Vehicle System Dynamics, 31 (1999), pp.313–330 0042-3114/99/3204-313$15.00

Optimal Adaptive Cruise Control with Guaranteed String Stability

CHI-YING LIANG[1] and HUEI PENG[2]

SUMMARY

A two-level Adaptive Cruise Control (ACC) synthesis method is presented in this paper. At the upper level, desired vehicle acceleration is computed based on vehicle range and range rate measurement. At the lower (servo) level, an adaptive control algorithm is designed to ensure the vehicle follows the upper level acceleration command accurately. It is shown that the servo-level dynamics can be included in the overall design and string stability can be guaranteed. In other words, the proposed control design produces minimum negative impact on surrounding vehicles. The performance of the proposed ACC algorithm is examined by using a microscopic simulation program—ACCSIM created at the University of Michigan. The architecture and basic functions of ACCSIM are described in this paper. Simulation results under different ACC penetration rate and actuator/engine bandwidth are reported.

1. INTRODUCTION

Adaptive Cruise Control (ACC) systems were proposed as an enhancement to classical cruise control systems for ground vehicle speed regulation. ACC system controls the vehicle speed to follow a driver's set value when no lead vehicle is in sight. When a slower leading vehicle is present, the ACC controlled vehicle will follow the lead vehicle at a safe distance. ACC research first began in the 1960s [1], and has received evergrowing attention in the last decade. Their commercial implementation is not possible until recently with significant progresses in sensors, actuators, and other enabling technologies.

It has been shown that PID or its variations produce satisfactory control results [2]. ACC algorithms of more complex forms have also been proposed and analyzed ([3], [4], [5]). Two large-scale field tests were also performed recently ([8], [9]), with regular drivers driving ACC vehicles on normal highways. Preliminary test results have shown that these simple ACC control algorithms work reliably in real traffic conditions.

[1] Graduate student
[2] Associate Professor, Department of Mechanical Engineering and Applied Mechanics, University of Michigan; E-mail: hpeng@umich.edu

A two-level approach is commonly used for ACC control design. At the higher level, desired force or acceleration is computed based on vehicle range, range rate and other signals. At the lower (servo) level the throttle/brake are manipulated to follow the desired force or acceleration command closely. It should be noted that this two-level design concept is also common in automated highway research [10] although in a slightly different form. This two-level approach will be adopted in this paper as well. A major contribution that differs this paper from previous research, however, is that the effect of ACC on surrounding vehicles will be addressed explicitly. In our two-level design, the upper level controller addresses the performance of other vehicles and the ACC design will have guaranteed string stability.

The "string stability" problem has been studied as early as 1977 [11]. A platoon of vehicles are said to be string stable if, under no other excitations, the range errors decrease as they propagate along the vehicle stream. In mathematical terms, if the transfer function from the range error of a vehicle to that of its following vehicle has a magnitude less than or equal to 1, it is string stability [15]. To achieve string stability with constant inter-vehicle spacing, vehicle-to-vehicle communication was shown to be necessary [12]. Yanakiev and Kanellakopoulos [14] used a simple spring-mass-damper system to demonstrate the idea of string stability and show the string-stability criterion for constant time-headway and variable time-headway policies. Swaroop and Hedrick [15] showed that if the coupling between two vehicles is weak enough, the controlled system is string stable. This conclusion implies that there is a trade-off between traffic-friendliness and vehicle speed regulation performance.

Although the criteria for guaranteed string-stability are well known, no *synthesis* method has been proposed. The choice of the control gains is mostly based on trial-and-error. In this paper we propose an optimal control strategy which can be used to obtain the gains of the upper-level ACC controller. The cost function of this optimization problem consists of penalty for the range and range rate error terms for all the vehicles in the string. Therefore, string stability is guaranteed. This problem was solved by applying a time-space transformation technique proposed by Levin and Athans [1]. In their study, the optimal controller was assumed to use the information of all the vehicles in the string. In other words, they solved a centralized control problem. A decentralized technique was developed later by K. C. Chu [13] for coupled dynamic systems. In this paper, we apply the formulation proposed by Chu to the ACC problem. The result is a decentralized algorithm suitable for ACC implementation. In other words, only local range and range rate signals are needed for the control of individual vehicles. The performance of the controller can be adjusted by changing the penalty coefficients of the cost function, just like standard LQ optimization problems.

In the past, ACC algorithms were evaluated either using a complex vehicle model with only two vehicles (a lead vehicle and a controlled vehicle) or a simplified vehicle model with a string of identical vehicles. In this research, we propose to use a microscopic simulator, which makes it possible to study the effect of the most important disturbance—lane changes. In this simulation program, the vehicle longitudinal dynamics is simple, but a complex lane change behavior is included. A two-lane closed-circuit highway is constructed in which autonomous lane changes will occur. The fact that we are simulating individual vehicles enables us to study safety and traffic flow characteristics more accurately.

The remainder of this paper is organized as follows: string stability analysis of a platoon of vehicles is shown in Section 2. In Section 3, a synthesis method for ACC control algorithm with guaranteed string stability is presented. ACCSIM—a microscopic simulation tool we developed to evaluate various ACC systems is described in Section 4. A servo-level control design is also presented to confirm that the servo-level dynamics can be robustly regulated. Representative simulation results are given in Section 5. Finally, conclusions are drawn in Section 6.

