A physically sound vehicle-driver model for realistic microscopic simulation

Dr. Marc Philipp Miska
Smart Transport Research Centre,
Faculty of Build Environment and Engineering,
Queensland University of Technology
2 George St GPO Box 2434, Brisbane QLD 4001 – Australia
marc.miska@qut.edu.au

Dr. Alfredo Nantes
Faculty of Science and Technology,
Queensland University of Technology
2 George St GPO Box 2434, Brisbane QLD 4001 – Australia

Dr. Brian Lee
Smart Transport Research Centre,
Faculty of Build Environment and Engineering,
Queensland University of Technology
2 George St GPO Box 2434, Brisbane QLD 4001 - Australia

Prof. Dr. Edward Chung
Smart Transport Research Centre,
Faculty of Build Environment and Engineering,
Queensland University of Technology
2 George St GPO Box 2434, Brisbane QLD 4001 - Australia

Word count	
Abstract	142
Main text	4765
Figures (7 x 250)	1750
Tables (0 x 250)	
Total	**6657**

ABSTRACT

In microscopic traffic simulators, the interaction between vehicles is considered. The dynamics of the system then becomes an emergent property of the interaction between its components. Such interactions include lane-changing, car-following behaviours and intersection management. Although, in some cases, such simulators produce realistic prediction, they do not allow for an important aspect of the dynamics, that is, the driver-vehicle interaction. This paper introduces a physically sound vehicle-driver model for realistic microscopic simulation. By building a nanoscopic traffic simulation model that uses steering angle and throttle position as parameters, the model aims to overcome unrealistic acceleration and deceleration values, as found in various microscopic simulation tools. A physics engine calculates the driving force of the vehicle, and the preliminary results presented here, show that, through a realistic driver-vehicle-environment simulator, it becomes possible to model realistic driver and vehicle behaviours in a traffic simulation.

INTRODUCTION

As far as time is concerned, traffic simulator can be *discrete* or *continuous*, depending on whether the state of the system is sampled at discrete or continuous intervals. According to the level of detail requested by the simulation, approaches are in general classified as either *macroscopic*, *mesoscopic* or *microscopic* (Tango, 1997) In microscopic traffic simulators, the interaction between vehicles is considered. The dynamics of the system then becomes an emergent property of the interaction between its components. Such interactions include lane-changing, car-following behaviours and intersection management.

Although, in some cases, such simulators produce realistic prediction, they do not allow for an important aspect of the dynamics, that is, the driver-vehicle interaction. Caccibue et al. are investigating driver-vehicle-environment (DVE) interaction models that can predict the human behaviour under different traffic scenarios. According the authors, such models will allow to prevent erroneous and risky manoeuvres, as well as implement means of prompt intervention. The Adaptive Integrated Driver-Vehicle Interface (AIDE) project[1] has been set up to realize the DVE concepts in order to build safer road vehicles.

In this paper, we introduce a physically sound vehicle-driver model for realistic microscopic simulation, grounded on the DVE paradigm. We postulate that such a new scheme will enhance accuracy of current traffic models, thus broadening the scope within which traffic simulation can take place.

THE ENVIRONMENT-DRIVER-VEHICLE INTERACTION

In current microscopic traffic models, the traffic dynamics is decided by the simulator through constraints and rules applied to each vehicle. These include, speed limitations, source and destination points, car-following, lane changing and road intersection policies. The simulator then generates the (continuous or discrete) time trajectories for each vehicle and produces (numerical and/or visual) results about the evolution of the simulation (Tango, Montanari et al. 2007).

As observed by Härri et al. (2006), however, most traffic models available presently neglect several important building blocks. These are the ones determining the feedback from vehicles to the driver (e.g. actual speed, acceleration, deceleration, forces, moments and vibrations), and the ones synthesizing human driving patterns (e.g. preferred path, driving style, and context-dependent path planning), amongst others. Including such blocks may significantly increase the realism of the simulation, as pointed out in various research papers (Harri, Filali et al. 2006; Tango, Montanari et al. 2007; Casucci, Marchitto et al. 2010).

In the model we propose, the behaviour of each vehicle is controlled by a *driver module* which constantly receives information from both the *vehicle module* and the *environment module*. The driver actions influence the dynamics of the vehicle, which in turn may influence future driver choices. The environment becomes a continuous stream of information for all vehicles in the simulation. In particular, the environment information includes the status of all vehicles (e.g. model, position, velocity, acceleration and yaw angle), roads (e.g. lane width, radius, inclination angle and surface properties), obstacles (e.g. type, shape and pose), traffic lights (e.g. position and colour) and road signs (e.g. type and position). This type of simulation is known as Driver-Vehicle-Environment paradigm (Cacciabue, Re et al. 2007).

