2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Selecting Flow Optimal System Parameters for Automated Driving Systems

Florian Hauer *Member, IEEE*, Raphael Stern *Member, IEEE*, and Alexander Pretschner *Member, IEEE*

***Abstract*— Driver assist features such as *adaptive cruise control* (ACC) and highway assistants are becoming increasingly prevalent on commercially available vehicles. These systems are typically designed for safety and rider comfort. However, these systems are often not designed with the quality of the overall traffic flow in mind. For such a system to be beneficial to the traffic flow, it must be string stable and minimize the inter-vehicle spacing to maximize throughput, while still being safe. We propose a methodology to select autonomous driving system parameters that are both safe and string stable using the existing control framework already implemented on commercially available ACC vehicles. Optimal parameter values are selected via model-based optimization for an example highway assistant controller with path planning.**

I. Introduction

Driver assist features such as *adaptive cruise control* (ACC) and highway assistants promise to be the first step toward an autonomous future. Such products have become commonplace on many new commercially available vehicles that are now on the market. These features are typically designed for safety and rider comfort [1], but do not necessarily take the overall traffic flow into account.

Changing the dynamics of a small number of vehicles within the traffic has been shown to change the emergent properties of the underlying flow [2]. This can influence traffic string stability, which determines whether traffic jams will arise due to the vehicle driving behavior alone. These string instabilities have been shown to greatly increase fuel consumption and vehicle emissions of the entire traffic flow [2], [3]. To avoid these adverse impacts of string instability, it is important that new automated driving systems that are deployed be string stable, or close to string stable, in that small disturbances must travel through a long platoon of vehicles before the initial disturbance grows appreciably.

The design of string stable automated driving systems has been a major focus for several years [4]–[12]. However, experimental work by Melanés and Shladover [13] showed that at least one commercially available ACC vehicle was string unstable. More recently, work by Gunter, *et al.* has shown that of eight different commercially available automated driving systems tested, all are string unstable [14], [15]. Thus, while commercially available automated driving systems may be designed to be string stable, it is possible that the overall vehicle system when implemented is not string stable as shown by [13], [14]. For automated driving systems, which typically take over additional driving tasks beyond longitudinal control of the vehicle, it may be even more difficult to ensure string stability by design.

One possible way to achieve string stability is by increasing the inter-vehicle spacing, or space gap. However, since the space gap is inversely proportional to the throughput, increasing space gap will decrease the capacity of the roadway. Therefore, we introduce the designation *flow optimal*, with which we mean that the automated driving controller behaves in such a way that it maintains as small of a space gap as possible while being string stable. Moreover, the design of automated driving systems should be both safe and flow optimal to not only benefit the driver of the vehicle but also be satisfactory from a traffic flow perspective.

While automated driving controllers that are string stable have been proposed in the literature, many commercially implemented controllers are not string stable [13]–[15]. To address this, we propose model based system parameter optimization approach to select automated driving control parameters that are both safe and flow optimal [16]. This framework can be applied to any number of proprietary automated driving controllers. However, for demonstration, we implement the framework on a specific highway assistant. It is a model-predictive control approach that optimizes a driving trajectory within a safety corridor [17], [18].

The goal of this article is to provide a novel methodology that can be implemented in the prototyping phase of the vehicle design to select automated driving controller parameters that are both safe and flow optimal. Thus, the proposed method is control agnostic, since no specific knowledge of the automated driving controller beyond the controller parameters is required to identify a set of flow optimal parameter values. To do this, a high resolution simulation of the car following behavior is conducted, as is common during the prototyping phase of automotive development. The lead vehicle and following vehicle trajectories are used to analyze the system for safety and throughput and to estimate a speed-to-speed transfer function, which is analyzed for string stability. Based on this analysis, optimal system parameter values are found. The novel aspect of this work is the formulation of a fitness function that considers both safety and flow optimality, to then apply standard search-based techniques for system optimization.

The remainder of this article proceeds as follows. A introduction to string stability is presented in Section II. System optimization is presented in Section III. Specifically, the search based optimization is outlined in Section III-B and the fitness function is derived in Section III-C. The designed methodology is demonstrated on an example autonomous

F. Hauer and A. Pretschner are with the Department of Informatics at the Technical University of Munich, Germany. (email: {florian.hauer, alexander.pretschner}@tum.de).

