Model Predictive Control-Based Trajectory Shaper for Safe and Efficient Adaptive Cruise Control

Anye Zhou
Oak Ridge National Laboratory
Oak Ridge, TN, USA
zhoua@ornl.gov

Zejiang Wang
Oak Ridge National Laboratory
Oak Ridge, TN, USA
wangz2@ornl.gov

Adian Cook
Oak Ridge National Laboratory
Oak Ridge, TN, USA
cookas@ornl.gov

***Abstract*—Recent studies show that commercially-available adaptive cruise control (ACC) systems are string-unstable, indicating that ACC-driven vehicles amplify speed fluctuations from downstream traffic and induce stop-and-go waves. Moreover, it is challenging to revise the original control algorithm of an ACC system to achieve string stability due to its internal complexity and powertrain uncertainties. To achieve desired control performance given a string-unstable ACC system and circumvent revising the original control algorithm, this study proposes a model predictive control-based trajectory shaper (MPC-TS), which only modifies the sensor-measured trajectory information (i.e., position and speed) of the preceding vehicle. The proposed MPC-TS leverages the input shaping technique to generate reference trajectory to improve string stability, while incorporating tracking errors and vehicle acceleration/deceleration magnitude in the MPC cost function and constraining fluctuations of vehicle speed and spacing to ensure desired car-following performance. Numerical experiments validate the control performance of ACC with the proposed MPC-TS in terms of string stability, safety, traffic efficiency, and comfort.**

***Index Terms*—adaptive cruise control, string stability, safety, input shaping, model predictive control**

I. Introduction

The rapid advances in vehicle automation and sensing technologies enable adaptive cruise control (ACC) systems to be widely equipped on commercially-available vehicles. An ACC system consists of two main components: a high-level planner (HP) and a low-level controller (LC). The HP uses sensor-measured trajectory information (i.e., position and speed) of the preceding vehicle to compute desired ego vehicle trajectory (e.g., future speed/acceleration). Next, the LC computes a proper gas/brake command and delivers it to the vehicle powertrain for execution. ACC systems can effectively alleviate driving fatigue and improve the experience of commuting by automatically realizing car-following (CF) and cruising functionalities. However, recent studies have shown that the current commercially-available ACC systems are string-unstable [1]–[3], indicating that vehicles driven by ACCs will amplify speed fluctuations from downstream traffic and induce undesired stop-and-go waves.

To mitigate string instability, the most direct and intuitive approach is to fine-tune or modify the control algorithms of an ACC system. Several studies have developed HPs to compute desired trajectories. For instance, proportional-derivative (PD) type linear CF controller and its variants have been extensively studied to ensure string stability and desired CF performance [4]–[6]. Model predictive control (MPC) type planners have also been widely studied to achieve optimal trajectory with operational constraints (e.g., speed limit, minimum safe spacing, and actuator saturation) [7]–[9]. Numerous methods for LCs have also been proposed to execute the planned trajectory. For example, proportional-integral-derivative control [10], feedback linearization control [11], and loop shaping control [12] have been developed to minimize the tracking error between the actual and planned trajectories. However, directly modifying the control algorithms of an ACC system to achieve string stability suffers from the following challenges. Firstly, the performance of LC can significantly influence string stability [10], which has been neglected in HP design and stability analysis. However, the interaction between HP and LC is complicated and difficult to analyze because it involves: (i) numerous parameters, (ii) powertrain uncertainties rooted from real-world operations, and (iii) unknown controller dynamics due to commercial propriety. These factors hinder rigorous analysis considering HP, LC, and vehicle powertrain dynamics to achieve practical string stability, entailing laborious tuning works based on trial-and-error. Second, there is a trend of switching to learning-based control methods from rule-based control methods in the industry [13]. The complexity and intractability of deep neural networks used in deep (reinforcement) learning can substantially increase the effort and cost to tune and revise ACC control algorithms to achieve string stability.

To circumvent directly modifying the existing control algorithms of ACC systems and achieve string stability, this study develops a trajectory shaper (TS)-based method. The TS-based method only modifies the sensor-measured trajectory information of the preceding vehicle before it is used by the ACC control algorithm. Specifically, the TS-based method is inspired by the input shaping technique used to alleviate residual vibrations in mechanical systems [14]–[16]. Func-

tioning as an input signal filter, the TS-based method shapes the trajectory information of the preceding vehicle based on the estimated natural frequency and damping ratio of an ACC system, such that a string-unstable ACC system can behavior in a string-stable manner. This study first introduces a vanilla trajectory shaper (VTS) by adapting the zero-vibration shaper (ZVS) technique [14] to ACC systems. However, the string stability achieved by VTS may introduce delay to vehicle trajectories and sacrifices other CF performance metrics (i.e., safety and traffic efficiency). Thus, to balance different CF performance metrics, this study further proposes an MPC-based trajectory shaper (MPC-TS). The MPC-TS aims to track the trajectory generated by VTS in the prediction horizon to enhance string stability while incorporating: (i) objectives including comfort and maintaining desired time headway and identical speed with the preceding vehicle; and (ii) safety, efficiency, and feasibility-related constraints on vehicle states to achieve a well-balanced CF performance. The emphasis on specific control performance is regulated by the corresponding weighting coefficient in the cost function. Remarkably, the formulation of MPC-TS enables a framework to alter the sensor measurement based on the different CF performance metrics rather than merely focusing on string stability. Numerical experiments show that the proposed MPC-TS outperforms VTS in terms of safety and traffic efficiency (i.e., maintaining small time headway to improve roadway capacity), while maintaining an almost identical string stability performance.

