Guidance and Control of a Platoon of Vehicles Adapted to Changing Environment Conditions*

Xavier Huppé, Jean de Lafontaine, Mathieu Beauregard, François Michaud
Department of Electrical Engineering and Computer Engineering
Université de Sherbrooke
Sherbrooke, Québec, Canada
xh2@hermes.usherb.ca

Abstract – *This paper describes the decentralized longitudinal control of a ten-vehicle platoon. An adapted time-headway distance approach is adopted, i.e., that each vehicle has to respect a "time-to-contact" with its preceding vehicle. The adaptation is made relatively to the vehicle dynamics and the road conditions. A two-level longitudinal controller is developed using the Linear Quadratic Regulor (LQR) and feedback linearization methods. String stability analysis is discussed resulting in a definition for the adopted control strategy. A pentic polynomial guidance algorithm is used to limit the acceleration when a vehicle is reaching a platoon. Simulations showing the leader velocity variation for normal, icy and snowy roads and the pentic guidance approach are presented.*

Keywords: Vehicle longitudinal control, decentralized control, pentic guidance, adapted time-headway, string stability.

1 Introduction

Major cities in the world have to cope with an increasing traffic flow. At the same time, many of their roadway infrastructures are unable to handle this demand. The result is seen in daily traffic jams causing accidents, driving stress, loss in efficiency and increase in fuel consumption and pollution.

Automated Highway Systems (AHS) [5] are potential solutions to resolve these problems. One type of such systems are the *Collaborative Driving Systems (CDS)* that ensure safe and efficient coordination of vehicles. For instance, it can be a platoon in which the vehicles are automatically controled to follow each other with a safe distance between them.

This paper describes the development and simulation of a part of the CDS. It focusses on the longitudinal guidance and control of a ten-vehicle platoon. First, the CDS concept is briefly explained in Section 2. Section 3 gives an idea of the developed vehicle and platoon simulators and is helpfull in understanding the vehicle control part described in Section 4. A string stability analysis is conducted in Section 5, followed by a description of platoon guidance approach in Section 6. Section 7 presents simulation results.

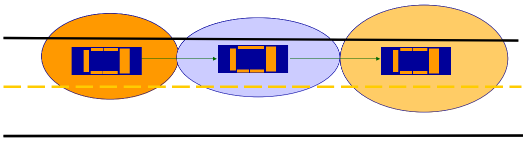

Figure 1: Virtual "safety bubble"

2 Collaborative Driving System

The global objective of such systems is to insure a safe and coordinated movement of a group of vehicles. Our approach is based on the concept of a virtual "safety bubble" surrounding each vehicle, as shown in Figure 1. This space is determined according to each vehicle dynamics and road conditions. The main idea is that no collision should occur in case of emergency braking.

In this work, we focus on longitudinal guidance and control of the platoon of vehicles, i.e., in the moving direction of the platoon when following a straight line. The adopted distance policy, derived from the virtual "safety bubble" concept, is an adapted time-headway. Each vehicle has to follow a safe time-headway relative to the front vehicle resulting in different distances between the vehicles in the platoon. A time-headway h_d is defined as the time it takes to cover the distance between the rear bumper of the preceding car and the front bumper of the following vehicle.

2.1 Decentralized guidance and control system architecture

We propose a decentralized approach to guide and control vehicles in the platoon. This gives the opportunity to create independant local controllers resulting in distributed computing over the platoon. There are mainly two avantages of using a decentralized control approach. First, safety is improved since the decisions

