Longitudinal Control of the Lead Car of a Platoon

Datta Godbole, John Lygeros

November 1993

PATH TECHNICAL MEMORANDUM 93-7

CALIFORNIA PARTNERS FOR ADVANCED TRANSIT AND HIGHWAYS
This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.
Abstract

We present longitudinal control laws for vehicles moving in an Intelligent Vehicle Highway System (IVHS) [1] [2]. In particular the scenario where cars move along the highway in tightly spaced platoons is considered. We present control schemes that perform the major longitudinal tasks that will be required from the lead vehicle of a platoon moving on an automated highway. More specifically schemes that maintain safe spacing, track an optimal velocity and perform various maneuvers (forming, breaking up platoons and changing lanes) are described. Simulation results are given in the Appendix.

1 Introduction

The work presented here was carried out with the particular IVHS structure of [3] in mind. In this context it is assumed that traffic in a lane travels in platoons. Inside a platoon all the vehicles follow the leader with a small intra-platoon separation of 1 meter. The inter-platoon spacing is assumed to be large so as to isolate the platoons from each other.

Obviously the control of such a large scale system poses a formidable problem. The control structure suggested in [3] consists of three layers. The top layer, called the link layer, coordinates the operation of the whole highway, the second layer, called the coordination layer, coordinates the operation of neighboring platoons and the bottom layer, the regulation layer, deals with the dynamics of individual vehicles. Our work focuses on the regulation layer.

Within the platooning framework the regulation layer operation can be divided into two main modes: leader and follower. Controllers for the latter have already been designed and tested both in simulation and in experiments (see [4], [5], [6], [7]). We propose a controller for the leader mode. Such a controller should be able to perform three main tasks. The first is to follow the preceding platoon at a safe distance. The distance between two platoons is chosen so that if the platoon in front applies maximum deceleration and stops, then the platoon behind should be able to respond, without a collision occurring. We consider a safety distance defined in the following way

\[ D_i = \lambda_a \ddot{x}_i + \lambda_v \dot{x}_i + \lambda_p \]

(1)

where \( \dot{x}_i \) and \( \ddot{x}_i \) denote the velocity and acceleration of platoon \( i \) respectively. For normal operation, we take \( \lambda_a = 0, \lambda_v = 1\text{sec}, \lambda_p = 10\text{m} \); in other words we use a constant time separation between two platoons. If the preceding platoon uses maximum deceleration to come to a complete halt the following platoon gets at least \( \lambda_v \) seconds to reach maximum deceleration and still come to a halt without a collision occurring. \( \lambda_p \) specifies the separation between platoons when they are stopped and also provides a safety margin (the "at least" term in the above sentence).

The second task that the controller should perform is to track an optimal velocity as closely as possible. The optimal velocity is calculated by the link layer and is such that the flow of cars on the highway is maximized. Of course maintaining the safety distance has a higher priority. A controller design
that performs both these tasks is described in Section 3.

Finally, in the platooning framework, the lead controller should perform a third task, namely engage the lead cars in certain maneuvers. Leaders should be able to join other platoons, take over when a platoon splits up and change lanes (or decelerate in order to facilitate lane changes). Controllers that calculate and track trajectories for such maneuvers are described in Section 4.

1.1 Control Objectives, Notation and Assumptions

Figure 1 shows two platoons numbered $i$ and $i-1$ following each other on an automated highway. The inter-platoon distance is defined as:

$$d_i = x_{i-1} - x_i - L_{i-1}$$

where $x_{i-1}$ and $x_i$ are the distances of the leaders of the individual platoons from a fixed roadside reference and $L_{i-1}$ is the length of the $(i-1)^{th}$ platoon.

The lead car of each platoon should maintain the inter-platoon distance equal to the safety distance given by (1) while following the speed of the platoon in front. If there is no car in the range of the distance sensor of the leader, then the lead car should follow the optimal speed provided by the link layer. These two objectives specify the steady state operation of the leader.

The lead vehicle will inevitably be disturbed from its steady state by the merge, split and change lane maneuvers. In addition to these routine maneuvers, there will be some occasional disturbances because of unexpected conditions on the road; for example, cars changing lane without communicating, break downs etc. Many of these disturbances can propagate along the highway and may even get magnified as they do so, causing an accident up stream of the disturbance. The lead controller should be able to respond to such emergencies while avoiding this slinky effect and maintaining its state as close to the steady state as possible.

Another important aspect to consider while designing a lead controller is the ride quality for the passengers and the limits on the achievable acceleration and jerk imposed by the car. The bounds on the acceleration are determined by the potential of the car and the road conditions. For our design we take them as $-5m/s^2$ for deceleration and $2m/s^2$ for acceleration. These are reasonable limits for normal operation as, under fair road conditions, modern cars can easily achieve and exceed them. Under adverse conditions (rain, breakdowns) the bounds on the acceleration can be considerably tighter. Investigation of the proposed controller performance under such conditions is currently under way. The bounds on the jerk are imposed primarily by the requirement for a comfortable ride and not by vehicle limitations. We take them as $\pm 5m/s^3$. Because the limits on the jerk are not imposed by saturation, they can be violated if there is some safety problem.

