README for “Multiple Model Particle Filter for Traffic Estimation and Incident Detection.”.

Ren Wang and Daniel B. Work

June 10, 2015

Abstract

This document describes the implementation of the multiple model particle filter(MMPF) introduced in the article ’Multiple Model Particle Filter for Traffic Estimation and Incident Detection.” by Wang, Work and Sowers, submitted to the IEEE Transaction on Intelligent Transportation Systems A preprint of the article is available for download on the second author’s website. The source code is hosted at `https://github.com/Lab-Work/MMPF_Traffic_Estimation_Incident_Detection`.

1 License

This software is licensed under the *University of Illinois/NCSA Open Source License*:

Copyright (c) 2014 The Board of Trustees of the University of Illinois. All rights reserved.

Developed by: Department of Civil and Environmental Engineering University of Illinois at Urbana-Champaign `https://github.com/Lab-Work/MMPF_Traffic_Estimation_Incident_Detection`

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution. Neither the names of the Department of Civil and Environmental Engineering, the University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

2 Running the code

The provided .py files can be used to reproduce the results presented in the publication.

1. Download CORSIMOutPutTextFile.zip and CORSIMOutPutTextFile2.zip from the link:

2. Unzip CORSIMOutPutTextFile.zip and copy it to folder: /CORSIM filter factor/

3. Unzip CORSIMOutPutTextFile2.zip and copy it to folder: /CORSIM_FD/

4. Go to the folder: /CORSIM filter factor/PythonCode and running:

   ```
   extract_keyData.py

   extract_trajectory.py

   generate_measurements.py

   generate_trajectory.py

   generate_true_state.py
   ```

5. Go to the folder /CORSIM_FD/PythonCode and running:

   ```
   extract_keyData_FD.py

   process_keyData_FD.py
   ```

6. To generate figure 2, go to the folder /CORSIM_FD/PythonCode and running:

```
generate_keyData_FD.py
```

7. To generate figure 3, go to the folder/test/MMPF/ and running

   ```
   Main_MMPF.py
   ```

   ```
   change the PR value and lag value to generate results for different penetration rates a
   ```

8. To generate figure 4, go to the folder/test/MMPF/ and running

   ```
   Main_MMPF_PR.py
   ```

   ```
   Error_PR.py
   ```

9. To generate figure 5, go to the folder/test/MMPF/ and running

   ```
   Main_MMPF_inflow.py
   ```

   ```
   change the inflow value to generate results for different inflows
   ```