*0-7803-7952-7/03/$17.00 © 2003 IEEE

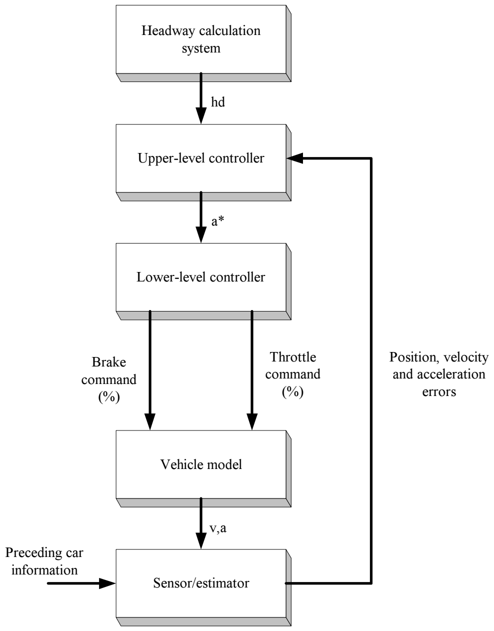

Figure 2: Single vehicle system architecture

of each vehicle do not rely only on one centralized computer. Second, because the time-headway distance between vehicle approach is used, it has been shown that no control data has to be transmitted from the platoon leader to the following vehicles [10] to ensure stable platoon control. Only measured data and local variables are necessary, avoiding communication failures and associated costs [9].

As shown in Figure 2, there are five modules in our system. The first one is the *Headway Calculation System*. This module gives the time-headway that each vehicle has to follow based on environment conditions changes, vehicles dynamics and safety considerations. The upper-level and lower-level controller modules are presented later in Section 4. The vehicle model is presented in Section 3 as a part of the platoon simulator developed under MATLAB/SIMULINK. The *Sensor/estimator* module calculates the relative distance, velocity and acceleration between the preceding and the controlled vehicles. At the moment, this part is simulated by a perfect measurement of those three variables.

3 Vehicle and platoon model

Figure 3 represents the components of the vehicle model. The vehicle engine is simulated by an engine map that provides a torque relative to the engine revolution and its throttle. This information is made available by some car manufacturers. The transmission, torque converter and differential bloc contains all the parts that simulate the power transfer from the engine torque to the wheel torque. The model uses automatic transmission and the torque converter model simulates the non-linearities that caracterise this part of the vehicle [3]. The wheel model takes into account the road surface and has been developed using the theory described in [6]. The braking system is an anti-lock braking system (ABS) described in [2]. The frame forces are the rolling resistance, the aerodynamic drag and the road grade force. All the resulting forces are summed and integrated to give the vehicle acceleration and velocity.

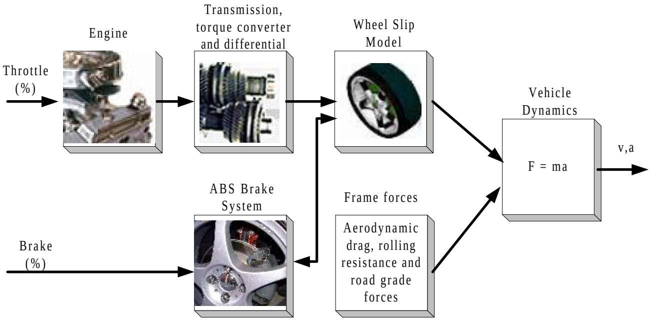
Engine
Transmission, torque converter and differential
Wheel Slip Model
Throttle (%)
Vehicle Dynamics
F = ma
v,a
ABS Brake System
Frame forces
Aerodynamic drag, rolling resistance and road grade forces
Brake (%)

Figure 3: Complete vehicle model schematic

Using the vehicle model, a ten-vehicle platoon simulator has been developed. Each vehicle is independant and can have different caracteristics (engine, transmission, weight, . . .). The platoon simulator also provides the possibility to create different road surfaces (normal, wet, snowy and icy), wind schedules and road grades that change with vehicle absolute position.

4 Vehicle control

The vehicle control approach has two levels. The upper level calculates the desired control effort for the vehicle. The control effort, u_i, can be seen as the vehicle desired acceleration in (m/s^2). The lower level takes the control effort and computes the corresponding throttle and brake commands.

4.1 Upper level controller

The upper level controller calculates the desired control effort as shown in Equations (1) to (4). In these equations, x_i, $\dot{x}_i$ and $\ddot{x}_i$ are respectively the absolute position, the velocity and the acceleration of the controlled vehicle. k_{x_i}, k_{v_i} and k_{a_i} are control gains. Note that the subscript i refers to the position of the vehicle in the platoon.