We assume that the lead vehicle is equipped with sensors which can provide measurements of the relative distance and speed of the vehicle in front, as well as its own velocity and acceleration. We also assume that the sensors can provide accurate measurements for distances up to 60 meters. Finally, we assume no communication between platoons. In this respect our lead controller can be classified as an Autonomous Intelligent Cruise Control (AICC) law.

With these assumptions and objectives in mind, we present the longitudinal controller for the lead car of a platoon. Section 2 describes the model of the car. In Section 3 we present the lead controller which will be used all the time except when the lead car is engaged in a maneuver. Section 4 outlines the design of specialized control laws for the individual maneuvers of merge, split and change lane. Simulation results are presented in the Appendix.

2 Model of the car

The equations used to model the motion of the car are the same as the ones used in [8] and [9]. The motion is summarized by a non linear, third order, ordinary
differential equation:

\[
\ddot{x}_i = b_i(\dot{x}_i, \ddot{x}_i) + a_i(\dot{x}_i)u_i \tag{2}
\]
\[
a_i(\dot{x}_i) = \frac{1}{m_i \tau_i(\dot{x}_i)} \tag{3}
\]
\[
b_i(\dot{x}_i, \ddot{x}_i) = -\frac{2K_{d_i}}{m_i} \dot{x}_i \ddot{x}_i - \frac{1}{\tau_i(\dot{x}_i)}[\ddot{x}_i + \frac{K_{d_i}}{m_i} \ddot{x}_i^2 + \frac{d_{mi}}{m_i}] \tag{4}
\]

where the subscript \(i\) indicates the leader of the \(i^{th}\) platoon. \(x_i\) is the position of this vehicle with respect to a fixed roadside reference (therefore \(\dot{x}_i, \ddot{x}_i\) are its velocity and acceleration, respectively), \(m_i\) is its mass, \(\tau_i\) is the time constant of its engine, \(u_i\) is the engine input, \(K_{d_i}\) is the aerodynamic drag coefficient and \(d_{mi}\) is the mechanical drag.

For the design of the longitudinal controller for the leader of a platoon the following simplifying assumptions were made about the state of the vehicle and the parameters of the model:

1. The whole state \((x_i, \dot{x}_i, \ddot{x}_i)\) can be measured directly, so that full state feedback is possible without the use of an observer.

2. \(m_i, \tau_i, K_{d_i}, d_{mi}\) are known. This is a strong assumption as these quantities are usually known only approximately and might change with time, even for the same car. However it allows us to linearize the model by state feedback as discussed in the next section (see also [8], [9]). An adaptive version of the controller may be designed at a later stage to relax this assumption somewhat (see [10]).

3 Intelligent Cruise Control Design

The controller design was carried out in 2 stages, as outlined in Figure 2. In the first stage (inner loop) nonlinear feedback was used to make the intermediate closed loop system input-output (from \(v\) to \(y\)) linear. In the second stage (outer loop) controllers for the linear system were designed.

3.1 Linearizing Control

For a particular class of nonlinear systems it is possible to find a state feedback control law such that the resulting closed loop system is linear from the input-output point of view. The conditions that characterize this class of systems can be found in the literature ([11] and [12]). The system in question, however, is simple enough to allow us to obtain the linearizing state feedback law by inspection, without having to go into the details of nonlinear control theory. The requisite control law is:

\[
u = \frac{1}{a(\dot{x}_i)}[-b(\dot{x}_i, \ddot{x}_i) + v]
\]

The resulting linear system is in a controllable canonical form:

\[
\frac{d}{dt} \begin{bmatrix} x_i \\ \dot{x}_i \\ \ddot{x}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \\ \ddot{x}_i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \tag{5}
\]

The objective now is to choose a suitable \(v\) to achieve the desired performance.

3.2 Linear Controller Design

Let us define two new variables:

\[
v_i = \frac{100(\dot{x}_{i-1} - \dot{x}_i)}{\dot{x}_i} \quad e_i = x_{i-1} - x_i - L_{i-1} - (\lambda_v \dot{x}_i + \lambda_p)
\]

where \(v_i\) is the relative velocity of platoons \(i\) and \(i-1\) expressed as a percentage of the velocity of the \(i^{th}\) platoon and \(e_i\) is the error between the actual and safe inter-platoon spacing. In these coordinates the steady state operating condition defined in the previous section is given by \(v_i = 0\) and \(e_i = 0\).

For small disturbances which keep the state close to the operating condition, we can use the following PID controller, suggested by S. Sheikholeslam [6]:

\[
\ddot{x}_i = v
\]
= c_i \int e_i \, dt + c_p e_i + c_v \dot{e}_i
+ k_v (\dot{x}_i - \dot{x}_i(0)) + k_a (\ddot{x}_i - \ddot{x}_i(0)) \tag{7}

Under this control law the closed loop transfer function relating the spacing error experienced by platoon $i-1$ to the spacing error experienced by platoon $i$ is:

$$H(s) = \frac{c_i(s)}{e_{i-1}(s)} = \frac{N(s)}{D(s)} \tag{8}$$

where,

$$N(s) = c_v s^2 + c_p s + c_i$$
$$D(s) = s^4 + (\lambda_v c_v - k_v) s^3 + (c_v - k_v + \lambda_v c_p) s^2 + (c_p + \lambda_v c_i) s + c_i$$

The controller parameters, $c_p, c_v, c_i, k_v$ and $k_a$, should be chosen so that:

- The transfer function $H(s)$ is stable.
- The impulse response $h(t)$ is greater than zero for all $t$, to ensure that there will be no overshoot. This is desirable as it guarantees that the disturbance will not be magnified as it propagates form one platoon to the next.
- $\|H(j\omega)\| < 1$, $\forall \omega$ This condition is required for disturbance attenuation and avoidance of the slinky effect between platoons. Refer to [8], [3] for details.