---

[1] www.aide-eu.org/

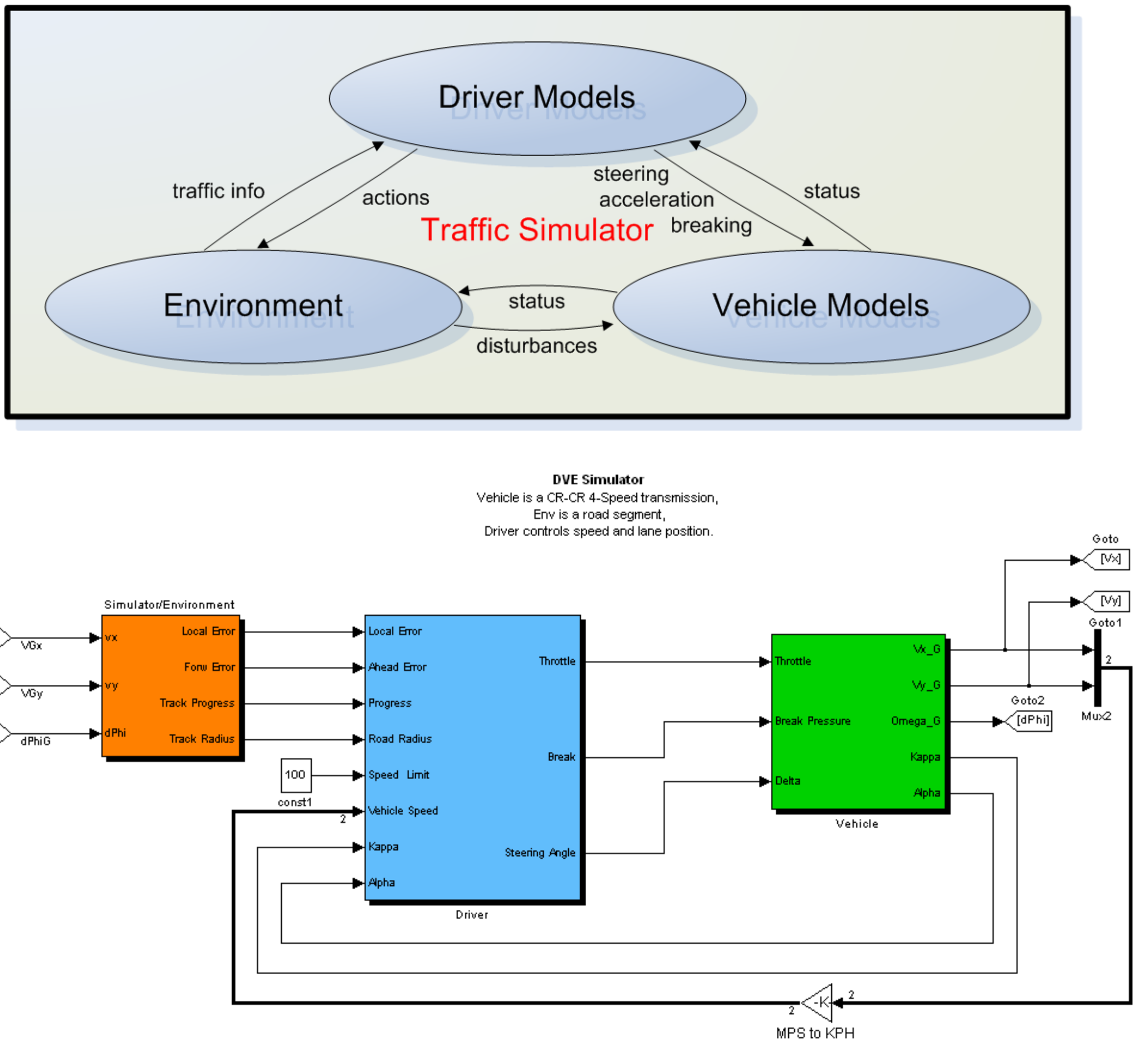

Figure 1: Schematics of the DVE model (top) and the DVE traffic simulator prototype proposed in this paper (bottom).

THE VEHICLE MODEL

The vehicle model used in our experiments is quite comprehensive. The lateral and longitudinal dynamics of the vehicle (assumed to be a rigid body) are governed by the following set of equations:

$$m(\dot{u} - \dot{\psi} v) = \sum_{\forall i} F_{x_i} \cos(\delta_i) - \sum_{\forall i} F_{y_i} \sin(\delta_i) - \frac{1}{2}\rho v^2 S C_x - mg\sin(\beta)$$

$$m(\dot{v} - \dot{\psi} u) = \sum_{\forall i} F_{x_i} \sin(\delta_i) + \sum_{\forall i} F_{y_i} \cos(\delta_i) + \frac{1}{2}\rho v^2 S C_y + mg\sin(\beta_t)\, F_{y_e}$$

$$J_z \ddot{\psi} = \sum_{\forall i} F_{x_i} \sin(\delta_i) x_i + \sum_{\forall i} F_{y_i} \cos(\delta_i)\, x_i - \sum_{\forall i} F_{x_i} \cos(\delta_i) y_i + \sum_{\forall i} F_{y_i} \sin(\delta_i) y_i + \sum_{\forall i} M_{z_i} + \frac{1}{2}\rho V_r^2 S l C_{M_z}$$

Here, m is the total mass of the vehicle, and u and v the longitudinal and lateral velocities, respectively, in the direction of the local x and y axes (i.e. the axes of the non-inertial reference frame). $\ddot{\psi}$ represents the angular vehicle acceleration and J_z is the vehicle inertia along the z direction. The index i of the summations indicates the i-th wheel. F_x and F_y are the combined driving and breaking force and rolling resistance applied to the wheel, in the x and y direction respectively.

The parameter β is the longitudinal grade angle, and β_t is the transversal slope angle of the road. δ_i denotes the steering angle of the i-th wheel . The quantities $1/2\rho v^2$ and $1/2\rho V_r^2$ express the dynamic pressure of the airflow applied to the vehicle, with ρ being the density of the air and V_r the air speed. C_x , C_y and C_{M_z} are non-dimensional longitudinal, lateral and moment drag coefficient; S and l are the reference surface of the vehicle and wheelbase respectively.

Longitudinal forces, F_x , and lateral forces, F_y , from each tire are computed as a function of the tire load, F_{z_i}, longitudinal slip, k, and sideslip angle, α. The empirical equations that we used to compute such forces are due to Pacejka (2006), and read:

$$F_x = D_x \sin\left(C_x \tan^{-1}\left\{B_x(1-E_x)\left(k+S_{h_x}\right) + E_x \tan^{-1}\left[B_x\left(k+S_{h_x}\right)\right]\right\}\right) + S_{v_x}$$

$$F_y = D_y \sin\left(C_y \tan^{-1}\left\{B_y(1-E_y)\left(\alpha+S_{h_y}\right) + E_y \tan^{-1}\left[B_y\left(\alpha+S_{h_y}\right)\right]\right\}\right) + S_{v_y}$$

where D, C, B, E, $S_h$and S_v are functions of the vertical load, F_z , applied to the wheel, and other tire parameters. The quantities k and α are the longitudinal slip and sideslip angles respectively, computed as:

$$k = V_x - r_e\Omega$$

$$\tan(\alpha) = -\frac{V_y}{V_x}$$

Here, r_e and Ω are the wheel effective rolling radius and angular velocity respectively; V_x and V_y are the forward and lateral velocity of the wheel. Finally, the front and rear tire load is determined via the following load transfer functions:

$$F_{z_f} = \frac{h_G}{a+b}\left(\sum_{\forall i} F_{x_i}\cos(\delta_i) - \sum_{\forall i} F_{y_i}\sin(\delta_i)\right) + \frac{b}{a+b} mg\cos(\beta)$$

$$F_{z_r} = \frac{h_G}{a+b}\left(\sum_{\forall i} F_{x_i}\cos(\delta_i) - \sum_{\forall i} F_{y_i}\sin(\delta_i)\right) + \frac{a}{a+b} mg\cos(\beta)$$

where F_{z_f} denotes the load applied to the front tires, and F_{z_r} the load applied to the rear tires. The parameter h_G refers to the height of the centre of gravity (CG) of the vehicle with respect to the road surface; a and b denote the horizontal distances from CG to the front and rear axle respectively. Figure 2 gives an overview of the vehicle model.

