JOSÉ GERARDO CARRILLO-GONZÁLEZ, Ph.D.[1,2]
(Corresponding autor)
E-mail: j.carrillo@correo.ler.uam.mx
JESÚS ARÁMBURO-LIZÁRRAGA, Ph.D.[3]
E-mail: jaramburo@cucea.udg.mx
LILIANA IBETH BARBOSA-SANTILLÁN, Ph.D.[3]
E-mail: ibarbosa@cucea.udg.mx
[1] National Council of Science and Technology (CONACYT)
México City, México
[2] Department of Information Systems and Computational
Sciences, Universidad Autónoma Metropolitana,
Unidad Lerma, Av. Hidalgo Pte. 46, Col. La Estación,
Lerma de Villada, 52006, Edo. de Méx., México
[3] Universidad de Guadalajara
Av. Juárez # 976, col. centro, 44100, Guadalajara, Jalisco, México

Traffic Engineering
Preliminary Communication
Submitted: 9 Dec. 2016
Accepted: 2 Feb. 2018

ACCELERATION (DECELERATION) MODEL SUPPORTING TIME DELAYS TO REFRESH DATA

ABSTRACT

This paper proposes a mathematical model to regulate the acceleration (deceleration) applied by self-driving vehicles in car-following situations. A virtual environment is designed to test the model in different circumstances: (1) the followers decelerate in time if the leader decelerates, considering a time delay of up to 5 s to refresh data (vehicles position coordinates) required by the model, (2) with the intention of optimizing space, the vehicles are grouped in platoons, where 3 s of time delay (to update data) is supported if the vehicles have a centre-to-centre spacing of 20 m and a time delay of 1 s is supported at a spacing of 6 m (considering a maximum speed of 20 m/s in both cases), and (3) an algorithm is presented to manage the vehicles' priority at a traffic intersection, where the model regulates the vehicles' acceleration (deceleration) and a balance in the number of vehicles passing from each side is achieved.

KEY WORDS

simulation; deceleration; mathematical model; algorithm; intersection;

1. INTRODUCTION

The goal of this paper is to report a car-following model that gradually accelerates (decelerates) while maintaining safe conditions and at the same time supports time delays when refreshing the vehicles' position coordinates, that is, the data required by the model. In the ideal case no delay is present, as in [1], where two heavy-duty vehicles in motion (with identical braking capabilities) can maintain a minimum distance of 1.2 m. Besides, if there are no delays in the feedback system and optimal controls are used in the system design, then the following factors make it possible to determine the minimum relative distance: velocity, relative velocity, and vehicle characteristics. Another factor (if human behaviour is considered) is the time required by the driver to react (the analogy is the time delay of an autonomous system). In [2], a basic Intelligent Vehicle/Highway System (IVHS) is proposed. It is assumed that organizing the traffic in platoons will increase the highway capacity; however, if the distance between the vehicles in the platoon is 1 m and considering the human delay of 0.25 to 1.2 s, it is concluded that the deceleration control of the vehicle should be automatic (to decelerate immediately) and the variables of the vehicle in front must be sensed. The work presented in [3] uses a time cycle of 0.1 s for data transmission and has a similar intention to the present work. Therefore, we focus on developing a model capable of regulating the vehicles' motion (in car-following) when data are refreshed in time periods ≥1 s.

There are works that design procedures to decelerate vehicles without considering the time at which the model should be updated to avoid conflicts (as collisions) and maintain the desired behaviour. In [4], a deceleration model is introduced, with a polynomial model being proposed for cars and a dual regime model for other vehicles. Other works focus on human behaviour. In [5], the discomfort caused by longitudinal deceleration is approached and a braking comfort model is introduced to be used in car-following situations. The model is implemented according to the relation between the vehicle deceleration and the passenger comfort levels. In [6], the velocity, time gap, deceleration, and braking system response were the variables selected that influence the follower's braking response time, and it was found that the time gap has the greatest influence, with response times of 1.34, 1.67, and 2.23 s for time gaps of 1, 2, and 3 s, respectively. Then, if the objective is to imitate human behaviour with a model (under certain conditions, e.g. with specific values of the variables considered in [6]), it must support (at least) the maximum response time required (by humans) to initiate the vehicle deceleration. In [7], a headway threshold was considered to choose between free flow and a car-following regime; the former was modelled considering a random term (associated with the follower acceleration at time t), the stimulus (relative speed between leader and follower at time t-τ), and the sensitivity (a function of the follower speed, space headway, and traffic density at t-$\xi\tau$), where t is the current time, τ is the driver reaction time, and ξ is used to regulate the delay. If ξ=1, lag for sensitivity and stimulus are equal. If ξ<1, the delay for the sensitivity is less than the one for the stimulus and it also means that the driver is updating their perception of the environment during the acceleration (deceleration) manoeuvre.

We found in literature that with simulations different traffic situations are represented. In [8], an experimental platform that integrates a Traffic Simulator with Multiple Driving Simulators (TSMDS) is developed, allowing the simulation of special driving behaviour under realistic traffic conditions. Traffic situations involving autonomous driving are also approached through simulations. In [9], steering behaviours to control the vehicle's motion in a virtual environment are presented, the vehicle's motion is controlled by different behaviours, each responsible for a driving task. The collision avoidance problem is also considered with simulations. In [10], collision avoidance through a velocity correction control law is introduced. In [11], pursuit-evasion games to avoid collisions are analysed.