The parameter values we used were $c_i = 81$, $k_v = -24.75$, $c_p = 27$, $k_a = -9.75$, $c_v = 2.25$ which result to a transfer function $H(s)$ given by:

$$H(s) = \frac{(s + 6)^2}{(s + 3)^4} \tag{9}$$

For this choice of parameters the transfer function satisfies all the conditions listed above. The control law in (7) was tested by simulation. Although it guarantees rapid convergence to the desired equilibrium point (safe spacing and zero velocity mismatch), it can only be used locally, in a small neighborhood of this equilibrium point. The reason is that this controller tends to produce very large accelerations and decelerations (well outside the [-5,2] m/s^2 specification) even when presented with moderate displacements from its equilibrium. In a typical example the response due to a step input in the spacing error of a few meters (e.g. a reduction of 10 meters) displayed decelerations as large as 20 m/s^2. An input like this is plausible: it may correspond to a vehicle moving into the lane ahead of a platoon or the lead control taking over after a merge maneuver was aborted half way through. The reason for the poor performance is the fact that the linear control law in (7) requires exponential convergence of the state, which in turn requires large inputs.

In an effort to resolve the problem without extensive redesign, the location of the poles was augmented to produce a milder response. The improvement obtained in this way was minor. The accelerations produced for moderate inputs were still outside the required bounds. In a further attempt, a saturation function was introduced to “cut” the jerk input whenever the acceleration bounds were reached. Again the results were unacceptable — instability occurred in most cases.

Extensive simulation and experimentation with various control laws suggested that it is very unlikely that a single law will be able to produce acceptable performance (i.e. satisfy the acceleration and jerk bounds), when faced with various large displacements from the equilibrium point. Note that performance under such large displacements is an important requirement for an AICC control law in an IVHS framework, as will become apparent by subsequent comments. The controller we propose below utilizes different laws in different regions of the state space. Simulation results, summarized in the Appendix, indicate that, for realistic inputs, our controller satisfies the acceleration and jerk bounds. We are currently working on an analytic proof of the above statement.

We divide the $\epsilon - \nu$ plane of the state space of each leader into four regions according to Figure 3. The names of the regions indicate the nature of the control law used. The following cases represent typical incidents on a highway which will cause a transition of the state of leader from the nominal operating condition ($\epsilon_i = \nu_i = 0$) to one of the regions.

- **Region 1: Modified Pole Placement**
  Consider the state of platoon $i$ in the nominal operating condition $(0,0)$ and assume a vehicle changes lane in front of it, thus making $\epsilon_i < 0$. Then, if the velocity of the intruding vehicle is close to $\dot{x}_i$, the lane change will cause the state of the leader $i$ to move from the origin to some point in Region 1.
• Region 2: Track Self Velocity
Suppose there is no vehicle in front of platoon \( i \) for \( t < 0 \). At \( t = 0 \), a vehicle moving faster than \( \dot{x}_i \) changes lane in front of leader \( i \) at a distance which is less than the safety distance. This will cause the state of the \( i^{th} \) leader to move to some point in Region 2 where \( e_i < 0 \), and \( v_i > 0 \).

• Region 3: Track Velocity of Previous Car
Points in Region 3 can be reached if a vehicle changes lane in front of platoon \( i \) and its velocity is less than \( \dot{x}_i \), thereby making \( e_i < 0 \) and \( v_i < 0 \). The points in this region are critical from a safety point of view.

• Region 4: Track Optimal Velocity
There are two scenarios that can cause a leader to be in Region 4:

1. If there is no car in the range of the distance sensor of leader \( i \), then the only control objective is to track the optimal velocity provided by the link layer. The error in the safety distance is undefined in this case as there is no car in front and can be assumed to be \( e_i \gg 0 \).

2. In the previous scenario the distance sensor may detect a car far ahead (say at 60 m). If the velocity of this car is lower than \( \dot{x}_i \) the state of leader \( i \) will be such that \( e_i > 0 \) and \( v_i < 0 \). This corresponds to the fourth quadrant of the \( e - v \) plane. The opposite (a faster car) will lead to a point in the first quadrant.

Control laws were designed for each region taking into account passenger comfort and safety requirements. Then a switching mechanism was introduced to combine these different regional control laws to get a single controller. The goal of every regional control law is to take the system to the operating condition given by \( e_i = v_i = 0 \). At steady state we will use the PID controller of (7), that will take care of minor disturbances. The movement of other vehicles on the highway will change the state of the leader \( i \) from the steady state operating point to a point in one of the regions. Then the unified control law will take it back to the steady state operating point without violating the constraints. We now describe in detail the working of the individual controllers.