$$u_i = k_{x_i} e_i + k_{v_i} \dot{e}_i + k_{a_i} \ddot{e}_i \tag{1}$$

$$e_i = x_{i-1} - x_i - h_{d_i} v_i \tag{2}$$

$$\dot{e}_i = \dot{x}_{i-1} - \dot{x}_i \tag{3}$$

$$\ddot{e}_i = \ddot{x}_{i-1} - \ddot{x}_i \tag{4}$$

To calculate these gains, a simple third-order vehicle model has been used. It is described by the Equation (5). This model approximates the servo-loop dynamics by a first-order lag (τ_i).

$$\dddot{x}_i = \frac{u_i - \ddot{x}_i}{\tau_i} \tag{5}$$

A state-space model has been created for the distance control, as in [8]. It is described in Equations (6) to (11). The states variables are respectively x_i, $\dot{x}_i$ and $\ddot{x}_i$.

$$X_i = \begin{bmatrix} x_i & \dot{x}_i & \ddot{x}_i \end{bmatrix}^T \tag{6}$$

$$U_i = u_i \tag{7}$$

$$\dot{X}_i = A_i X_i + B_i U_i \tag{8}$$

$$A_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{-1}{\tau_i} \end{bmatrix} \tag{9}$$

$$B_i = \begin{bmatrix} 0 & 0 & \frac{1}{\tau_i} \end{bmatrix}^T \tag{10}$$

$$C_i = \begin{bmatrix} -1 & -h_{d_i} & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{11}$$

The C matrix supposed that the vehicle in front is moving at constant speed. By doing this simplification, it is possible to find control gains without knowing the front vehicle dynamics.

To find the gains, the *Linear Quadratic Regulator (LQR)* method has been choosen. This method has the advantage of finding optimal gains by fixing a trade-off between the control effort and the state measurements.

To use the *LQR*, two extra weight matrices have to be defined. They are defined in Equations (12) and (13) where $q1$, $q2$, $q3$ and r_i are user-selected constants.

$$Q_i = \begin{bmatrix} q1 & 0 & 0 \\ 0 & q2((hd_i)^{-1})^2 & 0 \\ 0 & 0 & q3(\tau_i) \end{bmatrix} \tag{12}$$

$$R_i = \begin{bmatrix} r_i \end{bmatrix} \tag{13}$$

The definitions of the matrix Q and R give a relative weighting between the error and the command. If the matrix R is fixed and the weights of Q increased, the steady-state errors are reduced and a higher control effort is given. If we set r_i, $q1$, $q2$ and $q3$, the Q matrix changes with two variables, h_{d_i} and τ_i. Following this logic, if h_{d_i} increases, the control effort decreases. A vehicle with a higher time-headway has a higher gap relative to the preceding car. For the same relative error, the pourcentage of error is lower if h_{d_i} increases. Then, the control effort can be lower. If τ_i increases, the control effort is going up. Considering that there is no saturation, a system with a smaller actuator lag respond more quickly than one with a high delay. Then, the command can be lower for a faster system.

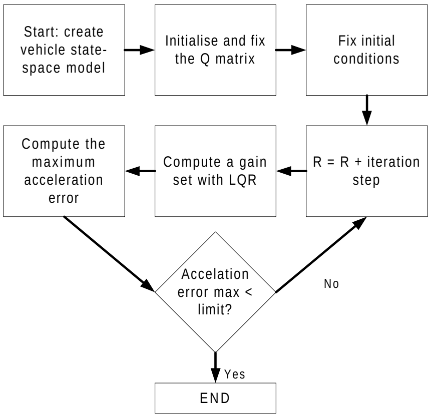
Start: create vehicle state-space model
Initialise and fix the Q matrix
Fix initial conditions
R = R + iteration step
Compute a gain set with LQR
Compute the maximum acceleration error
Accelation error max < limit?
No
Yes
END

Figure 4: Acceleration limitation algorithm

4.1.1 Acceleration limitation algorithm

One way to validate the control structure described above is to test it under initial conditions, like an error in relative position. By looking at the states variables, it is possible to evaluate the vehicle error in acceleration that this initial error in position creates. If the preceding vehicle is in steady-state, the error in acceleration becomes the vehicle acceleration. Else, by changing gains, it is possible to limit the vehicle acceleration depending on those fixed initial conditions. The block diagram shown in Figure 4 explains this acceleration limitation algorithm.

