
>
     <rosparam file="$(find iri_robot_assaop)/config/dabo/akp_local_planner_params.yaml"
command="load" />
     <rosparam file="$(find iri_dabo_rosnav)/params/amcl_companion_test_ely2021_mapFME.yaml"
command="load" />
 <!--<rosparam file="$(find iri_dabo_rosnav)/params/global_planner/global_planner_params.yaml"
command="load" />
 <rosparam file="$(find iri_dabo_rosnav)/params/mux.yaml" command="load" />-->
     <param name="AkpLocalPlanner/destination_map_path" value="$(find iri_robot_assaop)/maps/
master_big_destinations_Gazebo_sim.txt" />
       <param name="AkpLocalPlanner/results_filename_Zanlungo" value="name_file.txt" />
     <param name="AkpLocalPlanner/robot" value="dabo"/>
     <param name="AkpLocalPlanner/person_goal_id"  value="200"/>
     <param name="AkpLocalPlanner/speed_k"  value="1"/>
     <param name="AkpLocalPlanner/person_radi_amp"  value="0.2"/>
     <param name="AkpLocalPlanner/change_treshold_distance_between_steps"  value="0.3"/>
       <!-- for SIM-->
       <param name="AkpLocalPlanner/change_sim"  value="False"/>
       <param name="AkpLocalPlanner/sim_target_per"  value="False"/>
       <param name="AkpLocalPlanner/frame_robot_footprint"  value="dabo/base_link"/>
   </node>
 </group>
```

Note: In these launchs change all the remaps for each robot, all the configurations of the local-planner which implements the accompaniment due to the different characteristics of each robot; and all the used robot frames.

3. Include more static obstacles in the map

You can create maps that include more obstacles using the maps included in the docker. For example, the map **fme_door_open_with_obst** is obtained from the **fme_door_open** only adding two static obstacles at the middle of the environment that simulate a “door”.



These maps are included in folders:

1. For gazebo: catkin_ws/src/iri_navigation/iri_gazebo_worlds/worlds

In order to include these static obstacles in the map o fme_door_open.world, we need to open this hmlt document and change it to obtain the hmlt document fme_door_open_with_obst.world. To do it, we need to include a piece of code that includes 8 boxes that perform these two square obstacles.

fme_door_open.world

```
<?xml version="1.0" ?>
<sdf version="1.4">
  <world name="fme_door_open">
    <!--
    <include>
      <uri>model://sun</uri>
    </include>
    -->

    <light type="directional" name="sun">
      <cast_shadows>true</cast_shadows>
      <pose>0 0 10 0.2 0.2 0.2</pose>
      <diffuse>0.8 0.8 0.8 1</diffuse>
      <specular>0.2 0.2 0.2 1</specular>
      <attenuation>
        <range>1000</range>
        <constant>0.9</constant>
        <linear>0.01</linear>
        <quadratic>0.001</quadratic>
      </attenuation>
      <direction>-0.5 0.1 -0.9</direction>
    </light>

    <include>
      <uri>model://ground_plane</uri>
    </include>

    <model name="fme_door_open">
      <static>true</static>
      <link name="link">
        <pose>0 0 0 0 0 0</pose>
        <collision name="collision">
          <geometry>
            <mesh>
              <uri>model://meshes/fme_door_open.dae</uri>
              <scale>1 1 1</scale>
            </mesh>
          </geometry>
        </collision>
        <visual name="visual">
          <geometry>
            <mesh>
              <uri>model://meshes/fme_door_open.dae</uri>
              <scale>1 1 1</scale>
            </mesh>
          </geometry>
        </visual>
      </link>
    </model>

<gui fullscreen='0'>
      <camera name='user_camera'>
        <pose frame=''>10.0958 -9.41664 10.969 0 0.667643 2.2282</pose>
        <view_controller>orbit</view_controller>
        <projection_type>perspective</projection_type>
```

fme_door_open_with_obst.world

```
<?xml version="1.0" ?>
<sdf version="1.4">
  <world name="fme_door_open_with_obst">
    <!--
    <include>
      <uri>model://sun</uri>
    </include>
    -->

    <light type="directional" name="sun">
      <cast_shadows>true</cast_shadows>
      <pose>0 0 10 0.2 0.2 0.2</pose>
      <diffuse>0.8 0.8 0.8 1</diffuse>
      <specular>0.2 0.2 0.2 1</specular>
      <attenuation>
        <range>1000</range>
        <constant>0.9</constant>
        <linear>0.01</linear>
        <quadratic>0.001</quadratic>
      </attenuation>
      <direction>-0.5 0.1 -0.9</direction>
    </light>

    <include>
      <uri>model://ground_plane</uri>
    </include>

    <model name="fme_door_open">
      <static>true</static>
      <link name="link">
        <pose>0 0 0 0 0 0</pose>
        <collision name="collision">
          <geometry>
            <mesh>
              <uri>model://meshes/fme_door_open.dae</uri>
              <scale>1 1 1</scale>
            </mesh>
          </geometry>
        </collision>
        <visual name="visual">
          <geometry>
            <mesh>
              <uri>model://meshes/fme_door_open.dae</uri>
              <scale>1 1 1</scale>
            </mesh>
          </geometry>
        </visual>
      </link>
    </model>

<model name="my_model1">
  <pose>-2.0 1.5 0 0 0 0.0</pose>
  <static>true</static>
    <link name="link">
      <inertial>
        <mass>1.0</mass>
        <inertia>
```

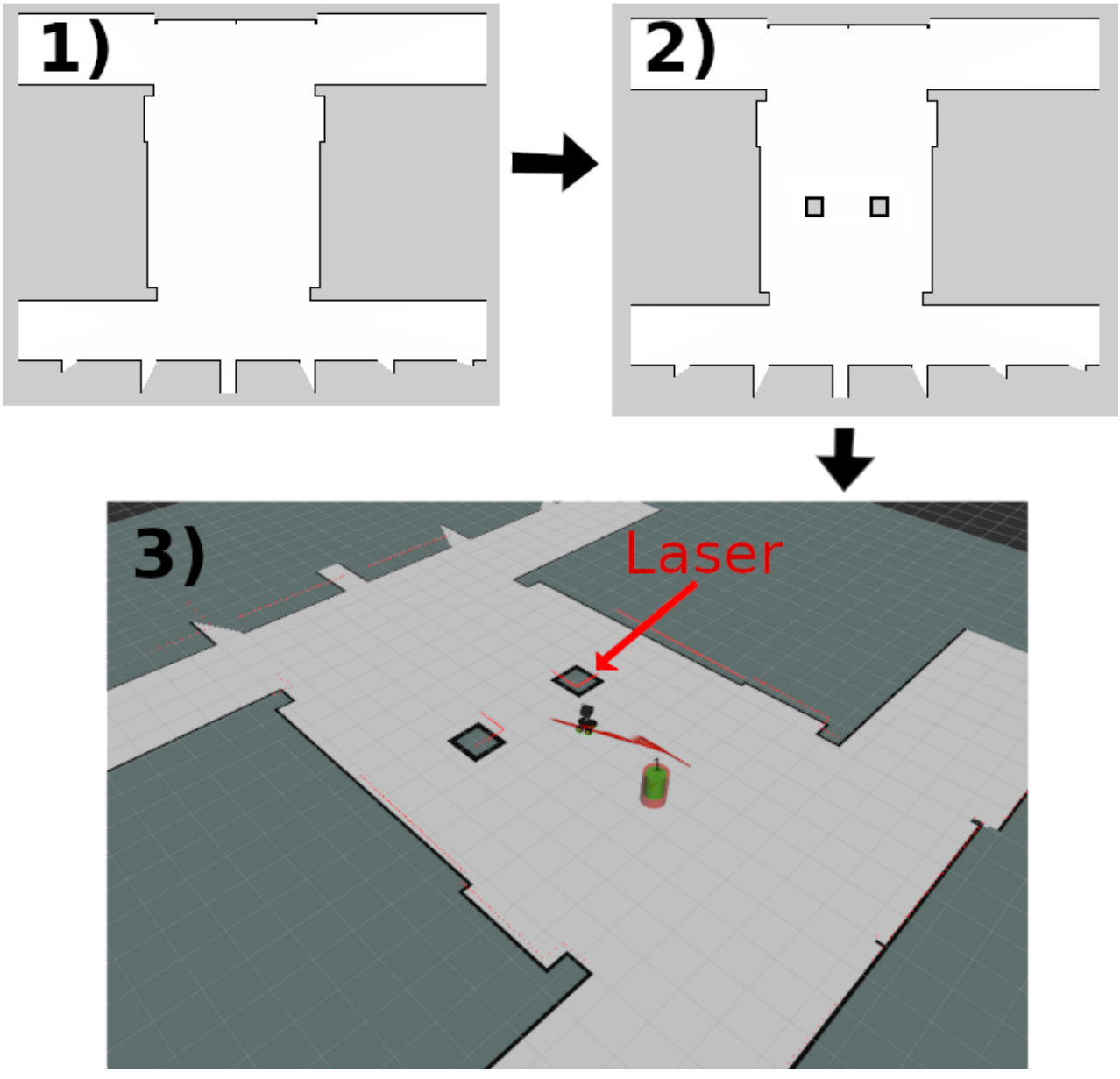
```
          <ixx>1.0</ixx>