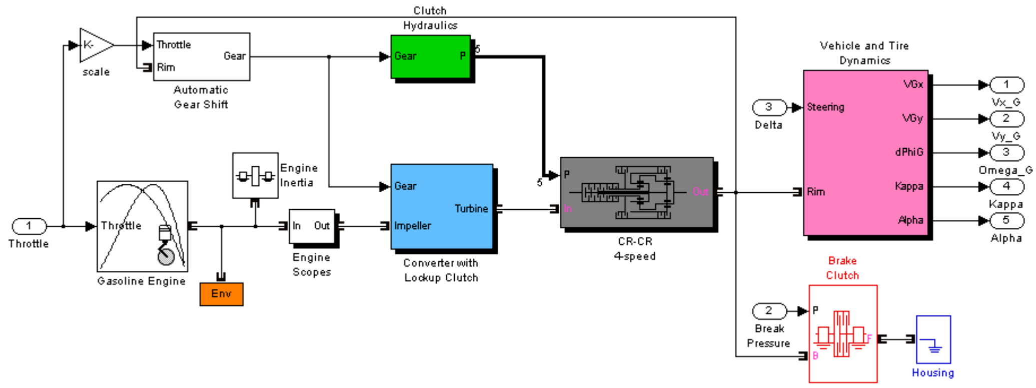

Figure 2: Vehicle model. The model includes gasoline engine, torque converter, automatic gear shift, CR-CR 4 speed system and vehicle body and tire dynamics

Acceleration and breaking torque is generated by a vehicle engine module, upon throttle and breaking stimuli from the driver controller. The torque is then transferred to the wheels. The engine module implements a CR-CR 4-speed gasoline engine, which includes torque converter, clutch and transmission system. In order to implement gear shifting, an Automatic Transmission Control (ATC) mechanism was used (1998). The engine model used was the one available from the Matlab® SimDriveline™ toolbox. Figure 2 depicts a (Matlab® Simulink®) schematics of the engine model.

THE DRIVER MODEL

Overview

The purpose of the driver model is to control the vehicle in order to carry out goals and actions planned by the driver, as the driver-vehicle-environment interaction unwinds. As Cacciabue et al. pointed out (2007), driving can be regarded to as a set of goals, interconnected by functional dependencies. In framework initially proposed by Misky (1975), and implemented in the SSDRIVE simulator (Casucci, Marchitto et al. 2010), tasks are enabled by *pre-conditions* and disabled or closed by *post-conditions*. When a task is enabled, it implements a sequence of *elementary* actions. For example, the task of attaining higher speed is enabled by the pre-condition that there is no ahead obstacle with lower speed. The sequence of elementary actions is 'accelerate', 'check speedometer' and 'maintain speed'. The post-condition consists of reaching the selected speed. The tasks are continuously generated according to purpose and knowledge base of the driver and the perception and interpretation of the information coming from the vehicle and the environment. Some tasks do not require preconditions to be launched, but they are rather carried out permanently in the DVE interaction. These tasks are referred to as *permanent* and concern higher priority manoeuvres, such as the maintenance of longitudinal and lateral safety margins. In this work, we focus on the elementary driving actions (i.e., increase/decrease throttle and brake pressure, steering), to maintain the desired speed. More complex tasks will be object future research.

Speed and Steering Control

Our current driver model includes speed and steering control mechanisms, implemented through fuzzy logic controllers; these are the speed and steering controllers. The speed controller is used to determine the desired speed for the vehicle, upon the stimuli received from both the environment and the car. This controller is, in fact, composed by two sub-systems; the one computing the intended speed (IS); and the one computing the throttle and breaking pedal position (TB). Finally, the steering controller (ST) determines the angular position of the steering wheel, in order to direct the vehicle towards the intended orientation.

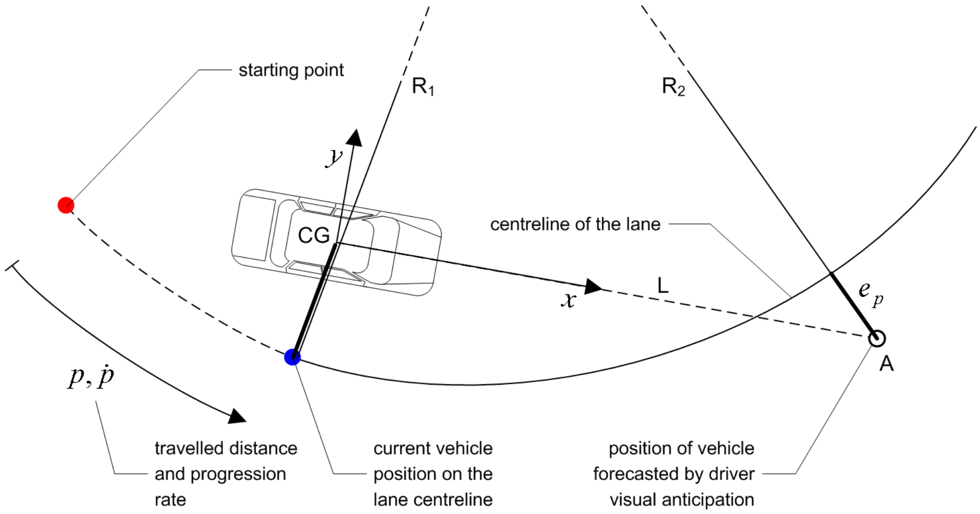

Figure 3: Illustration of the parameters used by the driver controller. The centre of gravity of the vehicle is denoted by CG; L is the visual anticipation distance. R_1 and R_2 are the road radii at the CG and A position respectively.

