Traffic simulation for interactive applications

*Luis Loureiro, Instituto Superior Técnico*

**Abstract** — The road traffic simulation is a crucial component for interactive applications, entertainment and educational purposes, such as the driving simulators. The objective is to create realistic and plausible situations, where the human player can interact with other agents while driving his virtual vehicle. The key to a realistic motorized vehicle simulation is the sub-microscopic model, where the decision process is based on the physical, mechanical characteristics of the vehicle and the human driver mental processes. This model also allows the interaction with physics simulation engines to improve the realism of the simulation. The actions of the vehicles and pedestrians are based on steering behaviors specified by Craig W. Reynolds and adapted for the dynamics and physics properties of the agents.

**Index Terms** — Traffic Simulation, sub-microscopic, agents, vehicles, mechanics, pedestrians.

I. INTRODUCTION

The road traffic is diverse and complex system, which allows the transport of goods and the movement of people through motorized, un-motorized vehicles and pedestrians on roads.

This paper will focus on motorized vehicles and pedestrians because they are the most common elements in urban environments. The vehicles traffic simulation can be achieved through the microscopic models or macroscopic models (Sewall, Wilkie, Merrell, & Lin, 2010).

The microscopic model allows the individual simulation of each vehicle with a mathematical expression which mimics the vehicle dynamics. The macroscopic models are based on equations for averaged quantities, such as the vehicle density or the average flux along a section of road.

Although these models allow the simulation of road traffic, they are inadequate for realistic physics simulation, since they replace the physics equations with their own proposed models.

In order to improve the simulation reliability a sub-microscopic model is used in this study. This allows the vehicle control to be processed in the same way as the human driver interacts with the automobile, through the throttle pedal, brake pedal, steering and the gearbox.

The sub-microscopic model described is this paper is an attempt to map the mental processes used to drive a motorized vehicle, by mathematical equations that model behaviors for agents driving.

The pedestrians are simulated through the Craig W. Reynolds steering behaviors and are used to improve the simulation reliability.

II. STRUCTURE

Each agent is simulated through three crucial components: physics, sensorial and behavioral.

The physic component is responsible for supporting the agent rigid body simulation, using the Newton-Euler equations for dynamics. This component also provides an additional layer for mechanics simulation in the vehicles.

The sensorial component allows the perception of other agents and road elements existing in the virtual world. Each sensor has specific characteristics which affect the way the sensor perceives. These characteristics are related to some human features such as the field of view, reaction time and maximum perceived distance. For each perceived element type is given a dedicated sensor and inter-sensor dependencies, to improved computational resources usage.

The behavioral component is responsible for agents actions control, using the feedback received from the sensors and the physical component. Each action refines the vehicle behavior (Fig. 1), allowing a correct and safe vehicle simulation.

![Fig. 1 - Vehicle behavior execution sequence.](image-url)
The simulation runs in a QuadTree (Finkel & Bentley, 1971) and is subdivided according to agent’s perception characteristics to improve the speed of the search queries. In the deepest tree leafs it’s stored the agents paths, agents and road elements such as the traffic signals.

The agent generic structure is then formed by a group of sensors, behaviors, parameters and current state, which are specified through classes-based programming. This structure also facilitates the agent capabilities scalability, adding or removing features easily (Fig. 2).

**III. Modeling**

A. Physics Modeling

The agent physics component allows a realistic rigid body motion simulation using the Newton-Euler equations, converting the applied force to an angular and linear motion.

The pedestrian can be considered a particle steered through the world according to the applied forces to the rigid body. The stopping distance is null, since the pedestrian top speed is lower compared to a vehicle speed.

The motorized vehicle requires an extra layer of complexity for engine propulsion simulation, in order to convert the throttle pedal action in mechanical force to be applied to the vehicle rigid body.

1) Motorized vehicle transmission

The main components used in the motorized vehicle for the propulsion system are the engine, clutch and gearbox (Fig. 3). The engine can be simulated through a curve which gives the torque in relation to engine RPM.

The clutch is responsible for the smooth torque transition at gear shift event. The critical issue is when the vehicle starts moving, because the wheels are stopped and the gearbox axel rotations are zero. If the clutch plates are joined the motor rotations would be null. To avoid this, the clutch slips to allow a progressive torque passage, while the vehicle speed isn’t enough to achieve the minimum engine rotations. In this phase the engine rotations are calculated directly through the throttle pedal pressure.

