String instability mitigation of adaptive cruise control without modifying control laws: Trajectory shaper and parameter estimation

Anye Zhou[a*], Hao Zhou[b], Jorge Laval[c], and Srinivas Peeta[c]

[a]*Buildings and Transportation Science Division, Oak Ridge National Laboratory, Oak Ridge, USA;* [b]*Civil and Environmental Engineering, University of South Florida, Tampa, USA;* [c]*Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, USA*

*Corresponding author, Buildings and Transportation Science Division, Oak Ridge National Laboratory, Knoxville, TN 37932, USA; e-mail: zhoua@ornl.gov

String instability mitigation of adaptive cruise control without modifying control laws: Trajectory shaper and parameter estimation[*]

Vehicle automation technologies enable vehicles to be equipped with adaptive cruise control (ACC) systems, which relieve highway driving fatigue. However, recent studies have shown that the current ACC systems on commercially-available vehicles are string-unstable (i.e., exacerbate shockwave propagation and congestion). To achieve string stability, most existing studies seek to directly modify the control algorithms of ACC systems. Alternatively, this study proposes a trajectory shaper (TS)-based method, which only modifies the trajectory information (i.e., position and speed measured by sensors) of the predecessor vehicle, so that the ego vehicle driven by a string-unstable ACC system leverages the modified trajectory information to achieve string stability. To devise this TS-based method, the batch-optimisation and extended Kalman filter methods are applied to estimate the parameters of an ACC system in an offline-online fashion. The conditions of persistent excitation and observability-identifiability are analysed to determine the sufficient conditions for obtaining accurate parameters of an ACC system. The proposed TS-based method is cost-effective during implementation, as they avoid modifying existing ACC control algorithms (which entails complex analysis of ACC control systems and parameter tuning). The effectiveness of the proposed TS-based method is validated through numerical experiments. The results suggest that they can achieve desired string stability without modifying existing ACC control algorithms.

Keywords: adaptive cruise control; string stability; stop-and-go traffic; trajectory shaper; parameter estimation

---

Introduction

Due to the rapid advances in vehicle automation technologies, adaptive cruise control (ACC) systems are now widely installed on commercially-available vehicles around the world. As illustrated in the purple box of Figure 1, an ACC system consists of a high-level planner (HP) and a low-level controller (LC) (Zhou et al., 2022b; Naus et al., 2010). The HP uses the sensor-measured trajectory information (i.e., position, speed) of the predecessor vehicle to plan the desired trajectory (e.g., acceleration or speed in future time steps) so that the desired car-following (CF) control performance (e.g., tracking, collision avoidance, comfort, fuel economy, etc.) can be achieved for the ego vehicle. The LC determines the appropriate gas/brake command to be sent to the vehicle actuators (i.e., engine, transmission, brake system), so that the planned acceleration/speed can be executed on the ego vehicle plant. As a critical component of an Advanced Driver Assistance System (ADAS), the ACC system can alleviate drivers' fatigue and enhance safety and comfort in the driving task. However, recent studies suggest that the ACC systems available on commercially-available vehicles are string-unstable (Li et al., 2021; Makridis et al., 2021), indicating that vehicles driven by ACC systems will amplify speed fluctuations originating from the downstream traffic, which can amplify traffic oscillations and lead to undesired stop-and-go waves.

To guarantee string stability for dampening speed fluctuations and mitigating traffic oscillations, the most intuitive approach is to modify the ACC control algorithms (i.e., tuning parameters of HP and LC, or even revising the control algorithms). Several studies have investigated the design of HP to achieve string stability. For instance, proportional-derivative (PD) type linear planner and its variants have been studied extensively (Gong, Zhou, and Peeta, 2019; Zhou et al., 2020; Zhou and Ahn, 2019). Model predictive controller (MPC) type planners have also been investigated to

incorporate constraints related to safety, speed limit, and desired acceleration/deceleration limit (Gong, Shen, and Du, 2016; Wang et al., 2019; Zhou, Wang, and Ahn, 2019). The proportional-integral-derivative (PID) control (Zhou et al., 2022b), feedback linearization control (FLC) (Lu and Shladover, 2018), and loop shaping control (Shladover, 2009) techniques have been studied in the LC to minimize the tracking error for realizing the planned string-stable trajectory. However, directly modifying the algorithms of an existing ACC system to achieve string stability suffers from the following three challenges. First, the performance of the LC can also significantly impact string stability (Zhou et al., 2022b), which has been neglected in most studies. The HP and LC impact each other during ACC operations; the trajectory planning in the HP addresses the imperfect execution from the LC, while the LC counteracts the variations in planned trajectory from the HP. Hence, their interactions induce complex correlations that makes string stability analysis even more difficult (typically entails numerous trial-and-error attempts). Second, the control algorithms of vehicle engine, transmission, and brake system are determined by the original equipment manufacturers (OEMs) (components in the dash-line box in Figure 1). As these are proprietary information of the OEMs, it is difficult to access these control algorithms (in the factory powertrain control unit) and the corresponding actuator dynamics. Moreover, retrofitting the actuator response typically entails extensive experiments to construct look-up tables and fit empirical functions, which is laborious and time-consuming. If the vehicle actuator dynamics are not accurately modelled or are unknown, how the ego vehicle responds to the LC given the real-world disturbances (e.g., air drag, rolling resistance, road grade, etc.) and measurement noise during ACC operations (as shown in Figure 1) cannot be robustly predicted. The challenges in explicitly modeling the actuator dynamics would substantially degrade the tracking performance of the LC, which compromises string

stability. Third, as the autonomous vehicle industry is shifting from rule-based control methods (i.e., control decisions with explicit derivations and closed-form expressions) to end-to-end learning-based control methods (i.e., directly mapping sensor measurements to the control decision without modularizing HP and LC) (Zhou et al., 2022a), the complexity and intractability of deep neural networks substantially increase the effort and cost to tune and revise existing ACC control algorithms. Hence, based on these three challenges, directly tuning and revising the existing ACC control algorithm in a commercially-available vehicle to achieve string stability can be rather expensive, difficult and laborious.

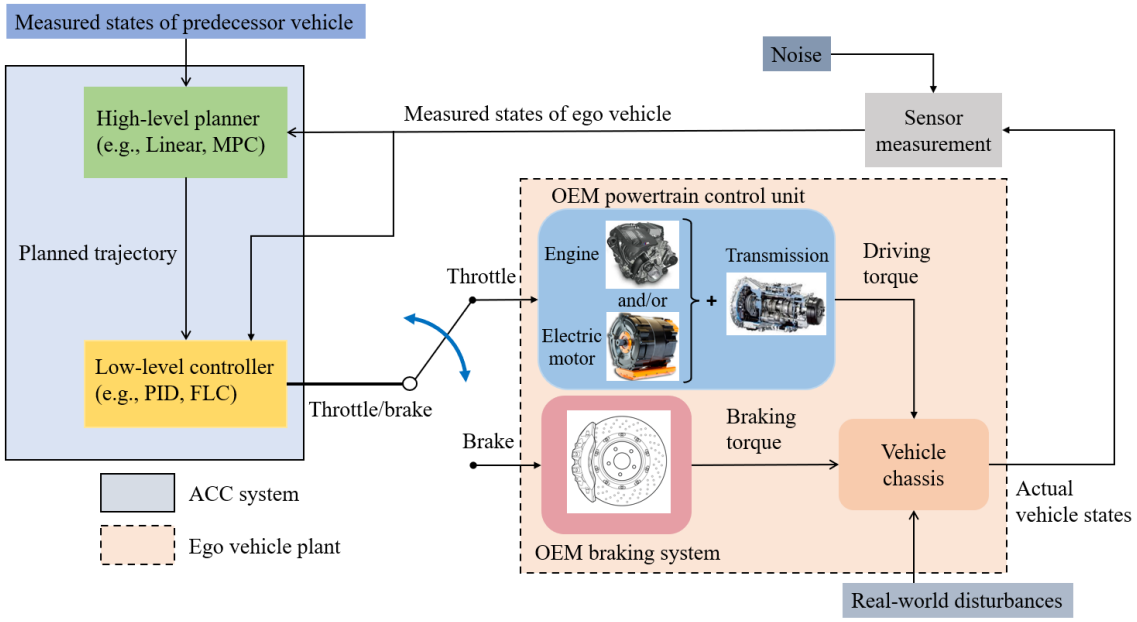


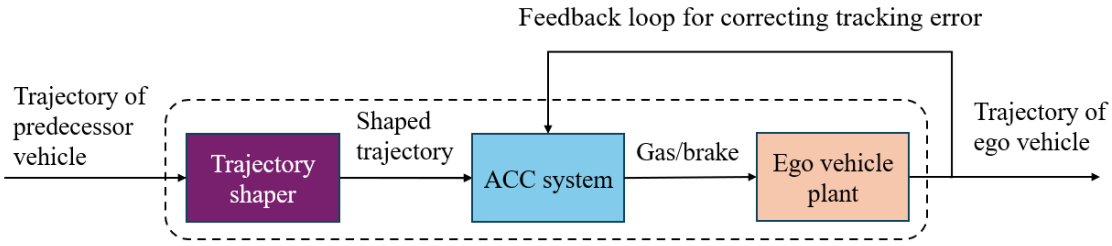
Figure 1. Ego vehicle control by ACC: gasoline vehicle uses engine; electric vehicle uses electric motor; hybrid vehicle uses both engine and electric motor

To circumvent the need to directly modify the existing algorithms of an ACC system and achieve string stability in a cost-effective manner, this study proposes trajectory shaper (TS)-based method which only modifies the sensor-measured trajectory information (i.e., position, speed signal) of the predecessor vehicle before it is used by the ACC system of the ego vehicle, as shown in Figure 2(a). The proposed TS-based method leverages the fact that the trajectory information of the predecessor vehicle

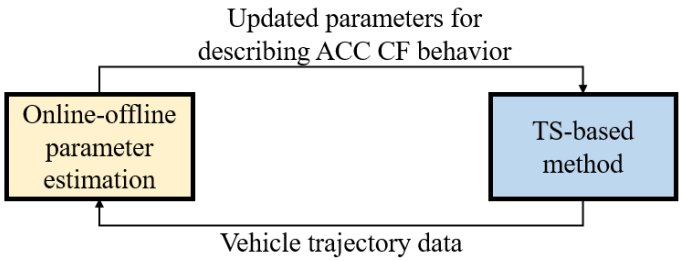
influences the trajectory of the ego vehicle. Specifically, the trajectory information of the predecessor vehicle can impact the CF behaviour of the ego vehicle. For instance, the predecessor vehicle can adjust its speed to achieve desired spacing between itself and the ego vehicle, which will reduce the speed variations of the ego vehicle. The design of the TS is inspired by the input shaping approach (Singh and Singhose, 2002; Singhose, Seering, and Singer, 1996) applied to attenuate residual vibrations of mechanical systems (e.g., cranes, robot manipulators). During operations, the TS functions as a signal filtering plug-in in the autonomous driving perception module, processes the trajectory information of the predecessor vehicle, and removes its undesired characteristics that will induce string-unstable CF behaviour from an ACC system. The TS also shares a similar rationale with cyberattacks, while exercising totally opposite objectives. Both TS and cyber attacker seeks to vary the information used by vehicle control systems to alter the vehicle trajectories. However, the TS seeks to improve the performance of a suboptimal ACC to benefit the traffic flow, while a cyber attacker seeks to deteriorate vehicle controllers to perturb the traffic and reduce safety.

To ensure desired string stability performance of the TS-based method, only two parameters of an ACC system need to be accurately identified from historical trajectories: (i) natural frequency, and (ii) damping ratio. Estimating these two parameters can be significantly easier than identifying complicated vehicle actuator dynamics (for revising existing control algorithms). The natural frequency corresponds to the speed oscillation characteristics (i.e., the changing rate and period of speed overshoot/undershoot) of an ACC system after encountering speed perturbations from downstream traffic. The damping ratio describes the capability of an ACC system to alleviate speed fluctuations to achieve string stability (i.e., the measure of oscillation attenuation). The two parameters provide a direct interpretation for the string stability property of an ACC system without

the need to know the exact control architecture or algorithms. The magnitude of maximum overshoot/undershoot and the decaying/amplifying characteristics of speed fluctuations can be computed based on natural frequency and damping ratio, which enables the control strategies to mitigate the corresponding speed fluctuations and overshoot/undershoot. Thereby, given a string-unstable ACC system, if these two parameters are accurately estimated, a vanilla trajectory shaper (VTS) can be implemented to alleviate string-unstable CF behaviour. However, the estimated natural frequency and damping ratio may not be accurate because the trajectory data obtained from real-world measurements: (i) can be noisy due to the measurement noise of onboard sensors, and (ii) may not be informative enough to exhibit the CF characteristics of an ACC system comprehensively (e.g., the trajectory in the free-flow traffic condition cannot provide any characteristics related to string stability). Hence, we enhance the robustness of TS design by formulating a robust trajectory shaper (RTS) using a nonlinear program. The RTS can handle different oscillatory speed patterns covering a range of natural frequencies and damping ratios to achieve string stability, which appreciably mitigates the negative impacts of inaccurately estimated parameters. In this study, we first formulate the VTS to explain the fundamental concepts of the TS-based method, and then formulate the RTS for more reliable real-world application. The merits of the TS-based method are twofold. It entails a simple formulation and straightforward implementation (i.e., it can be added to an existing ACC system as a plug-in block, and only entails estimating two parameters rather than the complex vehicle actuator dynamics). Of critical practical importance, it circumvents the need to modify the existing ACC control algorithms within the vehicle.



(a) ACC setup with TS



(b) Automated workflow of proposed strategies

Figure 2. Description of proposed strategies

As the proposed TS-based method builds upon the damping ratio and the natural frequency of an ACC system, the estimation of these two parameters is crucial for real-world implementation. To achieve desired accuracy of parameter estimation, the batch-optimisation and extended Kalman filter (EKF) methods are incorporated to devise an offline-online parameter estimation method. The batch-optimisation method solves a nonlinear least-squares problem to determine the optimal parameters that minimize the difference between the ground-truth and estimated spacings, using the historical trajectory data. It is implemented in an offline fashion, due to the heavy computational burden involving numerous historical trajectories. The estimated parameters from the batch-optimisation method are then applied as the initial condition to initialize the online parameter estimation. In the online parameter estimation process, the EKF uses the real-time measurement information (e.g., position and speed of predecessor and ego vehicles) to iteratively update the estimated parameters of a CF model, by minimizing the difference between ground-truth measurement information and the measurement information computed using the estimated parameters. Then, the estimated parameters

will be utilized to compute the corresponding damping ratio and natural frequency. The offline-online parameter estimation method automates the implementation of TS-based method in real-world operations. As shown in Figure 2(b), the offline-online parameter estimation and the TS-based method forms a closed iterative loop. The TS-based method provides historical trajectories (e.g., data from prior operations) to consistently update the estimated parameter, while the updated estimated parameters are used in TS-based method so that the TS is computed based on accurate ACC car-following behaviour to enhance string stability and safety. This also enables cost-effective operations using the proposed method, and precludes human intervention and empirical efforts in the implementation of TS.

Remarkably, in the parameter estimation process, besides the parameter estimation method, the characteristics of the trajectory (i.e., position, speed) of the predecessor vehicle and the measurement information also exert great influence on the accuracy. Three conditions are sufficient to obtain an accurate estimation of parameters: (i) the trajectory of the predecessor vehicle needs to satisfy the persistent excitation condition (which manifests if certain trajectory data is capable of exciting a dynamical system to display all dynamic characteristics, e.g., magnitudes of overshoot/undershoot, convergence to steady state) (Ljung, 1999); (ii) the trajectory of the predecessor vehicle needs to include scenarios of maximum operating speed (reaching to speed limit) and stop-and-go movements; (iii) the measurement information needs to enable the observability-identifiability property of the CF dynamical system (i.e., involving CF model dynamics and the measurement information evolution). The observability-identifiability property indicates that all states and parameters of a dynamical system can be computed using attainable measurement information. These conditions can be difficult to satisfy in real-world operations, as the traffic condition may not be able to generate the

aforementioned scenarios. Besides, during daily driving, drivers may adjust the vehicle powertrain control modes (e.g., eco and sport) and ACC headway settings, which would alter the damping ratio and natural frequency of the ACC system. Hence, fixed estimated parameters cannot reflect the dynamics of an ACC system accurately, and will further degrade the string stability performance of TS. The proposed offline-online parameter estimation method can address these two issues. Specifically, the EKF will periodically update the estimated parameters based on real-time measurements, which consistently improves the accuracy of parameter estimation, and enables the TS-based method to factor time-varying dynamics during ACC operations. This also improves the reliability and safety of TS-based method.

