MEKF_VDPL LOCALIZATION ALGORITHM USER GUIDE

This manual has been prepared for the proper simulation implementation of the MEKF_VDPL localization algorithm using the Turtlebot3 mobile robot in the ROS Gazebo and Rviz environments.

1. Open the MEKF_map.yaml file and replace "ozan" with the name of your computer as follows. After making this change, put MEKF_map.pgm and MEKF_map.yaml files in the HOME folder. If this method does not work, you can create your own map using the Gmapping package.

```
image: /home/ozan/MEKF_map.pgm
resolution: 0.050000
origin: [-10.000000, -10.000000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

2. Install the turtlebot3 package inside the catkin_ws/ folder. This package contains the AMCL algorithm as the localization algorithm. In order to use the MEKF_VDPL algorithm we need to disable the AMCL algorithm. For this purpose, the following files should be opened in order and the AMCL-related code in the turtlebot3_navigation.launch file should be commented as follows:

   HOME-→catkin_ws-→src-→turtlebot3-→turtlebot3_navigation-→launch-→ turtlebot3_navigation.launch

```
<!-- Map server -->
<node pkg="map_server" name="map_server" type="map_server" args="$(arg map_file)"/>

<!-- AMCL -->
<!-- include file="$(find turtlebot3_navigation)/launch/amcl.launch"/ -->

<!-- move_base -->
<include file="$(find turtlebot3_navigation)/launch/move_base.launch">
  <arg name="model" value="$(arg model)" />
</include>
```

3. It is recommended to change some parameters within the DWA planner as follows: HOME-→catkin_ws-→src-→turtlebot3-→turtlebot3_navigation-→param-→ dwa_local_planner_params_waffle_pi.yaml

```
DWAPlannerROS:

# Robot Configuration Parameters
  max_vel_x: 0.15
  min_vel_x: -0.15

  max_vel_y: 0.0
  min_vel_y: 0.0
```

The velocity when robot is moving in a straight line
  max_trans_vel: 0.26
  min_trans_vel: 0.13

  max_rot_vel: 1.0
  min_rot_vel: 0.1

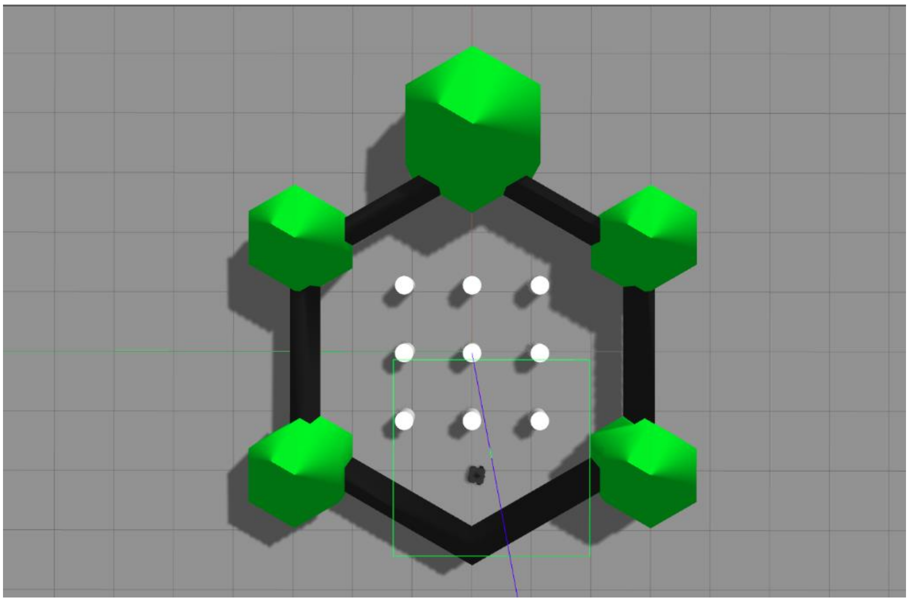
  acc_lim_x: 2.5
  acc_lim_y: 0.0
  acc_lim_theta: 2

** After following the above instructions exactly, please follow the instructions below for the navigation application in ROS environment using the MEKF_VDPL algorithm.

I. Open a new terminal and enter the following codes in order.

