BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LX No. 3B/2008	43 - 48	Seria Tehnică

A Modern Framework for Road Traffic Modelling and Dynamic Simulation

Gabriel Rădulescu*, Melania Cruceru**

* Universitatea Petrol-Gaze din Ploieşti, B-dul Bucureşti nr. 39, Ploieşti
e-mail: gabriel.radulescu@upg-ploiesti.ro

** Yokogawa Europe B.V. – Romania Branch, B-dul Dimitrie Pompeiu nr. 6, Bucureşti

Abstract

Modelling and simulation are one of the most used tools in processes investigation, especially when dealing with complex/intricate dynamic behavior. The road traffic is a typical example in this field, its complexity being a consequence of strong interaction between traffic participants, transport infrastructure and traffic controls. This paper proposes a modern modelling and dynamic simulation approach as framework for controlled traffic investigation. Based on Microsoft .NET platform and making use of OpenGL and GLUT graphical libraries, the simulator implemented by authors is a good way to perform advanced studies, from model developing and testing to traffic management and event prediction.

Key words: *road traffic, dynamic modelling, graphical simulation*

Introduction

The mathematical model-based simulation is a very efficient tool in various fields of activity, a typical example being the complex studies on road traffic. But modelling and simulation are not an easy task, mainly due to inherent subjective factors implication. More, a good simulation platform has to reveal with fidelity as many as possible real traffic situations, which involves an appropriate implementation for one very sensitive aspect: the *interaction* between traffic actors and road transport infrastructure [5].

Since 90's, the available commercial traffic simulators continuously improve their capacity to generate an adequate and accurate representation for the so-involved traffic infrastructure, active/passive actors and regular/unexpected road events [2]. But, at present, an alternative gains place on the market: instead of using a commercial simulator covering a large class of applications, it is better to use a general framework, which permits building-up customized simulators focused only on the specific problem to be treated.

The authors of this paper have recently started a research project focused on developing a modern framework for studies based on road traffic mathematical models, which makes possible an easy traffic simulators implementation. A project overview is here presented, covering from general aspects (as standard modeling and simulation techniques) to specific solved problems when building-up an application for traffic studying in a simple crossroad.

An overview on the Framework's Simulation Techniques

The traffic simulation applications make use of fundamental mathematical apparatus in order to dynamically describe the studied system behavior and estimate its performances through standard indicators as traffic flow, speed or density. Concerning the equations form, there are two main global approaches: macroscopic and microscopic models [5].

The macroscopic approaches do not focus on each independent vehicle, but on the global traffic, by assimilating it to a continuous stream. Such models usually involve a relative low building-up effort and need less mandatory input data, but have the disadvantage of lower detailed informational outputs.

By contrary, *the microscopic approaches* allow representing the individual behavior and performances for implied vehicles that mutually act in coordination with each other (as actors in the traffic scene). Although model formulation and software encoding need sometimes significant effort, the simulators based on these micro-models are able to offer a rich informational output (the major reason why our framework is oriented to this microscopic-type approach).

Microscopic representation basic principles

For each moving object, the traffic micro-simulation models associate a specific type (i.e. car, truck, bus) and performances (maximum values for speed, acceleration/deceleration, minimum turning angle and so on). In conjunction with drivers behavior mathematical representation (on a whole range from “very cautious” to “very aggressive”), such aggregated models offer a specific *profile* for each vehicle acting in the studied traffic scenario [5].

By adopting a microscopic representation, each vehicle's position and speed are periodically actualized (at each simulation step) upon an algorithm taking into account – as above stated – first the inner performances, respectively the associated driver's behavioral type and then other factors as surrounding vehicles actions, traffic control devices states, road surface quality etc.

Traffic modelling algorithms

After adopting appropriate models for cars performances and drivers' behavior, any individual object in the traffic scene moves accordingly with some particular traffic algorithms. We decided using three primary types (car following, lane changing and gap acceptance algorithms) and two secondary types (signal-based traffic controlling and car queue remodeling).

The car following algorithm describes how the vehicles interaction each other and, subsequently, how they are distributed within traffic stream, by determining the *headway* (this specific term denoting in fact the distance between two consecutive cars) [5]. On our simulation platform, the car following algorithm has an imposed (constant) headway target value, as well as a fixed maximum allowed speed. Another related simplifying assumption is that any driver has a determined and constant driving style, but these hypotheses will change in future framework's implementations.

The lane changing algorithms control how the vehicles act when crossing from one track to another, focusing on this procedure influence on the traffic flow [5]. This algorithm implementation is planned for the very next future.

