Vehicle Platoon Control in High-Latency Wireless Communications Environment

Model Predictive Control Method

Hao Zhou, Romesh Saigal, Francois Dion, and Li Yang

Recent developments in vehicle onboard computers and vehicle-to-vehicle communications technology allow automatic control of vehicles and the organization of vehicles into platoons with short intraplatoon distances. One of the major issues with platoon control is latency in wireless communications. Latency has a negative impact on safety and disrupts the stability of platoons. A decentralized longitudinal platoon-controlling mechanism that uses a model predictive control approach to control vehicles safely, even in harsh communications environments, is proposed. The sensitivity of this method was analyzed to derive the conditions for this method to work safely. A simulation test bed for this control method was implemented to test effectiveness and safety under two communications latency settings. The results showed that the model predictive control method could safely control the platoon even in harsh communications environments.

Because of the ever-increasing transportation demand throughout the world, traffic congestion and safety have become more and more important. One way to reduce the impact of congestion and improve safety is to use intelligent transportation systems (*1, 2*). The idea is to increase the capacity of highways by automatically coordinating and controlling vehicles to form platoons in which vehicles are kept at a small distance from each other (*3*). To facilitate the exchange of control information, vehicles are equipped with wireless communications devices, also known as dedicated short-range communication (DSRC) devices. Protocols such as IEEE 802.11p have been developed to enable vehicle-to-vehicle or vehicle-to-infrastructure communication.

The benefit of using intelligent transportation systems includes increased highway capacity, improved safety, and increased fuel efficiency. It has been shown that by using accurate sensors and appropriate vehicle control algorithms, highway capacity can be significantly improved (*4*). Meanwhile, highway safety can also be improved by broadcasting emergency messages to the entire platoon so that vehicles can brake in advance to avoid collisions (*5*). Intelligent transportation systems also have the potential to reduce fuel consumption because vehicle driving is better coordinated through wireless communications, and thus the amount of unnecessary acceleration or deceleration for each vehicle is reduced (*6*).

Early attempts to implement the idea of automatic vehicle control by the Partners for Advanced Transportation Technology (PATH) program at the University of California, Berkeley, used the concept of vehicle-follower control (trying to maintain a certain spacing with other vehicles) rather than point-follower control (trying to follow markers along the road) to operate vehicles in close-formation platoons (*7*). A hierarchical control scheme was introduced to accommodate the nonlinear dynamics of vehicle mechanical systems (engine, transmission, and drive train). The study also conducted a thorough review of communications methods and channel capacity requirements.

A safe control system requires sophisticated methods to handle the latency or delays brought by both the vehicle's mechanical system and the communications systems. Besides mechanical latency, a challenge in developing a safe intelligent vehicle control system is to adequately handle information delay. The relatively narrow radio spectrum and competing nature of wireless communications limit the data rates on the wireless channel (*8*). Moreover, the channel may be noisy and unreliable because of the reflections and attenuation of the wireless signal being transmitted. These effects inevitably introduce some random delay and packet losses (*9*). Experiments have for instance shown that latency is much higher on urban highways than in an open field (*10*) as a result of the signal distortions caused by buildings and highways.

A high number of vehicles using DSRC devices to exchange information may also eventually cause channel congestion and thus higher packet loss ratios. According to the study by Huang et al., latency also depends on the protocols used to implement the wireless data transmissions (*11*). The study notably pointed out that a trade-off exists between message transmission rate and packet-loss ratio: if one tries to increase the transmission rate, channel congestion will more likely happen, thus increasing the packet-loss ratio, and vice versa. Therefore, a sophisticated wireless channel control is needed to maintain a desirable latency level.

Communications delay may have two negative impacts on the automatic vehicle control system: increased risk of collision and violation of string stability. It has been shown by Liu and Dion that information delay of more than 0.5 s would increase the probability of collision significantly (*12*).

Another issue in controlling a platoon of vehicles is string stability. String stability of a platoon refers to a property that guarantees that the spacing error does not amplify as it propagates along a string of vehicles (*13*). Control methods were proposed to handle constant information delays by using leading and preceding vehicle information (*14*). However, as is shown by Liu et al., such systems do not necessarily create string-stable platoons when they only consider

H. Zhou, R. Saigal, and L. Yang, Department of Industrial and Operations Engineering, University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109. F. Dion, Partners for Advanced Transportation Technology, University of California, Berkeley, Building 452, 1357 South 46th Street, Richmond, CA 94804-4648. Corresponding author: H. Zhou, haozhou@umich.edu.

Transportation Research Record: Journal of the Transportation Research Board, No. 2324, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp. 81–90.
DOI: 10.3141/2324-10

information delay with the leading or preceding vehicle (*9*). Another method to control a platoon of vehicles is parallel estimation (*15*), in which each vehicle first estimates the state of the entire platoon and then updates its estimates by communicating with other vehicles. In this case, a special communications network topology is needed to achieve stability.

GOALS AND ASSUMPTIONS

Goals

A higher-level longitudinal control algorithm is to be developed (details discussed in the next section) for a vehicle platoon that can work under an unreliable wireless communications environment and achieve the following goals:

1. Improve highway safety,
2. Increase highway capacity, and
3. Improve energy efficiency.

Hierarchy of Control

Two types of control are crucial to intelligent vehicles: longitudinal control (throttle and brake) and lateral control (steering). In this paper, it is assumed that lateral control can be readily established by a separate controller, and the focus is primarily on the longitudinal control aspect.

When longitudinal control decisions are made, use of a hierarchical control system implementing an upper and lower control level similar to the one described by Rajamani et al. is also assumed (*16*). At each time step, it is assumed that the upper-level controller determines the desired acceleration for each vehicle based on the following two objectives:

1. To maintain appropriate spacing between vehicles and
2. To ensure string stability of the platoon.

The acceleration decision is made on the basis of the perceived state of other vehicles. Because of communications latency, outdated information regarding the position and velocity of surrounding vehicles may be used in the decision. Once determined, the acceleration or deceleration decision is passed to the lower-level controller to be executed.

