

Single Lane Mixed Simulation

Version 1

Provides for controlled changes of velocity and implements static predefined driving profiles.

Version 2

Introduces dynamically adjusting speed targets: when a car comes near to the car in front of it, the velocity is adjusted. This requires introduction of separate, interruptible processes

Version 3

The naïve dynamic adjustment of velocity results in potentially extreme deceleration, approaching g -force. It introduces car specific limits for maximum and minimum normal acceleration.

Version 4

Adds emergency braking to the main control loop of a car. Emergency Braking cannot be interrupted. Braking is recorded as special event and added to the plotting routine. We can see the behaviour of braking moving down a group of cars tailgating each other.

Version 5

Emergency braking avoids most crash situation, except when the car in the back is going far too fast. As this is a single lane simulation a crash stops all traffic thereafter.

Multi Lane Mixed Simulation

Version 1

We begin with constructing the motorway infrastructure. Initially we introduce the building blocks (lane segments) and construct long stretches of a single lane motorway. The single lane simulation model is translated into a special case of a multi lane model.

Version 2

Now we start constructing a multi lane motorway by widening a motorway. We test the static infrastructure with some printing tools. The first step is to introduce explicitly controlled lane change.

- There is a new section Version History at the end of the Jupyter Notebook files listing all minor fixes.

Version 3

Adds a new composite property Surround giving a snapshot of the vehicles surrounding another at a particular point in time.

Class Vehicle:

- The constructor has been extended to allow specifying a start position in the start lane. This helps building test cases.
- The update method utilises the Surround class for finding the vehicle in front.
- Make the change lane process was made interruptible. When interrupted, the car should go quickly back to the old Lane.

Version 4

Added automatic overtaking behaviour in class Vehicle.

- This required introduction of a status bit changingLane (to prevent triggering the overtaking over and over again).
- Utility methods isFasterThan and isSlowerThan have been added to class Vehicle.
- Trace functionality for overtaking has been added.
- A convenience method getEvents() has been added to the class SimpleRecorder

Integration

- Generalise constants into parameters of Entities (like CAR_LENGTH, a_min, a_max etc.)
- Incorporate freeMotorwaySpeed() from Week 9 Example 2
- Bundle all input parameter into a data frame X with one row for each vehicle going onto the motorway
- Extract relevant parameter from data frame rec.getData() similar to the reporting in Week 7 Example 1, if necessary put them into a data frame Y
- Create a simulation routine with input parameter X and output parameter Y.
- Run that simulation routine and create a report.