### 3.2.1 Region 1: Modified Pole Placement

Here \( e_i < 0 \) and the velocity mismatch is negligible. For this case, we modify the PID law of (7) in the following manner.

Define

\[
\dot{e}_i = e_i - \delta \lambda_p \tag{10}
\]

where

\[
\delta \lambda_p(0) = e_i(0) \tag{11}
\]

\[
\dot{\delta \lambda_p} = -\eta_p \text{sgn}(\delta \lambda_p) \tag{12}
\]

Now use \( \dot{e}_i \) instead of \( e_i \) in the controller equation (7). From equation (12), \( \delta \lambda_p \) returns to zero in finite time, in fact:

\[
\delta \lambda_p = 0 \quad \text{for} \quad t \geq e_i(0)/\eta_p \tag{13}
\]

which implies that:

\[
\dot{e}_i = e_i \quad \text{for} \quad t \geq e_i(0)/\eta_p \tag{14}
\]

Thus instead of reacting to a disturbance instantaneously, we can control the rate at which the system responds so that the trajectory remains inside the passenger comfort limits. In other words, we are converting the step change in \( e_i \) to a ramp change in \( \dot{e}_i \) (corresponding to a step change in \( v_i \)).

Note that this modification works only for \( |e_i(0)| \leq \lambda_p \). For \( |e_i(0)| > \lambda_p \), we can introduce \( \delta \lambda_v \) to get the same result in the following way. Consider
\[ \delta \lambda_p \text{ and } \delta \lambda_v \text{ defined as} \]
\[ \delta \lambda_p(0) = \lambda_p \text{sgn}(e_i(0)) \]
\[ \delta \lambda_v = -\eta_v \text{sgn}(\delta \lambda_p) \]
\[ \delta \lambda_v(0) = (\lambda_v + c_i(0))/\ddot{x}_i(0) \]
\[ \delta \lambda_v = -\eta_v \text{sgn}(\delta \lambda_v) \]

Redefine \( \dot{e}_i = e_i - (\delta \lambda_v \dot{x}_i + \delta \lambda_p) \) and apply
\[ \ddot{x}_i = c_i \int \dot{e}_i dt + \dot{\kappa}_p \dot{e}_i + \kappa_c \dot{e}_i + \kappa_c (\dot{x}_i - \ddot{x}_i(0)) \]

where
\[ \dot{\kappa}_p = \dot{c}_p - \delta \lambda_v c_i \]
\[ \kappa_a = k_a + \delta \lambda_v c_v \]
\[ \kappa_c = k_c + \delta \lambda_v (c_p - \lambda_v c_i - \delta \lambda_v c_i) \]

As before \( \delta \lambda_p, \delta \lambda_v \), \( \dot{e}_i - c_i \) converge to zero in finite time. Note that in this case the poles of the transfer function could shift because they depend on \( \lambda_v \) (as can be seen from equation 8). To avoid this we change the controller parameters \( (c_p, k_a, k_c) \) according to (15) to guarantee that the closed loop poles remain at \( s = -3 \).

### 3.2.2 Region 2: Track Self Velocity

In this region, the car in front is closer than desired but moving faster: \( e_i < 0 \) and \( v_i > 0 \). Therefore if the leader \( i \) keeps moving at constant velocity then, after some time, \( e_i \) will be greater than zero and the state will drift into the track optimal velocity region. The following control law will guarantee that the velocity remains constant.
\[ \ddot{x}_i = k_1 \ddot{x}_i + k_2 (\dot{x}_i - \dot{x}_i(0^-)) + k_3 (x_i - x_i(0^-) - \int \dot{x}_i(0^-) dt) \]

where we use \( k_1 = -1.5, k_2 = -0.75, k_3 = -0.125 \) to place the poles at \(-0.5\).

### 3.2.3 Region 3: Track Velocity of previous car

In this case, the car in front is closer than desired and moving slower, \( e_i < 0 \) and \( v_i < 0 \). As the safety distance is affine in the velocity, during the process of bringing \( v_i \) to zero, the safety distance and hence \( e_i \) are constantly changing. A controller of the form of (7) puts more emphasis on getting \( e_i \) to zero, whereas in this case, the velocity mismatch is more serious. Thus we design a state feedback controller which puts equal emphasis on \( v_i \) and \( e_i \). We define a desired velocity \( \dot{x}_{di} = \dot{x}_{i-1} \) and the desired spacing \( x_{di} = x_{i-1} - L_{i-1} - (\lambda_v \dot{x}_i + \lambda_p) \) and use a control law of the form:
\[ \ddot{x}_i = k_1 \ddot{x}_i + k_2 (\dot{x}_i - \dot{x}_{di}) + k_3 (x_i - x_{di}) \]
\[ = k_1 \ddot{x}_i + k_2 (\dot{x}_i - \dot{x}_{i-1}) - k_3 e_i \]

where \( k_1 = -3, k_2 = -3, k_3 = -1 \) to place poles at -1. As this situation is safety critical for some values of \( c_i(0) \) and \( v_i(0) \), the passenger comfort standards may be violated for a small amount of time until we attain the desired safety distance.