The gap acceptance algorithms make an early estimation of the gap between concurrent vehicles in a potential conflicting zone and, accordingly with the planned movement, traffic conditions, driver's behavioral type and an acceptance threshold, decide if the subject may or may not proceed with maneuver [1, 5]. Up to now we have experimented a simple gap acceptance algorithm (with fixed distances estimation), but open to any change in the future framework stages.

The above-mentioned secondary algorithms (*signal-based traffic controlling* and *car queue remodeling methods*) are now implemented at a very basic level in the simulation environment proposed by authors, but in the future they will play an important role in the realistic traffic dynamic representation.

Building-up the Proposed Framework for Studies on Road Traffic Modelling and Dynamic Simulation

As previously mentioned, by adopting the macroscopic representation, each vehicle is considered an individual actor, not being part of any macroscopic structure. Of course, at the macro-level one can still observe realistic behavioral effects, but all of these result only from individual (micro) interactions.

This section shows how the announced framework was functionally aggregated, starting from the micro-representation basics and mathematical modeling principles, at present it being ready to use in traffic dynamics studies.

The core mathematical model

In our cellular automaton based representation, each vehicle is determined by its *passive properties* (seen as model constant parameters: length l, maximal acceleration a_+, maximal deceleration a_-) and *active properties* (its allocated state variables: position in queue x, actual speed v and acceleration a – updated with each simulation step Δt). Acceleration is considered as the main state variable, because it strongly depends on the environment and, more, v and x can be easily calculated from a.

The first simplifying assumption is having a constant acceleration value for each Δt time horizon, the a value (positive for acceleration, negative for deceleration) being directly influenced by the driver's actions on gas pedal. Considering also that a should tend to its maximum allowed values (a_+ or a_-), the following equation is adopted in our model:

$$a(t) = \begin{cases} a_+ \tanh(\varepsilon/a_-), \text{if } \varepsilon < 0 \\ a_- \tanh(\varepsilon/a_-), \text{otherwise.} \end{cases} \quad (1)$$

ε is the deviation between *ideal* traffic conditions (free road, no supplementary speed restrictions) and *real* ones and may be defined in several ways, upon choice (as shown below).

It can be observed that equation (1) brings a realistic *primary value* of a depending on ε value, by using the hyperbolic tangent operator, denoting a stronger driver reaction on the gas pedal as the deviation (positive or negative) has a bigger absolute value.

Another assumption is that each vehicle runs with the maximum allowed speed, but eventually limited by the general and/or local factors: traffic rules, distances to fixed obstacles (traffic lights, signs) or moving ones (other vehicles) and so on. Without any restrictions, the primary-form speed equation we used in our model is

$$v(t + \Delta t) = \max(0, v(t) \times a(t)\Delta t), \quad (2)$$

while the position equation is represented by

$$x(t + \Delta t) = \max(0, x(t)\,\Delta t + 0.5a(t) \times (\Delta t)^2). \quad (3)$$

One can observe that equations (2) and (3) do not allow any negative values for v and x (meaning no turning back for all vehicles in the same queue).

In addition to these basic equations (1), (2) and (3), the mathematical model has to take into account, as told, two main restricting factors: the speed limits and obstacles presence. A unitary approach makes the deviance ε to reflect both cases, as shown below.

The speed limits, for instance, may give a value $\varepsilon = V - v$, where V denotes the maximum allowed speed. Equation (1) gives now a positive value for acceleration a if $v < V$, zero if $v = V$ or a negative one if $v > V$. It is easy to demonstrate that, after several number of time steps Δt, v will equal V, proving a good adapting feature for the model (when new limitations – shown by changes in V – happen to occur).

The obstacles, fixed or mobiles, also need a permanent (current) state evaluation, as their presence in the traffic scene may vary in time. But when dealing with any kind of obstacles, the general *safety stop time* rule applies; it correlates the drivers actions (changes in a) with current traffic conditions, in a way allowing the cars come near obstacles, but never touch them.