2. STRING STABILITY ANALYSIS OF A CONSTANT-HEADWAY ACC POLICY

Consider a platoon of identical ACC-controlled vehicles running on the highway in a string (see Fig. 1). This platoon is said to be string stable if the transfer function from the range error of a vehicle to that of its following vehicle has a magnitude less than or equal to one. For each vehicle, define

x_i : the position of the ith vehicle
v_i : the velocity of the ith vehicle
u_i : acceleration command of the ith vehicle (from the upper level ACC)
R_i : $\equiv x_{i-1} - x_i$, range signal measured by the ith vehicle

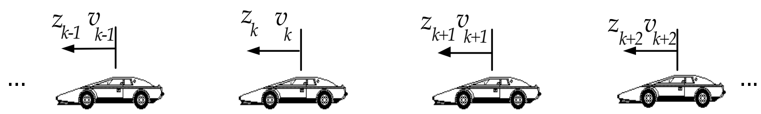


Fig. 1. Schematic diagram of the vehicle string.

We consider a simple proportional control algorithm where the only feedback signals available are range and range rate signals. Under these assumptions that, an ACC algorithm adopts a constant-separation policy (i.e., the desired range is a constant distance), the vehicle platoon cannot be string stable. In the following, we will assume that the constant-headway policy is adopted, the stability of the vehicle string is analyzed.

2.1. Ideal vehicles

For a constant-headway ACC, the control objective is to maintain the same speed as the preceding vehicle and keep the time headway at a constant value where the time headway h_i is defined as $h_i = \frac{R_i}{v_i}$. For ideal vehicles, the acceleration command is followed exactly, i.e., $u_i = \ddot{x}_i$. The control laws for the i[th] and (i-1)[th] vehicles are thus

$$\ddot{x}_{i-1} = K_1 \cdot (x_{i-2} - x_{i-1} - h_d \cdot v_{i-1}) + K_2 \cdot (v_{i-2} - v_{i-1}) \tag{1}$$

$$\ddot{x}_i = K_1 \cdot (x_{i-1} - x_i - h_d \cdot v_i) + K_2 \cdot (v_{i-1} - v_i) \tag{2}$$

From Eqs.(1) and (2), we have

$$\ddot{R}_i + (K_2 + K_1 \cdot h_d) \cdot \dot{R}_i + K_1 \cdot R_i = K_2 \cdot \dot{R}_{i-1} + K_1 \cdot R_{i-1} \tag{3}$$

Therefore,

$$\frac{R_i(s)}{R_{i-1}(s)} = \frac{K_2 \cdot s + K_1}{s^2 + (K_2 + K_1 \cdot h_d) \cdot s + K_1} \tag{4}$$

String stability can thus be guaranteed if and only if $\left| \frac{R_i(j\omega)}{R_{i-1}(j\omega)} \right| \le 1, \forall \omega$, which can be satisfied if and only if $K_2 > \frac{2 - k_1 h_d^2}{2 h_d}$. This implies that the feedback gain K_2 should be large enough to ensure string stability. The time headway also plays an important role. When K_2 is limited in magnitude by practical reasons (sensor noise, etc.), a large time headway should be used.

2.2 Vehicles with servo-loop dynamics

When the servo-loop is described by a first-order lag system $\dot{a}_k = \frac{1}{\tau_a}(u_k - a_k)$, the control laws for two consecutive vehicles become

$$\dddot{x}_{k-1} = \frac{1}{\tau_a}\left[K_1 \cdot (x_{k-2} - x_{k-1} - h_d \dot{x}_{k-1}) + K_2 \cdot (\dot{x}_{k-2} - \dot{x}_{k-1}) - \ddot{x}_{k-1}\right] \tag{5}$$

$$\dddot{x}_k = \frac{1}{\tau_a}\left[K_1 \cdot (x_{k-1} - x_k - h_d \dot{x}_k) + K_2 \cdot (\dot{x}_{k-1} - \dot{x}_k) - \ddot{x}_k\right] \tag{6}$$

Therefore,

$$\left| \frac{R_k(s)}{R_{k-1}(s)} \right| = \left| \frac{K_2 \cdot s + K_1}{\tau_a \cdot s^3 + s^2 + (K_2 + K_1 \cdot h_d) \cdot s + K_1} \right| \tag{7}$$

For string stability, K_1 and K_2 have to satisfy one of the following two constraints:

$$\begin{cases} K_2 + h_d K_1 \le \frac{1}{2\tau_a} \\ 2h_d K_2 + h_d^2 K_1 > 2 \end{cases} \quad \text{or} \quad \begin{cases} K_2 + h_d K_1 \ge \frac{1}{2\tau_a} \\ (K_2 - \frac{1}{2\tau_a})^2 < (\frac{h_d}{\tau_a} - 2)K_1 \end{cases} \tag{8}$$

3. OPTIMAL ACC ALGORITHM WITH GUARANTEED STRING STABILITY

Assuming that the dynamics of a single vehicle is described by

$$\dot{x}_k = A_k x_k + B_k u_k \tag{9}$$

where x_k denotes the state vector of the k^{th} vehicle (not just the travel distance). The feedback signals depend on the states of more than one vehicle and thus in general could have the form

$$y_k = \sum_{j=-\infty}^{\infty} G_{k-j} x_j \tag{10}$$

For simple proportional control based on range and range rate signals, the control law becomes

$$u_k = K_1 \cdot (x_{k-1} - x_k - h_d v_k) + K_2 \cdot (v_{k-1} - v_k) \equiv K \cdot y_k \tag{11}$$

