

Pekauto Test Task Documentation
Hamid Mahoodabadi
Mahmoodabadihamid@gmail.com

Introduction

The GNSS (Global Navigation Satellite System) module is a collection of functions designed to process and analyze raw GNSS data. It includes tasks such as parsing raw GNSS data, calculating the projection of GNSS points onto a moving plane, determining vehicle heading, and visualizing the GNSS data and its projection.

Functions definition:

1. 'raw_data_parser' Method

'''

@staticmethod

def raw_data_parser(text_data):

'''

This method is responsible for parsing raw GNSS data provided as text. The input is a string containing comma-separated values, with each line representing a data point. The method returns a list of lists, where each inner list contains the following information for a data point:

1. 'time_s': Time in seconds.
2. 'x_mm': X-coordinate in millimeters.
3. 'y_mm': Y-coordinate in millimeters.
4. 'roll_deg': Roll angle in degrees.
5. 'pitch_deg': Pitch angle in degrees.

2. 'calculate_projection' Method

'''

@staticmethod

def calculate_projection(gnss_data, altitude):

'''

This method calculates the projection of GNSS points onto a moving plane. It takes in the parsed GNSS data as well as an altitude value. The altitude represents the height above the ground plane at which the projections are to be calculated.

For each data point in the GNSS data, the method performs the following calculations:

1. Converts the roll and pitch angles from degrees to radians.

2. Uses trigonometry to calculate the X and Y coordinates of the projection by taking into account the altitude, roll angle, and pitch angle.

The method returns a list of tuples, where each tuple contains the following information for a

data point:

1. 'time_s': Time in seconds.
2. 'projection_x': X-coordinate of the projection.
3. 'projection_y': Y-coordinate of the projection.

3. calculate_heading' Method

'''

@staticmethod

def calculate_heading(gnss_data):

'''

This method calculates the heading of the vehicle based on the GNSS data. Heading represents the direction in which the vehicle is moving. It takes the parsed GNSS data as input.

For each consecutive pair of data points in the GNSS data, the method calculates the heading using the 'atan2' function and converts it from radians to degrees. The heading is the angle between the X-axis and the direction in which the vehicle is moving.

The method returns a list of tuples, where each tuple contains the following information for a

data point:

1. 'time_s': Time in seconds.
2. 'heading_deg': Vehicle heading in degrees.

4. 'visualize_data' Method

'''

@staticmethod

def visualize_data(gnss_data, projections):

'''

This method generates an animated visualization of the GNSS data and its projections using Plotly. It takes in the GNSS data and the calculated projections as input.

The visualization includes the following components:

- A line plot of the GNSS data points (blue).
- A line plot of the projection points (red).
- Animation frames that show the movement of the points over time.

The visualization is displayed in a web browser, and it allows you to play an animation of the vehicle's movement.

Example Usage

After defining the GNSS data in raw format and preprocessing it using the 'raw_data_parser' method, you can perform the following tasks:

1. Calculate projections using 'calculate_projection'.
2. Calculate vehicle headings using 'calculate_heading'.
3. Visualize the GNSS data and projections using 'visualize_data'.

Example Output

The output of this code is an interactive animation that visually represents the movement of the vehicle based on GNSS data and its projections.

Notes

- The code includes error handling to handle exceptions that may occur during calculations.
- Ensure that you have the required libraries ('numpy', 'plotly') installed to run this code successfully.

By using this GNSS module, you can efficiently process and visualize GNSS data for vehicle tracking and analysis.