PNC::obstacle_stop_planner::about calculate obstacle #2503

Unanswered   **JokerWhy233** asked this question in **Q&A**

JokerWhy233 on Mar 31, 2022

In obstacle_stop_planner module , we sub"
/perception/obstacle_segmentation/pointcloud" and"
/perception/object_recognition/objects"   when we find the nearest object,
We first use the point cloud information to find the collision point, and then
find matching Object in " /perception/object_recognition/objects"   Why not
directly use" /perception/object_recognition/objects" for collision detection
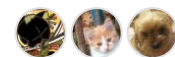and find corresponding obstacles

↑ 1

Category

🙏 Q&A

Labels

None yet

3 participants

---

0 comments · 2 replies

Oldest   Newest   Top

kenji-miyake on Mar 31, 2022

@yukkysaito @TakaHoribe Could you answer this or mention appropriate
people? 🙏

↑ 1   0 replies

TakaHoribe on Apr 22, 2022   Maintainer

@JokerWhy233
The current design concept is like below.

First, these two types of information have the following characteristics:

- Dynamic Object (`/perception/obstacle_segmentation/pointcloud`):
  has time-series information through clustering and tracking, allowing
  estimation of velocity. On the other hand, positional accuracy is
  generally poor compared to the `Object Segmented Points` due to the
  performance of clustering algorithms.
- Object Segmented Points (`/perception/object_recognition/objects`):
  is the sensor point cloud information segmented as an obstacle, and the
  position information is very accurate. On the other hand, speed
  estimation is not that easy.

To efficiently use these advantages, the `obstacle_stop_planner` first calculates the position where the planned-velocity is inserted from the `/perception/obstacle_segmentation/pointcloud`, and then the velocity to be inserted is calculated using `/perception/object_recognition/objects`.

Note that it might be not ideal, and should be discussed for the global design with sensing/perception/planning components.

↑ 1   0 replies