One of the scenarios considered to test our model was an intersection as shown in [12, 13, 14, 15]. Therefore, an algorithm has been developed which focuses on allowing equal traffic flow from each side of the entries. A common approach used to control traffic at intersections is to represent the vehicles as agents. In [16], the priority relations (defined in the Highway Code) are considered at intersections, where a multi-agent model based on opportunistic individual behaviours and anticipatory individual abilities is proposed to simulate traffic at road junctions. In [17], the vehicles are presented as agents and game decision matrices are considered to solve the vehicle priority at junctions.

The implementation of autonomous systems eliminates the need of physical human intervention, with the intention of avoiding human driving mistakes. The case presented in this paper to regulate the traffic at the intersection seeks to eliminate human intervention. The algorithm introduced is designed to regulate the vehicle priority for streets where a balance in the number of vehicles passing from each side is required and uses the proposed model to smoothly accelerate (decelerate) the vehicles. To determine which vehicle should pass the intersection, the vehicles approaching from each direction have been compared (the variables involved are the distance to the intersection, braking distance, current speed, and deceleration capacity).

The paper is organized as follows: Section 2 provides the model and describes the intersection scenario and the intersection algorithm. Section 3 explains the simulations. Section 4 presents the conclusions.

2. METHOD

2.1 Acceleration (Deceleration) model

The model to set the acceleration (r_a) or deceleration (r_d) of a vehicle is presented in *Equation 1*:

$$\begin{aligned} r_a &= (1 - \frac{ds}{d})m_a \\ r_d &= (1 - \frac{d}{ds})m_d \end{aligned} \tag{1}$$

The maximum acceleration and deceleration are m_a=2 m/s^2 and m_d=-4 m/s^2, respectively (around the typical maximum values [18]). The distance ds [19] is the one in which a vehicle will stop if it applies a constant deceleration (d_m) at the current speed (v); d is the Euclidian distance (from the vehicles centre) between the leader's position (x_l, y_l) and the follower's position (x_f, y_f) minus the expected distance between the vehicles (d_e); for example, setting d_e=10 m, the centre-to-centre distance between two vehicles will be 10 m when d=0 (with a vehicle length of 4 m, there will be a distance of 6 m between the back of the leading vehicle and the front of the following vehicle). Distances ds and d are expressed by *Equations 2* and *3*, respectively.

$$ds = \frac{v^2}{-2 \cdot d_m} \tag{2}$$

$$d = \sqrt{(x_f - x_l)^2 + (y_f - y_l)^2} - d_e \tag{3}$$

In addition, a rule is introduced: if d<0, then d=0. This rule keeps the follower braking at m_d if the Euclidian distance is less than d_e between the follower and the leader. The need to decelerate occurs when d<ds. From *Equation 1*, with 1-(d/ds), we regulate the amount of m_d that is applied to decelerate a vehicle; for example, if ds=d then 1-(d/ds)=0 and no action is required, if ds doubles the value of d then 1-(d/ds)=0.5, and this last value increases as d/ds decreases. From *Equation 2*, we select d_m=-1 m/s^2 and test the model behaviour with different values of m_d (see Section 3.1). From *Equation 3*, (x_f, y_f) and (x_l, y_l) are the position coordinates of the follower and the leader, respectively. A vehicle accelerates (r_a) or decelerates (r_d) according to *Equation 4*:

$$a = \begin{cases} r_a & if \quad ds < d \\ 0 & if \quad ds = d \\ r_d & if \quad ds > d \end{cases} \tag{4}$$

To calculate the vehicles' speeds, *Equation 5* is employed, where Δt is the time step and a is the applied acceleration or deceleration.

$$v^t = v^{t-\Delta t} + a\Delta t \tag{5}$$

The model in *Equation 1* uses the positions of the leader and the follower; the analogy with the real world is that vehicles are equipped with Global Positioning Systems (GPSs) and no other sensors are required (a similar assumption is made in [20]). In addition, it is assumed that the sensors operate without errors.

2.2 Intersection scenario

The intersection is composed of two perpendicular streets allowing one-way travel without turnings. The vehicles travelling along the south-north road are numbered with sub index i=1...n_1 and those travelling from east to west are labelled as j=1...n_2, as can be observed in *Figure 1* (x-axis positive to the west and y-axis positive to the north). In addition, the origin of the intersection is (x_{in}, y_{in}), the position (in the centre) of the i-th vehicle travelling in the south-north direction is (x_p^i, y_p^i) and its target coordinate is (x_{sn}, y_{sn}). For the east–west direction the corresponding coordinates are (x_p^j, y_p^j) and (x_{ew}, y_{ew}), respectively.

