A Novel Framework for Detecting Dangerously Slow Vehicle on Highway Corridor in a Partially Connected Environment

Kinjal Bhattacharyya, Pierre-Antoine Laharotte, Nour-Eddin El Faouzi

A Novel Framework for Detecting Dangerously Slow Vehicle on Highway Corridor in a Partially Connected Environment

Kinjal Bhattacharyya, Pierre-Antoine Laharotte, Nour-Eddin El Faouzi
Univ. Gustave Eiffel, Univ. Lyon, ENTPE, LICIT-ECO7 UMR T9401, F-69675, Lyon (France)

Introduction

- Slow Vehicles in free flow conditions are affecting the global driving safety with required maneuvers for other drivers, like « mandatory lane changes ». We assume that such behaviour should affect the natural spatial distribution of lane-changes and be featued by a recognizable pattern.

- **Main contributions :**
  - Introduction of a **lane change detection process with low requirements regarding the input data**, only requirement: the speed and heading time-series of Connected Vehicles (CVs) natively left by CVs through Cooperative Awareness Messages.
  - Introduction of a **slow vehicle detection process** based on the **weighted local density of lane changes** along a highway.

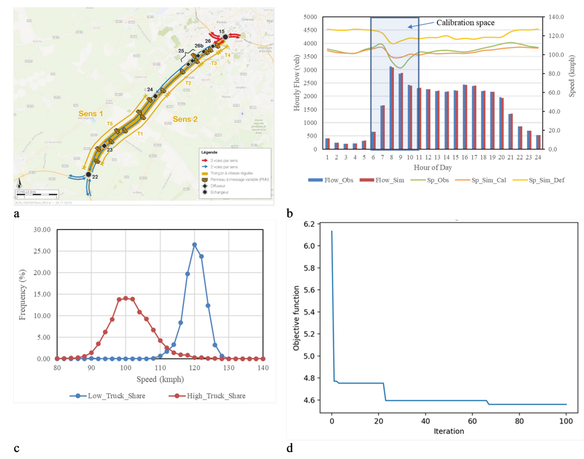
Experimental context

A fully simulated framework

- **SUMO [1]** is implemented with the default lane-change model [2] and a Krauss [1] car-following model calibrated thanks to a Genetic Algorithm.
- **TraCI** is used to dynamically collect data on Connected Vehicles with a refreshment period of 10 Hz.

TABLE 1: Default and calibrated values of selected parameters

Parameter	Default Value	Calibrated Value	
		CAR	TRUCK
maxSpeed (m/s)	36.11	32.2	26.4
minGap (m)	2.5	2.4	3.15
accel (m/s^2)	2.6	2.5	
decel (m/s^2)	4.5	4.65	
sigma	0.5	0.7	0.75
tau (s)	1	1.3	1.5



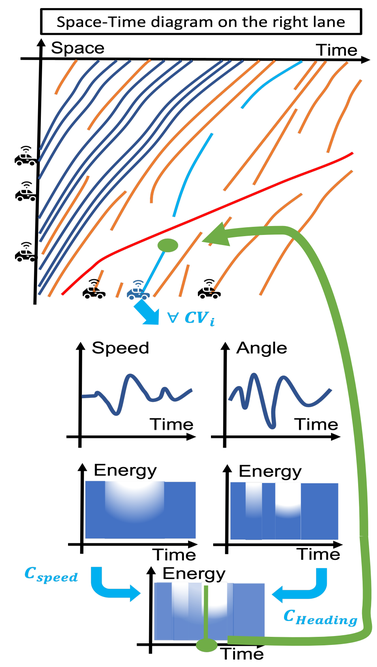
Methodology

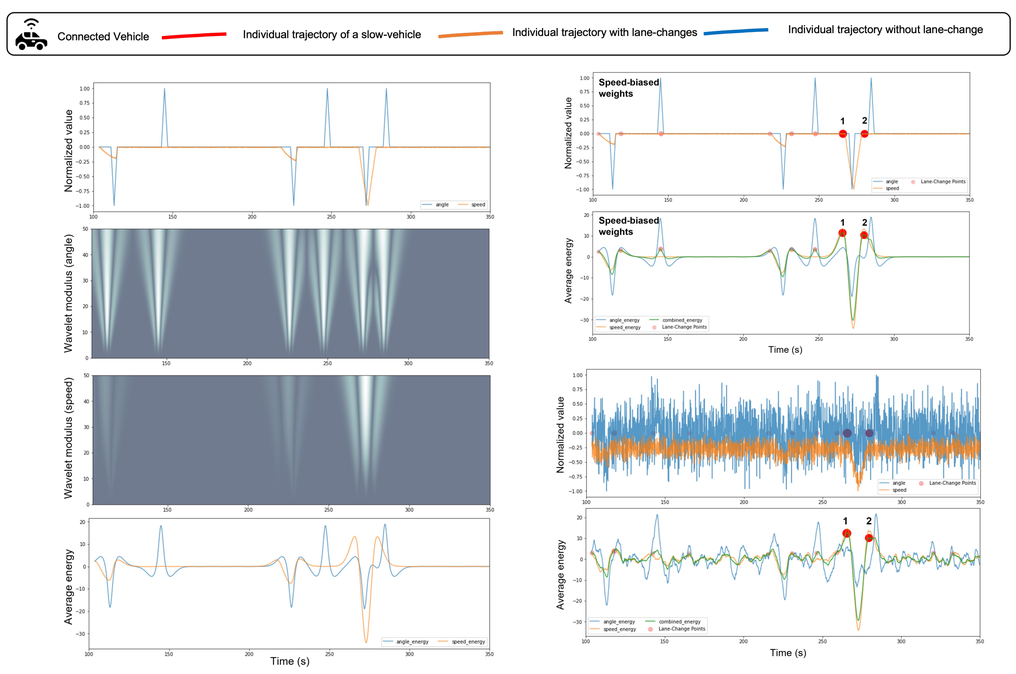
Step 1: Detecting lane-changes based on modifications of CV's energy computed by Wavelet Tranform