The lower-level controller is responsible for applying the throttle and brake actuator to ensure that the desired acceleration is achieved. The design of the lower-level controller is a complex problem because it is necessary to understand not only the mechanical system of engine transmission and drive train but also the effect on tire and road condition. It is also necessary to consider the mechanical latency between the upper- and lower-level controllers. Analysis of lower-level controllers (*16*) has been extensively considered in the literature. Since the project described in this paper mainly focuses on the upper-level control, it will be assumed that a lower-level controller is readily built and usable with constant mechanical latency.

Assumptions

To solve the vehicle platoon control problem, the following assumptions are made.

Full Automation

In terms of level of automation, there are three major types of systems:

1. Emergency warning systems that alert the driver when upstream incidents occur,
2. Semiautomatic cruise control systems that can take over parts or all of vehicle control but do not coordinate with other vehicles, and
3. Fully automatic control systems that can completely control the vehicle when on the highway and that will coordinate with other vehicles to maintain a safe distance and provide steering control to stay within a lane.

It was argued by Varaiya that although a partially automated system may improve safety, only full automation can achieve significant capacity increases (*3*). Therefore, it is assumed that all vehicles are fully controlled by computers.

Identical Vehicles

To simplify the model description, it is assumed that all vehicles within a platoon are identical. However, as is shown in the following section, this assumption can be relaxed by simply replacing a constraint (Equation 7) in the optimization problem to allow the modeling of different types of vehicles in the platoon.

Decentralized Control

Each vehicle has its own controller. In each time step, each vehicle makes its own decision on acceleration and steering control. There is no central controller for each vehicle. However, although it is making individual decisions, each vehicle needs to coordinate its actions with neighboring vehicles.

The benefit of decentralized control is twofold: first it requires less communication capacity than centralized control, thus reducing the likelihood of channel congestion. Second, the decentralized system is more robust because the overall safety of a platoon is not compromised if one or more controllers fail.

Vehicle Spacing Policy

Each vehicle is required to keep a safe distance from its preceding vehicle. There are many spacing policies to choose from (*17*). Among them the constant spacing and constant time headway spacing are frequently used for platoon control. Constant spacing refers to the policy of keeping a constant distance between consecutive vehicles no matter how fast they are traveling. Although this policy can achieve very great highway capacity, it may also lead to higher risks of collision when emergency braking occurs.

The spacing policy used in this project is the constant time headway policy, which tries to keep the ratio of vehicle spacing and velocity a constant. This policy has been shown to provide a high level of safety (*12*).

Leading Vehicle Control

The platoon control scheme does not include leading vehicle control. It was assumed that the motion of the leading vehicle is exogenous to the model, controlled either by a human driver or by an automatic guidance system.

Wireless Communications

Each vehicle is equipped with an IEEE 802.11p (DSRC) transceiver and sends out a message containing its position, speed, and acceleration (often known as the "Here I am" message). The following further assumptions were made for the communication system:

1. For each vehicle, messages are sent every K_s time steps from time 0.
2. After each message is sent, it takes τ_0 seconds for encoding and decoding the message.
3. When one message is sent, it is received by each vehicle independently with probability $1 - \rho$, where ρ is the message loss rate. This rate ρ was assumed to be a constant for all sender–receiver pairs and for all time.

DESCRIPTION OF CONTROL METHOD

Problem Analysis

Centralized Control with No Latency

If there is a platoon of N vehicles and a centralized controller that has perfect real-time information about every vehicle, the following is the basic state-space model for (global) control:

$$\dot{X} = AX + BU$$

where

$X = [x_1, \dots, x_N, \dot{x}_1, \dots, \dot{x}_N]^T$, where T = transposition of the matrix;
$U = [\ddot{x}_1, \dots, \ddot{x}_N]^T$;
x_i = position of ith vehicle; and
A and B = matrix of appropriate size.

This model will work if it is assumed that every vehicle can send its information (position, velocity, etc.) to the central controller without delay. However, in a real-world scenario, there is a delay in sending and receiving messages through a wireless channel. Usually the delay is a random variable. Liu et al. showed that if the delay is not a constant, the system is not guaranteed to be stable (*9*). Moreover, as is discussed in the previous section, the centralized control is vulnerable to disruption of wireless communications or failure of the central controller. A decentralized control model is now established to handle these issues.

Decentralized Controller with Latency

It is assumed that every vehicle in the platoon has its own controller. To make a control decision, each vehicle's controller needs to know how other vehicles are moving. So it is assumed every vehicle broadcasts its information to all other vehicles (thus every vehicle also receives information from all other vehicles).

Because of the communications latency, each vehicle may not have current information about other vehicles but most likely has the information sent by other vehicles a fraction of a second before.

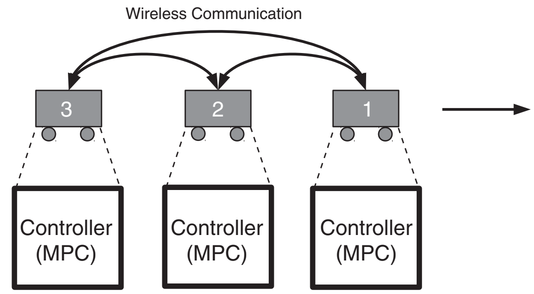

FIGURE 1 Communications and control scheme.

In order to properly handle this outdated information, a prediction model is used together with an optimization algorithm, also known as the model-predictive control (MPC) method, discussed in the next subsection. Figure 1 shows an overview of the communications and control scheme of a three-vehicle platoon.

MPC Method

The MPC method is used to control vehicles in the platoon. MPC has been applied in the processing industry and many other control applications (*18, 19*) and was implemented to control the lane allocation of intelligent vehicles (*20*).

MPC is based on a prediction model and an online optimization to obtain optimal control actions for the system. First the time horizon is discretized and the sampling period is set to T. At each time step k, the controller measures the current state of the system and uses a predictive model to predict the future states of the system, that is, from time step $k + 1$ to $k + K_p$, where K_p is the prediction horizon. Then the predicted future states are used as parameters of an optimization problem that minimizes some objective function $J(k)$ over the control variables $u(k), \dots, u(k + K_p)$. The general process of the MPC method is shown in Figure 2 [also in the work by Baskar et al. (*20*)].