```
export TURTLEBOT3_MODEL=waffle_pi
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

After entering these codes in the terminal, a platform like the following will open in the Gazebo environment

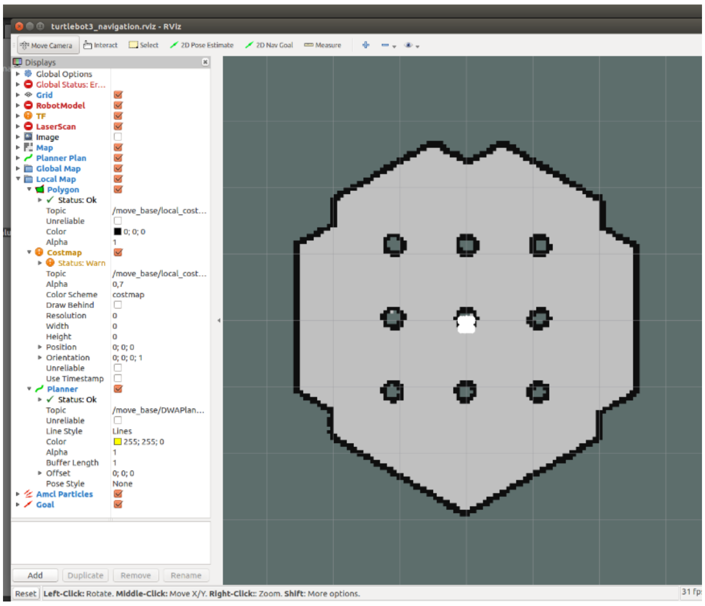


Turtlebot3 robot model can shift from its initial position after some time has passed. To bring it back to its initial position, press Ctrl + R keys together.

II. Open a new terminal and enter the following codes in order.

```
export TURTLEBOT3_MODEL=waffle_pi
roslaunch turtlebot3_navigation turtlebot3_navigation.launch
map_file:=$HOME/MEKF_map.yaml
```

After entering these codes in the terminal, a platform like the following will open in the Rviz environment

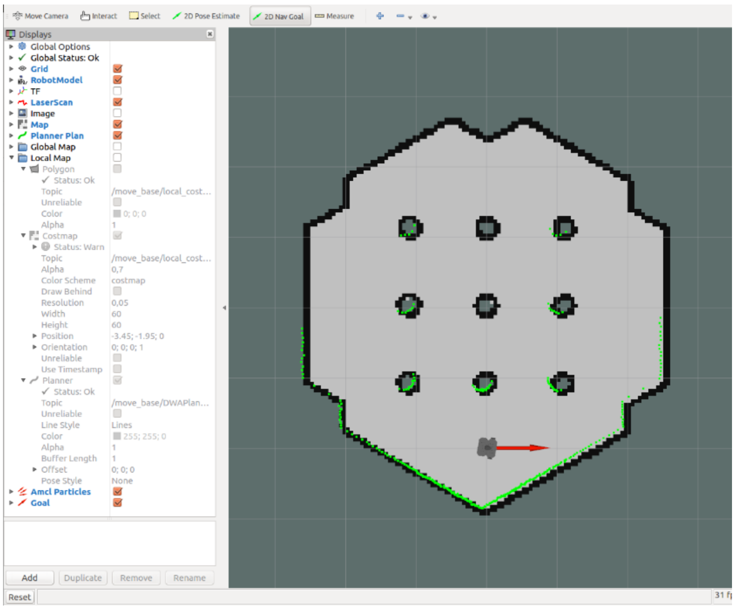


III. Run the MATLAB script in the folder (MEKF_VDPL_Pose.m [1])

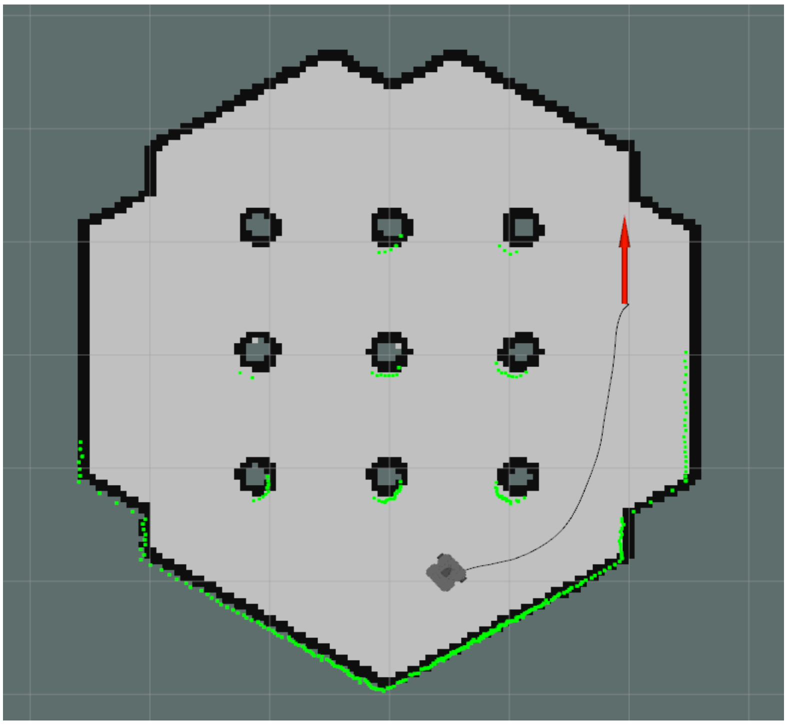
IV. Inside the catkin_ws folder put the map_to_odom pyhton node and open a new terminal and enter the following code.

```
cd catkin_ws/
python map_to_odom_transform_publisher.py
(python3 map_to_odom_transform_publisher.py)
```