An optimal control framework can then be formulated to minimize the range and range rate errors for all the vehicles in the string. The performance index is defined as:

$$J = \frac{1}{2} \int_0^{\infty} \sum_{k=-\infty}^{\infty} [q_1 (x_{k-1} - x_k - h_d v_k)^2 + q_2 (v_{k-1} - v_k)^2 + r u_k^2] dt \tag{12}$$

where the coefficients q_1, q_2 and r can be selected for performance/effort trade-off. Eq.(12) can be rewritten as

$$J = \frac{1}{2} \int_0^{\infty} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} (\underline{x}_j^T Q_{j-k} \underline{x}_k + u_j^T R_{j-k} u_k) \cdot dt \tag{13}$$

The major technical challenge is then to compute the control gains (K_1 and K_2) given the weighting coefficients q_1, q_2 and r.

3.1. Optimal ACC algorithm

To solve the optimization problem described above, a bilateral z transformation technique [13] is applied. This transformation is defined as follows:

$$Z\{A_k\} = \sum_{k=-\infty}^{\infty} A_k z^{-k} = \mathrm{A}(z) \tag{14}$$

where z is a "space advance" operator and corresponds to moving upstream in the vehicle string (see Fig. 1). An inverse z transformation is also defined:

$$A_k = \langle \mathrm{A}(z) \rangle_k = \frac{1}{2\pi i} \int \mathrm{A}(z) z^{k-1} dz \tag{15}$$

where the path of integration is the unit circle in the complex plane. Notice that the inverse z-transformation will be denoted as $\langle \cdot \rangle_k$ in the remainder of this paper. Converting Eqs.(9)-(11) and (13) into the z domain, we have

$$\dot{x}(z) = A(z)x(z) + B(z)u(z) \tag{16}$$

$$u(z) = Ky(z) \tag{17}$$

$$y(z) = G(z)x(z) \tag{18}$$

$$J = \frac{1}{2} \int_0^\infty \left\langle x(z^{-1})^T Q(z)x(z) + u(z^{-1})^T R(z)u(z) \right\rangle_0 dt \tag{19}$$

From Eqs.(16)-(18), we have $\dot{x}(z) = [A(z) + B(z)KG(z)]x(z)$. Define $D(z,K)=A(z)+B(z) KG(z)$, $x(z)$ is then given by its initial condition $x_0(z) \equiv x(z)\big|_{t=0}$ from

$$x(z) = \exp(D(z,K) \cdot t) \cdot x_0(z)$$

The performance index (19) can then be written as

$$J = \frac{1}{2} \int_0^\infty \left\langle r(\exp(D(z^{-1},K)^T \cdot t) M(z,K) \exp(D(z,K) \cdot t) x_0(z) x_0(z^{-1})^T) \right\rangle_0 \cdot dt \tag{21}$$

where $M(z,K) = Q(z) + G(z^{-1})^T K^T R(z) KG(z)$. Assuming that the initial states of the vehicles are randomly distributed, an initial-state independent optimization problem can then be defined which minimizes a slightly different cost function:

$$\hat{J} = \left\langle \frac{1}{2} tr \int_0^\infty \exp(D(z^{-1},K)^T \cdot t) M(z,K) \exp(D(z,K) \cdot t) \cdot dt \right\rangle_0 \tag{22}$$

which is satisfied by the optimal gain $K = K^*$ only if we have

$$\frac{\partial \hat{J}}{\partial K^*} = \left[\frac{\partial \hat{J}}{\partial K_1^*} \quad \frac{\partial \hat{J}}{\partial K_2^*} \right]^T = 0.$$

Eq.(22) can be written as

$$\hat{J} = \left\langle \tfrac{1}{2} tr(M(z,K)\cdot L(z,K)) \right\rangle_0 = \left\langle \tfrac{1}{2} tr(P(z,K)) \right\rangle_0$$

where two extra matrices are defined:

$$P(z,K) = \int_0^\infty \exp(D(z^{-1},K)^T \cdot t)\cdot M(z,K)\cdot \exp(D(z,K)\cdot t)\cdot dt$$

and

$$L(z,K) = \int_0^\infty \exp(D(z,K)\cdot t)\cdot \exp(D(z^{-1},K)^T \cdot t)\cdot dt \cdot$$

The necessary condition for optimization is then

$$\frac{\partial \hat{J}}{\partial K} = \left\langle \tfrac{1}{2} tr(\frac{\partial M(z,K)}{\partial K}\cdot L(z,K)) \right\rangle_0 + \left\langle \tfrac{1}{2} tr(M(z,K)\cdot \frac{\partial L(z,K)}{\partial K}) \right\rangle_0 = 0 ,$$

or

$$\begin{aligned} &\left\langle R(z)KG(z)L(z,K)G(z^{-1})^T + R(z)^T KG(z^{-1})L(z,K)^T G(z)^T \right\rangle_0 + \\ &\left\langle B(z^{-1})^T P(z,K)L(z,K)G(z^{-1})^T + B(z)^T P(z,K)^T L(z,K)^T G(z)^T \right\rangle_0 = 0 \end{aligned} \tag{23}$$

