10 A Framework for Electric Bus Powertrain Simulation in Urban Mobility Settings: coupling SUMO with a Matlab/Simulink nanoscopic model

Jose Macedo, Guilherme Soares, Zafeiris Kokkinogenis, Deborah Perrotta, Rosaldo J. F. Rossetti;
Artificial Intelligence and Computer Science Laboratory (LIACC), Department of Informatics Engineering (DEI), Portugal
Zafeiris Kokkinogenis;
Institute of Mechanical Engineering (IDMEC), Faculty of Engineering, University of Porto, Portugal
Deborah Perrotta;
Centro Algoritmi, Universidade do Minho, Portugal

10.1 Abstract

The increasing traffic volumes in urban areas result in heavy environmental condition. The necessity for a rapid and sustainable transport of people and goods became one of the high priorities in the public governance's agenda. Government agencies and organizations try to develop more stringent standards for the fuel consumption and gas emissions through reduction of the congestion on network infrastructures. Among the different solutions under exploration to tackle congestion problems is the development of Electric Vehicles for personal and public transport as an alternative to internal combustion engine vehicles. Albeit this direction seems to be promising, there are still open concerns associated with the consumption of energy and other performance measures for considering the implementation of electric buses in urban settings as a cost-effective solution. An important aspect in evaluating the performance and adequateness of such vehicles is the fact they must be immersed in the urban environment context. This work presents the distributed simulation architecture for electric bus powertrain simulation within a realistic urban mobility context. The architecture uses the microscopic simulator SUMO to represent the urban reality coupled with a nanoscopic model of an electric bus powertrain subsystem implemented in Matlab/Simulink. The paper discusses on how such a platform will be important for analysing the way traffic flow and its dynamics affect the performance of electric buses when there are obstructions or intense traffic conditions.

Keywords: Microscopic simulation, nanoscopic model, SUMO, Matlab/Simulink, electric vehicles, public transport.

10.2 Introduction

In the last decades we witnessed a large increase in traffic and transport demand that has created and aggravated capacity problems in the infrastructure causing traffic congestions and delays. Heavy traffic conditions not only affect the welfare of citizens from the economic point of view but they are also related to their health status speaking both psychologically, due to stress accumulated during their travels, and physically, due to high air pollution levels. In fact, a problem associated with the increasing use of personal vehicles is the carbon emissions. According to the 2009 Urban Mobility Report [1] the congestion led urban Americans to travel 4.2 billion hours more, which resulted in 2.8 billion gallons of extra fuel, an increase of more than 50% over the previous decade. Government agencies and organizations try to develop more stringent standards for the fuel consumption and gas emissions through reduction of the congestion on network infrastructures [2], [3].

In this sense, incentives and investments on public transport modes as well as research on more eco-sustainable solutions have been performed as attempts to minimize both the air pollution and congestion problems. One of the approaches currently investigated and implemented to provide an eco-sustainable solution for the public transport is related to the employment of electric bus powertrains in metropolitan transportation as an alternative to internal combustion engine buses [4]. However, there are still open issues related to the consumption of energy and other performance measures for considering the adoption of electric buses in urban scenarios as a cost-effective solution [5].

In order to analyse the adoption of electric buses as a cost-effective solution, it is necessary an evaluation of the performance and adequateness of such vehicles while immersed into an urban environment context. Albeit there are different tools and models to assess the behaviour of electric buses, such evaluations often lack the aforementioned integration with the traffic dynamics. This work intends to contribute to the improvement of transportation and traffic analysis tools and presents a flexible integrated architecture for electric bus powertrain simulation in urban mobility settings. Such a platform will be important for analysing how traffic flow and its dynamics affect the performance of the electric bus when there are obstructions or intense traffic conditions.

The paper describes how the microscopic traffic simulator SUMO (Simulation of Urban Mobility) [6] is used as a means to represent an urban traffic scenario and its integration with a model of an electric bus powertrain subsystem [7], implemented in MatLab/Simulink. The remainder of the paper is organized as follows. Section 10.3 introduces the proposed architecture. Section 10.4 presents a simple driving scenario in order to show the usefulness of the platform architecture. Section 10.5 draws some conclusion and discusses future developments.