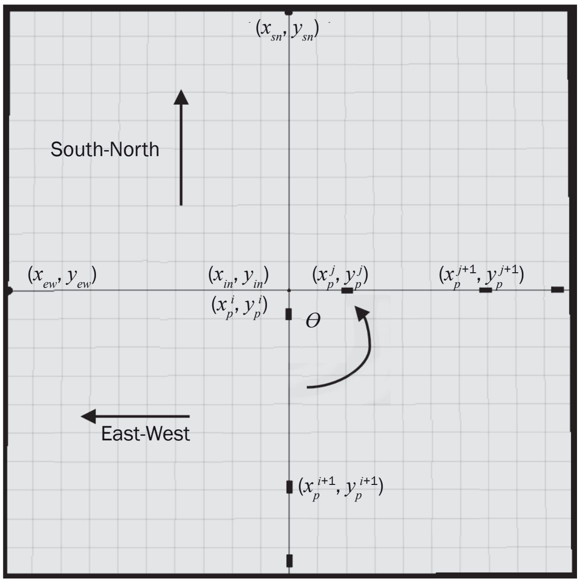

Figure 1 – Intersection scenario

With vehicles coming from both directions, the vehicle for which the condition $dc \leq ds+s$ gets the priority to pass and the vehicle coming from the other direction decelerates if required, that is, if the condition is true and the vehicle travelling from the other direction has not passed the intersection. From the condition, $ds=v^2/4$, and $dc = \sqrt{(x_p - x_{in})^2 + (y_p - y_{in})^2}$, the vehicle position is (x_p, y_p) and the intersection position is (x_{in}, y_{in}). In addition, with θ=90°, it was found that s=6 m is a suitable value.

To determine whether a vehicle has passed the intersection origin (x_{in}, y_{in}), *Equation 6* is used. The circulation from east to west is enabled if $\alpha \geq \beta$ (inequality involving the i-th vehicle position), while the circulation from south to north is enabled if $\delta \leq \gamma$ (inequality involving the j-th vehicle position). From *Equation 6*, c is the distance that a vehicle needs to cover after the intersection origin to enable the counterpart traffic flow; for vehicles with a length of 4 m, c=2.8 m is selected.

$$\begin{aligned} &\alpha = y_p^i \\ &\beta = y_{in} + c \end{aligned} \quad if \quad y_p^i < y_{sn}$$

$$\begin{aligned} &\delta = x_p^j \\ &\gamma = x_{in} - c \end{aligned} \quad if \quad x_p^j > x_{ew} \tag{6}$$

2.3 Intersection algorithm

The intersection algorithm is presented in Algorithm 1. A vehicle accelerates (decelerates) according to *Equation 4* except when it has no priority (the vehicle with priority has not passed the intersection yet) and the condition $dc \leq ds+s$ is true; in that moment, the vehicle without priority decelerates at a constant rate of a=-2 m/s^2 ; then $ds=v^2/2 \cdot a$ is the distance required for the vehicle to brake before the intersection; with s=6 m, after braking there will be 4 m between the front of the vehicle and the intersection centre.

Algorithm 1. Intersection Algorithm

Initial conditions:
flag1=true, flag2=true, exit=false, i=1, j=1, e=2,
n_1=total cars of south-north direction,
n_2=total cars of east-west direction.

```
do{
Comparison 1:
$if(dc_i\text{-}s \leq ds_i \quad \wedge \quad flag1{=}{=}true)$
        $\{flag2{=}false;$
        $if(dc_j\text{-}s \leq ds_j)$
            $\{flag1{=}false;$
            $Decelerate\ vehicle_j\ ;\}\}$
Comparison 2:
$if\ (dc_j\text{-}s \leq ds_j \quad \wedge \quad flag2{=}{=}true)$
        $\{flag1{=}false;$
        $if\ (dc_i\text{-}s \leq ds_i)$
            $\{\ flag2{=}false;$
            $Decelerate\ vehicle_i\ ;\}\}$
Check 1:
$if\ (\alpha_i \geq \beta_i \wedge c_1 \leq n_1)$
```

```
    {$Accelerate\ vehicle_j$;
    $flag1=true$; $flag2=true$;
    $if\ (v_{j-e}\leq 0 \land j>e)$
        {$Decelerate\ vehicle_j$ ;
        $Accelerate\ vehicle_i$ ;
        $flag1=false$; $flag2=false$;}
        $if\ (i<n_1)\{i=i+1;\}$
            else {exit=true}}
Check 2:
$if\ (\delta_j\leq \gamma_j \land c_2\leq n_2)$
    {$Accelerate\ vehicle_i$;
    $flag1=true$; $flag2=true$;
    $if\ (v_{i-e}\leq 0 \land i>e)$
        {$Decelerate\ vehicle_i$ ;
        $Accelerate\ vehicle_j$ ;
        $flag1=false$; $flag2=false$;}
        $if\ (j<n_2)\{j=j+1;\}$
            else {exit=true}}}
  while {exit==false}
```

3. SIMULATIONS

3.1 Model simulations

The simulations were developed with the Unity Engine (used in robotics [21] and optimization problems [22]). The simulations were designed to test the model presented in *Equation 1*.

Next simulations show the results with m_d=-4 m/s^2 and d_m=-1 m/s^2. As the initial conditions, four vehicles spaced at 10 m begin to accelerate; then, when all the vehicles reach the maximum allowed speed (in this case, 10 m/s), the leader decelerates constantly at –2 m/s^2 until it stops. *Figure 2* shows the distance between each vehicle and its respective leader, with all vehicles stopping at a distance of 10 m from each other (from *Equation 3*, d_e=10 m). *Figure 3* presents a graph of time versus speed. *Figure 4* shows that the maximum deceleration applied by the second, third, and fourth cars (after the first car decelerates) is less than –2 m/s^2.

The previous simulation is repeated with m_d=3.5d_m (from [4], the comfortable deceleration rate proposed by the American Association of State Highway and Transportation Officials is 3.4 m/s^2); again, the vehicles stop at the expected distance (10 m) with an error of approximately –0.01 m (the negative sign indicates less distance than expected). The error increases if m_d=3.25d_m (see *Table 1*).