In the following subsections, the components of the MPC method are discussed in detail.

Discretization of Time

Usually platoon-controlling models are continuous in time (*16*). However, the discretized-time model was chosen over the continuous-time model for the following reasons: (*a*) the discretized model is easier to fit in the prediction and optimization algorithm and (*b*) it coincides with the discretized nature of computerized automated control.

The time horizon is discretized into time intervals of length T s (i.e., the sampling period is T) and the control action $u(k)$ is applied when time $t = kT$, $k = 0, 1, \dots$, and holds constant within time period $[k, k + 1)$ (lower part of Figure 2). Similarly, it is assumed that messages are sent and received only at time $t = kT$, $k = 0, 1, \dots$. This assumption holds true in practical applications: normally communications devices have buffers that can hold messages sent or received during time $((k - 1)T, kT)$ and deliver them to the sender or receiver at time kT.

Sampling period T can be adjusted according to the needs of different applications. For instance, in the simulation experiments in the following sections, $T = 0.05$ was chosen after the trade-off between accuracy and simulation speed was considered.

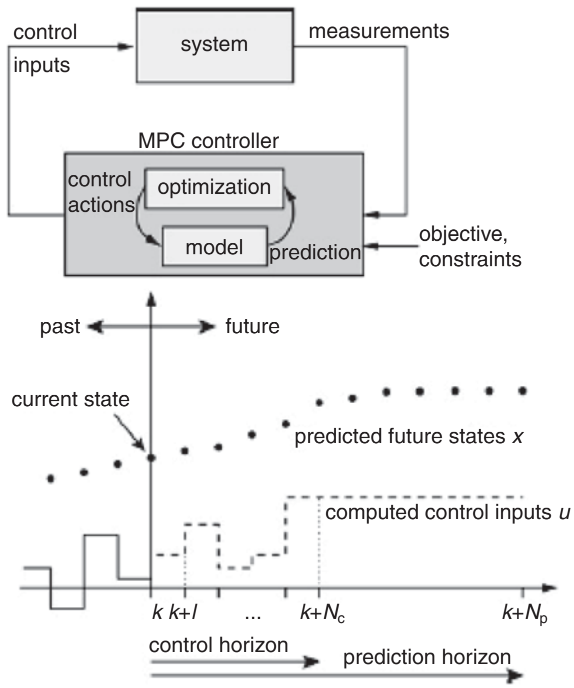

FIGURE 2 MPC control scheme.

Prediction Model to Handle Latency

A set of variables is defined that describes the car movement in discrete time: let $x_i(k)$, $v_i(k)$, and $a_i(k)$ be the position, speed, and acceleration of the ith vehicle at time kT, respectively. As discussed earlier, each car i at time kT sends $x_i(k)$, $v_i(k)$, and $a_i(k)$ to all the other cars in the platoon. But because of the stochastic nature of communications latency, this information arrives at destination car j at time kT with delay $\tau_{i,j}(k)$ time steps, where $j = 1, \ldots, N$. Therefore, at any given time kT, car j has information with different "ages" from different cars. Figure 3 demonstrates this asynchronous information transmission: white boxes represent information that is received by Car 2, whereas grey boxes represent information not known to Car 2. In this case, $\tau_{1,2}(k) = 3$, $\tau_{3,2}(k) = 2$, and $\tau_{4,2}(k) = 3$.

To handle this asynchronous information transmission, it is assumed that every vehicle has a buffer that can hold information of up to $\tau_{max} + 3$ past time steps, where τ_{max} is the maximum delay counted in time steps. With this buffer, this historic information about other cars and a statistical model can be used to fill in the gaps created by latency. An effective statistical prediction model is one that predicts how the other cars are moving (at the current time step) on the basis of the historical information stored in the buffer.

The statistical model used here to predict the movement of cars is the autoregressive and moving average with exogenous model, ARMAX(3,2,1) (*21*). The ARMAX model is a regression model that incorporates past observations of data as well as estimation errors to predict future data series. For example, if a vehicle j at time step k wants to know what vehicle i's speed is at the current time step, but it only has the speed of car i from period $k - \tau_{max} - 3$ to period $k - \tau_{i,j}(k)$, the following equation can be used to estimate the speed of vehicle i during period $[k - \tau_{i,j}(k) + 1, k]$:

$$\hat{v}_{i,j}(\kappa) = \phi_1 \hat{v}_{i,j}(\kappa - 1) + \phi_2 \hat{v}_{i,j}(\kappa - 2) + \phi_3 \hat{v}_{i,j}(\kappa - 3) + \varepsilon_i(\kappa) - \theta_1 \varepsilon_i(\kappa - 1) - \theta_2 \varepsilon_i(\kappa - 2) + \eta_1 \hat{a}_{i,j}(\kappa - 1)$$
$$\kappa = k - \tau_{i,j}(k) + 1, \ldots, k \qquad (1)$$

where

$\hat{v}_{i,j}(\kappa)$ = estimated speed of car i using car j's information at time κ,
$\hat{a}_{i,j}$ = estimated acceleration of car i at time κ,
$\varepsilon_i(\kappa)$ = estimation error of car i at time κ, and
ϕ, ε, η = coefficients estimated from data in buffer: $v_i(k - \tau_{max} - 3), \ldots, v_i(k - \tau_{i,j}(k))$, using least-squares estimate.