          <ixy>0.0</ixy>
          <ixz>0.0</ixz>
          <iyy>-1.0</iyy>
          <iyz>0.0</iyz>
          <izz>1.0</izz>
        </inertia>
      </inertial>
      <collision name="collision">
        <geometry>
          <box>
            <size>1.0 0.1 3</size>
          </box>
        </geometry>
      </collision>
      <visual name="visual">
        <geometry>
          <box>
            <size>1.0 0.1 3</size>
          </box>
        </geometry>
          <material>
            <script>
              <uri>file://media/materials/scripts/gazebo.material</uri>
              <name>Gazebo/Turquoise</name>
            </script>
          </material>
      </visual>
    </link>
  </model>
  <model name="my_model8">
    <pose>-2.0 0.5 0 0 0 0.0</pose>
    <static>true</static>
      <link name="link">
        <inertial>
          <mass>1.0</mass>
          <inertia>
            <ixx>1.0</ixx>
            <ixy>0.0</ixy>
            <ixz>0.0</ixz>
            <iyy>-1.0</iyy>
            <iyz>0.0</iyz>
            <izz>1.0</izz>
          </inertia>
        </inertial>
        <collision name="collision">
          <geometry>
```

... here are included the 8 boxes that form the additional obstacle added in the map

```
          </material>
      </visual>
    </link>
  </model>
<model name="my_model7">
  <pose>-2.45 1.0 0 0 0 0.0</pose>
  <static>true</static>
    <link name="link">
      <inertial>
        <mass>1.0</mass>
        <inertia>
          <ixx>1.0</ixx>
          <ixy>0.0</ixy>
          <ixz>0.0</ixz>
          <iyy>-1.0</iyy>
          <iyz>0.0</iyz>
          <izz>1.0</izz>
        </inertia>
      </inertial>
      <collision name="collision">
        <geometry>
          <box>
            <size>0.1 0.9 3</size>
          </box>
        </geometry>
      </collision>
      <visual name="visual">
        <geometry>
          <box>
            <size>0.1 0.9 3</size>
          </box>
        </geometry>
          <material>
            <script>
              <uri>file://media/materials/scripts/gazebo.material</uri>
              <name>Gazebo/Turquoise</name>
            </script>
          </material>
      </visual>
    </link>
  </model>


<gui fullscreen='0'>
      <camera name='user_camera'>
        <pose frame=''>10.0958 -9.41664 10.969 0 0.667643 2.2282</pose>
        <view_controller>orbit</view_controller>
        <projection_type>perspective</projection_type>
      </camera>
    </gui>

  </world>
</sdf>
```

2. For all the nodes: catkin_ws/src/iri_navigation/iri_maps/maps

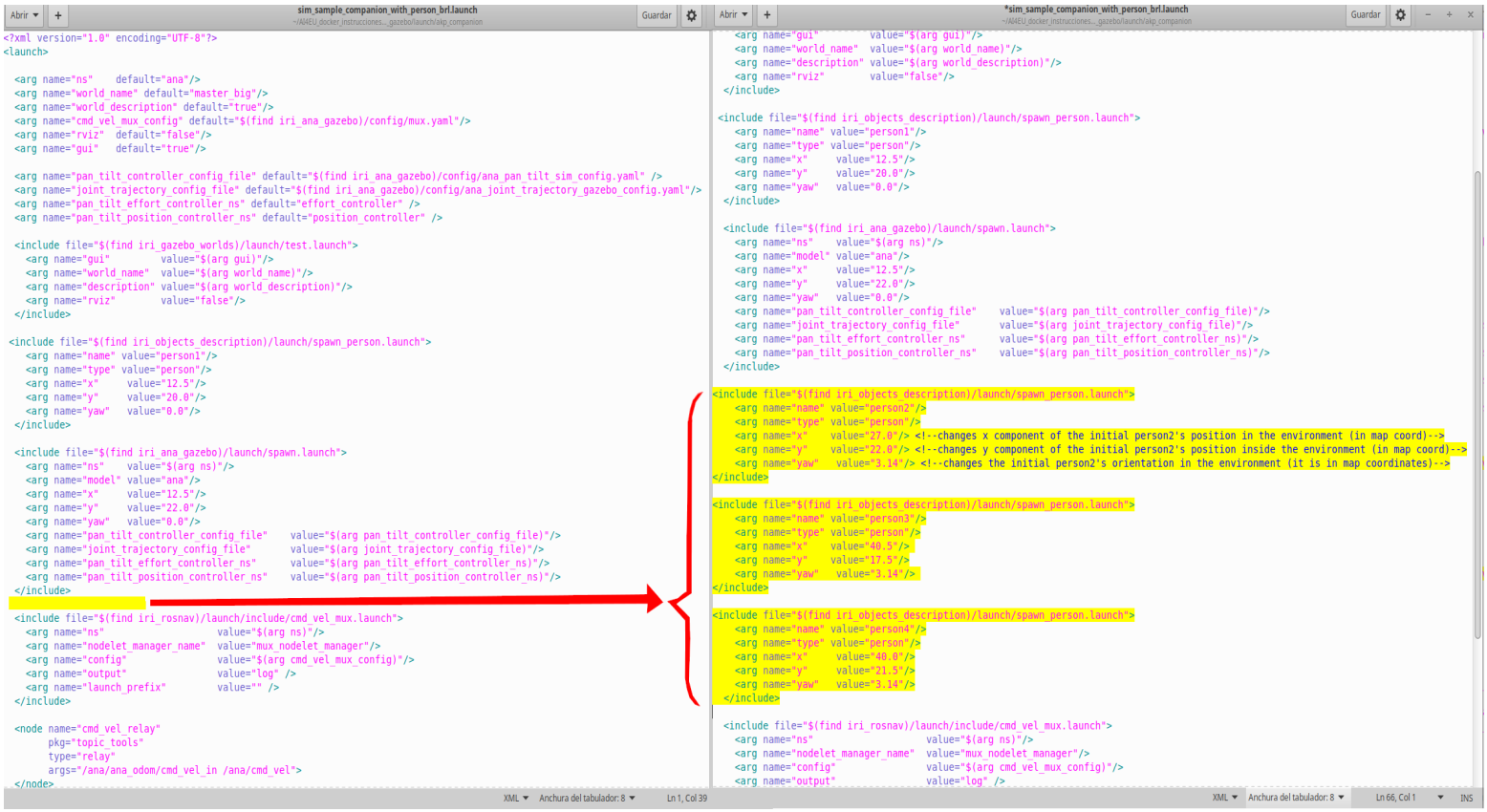
In order to include these static obstacles in the map **fme_door_open** and obtain the new map of **fme_door_open_with_obst,** we need to draw these obstacles in this **fme_door_open.pgm** to obtain the **fme_door_open_with_obst.pgm** image with any painting software. After that, we need to be sure that the robot detects these obstacles over the position of the obstacles in the image, by launching the launch *roslaunch iri_ana_gazebo sim_sample_companion_with_person.launch* ***world_name:=fme_door_open_with_obst***, and checking it.



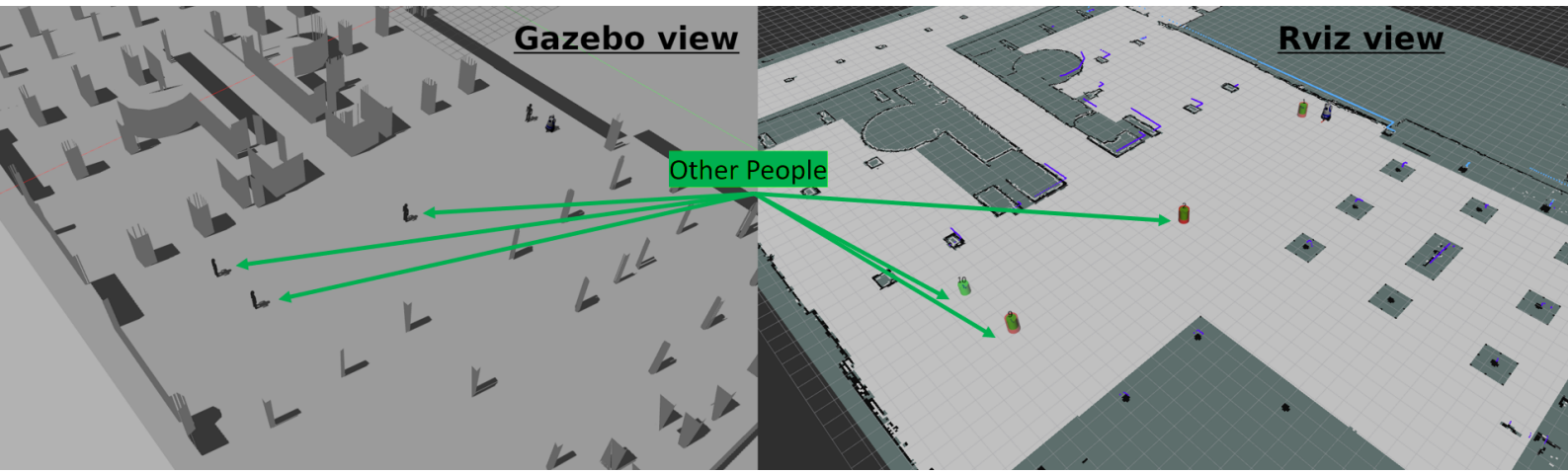
4. Add more people in the environment

You can add other people in the environment to cross the group's path and see the dynamic people avoidance of the robot's accompaniment.

In order to include these people in the environment, you need to add them in the gazebo-launch. For example in: roslaunch iri_ana_gazebo sim_sample_companion_with_person_brl.launch world_name:=master_big. Then, you need to open the launch file sim_sample_companion_with_person_brl.launch and include more people, like the next image shows:



Now, if you include the same people in the launchs of Dabo-robot and execute the launch that includes Gazebo and the rviz (roslaunch iri_dabo_gazebo sim_gazebo_dabo_companion.launch), you see the result of the next image:



If you want to move these people, you have to use the next teleop-rosrun-commands inside a Ubuntu-terminal:

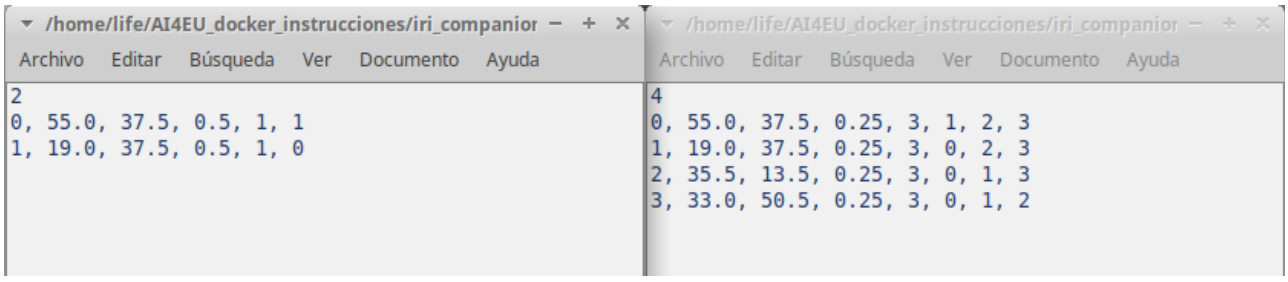
$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py cmd_vel:=/**person2**/cmd_vel __name:=**person2**

$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py cmd_vel:=/**personX**/cmd_vel __name:=**personX**

Where personX is the name of the person included, which needs to match with the name included in the launch to add this person.

5. Change the static destinations of the planner inside any environment

You can modify or add destinations in the environment to include other interesting points where people should go, like doors, corridors, stairs, wending machines, entrances or exits of streets or squares, etc. For example, we can add two more destinations in the BRL environment and pass from the file *master_big_destinations_Gazebo_sim.txt* to the file *master_big_destinations_Gazebo_sim_more_dest.txt*, in folder: *iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/iri_navigation/iri_robot_aspsi/maps/*



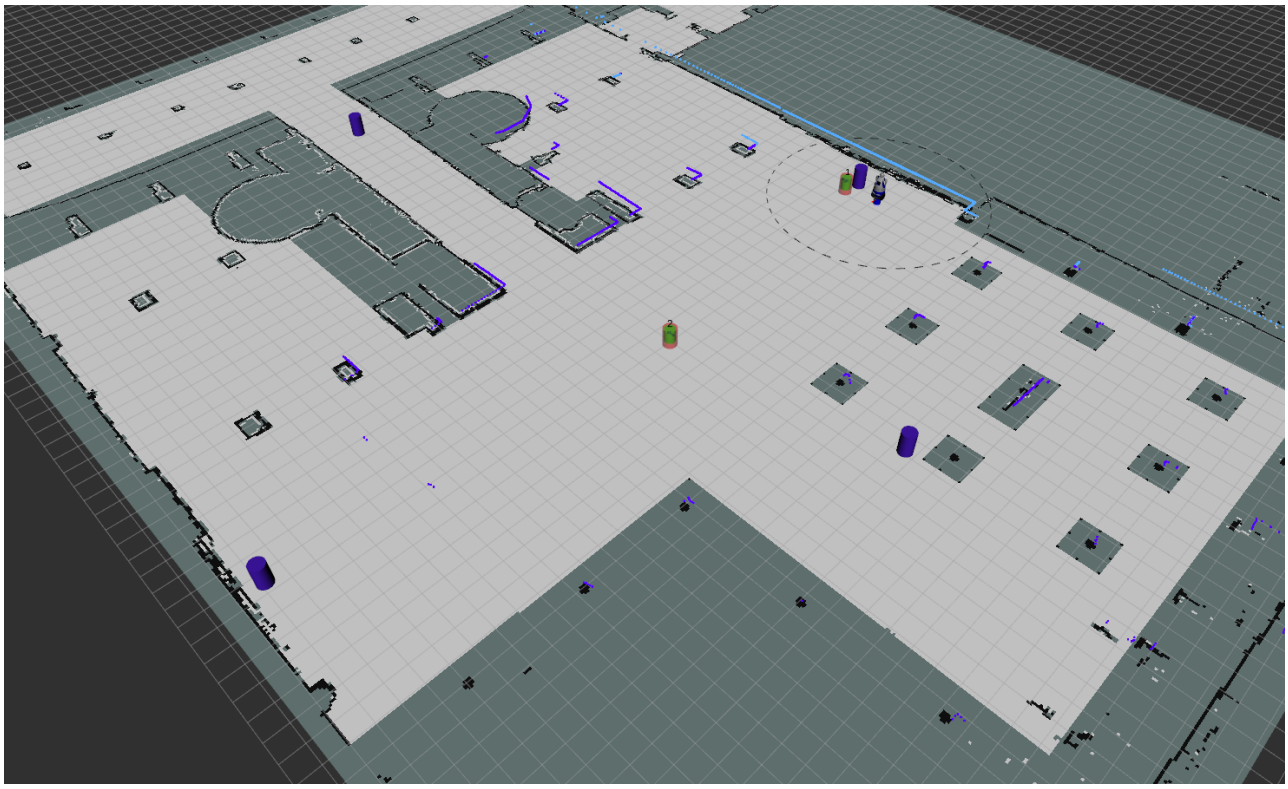
To understand how to change the files of the static destinations in any environment, you need to know that:

- The 4 is the number of destinations. Each destination is defined in one of the next four rows.