4.2 Lower level controller

For more complex vehicle models, the desired acceleration (u_i) has to be transformed in corresponding throttle and brake commands [11]. Figures 5 and 6 show how the lower-level controller works.

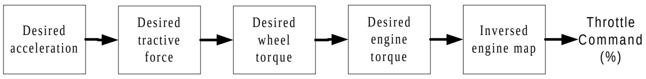

Figure 5: Throttle control algorithm

The throttle control algorithm neglects the torque converter effect and considers it to be locked. Also, it neglects the wheel traction on the road considering that the torque is totally transfered to the vehicle dynamics module. The inversed engine map is calculated according to the engine map. It is then possible to obtain a throttle relative to a desired engine torque and the actual vehicle engine revolution.

The brake control algorithm takes the desired acceleration and substracts it to the actual vehicle acceleration and provides a brake torque. The switching between the throttle and brake commands is done under two conditions: 1) The throttle command must be equal to 0 ; 2) $u_i - a_i \leq 0$. Then, the negative engine torque occurring when the throttle is released can be used to slow

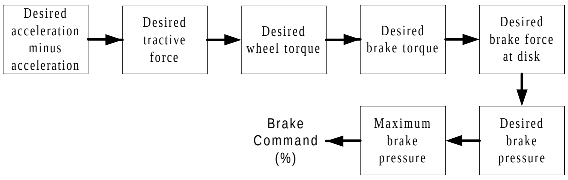

Figure 6: Brake control algorithm

down the vehicle wihout automatically switching to the brake system when u_i is negative. The brake system only works when the engine torque is unable to provide the desired deceleration.

5 String stability analysis

The string stability of a platoon has been addressed in many works and is the major point to consider when developing a platoon controller [4]. Usually, the string stability definition combines two aspects. First, all the vehicles in the platoon must be stable when reacting to the preceding vehicle behavior. Second, according to the classical string stability definition, the inter-vehicle spacing error must be attenuated starting from the leader to the last car in the platoon.

This definition is adequate for a platoon where all the vehicle gaps (h_{d_i}) are equal. In [1], other definitions of string stability have been proposed for the case where each vehicle has a different time-headway to respect. Considering the input-output behavior of the i^{th} vehicle, named $G_i(s)$, the string stability for a platoon of k vehicles is defined as in Equation (14).

$$\parallel G_1 \cdot G_2 \ldots G_k \parallel \leq 1 \tag{14}$$

Instead of evaluating the gap error between the vehicle, this evaluates each vehicle oscillatory behavior in absolute position. Then, it can be considered that a platoon is string stable if the following vehicle has a smaller sinusoïdal variation in position than the preceding vehicle.

Equation (15) defines the position transfer function for our control approach. The calculation is based on the model developped in Equations (1) to (5).

$$G_i(s) = \frac{k_{a_i}s^2 + k_{v_i}s + k_{x_i}}{\tau_i s^3 + (1 + k_{a_i})s^2 + (k_{v_i} + k_{x_i}h_{d_i})s + k_{x_i}} \tag{15}$$

If $|G_i(jw)| \leq 1$ for each vehicle, then the platoon string stability is assured. Also, looking at Equation (15), if h_{d_i} increases lightly and the gains stay the same, the condition for $G_i(jw)$ still holds. Then, if for safety reasons the *Headway Calculation System* increases the time-headway of a vehicle, the string-stability is preserved.

6 Pentic guided platoon approach

When a vehicle is too far from the preceding vehicle, the local controller generates high control effort to reduce the errors. Unfortunatly, this often causes control saturation that can lead to unexpected results.

One way to resolve this problem is to design controllers that limit the control effort in a way to avoid saturation. To handle such manoeuver, we use a guidance algorithm based on *pentic polynomial in time*. This is similar to the Apollo lunar descent guidance algorithm [7]. The goal is to create three trajectories to feed a controller so that the vehicle reaches its operating point within a certain acceleration limit. Equations (16) and (17) describe how to calculate the trajectory points in-time. Note that the $*$ means desired.