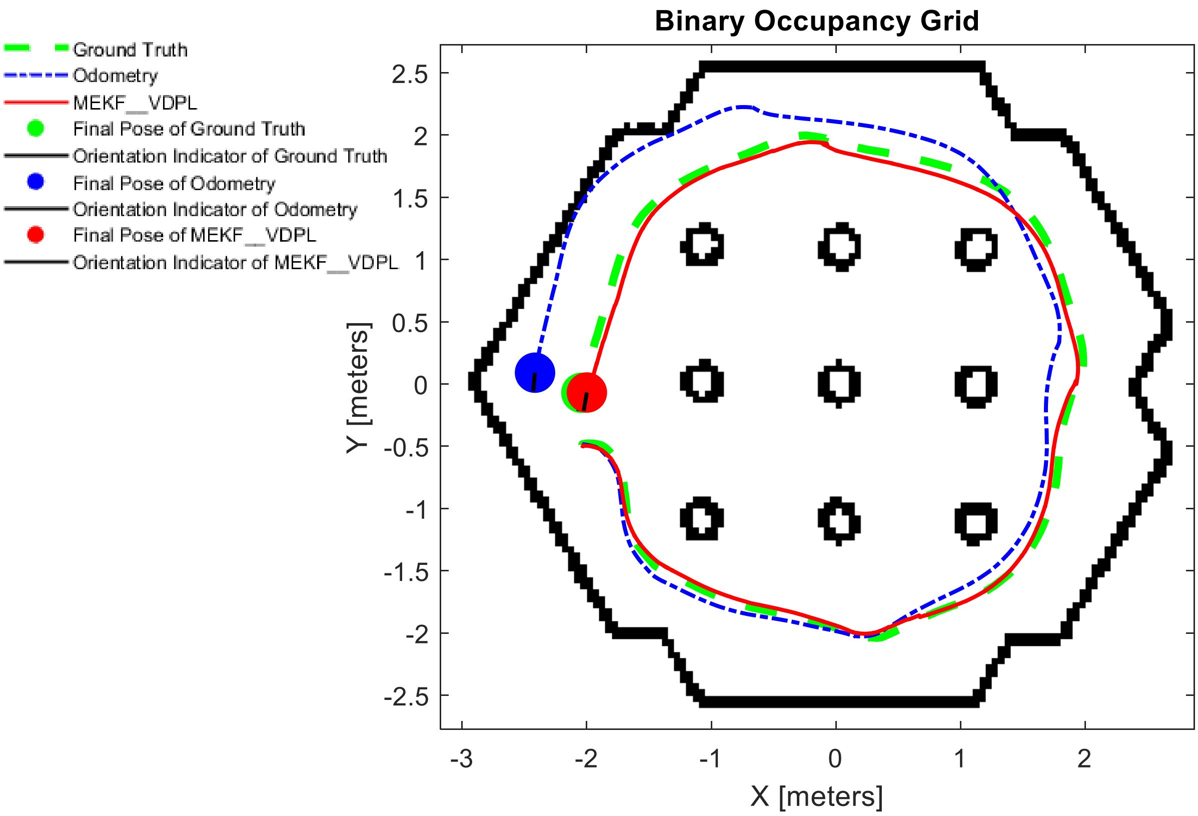
After the Python code runs, the Rviz environment will look like the following.



Giving a goal in the Rviz environment, it can be ensured that the robot goes to the goal in fully autonomous as shown in the figure below.



Given a target at certain points, the route followed by the robot with the MEKF_VDPL algorithm using the erroneous velocity data is given below. With Odometry, localization is done using only the erroneous velocity data and there is no correction.



REFERENCES

[1] O. V. Altinpinar and V. Sezer, “A novel indoor localization algorithm based on a modified EKF using virtual dynamic point landmarks for 2D grid maps,” Robotics and Autonomous Systems, vol. 170, (2023): 104546.