The contributions of this study are two-fold. First, an alternative approach to achieve string stability is proposed, which can save costs and efforts in tuning/revising complicated ACC control algorithms. Second, the proposed MPC-TS provides a general framework to balance multiple CF performance metrics, such that ensuring string stability will not sacrifice other performance metrics during ACC operations.

The remainder of this study is as follows. Section II introduces the input shaping technique and VTS. Section III formulates the MPC-TS. Section IV provides numerical experiments to validate the proposed approach. Section V concludes this study and points out future research directions.

II. Vanilla Trajectory Shaper

This section introduces the VTS based on the input shaping technique to alleviate the string instability of ACC systems.

A. Input Shaping

The input shaping technique seeks to alleviate the residual vibrations in mechanical systems via shaping the input signal (e.g., reference trajectory to track) [14], [15]. The dynamics of vibrations can be approximated as a second-order dynamical system formulated as the following transfer function [16]:

$$\frac{Y(s)}{U(s)} = \frac{\kappa\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{1}$$

where s is the Laplace operator, $Y(s)$ and $U(s)$ are the output and input of the system (e.g., $Y(s)$ can be the position of the mass of a mass-spring-damper system, and $U(s)$ is the input force applied on the spring and damper), respectively. κ is the static gain which scales the magnitude of the input signal, ω_0 is the natural frequency which describes the oscillation period and speed of the system, ζ is the damping ratio which reflects the capability of alleviating oscillations.

The core idea of input shaping is to apply an impulse sequence $\{(t_j, M_j), j = 1, ..., N\}$ to shape the original input signal U via convolution, such that the shaped input signal will not excite any vibrations from the dynamical system. M_j and t_j are the magnitude and time instance of jth impulse, respectively. Correspondingly, in the Laplace domain, the shaped input can be expressed as:

$$U_{\text{shaped}}(s) = \sum_{j=1}^{N} M_j U(s) e^{-t_j s} \tag{2}$$

Specifically, to design the impulse sequence, a performance metric called residual vibration percentage is defined as:

$$V(\omega_0, \zeta) = e^{-\omega_0\zeta t_N}\sqrt{S(\omega_0,\zeta)^2 + C(\omega_0,\zeta)^2} \tag{3}$$

where $S(\omega_0,\zeta) = \sum_{j=1}^{N} M_j e^{\omega_0\zeta t_j}\sin\omega_0\sqrt{1-\zeta^2}t_j$, $C(\omega_0,\zeta) = \sum_{j=1}^{N} M_j e^{\omega_0\zeta t_j}\cos\omega_0\sqrt{1-\zeta^2}t_j$. The residual vibration percentage measures the vibrations after applying N_j impulses to the second-order dynamical system in (1). Enforcing (3) to zero indicates the impulse response of a mechanical system will not exceed the input level, and any overshoot/undershoot will be eliminated. Correspondingly, in the implementation of zero-vibration shaper (ZVS) [14] (the simplest input shaping technique), two impulses are applied to eliminate the vibration (i.e., $V = 0$ and $N = 2$), which can be computed by solving the following equations:

$$S(\omega_0, \zeta) = 0 \tag{4a}$$

$$C(\omega_0, \zeta) = 0 \tag{4b}$$

$$\sum_{j=1}^{N} M_j = 1 \tag{4c}$$

Correspondingly, by setting $t_1 = 0$ as the time instance of initializing ZVS, solving (4) yields the following impulse sequence of ZVS:

$$M_1 = \frac{e^{\zeta\pi/(1-\zeta^2)}}{1 + e^{\zeta\pi/(1-\zeta^2)}} \tag{5a}$$

$$M_2 = 1 - M_1 \tag{5b}$$

$$t_2 = t_1 + \frac{\pi}{\omega_0\sqrt{1-\zeta^2}} \tag{5c}$$

The impulse sequence is then convoluted with the original input signal (following (2)) to achieve zero residual vibrations.