With the estimated speed, the estimated positions of all the other cars can be obtained. These estimates are fed into the optimization

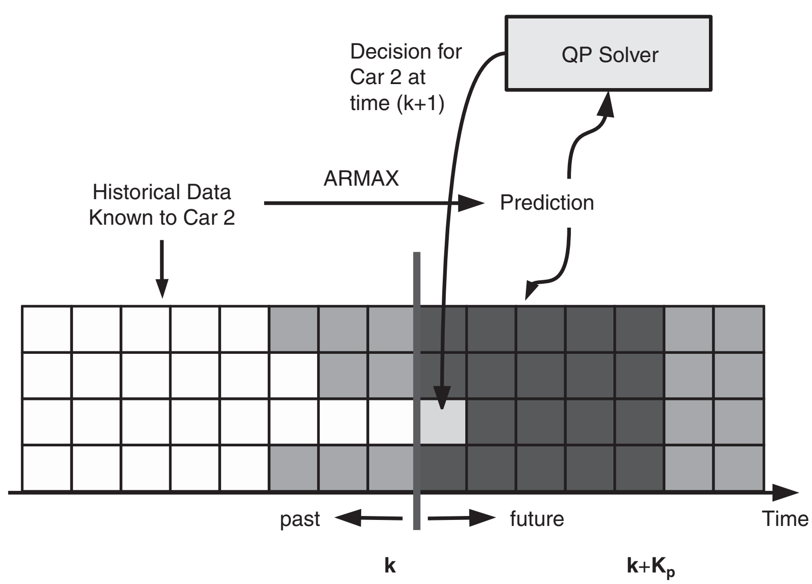

FIGURE 3 Decisions made by Car 2 at time step *k* (QP = quadratic programming).

problem $P(j, k)$ described next, which is then solved for optimal acceleration of car j at time step k.

Optimization Problem

Under the MPC framework, after the estimated speed and position of other cars have been determined, the controller of car j will construct an optimization problem to compute the optimal actions for the next K_p time periods. The optimization problem consists of an objective function $J(k)$ expressing the goal and a set of constraints that guarantee that the system is working in a specified manner and within certain conditions.

To achieve lower fuel consumption, one possible objective function is

$$J(k)=\sum_{i=1}^{N}\sum_{k=1}^{K_p}\left(a_i(k)\right)^2$$

By minimizing this objective function, one minimizes the amount of acceleration (or deceleration) in the following K_p time steps. However, in order to maintain the cars with equal spacing in a platoon, the following objective function is proposed:

$$J(k)=\sum_{i=1}^{N}\sum_{k=1}^{K_p}\left(a_i(k)\right)^2+W\cdot\sum_{i=1}^{N}\sum_{k=1}^{K_p}\left(x_i(k)-x_{i-1}(k)-Hv_i(k)\right)^2 \quad (2)$$

where H is the desired time headway (time used to travel the distance between two consecutive cars at current speed), and W is a constant that determines the penalty of deviating from the desired headway. The second term in the objective function penalizes actions that will bring two consecutive vehicles too close to or too far away from each other, and thus attempts to maintain a stable platoon system.

With the objective function (Equation 2), the optimization problem $P(j, \kappa)$ for car j, $j = 1, \ldots, N$ at time step κ is defined as follows:

$$P(j,\kappa): \quad (3)$$

$$\text{minimize } J(k)=\sum_{i=1}^{N}\sum_{k=\kappa}^{\kappa+K_p}\left(a_i(k)\right)^2+W\cdot\sum_{i=1}^{N}\sum_{k=\kappa}^{\kappa+K_p}\left(x_i(k)-x_{i-1}(k)-Hv_i(k)\right)^2 \quad (4)$$

subject to

$$x_i(k+1)=v_i(k)T+x_i(k) \qquad \text{for } i=2,\ldots,N;\ k=\kappa,\ldots,\kappa+K_p \quad (5)$$

$$v_i(k+1)=a_i(k)T+v_i(k) \qquad \text{for } i=2,\ldots,N;\ k=\kappa,\ldots,\kappa+K_p \quad (6)$$

$$a_{\min}\le a_i(k+1)\le a_{\max} \qquad \text{for } i=2,\ldots,N;\ k=\kappa,\ldots,\kappa+K_p \quad (7)$$

$$Lv_i(k)\le x_i(k+1)\le Uv_i(k) \qquad \text{for } i=2,\ldots,N;\ k=\kappa,\ldots,\kappa+K_p \quad (8)$$

where L and U are the lower and upper bound of time headway, respectively.

$$x_1(k)=\hat{x}_1(k) \qquad \text{for } k=\kappa,\ldots,\kappa+K_p \quad (9)$$

$$v_1(k)=\hat{v}_1(k) \qquad \text{for } k=\kappa,\ldots,\kappa+K_p \quad (10)$$

$$x_i(\kappa)=\hat{x}_i(\kappa) \qquad \text{for } i=2,\ldots,N \quad (11)$$

$$v_i(\kappa)=\hat{v}_i(\kappa) \qquad \text{for } i=2,\ldots,N \quad (12)$$

where $a_{\min}$ and $a_{\max}$ are the minimum and maximum accelerations, respectively. Constraint 7 guarantees that the maximum acceleration and deceleration will not exceed the car's mechanical limits. For the convenience of demonstration, all cars are set to have the same maximum acceleration, but one can easily change the value of these two parameters to apply the model platoons with nonidentical vehicles.

Constraints 5 and 6 describe the movement of every vehicle in time and space appropriately. Constraints 9 to 12 are the initial conditions of the model, the right-hand side of which comes from the prediction by the ARMAX model in the previous section.

Since this problem has linear constraints and a quadratic objective function, it is a quadratic programming (QP) problem, and thus can be efficiently solved with a QP solver.

Overview of Platoon Control with MPC

The overview of the control algorithm on Car 2 is shown in Figure 3. In the overview, it is assumed that the platoon has only four cars. Each box in the grid represents the state of one vehicle at a certain time step. The vehicle state contains the position, velocity, and acceleration. The white boxes represent information already known to Car 2, and the grey boxes represent information not known to Car 2 because of communications latency. As shown, Car 2 knows its every movement up to time step k, but only has information on Car 1 and Car 4 up to time $k - 3$. The black boxes represent predicted vehicle movement, which is generated by the ARMAX model. This predicted information will be used as the parameters of the QP model $P(j, \kappa)$ described earlier. Then the QP solver will give the optimal acceleration at time $k + 1$ for Car 2.

ANALYSIS OF ROBUSTNESS AND COMPUTATIONAL COMPLEXITY

Because the ARMAX model is used to predict future states of vehicles, it will inevitably introduce prediction errors into the MPC model. To test whether the MPC method is reliable enough, one needs to test how the prediction error affects the subsequent solution obtained by the optimization problem solved in each vehicle. Sensitivity analysis on the optimization problem $P(j, \kappa)$ is used to investigate how the optimal solution changes as the prediction of speed and position changes. The goal of this analysis is twofold: (*a*) to test under what conditions the solutions can be used to maintain a stable system, and (*b*) how large an estimation error can be tolerated by the optimization problem without jeopardizing the safety of the platoon.