- If we focus on the first row, from left to right. The 0 is the identifier of this destination, the 55.0 is the x coordinate of the position of the destination, the 37.5 is the y coordinate of the position of the destination, the 0.25 is the probability of this destination to be selected from all the possible ones (We use equal probability for all the destinations, but you may change it if you know that one of them are most probable than the others), 3 is the number of other destinations that can be selected after arrive to the actual destination, and the 1, 2, 3 are the destinations that can be chosen after arrive to the actual destination. These last parameters are useful to include several groups of people that walk randomly between a subset of

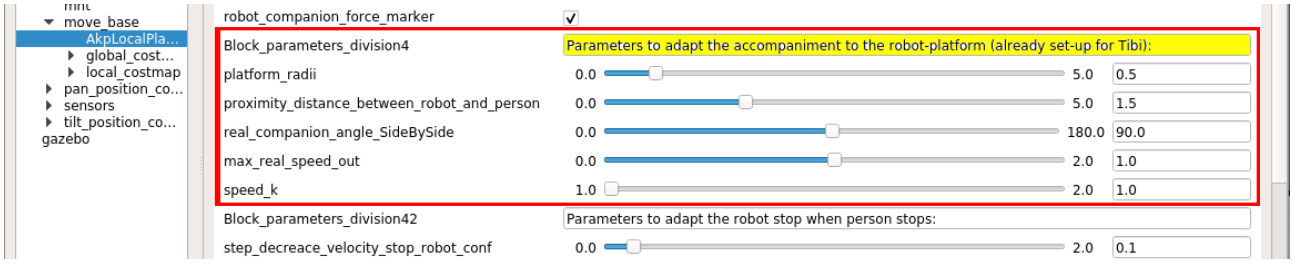
destination.

If we launch this launch for Dabo-robot *roslaunch iri_robot_aspsi gazebo_ASPSI_BRL_OK_more_dest.launch* instead the normal one r*oslaunch iri_robot_aspsi gazebo_ASPSI_BRL_OK.launch*, we can see the 4 destinations in the BRL and not only two like in the left image. This launch is to launch the accompaniment, which is the node that uses theses final static destinations inside the environment.



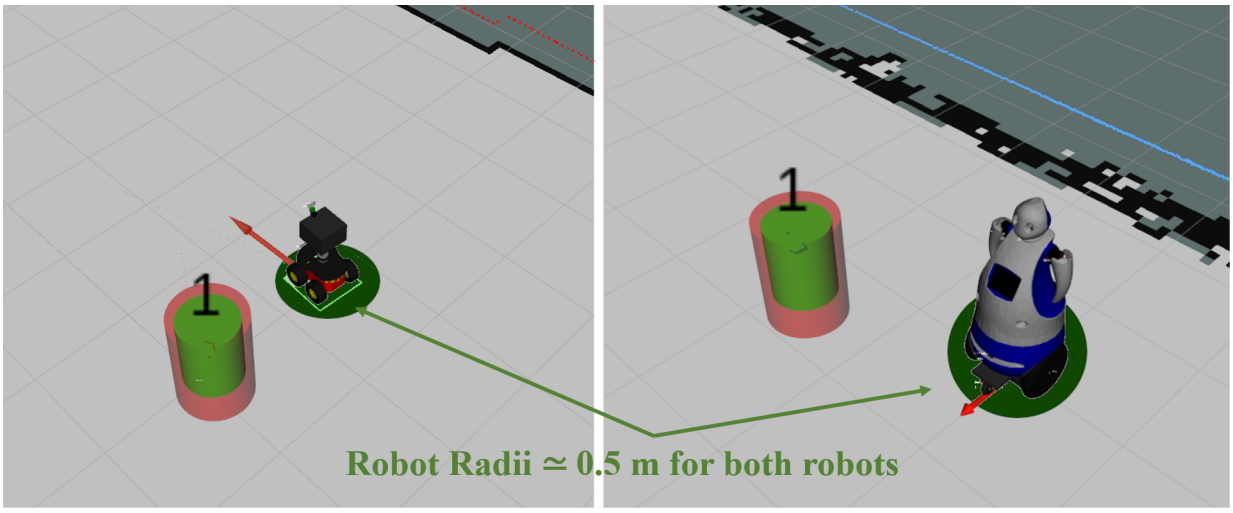
6. Adapt the parameters of accompaniment to the person preferences

The side-by-side accompaniment can be customized from outside using the *rqt_reconfigure*. Also this customization can convert the side-by-side accompaniment in other type of geometric-formation between the robot and the accompanied person.

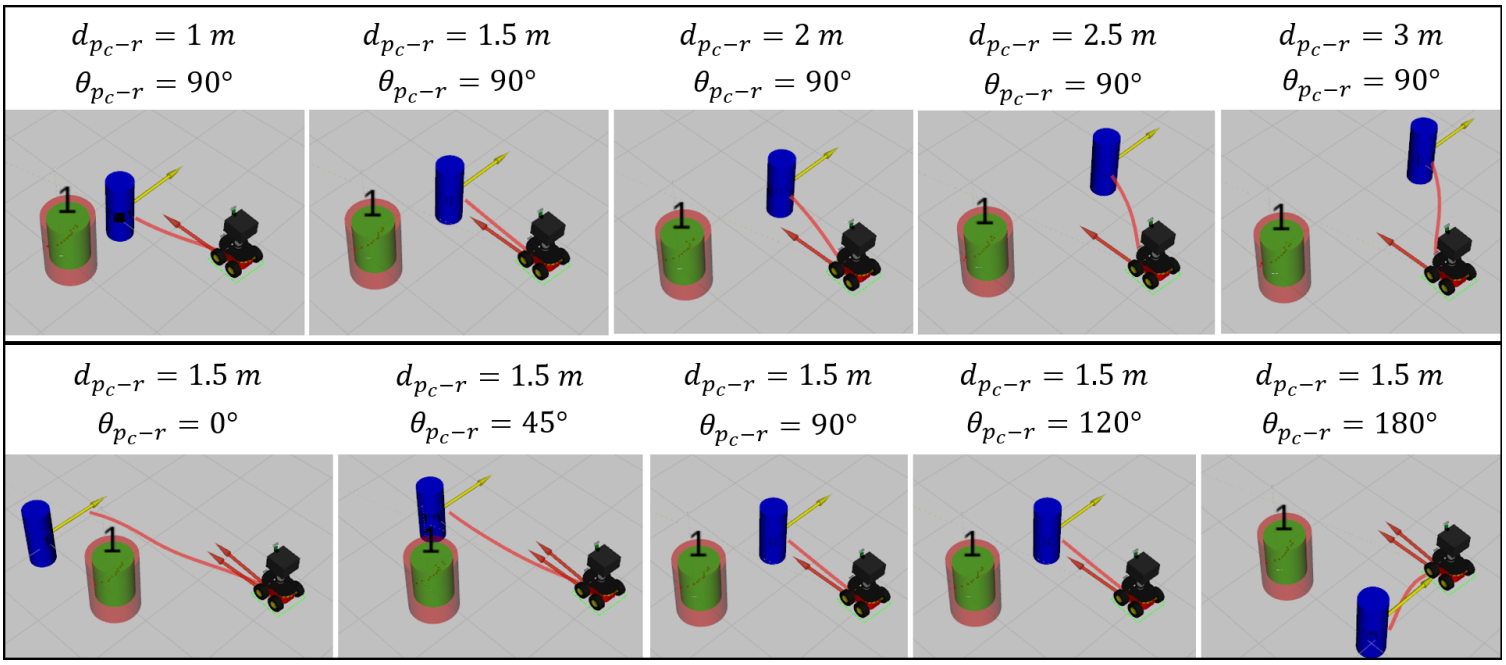


There are five parameters that you can customize: the platform_radii, the proximity_distance_between_robot_and_person, the real_companion_angle_sideBySide, the max_real_speed_out and speed_k.

- The platform_radii need to be set up to the radii of the platform of your robot. All our robots have more or less 0.5 meters of platform radii, then we had set-up this parameter to this value.



- The proximity_distance_between_robot_and_person and real_companion_angle_sideBySide customize the type of geometric-formation between the robot and the accompanied person, in our case the distance between them is 1.5 meters (include the person and robot radii, it is to say, the free space between them is between 0.7 and 0.5 meters). Also, you can reduce this distance but you need to be sure to do not collide in any case with the accompanied person, for security reasons. The real_companion_angle_sideBySide customizes the angle of accompaniment, in our case side-by-side, then is set-up to 90 degrees. For other type of formations you need to increase or decrease this value in the semi-circle of the lateral of the person. The 0.0 degrees should correspond to one-by-one formation with the robot in front of the person and the 180 degrees should correspond to one-by-one formation with the robot behind the person.



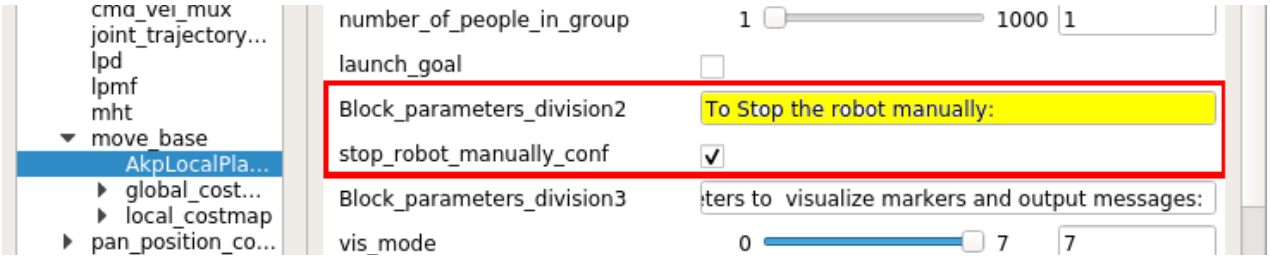
- The max_real_speed_out is the maximum real speed of the robot, in our case should be between 1m/s and 1.2 m/s. To set-up this parameter you need to know the maximum

velocity of your robot, the available free space of the environment and test if for security. We normally use these values because are the more safety ones and are enough to accompany people doing a leisure stroll.

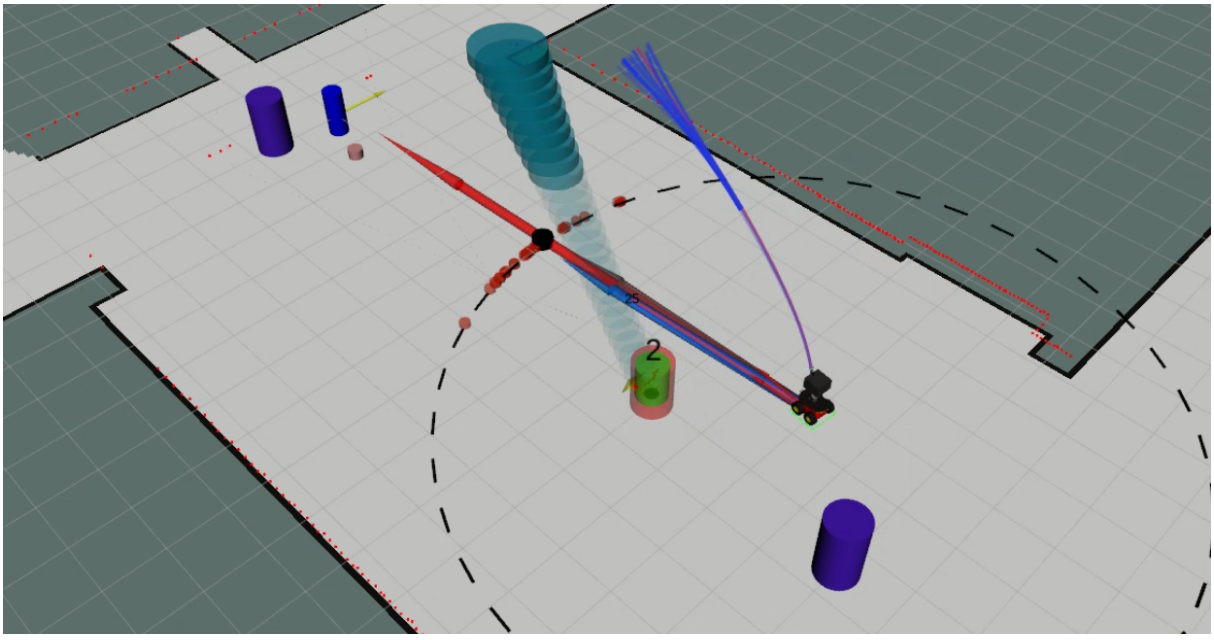
- The speed_k is due to our platform controller. In the real robot our controller divides the velocity and in consequence reduce it, then to correct this fact and send the real velocity to the real-robot, we need to multiply by 1.5. In simulation this value is not needed because the velocity are directly send to the platform. You need to know if your robot platform does something similar, because if you do not detect this fact the robot behavior is not the optimal one and it is not due by a wrong behavior of the planner.

