FMU and Proposed Solution

Introduction

https://fmi-standard.org/

https://fmi-standard.org/docs/3.0-dev/#fmi-for-co-simulation

FMU version : 2.0

FMI for Model Exchange

The goal of the Model Exchange interface is to numerically solve a system of differential, algebraic and discrete-time equations

FMI for Co-Simulation

Other questions

- How does VTD control vehicle (change throttle/steering)
    - Is vehicle follow the waypoint pathshape to move?
    - How to define and get baseline (real-time position) of vehicle? (Localization)
    - How does VTD define steering angle/ throttle?

▼ How vehicle change the pathshape to switch another waypoint path?

The controller will be introduced to implement this function.

How controller(FMU) interference with other components (VTD & VSM)?

▼ Is obstacle defined in VTD?

- If it is, how the bus detect whether there is obstacle?
- Is obstacle has pose (x,y,z) or waypoints(trajaceny) to follow?
- If we need to set a controller to detect whether there is a obstacle, the controller should be inside or outside VTD?

Solution

Edit scenior xml file to predefine the path of bus and obstacle position (New player1).

For New player1, position is fixed in original postion and speed equals to 0.

- Use Buggy Code

Buggy Code

There are three modules are used for our project :

> lidar_localizer

> waypoint_follower

> waypoint_maker

lidar_localizer

▼ **ndt_mapping**

- Inputs : real-time pointcloud data.
- Output : pcd files (stored in .ros)

approximate_ndt_mapping is different version of ndt_mapping which will generate sub-maps and clear memory continuously.

▼ **ndt_matching**

- Input : compare real-time pointcloud with pcd files
- Output : **POSE**

▼ **pcd_map_saver**

- Input : pointcloud data
- Output : pcd files (stored in .ros)

Waypoint _maker

▼ waypoint_loader

- Input : waypoint data
- Output : pcd files (stored in .ros)

▼ velocity_replanner

Change velocity

Waypoint _follower

▼ pure_pursuit_core

Adjust speed(linear.x) and steering angle(angular.z)

Obstacle Detection

Buggy use pointcloud data to analyze obstacle's position. With waypoints, we can find stopline to stop/ change speed of vehicle.

After successfully build these three modules, we can

```
# Run ros nodes
rosrun lidar_localizer ndt_mapping

# Run launch files
cd src/CustomizedBuggy/lidar_localizer/launch/
roslaunch buggy.launch
roslaunch ndt_mapping.launch
```

Open Questions

If we want to build ros code in fmu, do we need to put all things inside fmu creation project?