The contributions of this study are threefold. First, we illustrate the impacts of the predecessor vehicle trajectory on the string stability of an ACC system. Second, the proposed TS-based method can achieve string-stable CF behaviour without modifying the control algorithms in the HP and the LC of a string-unstable ACC system, which offers a cost-effective alternative to improve existing ACC systems. That is, we can circumvent intrusive modifications of OEM control laws in existing vehicles by using plug-in blocks for signal processing. Third, we rigorously articulate the sufficient conditions for identifying the accurate parameters of an ACC system. Based on the sufficient conditions, we further propose an offline-online parameter estimation method to automate the real-world implementation of TS-based method, which enhances its reliability and reduce operational costs.

The remainder of the paper is organized as follows. The next section introduces the formulation of the TS-based method. The offline-online parameter estimation method and the sufficient conditions for identifying accurate parameters are articulated in the section thereafter. The next section validates the proposed TS-based method and offline-

online parameter estimation method using numerical experiments. Last, concluding comments and future directions are provided.

Trajectory Shaper Design

This section first formulates an ACC system to characterize it as a second-order dynamical system. Next, TS-based method is developed based on the CF dynamics and string stability analysis of an ACC system.

ACC Formulation

For tractable analysis, we consider the point-mass type vehicle dynamics, which is a common assumption in string stability-related ACC studies (Gong, Shen, and Du, 2016). Thus, the vehicle motion of the ego vehicle is described as follows:

$$\dot{p}_{\text{ego}}(t) = v_{\text{ego}}(t) \tag{1}$$

$$\dot{v}_{\text{ego}}(t) = a_{\text{ego}}(t) \tag{2}$$

where $p_{\text{ego}}(t)$, $v_{\text{ego}}(t)$, and $a_{\text{ego}}(t)$ are the position, speed, and acceleration of the ego vehicle controlled by ACC, respectively.

The ego vehicle uses onboard sensors (e.g., radar, camera, etc.) to obtain trajectory information of the immediate predecessor vehicle to implement the ACC. Correspondingly, we use a frequently-used constant time headway linear model (CTH-linear) (Gong, Zhou, and Peeta, 2019; Zhou et al., 2020) to describe the ACC system, whose control decision (i.e., vehicle acceleration) is as follows:

$$a_{\text{ego}}(t) = k_p e_p(t) + k_v e_v(t) \tag{3}$$

where $a_{\text{ego}}(t)$ is derived as the control decision of ACC system. $e_p(t) = p_{\text{pred}}(t) - p_{\text{ego}}(t) - hv_{\text{ego}}(t)$ is the spacing error, and $e_v(t) = v_{\text{pred}}(t) - v_{\text{ego}}(t)$ is the speed tracking error. $p_{\text{pred}}(t)$ and $v_{\text{pred}}(t)$ are the position and speed of the immediate

predecessor vehicle, respectively. k_p is the proportional control gain for correcting the spacing error, k_v is the derivative control gain for correcting the speed tracking error, and h is the desired time headway to be maintained.

String Stability and Speed Damping

An ACC system can be interpreted as a second-order dynamical system (Haidekker, 2020), as the control decision (i.e., acceleration) is the second-order derivative of the vehicle position. Correspondingly, in the Laplace domain, the string stability transfer function (SSTF) (Feng et al., 2019) can be expressed as:

$$\Gamma(s) = \frac{\Xi_{\text{ego}}(s)}{\Xi_{\text{pred}}(s)} = \frac{k_v s + k_p}{s^2 + \left(k_p h + k_v\right)s + k_p} \tag{4}$$

where $\Xi_{\text{ego}}(s)$ and $\Xi_{\text{pred}}(s)$ are the positions of the ego and predecessor vehicles in the Laplace domain, respectively. $s = j\omega$ is the Laplace operator, $j = \sqrt{-1}$ is the indicator of complex number, and ω is the angular frequency. The SSTF quantifies how the ego vehicle states vary given the stimulation from the predecessor vehicle.

String stability indicates that once a predecessor vehicle deviates from the equilibrium (i.e., desired spacing and speed) due to a speed perturbation, the speed perturbation will not be amplified by the ego vehicle. This study applies the l_2 string stability criterion as it can be linked to the frequency domain analysis (which is analogous to the TS-based method), provides a direct constraint on speed fluctuations, and enables elegant mathematical analysis. The l_2 string stability states that the l_2 norm of the speed deviation of the ego vehicle is smaller than that of its predecessor vehicle:

$$\left\| v_{ego}(t) - \bar{v}_0 \right\|_{l_2} \leq \left\| v_{pred}(t) - \bar{v}_0 \right\|_{l_2} \tag{5}$$

where $\bar{v}_0$ indicates the operating speed at equilibrium. Correspondingly, for the control design of an ACC system, this phenomenon can be interpreted as: the impulse response

of ego vehicle is bounded by that of its predecessor vehicle. This corresponds to the H_∞ norm of the SSTF (4) being no greater than one:

$$\|\Gamma(j\omega)\|_{\mathcal{H}_\infty} = \sup_{\omega}|\Gamma(j\omega)| \leq 1 \tag{6}$$

where $\|\cdot\|_{\mathcal{H}_\infty}$ denotes the maximum magnitude over the frequency range $[0, \infty)$. Equation (6) can be satisfied through the following inequality (Feng et al., 2019; Naus et al., 2010):

$$\left\| \frac{jk_v\omega + k_p}{j\left(k_p h + k_v\right)\omega + \left(k_p - \omega^2\right)} \right\|_2 \leq 1 \tag{7}$$

which is equivalent to the following condition after arithmetic simplification:

$$k_p^2 h^2 + 2k_p(k_v h - 1) \leq 0 \tag{8}$$

The string stability condition in Equation (8) can ensure that the impulse-type speed perturbation (e.g., sudden braking followed by acceleration, or sudden acceleration followed by braking) will be dampened. The impulse-type speed perturbation frequently appears in real-world driving. However, for a step-function type speed perturbation (e.g., speed suddenly increasing/decreasing to another setpoint), a further constraint on controller parameters is required. Specifically, given a step-function type speed perturbation, the phenomenon of overshoot/undershoot of a second-order dynamical system corresponds to string-unstable CF behavior of the ACC system. The overshoot/undershoot indicates that the speed fluctuation of ego vehicle exceeds the speed variation of its predecessor vehicle. Thereby, the objective of achieving string stability under a step-function type speed perturbation is equivalent to avoiding overshoot/undershoot of the second-order dynamical system in Equations (3) and (4).

The SSTF in Equation (4) can then be equivalently rewritten as a second-order dynamical system (Haidekker, 2020) as follows:

$$\frac{Y(s)}{U(s)} = \frac{(2\zeta\omega_0 - \omega_0^2 h)s + \omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{9}$$

where $Y(s)$ is the output of the system (e.g., trajectory of ego vehicle), and $U(s)$ is the input of the system (e.g., trajectory of predecessor vehicle). $\omega_0 \in \mathbb{R}^+$ is the natural frequency that corresponds to the oscillatory speed patterns of the system (e.g., speed fluctuations, overshoots/undershoots). $\zeta \in \mathbb{R}^+$ is the damping ratio, which describes the capability of attenuating oscillatory speed patterns (i.e., suppressing speed fluctuations and attenuating oscillating movements). To eliminate overshoot/undershoot of a second-order dynamical system, and ensure string stability, the damping ratio should satisfy (Haidekker, 2020):

$$\zeta \geq 1 \tag{10}$$

A system with $\zeta = 1$ is called critically damped, while a system with $\zeta > 1$ is called over-damped. Both critically damped and over-damped systems can guarantee that no overshoot/undershoot will occur, but an over-damped system will converge to the desired setpoint at a slower pace. A system with $0 < \zeta < 1$ is labeled as under-damped, which will exhibit string-unstable CF behavior given a step-function type speed perturbation. From Equations (3) and (4), we have the following equalities:

$$\zeta = \frac{k_p h + k_v}{2\sqrt{k_p}} \tag{11a}$$

$$\omega_0 = \sqrt{k_p} \tag{11b}$$

Thus, to guarantee string stability under step-function type speed perturbation, we require the following constraint on the ACC parameters to avoid under-damped situations:

$$\left(k_p h + k_v\right)^2 - 4k_p \geq 0 \tag{12}$$

Consequently, to dampen both impulse-type and step-function type speed perturbations from the trajectory of the predecessor vehicle, the string stability condition should involve Equations (8) and (12). As illustrated in Figure 3, with time headway set as $h = 1$, the red dotted line is the boundary of the string stability condition in Equation (8), below

(above) which the region is string-unstable (stable). The black solid curve indicates the critically-damped condition, with the blue area below being under-damped and the white area above it being over-damped. Thus, the string stability region for dampening both impulse-type and step-function type speed perturbations is the white area above the yellow dashed curve (which satisfies Equations (8) and (12)). In real-world operations, an ACC system satisfying conditions in Equations (8) and (12) will produce damped speed profiles and alleviate speed fluctuations compared to the predecessor vehicle.

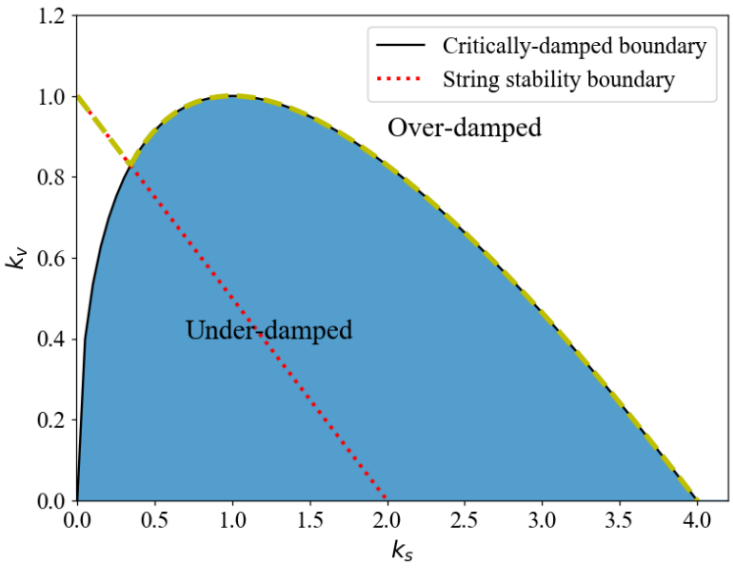
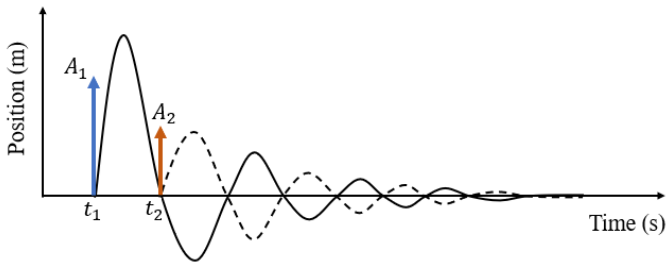


Figure 3. Regions of string stability and over/under-damped system

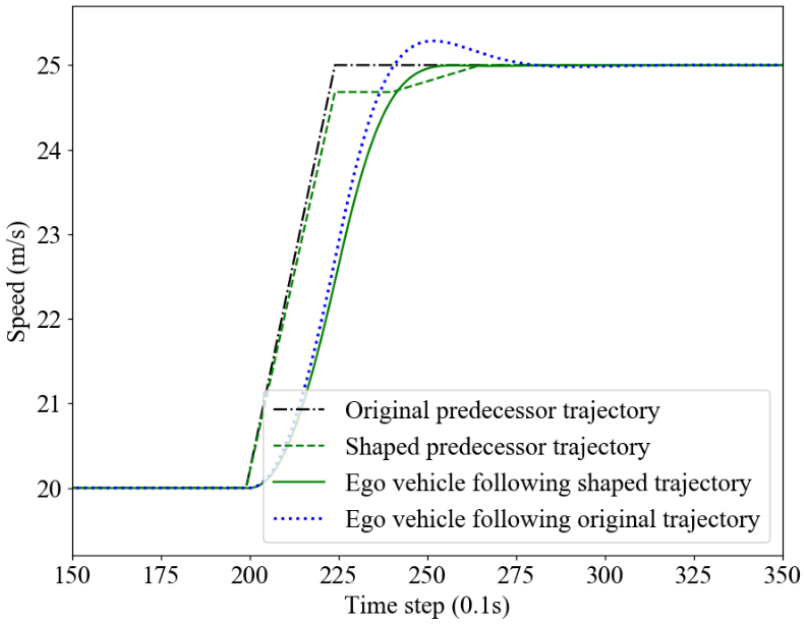
Vanilla Trajectory Shaper

The VTS design is based on the zero vibration shaper (ZVS) method which is designed to attenuate the residual vibration of an mechanical vibration system (Haidekker, 2020). The core idea of ZVS is to appropriately design two sequential impulses to negate residual vibrations, as shown in Figure 4(a). The first impulse A_1 induces a vibrating response (solid curves), while the second impulse A_2 induces another vibrating response (dashed curves). By sequentially applying impulses A_1 and A_2 to the mechanical vibration system, the vibrations induced from these two impulses will cancel each other to achieve a

vibration-free response. The residual vibrations are analogous to the speed fluctuations of an ACC system (as both the mechanical vibration system and the ACC system can be described using the second-order dynamical system model in Equation (9)). Thereby, the core idea of the VTS is the same as that of ZVS. Specifically, the VTS applies impulses A_1 and A_2 to convolute with the speed information of the predecessor vehicle so that the speed variations of the predecessor vehicle are appropriately altered (see the green dashed line in Figure 4(b). The first impulse brings the speed information to an intermediate setpoint for the ego vehicle to respond. Next, when the ego vehicle is about to overshoot, the second impulse drives the speed information to the original speed level, which ensures that the ego vehicle converges to the speed of the predecessor vehicle without overshoot (as illustrated by the blue dotted curve and green solid curve in Figure 3(b)).