7. Stop manually the robot to see the planer behavior well

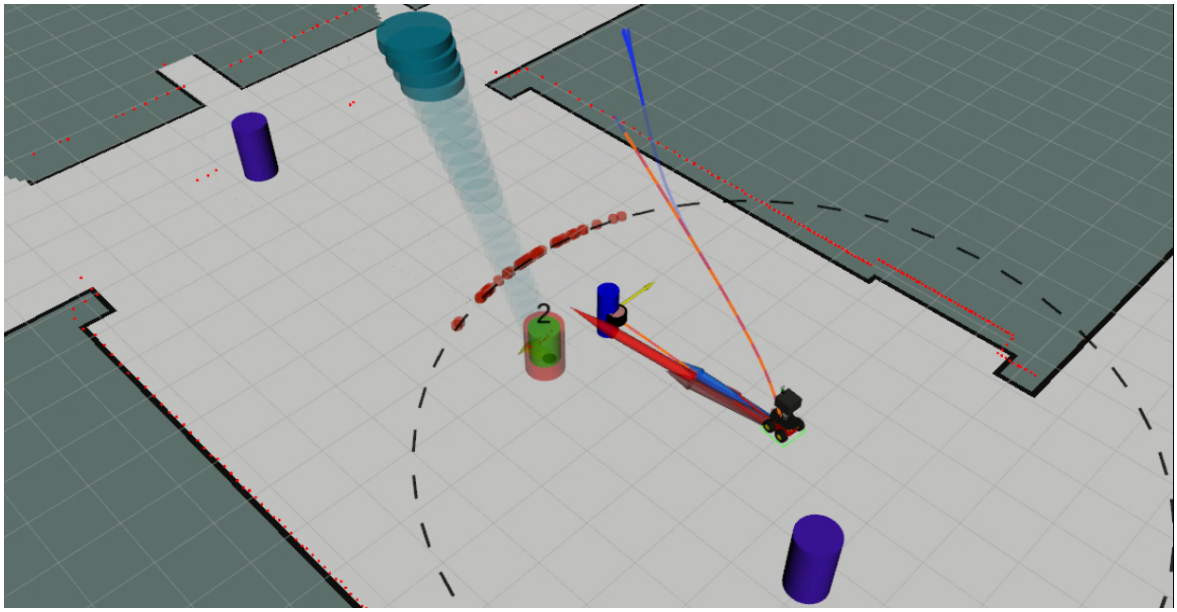
Using the rqt_reconfigure, we can stop manually the robot and see the planner behavior when we move the person inside the environment. To stop the robot you need to change to true the Boolean of stop_robot_manually_conf, like in the next image.



Here we see how the robot is planning all the paths to accompany the person in order to arrive to the final destination, because the person is less than 3 meters from it.

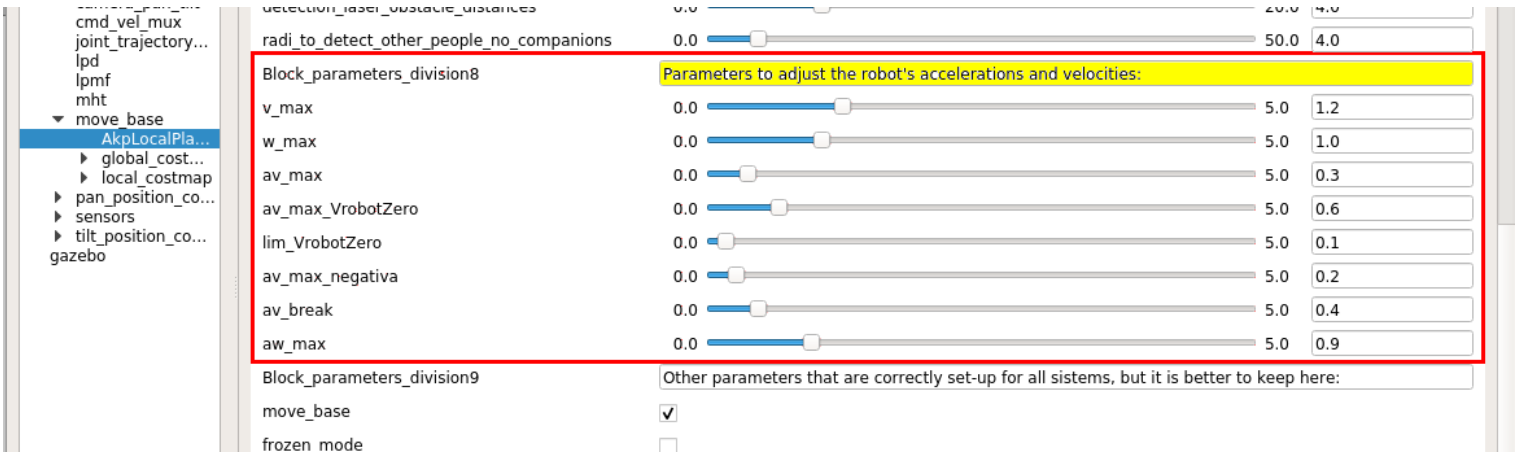


If the person is farther than 3 meters from the robot, it tries to arrive to the best accompaniment position respect to this person. In our case side-by-side with 90 degrees and 1.5 meters between them.



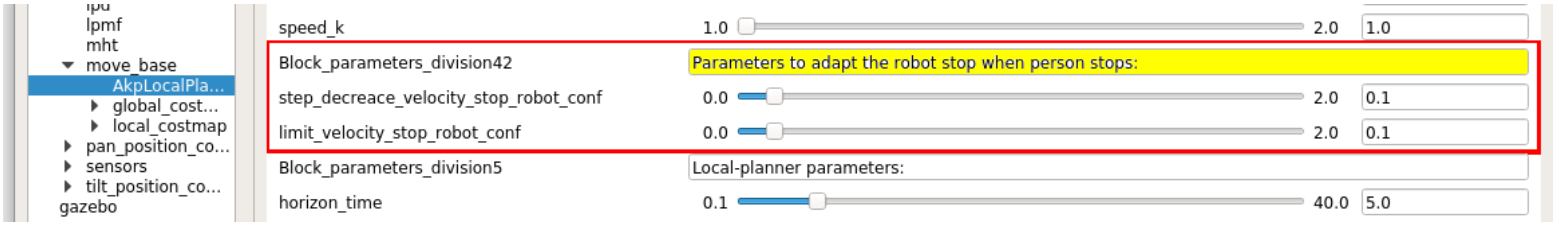
8. Adapt the robot velocities and accelerations to your robot

In the case that you change the robot model and use a different robot than our robots, you may net to adapt the robot velocities to the mass, size and capabilities of your robot. Then, using the rqt_reconfigure you can change these velocities to test the robot behavior before changing the velocities definitively. These are the limits of linear and angular velocities and accelerations to move the robot using the Social Force Model (SFM). If you need, you can change the maximum linear velocity (v_max), the maximum angular velocity (w_max), the maximum positive linear acceleration (av_max), the maximum negative linear acceleration (av_max_negativa), the linear acceleration of break (av_break), the maximum angular velocity (aw_max), and two linear acceleration and limit of this one to customize the robot stop (the av_max_VrobotZero and lim_VrobotZero). These accelerations and velocities need to be set-up experimentally to be customized for different robots.



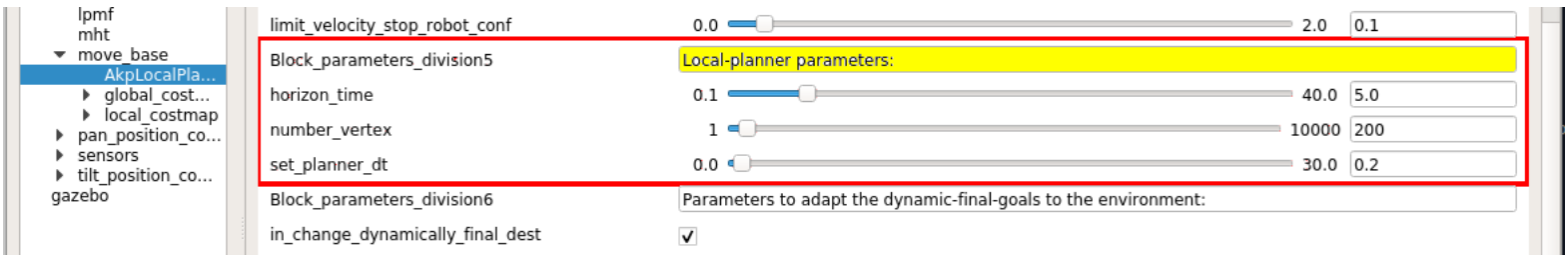
9. How to adapt the smooth stop for your robotic platform

We have two parameters to customize the smooths stopping robot's behavior. These parameters need to be set-up experimentally to be customized for different robots like in the case of the robot's maximum velocities and accelerations. You can change the limit of the velocity to stop the robot abruptly (limit_velocity_stop_robot_conf) and the increment of the velocity that the robot reduces in each iteration when the accompanied person is stop (step_decrease_velocity_stop_robot_conf).



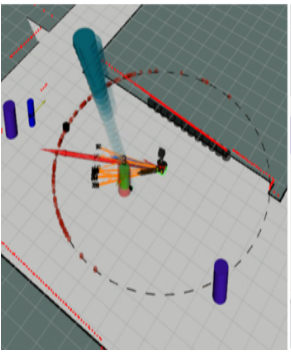
10. How to change the planner parameters (local-window, max vertex and step size)

We can change different parameters that customize the size of the local-window of the local planner, the size of each step of the paths and the maximum number of vertex to compute all the paths. The time of the local-window limits the area to plan the local paths and can be changed with the parameter horizon_time. The number of vertex augments the number of planned paths for the local planner and can be changed using the parameter number_vertex. The size of each step of the paths has to corresponds with the maximum iteration time to compute all the paths again and can be customized using the parameter set_planner_dt. The optimum values for these parameters taking into account our computer and robot capabilities are horizon_time=5 seconds, number_vertex= 500 or 200, and set_planner_dt=0.2 seconds.

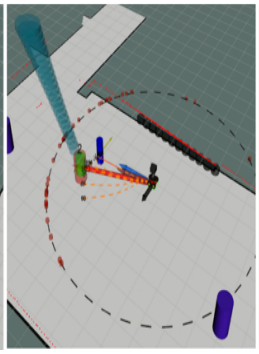


In the next images, we show the differences in the local planner due to change the horizon time and the number of vertex. Changes in the dt can only be appreciated experimentally, seeing that the robot has to continuously correct errors in the planning due to incorrectly computed speeds and acceleration by using an step time between iterations different form the required for this robot.