In order to determine the intended speed, a number of parameters are taken into account, both from the environment and the vehicle. Specifically, the simulator computes the inputs to the IS controller, according to the topology of the road driven by the vehicle (i.e. track points position, tangent vectors, normal vectors and radii); to vehicle position and orientation with respect to the environment coordinate system; and to the sideslip angles from the vehicle. These parameters shall be described shortly.
The other component of the speed controller is the one determining the throttle and breaking pedal positions of the vehicle. This TB mechanism receives inputs from the environment, the vehicle and IS. The desired speed is computed as the minimum value between the speed limit of the driven road, and the intended speed computed by the IS system. In order to reduce longitudinal and lateral sliding, the TB controller also accounts for longitudinal slip and sideslips angles.

Intended Speed
In our current implementation, the intended speed is computed so as to minimize the time to reach the intended destination, while ensuring safety conditions. In other words, the vehicle is driven at the minimum between the *highest allowed speed* and the *maximum safe speed*, computed as:

$$IS = \min(AS, SS)$$

In our formulation, the maximum allowed speed AS is simply the speed limit of the road driven by the SS is the maximum safe speed. The latter, is a function of environment and vehicle variables, such as *road radius*, R, *position error,*e_p , *sideslip angle*, α, and *trip progression rate*, $\dot{p}$. Two road radii are measured; one (R_1), at the position of the vehicle centre of gravity (CG), and the one (R_2), at a distance L ahead from vehicle; the road radius R is then computed as $\min(R_1, R_2)$. The position error is the distance from the CG to the lane centreline, computed at the visual anticipation distance L. This error is positive if the vehicle position is on the left hand side of the lane centreline; and negative otherwise. As it has been (Genta 1997; Rajamani 2006), bigger visual anticipation distances produce more stable systems[2]. The angle is a function of the forward and lateral velocity of the wheels, as explained before. Finally, the trip progression rate, $\dot{p}$, is the time derivative of the vehicle distance, from the starting point. These quantities illustrated in

[2] Bigger anticipation distances, L, entail better damped zeros of the transfer function between the steering angle and the lateral position of the vehicle with respect to the centre line of the lane.

Figure 3.

Our fuzzy map for the maximum safe speed reads:

$$SS = f(R, e_p, \alpha, s)$$

$$SS \begin{cases} High & \textbf{if } R \text{ is large } \textbf{and } |e_p| \text{ is small } \textbf{and } |\alpha| \text{ is small } \textbf{and } s \text{ is positive} \\ Low & \textbf{if } R \text{ is not large } \textbf{or } |e_p| \text{ is not small } \textbf{or } |\alpha| \text{ is not small } \textbf{or } s \text{ is negative} \end{cases}$$

Here, the membership functions R, e_p, α and $\dot{p}$ are time-independent Gaussian curves. The output membership function consists of two triangular functions, denoting Low and $High$ speed. Implication, in the fuzzy logic controller, is performed through the *min* (minimum) method; aggregation is achieved via the *max* (maximum) method. Finally, defuzzification is carried out by using the *centroid* rule.

In the work presented by Cacciabue et al. (Cacciabue, Re et al. 2007), the intended speed is a function of driver and environment parameters such as the driver experience and attitude, the visibility and complexity of traffic and the condition of the road. Driver experience and attitude, as well as road condition are independent, discrete variables; the task demand, depends on the driver visibility and traffic complexity. We believe that the driver status is an important aspect to consider when computing the intended speed. In our current implementation, however, this component has been neglected.

Throttle and Breaking

After determining the intended speed, the velocity of the vehicle is adjusted accordingly, by acting on throttle and breaking pedals. As for the IS controller, the Throttle and Breaking (TB) system implements a fuzzy mapping between inputs and control variables. The inputs to the TB system are the *speed error,* e_s, the *speed error rate,* $\dot{e}_s$, the *sideslip angle,* α, and the *longitudinal slip,* k. The control variables are the acceleration and breaking signals. The speed error represents the difference between intended and actual vehicle speed. The speed rate is the time derivative of the vehicle speed. The longitudinal slip, expresses the difference between the vehicle longitudinal velocity at the axel of the wheel, and the equivalent rotational velocity of the wheel. Finally, the sideslip angle denotes the angle between wheel orientation and the orientation of the velocity vector of the wheel, as described before. The TB function, is thus expressed as:

$$TB = f(e_s, \dot{e}_s, \alpha, k)$$

and implemented through the rule set:

$$TB \begin{cases} Break\_Full & \textbf{if } e_s \text{ is big\_negative} \\ No\_Action & \textbf{if } e_s \text{ is small\_neg } \textbf{and } \dot{e}_s \text{ is positive} \\ Throttle\_Full & \textbf{if } e_s \text{ is big\_positive} \\ Throttle\_Half & \textbf{if } e_s \text{ is small\_pos } \textbf{and } \dot{e}_s \text{ is negative} \\ No\_Action & \textbf{if } |e_s| \text{ is small } \textbf{or } |\alpha| \text{ is not small } \textbf{or } |k| \text{ is not small} \end{cases}$$

As with the IS controller, the input membership functions are Gaussian, whereas the output functions defining the output are triangular. The output of the system becomes positive when the throttle needs to be opened and negative when a breaking action is required. Using a single output guarantees that the acceleration and deceleration pedals are not pressed at the same time. This choice is motivated by the observation that human drivers, typically, do not act on the gas brake pedals concurrently..