In the normal situation, when the vehicle speed is sufficient to keep the engine running, the throttle pedal adjusts the amount of torque transmitted.

The gearbox acts as a torque and RPM converter allowing an efficient use of engine power.

![Agent generic UML structure.](image1)

**Fig. 2 - Agent generic UML structure.**

![Model of vehicle transmission simulation.](image2)

**Fig. 3 - Model of vehicle transmission simulation.**

2) Friction Forces

The friction forces applied to the vehicle body have an important effect on motion, since they are related with the aerodynamic, rolling and gravity forces.

The aerodynamic friction force (1) is related to the motion of the vehicle body through a gas (\( \rho \)-Gas Density, \( A \)-Vehicle Frontal Area, \( c \)-Vehicle Form Factor) and it’s proportional to the squared speed of the vehicle (\( v \)).

\[
F_{\text{aerodynamic}} = \frac{\rho Av^2c}{2} \quad (1)
\]

The rolling friction (2) is related to tire deformation (\( \mu_g \)) and the gravity force is related with vehicle inclination. For simulation purposes it’s only considered the aerodynamic and rolling friction, since the simulated world is flat.

\[
F_{\text{Rolling}} = g\mu_g \quad (2)
\]

The integration of the equations (1) (2) will be helpful for the braking behavior, allowing the prediction of the needed time to slowdown the vehicle, without using the throttle or brake.

\[
v_{\text{front \ vehicle}} = v_{\text{vehicle}} - \int_0^t \left( \frac{\text{dx}}{\text{dt}} \right)^2 \frac{\rho Av^2c}{2m} + g\mu_g \, \text{dt} \quad (3)
\]

This equation (3) isn’t solvable analytically and it’s necessary to use the Euler integrator with adaptive steeps to speed up the solving.

3) Braking Distance

The vehicle braking distance (4) depends on vehicle speed and tires friction factor (\( \mu_c \)). This is an important value because it affects the limits of
perception for each agent, allowing a safe detection of the agents before the collision is inevitable.

\[ d_f = \frac{v^2}{2\mu g} \quad (4) \]

The tire friction factor used is 0.9 which corresponds to semi-new tire on a dry tar road (Robert Bosch GmbH, 2002).

### B. Sensorial Modeling

The sensorial model defines how the agent’s sensors perceive the environment and how the elements and agents should be arranged in the world to lower the queries cost.

The following points address the perception latency, perception distance, word dimensioning, road elements and the used sensors.

1) Perception Latency

Due to internal and external factors while driving a vehicle, each human driver has different reactions times. This latency to perceive a certain event or obstacle can be used in simulation favor to improve the performance, using periodic search queries with latency bigger than the simulation step latency. This concept can be achieved only by taking in count the effect of the latency \((\tau_p)\) in the query perception distance \((5)\).

\[ d_p = v\tau_p \quad (5) \]

The update latency used is 500 ms which is between 300 ms and 1700 ms, typical values for human reaction time (Robert Bosch GmbH, 2002).

2) Perception Distance

The perception distance is an important value which specifies the maximum perceived distance for each agent according to vehicle speed \((v)\), reaction time \((\tau_p)\) (sensor latency), gap between vehicles \((d_{gap})\), maximum element radius \((d_{rmax})\) and safety factor \((f_{safety})\). This equation \((6)\) is based in the worse driving case, which requires the longest braking distance to avoid colliding with an immobilized obstacle on the road.

The safety factor is a multiplier that increases the vehicle braking distance, allowing a smoother braking profile.

\[ d_{perception} = d_f(v, \tau_p) + d_f(v, \mu_v) * f_{safety} + d_{gap} + d_{rmax} \quad (6) \]

3) World Dimensioning

The world QuadTree should be dimensioned correctly to allow the detection of the agents and elements within a certain distance and with enough time to avoid accidents. All the deepest leaves should be connected according to Morton scheme, allowing the vehicles to search in the neighbor leaves.

The criteria to resize the QuadTree deepest leafs is the perception distance of the fastest vehicle with the biggest safety factor. This allows a vehicle in the frontier, with other leaf, to perceive correctly any agent or element.

After calculating the leaf minimum size, the word space is resized to fit all elements and agents according to a minimum number of cells multiple of four. With this approach it’s possible to calculate the needed QuadTree depth.

![World resizing to fit all agents and elements.](image)

4) Road Elements

The simulated world layout is designed in Blender and loaded in the simulation program with the help of bXporter SDK\(^3\).