R. Stern is with the Department of Informatics at the Technical University of Munich, Germany and the Department of Civil, Environmental, and Geo-Engineering at the University of Minnesota, USA (email: rstern@umn.edu).

driving system with path planning in Section IV and flow optimal system parameter values are identified and tested in simulation. We conclude that the proposed framework can be applied to a variety of systems in Section V.

II. TRAFFIC STRING STABILITY

For a platoon of vehicles, it is possible to define an equilibrium flow where each vehicle drives with the same speed and a fixed spacing (and thus no acceleration). If we introduce a small perturbation to this equilibrium flow, it will propagate from one vehicle to the next along the platoon of vehicles. String stability tells whether this perturbation will amplify or dissipate as it propagates through the platoon of vehicles. Specifically, for a string stable platoon, this perturbation will eventually dissipate until successive vehicles are not affected by the perturbation. If the platoon is string unstable, this small initial disturbance will amplify as it propagates from one vehicle to the next [19]. Therefore, the string stability of each vehicle's following behavior depends on the two-vehicle interaction between each successive vehicle pair.

Practically, string stability of a following vehicle can be assessed by estimating a speed-to-speed transfer function $\Gamma(s)$ for a two-vehicle trajectory pair as a function of the complex frequency s, which explains the frequency response of a system to a specific input [20]. A sufficient condition for string stability is then [12]:

$$|\Gamma(s)| \leq 1 \ \forall s. \tag{1}$$

Using the transfer function, it is also possible to construct a Bode plot to identify which frequency ranges are amplified, and at what amplification amplitude. Further, the maximum amplification amplitude can be used as a measure for how well-behaved a system is with respect to amplifying disturbances. If the maximum amplification amplitude from the Bode plot is small, disturbances may not grow sufficiently large on a reasonable timescale that is relevant for traffic. More information can be found in the article by Wilson and Ward [19].

Thus, the relevant measure when designing automated driving for practical implementation is the amplification amplitude A:

$$A = \max_{s} \ 20 \log_{10} |\Gamma(s)|, \tag{2}$$

where A is the amplitude in decibels. Specifically, A tells how quickly disturbances are amplified as they propagate from one vehicle to the next. Therefore, the magnitude of A is used as a measure of string instability, with larger values of A being considered further from string stable. This is similar to the notion of stability margin introduced by Wang [21].

III. SYSTEM OPTIMIZATION

In this section, the methodology for the system optimization is introduced, and the fitness function used for the optimization is described.

A. Methodology overview

During the automotive design process for an automated system, the entire system is first designed at an abstract level. At this point, there are certain design decisions that need to be made early on in the design process so that individual teams can use these specifications to design individual components independently. For example, the computation cycle time is often determined early in the design process, since it affects the computation schedule of a variety of hardware and software components. However, other aspects of the project fall under the category of design decisions that can be postponed by parameterizing attributes $p_i \in P$ ($\forall i < n = |P|$) and their domains $D_i \in D$. For example, the concrete (safety) distance that is kept to other traffic participants may be adjusted later on. Thus, the exact configuration of the system can be determined at a later point. The domains span a space of possible system configurations $C = D_1 \times D_2 \times ... \times D_n \subset \mathbb{R}^n$. Assigning specific values to each p_i yields a specific configuration. Selecting these parameters must then be done in such a way as to satisfy any design requirements (e.g., safety, string stability, etc.). These can then be selected to be the optimal parameters based on some optimization routine.

B. Search-Based Optimization with Scenarios

Search-based techniques can be used to identify the best candidate in a search space with the help of a fitness function as seen in Figure 1 where an initial set of candidate system parameters is created either by reusing existing system configurations, by using manually created ones by experts, or selected at random. These candidates are then tested in a simulation and the simulation results are evaluated via the fitness function, which returns a quantitative measure of the quality of the respective system configuration. According to these fitness values, the optimization algorithm updates the parameter values to obtain a new system configuration that performs better in the fitness function than the original parameter values. This iteration may be continued until a maximum number of iterations is reached, the assigned computation time is spent or the optimizer fails to find a better solution. Ideally, search-based techniques are able to find the global optimum, which is the best system configuration with respect to the fitness function. In this work, we use a multi-objective optimization technique (e.g., NSGA-II [22]), which make use of Pareto optimality. Instead of a single fitness value, a vector of fitness values is used with each element of the vector representing one of the three desired objectives described next (safety, string stability, and minimum following timegap). Note that in this work we use **minimization** to find the system optimal parameter values. Thus, a vector x is better than another vector y if all $x_i \leq y_i$ and at least one $x_i < y_i$.