10.3 Proposed architecture

Figure 10-1 depicts the software architecture for the electric bus simulation in a traffic environment. The proposed design is distributed combing the microscopic and nanoscopic aspects of the real system.

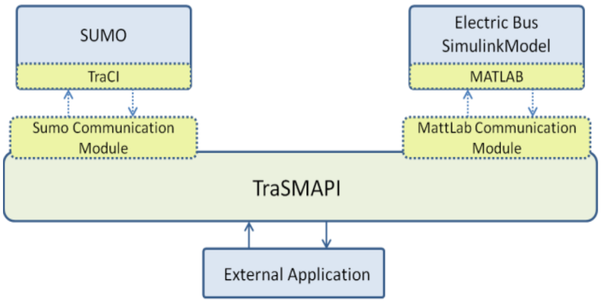
SUMO
TraCI
Electric Bus SimulinkModel
MATLAB
Sumo Communication Module
MattLab Communication Module
TraSMAPI
External Application

Figure 10-1: Proposed architecture.

The architecture consists of six modules, shortly described in subsections that follow.

10.3.1 SUMO Microscopic Traffic Simulator

SUMO (Simulation of Urban MObility) is a well-known open-source microscopic traffic simulation package that appears for the first time in 2001. Today SUMO is already in its 0.16 version and is one of the most commonly used microscopic traffic simulator by the research community. SUMO became an important tool in many researches in the urban traffic and transportation domains. It has gained its place among the academic community with many of scientific papers referring to it. Today, SUMO is not just a traffic simulator, but rather a suite of applications which help to prepare and to perform the simulation of traffic in complex settings.

SUMO can be seen as a pure microscopic traffic simulation: each vehicle is associated with an identifier, the departure time, and the vehicle’s route through the network. It is also possible to describe each vehicle in more detail. For example, it is possible to define arrival properties of the vehicle, which lane it could use, its velocity and its position. Also, each vehicle can be assigned to a vehicle-type which describes its physical properties and variables of the used movement model. This is an important feature for this work since it allows us to make a one-to-one association between the bus vehicle in SUMO and the electric bus powertrain subsystem (EBPS) of the electric bus.

It represents and simulates the traffic environment that is the road infrastructures and the vehicles moving on it providing a microscopic level of systems resolution. Within this module the TraCI API is integrated for gaining access to the SUMO simulator.

10.3.2 The MatLab/Simulink Electric Bus Powertrain Subsystem

The Electric Bus Powertrain subsystem, or EBPS for short, is a mathematical model of an electric bus subsystem implemented in MATLAB’s Simulink (Figure 10-2). MATLAB (MATrix LABoratory) is a high performance software suit for numerical computation, data visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. It is an interactive system and a programming language for computing technical and scientific cooperation in general [8]. MATLAB features a family of specific applications called toolboxes, very important to most users of that suit.

Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Domains to which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others [9].