B. VTS for String Stability

The string instability issue of an ACC system is analogous to the residual vibration problem of mechanical systems for the following reasons. First, the l_∞ string stability criterion:

$$\left\| \frac{Y_n(s)}{Y_{n-1}(s)} \right\|_\infty \leq 1 \tag{6}$$

states that the impulse response of the ego vehicle ($Y_n(s)$) triggered by the preceding vehicle should not exceed the perturbation of the preceding vehicle ($Y_{n-1}(s)$) [17]–[19]. This is identical to the rationale of residual vibration percentage, as eliminating overshoot/undershoot will also prevent string-unstable responses. Second, the trajectory control of an ACC system can also be modeled as a second-order dynamical system, and the input signal of an ACC system is the speed of the preceding vehicle [20]. Specifically, the longitudinal motion of the ego vehicle n is modeled as a double integrator [4], [21], [22]:

$$\dot{p}_n(t) = v_n(t) \tag{7a}$$

$$\dot{v}_n(t) = a_n(t) \tag{7b}$$

where $p_n(t)$ and $v_n(t)$ are position and speed of ego vehicle n at time t, respectively. The acceleration $a_n(t)$ of the ACC-driven ego vehicle can be described using the PD controller type constant time headway-relative velocity model (CTH-RV) [2], [10], [23], which rectifies the spacing error and speed tracking error in the CF process:

$$a_n(t) = k_p e_n^p(t) + k_v e_n^v(t) \tag{8}$$

where $e_n^p(t) = p_{n-1}(t) - p_n(t) - hv_n(t) - d_0$ is the spacing error under the constant time headway policy, $e_n^v(t) = v_{n-1}(t) - v_n(t)$ is the speed tracking error. $p_{n-1}(t)$ and $v_{n-1}(t)$ are position and speed of the preceding vehicle $n-1$ at time t, respectively. h is the desired time headway to maintain between vehicles for ensuring safety and traffic efficiency. k_p and k_v are the control gains to rectify spacing error and speed tracking error, respectively. d_0 is the vehicle bumper-to-bumper distance, which is set as 4m in this study.

Next, to achieve string stability using the input shaping technique, we first derive the transfer function of an ACC system based on (7) and (8) as:

$$\frac{\mathcal{L}(p_n)}{\mathcal{L}(p_{n-1})} = \frac{k_v s + k_p}{s^2 + (k_p h + k_v)s + k_p} \triangleq \frac{(2\zeta\omega_0 - \omega_0^2 h)s + \omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{9}$$

where $\mathcal{L}(\cdot)$ stands for Laplace transform of $\cdot$. Correspondingly, the natural frequency and damping ratio of an ACC system can be computed as:

$$\omega_0 = \sqrt{k_p} \tag{10a}$$

$$\zeta = 0.5(k_p h + k_v)/\sqrt{k_p} \tag{10b}$$

The magnitudes and time instances of impulses can then be computed by substituting (10) into (5). Next, the impulse sequence $\{(t_j, M_j), j = 1, 2\}$ is convoluted with the measured position and speed information of the preceding vehicle to obtain the shaped trajectory information:

$$p_{n-1}^{\text{shaped}}(t) = M_1 p_{n-1}(t - t_1) + M_2 p_{n-1}(t - t_2) \tag{11a}$$

$$v_{n-1}^{\text{shaped}}(t) = M_1 v_{n-1}(t - t_1) + M_2 v_{n-1}(t - t_2) \tag{11b}$$

$p_{n-1}^{\text{shaped}}(t)$ and $v_{n-1}^{\text{shaped}}(t)$ are used to replace $p_{n-1}(t)$ and $v_{n-1}(t)$ for computing $e_n^p(t)$ and $e_n^v(t)$ in (8), respectively. Note that although VTS can ensure string stability with the estimated damping ratio and natural frequency of an ACC system, it cannot guarantee other performance metrics (e.g., safety and efficiency). In the next section, MPC method is incorporated into the design of TS to enhance the CF performance in terms of safety, efficiency, and comfort.

Remark 1: The implementation of VTS relies on the estimation of k_p, k_v, and h (to compute ω_0 and ζ), which can be achieved via online parameter estimation algorithms based on real-time measurements [23]–[26] and offline CF model calibration based on historical data [2], [27], [28]. This study also assumes the perception module has removed the measurement noise effect.

III. MPC-Based Trajectory Shaper

The MPC-TS optimizes the shaped position and speed information of the preceding vehicle at each time step. It leverages VTS to generate a string-stable reference ego vehicle trajectory and mimics the reference trajectory to improve string stability performance. The tracking errors of desired spacing and speed with respect to the preceding vehicle are also embedded in the MPC cost function to guarantee desired tracking performance. Moreover, the constraints on safe spacing, speed fluctuations, and vehicle acceleration/deceleration limit are incorporated in MPC-TS to guarantee safety, efficiency, comfort, and feasible vehicle trajectories in real-world operations. Specifically, the MPC-TS is formulated as follows.