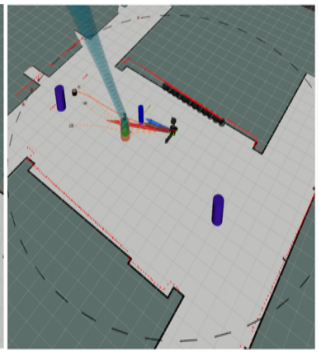
horizon_time= 5 seg
number_vertex= 700



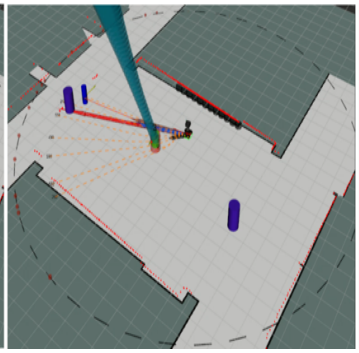
horizon_time= 5 seg
number_vertex= 200



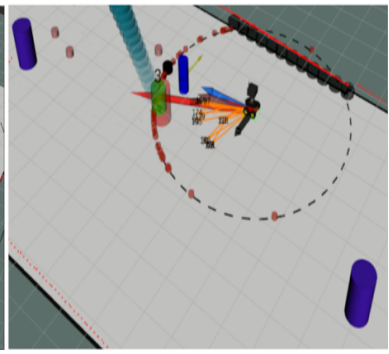
horizon_time= 10 seg
number_vertex= 200



horizon_time= 10 seg
number_vertex= 700



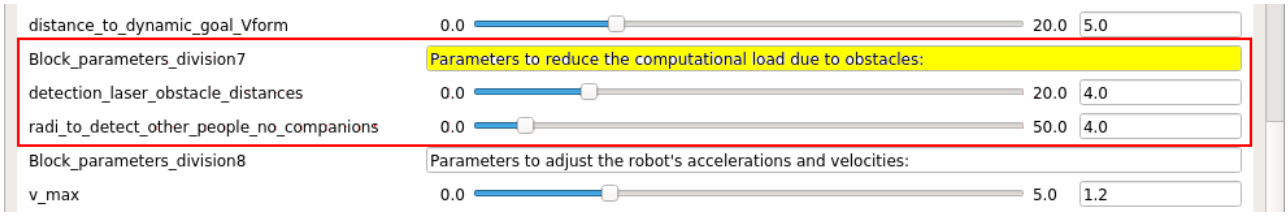
horizon_time= 3 seg
number_vertex= 200



11. How to reduce the radii to detect people and static obstacles

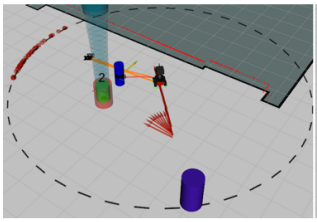
You can reduce the radii around the robot to detect static obstacles and other people. This reduction allows a minimization of the collisions computed, which reduces the computational load of the algorithm.

11.1 Reduce the radii to detect static obstacles: You need to reduce this radii by changing the value of the variable *detection_laser_obstacle_distances* included in the rqt_reconfigure of the companion algorithm.

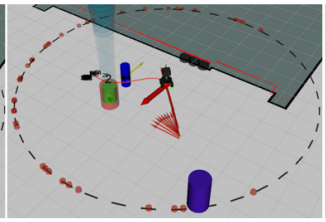


Next images show several examples of how affect the value of the parameter *detection_laser_obstacle_distances* in the distance around the robot position to detect the static obstacles. This distance is in meters. In the image the static obstacles are the black cylinders over the wall of the map of the FME.

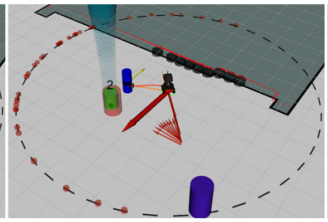
Distance = 1 m



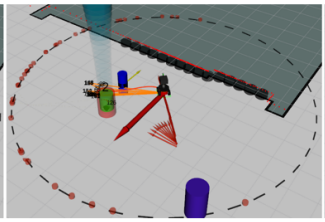
Distance = 2 m



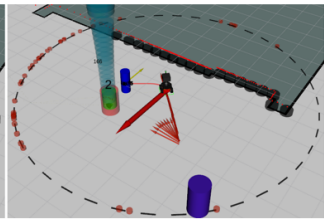
Distance = 3 m



Distance = 4 m

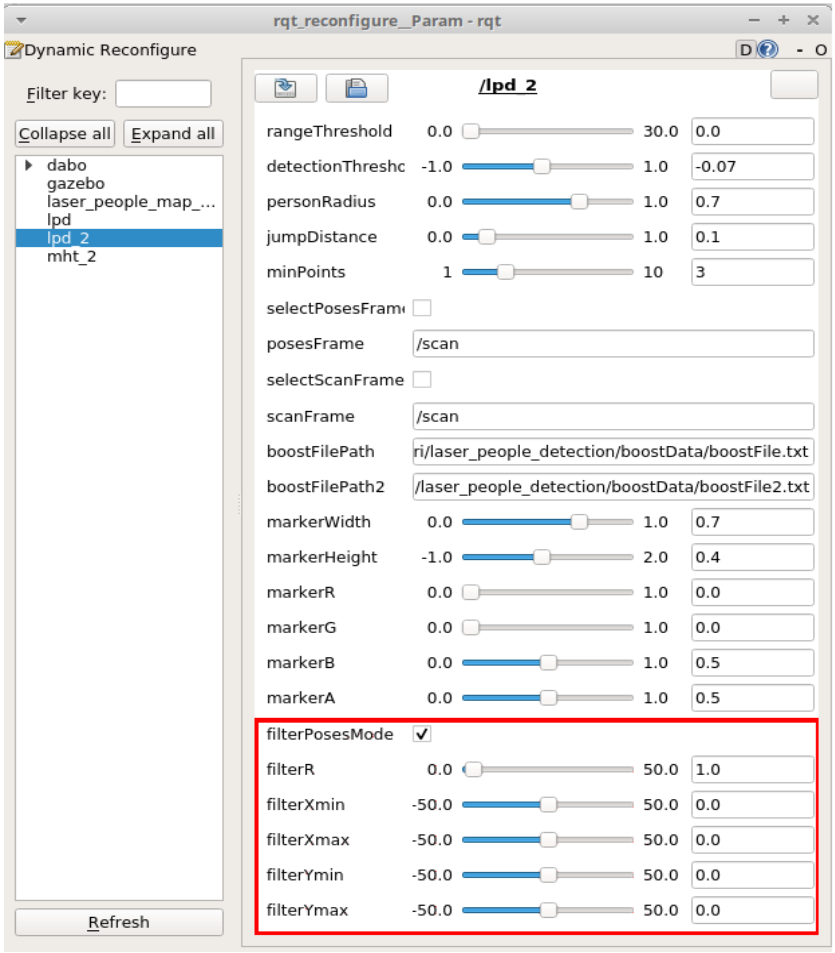


Distance = 5 m



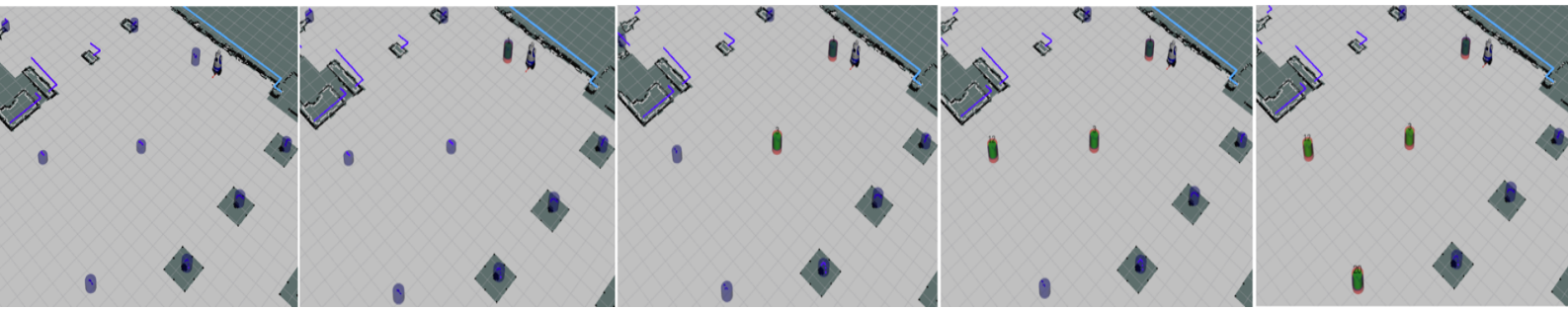
11.2 Reduce the radii to detect other people: The reduction of this radii is better to be reduced directly in the laser-leg-detector, because you reduce at the same time the computational load of the tracker and the planner to do the accompaniment. In the rqt_reconfigure you need to arrive at lpd_2 or lpd (laser people detector). The parameters that you need to change are first the Boolean to allow

the filtering, filterPosesMode=true, and second include which radii you want to filter around the robot by changing the value of filterR. Also, you may filter using different distance in X and in Y by using the parameters filterXmax and filterYmax, respectively.



Next images show several examples of how affect the value of the parameter filterR in the distance around the robot position to track people and use these tracks in the planner to do the accompaniment. This distance is in meters. In the image the people that uses the robot to do the plan are the only ones that have track with identifier associated (in red and green with an id over them). The blue cylinders are all people detection, which are filtered at the output of the node using the value of the parameter filterR. With filterR=50 m you do not filter any thing. With filterR=1 m you filter all the people (also the accompanied one). The most recommended values are filterR=5 m, filterR=10 m or filterR=15 m, to choose one or other depend on the computational power of your system.

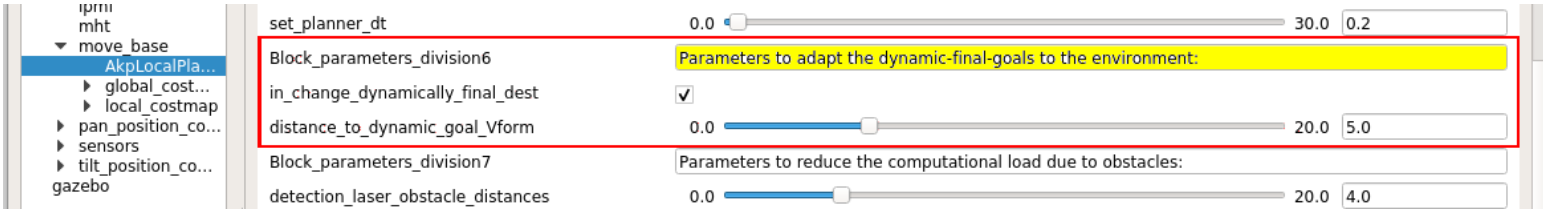
filterR = 1 m filterR = 5 m filterR = 10 m filterR = 15 m filterR = 50 m



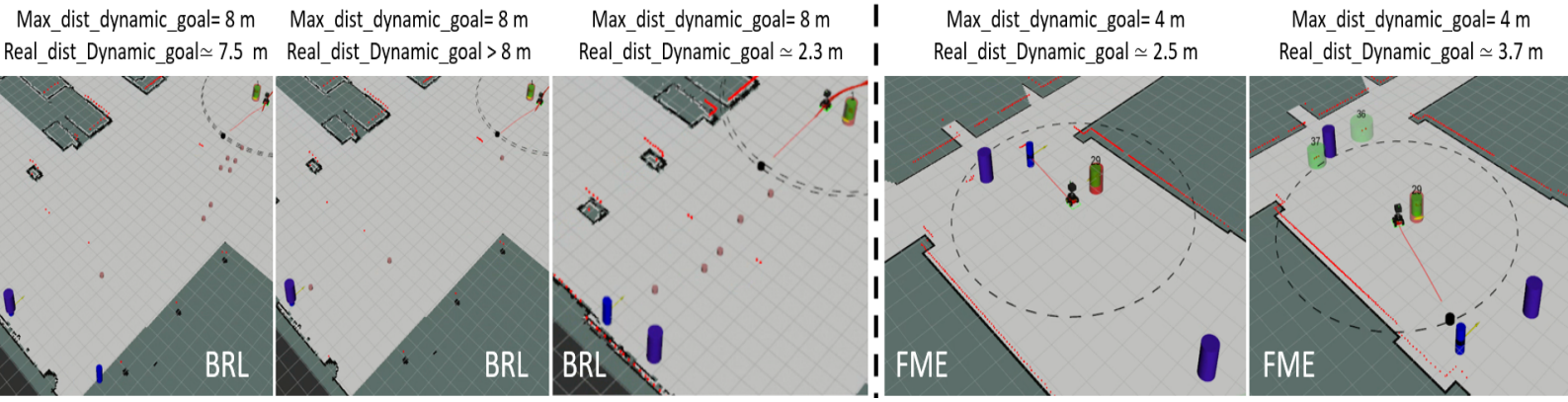
12. Change the dynamic destination parameters to adapt to any environment

We use a dynamic destination created using the static destination of the environment and the direction of the movement of the group (robot and accompanied person). This dynamic destinations allows the robot to do a perfect formation with the person in huge spaces, where people do not go directly to the punctual static destinations of the environment, due to different reasons: these destinations are stairs or entrances to streets or squares, which may have a large area, not a point; the group need to avoid obstacles before arrive to the destination and they do not go always directly to the destination; or maybe there are more destinations not included in our file of all environment destinations.