(a) Mechanism for eliminating residual vibrations



(b) Mechanism for eliminating speed overshoot from ego vehicle

Figure 4. Mechanism of ZVS and VTS

For a vibrating second-order dynamical system, the performance index of vibration attenuation is defined as the residual vibration percentage under the impact of n impulses (Zhao et al., 2016):

$$V(\omega_0, \zeta) = e^{-\omega_0 \zeta t_n} \sqrt{S(\omega_0, \zeta)^2 + C(\omega_0, \zeta)^2} \tag{13}$$

where $S(\omega_0, \zeta) = \sum_{i=1}^{n} A_i e^{\omega_0 \zeta t_i} \sin\left(\omega_0 \sqrt{1-\zeta^2} t_i\right)$, $C(\omega_0, \zeta) = \sum_{i=1}^{n} A_i e^{\omega_0 \zeta t_i} \cos\left(\omega_0 \sqrt{1-\zeta^2} t_i\right)$, A_i and t_i are the magnitude and occurrence time of impulse i, respectively. The residual vibration percentage measures the vibrations (fluctuations) after applying n impulses to the second-order dynamical system (9). Correspondingly, for the ZVS, to eliminate residual vibrations of a mechanical vibration system using two sequential impulses (i.e., $V = 0, n = 2$), we need to solve the following set of equations:

$$S(\omega_0, \zeta) = 0 \tag{14a}$$

$$C(\omega_0, \zeta) = 0 \tag{14b}$$

$$\sum_{i=1}^{2} A_i = 1 \tag{14c}$$

where Equations (14a) and (14b) seek to push the residual vibration percentage to zero, and Equation (14c) ensures that the system response can still converge to the original reference. The variable t_1 is the time stamp at which the ZVS is initialized (e.g., $t_1 = 0$). After solving Equation (14), we obtain the magnitudes and time stamps (time when an impulse is applied) of the impulse sequence as follows:

$$A_1 = \frac{e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}}{1 + e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}} \tag{15a}$$

$$A_2 = 1 - A_1 \tag{15b}$$

$$t_2 = t_1 + \frac{\pi}{\omega_0\sqrt{1-\zeta^2}} \tag{15c}$$

Using the same idea and similar procedures, the VTS applies two sequential impulses to modify the original trajectory information of the predecessor vehicle so that the shaped trajectory induces no overshoot/undershoot from the ego vehicle (which ensures its string stability). Correspondingly, by convoluting the impulse sequence with the original position p_{pred} and speed v_{pred} of the predecessor vehicle, the shaped speed and position in VTS can be expressed as:

$$p_{\text{pred}}^{\text{shaped}}(t) = A_1 p_{\text{pred}}(t - t_1) + A_2 p_{\text{pred}}(t - t_2) \tag{16a}$$

$$v_{\text{pred}}^{\text{shaped}}(t) = A_1 v_{\text{pred}}(t - t_1) + A_2 v_{\text{pred}}(t - t_2) \tag{16b}$$

Then, $p_{\text{pred}}^{\text{shaped}}$ and $v_{\text{pred}}^{\text{shaped}}$ will replace the original sensor measurements p_{pred} and v_{pred} in Equation (3) to implement the ACC system. Note that convolution (16) introduces the delay effect, enforcing an ACC to be more dependent on the predecessor vehicle's past trajectories. This can further induce a slower convergence to the desired speed and spacing than by following the original trajectory. The delay effect becomes more obvious with longer impulse sequence, making a shorter impulse sequence more ideal for efficient car-following operations. Moreover, as the VTS assumes ACC parameters have been accurately estimate, two impulses ($n = 2$) are applied to minimize the delay of ACC response and enhance traffic efficiency.

Considering the performance of the VTS can degrade under inaccurately identified parameters, the next subsection addresses this aspect by introducing the RTS.

Robust Trajectory Shaper

In real-world applications, the perfect identification of ω_0 and ζ of an ACC system can

be difficult due to noisy, or even inaccurately measured trajectory information. The VTS implemented using inaccurate ω_0 and ζ will generate an erroneous shaped trajectory, degrading the performance of attenuating speed fluctuations. Thus, robustness should be incorporated into TS to counteract the string-unstable ACC system factoring a range of ω_0 and ζ. Specifically, instead of requiring the residual vibration percentage at a specific ω_0 and ζ to be zero, the RTS adapts the idea of specified insensitivity shaper (Singh and Singhose, 2002; Zhao et al., 2016) to ensure that the residual vibration percentage over a span of ω_0 and ζ (which include the actual ω_0 and ζ corresponding to traffic oscillations) is less than a specified tolerance level.

Considering a longer impulse sequence can trigger more relaxed and smoother response to attain more margin for robustness, the RTS incorporates three impulses ($n = 3$). Then, defining the set of natural frequencies and damping ratios that can induce string-unstable CF behaviour as Ω and Ψ, respectively, we have the following nonlinear program to obtain the RTS (i.e., solving for the magnitudes and time stamps of 3 impulses):

$$\min_{A_1,\ldots,\ A_3,\ t_1,\ldots,\ t_3} t_3 \tag{17a}$$

$$\text{s.t. } t_{i+1} - t_i \geq 0, i = 1, \ldots, 2 \tag{17b}$$

$$A_i \geq 0, i = 1, \ldots, 3 \tag{17c}$$

$$\sum_{i=1}^{3} A_i = 1 \tag{17d}$$

$$V\left(\omega_0^j,\ \zeta^j\right) \leq V_{\text{tol}},\ \omega_0^j \in \Omega,\ \zeta^j \in \Psi \tag{17e}$$

The objective function in Equation (17a) aims to minimize the time stamp of the last impulse so that the delay effect introduced by convolution is minimized. Equation (17b) states that the time stamp of an impulse occurring later should be greater than the time stamp of an impulse occurring earlier. Equation (17c) requires the magnitudes of impulses to be positive, which reduces the variations in the shaped trajectory. Equation

(17d) ensures that the shaped trajectory can maintain the same magnitude as the original trajectory, so that the ACC system can produce a desired CF behaviour with respect to the original trajectory. Equation (17e) bounds the residual vibration percentage to below the tolerance level V_{tol} for all natural frequencies in set Ω and all damping ratios in set Ψ. The impulse sequence of RTS can be efficiently computed using nonlinear solvers. This study applies the "active set" algorithm embedded in MATLAB 'fmincon' function to solve the nonlinear program in Equation (17). Correspondingly, the shaped speed and position of the predecessor vehicle in the RTS can be expressed as:

$$p_{\text{pred}}^{\text{shaped}}(t) = \sum_{i=1}^{3} A_i \, p_{\text{pred}}(t - t_i) \tag{18a}$$

$$v_{\text{pred}}^{\text{shaped}}(t) = \sum_{i=1}^{3} A_i \, v_{\text{pred}}(t - t_i) \tag{18b}$$

The comparison of robustness between VTS and RTS is illustrated in Figure 5, where given a damping ratio ζ, the RTS dampens the speed oscillations (i.e., enforces $V(\omega, \zeta) \leq V_{\text{tol}}$) covering a much wider frequency range compared to the VTS.

Remark 1: The sets of natural frequencies Ω and damping ratios Ψ can be constructed by analysing historical data. Specifically, the variations and bounds of ω_0 and ζ during the ACC operations can be obtained using parameter estimation methods (e.g., EKF discussed in next section) which provide a set of values of ω_0 and ζ that will influence string stability. It is also worth noting that requiring excessively large sets of ω_0 and ζ can overdamp speed fluctuations, jeopardizing ACC tracking performance.

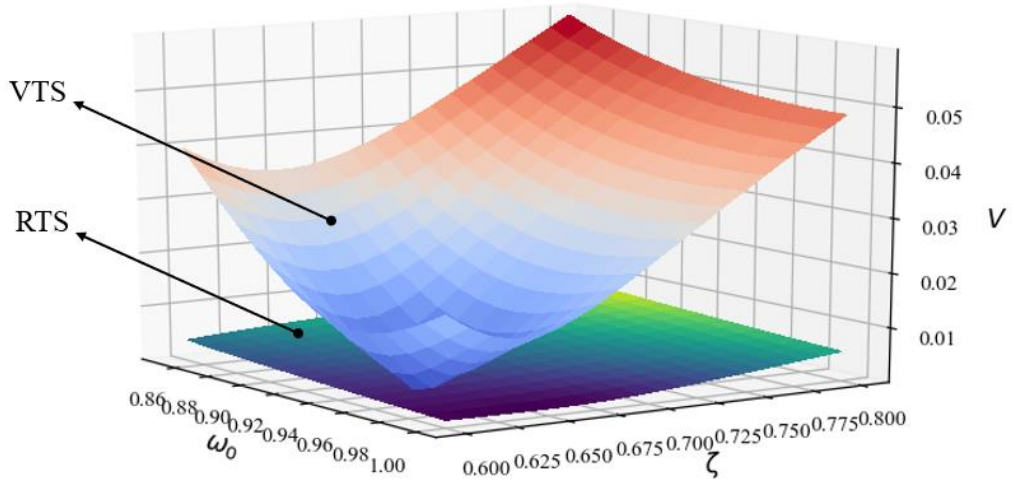


Figure 5. Comparison of VTS and RTS

Parameter Estimation of ACC System

The parameter estimation underpins and automates the real-world implementation of the TS-based method. Though the RTS can handle inaccurately identified parameters, its performance improves with accurately identified parameters (as seen in Figure 4). In this section, we discuss the sufficient condition for obtaining accurately estimated parameters of a CF dynamical system, and introduce the proposed offline-online parameter estimation method to enable real-world implementation of the TS-based method.

To obtain accurate parameters, the CF dynamical system of the ego vehicle needs to possess the observability-identifiability property (i.e., being able to estimate all the vehicle states and parameters given the sensor measurements), while the input signal (i.e., speed of predecessor vehicle) should satisfy the persistent excitation condition (Ljung, 1999) and include desired operating speed and stop-and-go movements (Zhou et al., 2024). Only if this property and the excitation condition are satisfied, the CF operations can fully unveil the CF dynamics of the ego vehicle and ensure that the estimated parameters converge to ground-truth values. Accurately estimated parameters will aid the string stability performance of the TS-based method. Next, we articulate the

observability-identifiability property of a CF dynamical system, based on which we discuss the significance of the predecessor vehicle trajectories in parameter estimation.

Observability-identifiability of a car-following dynamical system

This subsection starts with the formulation of the CF dynamical system. We first define the state vector of the ego vehicle as $x(t) = \left[\Delta p(t), v_{\text{ego}}(t)\right]^{\mathsf{T}}$, and its augmented state vector as $\tilde{x}(t) = [x(t), \theta]^{\mathsf{T}}$, where $\Delta p(t)$ and $v(t)$ denote the spacing with respect to the predecessor vehicle and the speed of the ego vehicle, respectively, and $\theta \in \mathbb{R}^{n_\theta}$ denotes the parameters of the ego vehicle. Then, the CF dynamical system of the ego vehicle can be formulated as follows:

$$\dot{\tilde{x}}(t) = F\big(\tilde{x}(t), u(t)\big) \tag{19a}$$

$$y(t) = g\big(\tilde{x}(t), u(t)\big) \tag{19b}$$

where $F\big(\tilde{x}(t), u(t)\big) = \left[v(t), a(t), 0_{n_\theta}\right]^{\mathsf{T}}$ describes the state evolution of the CF dynamical system, 0_{n_θ} is a n_θ-dimensional zero vector. $u(t)$ is the speed of the predecessor vehicle, which can be interpreted as the input signal of the CF dynamical system. $g\big(\tilde{x}(t), u(t)\big)$ describes the sensor measurement signal $y(t)$ (e.g., position, speed of ego vehicle measured by sensors) based on augmented states $\tilde{x}(t)$ and input signal $u(t)$. A CF dynamical system describes the vehicle state evolution and the sensor measurement information of the ego vehicle during the ACC operation.

Next, based on the formulation of the CF dynamical system, we introduce the definition and sufficient conditions for observability, and connect it to the identifiability of a CF dynamical system to obtain unique parameter estimation results.

Definition 1 (Observability). A system with an initial state x_0 is observable if and only if the value of the initial state can be determined from the system measurement signal $y(t)$ through the time interval $t_0 < t < t_f$ (Åström and Murray, 2010).

Correspondingly, we define the observability matrix $\mathcal{O}(x) \in \mathbb{R}^{n_x \times n_x}$ as follows:

$$\mathcal{O}(x) = \begin{bmatrix} \frac{\partial}{\partial x} y(t) \\ \frac{\partial}{\partial x} \dot{y}(t) \\ \frac{\partial}{\partial x} \ddot{y}(t) \\ \vdots \\ \frac{\partial}{\partial x} y^{n_x - 1}(t) \end{bmatrix} \tag{20}$$

The observability matrix $\mathcal{O}(x)$ links the variations of measurement signal to the states, which will be used to validate the observability in the following theorem.

Theorem 1. A dynamical system is (locally) observable in the neighborhood of x_0 if and only if the observability matrix $\mathcal{O}(x_0)$ has a rank of n_x (Khalil, 2002).

Theorem 1 implies that if a CF dynamical system is observable, all vehicle states can be estimated based on the measurement signal $y(t)$. This is important for the implementation of vehicle control, as not all vehicle states can be directly measured. The observability property enables us to track the variations of all states using only feasible sensor measurements. Next, we introduce the definition of identifiability to obtain a unique estimated parameter $\hat{\theta}$.

Definition 2 (Identifiability). A CF dynamical system given by Equation (19) is locally identifiable if for any θ^* there exists a neighborhood $\mathcal{N}(\theta^*)$ where the following bijection holds (Villaverde et al., 2019):

$$y\left(t, \hat{\theta}\right) = y(t, \theta^*) \iff \hat{\theta} = \theta^* \tag{21}$$

In the differential geometry framework (Villaverde et al., 2019), the observability-identifiability is an augmented observability property, where the augmented state vector $\tilde{x} = [x, \theta]^\intercal$ replaces the state vector x in Equation (20) and uses $\tilde{x}$ to compute the observability-identifiability matrix $\mathcal{O}(\tilde{x})$. The following theorem articulates the condition

for ensuring the observability-identifiability property, based on which all the states and parameters can be obtained using the measurement signal $y(t)$.

Theorem 2. A dynamical system is locally observable and locally identifiable in the neighborhood of $\tilde{x}_0$, if the observability-identifiability matrix $\mathcal{O}(\tilde{x}) = \begin{bmatrix} \frac{\partial}{\partial \tilde{x}} y(t) \\ \frac{\partial}{\partial \tilde{x}} \dot{y}(t) \\ \frac{\partial}{\partial \tilde{x}} \ddot{y}(t) \\ \vdots \\ \frac{\partial}{\partial \tilde{x}} y^{n_{\tilde{x}}-1}(t) \end{bmatrix}$ has a rank of $n_x + n_\theta$ at $\tilde{x}_0$ (Wang et al., 2022).

In addition, with an input signal $u(t)$ (e.g., speed of predecessor vehicle) in CF dynamical systems, the condition for observability-identifiability property is as follows:

Theorem 3. A dynamical system is locally observable-identifiable in a neighborhood of $\tilde{x}_0$, if the observability-identifiability matrix $\mathcal{O}(\tilde{x}) = \begin{bmatrix} \frac{\partial}{\partial \tilde{x}} g\big(\tilde{x}(t), u(t)\big) \\ \frac{\partial}{\partial \tilde{x}} \Big(L_f g\big(\tilde{x}(t), u(t)\big)\Big) \\ \frac{\partial}{\partial \tilde{x}} \Big(L_f^2 g\big(\tilde{x}(t), u(t)\big)\Big) \\ \vdots \\ \frac{\partial}{\partial \tilde{x}} \Big(L_f^{n_{\tilde{x}}-1} g\big(\tilde{x}(t), u(t)\big)\Big) \end{bmatrix}$ has a rank of $n_x + n_\theta$ (Villaverde et al., 2019).

Note that $L_f^i g\big(\tilde{x}(t), u(t)\big) = \frac{\partial L_f^{i-1} g\left(\tilde{x}(t), u(t)\right)}{\partial \tilde{x}} f\big(\tilde{x}(t), u(t)\big) + \sum_{j=0}^{\infty} \frac{\partial L_f^{i-1} g\left(\tilde{x}(t), u(t)\right)}{\partial u} u(t)^{(j+1)}$, and $L_f g\big(\tilde{x}(t), u(t)\big) = \frac{\partial g\left(\tilde{x}(t), u(t)\right)}{\partial \tilde{x}} f\big(\tilde{x}(t), u(t)\big) + \sum_{j=0}^{\infty} \frac{\partial g\left(\tilde{x}(t), u(t)\right)}{\partial u} u(t)^{(j+1)}$ are the Lie-derivatives corresponding to the augmented states of the ego vehicle. The Lie-derivatives describe how the variation in measurement signal depends on the variations in augmented state $\tilde{x}(t)$ and input signal $u(t)$, which can be used to investigate whether the variations in augmented state $\tilde{x}(t)$ can be interpreted from the variation in measurement signal. The property of observability-identifiability of a CF system indicates the feasibility of obtaining all vehicle states and CF parameters using

available sensor measurements. This is important for the TS-based method, as we need the estimated CF parameters based on sensor measurements to design the TS in real-world operations.