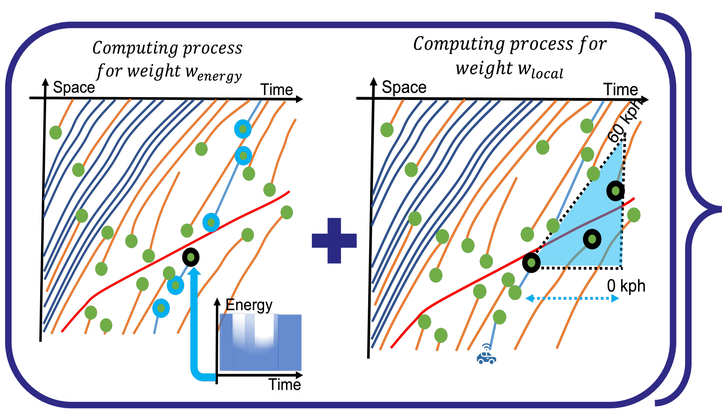
- For a fixed time-window (60s long, updated every 30s),
  - For any Connected Vehicles (CVs),
    - The speed and steering angle profiles of Connected Vehicles (CVs) are collected and feed a Wavelet Transform (WT) analysis.
    - The energy resulting from the WT of speed and steering angle is combined into a global indicator E^{global}.

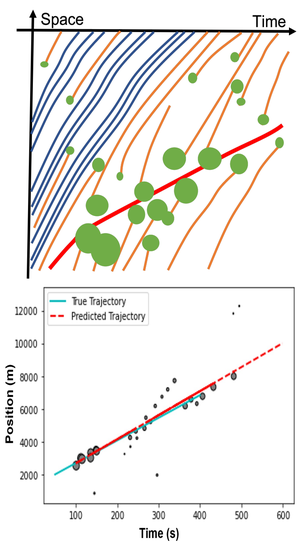
$$\forall b, \quad E^{global}(b) = C_{speed} \times E^{speed}(b) + C_{angle} \times E^{angle}(b)$$

- The global indicator E^{global} is compared to a threshold, when it is larger than 2.5, a lane-change is detected.









Step 2: Scatter plot analysis for slow-vehicles detection

- Preliminary identified lane-changes are discriminated by computing two weights aiming at reinforcing the distinction between patterns related to slow-vehicles and other lane changes
  - A weight related to the average WT energy
  - A weight related to the local density of lane-changes
- A (combination of) Gaussian (Weighted) Linear Regression is then applied to approximate the potential trajectory of the slow vehicle and threshold-based criterion are set up to validate the existence of a slow vehicle.

A 2-levels Gaussian Linear Regression based algorithm

Algorithm 1 Slow vehicle detection process algorithm

Require: WeightedScatterPlot

```
WLROuput ← WLR(WeightedScatterPlot)
                                  ▷ Apply LR with weighted samples
if WLROuput.Adjusted R^2 ≥ 0.9 then
                                  ▷ Check that WLR fit properly
    SlowVehicleSpeed ← WLROuput.SlopeCoefficient
                                  ▷ Collect speed estimation
    SlowVehicleInitialPosition ← WLROuput.InterceptCoefficient
                                  ▷ Collect initial position estimation
else
    Errors ← WLROuput.residuals
                                  ▷ Collect residuals errors from WLR
    MedError ← median(Errors)
                                  ▷ Calculate the median residual error
    SubsetScatterPlot ← WeightedScatterPlot [ Errors ≤ MedError ]
                                  ▷ Select only points with error lower than the median error
    LROutput ← LR(SubsetScatterPlot)
                                  ▷ Apply Linear Regression to the subset
    SlowVehicleSpeed ← LROuput.SlopeCoefficient
                                  ▷ Collect speed estimation
    SlowVehicleInitialPosition ← LROuput.InterceptCoefficient
                                  ▷ Collect initial position estimation
end if
if SlowVehicleSpeed ≤ 60 km/h then
    SlowVehicleStatus ← True
else
    SlowVehicleStatus ← False
end if
    return {SlowVehicleStatus, SlowVehicleSpeed, SlowVehicleInitialPosition}
```