Therefore,

$$\begin{aligned} K = -\tfrac{1}{r} &\left\langle B(z^{-1})^T P(z,K)L(z,K)G(z^{-1})^T + B(z)^T P(z,K)^T L(z,K)^T G(z)^T \right\rangle_0 \\ &\cdot \left\langle G(z)L(z,K)G(z^{-1})^T + G(z^{-1})L(z,K)^T G(z)^T \right\rangle_0^{-1} \end{aligned} \tag{24}$$

where the matrices $P(z,K)$ and $L(z,K)$ are solved from the following algebraic equations:

$$P(z,K)D(z,K) + D(z^{-1},K)^T P(z,K) + M(z,K) = 0 \tag{25}$$

$$L(z,K)D(z^{-1},K)^T + D(z,K)L(z,K) + I_n = 0 \tag{26}$$

The optimal gain can be calculated numerically from Eqs.(23)-(26). Although the string stability condition is not explicitly addressed in the proposed control formulation, it is indirectly guaranteed in the cost function formulation. In the following theorem we will prove that when the optimal control gains K^* exist, they will guarantee string stability.

Theorem 1 For ideal servo-loop dynamics, i.e., $A(z)=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$, $B(z)=\begin{bmatrix}0\\ 1\end{bmatrix}$, $K(z)=[K_1 \quad K_2]$ and $G(z)=\begin{bmatrix}z^{-1}-1 & -h_d\\ 0 & z^{-1}-1\end{bmatrix}$, the optimal control parameters K^* which minimize the performance index $\hat{J}$ in Eq.(22) will result in a string stable ACC design.

Proof: The state $x(z)$ is dependent on its initial condition according to $x(z)=\exp(D(z,K)\cdot t)\cdot x_0(z)$. For any bounded initial condition $x_0(z)$, the necessary and sufficient condition for $x(z)$ to remain bounded is that the matrix $[\exp(D(z,K)\cdot t)]$ should be bounded for all t and that it approaches zero as t $\to \infty$ for all $|z|$ = 1. In other words, all the eigen-values of $D(z,K)$ should have negative real parts for all points on the unit circle $|z|$ = 1. Since $D(z,K)=A(z)+B(z)KG(z)=\begin{bmatrix}0 & 1\\ K_1(z^{-1}-1) & -K_1\cdot h_d+K_2(z^{-1}-1)\end{bmatrix}$, the characteristic equa- tion of $D(z,K)$ is $\lambda^2+(K_1\cdot h_d-K_2(z^{-1}-1))\lambda-K_1(z^{-1}-1)=0$. It can be seen that the necessary and sufficient condition for λ to have negative real part for all points $|z|$ = 1 is $K_2>\frac{2-K_1h_d^2}{2h_d}$. This condition is exactly the same as the string stability condition shown in Section 2.1. In other words, when $K_2>\frac{2-K_1h_d^2}{2h_d}$, $[\exp(D(z,K)\cdot t)]$ is bounded, $\hat{J}$ is thus bounded. Therefore, the optimal control parameters K^* which minimize the performance index $\hat{J}$ will produce a string stable design.

3.2. Ideal vehicles

When the servo-loop is perfect, i.e., the vehicle acceleration follows the acceleration command from the upper level exactly, we have $A(z)=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$, $B(z)=\begin{bmatrix}0\\ 1\end{bmatrix}$, $K(z)=[K_1 \quad K_2]$, $Q_{-1}=\begin{bmatrix}-q_1 & -q_1h_d\\ 0 & -q_2\end{bmatrix}$, $Q_0=\begin{bmatrix}2q_1 & q_1h_d\\ q_1h_d & q_1h_d^2+2q_2\end{bmatrix}$, $Q_1=\begin{bmatrix}-q_1 & 0\\ -q_1h_d & -q_2\end{bmatrix}$, $Q_k=0_{2x2}$, $\forall|k|>1$, $R_0=r$, and $R_k=0$, $\forall k\neq 0$. Example optimal ACC design results are shown below. Figure 2 shows the contour plot of the performance index $\hat{J}$ for $q_1=q_2=r=1$ and $h_d=1.4$. The solid line at the lower left corner shows the string stability criterion $K_2>\frac{2-K_1\cdot h_d^2}{2h_d}$. Figure 3 shows the optimal control gains for a set of 36 different q_1, q_2 combinations on the $K_1 - K_2$ plane. The optimal algorithm always produce string-stable control gains as the Theorem has predicted.

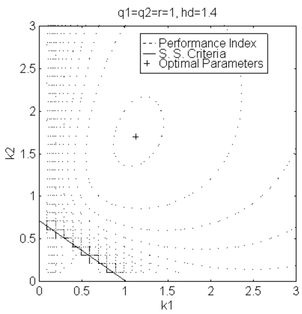


Fig. 2. Contour plot of the performance Index (ideal vehicles).

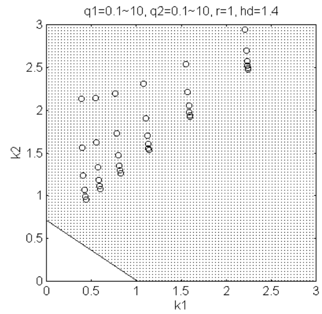


Fig. 3. Optimal control parameters plot (ideal vehicles).

