

# Vehicle Message Exchange Simulator VMES

## Matlab™ Simulator

### Specifications

Copyright Mira Bou Farah, David Mercier, François Delmotte and Eric Lefèvre: Université Lille Nord de France, UArtois, LGI2A, F-62400, Béthune, France (20/09/2014).  
mira.boufarah@gmail.com, firstname.lastname@univ-artois.fr.

M. Bou Farah, D. Mercier, É. Lefèvre, F. Delmotte. *A high-level application using belief functions for exchanging and managing uncertain events on the road in vehicular ad-hoc networks*, Annals of Telecommunications, pp. 185–199, 2014.

## Contents

1	Introduction	2
2	Launch Scenarios	2
3	Functions Diagram	4
	Bibliography	5

# 1 Introduction

VMES simulator is a computer Matlab<sup>TM</sup> program. Its purpose is to test methods based on belief functions which aim is to manage information telling about events on the road in V2V communication. Different scenarios has been implemented.

This software is governed by the CeCILL-B license under French law and abiding by the rules of distribution of free software. You can use, modify and/or redistribute the software under the terms of the CeCILL-B license as circulated by CEA, CNRS and INRIA at the following URL : <http://www.cecill.info>.

## 2 Launch Scenarios

Twelve scenarios are developed and tested. User can find the description and the results of these scenarios in [2, 1, 3].

- Scenarios 11, 12, 13 and 14 correspond respectively to the scenarios n°1, n°2, n°3 and n°4 for accident event type in M. Bou Farah thesis [2, Chapter 2].
- Scenarios 21, 22, 23, 24, 25 and 26 correspond respectively to the scenarios n°1, n°2, n°3, n°4, n°5 and n°6 for accident event type in M. Bou Farah thesis [2, Chapter 3].
- Scenario 31 corresponds to the scenario used to test the resistance of the methods to malicious nodes in M. Bou Farah thesis [2, Section 4.1].
- Scenario 33 corresponds to the scenario used to test the proposed improvement of the aging mechanism in M. Bou Farah thesis [2, Section 4.2].

To launch scenarios, user must first copy all the simulator folder.

Procedure to launch scenarios :

- To launch scenarios 11, 12 or 13, user must call the function *simulations* in the Matlab<sup>TM</sup> command window as follows : `simulations(N)`, where *N* is the scenario number (*N* is equal to 11, 12 or 13).
- To launch scenario 14 and test different debits, user must call the function *simulation* as follows :
  - Modify the variable *Flows* in the function *scenarioN*, for example : the debit  $Flows = [1/50]$ .
  - Call the *simulation* function in the Matlab<sup>TM</sup> command window: `simulation(14)`.
  - If user wants to test another debit, he must modify again the variable *Flows*, and call the *simulation* function.
- To launch scenarios 21 to 26 and test different methods, user must call the function *simulation* as follows :

- Modify the variable *Methodes.listeATester* in the function *scenarioN*, for example : *Methodes.listeATester* = [8]. Please do not test more than one method at a time for these traffic jam scenarios, because of a simulator limitation.
- Call the *simulation* function in the Matlab™ command window: **simulation(N)** where *N* is the scenario number.
- To launch scenarios 31 or 32, user must call the function *simulation* as follows :
  - Call the *simulation* function in the Matlab™ command window: **simulation(N)** where *N* is the scenario number.

**In summary :**

- **simulations(N)** : to launch scenarios 11 to 13.
- **simulation(N)** : to launch scenarios 14, 21 to 26, 31 and 32.

Procedure to create a new scenario *N*:

- Create a new Matlab™ function *scenarioN.m*.
- Create a new folder in the *simu* folder : *scenarioN*.
- If the user wants to launch many series of simulations (as scenarios 11 to 13), he must create folders *L1* to *LX* in *scenarioN* folder, where *X* is the number of series. For example, in scenarios 11 to 13, the number of series is equal to 10.

### 3 Functions Diagram

Figure 1 illustrates the diagram of simulator functions.

Each function is described in details (aim, inputs, outputs) at the beginning of its *.m* file. Each *.m* file corresponds to one function which name is identical to the file name. To get the description of a function, user can use the *help* Matlab™ command in the Matlab™ command window: `help functionName`.

In order to get simply the results of implemented scenarios, the function *resultats* has been implemented. Once the data of the simulations saved, load the data, then call the *resultats* function in the Matlab™ command window : `resultats()`.