### 3.2.4 Region 4: Track Optimal Velocity

As mentioned above, this region has to deal with two cases:

1. No car is within the sensor range or the car ahead is far away \( (e_i > 0) \) and moving faster than the optimal velocity. In this case the task is to track the optimal velocity provided by the link layer. This is done by creating a desired trajectory, \( x_{di}(t) \), that will bring the vehicle from the current to the optimal velocity. The trajectory we chose was obtained by linearly interpolating between the two velocities. Once the trajectory is available a feedback law like the one in (16) is used to guarantee tracking.

2. The previous car is far and is moving at optimal speed or less. Then the task is to track the velocity of the preceding car and maintain the safety distance. This scenario is very similar to the one in the Track velocity of previous car region. We will create a trajectory that will produce the desired car velocity and spacing corresponding to the car in front and then apply a control law similar to the one described above:
\[ \ddot{x}_i = k_1 \ddot{x}_i + k_2 (\dot{x}_i - \dot{x}_{i-1}) - k_3 e_i \]

with \( k_1 = -1.5, k_2 = -0.75, k_3 = -0.125 \). The only difference is that the poles are closer to the origin in this case, as the situation is not safety critical and slower (and therefore more comfortable) response is acceptable.

### 3.3 Smooth Switching

Whenever the state of the leader goes from one region to the other, the control law should also change
We want each $g_j$ to be

- Smooth function of $x$
- $\approx 1$ in the $j^{th}$ region
- $\approx 0$ outside the $j^{th}$ region

Consider the sigmoid function:

$$g(x) = \frac{1}{1 + e^{-\alpha (x-\beta)}}$$

For $\alpha$ large enough, $g(\cdot)$ fulfills all the above requirements (i.e., $\approx 0$ for $x < \beta$ and $\approx 1$ for $x > \beta$). We use such sigmoid functions as basic building blocks to get the actual switching functions. More specifically, at every point each scaling function is the product of two such sigmoids, one with $x = \epsilon_i$ and one with $x = v_i$. The values of $\beta$ in each case is chosen so that the sigmoid reaches 0.5 on the boundary of the corresponding region. This makes the transitions between the different regional controllers smooth.

Performance analysis with the unified controller under certain simplifying assumptions is presented in the Appendix, along with simulation results for typical traffic scenarios. We present the design of specialized control laws for merge, split and change lane in the next section.

### 4 Maneuver Control Laws

The controller described in the previous section is designed to maintain safe spacing and track the optimal velocity as closely as possible. Apart from this default mode of operation, however, the leader of a platoon will occasionally have to engage in other maneuvers as the platoon moves along the automated highway. There are three such basic maneuvers: /em merge, split and change lane (see [3] and [13] for the motivation behind this choice of maneuvers and their coordination layer implementation). All three have some common features and therefore the design of the controllers that carry them out is very similar.

**Merge** is the action taken by two platoons that want to become one. Of course both platoons should be in the same lane. The following platoon, which is the one that requests the merge, should accelerate, catch up with the leading platoon and join it. **Split** is exactly the opposite: a follower becomes the leader of all the cars that follow it (in the same platoon) and decelerates to a safe distance from the mother platoon. Finally **Change Lane**, the most complicated of the three maneuvers, is used whenever a free agent (one car platoon) wants to move from one lane to the next; it is mandated that only free agents will be able to carry out this maneuver. Lane changing involves the deceleration of either the free agent that wants to change lane or the platoon in the lane that the free agent wants to move to. The deceleration is such that the free agent ends up in a position relative to the adjacent lane traffic from where it can safely move over. So, apart from the obvious lateral action, change lane also involves longitudinal control.

It is clear that all these maneuvers involve both coordination and regulation layer control. The protocols required in the coordination layer have already been developed and verified (see [3] and [13]). We will only deal with how the regulation layer responds to commands issued by the coordination layer requesting some maneuver. The coordination of the two layers will be the subject of further research.

Note that split can already be handled by the controller described in the previous section (see for example Figure 11). Merge and change lane may also be performed in a similar way. These maneuvers, however, will be routinely carried out by the leader vehicles, therefore they need to be optimized with respect to time. If the maneuvers are carried out by the cruise controller of Section 3 they need about 20 seconds and utilize only a small percentage of the potential of the car for acceleration/deceleration. Therefore maneuvers under cruise control will be used only in the case of an emergency (e.g. aborted maneuver or a car changing lane without communicating). For regular operation, maneuvers that make use of a greater percentage of the cars potential while staying well within the required limits, were designed. The controllers considered for this task involve two stages:
first a trajectory that brings the cars from their current to their desired positions is calculated, then feedback is applied to guarantee trajectory tracking.

4.1 Merge

The objective is to take the vehicles from an initial spacing of $d_0$ (typically 30 meters) and velocity mismatch $\delta v_0$ (typically a few meters per second) to a final spacing equal to the intra-platoon spacing (typically 1 meter) and zero velocity mismatch. The whole maneuver should be carried out as fast as possible, but without going outside the limits of acceleration and jerk set above. No communication is assumed while the maneuver takes place so the controller has to rely only on the sensor readings for position and velocity of the car ahead.