In Blender environment, it’s specified the tracks for the vehicles and pedestrians through Bezier curves; road signals and crossroads are specified through simple 3D models with specific nomenclatures, to help the identification during the simulation loading process.

5) Sensors

a) Pedestrian Sensors

The pedestrian sensors are responsible for path and other pedestrian’s detection. The path detection allows the pedestrians to steer along the current path and permits the circulation in both directions over the same path. The pedestrian’s detection has the capability to detect up to six pedestrians and a field of view of 180°.

b) Vehicle Sensors

The vehicle sensors are responsible for path detection, path selection, road signals and light signals detection, crossroad detection and other vehicles detection.

Additionally it’s used a feedback sensor to avoid the vehicles forgetting the crossroads. When a vehicle detects a crossroad, it uses the path selection sensor to choose the next path and processes which direction indicator light should be on. If the vehicle needs to brake due to a queue, the perception distance reduces and the crossroad is wiped from the sensor. This may lead to an irregular direction indicator lights alternation, which may confuse the human player.

It’s also used a sensor to store the priority and top speed limit when the vehicle passes or gets enough near to the signal.

### C. Behavioral Model

a) Pedestrian Behavioral Model

The pedestrians use two behaviors to steer along the designated paths and avoid collision with other pedestrians.

The collision avoidance behavior is a mix of repulsion forces (Reynolds, 1987) with unaligned collision detection(Reynolds, 1999), selecting only the repulsion vectors from the agents which the collision will occur in the future.

It’s also used the obstacle avoidance behavior to steer away from the nearest threat. When the collision is unavoidable, the agent reduces speed.
b) *Vehicles Behavioral Model*

The vehicle uses a group of behaviors to ensure the safe driving; these must be executed in order to avoid conflicts. For example, the throttle behavior shouldn’t be executed after the collision avoidance or priority controller, avoiding throttling the vehicle in a dangerous situation.

\[
\frac{dv_{vel}(t)}{dt} = (v_{Des} - v)\beta (7)
\]

(1) **Throttle Controller**

This behavior controls the throttle integration (7) according to minimum value (\(v_{Des}\)) of the vehicle top speed, speed obligations and front vehicle speed.

(2) **Brake controller**

The brake controller isn’t implemented as a behavior class, but provides the breaking control decision in the behaviors which require it.

The decision process involves the vehicle speed, obstacle speed, inter-vehicle distance, minimum gap, safety factor plus other physical parameters.

The basic concept is controlling the brake only when the friction forces applied on vehicle rigid body aren’t enough to stop or reduce the velocity safely. Otherwise, if the integration of the equation (3) states that it’s possible to reduce vehicle speed in less time than the expected time to the collision it releases the throttle pedal.

If the brakes are needed, it is defined a two zone operation. If the vehicle is within the critical braking distance it must brake at 100%, otherwise if it’s within the safe braking distance it will brake more smoothly (Fig. 5).

The equation (8) defines the critical brake distance as the maximum value of differences between the braking of the two vehicles (the simulated and the obstacle, which can be another moving vehicle) and the vehicle distance traveled during the reaction time.

\[
d_{\text{critical distance}} = \max \{d_{\text{vehicle}} - d_{\text{obstacle}}, d_{\text{vehicle}} - d_{\text{OB}}\} (8)
\]

The safe distance is the critical brake distance multiplied by the safety factor.

With the critical brake distance and the safe distance it is possible to define an expression for brake control (9) as shown in Fig. 5.

\[
\text{brake} \% = \frac{d_{\text{safe distance}} \times d_{\text{gap}} - d_{\text{distance}}}{d_{\text{critical distance}} - d_{\text{safe distance}}} (9)
\]

If the vehicle speed is less than the obstacle speed and the distance is smaller than the safe break distance, the vehicle also brakes to increase the vehicle inter-distance.

(3) **Priority Controller**

This behavior is responsible for applying the priority law to the vehicles when they reach a crossroad. The applied rule states that any vehicle at the right has the priority to advance, only if the priorities are equal, otherwise, they should respect the given priorities according to the near road signals.

All vehicles detected in the neighborhood which are in the same crossroad, must be checked to see if any as a higher priority or if it’s already inside the crossroad. If a vehicle is already driving inside the crossroad, all other vehicles that may collide with it, must stop even if they have higher priority.

(4) **Collision Avoidance**

This behavior is responsible for vehicle collision avoidance, breaking the vehicles before the collision is inevitable. It uses the unaligned collision detection to predict the collision with other vehicle.