General Form of QP

To achieve these goals, a general QP problem is used to start. First, the variable y is denoted as follows:

$$y=(\mathbf{a},\mathbf{v},\mathbf{x})^T$$

where

$\mathbf{a} = (a_1(k), \ldots, a_1(k + K_p), \ldots, a_N(k), \ldots, a_N(k + K_p))^T$
$\mathbf{v} = (v_1(k), \ldots, v_1(k + K_p), \ldots, v_N(k), \ldots, v_N(k + K_p))^T$
$\mathbf{x} = (x_1(k), \ldots, x_1(k + K_p), \ldots, x_N(k), \ldots, x_N(k + K_p))^T$

The QP problem $P(j, \kappa)$ defined in Equations 4 to 12, is restated as follows:

$$\text{minimize } y^T Q y \tag{13}$$

subject to

$$Ay = b \tag{14}$$

$$By \le c \tag{15}$$

where Q is a positive semidefinite matrix and A and b are the coefficient matrix and right-hand side of Equality Constraints 6 and 5 and 9 through 12, respectively. B and c are the coefficient matrix of the left-hand side and right-hand side of Inequality Constraints 7 and 8, respectively.

Sensitivity Function

Then a small perturbation ε is applied to the right-hand side of the equalities constraints, and the sensitivity function $y(\varepsilon)$ is defined as follows:

$$\bar{y}(\varepsilon) = \text{argmin}_y y^T Q y \tag{16}$$

subject to

$$Ay = b + \varepsilon \tag{17}$$

$$By \le c \tag{18}$$

Here b is the right-hand side of Constraints 9 through 12:

$$b = \begin{pmatrix} \hat{x}_1(\kappa), \ldots, \hat{x}_1(\kappa + K_p), \hat{v}_1(\kappa), \ldots, \hat{v}_1(\kappa + K_p), \\ \hat{x}_2(\kappa), \ldots, \hat{x}_N(\kappa), \hat{v}_2(\kappa), \ldots, \hat{v}_N(\kappa) \end{pmatrix}^T$$

The estimation error ε is defined as the difference between the estimated and the actual value of b. The goal is to derive the parametric function $\Delta y(\varepsilon)$:

$$\Delta y(\varepsilon) = \bar{y}(\varepsilon) - \bar{y}(0) \tag{19}$$

Boundary-State Analysis

To facilitate this analysis, the following definition is given: The system 6 through 12 at any given time k is defined as in a boundary state if there exist $i \in \{2, \ldots, N\}$ such that at least one of the following equalities is true:

$$a_{\min} = a_i(k)$$

$$a_i(k) = a_{\max}$$

$$Lv_i(k) = x_i(k)$$

$$x_i(k) = Uv_i(k)$$

Then the following assumption is made: When the system is not in a boundary state, there exists a small enough perturbation ε such that the system at the next time step is still not in a boundary state.

By making this assumption, it can be assumed that Inequalities 7 and 8 are strict inequalities under small perturbation ε. Now the Karush–Kuhn–Tucker conditions are derived as the system of linear equations:

$$Q(y + \Delta y) + A^T(\mu + \Delta\mu) = 0$$

$$A(\bar{y} + \Delta y) = b + \varepsilon$$

where μ is the Lagrangian multiplier. Thus the relation of ε to Δy can be expressed as

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta y \\ \Delta\mu \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix}$$

Because Q is a singular matrix, it is partitioned into four blocks:

$$\begin{bmatrix} Q_B & 0 & A_B^T \\ 0 & 0 & A_N^T \\ A_B & A_N & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta y_B \\ \Delta y_N \\ \Delta\mu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \varepsilon \end{bmatrix}$$

where Q_B is the columns and rows of Q that are not all zeros and Δy_B are corresponding rows in Δy. It can be shown that Q_B is a nonsingular matrix. Thus the following system of equations results:

$$Q_B \Delta y_B + A_B^T \Delta\mu = 0$$

$$A_N^T \Delta\mu = 0$$

$$A_B \Delta y_B + A_N \Delta y_N - \varepsilon = 0$$

Solving this system by substitution,

$$\Delta y_N = \left(A_N^T \left(A_B Q_B^{-1} A_B^T\right)^{-1} A_N\right)^{-1} \left(A_B Q_B^{-1} A_B^T\right)^{-1} \varepsilon \tag{20}$$

Thus Equation 20 reveals the linear relation between the error of estimation and deviation from optimal control actions. It indicates how control decision $a_i(k)$, $i = 2, \ldots, N$ is affected by the error of the ARMAX model. For any given time, if there are an upper bound and lower bound on estimation error ε, how large the control error can be in each time step can be numerically computed.

Computational Complexity of MPC

Since each vehicle solves the optimization problem in each time step, it is critical that the optimization problem be solved quickly, otherwise the decisions cannot be made online.

Using the sensitivity analysis in the previous subsection, the following method can be developed to greatly reduce the computational time. In each time step, the MPC controller of each vehicle

1. Checks whether the system is in a boundary state;
2. If it is in boundary state, calls the QP solver to solve the optimization problem; and
3. If it is not in a boundary state, lets ε be the difference between the new estimation value and the old estimation value and plugs into Equation 20 to calculate the change in acceleration.

Because Equation 20 is linear, various linear solvers can be used to efficiently compute the acceleration decisions, thus avoiding use of the potentially slower QP solver in each time step.

SIMULATION TEST

In order to test whether this MPC method will work effectively under harsh communications conditions, a simulation platform was set up to test the performance of this algorithm under two scenarios. The platform and some implementation details are described first, and then the results of the two test scenarios are presented.

Simulation Setup

Data Structure

The simulation test bed and the control algorithm are implemented in MATLAB. Three vectors of size N are used to respectively represent vehicle positions, velocities, and accelerations. These vectors combined can be considered as the global system state. Also, each vehicle is programmed as a separate object in the simulation. Each individual object has different perceptions of the system state because of different communications latency. As is described in the previous sections, each vehicle has a message buffer of size $\tau_{max} + 3$, storing the history of movement of other vehicles.