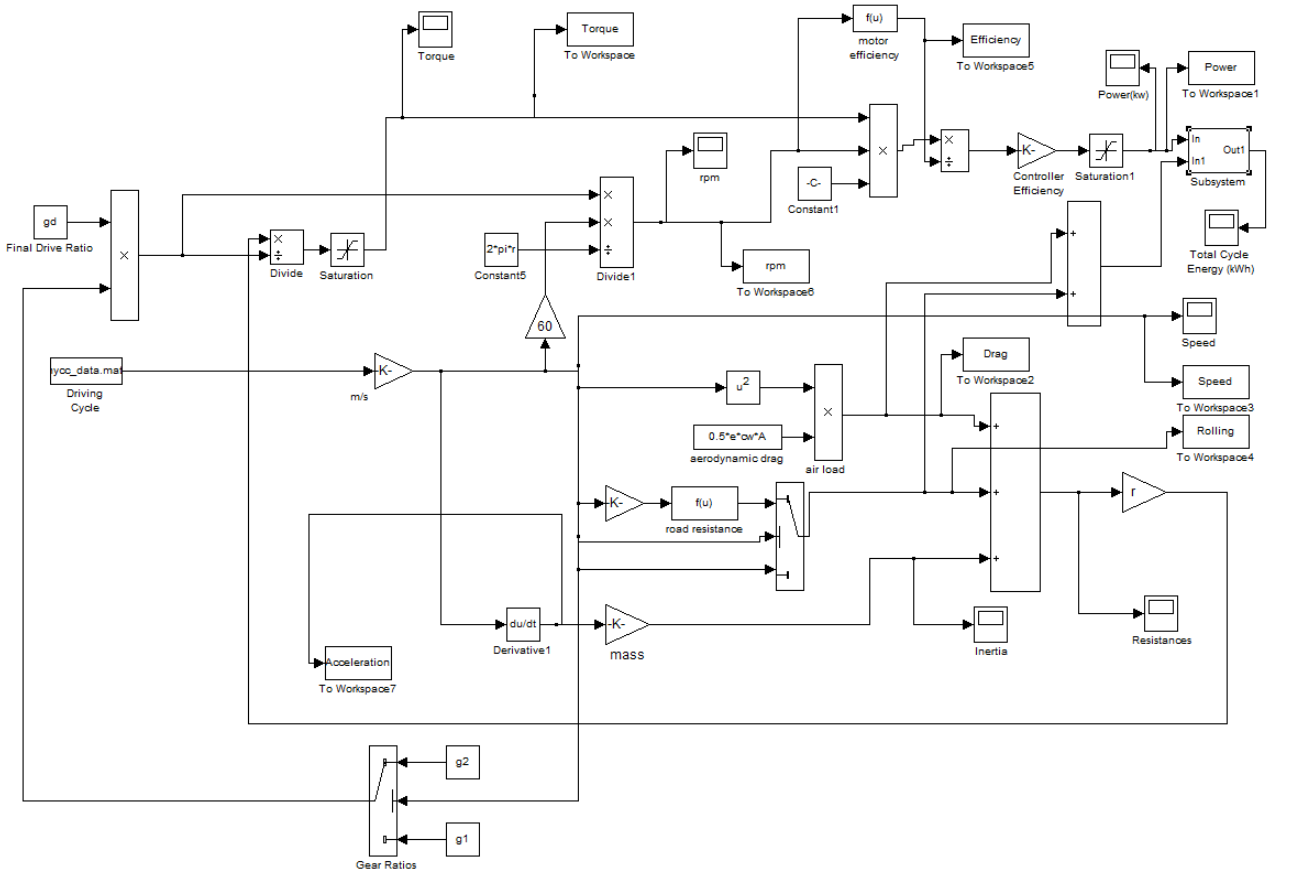

Figure 10-2: Main subsystem of EBPS model.

Simulink is a software package for modelling, simulating, and analysing dynamic systems. It supports linear and non-linear systems, modelled in continuous time, sampled time, or a hybrid of the two. Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. After defining a model, it is possible to simulate it, using a choice of integration methods, either from the Simulink menus or by entering commands in MATLAB's command window [10]. The simulation results can be put in the MATLAB workspace for post processing and visualization. And because MATLAB and Simulink are integrated, it is possible to simulate, analyse, and revise models in either environment at any point [11].

The model used has several subsystems, which are used to calculate specific parameters. One of these subsystems represents the vehicle's powertrain, taking into account the forces that work against its movement and the gear ratios involved.

An output of this subsystem computes the amount of required energy for a driving cycle to be completed. There is a third subsystem that calculates the amount of energy that may be possibly recovered from the regenerative braking, taking into account the kinetic energy of the vehicle. The two other subsystems are related to the batteries and the super-capacitors, evaluating whether they are capable of absorbing the energy from the braking [7]. Figure 10-2 illustrates the main subsystem of the model.