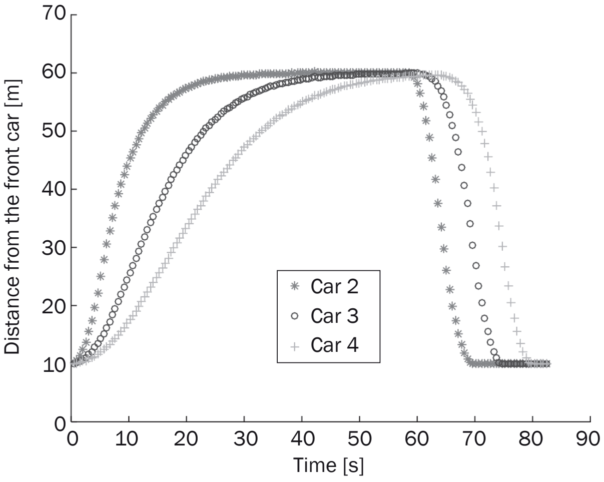

Figure 2 – Time vs. distance

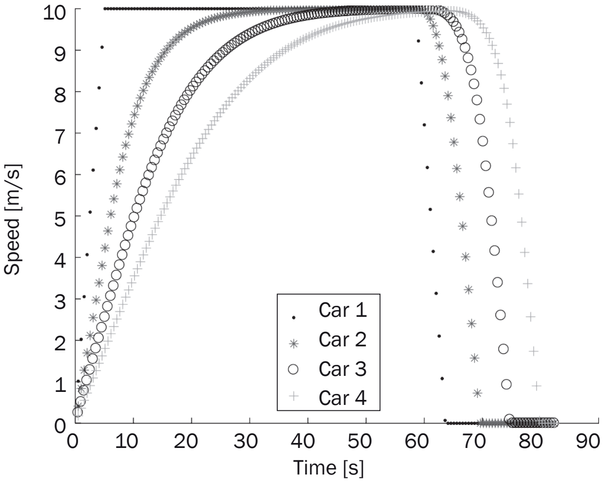

Figure 3 – Time vs. speed

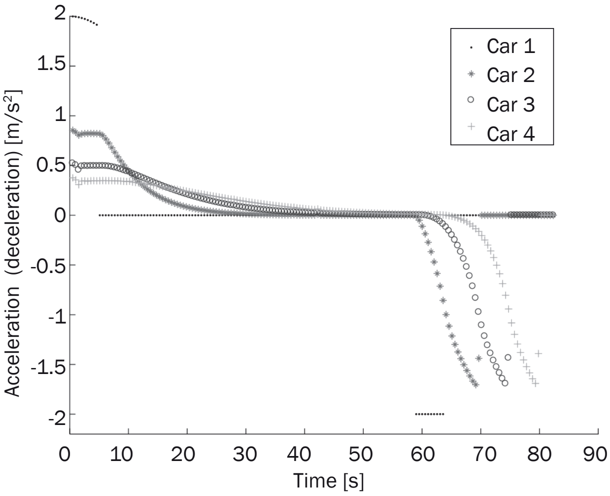

Figure 4 – Time vs. acceleration (deceleration)

Table 1 – Error in distance for each case (in metres)

	Error at m_d=3d_m	Error at m_d=3.25d_m	Error at m_d=3.5d_m	Error at m_d=4d_m
Car 2	-0.3718262	-0.1113281	-0.01239014	-0.00024414
Car 3	-0.3413696	-0.1013794	-0.01019287	0
Car 4	-0.3389893	-0.09716797	-0.01098633	-0.00067139

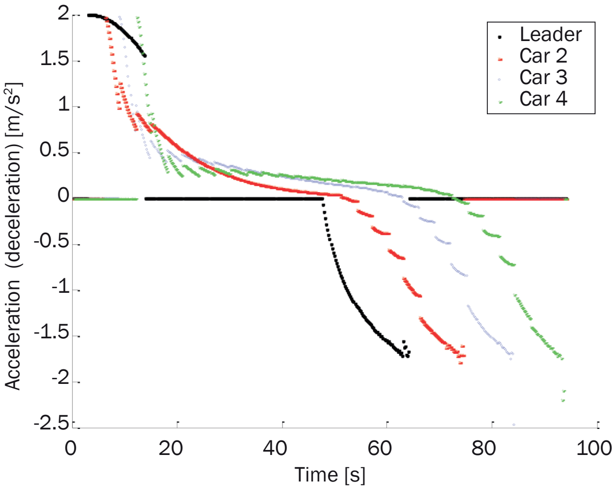

Figure 5 – Time vs. acceleration, 3 s time delay

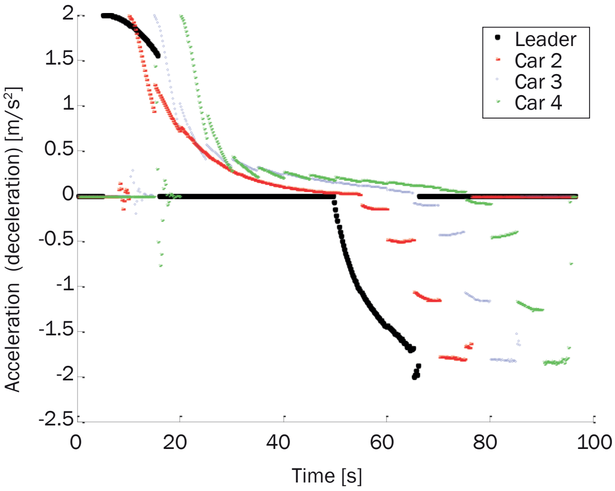

Figure 6 – Time vs. acceleration, 5 s time delay

In *Table 1* we can observe that a smaller error (zero if a car stops 10 m from the car in front) is obtained by approximating $m_d=4d_m$. Additionally, setting $d_e=5$ m, that is, considering a distance of 1 m from the leader's rear to the follower's front in *Equation 3*, the vehicles decelerate avoiding collisions (this simulation was performed with maximum vehicle speed=10 and 15 m/s).