The trajectory, $x_d(t)$, used is shown in Figure 4, where $\dot{a}_{\text{max}} = -\dot{a}_{\text{min}} = 5m/s^3$ and $a_{\text{max}} = -a_{\text{min}} = 2m/s^2$. Note that the time intervals $t_1$, $t_3 - t_2$, $t_5 - t_4$ are completely specified by the limits on acceleration and jerk and the initial acceleration $a_0$. Hence, the problem reduces to calculating $t_2 - t_1$ and $t_4 - t_3$ to achieve the desired final spacing and velocity. More specifically these time intervals should be chosen so that the area under the acceleration curve is equal to $\delta v_0$ while the area under the velocity curve is equal to $d_0$ minus the intra-platoon spacing. This leads to two equations, one linear and one quadratic, that, together with the three equations for $t_1$, $t_3 - t_2$, $t_5 - t_4$, can be solved to obtain the times $t_1$, $t_2$, $t_3$, $t_4$ and $t_5$.

Note that the above trajectory is designed under the assumption that the acceleration of the preceding platoon is zero, which, of course, need not necessarily be true. This is the main reason (in addition to stabilizing about the trajectory in case of disturbances) why asymptotic tracking is needed. The feedback controller used for this purpose is:

$$\ddot{x}_i = \ddot{x}_d + k_2(\dot{x}_i - \dot{x}_d) + k_1(\dot{x}_i - \dot{x}_{i-1} + \dot{x}_d) + k_0(x_i - x_{i-1} - L_{i-1} + x_d)$$

where $L_{i-1}$ is the length of the car and $k_2 = -9$, $k_1 = -27$, $k_0 = -27$ to place poles at -3. Note that because we assume that no communication between the platoons will take place, only the distance and velocity of the leading platoon, which are measured through sensors, are available to the controller. This is why the acceleration of the preceding platoon is not used in the feedback law.

The controller described above was implemented and tested in simulation. Typical results are given in the Appendix.

4.2 Split

The objective of a split maneuver is to take a pair of vehicles, initially at intra-platoon spacing and zero velocity mismatch, to inter-platoon safe spacing and zero velocity mismatch. The limits of acceleration and jerk should be maintained and we assume that communication is interrupted the moment the split begins, so that the maneuver has to be completed using only sensor readings. The trajectory used is the opposite of the one used for merge, but simplified.

Figure 4: Trajectories for Merge

Figure 5: Trajectories for Split
because both initial and final velocity mismatch is zero (Figure 5).

The jerk and acceleration limits are the same as above. Again the design involves choosing the time intervals \( t_2 - t_1 \) and \( t_4 - t_3 \) to achieve the desired spacing. Note that the final spacing depends on the initial velocity of the vehicles. The safe spacing at the end of the maneuver, however, depends on their final velocity. This is only a minor problem because the difference between final and safe spacing will be small (in the order of a few meters). As demonstrated in Section 3 the AICC control law is perfectly capable of dealing with such a mismatch, when it is activated after the completion of the split maneuver.

Alternatively a slight modification of the tracking controller can be used. Let \( d_{safe0} \) be the safe spacing used in the calculation and \( d_{safe} \) the safe spacing based on the current velocity of the car ahead. Then the tracking controller can be set to:

\[
\ddot{x}_i = \ddot{x}_d + k_2(\ddot{x}_i - \ddot{x}_d) + k_1(\dot{x}_i - \dot{x}_d) + k_0(x_i - x_{i-1} - \dot{x}_d) + k_0(x_i - x_{i-1} - L + x_d + (d_{safe} - d_{safe0}))
\]

(18)

By penalizing the mismatch between desired and actual safe spacing we force the final spacing closer to what it should be.

### 4.3 Change Lane

The longitudinal action required for changing lanes is summarized in the Figure 6.

Free agent A wants to change to the lane where platoon B is moving. It is assumed, for purposes of safety, that A can move over only if its speed is close to that of B and their spacing is close to the safety distance. There are three scenarios that allow A to move over in safety: A has to decelerate and move behind B, B has to decelerate and let A in ahead of it or B has to split and let A enter in the middle. Which of the three alternatives is chosen is decided by the coordination layer — the immediate superior of the regulation layer in the control hierarchy. The regulation layer is only asked to carry out the chosen maneuver.

All three maneuvers can be done by some more elaborate form of split (described above). The third maneuver involves a split of a platoon to a larger spacing than safety (approximately twice as much). The first can be visualized as a split of A from an imaginary platoon with the position and velocity of B but in the same lane as A (Figure 7). A similar technique can be used for the second scenario. The desired trajectories are therefore slightly complicated versions of the ones given for split.