Intended Steering

The steering behaviour of the driver is simulated through a controller (ST) which steers the vehicle towards the intended direction, while ensuring that the distance between vehicle and centreline of the lane is kept to a minimum. To that end, a fuzzy controller is used, which maps from three measurements (variables) of the vehicle-environment interaction, to the intended steering angle. The inputs to the ST controller are the position error, e_p, and its time derivative, $\dot{e}_p$, the sideslip angle, α, and the trip progression rate, $\dot{p}$, introduced earlier. Such a mapping is then formulated as:

$$ST = f\left(e_p, \dot{e}_p, \alpha, s\right)$$

and expressed by the rule set:

$$ST \begin{cases} Steering\_Left & \textbf{if } e_p \text{ is } big\_negative \\ Steering\_Right & \textbf{if } e_p \text{ is } big\_positive \\ No\_Steering & \textbf{if } \left|e_p\right| \text{ is } small \textbf{ or } \alpha \text{ is not } small \\ No\_Steering & \textbf{if } \dot{e}_p \text{ is } negative \textbf{ and } s \text{ is } positive \end{cases}$$

The input membership functions of the steering controller are Gaussian curves; the output functions are triangular for the $Steering_{left}$ and $Steering_{right}$ signals; and Gaussian for the $No\ Steering$ action. As it can be noted, the steering angle is positive (leftwards) when the position error is big and negative, that is, when the vehicle is heading towards the right hand side of the road; and negative otherwise. The variance of the Gaussian bell is such that the area under the curve is bigger than the area under the triangular functions. This allows giving the $No\ Steering$ action a bigger weight with respect to the steering commands, provided that the *centroid* rule is used for defuzzification.

Figure 4 shows a Matlab® Simulink® implementation of our driver model. The green boxes indicate the three fuzzy controllers, that is, IS, SS and ST.

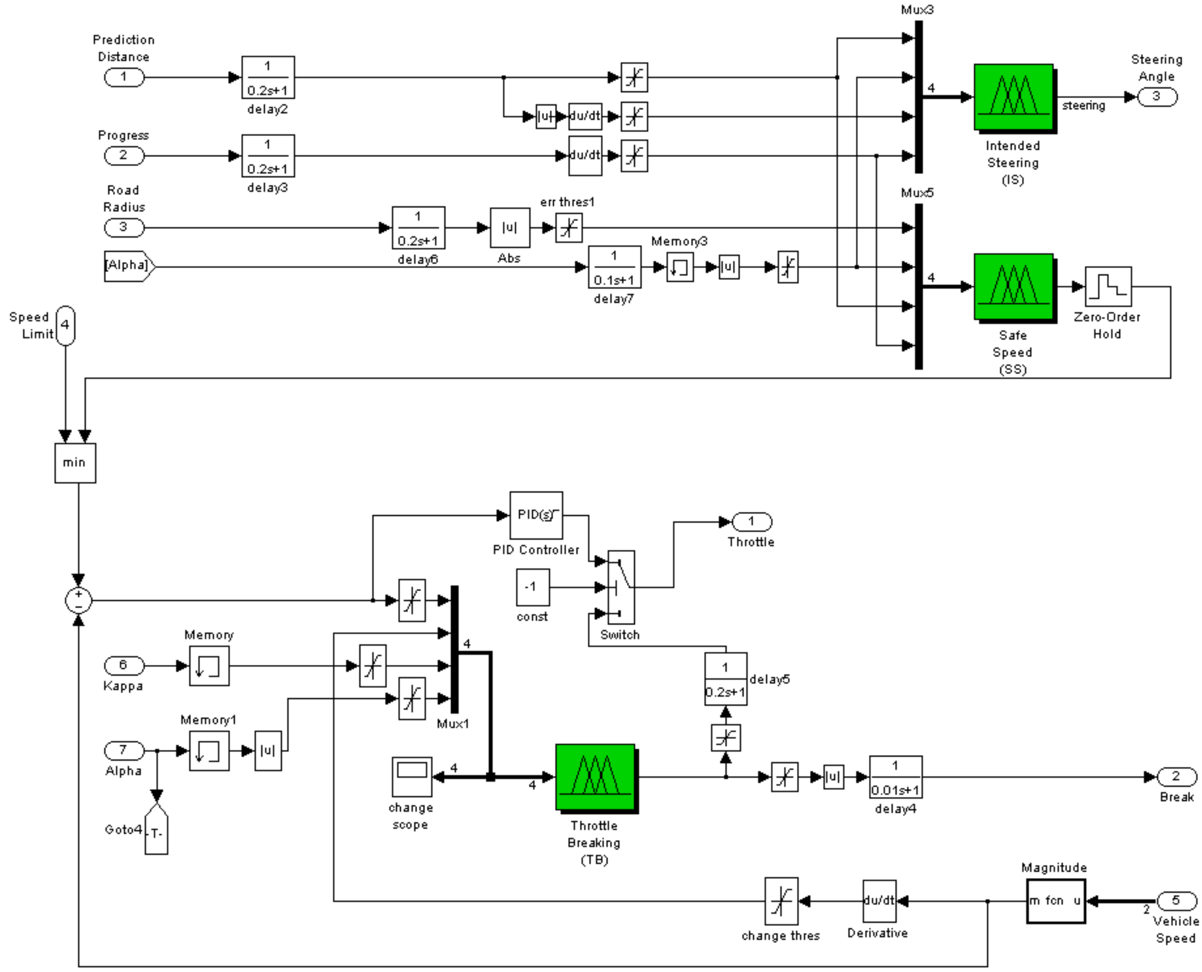

Figure 4: A Matlab (Simulink) implementation of our driver model

In driver-vehicle simulation environments, the driver is typically assumed to be a tracking system affected by delays. These, amount to the delay needed to elaborate the relevant information from the surrounding environment (*reaction time delay*), the time needed for the planned action to translate to muscular activity (*neuromuscular delay*), the time needed to perform the action (*execution delay*) and the *lead time*, to account for the human predictive capabilities (Genta 1997). Often, the lead time is neglected and all other delays are factored into one single delay τ, yielding the simple open loop transfer function:

$$\frac{y(s)}{u(s)} = K_d e^{-\tau s} \approx K_d \frac{1}{1+\tau s}$$

Where y, u, K_d, τ are respectively the output and input to the driver, the gain and the delay; s is the Laplace transform variable.

In our driver model we apply the transfer function above to all driver inputs to the IS and SS controllers, that is, the prediction distance, the trip progression rate, the road radius and the sideslip angle (see Figure 4). The inputs to the TB controller are not ‘filtered’ by the transfer function, for the TB controller receives inputs from the SS controller, to which the delay has been already applied.