Then, there are two parameters in the rqt_reconfigure to allow the computation of this dynamic destination or the customization of the dynamic destination to our environment.



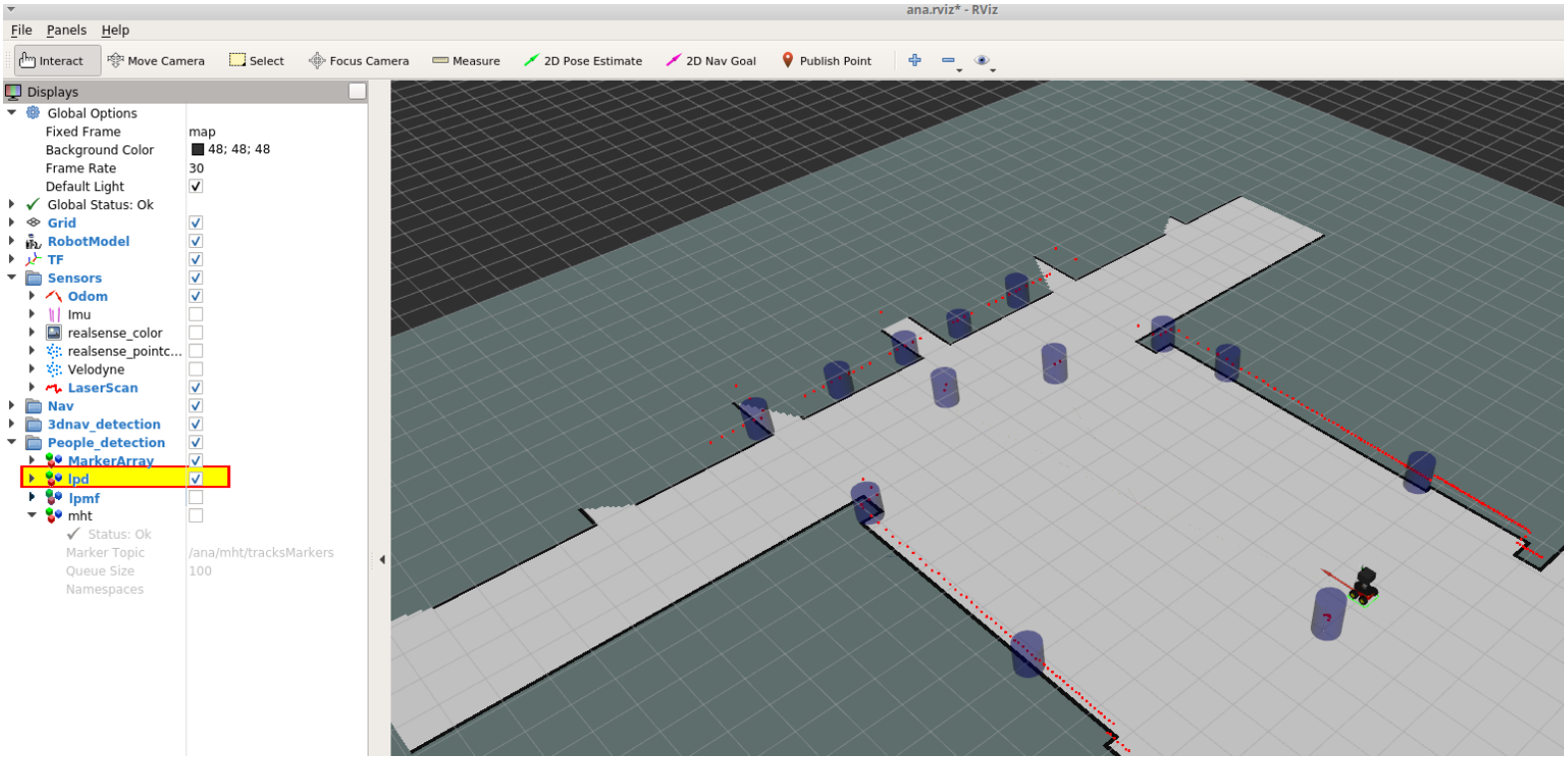
The next images show several examples of how we can use or customize this dynamic destination for different environments (FME and BRL). The Max_dist_dynamic_goal = distance_to_dynamic_goal_Vform and if the needed distance until the dynamic destination is more than 8 m, the robot considers the fixed static goal, like the dynamic one (to do not include collisions of this goal with the walls).



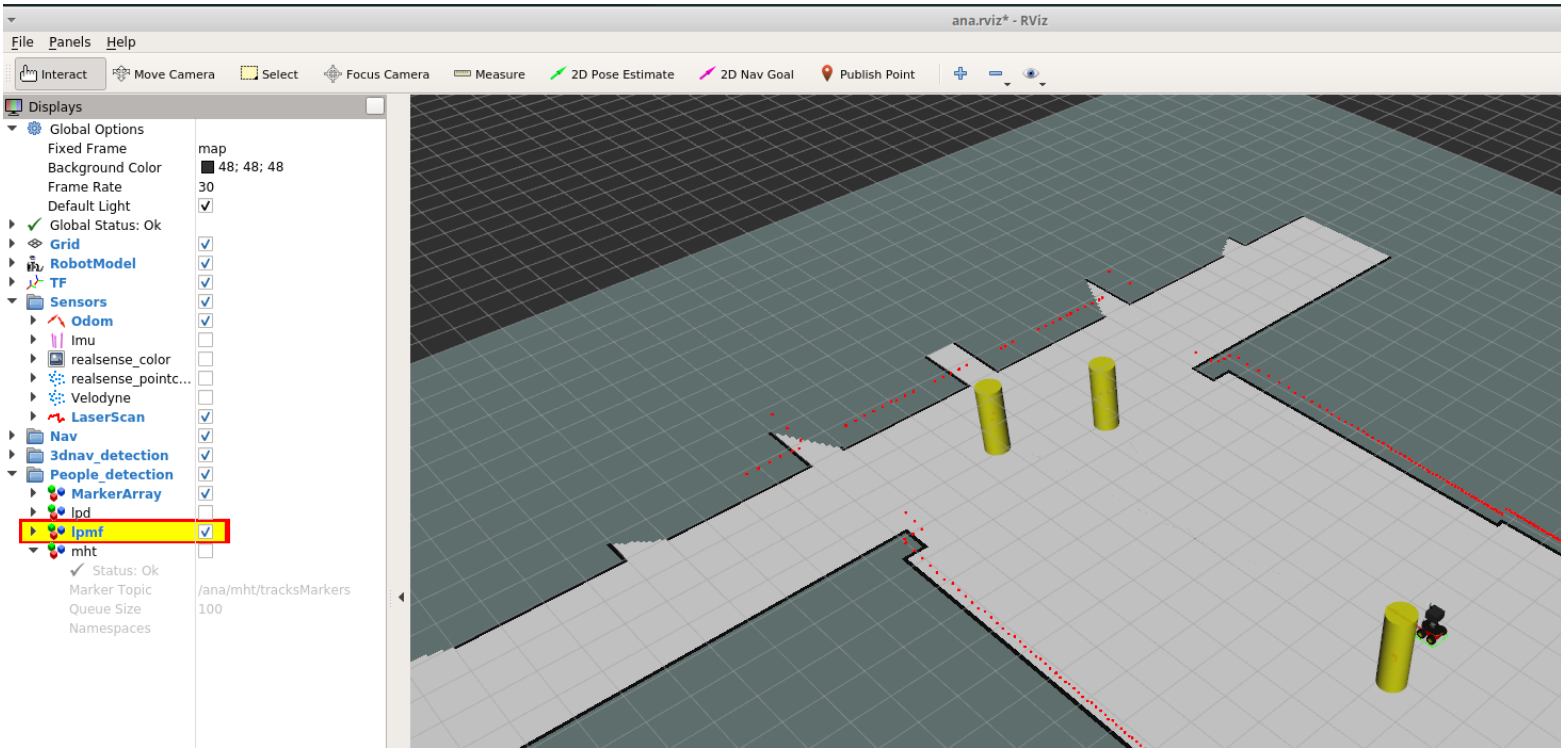
13. How to reduce the number of markers in rviz

You can reduce the number of markers to be seeing in the rviz. This reduction allows to obtain less computational load.

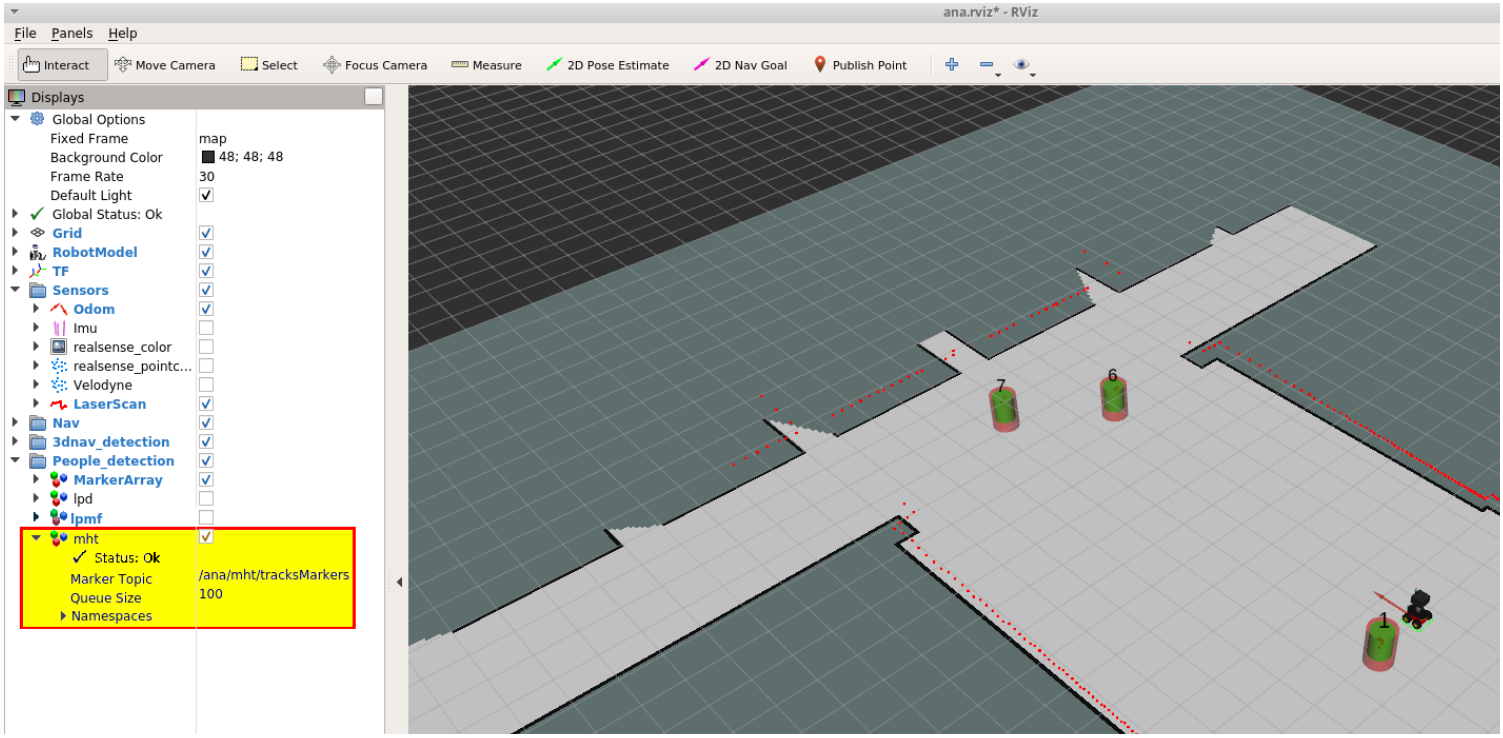
On the rviz you can disable the markers of the lase-people-detector by disabling the flag of these markers. The boolean of these markers is remarked in the next image (lpd). These markers are blue.