$$\begin{bmatrix}\Delta x_i * & \Delta \dot{x}_i * & \Delta \ddot{x}_i *\end{bmatrix}^T =$$
$$\begin{bmatrix} 1 & T_{go} & \frac{T_{go}^2}{2} & \frac{T_{go}^3}{6} & \frac{T_{go}^4}{24} & \frac{T_{go}^5}{120} \\ 0 & -1 & -T_{go} & -\frac{T_{go}^2}{2} & -\frac{T_{go}^3}{6} & -\frac{T_{go}^4}{24} \\ 0 & 0 & 1 & T_{go} & \frac{T_{go}^2}{2} & \frac{T_{go}^3}{6} \end{bmatrix} \begin{bmatrix} \Delta x_{i_{tgt}} \\ \Delta \dot{x}_{i_{tgt}} \\ \Delta \ddot{x}_{i_{tgt}} \\ \Delta x_{i_{tgt}}^{(3)} \\ \Delta x_{i_{tgt}}^{(4)} \\ \Delta x_{i_{tgt}}^{(5)} \end{bmatrix} \tag{16}$$

$$T_{go} = T_{tgt} - t \tag{17}$$

In these equations, the subscript *tgt* refers the target point of the trajectories. The terms Δx_i, $\Delta \dot{x}_i$, $\Delta \ddot{x}_i$, $\Delta x_i^{(3)}$, $\Delta x_i^{(4)}$ and $\Delta x_i^{(5)}$ are respectively the relative position, velocity, acceleration, jerk, snap and snap first time derivative relative to the target point. The exponent in parenthesis represent the n^{th} derivative of Δx_i where n is an integer. T_{go} means the time-to-go in the trajectory, T_{tgt} the time-at-target and t the elapsed time since the trajectory following started.

The goal is to compute the target point by fixing all the other variables. In the proposed designed, the desired relative position, relative velocity and relative acceleration are equal to zero. The relative jerk, relative snap and relative snap first time derivative at target are still unknowed. Then, it is necessary to compute these three missing target variables. The system in Equation (18) described the equations used for the computation of the missing variables.

$$\begin{bmatrix}\Delta x_{i_{ini}} & \Delta \dot{x}_{i_{ini}} & \Delta \ddot{x}_{i_{ini}}\end{bmatrix}^T =$$
$$\begin{bmatrix} \frac{T_{tgt}^3}{6} & \frac{T_{tgt}^4}{24} & \frac{T_{tgt}^5}{120} \\ -\frac{T_{tgt}^2}{2} & -\frac{T_{tgt}^3}{6} & -\frac{T_{tgt}^4}{24} \\ T_{tgt} & \frac{T_{tgt}^2}{2} & \frac{T_{tgt}^3}{6} \end{bmatrix} \begin{bmatrix} \Delta x_{i_{tgt}}^{(3)} \\ \Delta x_{i_{tgt}}^{(4)} \\ \Delta x_{i_{tgt}}^{(5)} \end{bmatrix} \tag{18}$$

When the target points are computed, it is possible to evaluate the maximum acceleration in the profile and to limit it by decrementing T_{tgt}.

With a perfect vehicle model and no perturbation, it is possible to follow the trajectories only by feeding the vehicle model with the acceleration profile. Realistically, the preceding vehicle can have some variations in relative position, velocity and acceleration and the controlled vehicle has to deal with some external perturbations and modeling uncertainties. This is why we need to use a controller to track the trajectories as described in Equation (19). The control gains are the same as in the one computed in Section 4. This has the advantage to compute only one gain set for the approach and the local controller.

$$u_{i_{approach}} = k_{x_i}(\Delta x_i - \Delta x_i*) + k_{v_i}(\Delta \dot{x}_i - \Delta \dot{x}_i*)$$

$$+k_{a_i}(\Delta \ddot{x}_i - \Delta \ddot{x}_i*) + \Delta \ddot{x}_i* \quad (19)$$

When the approach is completed, the desired relative position, velocity and acceleration are zero. The switching to the local vehicle controller is smooth because the effort is similar for both controller because they are using the same control gains.

7 Simulation results

In the simulations, the platoon leader (vehicle #1) is set to follow a front virtual point having a position, a velocity and an acceleration.

