**Launch file**

1. **Launch file for Process 1 (generation of .pbstream file)**

We used the launch file [test.launch](https://drive.google.com/file/d/13eyuRwFGDOjxWOOM-QIwOYEixDM0wAwV/view?usp=sharing) for the first process of generation of .pbstream file. The settings for this launch file were made using insights from the working configurations as uploaded by a user [here](https://drive.google.com/file/d/0B1KZT92BcdVNaHdkZVp5bkI0WDQ/view).

* The first change made to this was changing the cartographer\_ros package node from type="cartographer\_node" to type="cartographer\_offline\_node". This was done using insights from the default [offline\_backpack\_3d.launch](https://github.com/googlecartographer/cartographer_ros/blob/master/cartographer_ros/launch/offline_backpack_3d.launch) in Google Cartographer install\_isolated folder/launch folder. Using cartographer\_ros node does not generate .pbstream file and only gives the visualization in Rviz.
* Again, using insight from the same file, the lines -urdf\_filenames $(find cartographer\_ros)/urdf/backpack\_3d.urdf and -bag\_filenames $(arg bag\_filenames)" as done in the file. **(THIS MIGHT BE REDUNDANT).**

Drawing from the same, we also added

<node name="cartographer\_occupancy\_grid\_node" pkg="cartographer\_ros"

type="cartographer\_occupancy\_grid\_node" args="-resolution 0.05" />

* Remap was done in the cartographer\_ros package node to incorporate the names of the topics being published in our case.

<remap from="imu" to="imu/pixhawk"/>

<remap from="points2" to="velodyne\_points" />

The names on left correspond to the default topic IDs’ expected by Cartographer and the ones on the right correspond to the ones that are being published in our case.

* We removed the commands from the Rviz node as it was not required in our case.

<param name="robot\_description"

command="$(find jackal\_description)/scripts/env\_run

$(find jackal\_description)/urdf/configs/robot

$(find xacro)/xacro

$(find jackal\_description)/urdf/jackal.urdf.xacro" />

* We changed

<param name="robot\_description"

textfile="$(find jackal\_description)/urdf/jackal.urdf.xacro" />

to

<param name="robot\_description"

textfile="$(find cartographer\_ros)/urdf/backpack\_3d.urdf" />

for doing it for our case.

* We removed

<node name="tf\_remove\_frames" pkg="cartographer\_ros" type="tf\_remove\_frames.py">

<remap from="tf\_out" to="/tf" />

<rosparam param="remove\_frames">

- map

</rosparam>

</node> -->

Drawing from the same as well as the insights from the fact that this was not present in the default [offline\_backpack\_3d.launch](https://github.com/googlecartographer/cartographer_ros/blob/master/cartographer_ros/launch/offline_backpack_3d.launch) file, we also removed this:

<node name="playbag"

pkg="rosbag"

type="play"

args="--clock $(arg bag\_filename)" />

**Some tips and tricks**:

* The processing time depends on the size of your bag file. It’s better to split the bag file into smaller parts so as to quickly test and assess.

This can be done by using the command:

rosbag filter input.bag output.bag "t.secs <= 1284703931.86"

or

rosbag filter input.bag output.bag "t.secs >= 1531425960 and t.secs <= 1531426140"

The time mentioned here is the ROS time and can be obtained from the bottom left corner in the Rviz window launched during process 1. Make note of the start time when the ROS time when the process starts and then calculate the input to this parameter by adding the start ROS time the required number of seconds in the output.bag file i.e. if you want the output.bag file to contain 100 seconds and the ROS start time is x, the parameter value should be x+100.