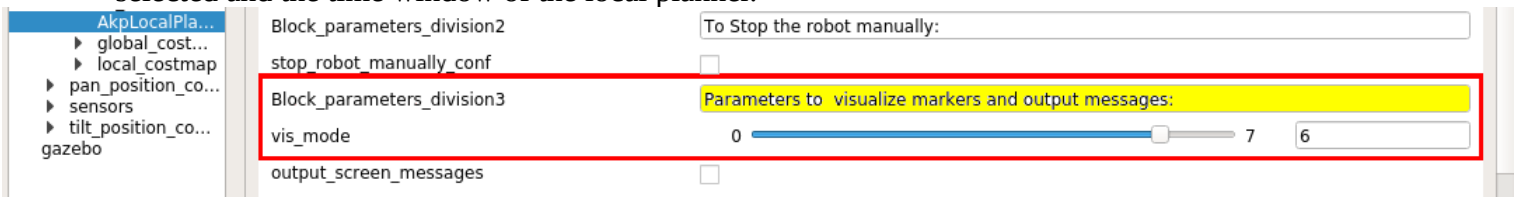
On the rviz you can disable the markers of the people-tracking by disabling the flag of these markers .The boolean of these markers is remarked in the next image (lpmf). These markers are yellow.



On the rviz you can disable the markers of the lase-people-filtering-using-the-map by disabling the flag of these markers, like the next image shows. The boolean of these markers is remarked in the next image (mht).

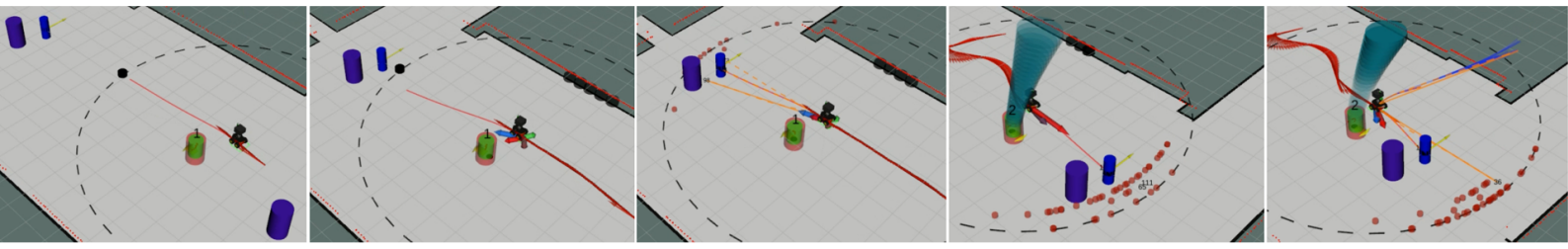


Also, if you reduce the vis_mode on the rqt_reconfigure in the planner used to do the accompaniment, you can reduce the number of markers that this algorithm shows in the rviz. The vis_mode=1 is the recommended mode to has less computational load. Also, this mode shows the minimum needed markers to see the accompaniment behavior: the final destination, the best path selected and the time window of the local planner.



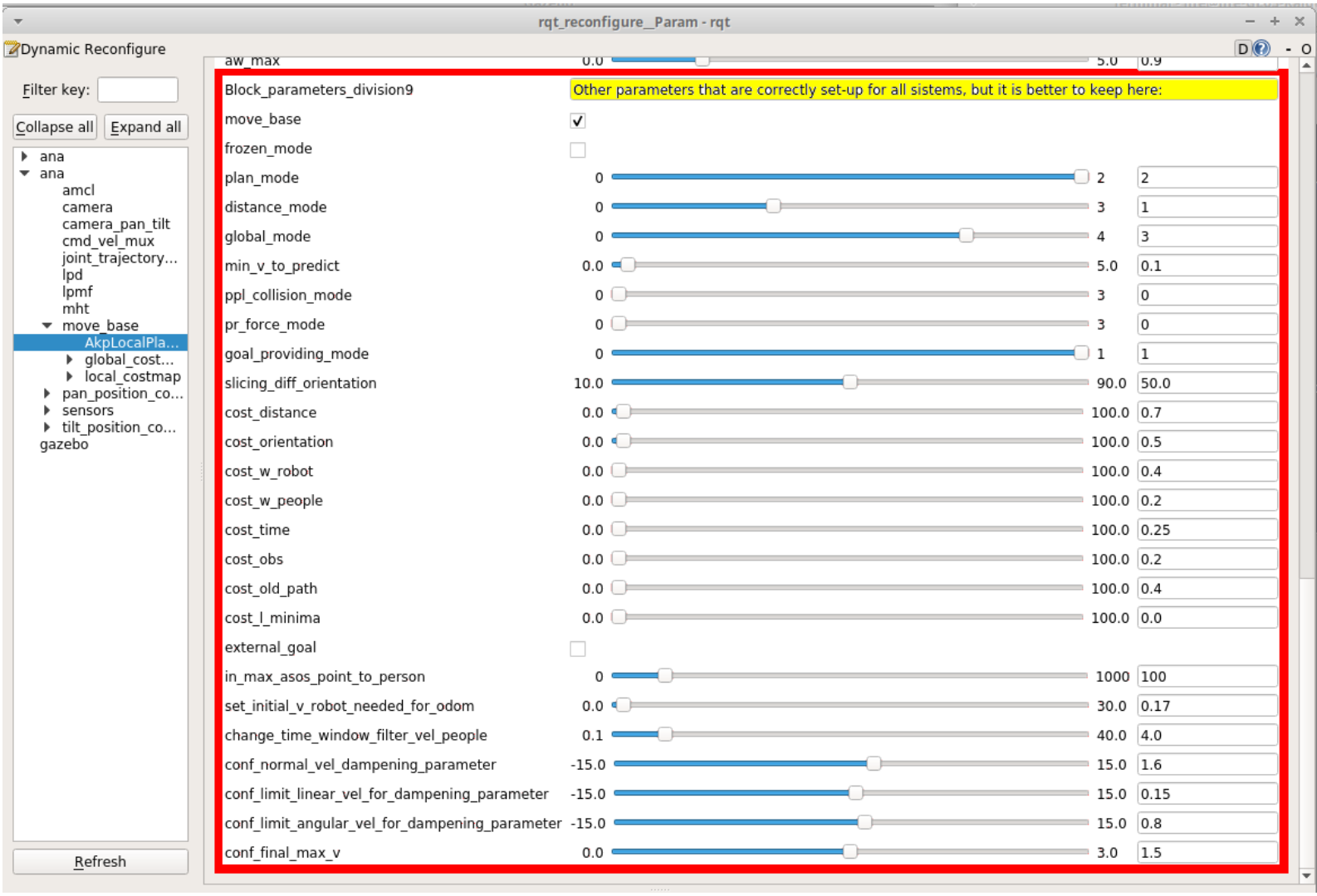
Next, we show the markers that you can visualize for each value of vis_mode. For vis_mode=1, we show the best path, the local-window, the static destinations of the environment and the dynamic destination. In vis_mode=2, are added the markers of the ESFM and the static obstacles. In vis_mode=3, we include all the most feasible paths in 2D. In vis_mode=4, is added all the people predictions (green for the other people and blue for the accompanied one). vis_mode=5 includes the most feasible paths in 3D.

vis_mode = 1 vis_mode = 2 vis_mode = 3 vis_mode = 4 vis_mode = 5



14. Parameters of the rqt_reconfigure that is better to do not change

In the rqt_reconfigure of the planner to do the accompaniment there are other parameters that you can change, but only if you are an “expert” or researcher in the area and you know how to change them.



15. Other configuration parameters of the local-planner and the global-planner that you can change only if you know how to change them well.

In any planner that uses ROS, there are several files to configure the parameters of the local-planner and the global-planner at the beginning of the execution. These files are included in iri_companion_docker_melodic_ana_y_dabo/catkin_ws/src/iri_navigation/iri_robot_aspsi/config/ inside the files: akp_local_planner_params.yaml, costmap_common_params.yaml, global_costmap_params.yaml, local_costmap_params.yaml, move_base_params.yaml. You can

find the concrete files that uses each robot in the launch of the accompaniment. For ana-robot: roslaunch iri_robot_aspsi gazebo_ASPSI_OK_ana_brl.launch and roslaunch iri_robot_aspsi gazebo_ASPSI_OK_ana.launch; for dabo-robot: roslaunch iri_robot_aspsi gazebo_ASPSI_BRL_OK.launch.

Here, we guide you a little in how to change the specific parameters of our local planner that are different for the parameters used in the local-planner and global planner of ROS. For the parameters of the ROS-planners the reader is referred to the tutorials of the ROS-wiki. The parameters of our planner are inside the file: akp_local_planner_params.yaml. You can see the content in the nex image.

Content of akp_local_planner_params.yaml file:

AkpLocalPlanner:

// We initially allow that the robot can move. If this parameter is set-up to false, the robot do not move. But is better that you use the parameter to stop the robot manually in the rqt_reconfigure instead this one.

move_base: True

// The best mode to do all the planning for us. In the thesis of Gonzalo Ferrer of the AKP, they test several types of planning modes, and also this is included and if you change this number you can test the other types of planning modes. But, we tell you that for us the best mode is the selected one.

plan_mode: 2

// The best mode to compute collision distances for us. In the thesis of Gonzalo Ferrer of the AKP, they test several types of collision distances, and also this is included and if you change this number you can test the other types of collision distances. But, we tell you that for us the best mode is the selected one.

distance_mode: 1

// The same than in the other cases, there are different types of global_mode that you can test it if you want. But, we tell you that for us the best mode is the selected one:

global_mode: 3

// The initial vis_mode. Remember that this two parameters can be changed in the rqt_reconfigure also.

vis_mode: 1

frozen_mode: False

// The initial number of vertex and horizon time for the local planner. Remember that this two parameters can be changed in the rqt_reconfigure also.

number_vertex: 200

horizon_time: 5.0

// The cost used to evaluate all the paths for the different characteristics that we evaluate (distance until the final destination, changes in orientation until the final destination, cost to control the robot, cost to avoid people, cost of time, cost to avoid obstacles, cost to be consistent in time because consecutive iterations always has paths very near to not change abruptly, cost of minimal distance for the paths.). Note that the companion cost is not here, we can change with the rqt_reconfigure directly:

cost_distance: 0.7

cost_orientation: 0.5

cost_w_robot: 0.4

cost_w_people: 0.2

cost_time: 0.25

cost_obs: 0.2

cost_old_path: 0.4

cost_l_minima: 0.0

// The limits of robot's velocities and accelerations:

v_max: 1.2

w_max: 1.0

av_max: 0.3

av_max_neg: 0.4

av_Vrobotzero: 0.6

lim_Vrobotzero: 0.1

av_break: 0.4

aw_max: 0.9

// The radii of the robot platform:

platform_radii: 0.5

// The tolerance in x and y to confirm that the robot arrived to the final goal:

xy_goal_tolerance: 0.3

// The distance to start stopping near the final goal:

distance_to_stop: 2.0

// The tolerance in velocity to confirm that the robot arrived to the final goal:

v_goal_tolerance: 0.1

// ESFM constants of the repulsive force between the robot and tall other people in the environment (marker of green arrow):

esfm_to_person_A: 5.05

esfm_to_person_B: 1.2

esfm_to_person_lambda: 0.25

// ESFM constants of the repulsive force between the people and the robot (marker of purple arrow):

esfm_to_robot_A: 5.04

esfm_to_robot_B: 1.2

esfm_to_robot_lambda: 0.25

// ESFM constants of the repulsive force between the people or robot and the static obstacles (marker of black arrow):

esfm_to_obstacle_A: 3.5

esfm_to_obstacle_B: 0.68

esfm_to_obstacle_lambda: 1.0

// ESFM constants. These two are common for the three before forces of the ESFM.