$$\min_{\boldsymbol{\delta} \in \mathbb{R}^{2\times T_p}} \quad \sum_{i=k}^{k+T_p-1} \|\boldsymbol{e}_n(i)\|_{Q_1}^2 + \|\boldsymbol{e}_n^{\text{VTS}}(i)\|_{Q_2}^2 + R a_n(i)^2 \tag{12a}$$

$$\text{s.t.} \quad \boldsymbol{x}(i+1) = A\boldsymbol{x}(i) + B a_n(i) + E a_{n-1}(i) \tag{12b}$$

$$a_n(i) = k_p(e_n^p(i) + \delta_p(i)) + k_v(e_n^v(i) + \delta_v(i)) \tag{12c}$$

$$a_{n-1}(i) = a_{n-1}(k) e^{-0.5[(i-k)\text{dt}]} \tag{12d}$$

$$p_n^{\text{VTS}}(i),\ v_n^{\text{VTS}}(i) \text{ from (11), (12b), (12c), (12d)} \tag{12e}$$

$$\boldsymbol{x}_{\min} \le \boldsymbol{x}(i) \le \boldsymbol{x}_{\max} \tag{12f}$$

$$a_{\min} \le a_n(i) \le a_{\max} \tag{12g}$$

where T_p is the optimization horizon. $\boldsymbol{\delta}(i) = [\delta_p(i), \delta_v(i)]^\top$ is the vector of control decision at time step i, $\delta_p(i)$ and $\delta_v(i)$ are the optimal shaped magnitudes to properly modify the measured position and speed information of the preceding vehicle $n-1$, respectively. $\boldsymbol{e}_n(i) = [e_n^p(i), e_n^v(i)]^\top$ is the vector of tracking errors with respect to the preceding vehicle, $e_n^p(i)$ and $e_n^v(i)$ are the actual discrete spacing error and speed tracking error, respectively. $\boldsymbol{e}_n^{\text{VTS}}(i) = [e_{n,p}^{\text{VTS}}(i), e_{n,v}^{\text{VTS}}(i)]^\top$ is the tracking error with respect to the reference trajectory generated by VTS, $e_{n,p}^{\text{VTS}}(i) = p_n^{\text{VTS}}(i) - p_n(i)$ is the difference between actual position and the position produced by the original ACC with VTS, $e_{n,v}^{\text{VTS}}(i) = v_n^{\text{VTS}}(i) - v_n(i)$ is the difference between actual speed and the speed produced by the original ACC with VTS. The acceleration of the preceding vehicle is assumed to be an exponential decay function (12d) to enhance

safety and trajectory smoothness in the optimization horizon [7]. This study sets $T_p = 10$.

Given a vector $\boldsymbol{y}$ and a positive-definite matrix X, $\|\boldsymbol{y}\|_X^2$ denotes the quadratic form $\boldsymbol{y}^\top X \boldsymbol{y}$. $Q_1 = \begin{bmatrix} w_p/(hv_{\text{lim}} + d_0) & 0 \\ 0 & w_v/\Delta v_{\text{max}} \end{bmatrix}$ is the normalized weighting matrix to regulate $\boldsymbol{e}_n(i)$, where increasing w_p and w_v will enforce spacing error and speed tracking error decay to zeros faster. $Q_2 = \begin{bmatrix} w_p^{\text{VTS}} & 0 \\ 0 & w_v^{\text{VTS}} \end{bmatrix}$ is the weighting matrix to regulate $\boldsymbol{e}_{\text{VTS}}(i)$, where increasing w_p^{VTS} and w_v^{VTS} will enforce the trajectory generated by MPC-based TS to be more similar to that of VTS (more string stable). R is the weighting coefficient to regulate vehicle acceleration/deceleration. Increasing R will induce more comfortable and smoother vehicle trajectories, while potentially compromising other performance metrics (e.g., safety and efficiency). The first term of (12a) seeks to ensure desired tracking performance and traffic efficiency, the second term of (12a) aims at improving string stability, the third term of (12a) is to enhance comfort and (potentially) fuel saving. Note that the weighting of each objective can be adjusted based on the requirements of real-world operations to balance different control performance metrics.

In the constraints, $\boldsymbol{x}_n(i) = [\Delta p_n(i), e_n^v(i), v_n(i)]^\top$ is the vehicle state vector, $\Delta p_n(i) = p_{n-1}(i) - p_n(i)$ is the spacing, $A = \begin{bmatrix} 1 & \text{dt} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} -0.5\text{dt}^2 \\ -\text{dt} \\ \text{dt} \end{bmatrix}$, $E = \begin{bmatrix} 0.5\text{dt}^2 \\ \text{dt} \\ 0 \end{bmatrix}$. dt is the step length, which is set as 0.1s in this study. $\boldsymbol{x}_{\text{min}} = [d_0, -\Delta v_{\text{max}}, 0]^\top$ is the lower-bound of vehicle states, $\boldsymbol{x}_{\text{max}} = [hv_{\text{lim}} + d_0, \Delta v_{\text{max}}, v_{\text{lim}}]^\top$ is the upper-bound of vehicle states, Δv_{max} is the maximum allowable speed difference to mitigate speed fluctuations, v_{lim} is the speed limit for enhancing safety, $hv_{\text{lim}} + d_0$ is the maximum allowable spacing to ensure desired tracking performance. a_{min} and a_{max} are the maximum vehicle deceleration and acceleration, respectively. The constraints on vehicle states and acceleration/deceleration are significant to enhance the safety, efficiency, and feasibility of trajectories computed by the TS-based method in real-world operations, which are not explicitly factored in VTS.