The period for the model to acquire data (the leader position) is called time delay. Next simulations consider time delay to model the vehicles' behaviour, with $ds=\frac{v^2}{-(2\cdot d_m)}+1$ if there is time delay. As initial conditions, vehicles are spaced at 10 m from each other, the maximum speed is set at 20 m/s. *Figure 5* shows a graph of time versus acceleration (deceleration) for a 3-second time delay and *Figure 6* shows the same for a 5-second time delay. Both graphs show acceleration (deceleration) discontinuities. During the time delay, the last registered leader speed is used to calculate the distance between leader and follower (the current position and speed of this last is known), then the model works with this information until data is acquired again (each time delay cycle).

With the intention of eliminating discontinuities, a second-order polynomial (from the least squares regression with time as an independent variable) was used to estimate the vehicles' speed and then the distance between each leader and follower. The simulation was repeated considering a time delay of 2 s. *Figure 7* shows time versus acceleration (deceleration) and *Figure 8* shows time versus observed and estimated leader speed (the first vehicle in the platoon). In addition, this simulation was repeated with a time delay of up to 5 s with no collisions occurring.

In the next simulation, the time delay is 1 s and the initial distance between vehicles is 20 m (measured from the centres). At the beginning, the vehicles constantly accelerate at 2 m/s^2 and reach the maximum speed of 20 m/s; then the leader (the first vehicle in the platoon) decelerates at –4 m/s^2. *Figures 9*, *10*, and *11* show the acceleration, speed, and distance from the

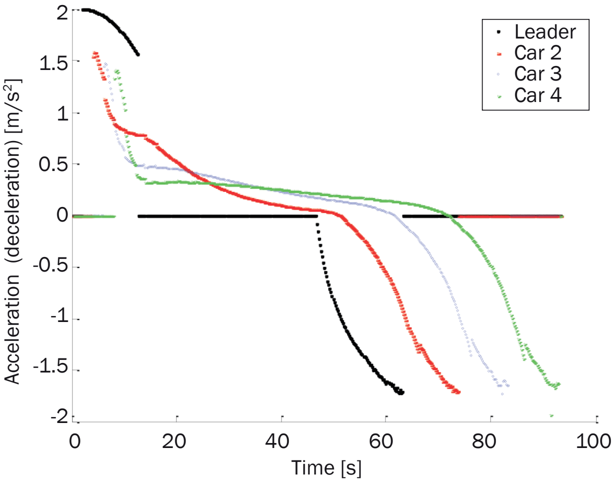

Figure 7 – Time vs. acceleration (deceleration)