10.3.3 TraSMAPI

TrasMAPI is a tool for the simulation of dynamic control systems in road networks, with special emphasis on Multi-Agent Systems. This tool allows real-time communication with microscopic simulators providing a framework for the development of multi-agent solutions [12]. The abstraction over the simulator enables it to run different traffic simulators using the same API (Application Programming Interface) allowing, for example, the comparison of results of the same application in different simulators [13].

This tool can be seen as a generic microscopic traffic simulator API that allows a higher level of abstraction and transparency, in terms of variable or simulation management. It provides methods to control the simulation life-cycle and allows the coupling of an external application for simulation data management.

However, the importance of TraSMAPI for this project is mainly related with the communication module that it provides. TraSMAPI will be mainly used on the integration of both systems, namely SUMO and EBPS. In this sense, its abstraction layer for interaction with microscopic simulators brings great value for the integration, allowing an easier control of the simulation process.

10.3.4 The MatLab Communication Module

MatLab could be seen as an API for Simulink since Simulink models can be controlled by MatLab methods. Simulink models are standalone models and exist in mutual symbiosis with the MatLab environment. The only way to access them externally is through MatLab's methods and calls. For this reason a "control" using MatLab's interface to Simulink was applied. In this sense, this module extends TraSMAPI's API to communicate with Simulink through MatLab.

In order to establish communication with MatLab, the *matlabcontrol* API was used. The *matlabcontrol* is a Java API to interact with MatLab allowing its methods

invoke behaviour of Java objects [14]. Using this, it was only necessary to create a proxy to work as an image of the MatLab application. After that, all the calls to MatLab are performed through this proxy.

Establishing the communication with MatLab was the first part in the development of this module. The second part was to control the Simulink model's step-by-step simulation through Matlab commands. To do so, the *set_param()* function was chosen. This function allows starting, pausing, stopping and resuming the simulation, as well as advancing one simulation step at a time.

10.3.5 The SUMO Communication Module

Unlike Matlab/Simulink, SUMO already comes with an API that provides a communication protocol. In fact, the TraCI interface provides a set of methods that allow an easy interaction with the simulator's state variables. In this sense, this module implements TraSMAPI's API to communicate with SUMO through TraCI's API. This module is composed of a set of functions, implemented in Java that comprises the necessary commands to interact with SUMO. TraCI has an extensive number of methods, each one associated with an entity in the simulation. For the scope of this project, the only TraCI's methods that were used are relative to the vehicle entity and its speed.

10.3.6 External Application

To perform the integration, the communication modules herein described were used to implement the TraSMAPI's interface. In this way, although each simulation tool has its own implemented methods, the external application could control and communicate with both simulation tools using TraSMAPI generic interface.

The application is responsible for the synchronization and the data exchange between the two environments. The main steps of the program are:

1. Launch the simulators, and establish a connection with them;
2. Start the simulation on both systems at the same time, and keep them waiting;
3. Advance simulation one step;
4. Get the speed of a SUMO vehicle and send it to the EBPS simulation process;
5. Get the EBPS outputs;
6. Advance simulation one step;
7. Stop both simulations and disconnect from respective simulators;
8. Close simulators.

Steps 4 to 6 are performed in a cycle until the last simulation step. In this integration a SUMO vehicle is delegated by the EBPS Simulink model to act on behalf of it. Thus a bus vehicle performs a trip on a given route of a SUMO's network and sends feedback to Simulink. The EPBS model computes a number of metrics, defined in [7], based on the information received by the SUMO vehicle.

10.4 Simulation scenario

As it has been mentioned previously, the devised integrated platform intends to offer a valid tool to traffic managers and practitioners for analysing how traffic flow and its dynamics affect the performance of electric buses when there are obstructions or intense traffic conditions. In this section we present a scenario where two different traffic flows were specified, so that the EBPS could be exposed to different solicitations due to daily traffic conditions. This illustration serves only as a demonstration of the possible type of analysis one can conduct using the proposed tool.