To be more specific, let us consider the arrival distance of the vehicle d_a (which, at the safety limit, when added to vehicle current position gives the next obstacle coordinate). It may be defined as sum between the reaction distance $d_r = v \times t_r$ (where, during the reaction time t_r, the driver sees and identifies the obstacle) and the stopping distance d_s (where, during stopping time t_s, the vehicle runs decelerated, accordingly with new commands from its driver). By applying well-known kinematic laws, d_a is

$$d_a = d_r + d_s = v \times t_r + v \times t_s + 0.5a \times t_s^2 = v \times t_r - 0.5v^2 \times a^{-1}. \qquad (4)$$

When a fixed obstacle is present into scene, the deviation ε should not be so strongly influenced by the difference $V - v$; instead it may be calculated as $\varepsilon = d_o - d_a$, where d_o denotes the distance where the obstacle is currently located. Re-written by using state variables, ε becomes

$$\varepsilon = x_0 - x - l - 2v - 0.5v^2 \times a_-^{-1}\,, \qquad (5)$$

where x_o is the obstacle coordinate.

A slight change in equation (5) leads to the mobile obstacles (usually cars in the front) general case,

$$\varepsilon = \begin{cases} x_0 - x - l - 2v - 0.5(v - v_o)^2 \times a_-^{-1}, & \text{if } (v - v_o) > 0 \\ x_0 - x - l - 2v & \text{otherwise,} \end{cases} \qquad (6)$$

v_o being here the obstacle speed. Like in the case of speed limiting, equation (6) gives in fact appropriate values for acceleration a and, more, it can be generally used – as (6) becomes identical with equation (5) when $v_o = 0$ (for a fixed obstacle). For a more complete image of the model the reader is referred to quoted literature [5, 8].

Software resources

At present, the proposed framework targets the Windows systems only, it being developed in Visual C++ (Microsoft Visual Studio 2005), which offers a powerful and flexible platform for advanced application programming [4]. These features are emphasized by integrating SDK (Software Development Kit) tools, with all benefits taken from auxiliary resources oriented to .NET Framework efficient usage [6]. Such a powerful development tool was primarily used for implementing the simulator engine (which solves the equations above and offer a robust communication interface both with operator and graphical system). In addition, a modern solution was adopted for the interactive results representation by using GLUT (OpenGL Utility Toolkit) libraries, which make possible a future portable solution for UNIX-type platforms [7].

Implementation overview

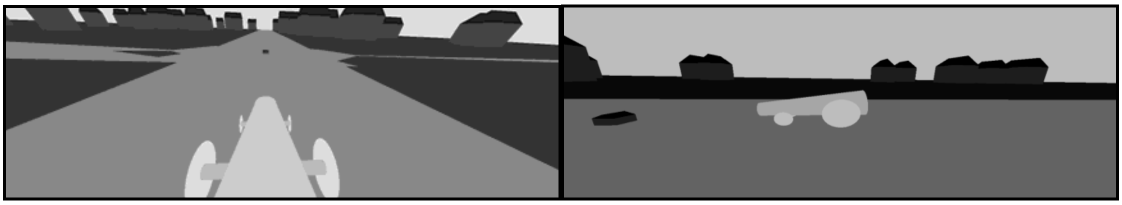
The microscopic modeling approach leads us representing each traffic actor as member of a determined list (organized as queue) [3, 4]. The list's elements store the main state variables and attributes that characterize (at a particular moment) vehicles in traffic scene, most important being their position (in 3D Cartesian coordinates), speed, acceleration, direction (3D Cartesian coordinates), bodywork length and color.

Specific functions apply on all lists, implementing the kinematic laws accordingly with the above core mathematical model, particular lists entries management rules (in order to create/delete the traffic actors from the studied scene) and graphical rendering procedures. As remark, although these management framework's functions are called in a synchronous manner by the simulator engine, they still keep the pseudo-asynchronous vehicles behavior, due to the independent kinematic rules implementation.

Case Study – Simulating a Simple Crossroad

In order to test the proposed framework's functionality, it was used in implementing a traffic simulator for simple crossroads (intersections between two one-way tracks), with traffic lights.

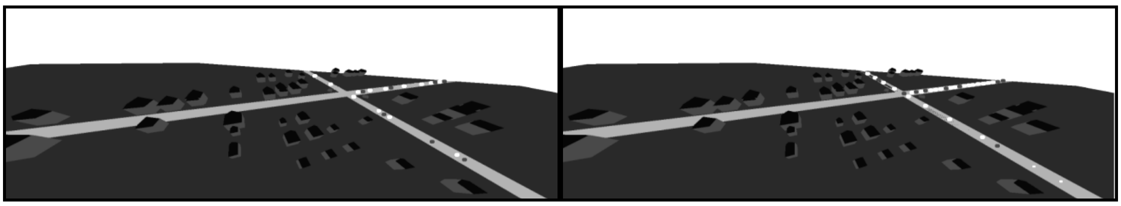
Figure 1 shows the functional flexibility of the OpenGL interface, allowing a free choice of the viewpoint over the traffic scene. The zooming factor and viewing angle may also be configured.