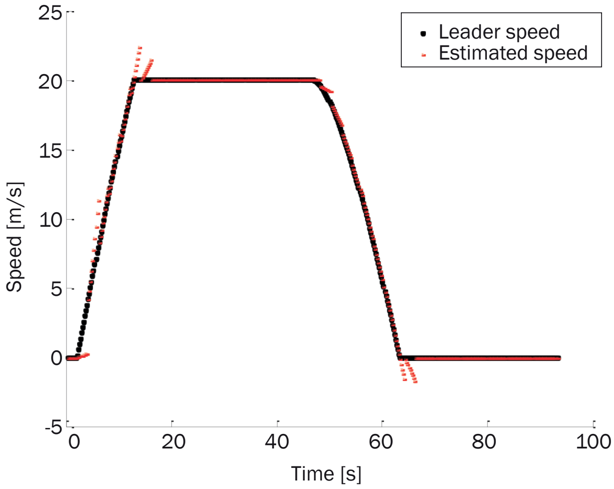

Figure 8 – Estimated and current leader speed

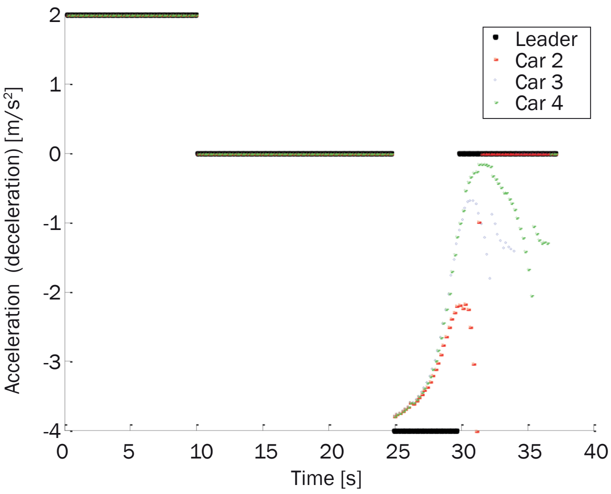

Figure 9 – Time vs. acceleration-deceleration

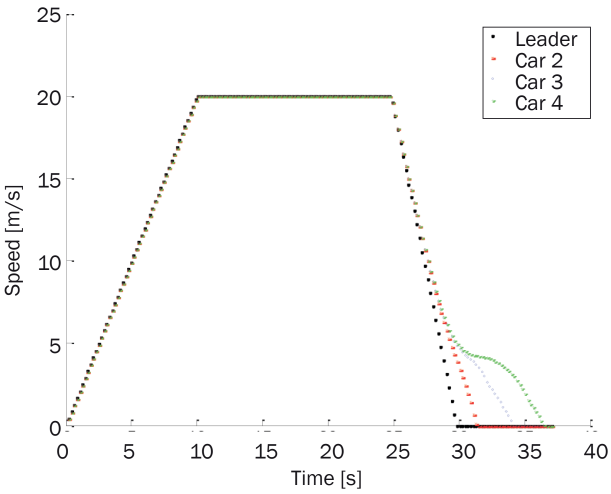

Figure 10 – Time vs. speed

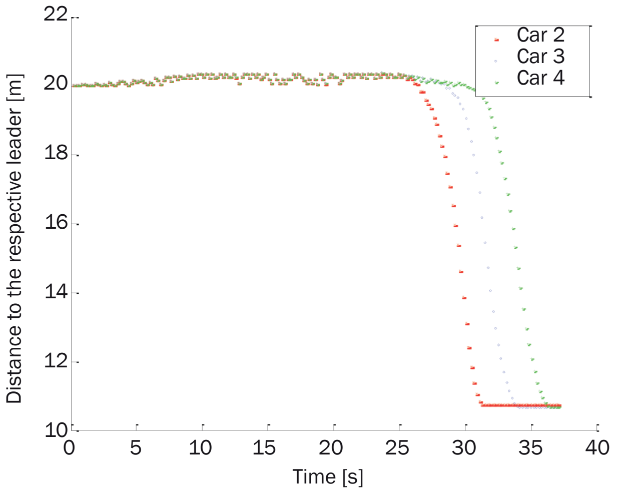

Figure 11 – Time vs. distance from a follower to its leader

follower to the respective leader. The simulation was repeated considering a time delay of 3 s without collisions. In addition, if the distance between the vehicles centres is reduced to 6 m (in the initial conditions) and the expected distance d_e in *Equation 3* is 5 m (or 1 m from the leader's rear to the follower's front), the vehicles effectively decelerate while preserving a distance of at least 1 m from each other at the end (with time delay=1 s).

3.2 Intersection simulations

The next simulation is conducted under the scenario presented in *Figure 1*. Algorithm 1 assigns the vehicles priority to pass the intersection and *Equation 4* regulates the vehicles' motion except when braking at constant speed before the intersection because another vehicle has the priority to pass first. Ten vehicles from each direction were created and the travel behaviour was recorded for the first and last three cars from both directions (before and after passing the intersection). The maximum deceleration is m_d=-4 m/s^2 and the maximum speed is 15 m/s. Under the initial conditions, we have the vehicles at rest and spaced 10 m apart from each other. *Figures 12* and *13* show the acceleration (deceleration) of the first three cars travelling from south to north and east to west, respectively. *Figure 13* shows a deceleration of –2 m/s^2 applied by the first car travelling from east to west due to the priority of the first car travelling from south to north (as the first vehicle travelling from south to north arrives first at the intersection, it receives priority), this induces a decrement of the acceleration slope of the second car (east–west). The first car (east–west) accelerates when it gets priority, which in turn causes an increment of the acceleration slope of the second car (east–west), this last decelerates when it loses the priority at the intersection but, when it is its turn to get