With speed as the measurement information, an ACC system based on CTH-linear CF dynamical system is locally observable-identifiable except at the CF equilibrium $(v_0 h, v_0)$, where v_0 is the desired operating speed. This elaborates that the trajectory corresponding to equilibrium driving cannot ensure accurate parameter estimation, which leads to the following analysis and requirements on the trajectory of the predecessor vehicle to enable observability-identifiability.

Persistent excitation condition of trajectory data

In addition to the observability-identifiability property of the CF dynamical system, the trajectory of the predecessor vehicle also plays an important role in parameter estimation. In this subsection, we first introduce the concept of persistent excitation, and then relate it to the trajectory of the predecessor vehicle for parameter estimation. The persistent excitation condition indicates that an input signal (e.g., the trajectory of the predecessor vehicle) used for parameter estimation is informative enough (e.g., displays sufficient speed fluctuations to trigger the ego vehicle to produce trajectories that can unleash all the CF characteristics) to obtain a unique and optimal estimation solution. It is formulated through the following definition.

Definition 3. A signal is persistently exciting of order n if the following limit exists:

$$R_u(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}[u(t)u(t-\tau)] \tag{22}$$

and the following matrix C_n is positive definite (Ljung, 1999).

$$C_n = \begin{bmatrix} R_u(0) & R_u(1) & \cdots & R_u(n-1) \\ R_u(-1) & R_u(0) & \cdots & R_u(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_u(1-n) & R_u(2-n) & \cdots & R_u(0) \end{bmatrix} \tag{23}$$

In particular, the condition of persistent excitation in Equations (22) and (23) is equivalent to the following theorem in frequency domain.

Theorem 4. In the frequency domain, a signal satisfying persistent excitation of order n has a power spectrum with nonzero values at n frequencies in the interval $(-\pi, \pi)$ (Ljung, 1999).

The property of persistent excitation illustrates the capability of an input signal to obtain unique and accurate parameters through the parameter estimation process. Based on this, the following theorem states the conditions for identifying (i.e., obtaining accurate parameters) a linear dynamical system (e.g., CF dynamical system which uses linear CF model to describe the state evolution).

Theorem 5. To identify an nth order linear dynamical system, the input signal needs to be at least persistently exciting of order $2n$ (Ljung, 1999).

As a linear CF dynamical system is a second-order linear dynamical system, theorems 4 and 5 suggest that the trajectory of the predecessor vehicle is persistently exciting of at least order 4 (i.e., having a power spectrum with at least 4 nonzero values in the interval $(-\pi, \pi)$).

Note that the analysis so far in this subsection is only sufficient for parameter estimation of linear systems, which cannot address characteristics of nonlinear CF dynamical systems (e.g., the state evolution is governed by intelligent driver model, optimal velocity model, Gipps' model, etc.). Hence, we identify two additional essential characteristics of the predecessor vehicle trajectory to enable accurate parameter estimation for nonlinear CF dynamical systems; they are: (i) stop-and-go movements, and (ii) operations at the desired maximum speed (e.g., speed limit). This is because the

nonlinearity causes significant impacts when a vehicle operates in stop-and-go movements and at the speed limit; specifically, they unveil certain safety-related characteristics in nonlinear CF dynamical systems that cannot be fully uncovered through trajectories satisfying only theorem 5.

The following proposition describes the sufficient conditions to accurately identify the parameters of a nonlinear CF dynamical system.

Proposition 1. To accurately identify the parameters of a CF dynamical system, the trajectory of the predecessor vehicle needs to satisfy the following conditions:

(i) the CF dynamical system is locally observable-identifiable;

(ii) the trajectory of the predecessor vehicle is persistently exciting of at least order four;

(iii) the trajectory of the predecessor vehicle contains scenarios of stop-and-go movements and operations at the speed limit.

Note that the conditions in Proposition 1 may be difficult to realize in real-world driving scenarios, especially under insufficient data. Hence, next, we propose an offline-online parameter estimation method to alleviate the negative impact of the predecessor vehicle trajectory on estimated parameters, and correspondingly guarantee desired performance of the TS-based method.

Offline-online parameter estimation method

The proposed offline-online parameter estimation method is illustrated in Figure 6. The parameter estimation starts from offline parameter estimation (as shown in the red solid box at the left bottom of Figure 6), where the batch-optimisation method uses a CF model (e.g., CTH-linear model in this study) and the historical trajectories of ego vehicle and its predecessor vehicle to estimate parameters $\hat{\theta}_0$. These estimated parameters $\hat{\theta}_0$ are used

as initial conditions (baseline) to initialize the online parameter estimation. Based on $\hat{\theta}_0$, an extended Kalman filter (EKF) uses the real-time trajectory measurements of the ego vehicle and its predecessor vehicle to perform the online parameter estimation iteratively (as shown in the blue dashed box in Figure 6). The trajectory used in the online parameter estimation is a discrete time series (measured by onboard sensors), and the online parameter estimation is performed at each time step t of the trajectory (i.e., each data point). In the online parameter estimation, we first specify a time interval T (i.e., a specific number of time steps) for updating the estimated parameters. This is because the estimated parameters will be used to update TS, and frequently updating the TS can reduce the control performance (e.g., by increasing speed fluctuations). In addition, after every time interval T (note that in Figure 6, $t\%T$ computes the residue of t/T), the newly estimated parameters $\hat{\theta}$ are also compared to the estimated parameters $\hat{\theta}_0$ in the previous EKF iteration. If the difference $\|\hat{\theta}_0 - \hat{\theta}\|$ is greater than a threshold ϵ, then the estimated parameter $\hat{\theta}$ will be used to update the TS, and $\hat{\theta}_0$ is replaced by $\hat{\theta}$. Otherwise, TS will not be updated, and $\hat{\theta}$ will be assigned to $\hat{\theta}_0$ for the next EKF iteration. This operation also aims to avoid frequent updates of TS and ensures smooth ACC operations. The TS is then implemented on the ACC system for operations. The vehicle trajectories collected from ACC operations are used in future EKF iterations.

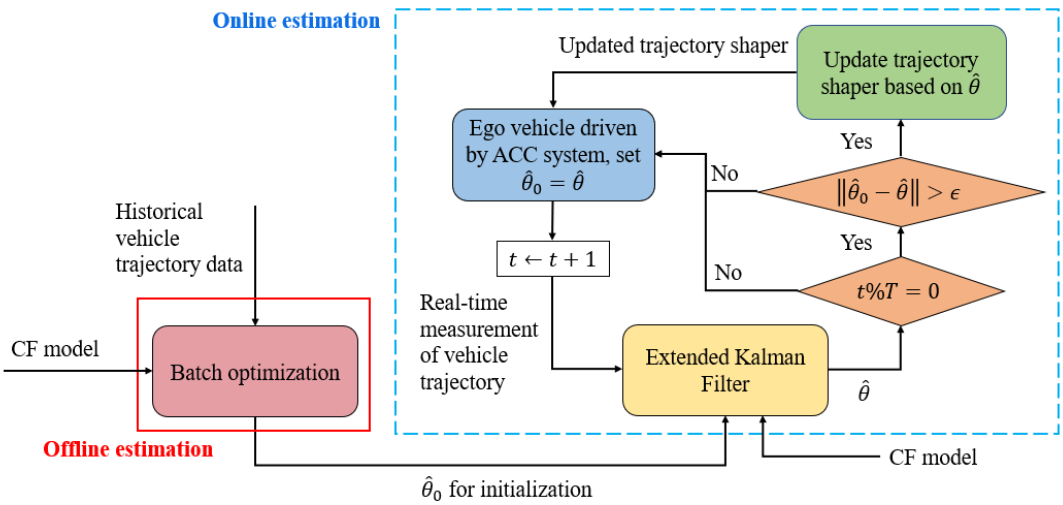


Figure 6. Offline-online parameter estimation

Note that a set of CF models can be parallelly applied in the offline-online parameter estimation process, so that the estimated parameters corresponding to the best accuracy (i.e., RMSE between the estimated and ground-truth trajectories is the smallest) can be selected for the TS-based method. Next, we describe the details of offline and online parameter estimation processes.

Offline parameter estimation

The batch-optimisation method aims to optimise the CF parameters $\hat{\theta}$ to minimize the root mean square error (RMSE) between the simulated spacing $s(\hat{\theta}, t)$ and the ground-truth spacing $\bar{s}(t)$ in the data. The RMSE of spacing is applied as the objective due to its capability for achieving high accuracy in the CF model calibration (Kesting and Treiber, 2008). The optimisation problem is formulated as:

$$\min_{\hat{\theta}} \sqrt{\frac{1}{\mathrm{T_{hist}}} \int_0^{\mathrm{T_{hist}}} \left(s(t) - \bar{s}(t)\right)^2 \mathrm{dt}} \tag{24a}$$

$$\text{s.t. } \dot{s}(t) = u(t) - v_{ego}\left(x, \hat{\theta}, t\right) \tag{24b}$$

$$\dot{v}_{ego}\left(x, \hat{\theta}, t\right) = f\left(x, \hat{\theta}, t\right) \tag{24c}$$

where $\bar{s}(t)$ is the ground-truth spacing data, and $u(t)$ is the speed of the predecessor vehicle. $\mathrm{T_{hist}}$ is the total length of the trajectory in the dataset. Equation (24b) states that the dynamics of spacing are equal to the speed difference between the ego vehicle and its predecessor. Equation (24c) indicates that the acceleration of the ego vehicle is factorized by a function f which can be any CF model (e.g., CTH-linear model, model predictive control, deep reinforcement learning-based method, etc.). Thus, the offline parameter estimation is an optimisation problem constrained by a CF dynamical system in Equations (24b) and (24c). Note that Equation (24a) can be nonlinear and nonconvex, which is

solved using the "interior point" algorithm many times with random initial values to improve solution optimality (Wang et al., 2022).

The offline parameter estimation uses the historical trajectories to compute the estimated parameter $\hat{\theta}_0$. $\hat{\theta}_0$ may be inaccurate for estimating the CF behaviour of an ACC system in real-world operations, because: (i) the historical trajectories may not satisfy Proposition 1; (ii) the batch-optimisation method may provide suboptimal solution which is not accurate; and (iii) the CF behaviour of an ACC system may vary during real-world operation (e.g., drivers may switch control modes in different traffic conditions), which cannot be accurately estimated using a fixed $\hat{\theta}_0$. Thus, online parameter estimation, which consistently updates the estimated parameters of an ACC system, is proposed next to improve the accuracy of estimated parameters during ACC operation.

Online parameter estimation

The Extended Kalman Filter (EKF) is applied to enable the online parameter estimation using real-time measurements. The EKF aims to minimize the trace of the error covariance matrix at each time step of the trajectory so that the uncertainty in parameter estimation is minimized, as follows:

$$\min_{\hat{\theta}} \operatorname{trace}\left(\mathbb{E}\left[(\theta-\hat{\theta})(\theta-\hat{\theta})^{\mathrm{T}}\right]\right) \tag{25}$$

To implement the EKF, we discretize the CF dynamical system with time step dt, and the corresponding dynamics of vehicle states and parameters at time step k are as follows:

$$\tilde{x}_k = \begin{bmatrix} p_k \\ v_k \\ \theta_k \end{bmatrix} = F(x_{k-1}, u_{k-1}, \theta_{k-1})$$

$$= \begin{bmatrix} x_{k-1} + v_{k-1}dt + 0.5a_k dt^2 + w_{k-1,1} \\ v_{k-1} + a_k dt + w_{k-1,2} \\ I_{n_\theta}\theta_{k-1} + w_{k-1,3} \end{bmatrix} \tag{26a}$$

$$y_k = g_k(x_k, u_k, \theta_k) = v_k + n_k \tag{26b}$$

where $w_{k-1,1}$ and $w_{k-1,2}$ are the uncertainties in position and speed dynamics, respectively. $w_{k-1,3}$ is the process noise in the evolution of the CF dynamical system. $a_k = f(x_{k-1}, u_{k-1}, \theta_{k-1})$ describes the acceleration variations of the ego vehicle.

For parameter estimation, if all vehicle states (i.e., speed, acceleration) are measured, then the focus is only on the evolution of parameters θ_k of the CF dynamical system. We define W and R as the covariance matrices of the disturbance in the parameter dynamics (i.e., $w_{k,3}$) and the measurement noise (i.e., n_k), respectively. The EKF for parameter estimation is illustrated in the following algorithm. The EKF starts by initializing the initial state x_0 (the vehicle state at the time step when EKF is activated), the initial value of estimated parameter $\hat{\theta}_0$ (obtained from the offline batch optimisation), and the error covariance matrix P_0 (the variance of parameter estimation error). Then, the EKF is executed for E_P iterations (E_P is also the total number of time steps of the trajectory data).

As shown in Algorithm 1, the iterative process starts with the prediction step where we assume the parameters are fixed during the current iteration to obtain the predicted parameter $\hat{\theta}_{k(-)}$ (as shown in line 4 of Algorithm 1). This mitigates abrupt changes in predicted parameters and improves accuracy. Next, the prediction error covariance matrix of $\hat{\theta}_{k(-)}$ is computed in line 5 of Algorithm 1, where $\phi_{k-1} = \frac{\partial F(x_{k-1}, u_{k-1}, \theta_{k-1})}{\partial \theta_{k-1}}\Big|_{\theta_{k-1}=\hat{\theta}_{k-1}} = I_{n_\theta}$ is an n_θ-dimensional identity matrix which describes

how $\hat{\theta}_{k(-)}$ vary with the estimated parameters $\hat{\theta}_{k-1}$ from the previous time step. $J_w = \frac{\partial F(x_{k-1},u_{k-1},\theta_{k-1})}{\partial w_{k-1}}|_{\theta_{k-1}=\hat{\theta}_{k-1}} = \mathbb{I}_{n_\theta}$ is a n_θ-dimensional unit vector which describes how predicted parameters are impacted by w_{k-1}. $P_{k(-)}$ indicates the inaccuracy of $\hat{\theta}_{k(-)}$, which is used to improve the estimation accuracy in the following steps. In line 6 of Algorithm 1, we minimize the trace of the prediction error covariance matrix (Equation (25)) to obtain the Kalman gain, where $H_k = \frac{\partial g_k(x_k,u_k,\theta_k)}{\partial \theta_k}|_{\theta_k=\hat{\theta}_{k(-)}}$ is a vector which shows how $g_k(x_k, u_k, \theta_k)$ is influenced by the predicted parameter $\hat{\theta}_{k(-)}$. Then, the Kalman gain is used to update the estimated parameter $\hat{\theta}_k$ in line 7 of Algorithm 1, where we correct the error between real-time measurement information y_k (obtained through onboard sensors) and the estimated measurement information computed by $g_k\left(x_k, u_k, \hat{\theta}_{k(-)}\right)$ (following Equation (26b),). The last step is to compute the estimation error covariance matrix P_k (in line 8 of Algorithm 1), where $J_n = \frac{\partial g(x_k,u_k,\theta_k)}{\partial n_k}|_{\theta_k=\hat{\theta}_{k(-)}} = 1$. P_k indicates the uncertainty of the estimated parameters and is used in the next iteration to further improve the estimation accuracy.

Algorithm 1: EKF for Online Parameter Estimation

1: **Input:** E_P, x_k, y_k, u_k, $g(x_k, u_k, \theta_k)$, W, R, ϕ_k, H_k, J_w, J_n

2: **Initialize** x_0, $\hat{\theta}_0$, P_0, $k = 0$

3: **for** $k = 1{:}\, E_\mathrm{P}$ **do**

4: Predicted parameters: $\hat{\theta}_{k(-)} = I_{n_\theta}\hat{\theta}_{k-1}$

5: Prediction error covariance matrix: $P_{k(-)} = \phi_{k-1}P_{k-1}\phi_{k-1}^\intercal + J_w W J_w^\intercal$

6: Kalman gain: $K_k = P_{k(-)}H_k^\intercal\left(H_k P_{k(-)}H_k^\intercal + J_n R J_n^\intercal\right)^{-1}$

7: Updated estimation of parameters: $\hat{\theta}_k = \hat{\theta}_{k(-)} + K_k \left(y_k - g_k\left(x_k, u_k, \hat{\theta}_{k(-)}\right)\right)$

8: Estimation error covariance matrix: $P_k = P_{k(-)} - P_{k(-)}H_k^{\mathrm{T}}\left(H_k P_{k(-)} H_k^{\mathrm{T}} + J_n R J_n^{\mathrm{T}}\right)^{-1} H_k P_{k(-)}$

9: **end for**

Remark 3: The EKF computes the estimated parameter at each time step of the vehicle trajectory, which can capture the time-varying CF behaviour of an ACC system and significantly improve the estimation accuracy. The accurately estimated parameters then enable reliable retrofitted CF behaviour of an ACC, which further ensures TS-based method can appropriately modify the sensor measurements and enhances safety. In addition, when implementing the TS-based method in the real world, frequently updating the TS based on the estimated parameters at each time step can induce additional speed fluctuations and compromise the control performance. Thus, recalling the discussion from Figure 6, we include the parameter updating time interval T and a threshold ϵ to avoid unnecessary updates of TS in real-world operations.