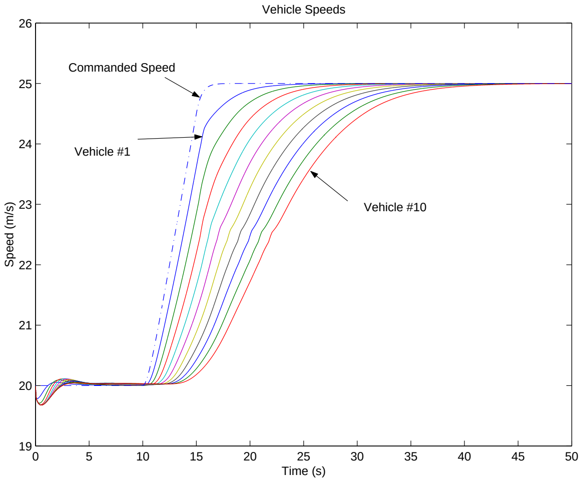

Figure 7: Velocity profiles on a normal road

In the first simulation, the platoon leader change its velocity from 20 m/s to 25 m/s in five seconds. The vehicles have all a one-second time-headway and the platoon is travellling on a normal road. Figure 7 shows the velocity profile for every vehicle. All the vehicles track the velocity change without overshoot and there is a velocity smoothing effect starting from vehicle #1 to vehicle #10. The figure showing the gaps between vehicle was ommitted because it is really similar to the velocity profile.

The second simulation scenario is similar to the first one except that the road conditions suddenly change from normal to a snowy and icy road. Compared to the normal road conditions, there is a gap overshoot among the first vehicles (see Figure 8). This is due to the fact that the first vehicle does not have enough traction on the road to follow the prescribed velocity profile. But, the platoon is still stable since all the gap variations are reduced going from the leader to the last vehicle in the platoon.

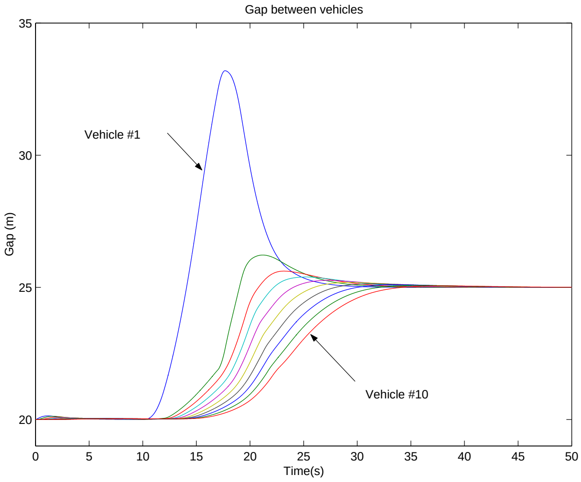

Figure 8: Gap between vehicle on an icy and snowy road

The third scenario shows the transient regime when the vehicle received a different time-headway notice. Ten seconds after the simulation began, all the time-headways changed slowly to become twice their initial value 20 seconds after. This simulate a security notice from the Headway Calculation System. This test have been performed on the snowy and icy road of the preceding simulation is shown in Figure 9. Note that each vehicle has a different time-headway.

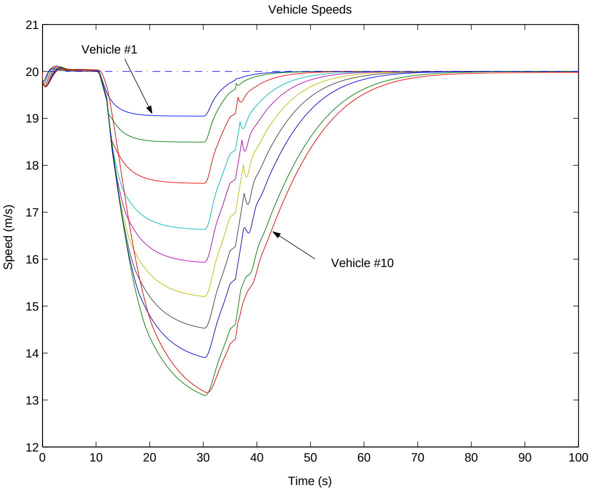

Figure 9: Velocity profiles for a time-headway change

Figure 10 shows the pentic guidance approach acceleration profile on one vehicle when it is entering a platoon. The vehicle has an initial error in position of 100 meters from the desired operating point. The maximum vehicle acceleration is limited to 1 m/s^2 and there is a smooth switching between the two controllers near 25 seconds after the simulation began.