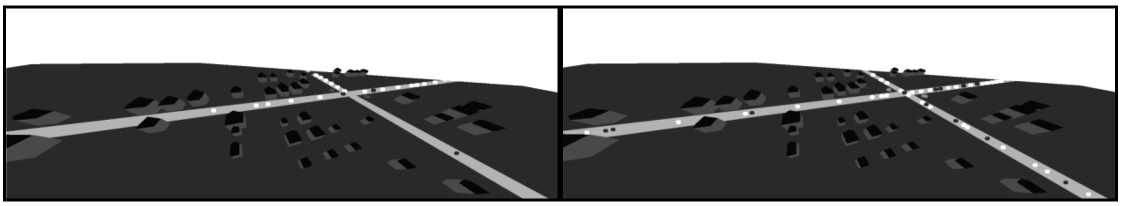
a **b**

Fig. 1. Different views allowed by the graphical interface:
a – from driver's viewpoint; b – car-aside view.

Next figure demonstrates that the mathematical model above allows an appropriate simulation, based on the proposed traffic example.



a **b**



c **d**

Fig. 2. Traffic situations: a – waiting queue at red signal on the right to left road;
b –the same as (a), but very close to a traffic jam; c – waiting queue at red signal
on the top to down road; d – priority-driven traffic.

Figure 2.a shows the case when the vehicles on the right to left road wait in a queue at the red light (while the top to down road is open to traffic). The same situation is depicted in figure 2.b, but with an increased traffic flow; one can see a significant high-density waiting queue, which could be very close to a traffic jam. By changing the traffic signals, figure 2.c shows a corresponding switch on the crossroad, now the waiting queue being located on the top to down track. The final example (figure 2.d) depicts the case when both traffic lights are in blinking yellow state. By considering the right to left road as having priority, the vehicles running from top to down are supplementary driven by the gap acceptance algorithm (shortly outlined above).

Conclusions

This work presented a framework oriented to controlled traffic dynamic modelling and simulation. Making use of modern software resources (Microsoft .NET, GLUT/OpenGL), the simulator implemented by authors is a good tool for advanced studies in this field. Starting with an overview on the used simulation techniques, the paper presented the building-up outline for such a framework (core model formulation, software encoding and implementation overview). At final, illustrative results for simulator's functionality were briefly presented.

References

1. Dan, F., Mureşan, R. – *Algoritmul de optimizare R-D pentru intersecţii*, Universitatea Tehnică Cluj-Napoca, 2002
2. Jones, S.L., Sullivan, A.J., Cheekoti N., Anderson, M. D., Malave, D. – *Traffic simulation software comparison study*, UTCA Report 02217, University of Alabama, 2004
3. Malik, D. S. – *C++ Programming: Program Design Including Data Structures*, Course Technology, 2008
4. Miller, A., Ford, L. F. – *Microsoft Visual C++ 2005 Express Edition Programming for the Absolute Beginner*, Course Technology, 2005
5. Queloz, P.-A. – *Modèle de trafic routier et simulateur massivement parallèle*, Master dissertation, University of Geneva, 1995
6. Shepherd, G. – *Microsoft ASP.NET 3.5 Step by Step*, Microsoft Press, 2008
7. Shreiner, D., Woo, M., Neider, J., Davis, T. – *OpenGL® Programming Guide: The Official Guide to Learning OpenGL®, Version 2.1 (6th Edition)*, Addison-Wesley, 2007
8. Yukawa, S., Kikuchi, M., Tadaki, S. – *Dynamical Phase Transition in One Dimensional Traffic Flow Model with Blockage*, Journal of the Physical Society of Japan, 63, 1994

O platformă modernă de modelare şi simulare în regim dinamic a traficului auto

Rezumat

Modelarea şi simularea sunt instrumente deosebit de puternice asociate studiului dinamicii traficului auto, proces caracterizat de o extraordinară complexitate, datorată în primul rând interacţiunilor existente între entităţile componente. Această lucrare descrie stadiul actual de dezvoltare a unei platforme destinată modelării matematice şi simulării dinamicii traficului rutier. Bazat principial pe mediul Microsoft .NET în conjuncţie cu resursele puse la dispoziţie de bibliotecile OpenGL şi GLUT, simulatorul implementat de autori se anunţă ca un mijloc de studiu adecvat domeniului propus, de la dezvoltarea, şi validarea modelelor de simulare până la testarea regulilor de management al traficului.