Numerical experiments

Parameter estimation performance

We first explore the sufficient conditions for obtaining accurate parameters, and then illustrate the performance of the proposed offline-online parameter estimation method. The parameter estimation method applied here is the EKF in Algorithm 1. The parameter setting of EKF is as follows: $W = 0.01 I_{n_\theta}$, $R = 0.01$, and $P_0 = I_{n_\theta}$.

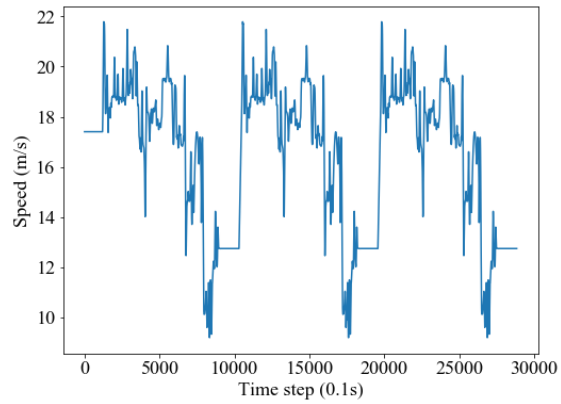
Validation of sufficient conditions for accurate parameter estimation

First, a vehicle trajectory obtained from the processed NGSIM dataset (Montanino and Punzo, 2015) is applied to the predecessor vehicle to perform parameter estimation of the ego vehicle driven by the CTH-linear model with predetermined parameters. The trajectory from this dataset is repeated three times (as shown in Figure 7(a)) to provide more data points for parameter estimation. Figure 7(b) shows that the NGSIM trajectory only has 2 distinct peaks (we neglect the peaks with extremely small values, as they are induced by numerical errors and will not influence the performance of parameter estimation) in the frequency range $[-\pi, \pi]$, which indicates that this trajectory is persistently exciting of order 2. Thus, it is not sufficient to accurately identify the parameters of the CTH-linear model according to Theorem 4, because a CF dynamical system is a second-order dynamical system and requires the trajectory of the predecessor vehicle to be at least persistently exciting of order 4. This is validated in Figure 8(a), where the estimated parameters diverge from the ground-truth values with large error covariances (red shaded areas).

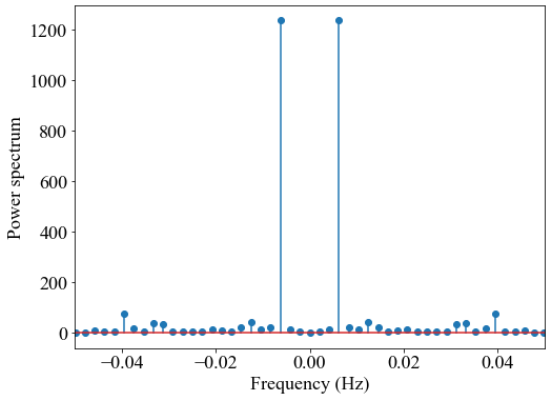
Next, a summation of sinusoids-type speed profile is considered, which sums up 4 sinusoidal signals with different angular frequencies and phase shifts: $v_{\mathrm{pred}}(t) = 17 + 6\sin\big(0.1(t-15)\big) + 5\sin\big(0.07(t-10)\big) - 2.5\sin(0.05t) + 5\sin\big(0.01(t+10)\big)$, as shown in Figure 7(c). Correspondingly, Figure 7(d) shows that this type of trajectory has 8 distinct peaks (again, neglecting the peaks with extremely small values) in the frequency range $[-\pi, \pi]$, which indicates this summation of sinusoids-type speed profile is persistently exciting of order 4. Correspondingly, it is sufficient to accurately identify the parameters of the CTH-linear model (i.e., Figure 8(b) illustrates that all parameters converge to the ground-truth values).

The summation of sinusoids-type trajectory is applied to estimate the parameters of the nonlinear intelligent driver's model (IDM) (Treiber, Hennecke, and Helbing, 2000) with predetermined parameters. As shown in Figure 9(a), the estimated parameters fail to converge to the ground-truth values steadily, because the nonlinearity of IDM requires more properties of the trajectory of the predecessor vehicle to be factored. Thus, we include the characteristics of: (i) reaching to maximum operating speed, and (ii) stop-and-go movements in $v_{\mathrm{pred}}(t)$, to uncover the nonlinearity of the IDM during the parameter estimation process. Correspondingly, Figure 9(b) illustrates that with these augmented features in the trajectory of the processor vehicle, the estimated parameters converge to the ground-truth values. This validates the sufficient conditions stated in Proposition 1 for obtaining accurate parameters.

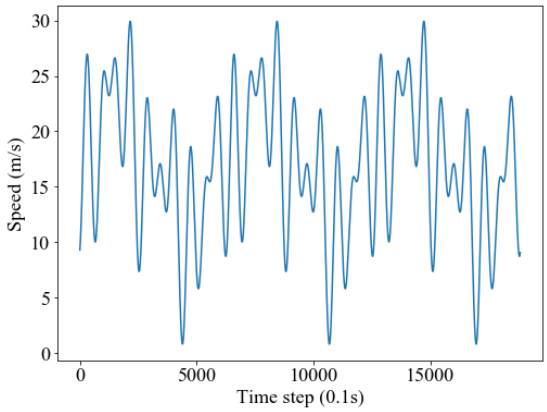
Note that the repeated patterns in Figures 7(a) and 7(c) do not indicate three cycles of identical trajectory are required when collecting real-world trajectories. They simply indicate the reuse of a segment of historical trajectory data three times, which aims to show that the predecessor vehicle trajectory with insufficient persistent excitation order will not lead to convergence when fitting the CTH-linear model, even with abundant data points (e.g., the lead trajectory in Figure 7(a) with three repetitions). The lead trajectory with a persistently exciting order of no less than four will guarantee convergence (e.g., the lead trajectory in Figure 7(c) enables convergence in less than one cycle). In real-world operations, if convergence is not reached, we can reuse the most recent trajectory data satisfying Proposition 1 to facilitate convergence. Meanwhile, online streaming trajectory data should also be consistently incorporated in parameter estimation to capture the time-varying CF characteristics.



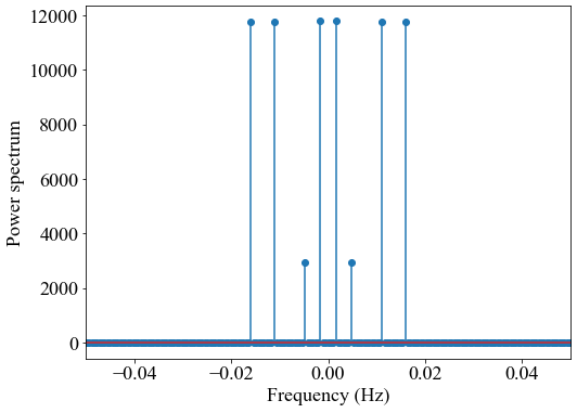
(a) NGSIM speed profile



(b) Power spectrum of NGSIM speed

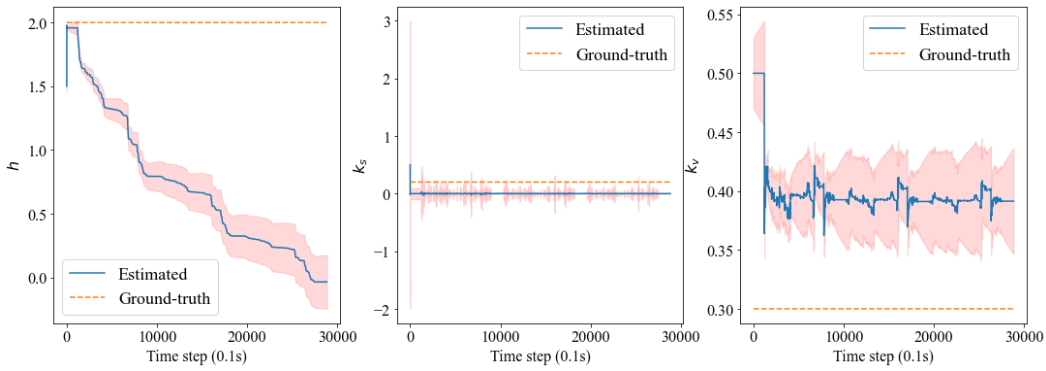


(c) Summation of sinusoids-type speed

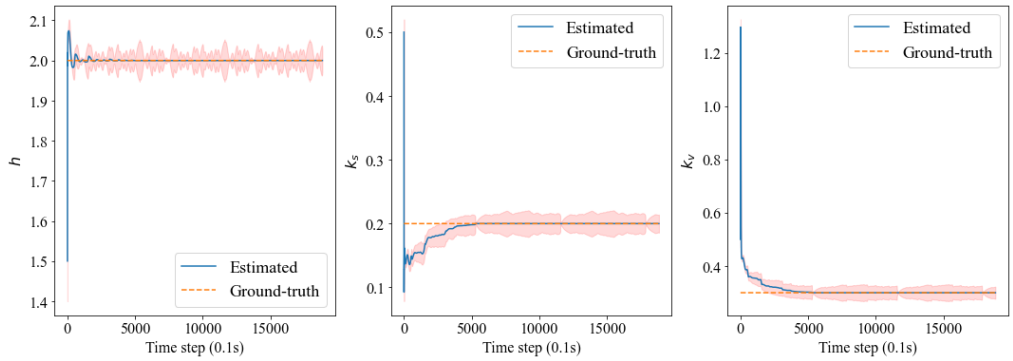


(d) Power spectrum of summation of sinusoids

Figure 7. Trajectories of predecessor vehicle and corresponding power spectrums

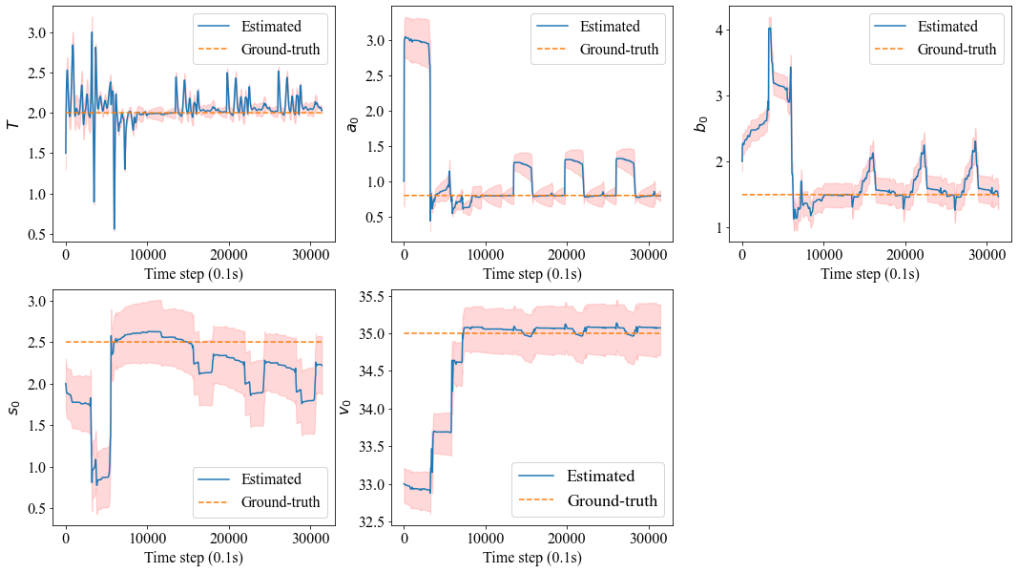


(a) The trajectory of predecessor vehicle does not satisfy the PE condition

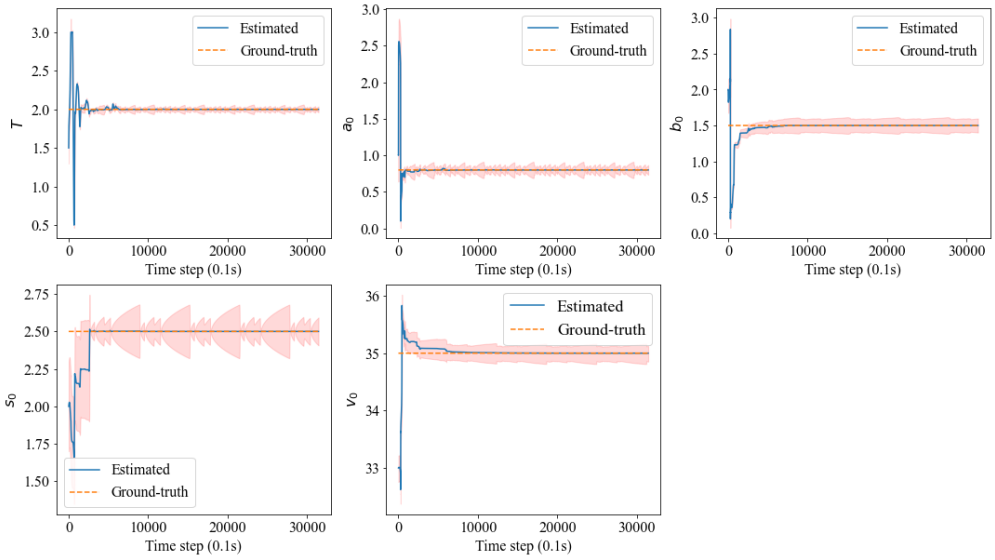


(b) The trajectory of predecessor satisfies the PE condition

Figure 8. Parameter estimation of CTH-linear model



(a) The trajectory of predecessor vehicle satisfies the PE condition but does not include stop-and-go movements and maximum operating speed



(b) The trajectory of predecessor vehicle satisfies the PE condition and includes stop-and-go movements and maximum operating speed

Figure 9. Parameter estimation of the IDM

Performance of offline-online parameter estimation method

This experiment uses real-world driving data to demonstrate the performance of the proposed offline-online parameter estimation method. CF trajectory data from a commercially available electric vehicle, collected on the State Highway 113 near Cartersville, GA, is used here. The ego vehicle (i.e., the commercially available vehicle) follows a human-driven predecessor vehicle (leader) which manually creates speed fluctuations to simulate the stop-and-go traffic condition. The ego vehicle (follower) is driven by the commercial ACC system. As shown in Figure 10, the ego vehicle experiences multiple acceleration/deceleration movements on the highway (i.e., green indicates high speeds, yellow indicates medium speeds, and red indicates complete stops).