The complication arises from the fact that, unlike the usual split case, both the initial spacing and the initial velocity mismatch are non-zero and may take values (positive or negative) in a very wide range. In contrast, the usual split only has to deal with situations where both cars move at the same speed and their original spacing is equal to a fixed, positive value. It turns out that it may not be possible to find a trajectory for the given pair of \( a_{max} \) and \( a_{min} \) in some cases. For example, suppose that the following situation is encountered: the platoons are initially close to the safety distance but the one that is ahead moves much faster. In this case starting by decelerating A up to \( a_{min} \) will lead to an error in the calculation of the time intervals \( t_2 - t_1 \) and \( t_4 - t_3 \) (more specifically \( t_2 - t_1 < 0 \)). This is, of course, unacceptable in the trajectory calculations. To be able to deal with problems like this, the controller adapts the values of \( a_{max} \) and \( a_{min} \) during the calculations. For example, in the above case when \( t_2 - t_1 < 0 \) is obtained, \( a_{min} \) is increased and the calculation is repeated. Similarly, whenever \( t_4 - t_3 < 0 \) is obtained,
Figure 7: Splitting from an imaginary platoon

\[ a_{\text{max}} \] is decreased. These modifications go on until acceptable values of \( t_2 \) and \( t_4 \) are reached. In extreme cases \( a_{\text{min}} > a_{\text{max}} \) may be needed. This implies that the platoon starts by accelerating (bringing the velocity mismatch down) and finishes by decelerating. In other words, it merges with an imaginary platoon situated at the safety distance and moving with the same velocity as B, without any inter-platoon spacing (Figure 8).

To insure that as few maneuvers as possible are aborted, a similar technique was also used in the calculation of the merge trajectories. Merge is ideally invoked when the cars are moving at similar speeds and approximately safe spacing, therefore this is probably redundant. It was added to make sure that unusual situations can also be handled, thus giving more flexibility to the design.

A control law based on such trajectories and the asymptotic tracking feedback in (18) was simulated. Typical results are given in the Appendix.

The biggest problem with the change lane longitudinal trajectories is deciding when to abort them. While decelerating to allow a change lane the platoon or free agent may need to take actions to avoid a hazardous situation (e.g., a broken down car in its lane). The criteria for aborting the deceleration and what to do with the change lane attempt from then on are not entirely clear and they are the subject of further research. Similar decisions for merge and split are a lot simpler as they involve cars in a single lane.

5 Conclusions

We presented a longitudinal control law for the lead vehicle of a platoon in an automated highway. The proposed controller achieves a delicate balance between safety and passenger comfort. Furthermore, because it does not rely on any communication between platoons, it can be used even without having a fully automated highway system, as an AICC law. Additional, specialized merge, split and change lane controllers were also designed to complete the maneuvers in minimum time, with safety and comfort.

The control laws successfully passed the simulation test. Some of the results are shown in the plots in the Appendix. To carry out more thorough tests some decision making capability was added to the regulation layer, in the form of a coordinating discrete event system. This addition also helped to make the regulation layer as autonomous as possible and to highlight its interaction with the coordination layer. The analysis of this interface design is presented in [14]. The resulting combination of the coordinator and the controllers presented here is currently being tested in the framework of the SmartPath simulation package (see [15]). The complete regulation layer was used to introduce dynamics into SmartPath and therefore help with the design and testing of the link layer controller. Hopefully the simulation results will also give some useful insight into the hybrid control problem obtained by coupling discrete event and continuous time controllers.

Note that the stability of this controller still remains to be proved. This, along with the verification of the coordinator and the hybrid system resulting from the interaction, is the subject of current research. In the long run, the controller will be tested on real cars and, if necessary, an adaptive version will be developed to estimate some of the model parame-
ters and update the control scheme accordingly.

Acknowledgment: The authors would like to thank Farokh Eskafi, Dr. Bobby Rao, Prof. Shankar Sastry and Prof. Pravin Varaiya for helpful discussions providing insight into this problem.

References


A Performance using a unified controller

The unified controller was tested by extensive simulations using SmartPath [15]. The controller performed well and did not result in any collisions when faced with typical highway situations. Typical simulation results are shown in the next section.

Although it is difficult to prove stability/performance for this kind of variable structure controller, we will provide some conservative bounds on its performance when it is faced with typical conditions on a highway. For this purpose, we will assume that the $i^{th}$ platoon is operating in its steady state. At $t = 0$, a disturbance is created in front of it (either a step in $e_i$ or a step in $v_i$ or both). We also assume that for $t > 0$, the acceleration of $(i-1)^{th}$ platoon is

---

11
zero. It is shown in [16] that, for any disturbance that takes the state of the platoon leader to any point in the regions Modified Pole Placement, Track Optimal Velocity or Track Self Velocity; the unified controller of Section 3 will return the state of the leader to its steady state operating point without a collision occurring.

This is not true for all the points of the Track Velocity of Previous Car region however. Suppose the \( i^{th} \) platoon is moving at 25 m/sec. If a vehicle changes lane 15 m in front of it (recall that safety distance at 25m/sec is 35m), then its velocity should be at least 3 m/sec to avoid collision. Note that this type of severe disturbance demands large accelerations from the leader of the \( i^{th} \) platoon. The collisions occur because the performance of the controller is limited by the capabilities of the vehicle. The following table provides the minimum velocity for vehicles which are at a less than safe distance from the \( i^{th} \) platoon leader so that a collision will not occur.

(see [16] for more details)

<table>
<thead>
<tr>
<th>( \dot{x}_i ) m/s</th>
<th>Safety Distance ( e_i ) m</th>
<th>Actual Separation ( e_{i-1} ) m</th>
<th>( \dot{x}_{i-1} ) m/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>35</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td>35</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>25</td>
<td>35</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>10</td>
<td>0.5</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>5</td>
<td>4.5</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The higher layers of the hierarchy (coordination and link layer) should make sure that the state of the lead controller will remain in the region of guaranteed performance as calculated above.