For this purpose a new road network was designed and created for SUMO. The network is a model of the central area of Porto's down-town (Figure 10-3) and it has been extracted from OpenStreetMaps database. The choice of this urban traffic area is motivated by the fact that it is an area of the centre of the city with high traffic densities and a large number of traffic lights.

In order to provide an example of what kind of study the framework can perform for the EBPS analysis, the *Acceleration* and *TotalEnergyCycle* metrics have been chosen among others. Other metrics are defined in [7].

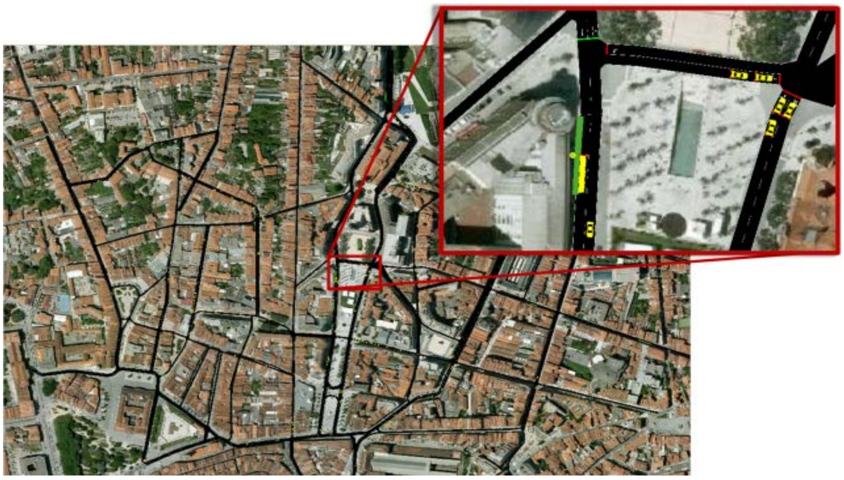

Figure 10-3: Test bed network of Aliados Avenue.

For the first set-up a repetition of five seconds has been used for the traffic flow generation. Since one second represents one step in simulation, this means that each trip is repeated by a new vehicle at every step. The second set-up of traffic flow has been generated with a repetition trip rate of one second. With this rate, rapidly the network begins to become overloaded, representing peak-hour periods.

The first parameter to be analysed is the acceleration profile of the electric bus under the different traffic conditions. It can be observed that under intense traffic conditions, there were performed tougher accelerations and decelerations (represented by the peaks on the plot of Figure 10-4). Besides that, the stop-and-go episodes were characterized by being performed on smaller periods of time when compared to low traffic. This type of behaviour have direct negative implications on the maintenance costs and on the efficiency of the bus, once the higher values of acceleration impact on the necessary energy for the bus to transverse a certain route.

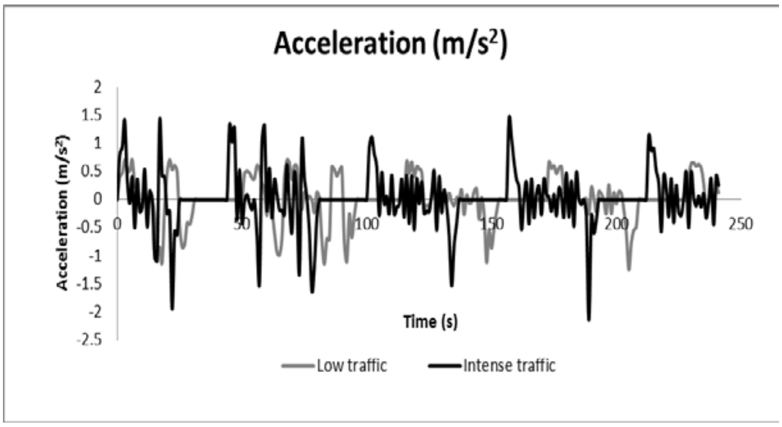

Figure 10-4: EBPS acceleration under different traffic conditions.