3.3. Vehicles with servo-loop dynamics

When the servo-loop is imperfect, the optimal ACC design needs to take the servo-loop dynamics into consideration. We consider the case when the servo-loop dynamics can be approximated by a first-order lag, i.e., $\frac{a_k}{a_{kd}} = \frac{1}{1+\tau_a s}$ where a_k is the vehicle acceleration and a_{kd} is the acceleration command from the upper-level. In this case, three state variables need to be defined for each vehicle, i.e., $x_k = [x_k \quad v_k \quad a_k]^T$. The state and input matrices are $A_k = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{-1}{\tau_a} \end{bmatrix}$ and $B_k = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau_a} \end{bmatrix}$. The "Q" matrices of the performance index need to be inflated in size (from 2x2 to 3x3) by adding zero elements. The optimization problem is otherwise identical to the ideal-vehicle case. Example optimal ACC design results are shown below. The time constant τ_a of the sub-loop dynamics is assumed to be 0.2 seconds.

Figure 4 shows the contour plot of the performance index $\hat{J}$ for $q_1 = q_2 = r = 1$ and $h_d = 1.4$. The solid line at the lower left and upper left corners show the string stability criterion shown in Eq.(8). Figure 5 shows the optimal control gains for a set of 49 different q_1, q_2 combinations on the $K_1 - K_2$ plane. Again, the optimal algorithm always produces string-stable control gains. Under similar penalty weighting, it can be noticed that the control gains are always smaller than those of the ideal-vehicle cases.

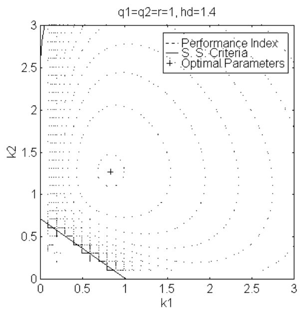


Fig. 4. Contour plot of the performance index (with sub-loop dynamics).

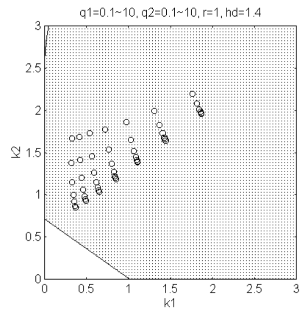


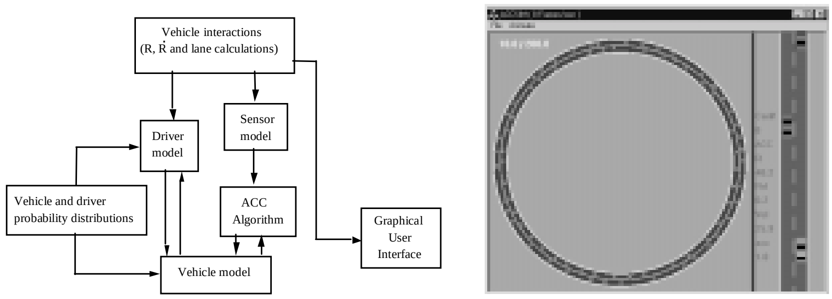
Fig. 5. Optimal control parameters plot (with sub-loop dynamics).

4. UOFM ACC SIMULATOR

A PC based simulation program ACCSIM (Adaptive Cruise Control SIMulator) has been developed at the University of Michigan to study the effect of ACC on the host vehicle as well as its surrounding vehicles. This is achieved by constructing a simulator with multiple vehicles. Variables such as position, speed, acceleration and lane locations of all the vehicles are simulated. The effect of ACC on safety, fuel consumption and throughput thus can all be investigated at the vehicle levels. The simulated highway is assumed to be a two-lane highway of 2km long with no on-off ramps.

The highway is closed-loop in nature so that if the ACC algorithm results in string instability, its effect will be amplified and become more visible.

The simulation program is divided into four parts: driver model, vehicle/sensor model, ACC model, and the interaction model (see Fig. 6). The driver model consists of a lane change model (for all vehicles) and a car-following model (only for manually controlled vehicles). The parameters of the driver model are obtained from field test data [6]. Details of this model can be found in [16]. The vehicles/drivers are initialized at the beginning of the simulation with a set of attribute parameters such as control type (manual vs. ACC), acceleration level, desired speed, desired headway, etc. When the vehicle is controlled by ACC, the ACC algorithm described in the previous section is used. The interaction model calculates the displacement, speed and lane locations of all vehicles, and consists of a collection of algebraic and integration routines. In the following, the vehicle dynamic model is described in details.



(a) (b)

Fig. 6. ACCSIM--(a): architecture, (b): screen shot.

4.1. Vehicle model

The longitudinal vehicle model is adopted from a work by Sommerville [17]. Figure 7 shows the Simulink block diagram of the vehicle model. Major sub-models are described in details below.

Engine model

A static look-up table is used to relate the engine states (throttle angle and engine speed) with its combustion torque (in N-m). Linear interpolation is used between data points. This engine map was obtained experimentally and the results were reported in [17]. After the engine output torque is obtained, the engine speed is calculated from

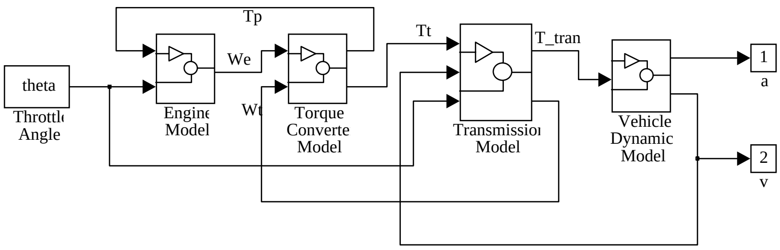


Fig. 7. Block diagram of the vehicle model.