Remark 2: The MPC-TS is formulated in a quadratic programming fashion, which can be solved efficiently by existing solvers and will not induce significant computational burdens on a vehicle. As the MPC-TS is executed on each ACC-equipped vehicle distributedly, it will also support the operation of a long platoon without incurring computational concerns as centralized MPC methods. The feasibility of MPC-TS is influenced by the initial conditions and constraints of states and vehicle acceleration/deceleration. In addition, if the initialization of the CF process does not violate constraints in (12) and the preceding vehicle does not accelerate/decelerate beyond the capability of the ego vehicle, the MPC-TS will always be feasible.

IV. Numerical Experiments

This section presents two numerical experiments to validate the CF performance of the TS-based method and compare it to other baselines. The first experiment applies a synthesized trapezoidal speed profile on the lead vehicle to evaluate the control performance under abrupt acceleration/deceleration. The second experiment applies naturalistic speed profiles from the processed NGSIM trajectory dataset [29] to thoroughly evaluate the control performance in real-world operations. In both experiments, the MPC-TS is programmed in CVXPY [30] and solved using OSQP solver [31]. Table I lists the parameters of the MPC-TS and the CTH-RV model.

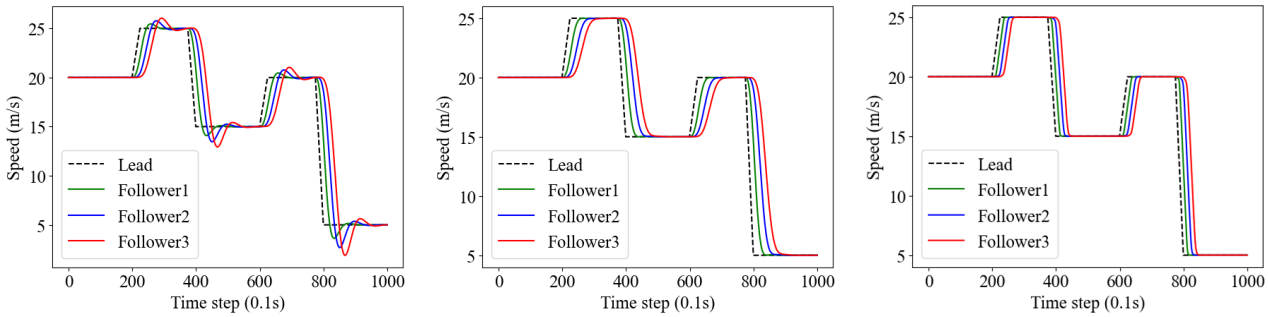
TABLE I
Parameters of MPC-TS

Parameter	w_p	w_v	w_p^{VTS}	w_v^{VTS}	R	k_1
Value	2	1	1	1	0.5	0.9
Parameter	k_2	h	a_{min}	a_{max}	v_{lim}	Δv_{max}
Value	0.3	1.25s	-6m/s^2	3m/s^2	30m/s	5m/s

A. Abrupt Acceleration/Deceleration Case

The abrupt acceleration/deceleration case is frequently used to provide insights on the stability, efficiency, and safety of ACC systems [3]. As shown in Fig. 1, the lead vehicle executes abrupt constant acceleration/deceleration for 2.5s and then maintains a steady speed level, forming a trapezoidal speed profile. The baseline ACC is simulated using the CTH-RV model with parameters listed in Table I. Fig. 1(a) shows that the three followers driven by the baseline ACC exhibit amplified speed fluctuations. By contrast, Fig. 1(b) illustrates the effectiveness of VTS in alleviating string instability to dampen traffic oscillations. Further, Fig. 1(c) shows that the proposed MPC-TS can also achieve string stability and converge to desired speed asymptotically. Moreover, the speed profiles produced by MPC-TS are more compact compared to those of VTS, indicating improved traffic efficiency (as the transient gaps are reduced and vehicles form a tighter formation). This is further validated by the average time headway shown in Fig. 2, where we can observe that the average time headway corresponding to the ACC with MPC-TS fluctuates less than that of the baseline ACC, and stays closer to the desired value (i.e., 1.25s) all the time. This manifests desired stability and efficiency achieved by MPC-TS. By contrast, the average time headway of the ACC with VTS deviates from the desired value the most, especially when the lead vehicle conducts a harsh brake (at around time step 900). In addition, Table II shows the minimum time-to-collision (TTC) of each follower in the CF process, which shows that the ACC with VTS yields the smallest TTC, while the ACC with MPC-TS leads to the largest TTC. The results indicate that although VTS can ensure string stability, it will inevitably sacrifice traffic efficiency and safety. The MPC-TS guarantees a desired balance among string stability, efficiency, and safety.