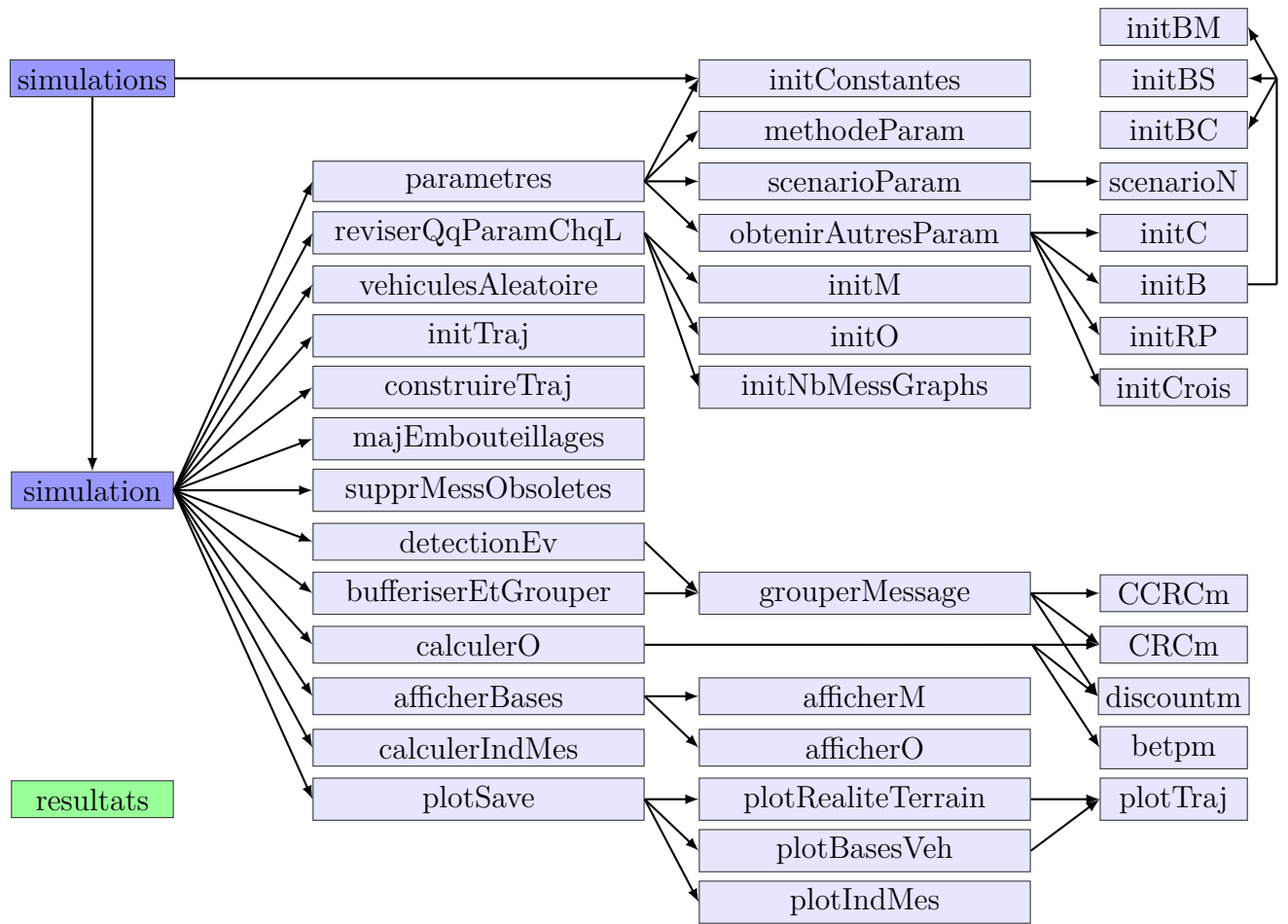


Figure 1: Functions diagram of the simulator

## References

- [1] M. Bou Farah, D. Mercier, É. Lefèvre, and F. Delmotte. Exchanging dynamic and imprecise information in V2V networks with belief functions. *16th International IEEE Conference on Intelligent Transport Systems, ITSC 2013, The Hague, The Netherlands*, pages 967–972, October 2013.
- [2] M. Bou Farah. *Méthodes basées sur des fonctions de croyance pour la gestion des informations imparfaites dans les réseaux de véhicules*. PhD thesis, Université d'Artois, 2014.
- [3] M. Bou Farah, D. Mercier, F. Delmotte, É. Lefèvre, and S. Lagrue. Methods handling accident and traffic jam information with belief functions in vanets. *3rd International Conference on Belief Functions*, 2014.