$$\dot{\omega}_e = \frac{1}{J_e}\left(T_e(\omega_e, \theta_{th}) - T_a(\omega_e) - T_p(\omega_e, \omega_t)\right) \tag{27}$$

where

θ_{th} = throttle angle,
ω_e = engine speed,
ω_t = torque converter turbine speed,

T_p = load torque from torque converter pump (to be described below)
J_e = engine inertia,
T_e = engine combustion torque,
T_a = engine accessory and load torque,

Torque converter model
The torque converter provides smooth coupling between the engine and the transmission. The pump side of the torque converter is connected to the engine and the turbine side is connected to the transmission. The behavior of the torque converter is modeled by a set of quadratic equations relating ω_t (turbine speed) and ω_p (pump speed) to turbine and pump torque.

$$\begin{aligned} T_p &= m_1 {\omega_p}^2 + m_2 \omega_t \omega_p + m_3 {\omega_t}^2 \\ T_t &= n_1 {\omega_p}^2 + n_2 \omega_t \omega_p + n_3 {\omega_t}^2 \end{aligned} \tag{28}$$

The torque converter can operate in one of three modes: converter mode, coupling mode and overrun mode. The mode and values of the parameters (m_1, m_2, m_3, n_1, n_2, and n_3) are determined by the speed ratio $SR = \frac{\omega_t}{\omega_p}$:

$SR < 0.842$: Converter mode
$0.842 \le SR \le 1.0$: Coupling mode
$SR > 1.0$: Overrun mode

Transmission model
The transmission is responsible for transferring engine torque to the driven wheels. Its gear ratio is determined by the vehicle speed and engine/load state. The gearshift in this model is based on symmetric linear shift lines described below:

1 ↔ 2 shift, $\omega_T = 0.77 \cdot \theta + 89.2$
2 ↔ 3 shift, $\omega_T = 2.77 \cdot \theta + 163.1$
3 → 4 shift, $\omega_T = 4.62 \cdot \theta + 224.8$
4 → 3 shift, $\omega_T = 7.85 \cdot \theta + 91.6$

where ω_T is the transmission output speed (rad/sec) and θ is the throttle angle (in degrees).

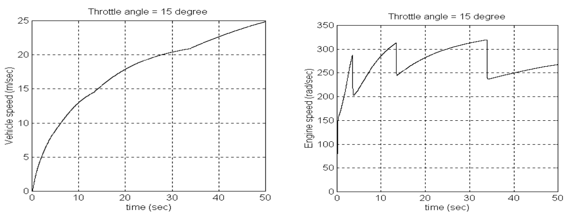
Vehicle dynamics model
Vehicle speed is calculated from the following equation based on Newton's second law.

$$\dot{v} = \frac{1}{rM_v}(G_T \cdot T_t(\omega_e, \omega_t) - r\gamma v^2 - rF_{rr} - T_b - T_g) \tag{29}$$

where

v = vehicle speed
r = tire radius
M_v = vehicle mass
G_T = overall gear ratio
γ = aerodynamic drag coefficient
T_T = torque from transmission
F_{rr} = rolling resistance force
T_b = braking torque
T_g = torque due to road inclination

Simulation results of a step-throttle acceleration run (with all other disturbance inputs set to zero) are shown in Figure 8.



(a) Vehicle velocity response (b) Engine speed response

Fig. 8. Simulation results of the vehicle response under 15 degree throttle angle.

4.2. Servo-loop control design

As shown in section 3, the higher-level ACC design considers the behavior between vehicles. A servo-level controller needs to be designed so that the acceleration command is followed closely. Ideally, the servo-loop controller should be designed so that its behavior is not only fast but also consistent. The vehicle model shown in the previous section is used for final simulations. It is, however, too complicated for control design purposes. A simpler model needs to be constructed. It is a well-known fact that the relationship between throttle angle and vehicle speed can be approximated by $\frac{V}{\theta_{th}} = \frac{K_v}{s + 1/\tau}$ where K_v and τ vary with vehicle speed V. Based on this simplified model, an adaptive PI controller is implemented to compensate for the variation of the vehicle parameters and to achieve steady-state tracking error. The block diagram of the control sub-loop is shown in Figure 9.

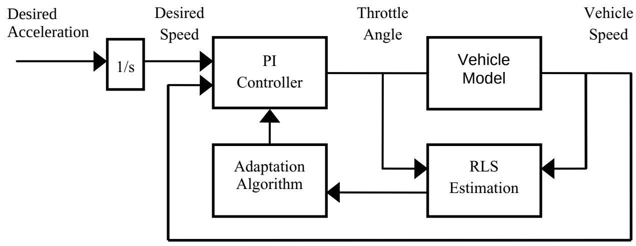


Fig. 9. Block diagram of the servo-loop system.

The objective of the servo-loop control design ($\dot{a} = \frac{1}{\tau_a}(a_d - a)$) is equivalent to have $\dot{V} = \frac{1}{\tau_a}(V_d - V)$. This goal is achieved by an adaptive PI control algorithm. The vehicle parameters K_v and τ are estimated and the PI control gains are updated accordingly. A standard Recursive Least-Squares (RLS) estimation method is used to estimated the parameters K_v and τ. Because the vehicle parameters are time-varying, the RLS method with a forgetting factor is used.

