Take Home Instructions

Objective

Your task is to implement a filtering algorithm to accurately estimate and track the position and velocity of a satellite using both GNSS (Global Navigation Satellite System) data. The solution should demonstrate your ability to apply state estimation techniques, emphasizing best practices in software development, such as code readability, modularity, and integration readiness.

Background

In aerospace engineering, precise tracking and estimation of space objects are crucial for applications such as navigation, remote sensing, and communication. Accurately estimating the position and velocity of a space object allows for better mission planning, collision avoidance, and performance assessment of space missions.

This assignment is designed to assess your proficiency in handling real-world satellite data and applying engineering principles to solve complex problems. We are particularly interested in your thought process, decision-making, and how you translate theoretical knowledge into practical implementations.

Tip: The goal of this exercise is to help us understand your thought process as you go about solving a state estimation problem. Please document any assumptions or decisions made throughout the process. The tasks have been designed around implementing a filtering algorithm to track a satellite position and velocity. Using best practices when it comes to developing code, comments, data manipulation, memory management, type hinting, etc. is encouraged.

Tasks

1. **GNSS Data Assimilation for Satellite State Prediction**
    - Select and implement a filtering algorithm to estimate and track the satellite's position and velocity using the provided GNSS data for a two-week period (May 12th to May 31st, 2024). In your solution, explain your choice of filtering algorithm, along with the reasoning behind the selection of key filter parameters such as the initial state covariance matrix, process noise, and measurement noise. You may use any freely available Python libraries or toolboxes to facilitate the implementation.
2. **Maneuver Detection**

- Implement a maneuver detection algorithm within your filter to detect and flag potential satellite maneuvers. Explain the logic behind your detection criteria.

3. **Satellite State Propagation and TLE Comparison Analysis**
    - Propagate the satellite's state and covariance for an additional three days (June 1st to June 3rd, 2024) without further data assimilation. Compare these propagated states with those generated by TLE data and analyze any discrepancies.

Optional Tasks/Stretch Goals

1. **Filter Parameter Tuning**
    a. Fine-tune the filter parameters to enhance the accuracy and stability of your tracking solution. Using quantitative metrics and qualitative assessments, compare the performance before and after tuning.

Materials

You have been provided with the following resources:

1. **GPS_meas.csv** file containing:
    a. **time** : column containing the timestamp at which the GPS measurement was recorded
    b. **sv** : Satellite ID
    c. **ECEF** : x, y z axis in ECEF frame
    d. **position** : position data in km
    e. **clock**: satellite's clock bias in microseconds
    f. **velocity** : velocity data in decimeters per second (dm/s)
    g. **dclock**: placeholder and can be ignored

Please note that the satellite under consideration is SWARM A, identified by Norad ID 39452. Be aware that the GPS position and velocity data may contain noise and potential corruption.

Deliverables

Your submission should include either a single or a collection of iPython Notebook(s) with the necessary Python scripts.

Please ensure that the iPython Notebook(s) are fully runnable from start to finish without any errors. This means anyone should be able to download your notebook(s), run all the cells sequentially, and reproduce our results.

The notebook should include:

1. **State estimation plots**: Plots illustrating the predicted state and velocity of the satellite, along with their associated covariance.
2. **Measurement residual plots**.
3. **Design documentation**: Explanation of design choices, including selecting filtering algorithms, parameter tuning strategies, and reasoning behind key decisions.
4. **Challenges and Solutions:** Discuss challenges encountered during implementation and strategies to address them.
5. **Additional insights:** Any further observations or insights gained from the implementation process, supported by relevant plots or visualizations. Consider discussing potential improvements, limitations of your approach, or future work that could enhance the solution.