Generating Latency

Since it is assumed that each message sent has an independent loss ratio ρ, communications latency from vehicle i to j at time kT is generated in the following manner:

$$\tau_{i,j}(k) = \tau_0 + nK_s + \left(k - K_s\left\lfloor \frac{k}{K_s} \right\rfloor\right) \tag{21}$$

where n is the number of times a message is sent (or re-sent) and is a geometric distributed random variable with success rate $(1 - \rho)$. Because every message is sent only at a time step that is a multiple of K_s, the term $k - K_s\lfloor k/K_s \rfloor$ indicates the periods since the last message was sent (or re-sent).

Simulation Initialization

At the beginning of the simulation, the program is initialized by setting the system at the stable state, in which every vehicle is driving at 30 m/s (67 mph) and at a distance of 30 m (98 ft) apart. During the first τ_{max} periods, no acceleration is applied to any vehicle, and thus vehicles run at a constant speed. The reason for this initialization is to fill up the message buffer before enabling the ARMAX prediction model to work. After τ_{max} time steps, a series of acceleration and deceleration impulses is applied to the leading vehicle, and a record is made of how the other vehicles react.

In the following two sections, the MPC method is tested under two scenarios: first in a good and then in a harsh communications environment. In both cases, the same set of acceleration commands is used for the leading car. Table 1 gives a brief description of parameter settings and their meanings in the model.

TABLE 1 Parameter Values in Simulation

Parameter and Value	Definition
$T = 0.05$ s	Sampling interval (length of time step)
$N = 4$	Number of vehicles in one platoon
$K_p = 10$	Number of time steps considered when solving QP problem
$\tau_{max} = 50$	Maximum delay in wireless communication
$W = 200$	Penalty coefficient
$H = 1$ s	Desired time headway
$L = 0.5$ s	Minimum time headway
$U = 1.5$ s	Maximum time headway
$D = 1$ s	Desired time headway
$a_{min} = -12 m/s^2$	Minimum acceleration
$a_{max} = 8 m/s^2$	Maximum acceleration
$\tau_0 = 1$	Message encoding and decoding delay counted in time steps
$K_s = 2$ or 6	Message sent interval
ρ = 10% or 25%	Probability that a message is lost during transmission

Scenario 1. Low-Latency Wireless Communications

The first test scenario is to simulate vehicles driving at a normal state, with only a few disruptions in the wireless channel. Therefore $K_s = 2$, meaning that messages are sent every 0.1 s. The message loss rate is set at $\rho = 0.1$, which means that 10% of the messages are lost during one transmission. Figure 4*a* shows the acceleration of vehicles in the platoon. The acceleration of Car 1 is not controlled by the MPC controller but by a prespecified program input.

Figure 4*c* shows the average latency vehicles experienced along time. Although most of the time latency is at 0.1 s, it will occasionally spike to 0.3 s. Despite the latency, the effect of sudden braking and accelerating of Car 1 is damped as its effect propagates toward the end of the platoon. Figure 4, *b* and *d,* also demonstrate the speed and spacing between vehicles.

Scenario 2. High-Latency Wireless Communications

The second scenario is to test how the MPC method withstands noisy wireless communications. $K_s = 6$ is set, meaning that messages