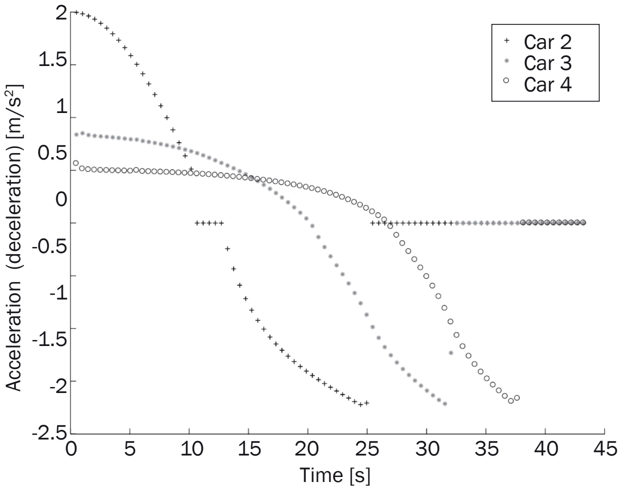

Figure 12 – First 3 cars in south-north direction

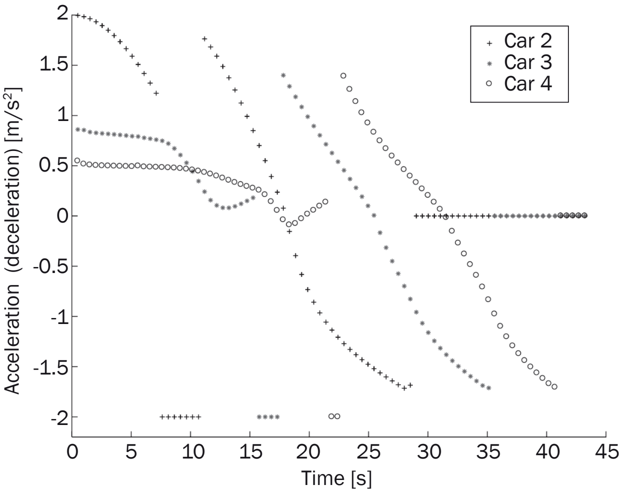

Figure 13 – First 3 cars in east-west direction

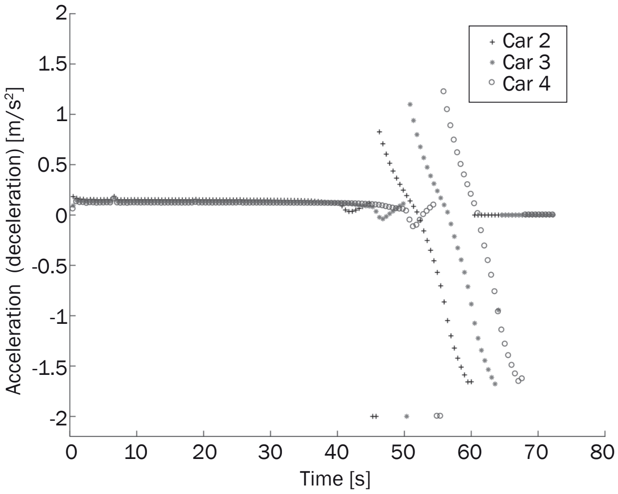

Figure 14 – Last 3 cars in south-north direction

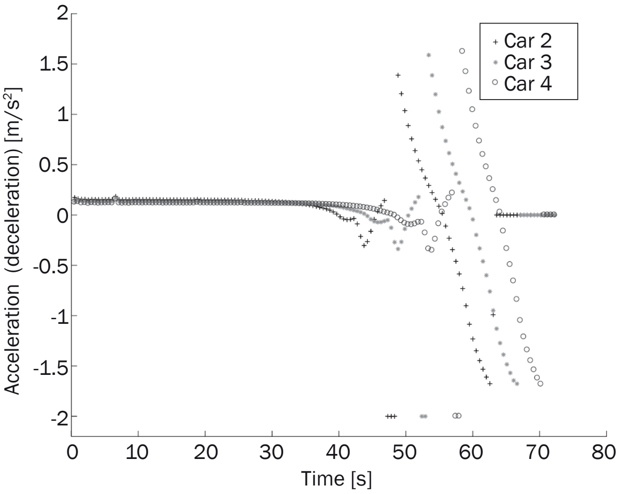

Figure 15 – Last 3 cars in east-west direction

priority, *Equation 4* regulates its acceleration (deceleration) behaviour again. A similar pattern is repeated by the consecutive cars travelling in the east–west direction.

Figures 14 and *15* show the acceleration or deceleration of the last three cars. In both plots it is observed that each vehicle decelerates at -2 m/s^2 and this causes a decrease of the acceleration slope of the corresponding following vehicle: the slope increases when the corresponding leader begins to accelerate because it gets the priority.

4. CONCLUSIONS

The relationship between m_d and d_m has been successfully established; that is, with the selected values the vehicles decelerate while avoiding collisions when the leader abruptly decelerates. In addition, if a time cycle to update the vehicle position is considered, the acceleration (deceleration) suffers discontinuities. Estimating the leader's speed and then calculating the distance between the vehicles solved this problem, where the model supports data time delays of up to 5 s under normal conditions (the model calculates the leader deceleration). If the distance between the vehicles (under the initial conditions) is set at 20 m and the maximum speed is 20 m/s, when the leader abruptly decelerates at -4 m/s^2, the maximum time delay supported is 3 s. The case with an initial distance of 6 m, the same maximum speed, and a final expected distance of 5 m supports a 1 s time delay. In the last case, which focuses on reducing the distance between the vehicles in a platoon, two benefits are detected: (1) the deceleration of the following vehicles (as a reaction if the leader decelerates) is smooth, not abrupt, and (2) the time cycle to refresh the data and maintain the desired inter-vehicle distance is 1 s (this time is a common GPS logging capacity [18]). The procedure in this paper reduces the time for a follower to get the information from its leader to operate in car-following situations, as the communications systems may have delays or errors in transmitting data, it is convenient to extend the time to acquire data.

With the simulations performed to manage traffic flow at the intersection (with Algorithm 1 and the proposed model), it is possible to maintain a balanced flow from both directions. This application of the model has the benefit that the only required sensor is one that delivers (with accuracy) the vehicles position. Also, the communication between the vehicles (to operate in car-following) will have the benefit of supporting delays (depending on the situation, as we conclude through different driving scenarios presented in this paper) when transmitting data.