THE ENVIRONMENT MODEL

A simulation environment suitable for realistic traffic simulations should allow reproducing real traffic contexts, such as environments of arbitrary complexity. At this stage of development, our prototype simulator features a rudimentary road editor whereby lane sections can be defined through *target points*. The desired lane is therefore generated by cubic *spline* interpolation (Knott 2000) of such sample points. The first and last target points introduced are considered, respectively, as source and destination points by the simulator. The editor also allows the user the selection of initial position and orientation of the vehicle, as well as visual anticipation distance, L. Once the lane has been determined, the editor then generates the data (i.e. lane tangents, normal, radii and vehicle initial status) used for the simulation. Figure 6 a, shows an example of road modelling through target points and related spline, generated by our environment editor. The environment system has been implemented in Matlab® Simulink®. As it can be observed from Figure 5, the system receives input signals (i.e. longitudinal and later velocities of the vehicle CG, and vehicle angular velocity) from the vehicle module and from the simulator (i.e. initial position of the vehicle CG, and initial vehicle orientation), and produces the related output to the driver module (i.e. position error, road radius and trip progress).

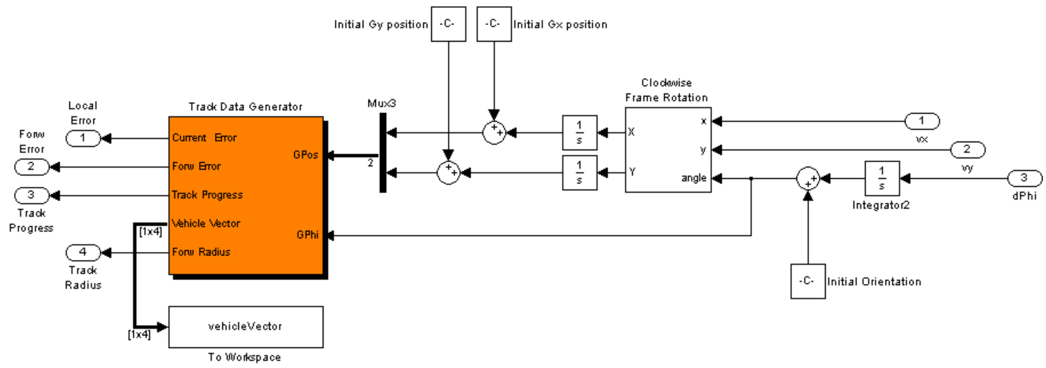

Figure 5: Environmental component of our model

CASE STUDY

The case study discussed in this section aims at demonstrating the functionality of our prototype simulator. In particular, we wish to show that different realistic vehicle dynamics can be reproduced, upon different input parameters. Since the dynamics of our simulator emerge from realistic interactions between driver, vehicle and environment, we postulate that this approach will enable the realization of realistic models. Thus, it will allow the simulation of traffic dynamics in a broad range of traffic scenarios. The examples discussed here are based on a simple path following task. The driver is to follow a pre-defined (intended) trajectory, defined through the environment editor, until the destination point is reached. All experiments are performed on the same road and with the same vehicle, though with three different driver models.

The vehicle model considered here is the *bicycle model* (i.e. a two-wheel model), extensively used in traffic simulation research (Im, Kageyama et al. 2000; Glaser, Rakotonirainy et al. 2007; Amiditis, Pagle et al. 2010). The motion of the vehicle considered is planar, that is, the vehicle travels on a flat surface ($\beta_t = 0, \beta = 0$). The road, represented in Figure 6, is a 696 metre long path, modelled through 100 segments, connected so as to interpolate the target points (red dots), through a cubic spline. The vehicle is located in the neighbourhood of the starting point. The visual anticipation distance, L, is set to 20m. Tire and vehicle parameters used in the simulation are the ones reported by Genta (1997), for a five-seat European saloon car. The relevant variables of the systems are monitored and will be discussed in the following sections:

In the first experiment considered here, the speed of the vehicle equals the speed limit specified by the environment model, for the given road section. In other words, the safe speed (SS) controller is disabled and, thus, the IS controller discussed earlier, becomes:

$$IS = \min(AS, SS) = AS$$

where AS denotes the speed limit. Throttle and brake pedals are controlled by TB controller in such a way as to minimize speed error, longitudinal slip and sideslip angles. The actual path, resulting from the simulation, is illustrated in Figure 6 b (Case 1). The speed profile, for both intended and actual speed, is shown in Figure 6 a, column 1. As it can be observed from Figure 6 c, the position error is kept at a minimum, up until the last turn negotiated after 25 seconds from the start of the simulation. From this time onwards, the position error becomes increasingly large, yielding oscillatory instabilities and strong lateral forces experienced by both the vehicle and the driver (Figure 6, column 2).