## B Simulation Results

The lead controller was successfully tested in simulations. We have attached certain illustrative plots at the end. In all cases at time \( t = 0 \) the platoon \( i \) is considered to be in its steady state with a velocity of 20m/sec. The plots correspond to following cases:

**Figure 9** The car in front of leader \( i \) slows down to 15m/sec resulting in a step change in velocity of 5m/sec. Note that the acceleration and jerk are within limits. The controller operates in the regions Track optimal velocity and Modified pole placement.

**Figure 10** A car changes lane in front of platoon \( i \) at a distance which is 20m less than the required safety distance and a velocity same as that of platoon \( i \). The controller gives a result which is well within bounds. The time of response can be changed by changing the parameters \( \eta_p \) and \( \eta_v \). As the initial error of 20m is greater than \( \lambda_p \), both \( \delta \lambda_p \) and \( \delta \lambda_v \) get activated. You can see that one of them returns to zero before the other, creating a small step in the velocity profile. The controller operates in the region Modified pole placement.

**Figure 11** This is the same situation as Figure 10, but the step in distance is 29m, leaving a separation of 1m to begin with. This shows that the default controller can work as a controller for split as well, but it takes a long time to complete the maneuver (approx 25 seconds). Thus, in the case of an emergency break up of a platoon, the cruise controller can take the platoon to safety. The controller operates in the region Modified pole placement.

**Figure 12** A car changes lane in front of platoon \( i \) reducing the safety distance by 10m but it is going faster than platoon \( i \) by 6m/sec. Thus platoon \( i \) keeps moving at the same velocity until \( e_i \) becomes positive. Then it tracks the optimal velocity of 23m/sec. The controller operates in the regions Track self velocity, and Track optimal velocity.

**Figure 13** Platoon \( i \) has no cars in its distance sensor for \( t < 0 \). At \( t = 0 \), it sees a car 60m ahead going at a much lower speed of 5m/sec. Here \( u_i < 0 \) and \( e_i > 0 \). Thus there is no immediate safety concern. But instead of waiting for \( e_i \) to become negative, the controller starts acting immediately and gets \( v_i \) to 5m/sec while inside the comfort limits. The controller operates in the regions Track optimal velocity and Modified pole placement.

**Figure 14** A slow car changes lane in front of platoon \( i \). Thus \( e_i \) and \( v_i \) both have a negative step
of 10. This is a safety critical situation. Thus, momentarily the jerk is above limits but then comes back in limits immediately. The controller operates in the regions *Track velocity of previous car* and *Modified pole placement*.

**Figure 15** The platoon merges to the one ahead. The leading platoon is decelerating at $0.5m/s^2$ while the maneuver takes place. This distorts the trajectories somewhat as the feedback tries to take care of the extra deceleration. Still, the position and velocity objectives are achieved (zero final velocity mismatch and intra-platoon spacing of $1m$).

**Figure 16** The car becomes a leader and splits from the mother platoon. The mother platoon accelerates at $0.5m/s^2$ while the maneuver takes place. Again the feedback distorts the trajectories but manages to achieve the objectives. Note that the final safety spacing depends on the final velocity.

**Figure 17** The platoon decelerates to facilitate a change lane. The maneuver is similar to a split from an imaginary platoon (refer to Figure 7). Note that in this case initial spacing may be less than zero.

**Figure 18** Same as above. In this case however due to the large mismatch in initial velocities, the maneuver looks more like a merge to an imaginary platoon (refer to Figure 8).

**Figure 9**: Velocity step of $-5m/s$. The car in front of leader $i$ slows down to $15m/sec$ resulting in a step change in velocity of $-5m/sec$.

**Figure 10**: Spacing step of $-20m$. A car changes lane in front of platoon $i$ at a distance which is $20m$ less than the required safety distance with the velocity same as that of platoon $i$. 


Figure 11: Spacing step of $-29m$ (emergency split). This is same situation as previous Figure, but the step in distance is $29m$ leaving a separation of $1m$ to begin with. This shows that the default controller can work as a controller for split as well.

Figure 12: Spacing ($-10m$) and velocity ($6m/s$) steps. A car changes lane in front of platoon $i$ reducing the safety distance by $10m$ but it is going faster than platoon $i$ by $6m/sec$.

Figure 13: Slow car at the edge of the sensor range. Platoon $i$ does not have any car in its distance sensor for $t < 0$. At $t = 0$, it sees a car $60m$ ahead going at a much lower speed of $5m/sec$.

Figure 14: Distance ($-10m$) and velocity ($-10m/s$) steps. A slow car changes lane in front of platoon $i$. 
Figure 15: Merge. The platoon merges to the one ahead. The leading platoon is decelerating at $0.5 \text{ m/s}^2$ while the maneuver takes place.

Figure 16: Split. The car becomes a leader and splits from the mother platoon. The mother platoon accelerates at $0.5 \text{ m/s}^2$ while the maneuver takes place.

Figure 17: Split from imaginary platoon (part of change lane maneuver)

Figure 18: Merge to imaginary platoon (part of change lane maneuver)