JOSÉ GERARDO CARRILLO-GONZÁLEZ, Dr.[1,2]
Correo: j.carrillo@correo.ler.uam.mx
JESÚS ARÁMBURO-LIZÁRRAGA, Dr.[3]
Correo: jaramburo@cucea.udg.mx
LILIANA IBETH BARBOSA-SANTILLAN, Dr.[3]
Correo: ibarbosa@cucea.udg.mx
[1] Consejo Nacional de Ciencia y Tecnología (CONACYT)
Ciudad de México, México
[2] Departamento de Sistemas de Información y
Comunicaciones, Universidad Autónoma Metropolitana,
Unidad Lerma, Av. Hidalgo Pte. 46, Col. La Estación,
Lerma de Villada, 52006, Edo. de Mex., México
[3] Universidad de Guadalajara, Av. Juárez # 976, col. centro,
44100, Guadalajara, Jalisco, México

MODELO PARA ACELERAR (DESACELERAR) SOPORTANDO RETRASOS DE TIEMPO PARA ACTUALIZAR DATOS

RESUMEN

Este artículo propone un modelo matemático para regular la aceleración (desaceleración) de vehículos que se conducen de manera autónoma en situaciones en las que un vehículo sigue a otro. Se ha diseñado un ambiente virtual para hacerle pruebas al modelo en diferentes situaciones: (1) los vehículos siguiendo al líder desaceleran a tiempo si el líder frena, donde es posible tener un retraso de hasta 5 s para actualizar los datos que requiere el modelo, es decir la posición de los vehículos, (2) con la intención de optimizar espacio, los vehículos se agrupan en pelotones, donde se puede tener un retraso de 3 s para actualizar los datos si los vehículos están espaciados a 20 m de centro a centro, y 1 s si el espacio es de 6 m, en ambos caso la velocidad máxima es 20 m/s, y (3) se presenta un algoritmo para administrar la prioridad de los vehículos en una intersección de tráfico, en este caso el modelo regula la aceleración y desaceleración aplicada por los vehículos, además con el algoritmo se logra un balance entre el número de vehículos que provienen de cada lado de la intersección.

PALABRAS CLAVE

Simulación; Desaceleración; Modelo Matemático; Algoritmo; Intersección;

REFERENCES

[1] Gattami A, Al Alam A, Johansson KH, et al. Establishing Safety for Heavy Duty Vehicle Platooning: A Game Theoretical Approach. *IFAC Proceedings Volumes*. 2011;44(1): 3818-3823.

[2] Varaiya P. Smart cars on smart roads: problems of control. *IEEE Transactions on automatic control.* 1993;38(2): 195-207.

[3] Fernandes P, Nunes U. Platooning of autonomous vehicles with intervehicle communications in SUMO traffic simulator. In: *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on;* 2010. p. 1313-1318.

[4] Maurya AK, Bokare PS. Study of deceleration behaviour of different vehicle types. *International Journal for Traffic & Transport Engineering.* 2012;2(3): 253-270.

[5] Wu Z, Liu Y, Pan G. A smart car control model for brake comfort based on car following. *IEEE Transactions on Intelligent Transportation Systems.* 2009;10(1): 42-46.

[6] Li G, Wang W, Li SE, et al. Effectiveness of flashing brake and hazard systems in avoiding rear-end crashes. *Advances in Mechanical Engineering*. Volume 2014.

[7] Ahmed KI. *Modeling drivers' acceleration and lane changing behavior*. Doctoral dissertation. Massachusetts Institute of Technology; 1999.

[8] Sun J, Ma Z, Li T, et al. Development and application of an integrated traffic simulation and multi-driving simulators. *Simulation Modelling Practice and Theory*. 2015;59: 1-17.

[9] Wang H, Kearney JK, Cremer J, et al. Steering behaviors for autonomous vehicles in virtual environments. *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*; 2005. p. 155-162.

[10] Carvalhosa S, Aguiar AP, Pascoal A. *Cooperative motion control of multiple autonomous robotic vehicles. Master's thesis*. Instituto Superior Técnico; 2009.

[11] Isler V, Sun D, Sastry S. Roadmap Based Pursuit-Evasion and Collision Avoidance. *Robotics: Science and Systems*. 2005;1: 257-264.

[12] Dresner K, Stone P. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*. 2008;31: 591-656.

[13] Makarem L, Gillet D. Fluent coordination of autonomous vehicles at intersections. *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on;* 2012. p. 2557-2562.

[14] Yan F, Dridi M, El Moudni A. An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach. *International Journal of Applied Mathematics and Computer Science.* 2013;23(1): 183-200.

[15] Hausknecht M, Au T-C, Stone P. Autonomous intersection management: Multi-intersection optimization. *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*; 2011. p. 4581-4586.

[16] Doniec A, Mandiau R, Piechowiak S, et al. A behavioral multi-agent model for road traffic simulation. *Engineering Applications of Artificial Intelligence*. 2008;21(8): 1443-1454.

[17] Mandiau R, Champion A, Auberlet J-M, et al. Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation. *Applied Intelligence*. 2008;28(2): 121-138.

[18] Bokare PS, Maurya AK. Acceleration-deceleration behaviour of various vehicle types. *Transportation Research Procedia*. 2017;25: 4737-4753.

[19] Boer ER, Kuge N, Yamamura T. Affording realistic stopping behavior: A cardinal challenge for driving simulators. *Proceedings of 1st Human-Centered Transportation Simulation Conference*; 2001.

[20] González JG, Arámburo-Lizárraga J. Digitalized roads based on GPS data in a virtual world. *Procedia Technology*. 2013;7: 20-29.

[21] Mattingly WA, Chang D, Paris R, et al. Robot design using Unity for computer games and robotic simulations. *Computer Games (CGAMES), 2012 17th International Conference on*; 2012. p. 56-59.

[22] Jie J, Yang K, Haihui S. Research on the 3D game scene optimization of mobile phone based on the Unity 3D engine. *Computational and Information Sciences (ICCIS), 2011 International Conference on*; 2011. p. 875-877.