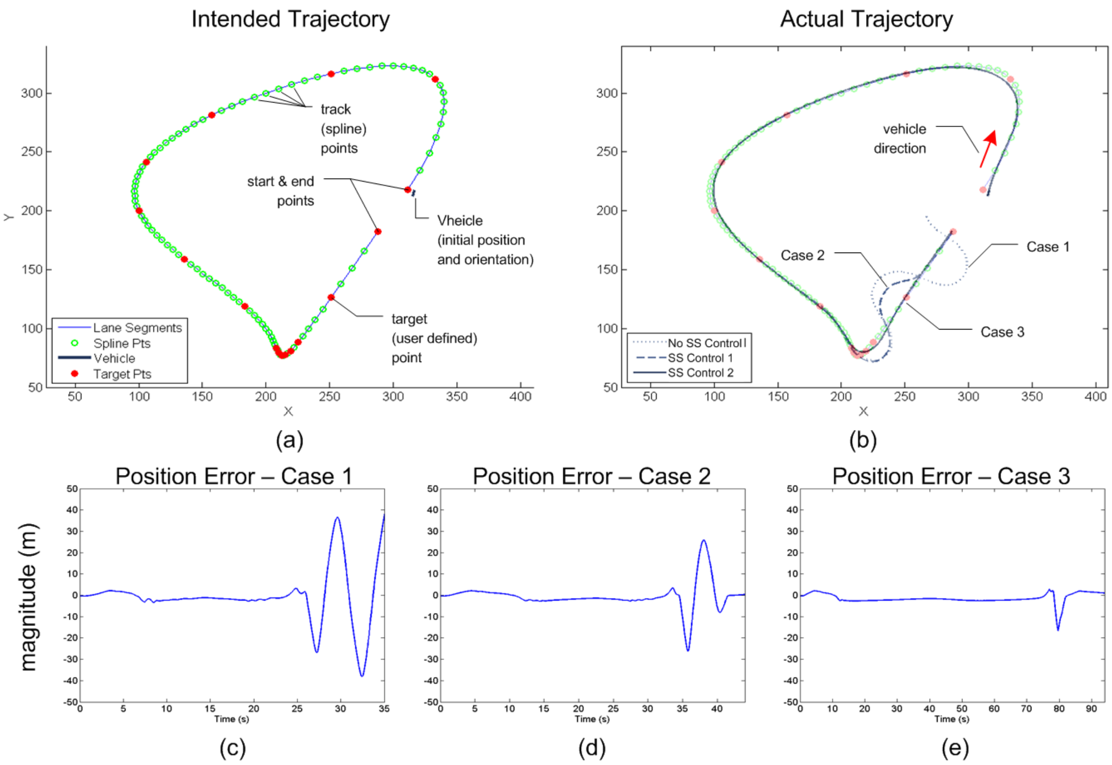

Figure 6: Example of road modelling through target points and related spline in comparison with the actual trajectory of the vehicle

In the second sample case, the intended speed is computed as :

$$IS = \min\,(AS, SS)$$

This time, the SS controller is enabled. The membership function representing the position error is a Gaussian function of zero mean and standard deviation of 40 (metres). This function produces a fuzzy controller that is quite tolerant to the position error. Also, because the safe speed depends on the radius of the road, the vehicle speed is significantly reduced in proximity of each turn (Figure 6, row b). Despite the large position error measured (Figure 6, d), the system is stable and the error is minimized, after negotiating the last sharp turn.

The third example is similar to the previous case, that is, the driver controller combines the behaviour of the IS, TB and ST system. The SS controller is enabled, but the Gaussian membership function representing the perceived position error has zero mean and a standard deviation of 1 (metre), with respect to the previous case.

The simulation outputs from this scenario are reported in Figure 6 b and e, and
row c. As it can be observed, the vehicle position and intended trajectory overlap almost perfectly. At the sharp turn, the predicted distance is significantly smaller than is case 1 and 2.

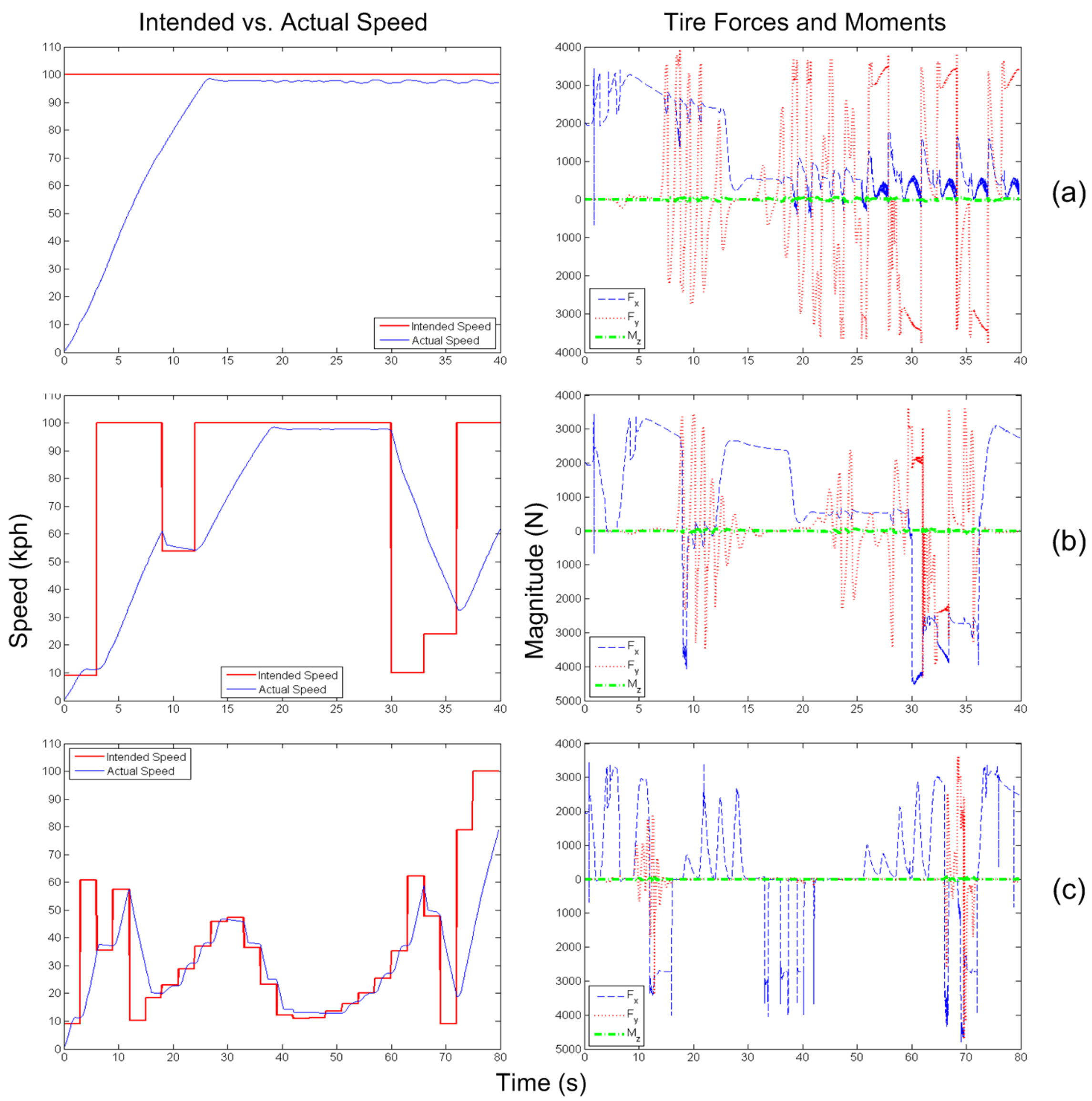

Figure 7: Simulation results from DVE system. The graphs of row (a) relate to Case *1*, and show the behaviour of the vehicle when driven at the speed limit (100 kph) with no IS control. Longitudinal forces, F_x lateral forces, F_y, and moments, M_z are also represented. Rows (b) and (c) relate to Case 2 and 3, respectively. This time, the IS controller is activated. Smaller velocities entail reduced lateral forces (dotted, red lines in the graphs of column 2).