Table III compares the computing time of three methods. The computing time of MPC-TS (i.e., mean of 2.06ms,



(a) Baseline ACC (b) ACC w/ VTS (c) ACC w/ MPC-TS

Fig. 1. Speed profiles of four-vehicle platoon

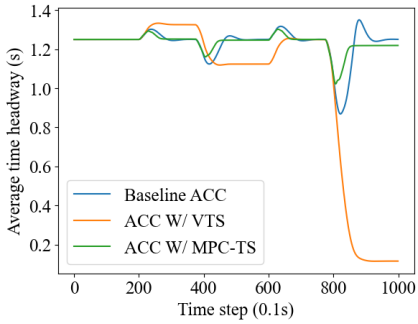


Fig. 2. Comparison of average time headway across three followers

TABLE II
COMPARISON OF MINIMUM TTC

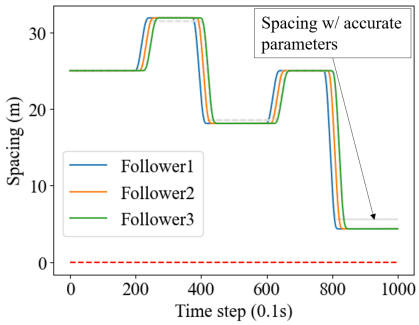
	Follower 1	Follower 2	Follower 3
Baseline ACC	1.35	1.56	1.54
ACC w/ VTS	0.92	1.15	1.38
ACC w/ MPC-TS	1.50	1.62	1.71

standard deviation of 1.35ms) is greater than the computing time of baseline ACC and ACC w/ VTS, as they both have closed-form solutions. However, the computing time of MPC-TS is still substantially smaller than the sampling time step (i.e., 0.1s) of trajectory planning, which enables MPC-TS to be implemented in real-world operations without incurring computational concerns. Moreover, if MPC-TS is programmed using a more efficient language (e.g., C++) and solved using commercial solvers, its computing time will be even smaller.

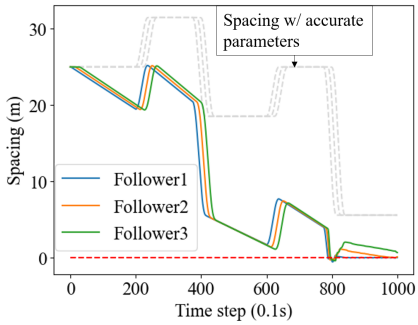
The robustness of the proposed MPC-TS is then tested by including errors to the parameters of the CTH-RV model, such that the CTH-RV model in (12) cannot describe the actual CF behavior of the real ACC system. First, we alter the k_s and k_v of the real ACC system to 0.6 and 0.15, respectively. Fig. 3(a) shows that if only control gains are inaccurate, the ACC with MPC-TS can still maintain robust and stable operations (with time headway slightly deviates from the desired one). Next, we further reduce h of the real ACC system to 1s. As shown in Fig. 3(b), the ACC with MPC-TS leads to unsafe operations (spacings lose track of desired values and shrink to zeros). This is because the ACC system seeks to maintain a different time headway compared to the TS, leading to fundamentally different CF behaviors and substantially large spacing errors. Thus, it is significant to accurately identify the desired time headway or spacing to implement the MPC-TS.

TABLE III
COMPARISON OF COMPUTING TIME

Method	**Baseline ACC**	**ACC w/ VTS**	**ACC w/ MPC-TS**
Computing time (ms)	< 0.01	< 0.01	2.06 (1.35)