esfm_k: 2.3

esfm_d: 0.2

// minimum velocity of people tracks to do people prediction. Thinking that the person is moving. With this limit we filter the movements due to the noise in the people detection and tracking:

min_v_to_predict : 0.1

// The best mode to detect collisions for us. In the thesis of Gonzalo Ferrer of the AKP, they test several types of collisions, and also this is included and if you change this number you can test the other types of collision computations. But, we tell you that for us the best mode is the selected one:

ppl_collision_mode : 0

// The best mode to compute the forces of the SFM for us. In the thesis of Gonzalo Ferrer of the AKP, they test several types of computation of the SFM, and also this is included and if you change this number you can test the other types of SFM computations. But, we tell you that for us the best mode is the selected one:

pr_force_mode : 0

// The same than in the other cases, there are different types of goal providingt you can test it if you

want. But, we tell you that for us the best mode is the selected one:

goal_providing_mode : 1

slicing_diff_orientation : 50.0

// ESFM constants of the repulsive force between the robot and the accompanied person (marker of green arrow between these two):

esfm_to_person_companion_A: 4.05

esfm_to_person_companion_B: 0.61

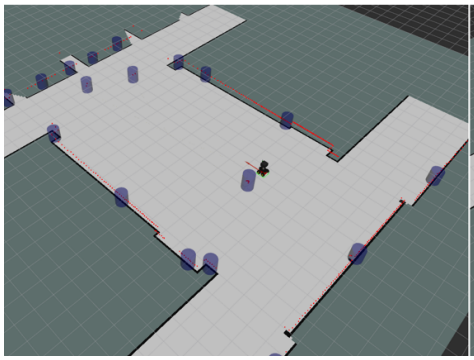
esfm_to_person_companion_lambda: 0.25

esfm_companion_d: 0.1

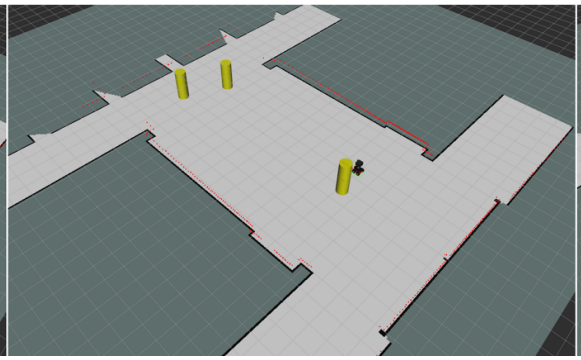
16. Detect and track several people

This two capabilities are from three ROS-nodes iri_laser_people_detection, iri_laser_people_map_filter (to filter the detections with the map) and iri_people_tracking_mht, which allow us to detect people by detecting the two semi-circles of the two legs seen by the 2D laser, filter these people detentions with any map of the environment and allow us to track people represented by the laser detection. These laser detection's are 2D points of the people position over the floor of any environment and the tracker is a multi-hypothesis tracker of detections represented by any 2D points in any 2D space. Next, we include an image that shows the outputs of each of these nodes represented by markers in the rviz. Also, we output detection and track messages with these ones to be used by other nodes.

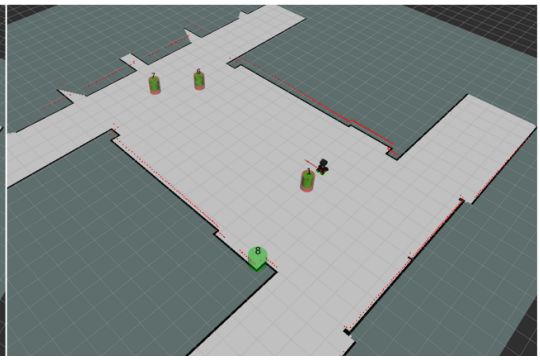
Markers iri_laser_people_detection



Markers iri_laser_people_map_filter



Markers iri_people_tracking_mht



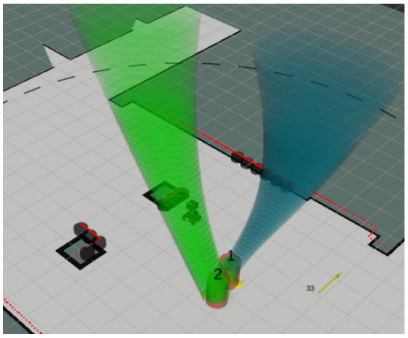
17. Predict people

This capability is embedded inside the node of accompaniment, but it uses different functions apart from the functions of the accompaniment and can be used independently if you want. The location of these functions is inside the c++ library, iri_companion_docker_ana_y_dabo/catkin_ws/src/labrobotica/people_prediction/src:

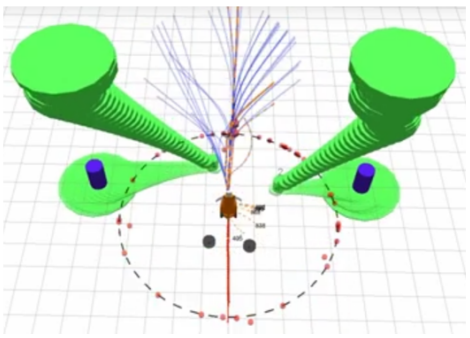
- prediction_bhmip.h and .cpp
- prediction_behavior.h and .cpp
- scene_elements/person_bhmip.h and .cpp
- scene_elements/person_behavior.h and .cpp

There are an example of how to use these functions to predict people behavior in *iri_companion_docker_ana_y_dabo/catkin_ws/src/labrobotica/people_prediction/src examples/prediction_example.cpp.* Also, you can explore the main function to do the people accompaniment (iri_companion_docker_ana_y_dabo/catkin_ws/src/labrobotica/people_prediction/ src/nav/plan_local_nav.h and .cpp) or to simulate an accompanied person (iri_companion_docker_ana_y_dabo/catkin_ws/src/labrobotica/people_prediction/src/nav/ plan_local_nav_person_companion.h and .cpp), and see how we use the prediction in our planning algorithms. Next, we include an image to see the markers of this people prediction that show the reader the type of prediction that we use in time and space, which also includes interactions between people and obstacles of the environment using the ESFM. For further information see Ferrer and Sanfeliu. Pattern Recognition Letters, 2014; Ferrer and Sanfeliu ICRA2014; and Ferrer and Sanfeliu Autonomous Robots 2019. Web page of the akp, where you can find several videos that include people predictions and the planner which is the basis of our companion planner: www.iri.upc.edu/groups/lrobots/akp/ .

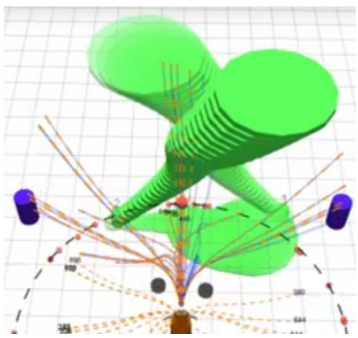
Predictions of people in normal situations when they walk far from the goal.



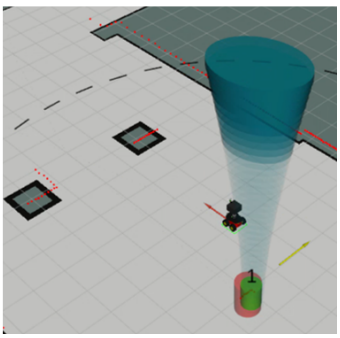
People predictions when people are reaching their final destinations.



People predictions in people crossing situations.



Person prediction when the person is stop.



18. Possible problems external to the proper behavior of the system and how to solve them

18.1 Problems due to computational cost overload:

In the system that uses the Dabo-robot, you can see this problematic behavior. In this case it is due to something wrong in the tf's of the robot model. If you see the messages when you launch the node, you can see how the tf's transformations take more time than needed.

IMPORTANT: Take care of this problem, because it may cause collisions with objects or people.

18.2 section describes a solution to avoid these collisions to ensure security when you use the system in a real robotic platform.

Sometimes, due to the mentioned computational cost overload, the robot may show a "S" navigation behavior which is not appropriate.

If you want to remove it, you could do the following:

18.1.1. If the problem is due to reduced computation capabilities of the Docker/virtual-machine:

Use the system directly on your Ubuntu operating system (you have not to use the docker/virtual-machine). To install the system on your Ubuntu, you need to adapt to your computer the commands included in iri_companion_docker_melodic_ana_y_dabo/Dockerfile.

18.1.2. If the problem persists and it is due to a high number of static or dynamic obstacles: You have several solutions for this problem. Let us go to see some of them.

> **18.1.2.1.1. Static obstacles, how to reduce the robot's radius that is used for detecting static obstacles in the rqt_reconfigure:** We detect the obstacles close to the robot using a circle centered in the robot geometric position. This circle is the range-laser detection from the robot to the obstacles. You can reduce this radius by decreasing the value of the variable detection_laser_obstacle_distances in the rqt_reconfigure. This distance is set up in meters. For more details, see the Document: ASP-SI_Tutorial_Capabilities_Document.pdf.
>
> **18.1.2.1.2. Static obstacles, how to reduce the computational cost in the detection of static obstacles (a second solution) (in this case, it implies additional code from your side):** Also, you can change the detector of the static obstacles by another more efficient detector. The present detector in the simulator, detects an obstacle as a group of small cylinders instead of an unique obstacle, which from the computational point of view could be not as efficient as if the detector detects only the unique object.
>
> **18.1.2.2. Dynamic obstacles (people), how to reduce the distance in rqt_reconfigure**: Reduce radius in the meters around the robot position to detect people, by reducing the value of the variable radi_to_detect_other_people_no_companions in the rqt_reconfigure. Sometimes, it is better to reduce it directly in the detector. In this case, you have to go to the lpd page in the rqt_reconfigure and reduce the value of meters of the variable filterR and set up filterPosesMode=true.

18.1.3. If the problem is directly the computational cost to generate all the paths: You can reduce the local window of the local planner and the number of vertex of all the paths. In the rqt_reconfigure, reduce the value of the horizon_time to diminish the size of the local window, and in the rqt_reconfigure, reduce the value of the number_vertex to diminish the number of vertices of all the paths. By default, these parameters are set up to the minimum value for a good robot behavior, horizon_time=4 segs and number_vertex=200. If your system can not handle these minimum values, you need a more powerful computer to do the simulations. The default parameters used in all our real experiments are: horizon_time=5 segs and number_vertex=500. We test the simulator in an Intel Core 2 Quad CPU @ 2.66 and 3.00 GHz, but if you need additional information, please refer to the publication of Repiso et al. IJSR2019.

18.1.4. If the problem is due to the high number of markers to visualize in Rviz: You can solve it by disabling the markers that you do not need to see: detector, detector-filtered by the map, etc. Also, you can reduce the visualization markers of the planner that implements the accompaniment diminishing the value of the variable vis_mode in the rqt_reconfigure.

18.2. Problems of robot proximity to any person due to system overload (this may cause collisions): If the system has a high computational load and the controller can not meet its desired rate, then the controller does not control the robot in real time. Then, it is good for safety reasons to include a new node that stops the robot at a safety distance, for example 0.3 cm. In our real-life system, we use a ROS-node that does this behavior, but the version of this node is not included in the system due to delays on the migration to ROS-melodic.

18.3. Problems of collisions of the dynamic final destination with the map's obstacles: When the simulator is computing the path of the accompaniment group (robot and person) to the final destination and then there are people interfering with the path or in general when there are obstacles that do not allow that the group reaches the final destination in a direct way, then the simulator has to create a “temporal final destination”. This new “temporal final destination”, which is denominated as “dynamic final destination”, is close to the final static destination and is separated by a specified distance that can has to be defined (see the article E. Repiso IROS2019 for additional explanation). The present simulator has already a default distance value that can be modified. The global ROS planner is very sensitive to this parameter, and can stop the robot’ planner path if it detects that it can not reach the final destination for the aforementioned reasons. Solutions: (1) disable the dynamic goal computation; or (2) reduce the distance that separates the dynamic final destination from the environment's final static destination. To disable the dynamic final destination computation do the following: in the rqt_reconfigure, change to false the boolean in_change_dynamically_final_dest. To reduce the dynamic goal's distance from the final static destination do the following: in the rqt_reconfigure, you have to reduce the value of the variable distance_to_dynamic_goal_Vform. This distance is in meters.

18.4. The robot movement is not smooth: We have a set-up of robot’s velocities and accelerations that could be different from your robot. Because your robot has other characteristics of mass, robot’s controller, robot’s dimensions, etc., you have to customize the linear, angular velocities and accelerations of the method. This information is included in the rqt_reconfigure, under the label “Parameters to adjust the robot's accelerations and velocities”.