According to the string stability analysis results presented in section 2.2, for an ACC system with a desired time headway of 1.4 sec, the servo-loop time constant τ_a must be less than 0.7 sec for string stability. We designed the servo-loop controller so that τ_a = 0.2 sec. The servo-loop system shown in Figure 9 is in fact a second-order system. We place a pole at -1/τ_a (= -5.0) and the other (faster) pole at -10/τ_a so that the overall response is dominated by the slower pole -1/τ_a. The RLS adaptive algorithm is implemented in discrete time with a sampling rate of 10 msec.

The simulation results of the sub-loop controller tracking step changes of acceleration command with magnitude of 0.1, 0.3, 0.5 and 0.7 m/s^2 at four different vehicle speeds are shown in Figure 10. It can be seen that the vehicle speed responses are all very close to the desired (first-order) form and the time constant for the 16 cases are all between 0.19 sec and 0.23 sec. In other words, the sub-loop controller not only achieves the desired dynamics, its behavior is consistent across a wide range of operating conditions.

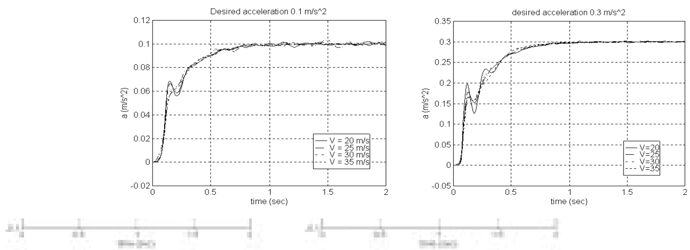


Fig. 10. Simulation results of the sub-loop controller under step response.

5. SIMULATION RESULTS

The purpose of this simulation study is to investigate the performance of ACC control algorithm under various penetration rate, and influence dynamics of the servo-loop on overall ACC performance.

The simulation scenarios are as follows: on the 2km 2-lane test track, 50 vehicles are created. Six runs were performed with different ACC penetration rate: 0%, 10%, 30%, 50%, 70% and 90%. The status of all the vehicles are recorded for a span of 2 hours. The vehicle desired speed, desired time headway, intention factor and safety factor are initialized according to the statistical distribution described in a previous paper [16]. Due to frequent lane changing, transient behavior (i.e., not platooning) of the ACC system becomes important. Figure 11 shows the root-mean-square (RMS) values of the range error, range rate error, and acceleration signal of manual vehicles vs. ACC vehicles. It can be seen that ACC vehicles achieve marked improvement in range and range rate with a significantly reduced acceleration level.

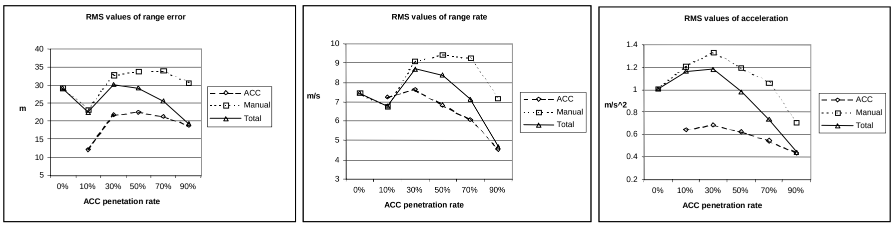


Fig. 11. Simulation results of ACC vs. manual vehicles.

Figure 12 shows the cummulative percentage of Time-To-Collision (TTC) under three ACC penetration rate: 0%, 50% and 90%. Under normal car-following conditions, TTC should approach infinity. The percentage of time the vehicles on the highway having a TTC less than, say, 5 seconds is a good index of safety. It can be seen that when most of the vehicles are controlled by ACC, the possibility of having TTC<5 seconds is greatly reduced.

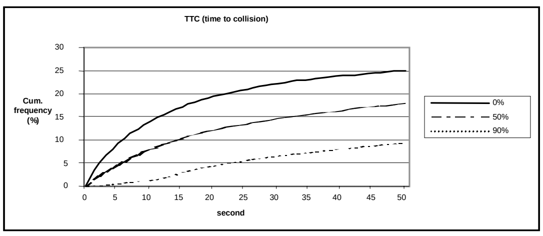


Fig. 12. Time-To-Collision of all vehicles.

We subsequently study the effect of engin/actuator lag on the performance of ACC. Three cases are simulated:

(a) Ideal vehicles/original control. In this case, the servo-level dynamics are assumed to be ideal, and the optimal control ACC results in a control gain of K_1 = 1.12 and K_2 = 1.70.

(b) Real vehicles/original control. In the simulations, the servo-level dynamics are introduced. The ACC algorithm, however, still uses the original control gains K_1 = 1.12 and K_2 = 1.70.

(c) Real vehicles/new control. In this case, the ACC control design considers the servo-level dynamics, and thus results in a new control gain (uses techniques shown in Section 3.3).

The servo-level dynamics is assumed to be described by $\frac{1}{1+0.2s}$ (which is achievable by an adaptive control algorithm, see Section 4.2), the RMS values of the range, range rate and acceleration signals are as shown in Table 1. It can be seen that

Table 1. Simulation results of three cases under servo-level dynamics $\frac{a}{a_d} = \frac{1}{1+0.2s}$ (50 vehicles/ 2km/ 2 lanes, 50% penetration rate).