Also, the lateral forces, F_y, are of reduced magnitude, yielding a supposedly more pleasant drive experience. Because the actual vehicle speed is almost always smaller than the speed limit, the vehicle takes twice as much of the time to reach its destination point (approximately 90 seconds for Case 3; 40 for case 1 and 2). Such a low speed (roughly 30 kph) is due to the limited grip offered by the tires of the vehicle in the bicycle model used.

DISCUSSION, CONCLUSIONS AND FURTHER STEPS

The preliminary results presented here, show that, through a realistic driver-vehicle-environment simulator, it becomes possible to model realistic driver and vehicle behaviours in a traffic simulation. The three cases analysed here show how the actual travelled path may differ from the intended trajectory, by only varying the parameters of the driver controller. For instance, the speed limit set for a road may prove not to be sufficient in order to ensure driver safety, that is, in order to maintain the lateral error to a minimum. In current traffic simulators, the speed of the vehicle is decided by the simulator, according to the current traffic conditions and the speed limit set for the environment and/or the vehicle. The physics barriers imposed by the vehicle (e.g. maximum acceleration and actual longitudinal and lateral forces), and the driver and environment behaviours are simulated through a limited number of parameters, by which the trajectory of the system is computed. As we have shown, however, the driver behaviour may need to divert from the traffic rules imposed by the system, in order to achieve the final task. The task itself may not necessarily come down to reaching the final destination point in the minimum time. Rather, it could involve meeting personal driving attitudes, such as carry out a quiet and safe, versus a sport and more risky, drive. A traffic simulator, in which the speed of the vehicles is entirely determined by pre-determined parameters (e.g. the speed limit and the velocity of the ahead vehicle), may not be able to capture behavioural perturbations like the ones presented in our experiments. On the one hand, adjusting the vehicle speed according to the ever-changing environment conditions – rather than static simulation rules – allows the modelling of more complex phenomena. This can be anything from a trajectory perturbation due to driver distractions or to complex vehicle dynamics; to a unintended acceleration/decelerations, do to friction losses. Moreover, increasing the level of realism of the system will enable the collection of more accurate information about the vehicle status, such as vibrations, understeering and oversteering conditions, amongst others (see Figure 5column 2). This additional data can then be used to predict traffic scenarios that are likely to be unpleasant or even risky drive experiences.

In agreement with Tango et al (2007), we believe that the development of more realistic traffic simulators requires a more accurate modelling of the three main actors, that is, the driver, the vehicle and the environment. In this work, we have presented a prototype simulator that is based on such concepts and that produces realistic, real-time traffic dynamics. In future work, we will extend our current bicycle vehicle model to a more realistic four-wheel vehicle.

As far as concerns the fuzzy controllers, we will incorporate a learning component to the simulator in order to determine the optimal parameters for the membership functions. Indeed, even though the membership functions used in our tests have proven to be effective in a large variety of road topologies, their parameters (i.e. means and variances of the Gaussian functions) have been tailored to the vehicle model. Through statistical learning methods (such as Expectation Maximization), the process of determining the best system parameters can be automated, via collecting enough training and validation data (i.e. longitudinal and lateral forces, self-aligning moments, prediction distance, velocity errors, longitudinal slip and sideslip angles) from various road topologies and vehicle types.

Finally, part of our future research will be directed towards the selection of the target points in the environment at run time, as the simulation dynamics unfolds. This will reflect more a realistic scenario in which the path generated by the simulator will allow for changes in the task and preferences of the driver, as well as feedback from the environment.

REFERENCES

Mathworks (1998), Using Simulink and Stateflow in Automotive Applications

Amiditis, A., K. Pagle, et al. (2010). "Driver-Vehicle-Environment monitoring for on-board driver support systems: lessons learned from design and implementation." Applied Ergonomics **41**(2): 225-235.

Cacciabue, P. C., C. Re, et al. (2007). Simple Simulation of Driver Performance for Prediction and Design Analysis. Modelling Driver Behaviour in Automotive Environments. P. C. Cacciabue. Dordrecht, Springer. **8:** 344-375.

Casucci, M., M. Marchitto, et al. (2010). "A numerical tool for reproducing driver behaviour: Experiments and predictive simulations." Applied Ergonomics **41**(2): 198-210.

Genta, G. (1997). Motor vehicle dynamics: modeling and simulation, World Scientific.

Glaser, S., A. Rakotonirainy, et al. (2007). An Integrated Driver-Vehicle-Environment (I-DVE) model to assess crash risks. the 2007 Australasian Road Safety Research, Policing and Education Conference, Melbourne.

Harri, J., F. Filali, et al. (2006). Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy. International Journal of Computer Applications, Technical Report RR-06-168, Institut Eurecom.

Im, E., I. Kageyama, et al. (2000). Construction of a traffic Simulator for an evaluation of ITS. IEEE Intelligent Vehicles Symposium 2000., Dearborn, MI.

Knott, G. D. (2000). Interpolating Cubic Splines. Boston, Birkhauser.

Minsky, M. (1975). A Framework for Representing Knowledge. The Psychology of Computer Vision. P. Winston. New York, McGraw-Hill**:** 211-277.

Pacejka, H. B. (2006). Tire and vehicle dynamics, Society of Automotive Engineers, Inc.

Rajamani, R. (2006). Vehicle Dynamics and Control. Boston, MA, Springer.

Tango, F., R. Montanari, et al. (2007). Present and Future of Simulation Traffic Models. Modelling Driver Behaviour in Automotive Environments. P. C. Cacciabue. Dordrecht, Springer. **8:** 400-428.