(a) k_s and k_v are inaccurate

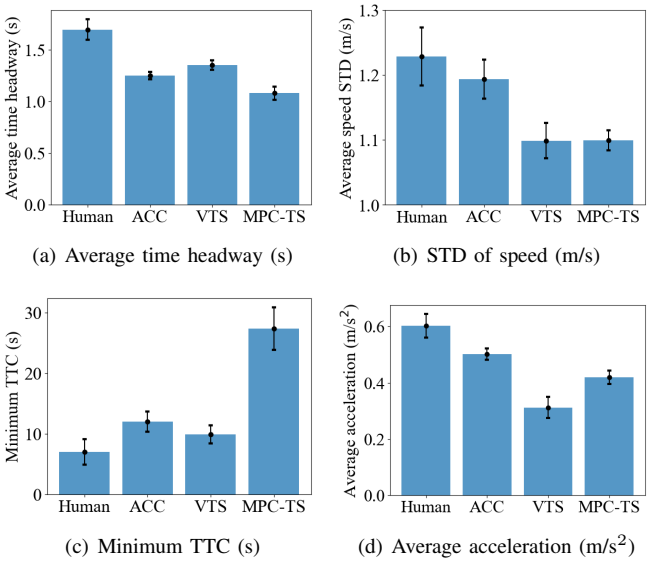


(b) k_s, k_v, and h are inaccurate

Fig. 3. Robustness evaluation of MPC-TS

B. Naturalistic Trajectory Case

This subsection applies 1341 naturalistic trajectories (containing positions, speeds, and accelerations) from the processed NGSIM dataset [29] to simulate the motion of the lead vehicle in real-world operations. The average length of the trajectories is around 24 seconds. The follower trajectory of each lead vehicle is also extracted from the dataset as the human driver baseline. Fig. 4 compares the CF performance of the follower vehicle driven by human drivers (labeled as human), the baseline ACC (labeled as ACC), the ACC with VTS (labeled as VTS), and the ACC with MPC-TS (labeled as MPC-TS). Fig. 4(a) shows that the ACC with MPC-TS achieves the best traffic efficiency, followed by the baseline ACC, the ACC with VTS, then the human driver. The black error bars show the 95% confidence intervals. The error bars with minimal overlapping regions indicate statistically significant differences and validate that the ACC with MPC-TS can improve traffic efficiency (achieving the smallest time headway) in real-world operations. Fig. 4(b) illustrates that the ACC with MPC-TS achieves the smallest standard deviation (STD) of vehicle speed, followed by the ACC with VTS, the baseline ACC, then the human driver baseline. The overlapping confidence interval indicates the ACC with MPC-TS achieves similar string stability performance to alleviate traffic congestion compared to the ACC with VTS. Fig. 4(c) shows that the ACC with MPC-TS can significantly improve safety by increasing TTC in the CF process. Fig. 4(c) also validates that the ACC with VTS will inevitably sacrifice safety (i.e., smaller TTCs) compared to the baseline ACC. Fig. 4(d) illustrates that the ACC with VTS achieves the smallest average acceleration/deceleration, then the ACC with MPC-TS, followed by the baseline ACC and human driver. This is because VTS only seeks to achieve string stability, which enforces the ACC to use smaller acceleration/deceleration to attain smoother trajectories. However, this sacrifices safety and traffic efficiency (as shown in Figs. 4(a) and 4(c)). By contrast, the ACC with MPC-TS applies slightly larger acceleration/deceleration to balance the performance of string stability, safety, and efficiency. Meanwhile, as the human driver baseline and the baseline ACC are string-unstable, they inevitably lead to amplified speed fluctuations and increased acceleration/deceleration.



(a) Average time headway (s) (b) STD of speed (m/s)

(c) Minimum TTC (s) (d) Average acceleration (m/s^2)

Fig. 4. Comparison of CF performance

V. Concluding Comments

This study proposes the TS-based method to mitigate string instability of commercially-available ACC systems. The proposed MPC-TS leverages the VTS and MPC to achieve desired string stability, safety, and traffic efficiency, given the original ACC systems with undesired CF performance. In particular, it functions as a signal filter to modify the sensor-measured trajectory information of the preceding vehicle, which reduces the costs of tuning and/or developing control algorithms to benefit traffic operations. The MPC framework of MPC-TS also enables ACC systems to achieve an optimal trade-off among different CF performance metrics. This study points to the following future directions: (i) design robust MPC to factor inaccurate dynamics model and measurement noise; (ii) extend TS to cooperative ACC, while considering dynamic communication channels [32]; and (iii) leverage TS in CF advisory system (e.g., advisory speed via head-up display) to stabilize human drivers in mixed-flow traffic.

References

[1] G. Gunter, D. Gloudemans, R. E. Stern, S. McQuade, R. Bhadani, M. Bunting, M. L. Delle Monache, R. Lysecky, B. Seibold, J. Sprinkle, B. Piccoli, and D. B. Work, "Are commercially implemented adaptive cruise control systems string stable?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6992–7003, 2021.

[2] G. Gunter, C. Janssen, W. Barbour, R. E. Stern, and D. B. Work, "Model-based string stability of adaptive cruise control systems using field data," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 90–99, 2019.

[3] M. Makridis, K. Mattas, A. Anesiadou, and B. Ciuffo, "Openacc. an open database of car-following experiments to study the properties of commercial acc systems," *Transportation research part C: emerging technologies*, vol. 125, p. 103047, 2021.

[4] G. J. Naus, R. P. Vugts, J. Ploeg, M. J. van De Molengraft, and M. Steinbuch, "String-stable cacc design and experimental validation: A frequency-domain approach," *IEEE Transactions on vehicular technology*, vol. 59, no. 9, pp. 4268–4279, 2010.

[5] A. Zhou, S. Gong, C. Wang, and S. Peeta, "Smooth-switching control-based cooperative adaptive cruise control by considering dynamic information flow topology," *Transportation Research Record*, vol. 2674, no. 4, pp. 444–458, 2020.

[6] A. Zhou, J. Wang, and S. Peeta, "Robust control strategy for platoon of connected and autonomous vehicles considering falsified information injected through communication links," *Journal of Intelligent Transportation Systems*, pp. 1–17, 2022.