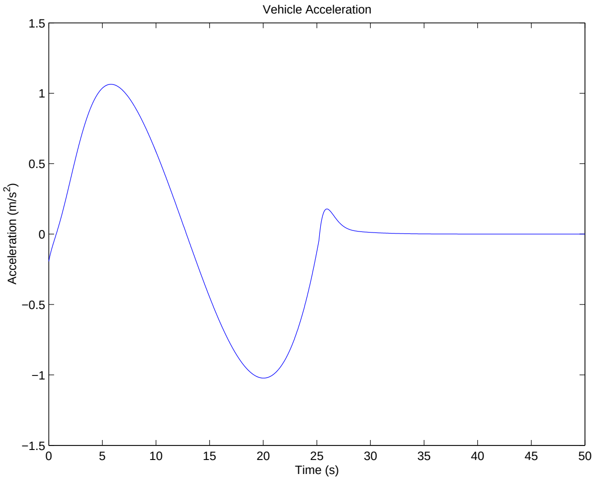

Figure 10: Vehicle acceleration during approach

8 Conclusion

This paper presents the development of a ten-vehicle longitudinal platoon controller. Using a time-headway distance policy, the adopted strategy decentralizes the control to each vehicle and uses local measurements for the two-level controller. The control gains are computed using the LQR method and an acceleration limitation algoritm. Results show that if the local controllers are stable and their transfer functions have a module equal or less than equal to one for all frequencies then the string stablity is ensured. Also, a pentic guided platoon algorithm allows smooth approach by a vehicle that joins the platoon by limiting its acceleration.

In future work, we plan to develop a module to estimate the relatice velocity and relative acceleration ($\dot{e}_i$ and $\ddot{e}_i$) based on the range measurements and the vehicle velocity.

Acknowledgment

This research is supported financially by Auto21st Network of Centre of Excellence of Canada. F.Michaud holds the Canada Research Chair (CRC) in Mobile Robotics and Autonomous Intelligent Systems.

References

[1] A. Bose and P. Ioannou. Analysis of traffic flow with mixed manual and semi-automated vehicles. Technical Report UCB-ITS-PRR-99-14, University of Southern California, 1999.

[2] T. D.Day and S. G. Roberts. A simulation model for vehicle braking systems fitted with ABS. Technical Report 2002-01-0559, SAE Technical Paper Series, 2002.

[3] M. C. Drew. A discussion of vehicle modeling for control. Technical report, University of California Berkeley, 2002.

[4] J. Eyre, D. Yanakiev, and I. Kanellakopoulos. String stability properties of AHS longitudinal vehicle controllers. In *Preprints of the 8th IFAC/IFIP/IFORS Symposium on Transportation Systems, Chania, Greece*, pages 69–74, 1997.

[5] D. N. Godbole and J. Lygeros. Safety and throughput analysis of automated highway systems. Technical Report UCB-ITS-PRR-2000-1, California PATH, January 2000.

[6] U. Kiencke and L. Nielsen. *Automotive Control Systems For Engine, Driveline and Vehicle.* Springer, 2000.

[7] A. R. Klumpp. Apollo lunar descent guidance. *Automatica*, Vol. 10:113–146, 1974.

[8] C.-Y. Liang and H. Peng. Optimal adaptive cruise control with guaranteed string stability. *Vehicle System Dynamics*, Vol. 31:313–330, 1999.

[9] X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick. Effects of communication delay on string stability in vehicle platoons. In *2001 IEEE Intelligent Transportation Systems Conference Proceedings - Oakland (CA), USA*, pages 625–630, 2001.

[10] D. Swaroop, J.K. Hedrick, C.C. Chien, and P. Ioannou. A comparision of spacing and headway control laws for automatically controlled vehicles. *Vehicle System Dynamics*, Vol.23:597–625, 1994.

[11] K. Yi, I. Moon, and Y. D. Kwon. A vehicle-to-vehicle distance control algorithm for stop-and-go cruise control. In *2001 IEEE Intelligent Transportation Systems Conference Proceedings - Oakland (CA), USA*, pages 478–482, 2001.