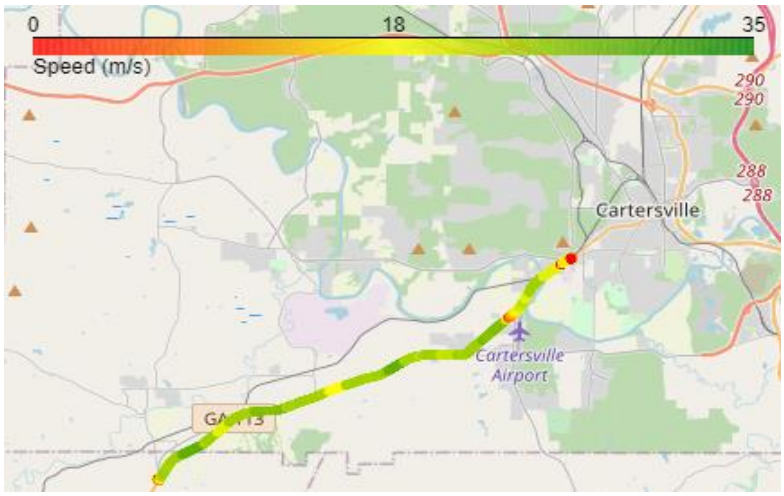


Figure 10. Car-following experiment

In the proposed offline-online parameter estimation method, the offline parameter estimation is devised using the batch-optimisation method on a 10-min trajectory. Then, the online parameter estimation is implemented on another 537-second trajectory. We set $T = 10s$, and $\epsilon = 0.05$. We then compare the proposed offline-online parameter

estimation method with the commonly used offline parameter estimation method (i.e., batch-optimisation method), and the EKF-based online parameter estimation (which starts with a random initial value and yields the estimated parameters at each time step), using the 537-second trajectory as the real-time measurement information. This 537-second trajectory will then be used for testing the performance of VTS and RTS.

Table 1 lists the RMSEs of the proposed offline-online parameter estimation method, the offline batch-optimisation method, and the EKF-based online parameter estimation. The offline-online parameter estimation method achieves the smallest RMSE among the three methods, indicating a capability for enabling the desired accuracy. In addition, Figure 11 illustrates that the speed of the estimated follower (blue dash curve, computed using CTH-linear model with estimated parameters) overlaps with that of the ground-truth follower (light blue solid curve, obtained from experimental data), which further reinforces the effectiveness of the offline-online parameter estimation method. Figure 10 also shows that the commercial ACC system is string-unstable, as it amplifies the speed fluctuations from its predecessor vehicle.

Table 1. Comparison of different parameter estimation methods

Parameter estimation method	Offline-online method	Batch-optimisation	EKF
RMSE	0.0318m/s	0.5155m/s	0.3071m/s

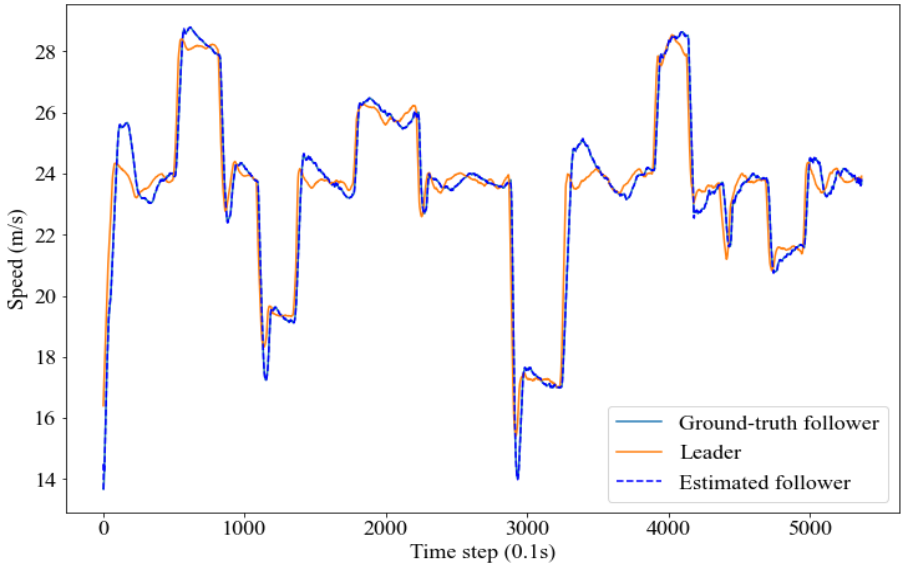


Figure 11. Vehicle speed profiles

Next, three CF models are compared in the implementation of the offline-online parameter estimation method: IDM, optimal velocity model (OVM) (Nakano, 1999) and CTH-linear model. Table 2 illustrates the performance of parameter estimation based on the different CF models. The CTH-linear model achieves the highest accuracy, followed by IDM, and then OVM. This is due to the potential linear CF behaviour of the commercial ACC system, which makes CTH-linear a more accurate behaviour estimator compared to the nonlinear OVM and IDM. In addition, the nonlinearity of OVM and IDM can create more complex non-convexity in model calibration, which yields a locally optimal solution with low accuracy. Correspondingly, more trajectory data and advanced optimisation algorithms should be incorporated to calibrate nonlinear models to achieve higher accuracy. Thus, the CTH-linear is selected to describe the CF behaviour of the commercial ACC system to determine its natural frequency and damping ratio. Next, the estimated CTH-linear model of the commercial ACC system is used to validate the performance of VTS and RTS.

Table 2. Comparison of different CF models in retrofitting commercial ACC system

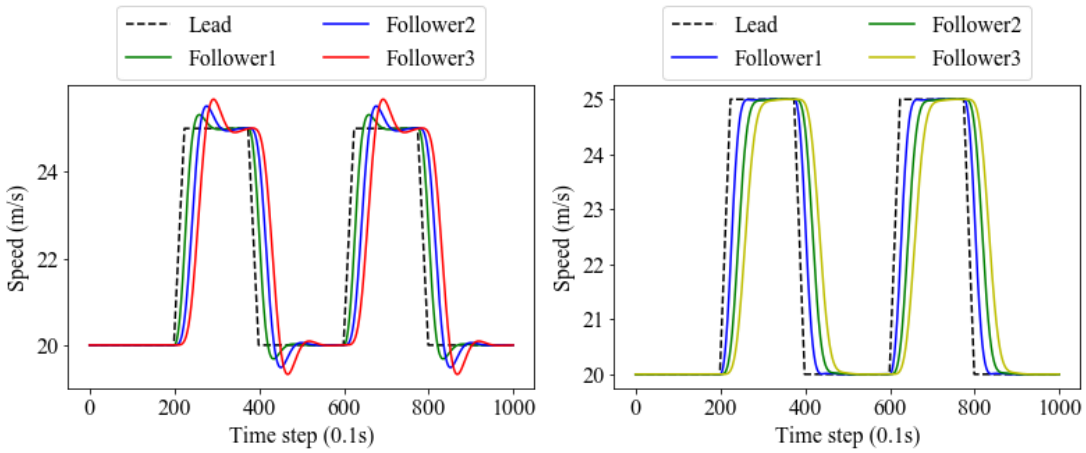
CF model	CTH-linear	IDM	OVM
RMSE	0.0318m/s	0.3218m/s	0.3296m/s

CF performance of TS-based method

Comparison of VTS and RTS using trapezoidal-type trajectory

Here, a four-vehicle platoon is used to analyse the performance of the TS-based method. The predecessor vehicle conducts a trapezoidal-type speed profile to simulate a typical congested traffic environment with abrupt acceleration ($2m/s^2$) and deceleration ($-2m/s^2$). Three followers are controlled using a string-unstable CTH-linear model (by setting parameters in the string-unstable region in Figure 3).

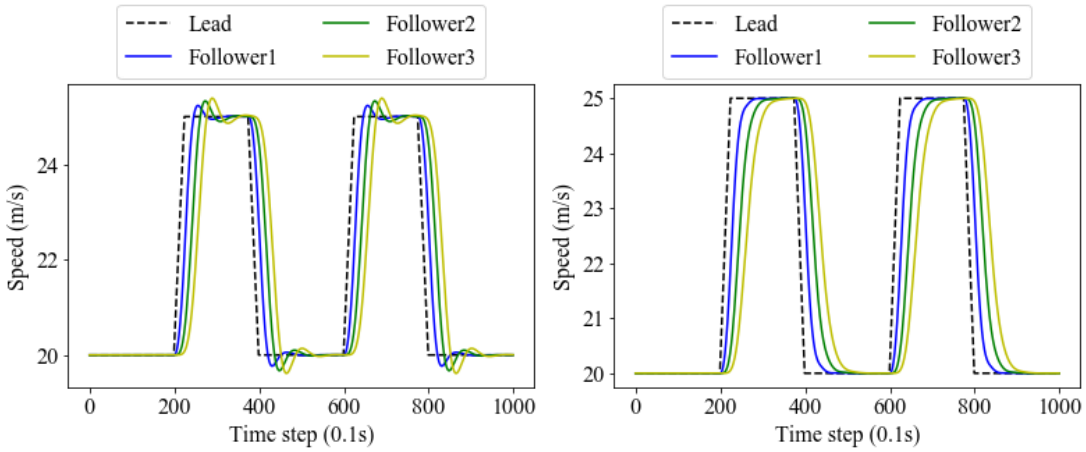
As shown in Figure 12(a), the speed fluctuations are enlarged upstream in the platoon due to the string instability of CTH-linear model. In this case, if ω_0 and ζ are accurately identified, the VTS can effectively mitigate the string instability of the three followers, as shown in Figure 12(b) where the followers smoothly converge to the leader speed levels without enlarging fluctuations.



(a) String-unstable CTH-linear model (b) String-unstable CTH-linear model with VTS

Figure 12. Performance of VTS with accurate parameters in time-based cycles

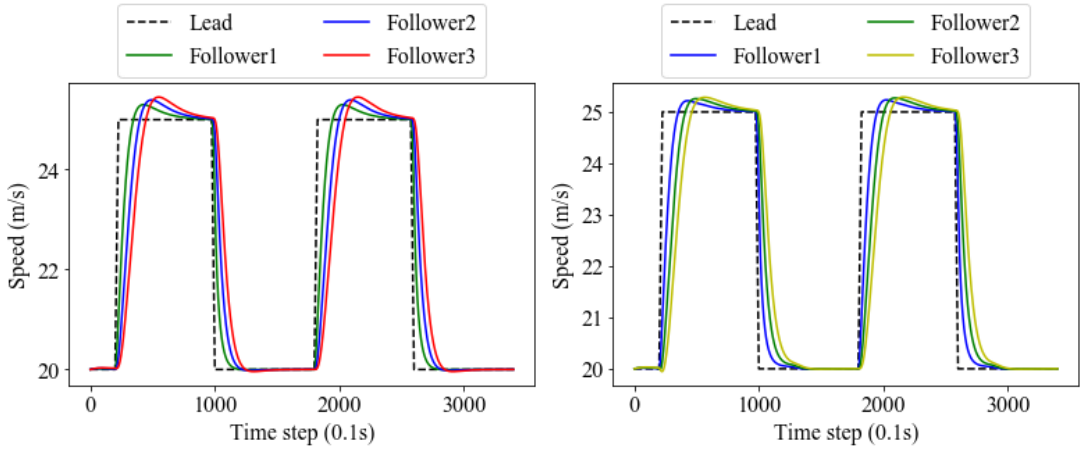
If ω_0 and ζ are not accurately identified, the VTS cannot achieve the desired performance in dampening the speed fluctuations (note the overshoots/undershoots in Figure 13(a)). By contrast, the extra robustness in the formulation of RTS enables it to effectively dampen the speed fluctuation even with inaccurate parameters (as shown in the smooth convergence in Figure 13(b)).



(a) String-unstable CTH-linear model with VTS (b) String-unstable CTH-linear model with RTS

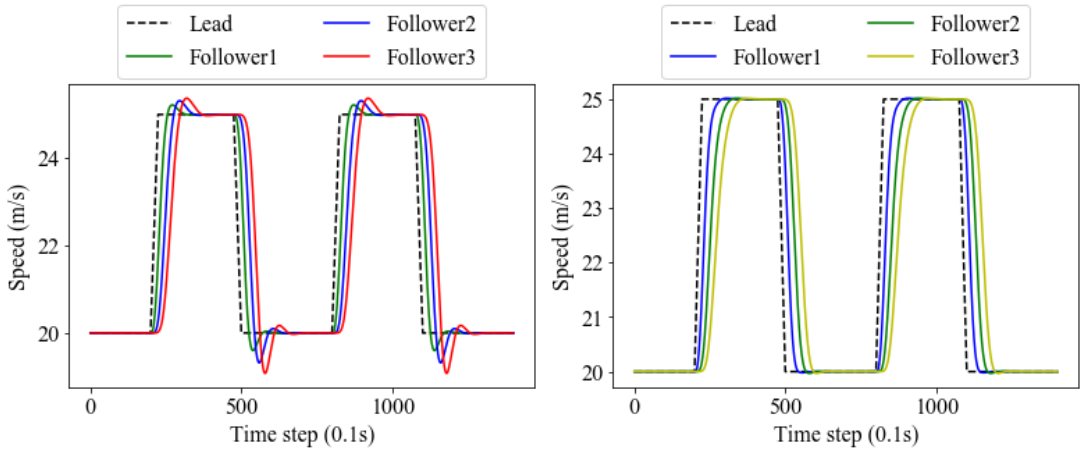
Figure 13. Comparison of VTS and RTS with inaccurate parameters

The nonlinearities arising in an ACC system can also limit the effectiveness of the VTS. Figures 14(a)-(d) illustrate that the VTS fails to mitigate the overshoots/undershoots from a string-unstable IDM, while effectively attenuating those from a string-unstable OVM. This is because the strong nonlinearity of IDM makes the fixed values of ω_0 and ζ insufficient to characterize its dynamics (time-varying values may be necessary to describe the IDM), but OVM's mild nonlinearity enables the fixed values of ω_0 and ζ to capture its dynamics. By contrast, as the RTS factors uncertainty, it can address the time-varying parameters induced by the nonlinearity in an ACC system. As illustrated in Figure 14(e), the RTS dampens undesired overshoots/undershoots from the string-unstable IDM.



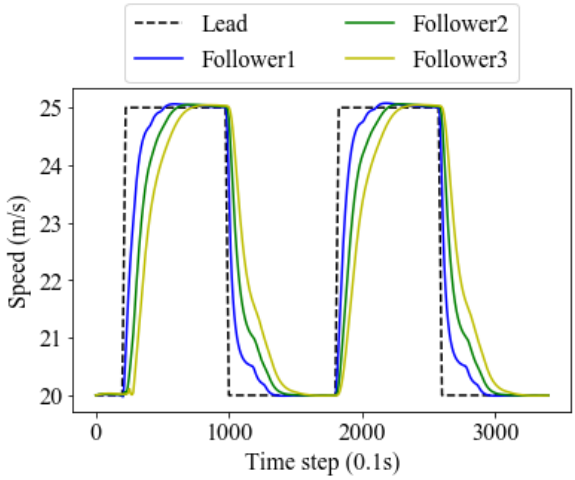
(a) String-unstable IDM

(b) String-unstable IDM with VTS



(c) String-unstable OVM

(d) String-unstable OVM with VTS

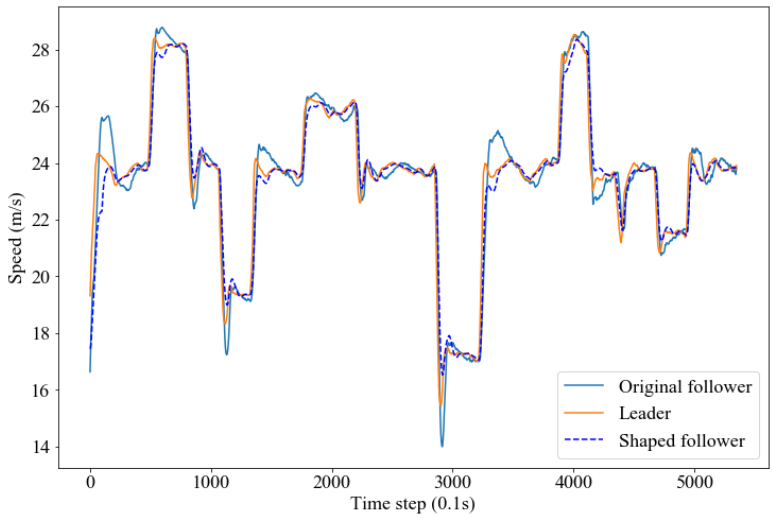


(e) String-unstable IDM with RTS

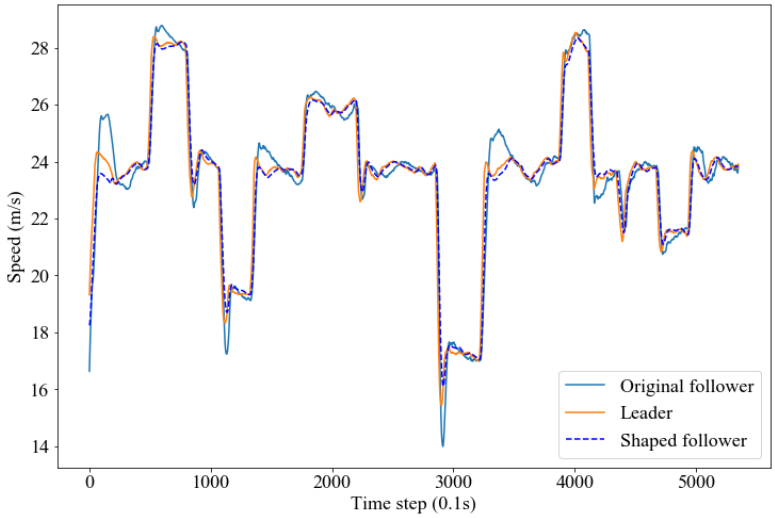
Figure 14. Comparison of VTS/RTS under IDM and OVM