[7] H. Zhou, A. Zhou, T. Li, D. Chen, S. Peeta, and J. Laval, "Congestion-mitigating mpc design for adaptive cruise control based on newell's car following model: History outperforms prediction," *Transportation Research Part C: Emerging Technologies*, vol. 142, p. 103801, 2022.

[8] J. Wang, S. Gong, S. Peeta, and L. Lu, "A real-time deployable model predictive control-based cooperative platooning approach for connected and autonomous vehicles," *Transportation Research Part B: Methodological*, vol. 128, pp. 271–301, 2019.

[9] Y. Lin, J. McPhee, and N. L. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 221–231, 2020.

[10] H. Zhou, A. Zhou, T. Li, D. Chen, S. Peeta, and J. Laval, "Significance of low-level control to string stability under adaptive cruise control: Algorithms, theory and experiments," *Transportation research part C: emerging technologies*, vol. 140, p. 103697, 2022.

[11] X.-Y. Lu and S. Shladover, "Truck cacc system design and dsrc messages," *FHWA Explor. Adv. Res. Program Cooperat. Agreement, Task*, 2018.

[12] S. E. Shladover, C. Nowakowski, D. Cody, F. Bu, J. O'Connell, J. Spring, S. Dickey, and D. Nelson, "Effects of cooperative adaptive cruise control on traffic flow: testing drivers' choices of following distances," 2009.

[13] H. Zhou, J. Laval, A. Zhou, Y. Wang, W. Wu, Z. Qing, and S. Peeta, "Review of learning-based longitudinal motion planning for autonomous vehicles: research gaps between self-driving and traffic congestion," *Transportation research record*, vol. 2676, no. 1, pp. 324–341, 2022.

[14] W. E. Singhose, W. P. Seering, and N. C. Singer, "Input shaping for vibration reduction with specified insensitivity to modeling errors," *Japan-USA Sym. on Flexible Automation*, vol. 1, pp. 307–13, 1996.

[15] T. Singh and W. Singhose, "Input shaping/time delay control of maneuvering flexible structures," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 3. IEEE, 2002, pp. 1717–1731.

[16] Y. Zhao, W. Chen, T. Tang, and M. Tomizuka, "Zero time delay input shaping for smooth settling of industrial robots," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2016, pp. 620–625.

[17] J. Ploeg, N. Van De Wouw, and H. Nijmeijer, "Lp string stability of cascaded systems: Application to vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 786–793, 2013.

[18] Y. Liu, A. Zhou, Y. Wang, and S. Peeta, "Proactive longitudinal control to manage disruptive lane changes of human-driven vehicles in mixed-flow traffic," *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 321–326, 2021.

[19] A. Zhou, S. Peeta, and J. Wang, "Cooperative control of a platoon of connected autonomous vehicles and unconnected human-driven vehicles," *Computer-Aided Civil and Infrastructure Engineering*, 2023.

[20] Y. Wang, M. L. Delle Monache, and D. B. Work, "Identifiability of car-following dynamics," *Physica D: Nonlinear Phenomena*, vol. 430, p. 133090, 2022.

[21] S. Feng, Z. Song, Z. Li, Y. Zhang, and L. Li, "Robust platoon control in mixed traffic flow based on tube model predictive control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 711–722, 2021.

[22] M. Montanino, J. Monteil, and V. Punzo, "From homogeneous to heterogeneous traffic flows: Lp string stability under uncertain model parameters," *Transportation Research Part B: Methodological*, vol. 146, pp. 136–154, 2021.

[23] Y. Wang, G. Gunter, M. Nice, M. L. Delle Monache, and D. B. Work, "Online parameter estimation methods for adaptive cruise control systems," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 288–298, 2020.

[24] J. Monteil, N. O'Hara, V. Cahill, and M. Bouroche, "Real-time estimation of drivers' behaviour," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 2046–2052.

[25] Z. Wang, X. Zhou, and J. Wang, "Algebraic car-following model parameter identification," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 864–869, 2021.

[26] ——, "An algebraic evaluation framework for a class of car-following models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 366–12 376, 2021.

[27] V. Punzo, Z. Zheng, and M. Montanino, "About calibration of car-following dynamics of automated and human-driven vehicles: Methodology, guidelines and codes," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103165, 2021.

[28] A. Zhou, Y. Liu, E. Tenenboim, S. Agrawal, and S. Peeta, "Car-following behavior of human-driven vehicles in mixed-flow traffic: A driving simulator study," *IEEE Transactions on Intelligent Vehicles*, 2023.

[29] M. Montanino and V. Punzo, "Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns," *Transportation Research Part B: Methodological*, vol. 80, pp. 82–106, 2015.

[30] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[31] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[32] Y. Bai, K. Zheng, Z. Wang, X. Wang, and J. Wang, "Mc-safe: Multi-channel real-time v2v communication for enhancing driving safety," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 4, pp. 1–27, 2020.