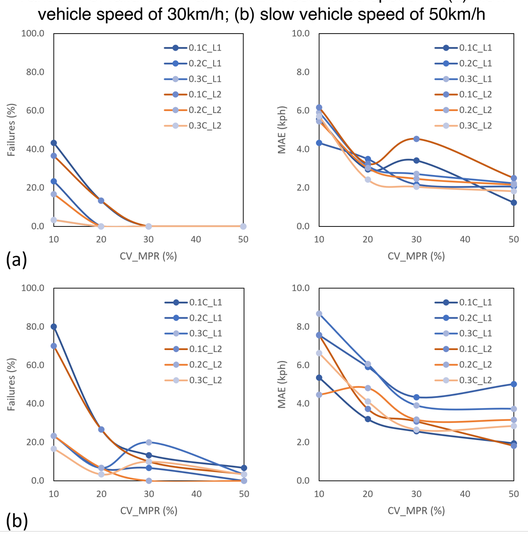
Main results

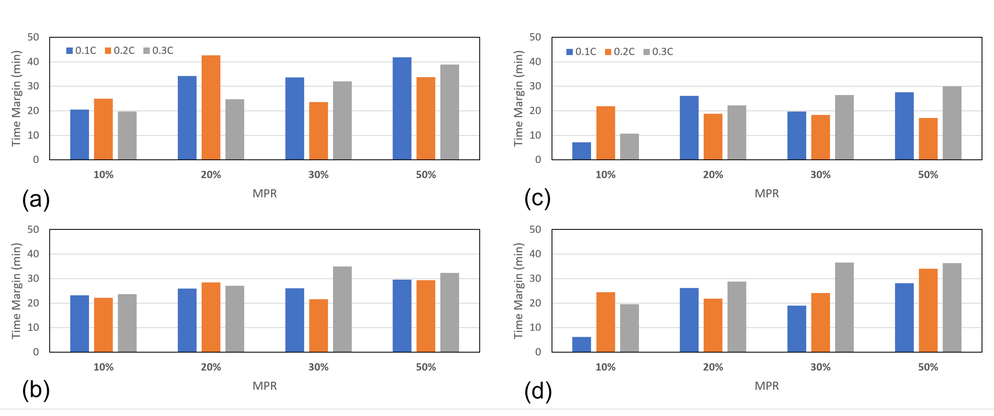
Experimental Design

- **Parameters under consideration**
  - Volume-to-Capacity ratio: 10%, 20%, 30%
  - CVs Market Penetration Rate: 10%, 20%, 30%, 50%
  - Speed of the slow vehicle: 30 kph, 50 kph
- Any unique combination of parameters was computed with 30 random seeds for stochasticity purpose

- **Key Performance Indicators**
  - Failure Rate
  - Mean Absolute Error (MAE) of Estimated Speed of Slow Vehicle
  - Time Margin for Warning

Variation of failure rate and MAE of estimated speed for (a) slow vehicle speed of 30km/h; (b) slow vehicle speed of 50km/h





Variation of time margin over different market penetration of CV and flow levels for (a) slow vehicle speed of 30km/h and level 1 regression; (b) slow vehicle speed of 30km/h and level 2 regression; (c) slow vehicle speed of 50km/h and level 1 regression; (d) slow vehicle speed of 50km/h and level 2 regression

Conclusions and Highights

- The **performance** of the slow vehicle detection algorithm **stabilizes** at **a market penetration of 30**% which emphasizes that a high market penetration might not be necessary for deployment ;
- Globally, the **MAE** is **below 6km/h and 8km/h** for slow vehicle speeds of 30km/h and 50km/h respectively ;
- Because with a slow vehicle driving at 50 kph, its speed is **closer to the flow's speed**, the **failure rate is higher** ;
- Mostly, the 2-levels regression outperforms the 1-level in terms of reduced failure rates and estimated errors.

Bibliography

[1] Lopez, P. A., M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Fl tter d, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wie ner, Microscopic traffic simulation using sumo. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 2575–2582.

[2] Erdmann, J., SUMO's lane-changing model. In Modeling Mobility with Open Data, Springer, 2015, pp. 105–123.

[3] Zheng, Z., S. 1 Ahn, D. Chen, and J. Laval, Applications of wavelet transform for analysis of freeway traffic: Bottlenecks, transient traffic, and traffic oscillations. Transportation Research Part B: Methodological, Vol. 45, No. 2, 2011, pp. 372–384.