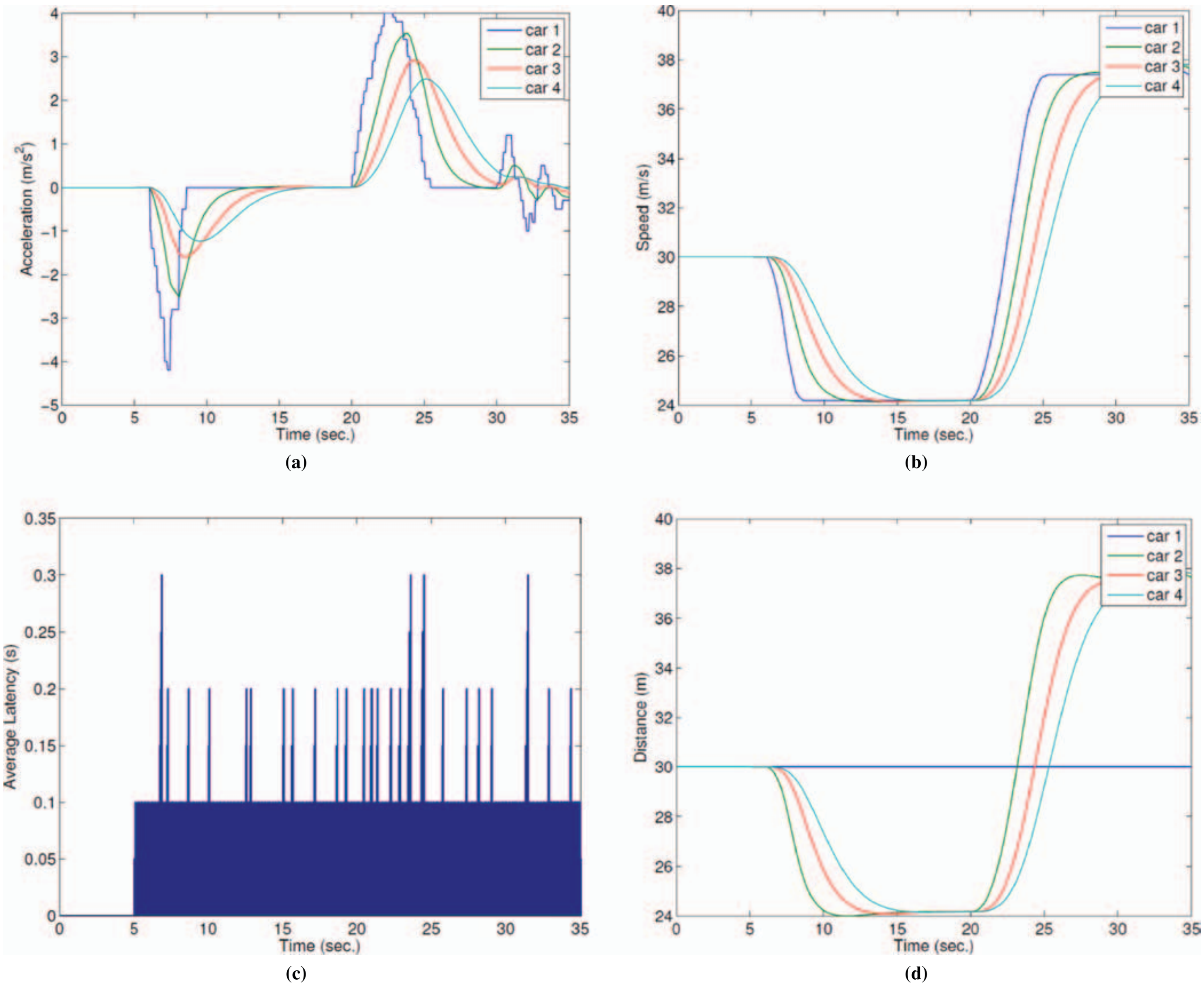

FIGURE 4 Scenario 1, low latency: (*a*) acceleration profile (L = 0.50, U = 1.50, D = 1.00, W = 200.000), (*b*) speed profile (L = 0.50, U = 1.50, D = 1.00, W = 200.000), (*c*) latency profile (T = 0.05, K_s = 2.0, τ_0 = 1.0, ρ = 0.10), and (*d*) spacing profile (L = 0.50, U = 1.50, D = 1.00, W = 200.000).

are sent every 0.3 s, and the message loss rate is set to $\rho = 0.25$, meaning that the probability that a message will be lost during transmission is 0.25. Figure 5*a* shows the acceleration of vehicles in the platoon. Acceleration of Car 1 is not controlled by the MPC controller but by a program input.

Figure 5*c* shows the average latency that vehicles experienced along time. In this case, communications latency is significantly higher than in the first case: most of the latency numbers are between 0.1 to 0.3 s, and at times they become more than 0.9. Although this case is very unlikely to happen in a real-world application, as is shown in experiments by Bai and Krishnan (*10*), it does provide a worst-case scenario to test the robustness of the MPC method.

The results indicate that large latency does affect the quality of control decisions of Car 2 and there are some jiggles in the acceleration of Cars 2 and 3. However, even in these extreme high levels of latency, the MPC algorithm still works properly and operates every vehicle safely.

CONCLUSION AND FUTURE WORK

A decentralized control method for controlling a platoon of vehicles under a high-latency communications environment is proposed. The MPC approach is used that combines a statistical prediction model with an optimization algorithm and gives optimal control action at each time step. The robustness of this method is also analyzed by using sensitivity analysis methods. Simulation experiments are performed to test the effectiveness and safety of this control method. It is shown that the MPC controller can react quickly to sudden braking or accelerating of the leading car and damps the effect of these actions as they propagate along the platoon. The simulation also demonstrates the potential of this method to operate vehicles safely in a severe communications environment.

Future research will include more extensive case studies to test controller performance under different parameter settings. Quantitative measurement of the performance of the control method (i.e., fuel effi-