	Case (a) Ideal vehicles Original control	Case (b) Real vehicles Original control	Case (c) Real vehicles New control
RMS of Range Error (m)	21.22	32.11	22.29
RMS of Range Error (m/s)	3.65	6.07	3.83
RMS of Accel. (m/s^2)	0.74	0.83	0.77

if servo-level dynamics exist but are ignored in the ACC design (case (b)), the performance may deteriorate significantly. When the ACC design considers servo-level dynamics (case (c), K_1 = 0.83 and K_2 = 1.26), the performance becomes quite similar to that of the ideal case (case (a)).

6. CONCLUSIONS

In this paper, a control synthesis method was presented for the design of Adaptive Cruise Control algorithms. This synthesis method is based on optimal control theory applied to a space-transformed vehicle string. Due to the fact that the control signal optimizes the range and range rate errors of all the vehicles in the string, string stability is guaranteed. The resulted upper-level ACC control algorithm generates desired acceleration command, which is then followed by a servo-level controller. Simulation were performed on a microscopic simulator which mimics a two-lane highway. It is shown that the servo-level dynamics need to be considered in the design of upper-level ACC to achieve optimal performance.

ACKNOWLEDGMENT

This work is supported by U.S. Department of Transportation under the contract DTFH61-94-C-00017-003 through the ITS Research Center of Excellence at the University of Michigan.

REFERENCES

1. Levine, W. and Athans, M., "On the optimal error regulation of a string of moving vehicles", *IEEE Trans. Automat. Contr., vol. 11, no. 3*, July, 1966, pp. 355-361
2. Ioannou, P., Xu, Z., Eckert, S. and Sieja, T., "Intelligent cruise control: theory and experiment", *Proc. of the 32nd Conference on Decision and Control*, San Antonio, December, 1993, pp. 1885-1890.
3. Ioannou, P. et al., "Intelligent Cruise Control: Theory And Experiment," *Proceedings of the 32nd IEEE Conference on Decision and Control*, San Antonio, TX, 1993, IEEE 93CH3307-6. 1996, pp.1885-1890.
4. Hatipoglu, C., Ozguner, U., and Sommerville, M., "Longitudinal Headway Control Of Autonomous Vehicles," *Proceedings of the 1996 IEEE Int. Conf. on Control Appl., IEEE 96CH35854*. 1996, pp.721-726.
5. Fancher, P., Peng, H. and Bareket, Z., "Comparison of Three Control Algorithms on Headway Control for Heavy Trucks," *Proceedings of the 1995 IAVSD Symposium*, Ann Arbor, MI, 1995a.
6. Fancher, P.S., Bareket, Z., Sayer, J.R., Johnson, G.E., Ervin, R.D., Mefford, M.L., *Fostering Development, Evaluation, and Deployment of Forward Crash Avoidance Systems (FOCAS*), ARR-5-15-95, University of Michigan Transportation Research Institute technical Report. 1995b.
7. Fancher, P.S., Bareket, Z., Sayer, J.R., MacAdam, C., Ervin, R.D., Mefford, M.L., Haugen, J., *Fostering Development, Evaluation, and Deployment of Forward Crash Avoidance Systems (FOCAS)*, ARR-12-15-96, University of Michigan Transportation Research Institute technical Report number 96-44. 1996,
8. Fancher, P. et al., "Tests Characterizing Performance of an Adaptive Cruise Control System," *Proceedings of the 1997 SAE International Congress and Exposition, Intelligent Transportation Systems, SAE SP-1230*, pp.155-164, SAE paper# 970458, 1997a.
9. Fancher, P. et al., "Intelligent Cruise Control Field Operational Test (Interim Report)," *Technical report, Dept. of Trans.*, DOT HS 808 622, Aug, 1997b.
10. McMahon, D.H., Narendran, V.K., Swaroop, D., Hedrick, J.K., Chang, K. S. and Devlin, P. E., "Longitudinal vehicle controllers for IVHS: Theory and experiment", *Proc. of the 1992 American Control Conference*, Chicago, 1992, pp. 1753-1757.
11. Caudill, R. J. and Garrard, W. L., "Vehicle-follower longitudinal control for automated transit vehicles", *J. of Dynamic Systems, Measurement, and Control*, December, 1977, pp. 241-248.
12. Sheikholeslam, S. and Desoer, C. A., "Longitudinal control of a platoon of vehicles", *Proc. of 1990 American Control Conference*, San Diego, 1990, pp. 291-296.
13. Chu, K. C., "Optimal decentralized regulation for a string of coupled systems", *IEEE Trans. Automat. Contr., vol. 19, no. 3*, June, 1974, 1974, pp. 243-246.
14. YanaKiev, D., Kanellakopoulos, I., "A Simplified Framework for String Stability Analysis in AHS," *Proc. of the 13th IFAC World Congress*, *Volume Q*, 1996, pp.177-182.

15. Swaroop, D., Hedrick, J.K., "String Stability of Interconnected Systems," *IEEE Trans. Automat. Contr., vol. 41, no. 3*, March, 1996, 1996, pp. 349-356.
16. Liang, C. and Peng, H., "Design and Simulations of A Traffic-Friendly Adaptive Cruise Control Algorithm," *Proceedings of the 1998 International Mechanical Engineering Congress and Exposition*.
17. Sommerville, M., "Implementation of a Longitudinal Controller for Use on an Automated Highway System", *Technical Report 96-03*, Center for Intelligent Transportation Research, the Ohio State University, Spring 1996.