Comparison of VTS and RTS using real-world driving trajectory

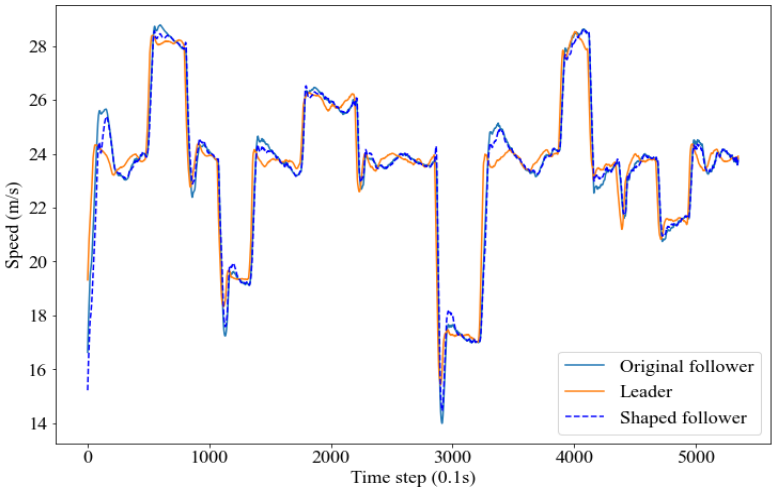
The CF behaviour of the commercial ACC system is retrofitted using the CTH-linear model with parameters estimated using the offline-online parameter estimation method. The natural frequency and damping ratio of the commercial ACC system are computed using the estimated parameters of CTH-linear model. Then, the VTS and the RTS are implemented to alleviate the string instability of the commercial ACC system. Figures 15(a) and (b) illustrate that both VTS and RTS can alleviate the string instability of the commercial ACC system: the speed fluctuations are attenuated at every hump. Correspondingly, the comparison of the speed fluctuations at every hump illustrates that the RTS achieves better string stability performance (i.e., less speed fluctuations) than the VTS (especially at around 300s when the VTS amplifies the overshoot). This is as expected due to estimation error occurring because the trajectory information may not be sufficiently informative to estimate accurate natural frequency and damping ratio. The potential nonlinearity in the commercial ACC system also leads to time-varying values of ω_0 and ζ, which can deviate from the estimated values. Then, the VTS is implemented with potentially inaccurate parameters, which will degrade its string stability performance (see Figure 5). By contrast, the RTS factors multiple natural frequencies and damping ratios that can cause string-unstable CF behaviour, which enables it to achieve desired robustness and better string stability performance. The performance of VTS and RTS under the batch-optimisation method is shown in Figures 15(c) and (d), respectively, where the inaccurately identified parameters degrade the string stability performance of both VTS and RTS, especially that of VTS. This further illustrates the need for the proposed offline-online parameter estimation method to implement the TS-based method.



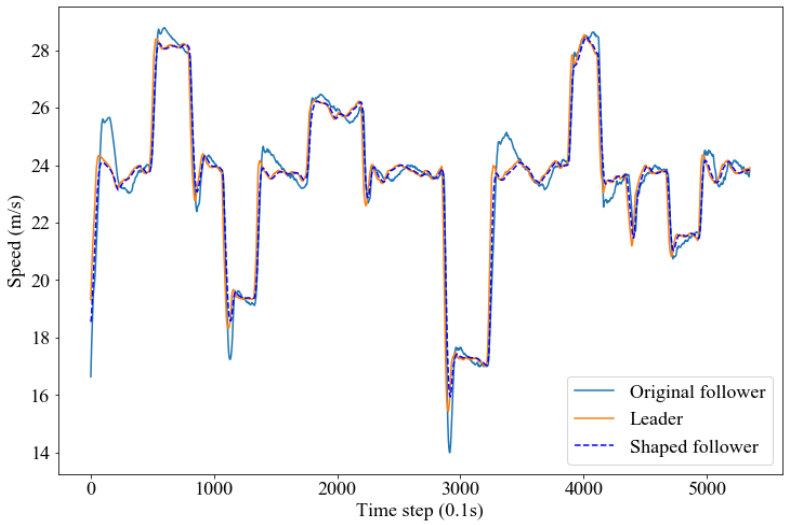
(a) Commercial ACC system with VTS implemented upon offline-online parameter estimation



(b) Commercial ACC system with RTS implemented upon offline-online parameter estimation



(c) Commercial ACC system with VTS implemented upon batch-optimisation parameter estimation



(d) Commercial ACC system with RTS implemented upon batch-optimisation parameter estimation

Figure 15. Performance of RTS and VTS using real-world CF data

Quantitative analysis of CF performance

This section statistically illustrates the CF performance of TS-based method. Specifically, 1341 lead vehicle trajectories extracted from the processed NGSIM dataset (Montanino and Punzo, 2015) are applied to the motion of the lead vehicle. A follower vehicle is controlled by either the original ACC (string-unstable CTH-linear model) or the ACC with VTS. The control parameters of the original ACC are: $k_s = 0.9$, $k_v = 0.15$, $h = 1$, $a_{\max} = 3\text{m/s}^2$, $a_{\min} = -6\text{m/s}^2$. The statistical significance level is set as 0.05.

Table 3 lists the mean values and the standard deviations (in parentheses) of average time headway across all time steps, speed standard deviation (STD), minimum time-to-collision (TTC), and maximum acceleration/deceleration of the follower vehicle. Negative TTCs and TTCs exceeding 10 seconds are dropped as they produce no meaningful implications on safety. With VTS, the average speed STD and maximum acceleration/deceleration are smaller than those of the original ACC, illustrating

improved traffic smoothness and comfort. The VTS can lead to increased average time headway and reduced minimum TTC. However, the minimum TTC still stays close to the satisfactory 5-second collision-avoidance value (Horst and Hogema, 1994), indicating no collision risks in stop-and-go traffic. The one-way ANOVA is then conducted. The large F-stats and p-values smaller than the significance level (0.05) validate the statistical significance of the results.

Table 3. Comparison of CF performance metrics

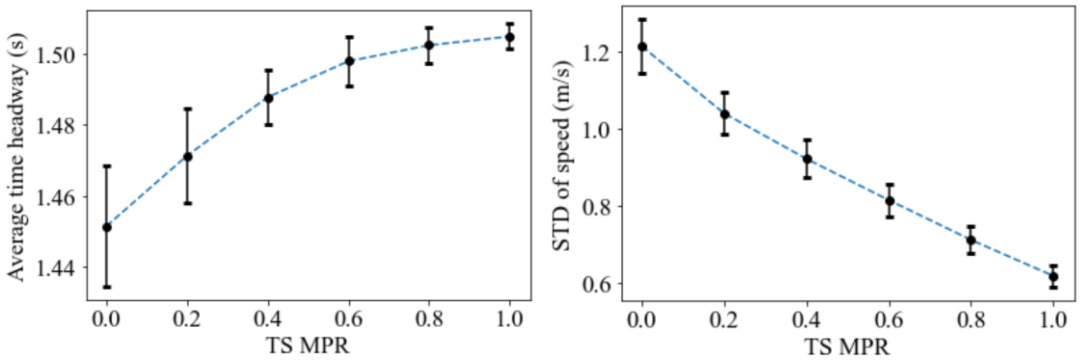
	ACC	ACC with VTS	F-stats	p-value
Average time headway (s)	1.221 (0.858)	1.487 (0.099)	127.425	<0.0001
Average STD of speeds (m/s)	1.325 (0.611)	1.101 (0.476)	110.967	<0.0001
Average minimum TTC (s)	5.415 (1.611)	4.459 (1.031)	517.079	<0.0001
Average maximum acceleration (m/s^2)	1.023 (0.347)	0.891 (0.302)	76.848	<0.0001
Average maximum deceleration (m/s^2)	-2.075 (1.669)	-0.892 (0.370)	641.699	<0.0001

Impacts of TS market penetration rate

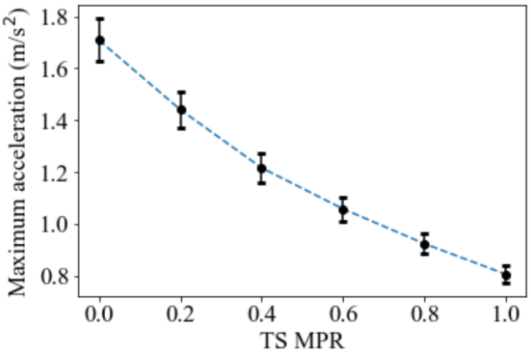
This section illustrates the impact of TS market penetration rate (MPR) on traffic flow by applying the 1341 lead vehicle trajectories from the processed NGSIM dataset (Montanino and Punzo, 2015). Ten follower vehicles controlled by either the original ACC (string-unstable CTH-linear model) or the ACC with VTS are incorporated to simulate the propagation of traffic congestion. The control parameters of the original

ACC are: $k_s = 0.9$, $k_v = 0.1$, $h = 1.2$. The simulation rollouts are conducted 100 times for each lead vehicle trajectory under different TS MPRs.

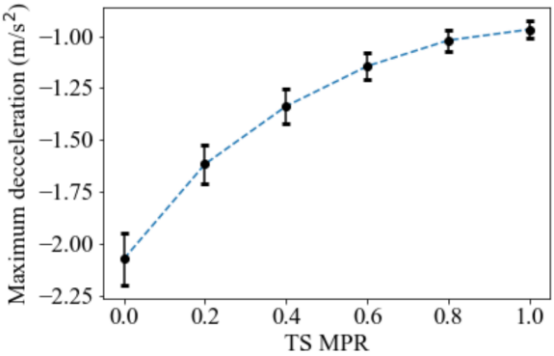
Figure 16 shows the mean values of average time headway across all time steps, speed STD, minimum TTC, and maximum acceleration/deceleration across all follower vehicles over the 100 rollouts. The black error bars are the 95% confidence intervals. Note that the negative TTC and TTC exceeding 10 seconds are dropped as they produce no meaningful implications on safety. Figure 16(a) shows that the average time headway increases with larger TS MPR, indicating more ACCs with TS-based method will induce sparser traffic flow. Figure 16(b) shows that speed STD reduces with larger TS MPR, illustrating that the TS-based method can create smoother speed profiles and alleviate the propagation of traffic oscillations. Figures 16(c) and 16(d) show that the maximum acceleration/deceleration decreases as TS MPR increases, implying that the TS-based method can improve riding comfort. Figure 16(e) shows that the average TTC of the platoon increases as TS MPR increases, indicating that the safety of a platoon can be improved by ACCs with the TS-based method. This is because string stability achieved by VTS enables vehicles away from the lead vehicle to experience significantly reduced shockwave and speed fluctuations, which correspondingly increase the TTCs. The increased TTCs dominate the reduced TTCs from the vehicles near the lead vehicle (i.e., the source of perturbation), leading to overall improved safety for the ACC platoon (contrary to the reduced minimum TTC of an immediate follower in Table 3).



(a) Average time headway

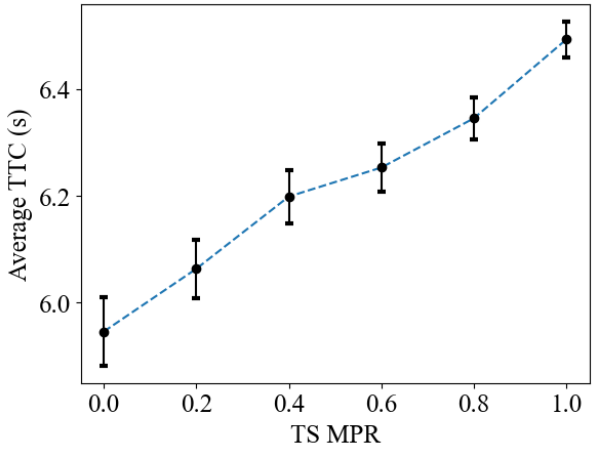


(b) Speed STD



(c) Maximum acceleration

(d) Maximum deceleration



(e) Minimum TTC

Figure 16. CF performance under different TS MPRs

Insights from abrupt harsh braking event

This section illustrates the impact of abrupt harsh braking event on TS-based method. The lead vehicle trajectory used here includes speed-varying operations and an abrupt harsh braking event at around time step 775. The parameters of the original ACC (controlled by string-unstable CTH-linear model) and VTS are: $k_s = 0.9$, $k_v = 0.15$, and $h = 1.2$.

To enhance safety performance under the harsh braking event, an automatic emergency braking (AEB) mechanism based on safe spacing (Zhu et al., 2020) is incorporated into the ACC with VTS and the original ACC: $a_{\text{ego}}(t) =$

$\begin{cases} a_{\min}, \ p_{\text{pred}}(t) - p_{\text{ego}}(t) \leq \ d_{\text{safe}} \text{ and } v_{\text{pred}}(t) \leq v_{\text{ego}}(t) \\ a_{\text{ego}}^{\text{shaped}}(t), \ \text{otherwise} \end{cases}$, where the safe spacing $d_{\text{safe}} = \frac{v_{\text{ego}}(t)^2 - v_{\text{pred}}(t)^2}{2|a_{\min}|}$ is the distance traveled by the ego vehicle after it decelerates to the speed of the predecessor vehicle with the maximum deceleration $a_{\min} = -8\text{m/s}^2$. Here, the AEB mechanism is assumed to be able to accurately capture vehicle states and react promptly.