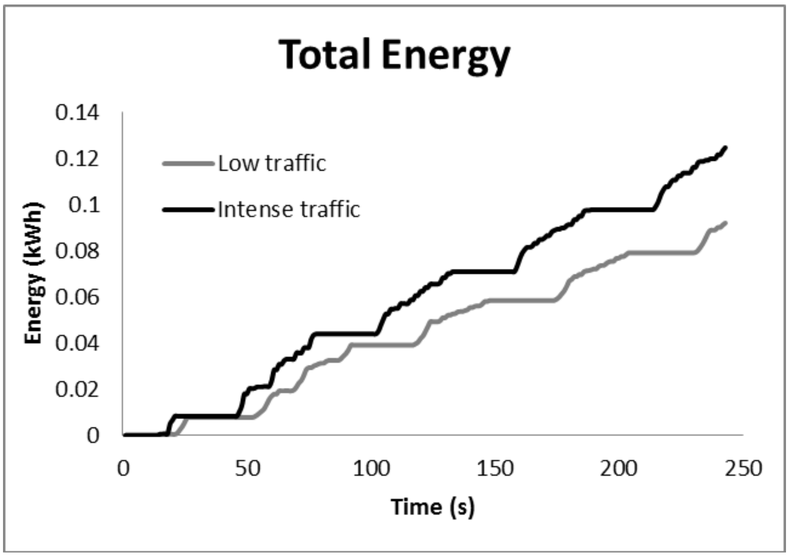

Figure 10-5: EBPS energy consumption under different traffic conditions.

After observing the acceleration profile, it was decided to analyse the amount of energy that the electric bus spent under the proposed conditions, as a way to corroborate the assumptions made previously. Not surprisingly, under intense traffic, the electric bus spent 36% more energy (0.125 kWh) when compared to the low traffic scenario (0.092 kWh), as it can be observed in Figure 10-5. This has immediate consequences on the costs associated with the recharge of the bus.

It could be concluded that intense traffic conditions affect negatively the performance of an electric bus, making it spend more energy to perform similar routes and also demanding more power from the bus motor, which implies on higher maintenance costs.

10.5 Conclusions

This paper discusses on the implementation of a distributed architecture for electric bus powertrain simulation in an urban mobility context using two types of simulation environments, namely the SUMO (Simulation of Urban Mobility) microscopic traffic simulator and the MatLab/Simulink environment. The proposed architecture envisions for a flexible approach to the integration of two simulators, one from the transportation area, and the other from automotive area. Its purpose is to offer a valid tool for traffic managers and practitioners so they can analyse how traffic flow and its dynamics affect the performance of electric buses (as well as other type of vehicles) when there are obstructions or intense traffic conditions. To illustrate the usefulness of the platform architecture a driving scenario has been presented and the preliminary results that can be obtained by the simulation have been shown.

The advent of promising technologies will increase the need for flexible R&D tools for such complex systems to be evaluated. Furthermore, there is an encouraging potential on this proposed approach able to bring together automotive and transportation research communities to work together on the development of Future Urban Transport Systems.

10.6 Acknowledgment

This project has been partially supported by FCT (Fundação para a Ciência e a Tecnologia), the Portuguese Agency for R&D, under grants SFRH/BD/67202/2009 and SFRH/BD/51256/2010.

10.7 References

[1] David Schrank and Tim Lomax. The 2009 URBAN MOBILITY REPORT. Technical report, Texas Transportation Institute, The Texas A&MUniversity System, 2009.

[2] R. Arnott, T. Rave, and R. Schöb. Alleviating urban traffic congestion. MIT Press Books, 1, 2007.

[3] Benjamin K. Sovacool. A transition to plug-in hybrid electric vehicles (PHEVs): why public health professionals must care. Journal of epidemiology and community health, 64(3):185–7, March 2010.

[4] N. Unger, T. C. Bond, J. S. Wang, D. M. Koch, S. Menon, D. T. Shindell, and S. Bauer. Attribution of climate forcing to economic sectors. Proceedings of the National Academy of Sciences, 107(8):3382–3387, 2010.