To improve the vehicle collision detection, it was created a method to get the nearest distance between two vehicles, considering them as capsules. With this method it’s possible for long vehicles to circulate in the roads.

Fig. 6 - Comparison with the unaligned collision detection and collision detection based on capsules.

(5) **Other behaviors**

There are other behaviors which are responsible for vehicle direction lights, braking at the red lights and gearbox shifting.

IV. **Results**

A. **Brake Controller**

The brake controller was tested with a simulation where two vehicles executed different actions to show how the controller acts. The simulation is composed with a vehicle simulated with the brake controller and other vehicle which acts as an obstacle.

1) **Immobilized Obstacle**

The first simulation shows the impact of the safety factor during the vehicle braking while the obstacle is immobilized. It was used a safety factor of 4 and 1.5 with a 300 meters distance between the vehicles.
Fig. 7 - Braking profile with a safety factor of 4.

Fig. 8 - Braking profile with a safety factor of 1.5.

It’s shown that a bigger safety factor allows a smoother braking while the smaller value has a more aggressive braking. It’s also verifiable some instability in the braking process while the vehicle speed is low.

2) Moving Obstacle

In the second simulation the obstacle moving at constant speed of 80 km/h and the simulated vehicle is allocated at 200 meters behind at 120 km/h, with a safety factor of 1.5 and a gap distance of 2 meters.

Fig. 9 - Braking, throttle and speed profiles of simulated vehicle with a moving obstacle.

It’s shown a sawtooth wave in the throttle, due to the braking behavior which reduces the throttle to decrease the speed through the aerodynamic and rolling forces (Fig. 9).

3) Sudden braking of simulated vehicle

This simulation shows the breaking profile when the simulated vehicle brakes due to excessive speed (Fig. 10). The obstacle is moving 50 km/h, the simulated vehicle is moving at 120 km/h, 25 meters apart from the obstacle, with a safety factor of 1.5 and a minimum gap distance of 2 meters.

Fig. 10 - Braking, throttle and speed profiles for the simulated vehicle in a sudden braking event.

4) Sudden obstacle braking

This situation shows the braking profile of simulated vehicle when the obstacle brakes suddenly (Fig. 11). Both vehicles are moving at 80 km/h, distanced apart 40 meters and the simulated vehicle has a safety factor of 3.0 with a minimum guard distance of 2 meters.

Fig. 11 - Braking, throttle and speed profiles of simulated vehicle, when the obstacle suddenly brakes.

B. Performance

1) Vehicles simulation

The vehicle simulation latency rises linearly with the number of simulation vehicles. The QuadTree is configured with a top speed of 120 km/h.

Fig. 12 – Vehicles simulation performance comparison with agent’s latency at 130ms and 500ms.
2) Pedestrians Simulation

The pedestrian’s simulation latency shows the dependency on the size of the QuadTree leafs and agents latency. The Fig. 13 shows the simulation with a QuadTree configured to 120km/h and the Fig. 14 shows the same simulation with a QuadTree configured for 40km/h.

![Simulation latency of pedestrians simulation with a QuadTree leafs configured for 120km/h.](image)

![Simulation latency of pedestrians simulation with a QuadTree leafs configured for 40km/h.](image)

V. CONCLUSION

The simulation of road traffic with sub-microscopic model is a complex task, because it requires the mapping of mental processes to mathematical expressions which can be used to control the vehicle mechanical parts.

The sensor latency aspect improves the overall simulation performance and it can also be used on some computationally heavy behaviors, such as the priority controller behavior.

The brake controller allows a more realistic vehicle control, and introduces the safety factor which allows the simulation of different kinds of vehicles, for example a bus requires a bigger safety factor to avoid sudden braking. The throttle controller could be improved to avoid the sawtooth wave form by using the microscopic dynamics equation as the throttle integrator.

The QuadTree leafs resizing according to the fastest agent improves the queries speed, but penalizes the pedestrians simulation due to their low speed. To solve this issue, it’s recommended to create a specific QuadTree configured for each kind of agent.

The pedestrian’s motion in crowds can be improved by using the Finite Velocity Obstacles, for the collision avoidance.

The collision avoidance behavior provides a useful mean to prevent the collision, but it doesn’t take in consideration the road paths, which means the vehicle will probably change direction in the predicted instant. This issue causes problems in curves and crossroads preventing the vehicles from braking in advanced or braking excessively.

VI. REFERENCES