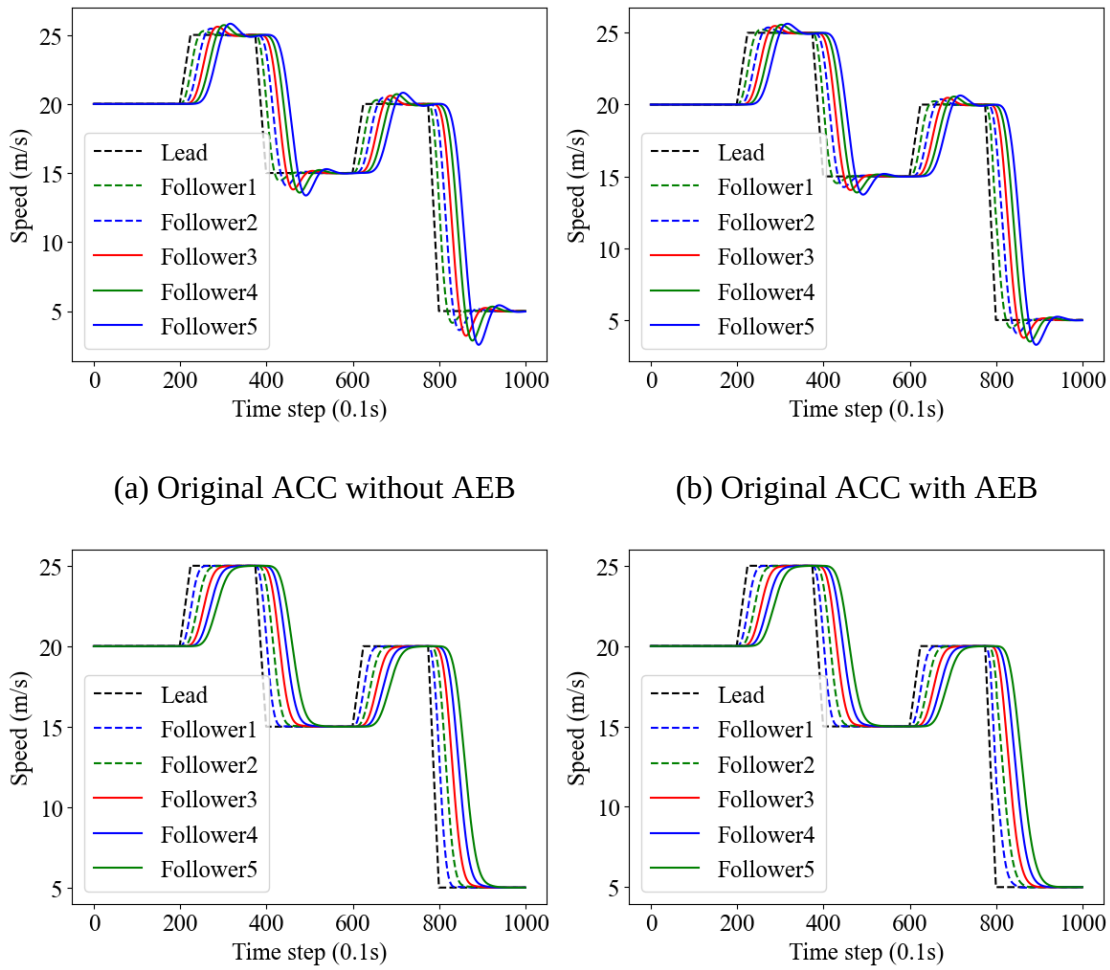
Figure 17 shows the speed profiles of five followers controlled by the original ACC and the ACC with VTS, with and without AEB. It can be observed that adding the AEB mechanism enables faster response to abrupt braking (i.e., faster varying speed curves). The AEB alleviates the undershoot of the original ACC (as shown in Figures 17(a) and 17(b)) and does not jeopardize the string stability achieved by the VTS (as shown in Figures 17(c) and 17(d)).

Table 4 shows the minimum TTC of the five followers under the different controllers. For the first two followers, the ACC with VTS reduces the minimum TTC of the original ACC. This is because string stability shrinks the transient spacings of vehicles close to the source of perturbation (e.g., the location of the harsh braking event). The value of minimum TTC is greater than the critical threshold of 1.4 seconds (Kusano and Gabler, 2011), indicating no safety hazards under the harsh braking event. With AEB, the minimum TTCs of VTS and the original ACC are both improved, and the minimum TTCs of the first two followers controlled by the ACC with VTS are almost identical to those of the original ACC. The results illustrate the need to incorporate an AEB mechanism and relevant abnormality detection system to improve safety of the TS-based method under emergency. Remarkably, the TTCs of the last three followers controlled by the ACC with VTS are greater than those of the original ACC, indicating an improved safety condition towards the tail of the platoon. Additionally, the average minimum TTC across

the platoon controlled by the ACC with VTS is also greater than that of the original ACC, implying that string stability can improve safety for the overall platoon.

Next, we apply the time exposed TTC (TET) to measure the duration of time when TTC is under a critical level of 3 seconds. TET is an aggregated measure to reflect safety throughout the CF process (instead of instantaneous performance at each time step described by TTC), where a smaller TET indicates a safer operation. Table 5 shows that the TET of the original ACC displays an increasing trend upstream the platoon, suggesting degraded safety from string instability. By contrast, the TET of ACC with VTS decreases upstream the platoon, and the TET of each vehicle is smaller than that of the original ACC. In addition, TET is substantially reduced when AEB is incorporated, indicating the effectiveness of AEB in improving ACC safety performance. The results validate that string stability can improve the overall safety of a platoon, while string instability will trigger safety concerns (especially when platoon size increases).

In summary, the results show that string stability can improve safety for the overall platoon under emergency. Meanwhile, as a string-stable follower close to the source of perturbation can experience a shrunk transient spacing (recall Table 3), the tolerance of system errors (i.e., maintaining safety under delays or abnormality of sensors and actuators) may be jeopardized during ACC operations. Thus, it is necessary to incorporate an AEB mechanism to enhance the safety of ACCs in case of extreme situations. The level of string stability should be adjusted based on traffic conditions and the location of the vehicle in a platoon to balance safety, efficiency, and stability.



(a) Original ACC without AEB

(b) Original ACC with AEB

(c) VTS without AEB

(d) VTS with AEB

Figure 17. Speed profiles with and without emergency brake

Table 4. Minimum time-to-collision

	ACC	ACC with AEB	ACC with VTS	ACC with VTS and AEB
Follower 1	1.833s	**2.161s**	1.693s	2.149s
Follower 2	2.334s	2.798s	2.319s	**2.934s**
Follower 3	2.460s	2.920s	2.719s	**3.261s**
Follower 4	2.515s	3.006s	3.073s	**3.544s**
Follower 5	2.536s	3.068s	3.395s	**3.809s**
Average	2.336s	2.791s	2.639s	**3.139s**

Table 5. Time exposed time-to-collision

	ACC	ACC with AEB	ACC with VTS	ACC with VTS and AEB
Follower 1	2.5s	**1.6s**	2.2s	2.0s
Follower 2	2.8s	1.2s	2.1s	**0.8s**
Follower 3	3.1s	0.7s	1.6s	**0s**
Follower 4	3.2s	0s	0s	0s
Follower 5	3.3s	0s	0s	0s
Average	2.98s	0.68s	1.18s	**0.56s**

Remark 2: In the architecture of a commercial ACC system, the AEB mechanism is typically required as an extra functionality to enhance safety, as the ACC emphasizes comfortable and smooth CF tasks, which cannot guarantee collision-free operations under emergency. The same control architecture should be applied to the TS-based method, as safety may be compromised in certain extreme cases.

Concluding Comments

This study proposes the TS-based method to modify the trajectory information of the predecessor vehicle to mitigate string instability under ACC. The VTS is proposed as a simple and straightforward shaper, but that it requires accurately identified parameters to achieve desired string stability performance. When the parameters of an ACC system are not accurately identified, its performance will degrade significantly. To address this, an RTS is proposed, which uses a nonlinear program to factor multiple natural frequencies and damping ratios to enhance robustness and maintain string stability under inaccurately identified parameters. To maintain reliable and safe operations under the TS-based method, the study develops an offline-online parameter estimation method, where the

batch-optimisation method is applied for the offline parameter estimation based on historical trajectory data and the EKF then iteratively improves the offline-estimated parameters using real-time measurements during ACC operations. It also explicitly discusses the sufficient conditions for obtaining the accurate parameters of an ACC system to devise the TS-based method.

The TS-based method can be implemented as a plug-in piggyback, entailing no laborious modifications to existing ACC control algorithms and codebase compared to developing new controllers. Thus, the corresponding computational complexity is also low, enabling rapid prototyping and efficient real-time implementation. VTS and RTS provide a cost-effective solution for automobile manufacturers, traffic-management administrative agencies, and after-market ADAS users to smoothen vehicle trajectories and mitigate traffic congestion.

The study insights suggest the following future research directions: (i) balancing the trade-off between traffic efficiency and string stability performance in TS-based method by developing more intelligent performance metrics and relaxing convolution operations; (ii) managing safety and string stability under different traffic conditions (e.g., harsh braking, cut-in, etc.) by optimising location-dependent string stability levels and incorporating proactive safety measures into the optimisation program; (iii) developing more advanced RTS program with the minimization of variance and expected performance metrics, while also factoring delayed and inaccurate measurements; (iv) extending the TS to cooperative platooning and lane-change control which factors trajectory information of multiple neighboring vehicles; and (v) conducting real-world CF experiments using the open-source autonomous driving platform “Openpilot” to test the proposed TS-based method.

Acknowledgements

This work is supported by Georgia Institute of Technology and National Science Foundation Cyber Physical System grant #1932451 and #1826162.

References

Åström, Karl J., and Richard M. Murray. 2010. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press. https://doi.org/10.5860/choice.46-2107.

Feng, Shuo, Yi Zhang, Shengbo Eben Li, Zhong Cao, Henry X. Liu, and Li Li. 2019. "String stability for vehicular platoon control: Definitions and analysis methods." *Annual Reviews in Control*. 47, 81-97.

Gong, Siyuan, Jinglai Shen, and Lili Du. 2016. "Constrained Optimisation and Distributed Computation Based Car Following Control of a Connected and Autonomous Vehicle Platoon." *Transportation Research Part B: Methodological*. https://doi.org/10.1016/j.trb.2016.09.016.

Gong, Siyuan, Anye Zhou, and Srinivas Peeta. 2019. "Cooperative Adaptive Cruise Control for a Platoon of Connected and Autonomous Vehicles Considering Dynamic Information Flow Topology." *Transportation Research Record*. https://doi.org/10.1177/0361198119847473.

Haidekker, Mark A. 2020. *Linear Feedback Controls: The Essentials*. Elsevier.

Van Der Horst, Richard, and Jeroen Hogema. 1994. “Time-to-collision and collision avoidance systems.” *In: 6th ICTCT workshop*. Salzburg, Austria.

Kesting, Arne, and Martin Treiber. 2008. "Calibrating car-following models by using trajectory data: Methodological study." *Transportation Research Record.* 2088(1), 148-156.

Khalil, Hassan K. 2002. *Nonlinear Systems* (3rd ed.). Prentice Hall. https://doi.org/10.1016/j.physa.2006.08.011.

Kusano, Kristofer D., and Hampton Gabler. 2011. "Method for estimating time to collision at braking in real-world, lead vehicle stopped rear-end crashes for use in pre-crash system design." *SAE International Journal of Passenger Cars-Mechanical Systems* 4 (2011-01-0576), 435-443.

Li, Tienan, Danjue Chen, Hao Zhou, Jorge Laval, and Yuanchang Xie. 2021. "Car-Following Behaviour Characteristics of Adaptive Cruise Control Vehicles Based on Empirical Experiments." *Transportation Research Part B: Methodological*, 147, 67–91. https://doi.org/10.1016/j.trb.2021.03.003.

Ljung, Lennart. 1999. *System Identification: Theory for the User* (2nd ed.). Pearson Education.

Lu, Xiao-Yun, and Steven E. Shladover. 2018. *Truck CACC System Design and DSRC Messages.* FHWA Exploratory Advanced Research Program Cooperative Agreement, Task 1–35.

Makridis, Michail, Konstantinos Mattas, Aikaterini Anesiadou, and Biagio Ciuffo. 2021. "OpenACC: An Open Database of Car-Following Experiments to Study the Properties of Commercial ACC Systems." *Transportation Research Part C: Emerging Technologies*, 125, 103047. https://doi.org/10.1016/j.trc.2021.103047.

Montanino, Marcello, and Vincenzo Punzo. 2015. "Trajectory Data Reconstruction and Simulation-Based Validation Against Macroscopic Traffic Patterns." *Transportation Research Part B: Methodological*, 80, 82–106. https://doi.org/10.1016/j.trb.2015.06.010.

Sugiyama, Yūki. 1999. "Optimal Velocity Model for Traffic Flow." *Computer Physics Communications*, 121, 399–401.

Naus, Gerrit JL, Rene PA Vugts, Jeroen Ploeg, Marinus JG van De Molengraft, and Maarten Steinbuch. 2010. "String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach." *IEEE Transactions on Vehicular Technology*, 59 (9), 4268–4279. https://doi.org/10.1109/TVT.2010.2076320.

Shladover, Steven E. 2009. "Effects of Cooperative Adaptive Cruise Control on Traffic Flow: Testing Drivers' Choices of Following Distances." California PATH Program, Institute of Transportation Studies, University of California Berkeley.

Singh, Tarunraj, and William Singhose. 2002. "Tutorial on Input Shaping/Time Delay Control of Maneuvering Flexible Structures." *Proceedings of the American Control Conference*, 3, 1717–1731.

Singhose, William E., W. P. Seering, and Neil C. Singer. 1996. "Input Shaping for Vibration Reduction with Specified Insensitivity to Modeling Errors." *Proceedings of the Japan/USA Symposium on Flexible Automation*, 1, 307–313.

Treiber, Martin, Ansgar Hennecke, and Dirk Helbing. 2000. "Congested Traffic States in Empirical Observations and Microscopic Simulations." *Physical Review E*, 62 (2), 1805–1824. https://doi.org/10.1103/PhysRevE.62.1805.

Villaverde, Alejandro F., Neil D. Evans, Michael J. Chappell, and Julio R. Banga. 2019. "Input-Dependent Structural Identifiability of Nonlinear Systems." *IEEE Control Systems Letters*, 3 (2), 272–277.

Wang, Jian, Siyuan Gong, Srinivas Peeta, and Lili Lu. 2019. "A Real-Time Deployable Model Predictive Control-Based Cooperative Platooning Approach for Connected and Autonomous Vehicles." *Transportation Research Part B: Methodological*, 128, 271–301. https://doi.org/10.1016/j.trb.2019.08.002.

Wang, Yanbing, Maria L. Delle Monache, and Daniel B. Work. 2022. "Identifiability of Car-Following Dynamics." *Physica D*, 430, 133090. https://doi.org/10.1016/j.physd.2021.133090.

Zhao, Yu, Wenjie Chen, Te Tang, and Masayoshi Tomizuka. 2016. "Zero Time Delay Input Shaping for Smooth Settling of Industrial Robots." *IEEE International Conference on Automation Science and Engineering*, 620–625. https://doi.org/10.1109/COASE.2016.7743459.

Zhou, Anye, Siyuan Gong, Chaojie Wang, and Srinivas Peeta. 2020. "Smooth-Switching Control-Based Cooperative Adaptive Cruise Control by Considering Dynamic Information Flow Topology." *Transportation Research Record*, 2674, 444–458. https://doi.org/10.1177/0361198120910734.

Zhou, Anye, Srinivas Peeta, Hao Zhou, Jorge Laval, Zejiang Wang, and Adian Cook. 2024. "Implications of stop-and-go traffic on training learning-based car-following control." *Transportation Research Part C: Emerging Technologies*. 168, 104578.

Zhou, Hao, Jorge Laval, Anye Zhou, Yu Wang, Wenchao Wu, Zhu Qing, and Srinivas Peeta. 2022a. "Review of Learning-Based Longitudinal Motion Planning for Autonomous Vehicles: Research Gaps Between Self-Driving and Traffic Congestion." *Transportation Research Record*, 2676(1), 324–341.

Zhou, Hao, Anye Zhou, Tienan Li, Danjue Chen, Srinivas Peeta, and Jorge Laval. 2022b. "Significance of Low-Level Controller to String Stability Under Adaptive Cruise Control: Algorithms, Theory, and Experiments." *Transportation Research Part C: Emerging Technologies*, 140, 103697.

Zhou, Yang, and Soyoung Ahn. 2019. "Robust Local and String Stability for a Decentralized Car-Following Control Strategy for Connected Automated Vehicles."

Transportation Research Part B: Methodological, 125, 175–196. https://doi.org/10.1016/j.trb.2019.05.003.

Zhou, Yang, Meng Wang, and Soyoung Ahn. 2019. "Distributed Model Predictive Control Approach for Cooperative Car-Following with Guaranteed Local and String Stability." *Transportation Research Part B: Methodological*, 125, 175–196. https://doi.org/10.1016/j.trb.2019.07.001.

Zhu, Meixin, Yinhai Wang, Ziyuan Pu, Jingyun Hu, Xuesong Wang, and Ruimin Ke. 2020. "Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving." *Transportation Research Part C: Emerging Technologies*, 117, 102662.