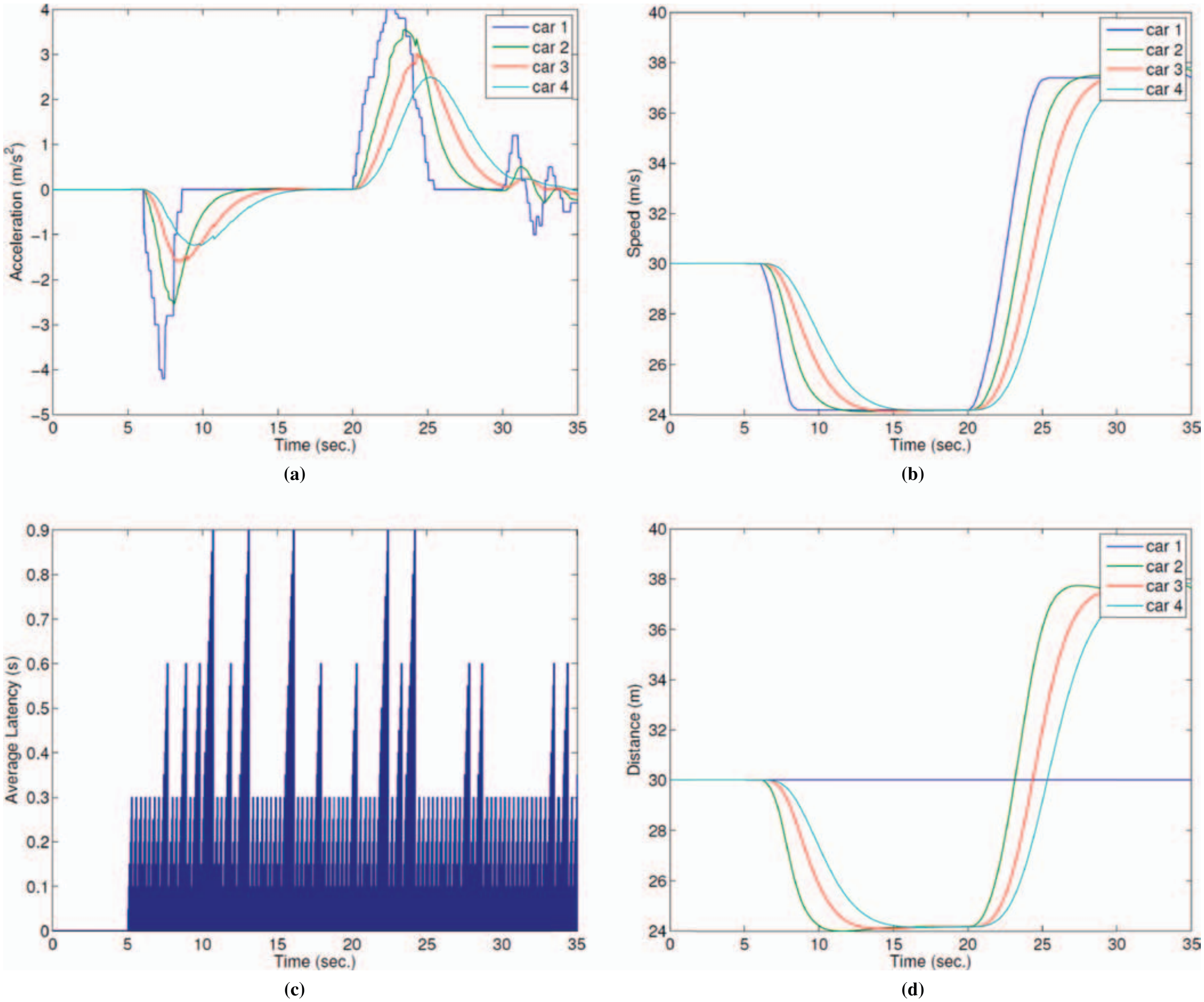

FIGURE 5 Scenario 2, high latency: (*a*) acceleration profile (L = 0.50, U = 1.50, D = 1.00, W = 200.000), (*b*) speed profile (L = 0.50, U = 1.50, D = 1.00, W = 200.000), (*c*) latency profile (T = 0.05, K_s = 6.0, τ_0 = 1.0, ρ = 0.25), and (*d*) spacing profile (L = 0.50, U = 1.50, D = 1.00, W = 200.000).

ciency, safety, ride quality) is needed to compare this model with other existing platoon control schemes. Another research topic is to reduce the size and complexity of the optimization problem so that it can be computed efficiently in inexpensive onboard computers.

ACKNOWLEDGMENT

This research was partly supported by a National Science Foundation grant.

REFERENCES

1. Horowitz, R., and P. Varaiya. Control Design of an Automated Highway System. *Proceedings of the IEEE,* Vol. 88, No. 7, 2000, pp. 913–925.
2. Varaiya, P., and S. Shladover. Sketch of an IVHS Systems Architecture. *Proc., Vehicle Navigation and Information Systems Conference,* Oct. 20–23, IEEE, New York, 1991, Vol. 2, pp. 909–922.
3. Varaiya, P. Smart Cars on Smart Roads: Problems of Control. *IEEE Transactions on Automatic Control,* Vol. 38, No. 2, 1993, pp. 195–207.
4. Rajamani, R., and S. E. Shladover. An Experimental Comparative Study of Autonomous and Co-Operative Vehicle-Follower Control Systems. *Transportation Research Part C: Emerging Technology,* Vol. 9, No. 1, 2001, pp. 15–31.
5. Biswas, S., R. Tatchikou, and F. Dion. Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety. *IEEE Communications Magazine,* Vol. 44, No. 1, 2006, pp. 74–82.
6. Baskar, L., B. De Schutter, J. Hellendoorn, and Z. Papp. Traffic Control and Intelligent Vehicle Highway Systems: A Survey. *IET Intelligent Transport Systems,* Vol. 5, No. 1, 2011, pp. 38–52.
7. Shladover, S. E., C. A. Desoer, and K. Hedrick. Automatic Vehicle Control Development in the PATH Program. *IEEE Transactions on Vehicular Technology,* Vol. 40, No. 1, 1991, pp. 114–130.
8. Gupta, P., and P. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory,* Vol. 46, No. 2, 2000, pp. 388–404.

9. Liu, X., A. Goldsmith, S. Mahal, and J. Hedrick. Effects of Communication Delay on String Stability in Vehicle Platoons. *Proc., Intelligent Transportation Systems Conference,* IEEE, New York, 2001, pp. 625–630.
10. Bai, F., and H. Krishnan. Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications. *Proc., Intelligent Transportation Systems Conference,* ITSC '06, Sept. 17–20, IEEE, New York, 2006.
11. Huang, C. L., Y. P. Fallah, and R. Sengupta. Adaptive Intervehicle Communication Control for Cooperative Safety Systems. *IEEE Network,* Vol. 24, No. 1, 2010, pp. 6–13.
12. Liu, Y., and F. Dion. Safety Assessment of Information Delay on Intelligent Vehicle Control System Performance. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1944,* Transportation Research Board of the National Academies, Washington, D.C., 2006, pp. 16–25.
13. Swaroop, D., and J. Hedrick. String Stability of Interconnected Systems. *IEEE Transactions on Automatic Control,* Vol. 41, No. 3, 1996, pp. 349–357.
14. Rajamani, R., and C. Zhu. Semi-Autonomous Adaptive Cruise Control Systems. *IEEE Transactions on Vehicular Technology,* Vol. 51, No. 5, 2002, pp. 1186–1192.
15. Smith, R., and F. Hadaegh. Closed-Loop Dynamics of Cooperative Vehicle Formations with Parallel Estimators and Communication. *IEEE Transactions on Automatic Control,* Vol. 52, No. 8, 2007, pp. 1404–1414.
16. Rajamani, R., H.-S. Tan, B. K. Law, and W.-B. Zhang. Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons. *IEEE Transactions on Control Systems Technology,* Vol. 8, No. 4, 2000, pp. 695–708.
17. Zhang, Y., B. Kosmatopoulos, P. Ioannou, and C. Chien. Autonomous Intelligent Cruise Control Using Front and Back Information for Tight Vehicle Following Maneuvers. *IEEE Transactions on Vehicular Technology,* Vol. 48, No. 1, 1999, pp. 319–328.
18. Maciejowski, J. *Predictive Control: With Constraints.* Pearson Education, Prentice Hall, 2002.
19. Camacho, E., and C. Bordons. *Model Predictive Control in the Process Industry: Advances in Industrial Control.* Springer-Verlag, Berlin, 1995.
20. Baskar, L., B. De Schutter, and H. Hellendoorn. Model-Based Predictive Traffic Control for Intelligent Vehicles: Dynamic Speed Limits and Dynamic Lane Allocation. *Proc., IEEE Intelligent Vehicles Symposium,* Eindhoven, Netherlands, June 4–6, IEEE, New York, 2008, pp. 174–179.
21. Rossiter, J. A. *Model-Based Predictive Control: A Practical Approach.* Control Series, CRC Press, Boca Raton, Fla., 2003.

The Intelligent Transportation Systems Committee peer-reviewed this paper.