C. Search Space and Fitness Function

Both a search space of potential system configurations and a fitness function is required for the optimization routine. The search space C is the space of parameters, depends

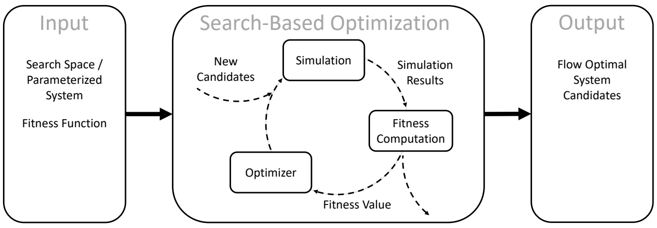


Fig. 1. Search-based techniques for scenario optimization.

on the specific autonomous system, and may differ between manufacturers. However, there are three aspects that hold for every driving system that has to take over the driving task (possibly among other driving tasks) of following a lead vehicle: the planned and controlled spacing, velocity, and acceleration. These are fundamental for the vehicle following behavior. For safety considerations, a safety distance to the lead vehicle always has to be maintained, while velocity and acceleration influence how a system adapts to dynamicity changes of the lead vehicle. There are multiple system parameters that influence those three aspects. Concrete values for those parameters cannot be chosen independently since each affects string stability and safety. Therefore, the optimal parameters that are safe and flow optimal are selected via model-based optimization over the candidate space C.

To identify an optimal system configuration, the desired system properties have to be incorporated into a fitness function. In addition to being safe and string stable, the system should keep minimum spacing to the lead vehicle. However, the desired system properties may be contradictory to each other. Specifically, maintaining a small timegap may contradict with the safety considerations and string stability. Therefore, there are multiple goals that are optimized, and a balance between these objects must be obtained. To address this, we use multi-objective optimization, which requires a fitness function that computes an array of fitness values, i.e., one fitness value per goal (safety, string stability, low spacing).

The overarching system requirement is safety since flow optimality is irrelevant if the system is not safe. In this work, we use a safety distance as a measurement for how safe the system is. A variety of different safety distance interpretations are presented in existing works. We use the one presented in [23], which is a physical interpretation of the safety distance described in the Vienna Road Convention [24].

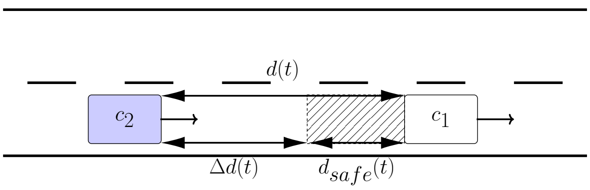


Fig. 2. Minimum safety distance at a specific moment in time t.

A vehicle c_2 that is following another vehicle c_1 as shown in Figure 2 has to keep sufficient distance from c_1 to avoid collision if c_1 suddenly slows down or stops. This notion defines a certain minimum safety distance $d_{safe}(t)$ which c_2 has to respect throughout the scenario. The remaining distance until violation $\Delta d(t)$ can be computed knowing the safety distance. The system may not violate the safety distance constraint during the entirety of a simulation. Thus, the remaining buffer until violation $\Delta d(t)$ must never be negative. This is the case if the minimum value of $\Delta d(t)$ is not negative. Let this minimum be denoted as α:

$$\alpha = \min(x_{c_1}(t) - x_e(t) - d_{safe}(t)) \tag{3}$$
$$= \min(d(t) - d_{safe}(t)) \tag{4}$$
$$= \min(\Delta d(t)), \tag{5}$$

where e is the ego vehicle, x_e its longitudinal position, and x_{c_1} the other car's longitudinal position. This measurement for safety is then used as part of the fitness function for the safety goal. All safe systems are considered equally safe and thus get the same fitness value. This means a large value of $\Delta d(t)$ is not necessary since a small one is sufficient for safety.

Since we minimize the value of the fitness function, the best system configuration needs to be assigned the smallest fitness value. For a non-safe system, $\alpha < 0$, therefore we minimize $-\alpha$ to obtain a safe system. The final fitness function for safety looks as follows:

$$f_{safety} = \begin{cases} -\alpha; & \alpha < 0 \\ 0; & otherwise. \end{cases} \tag{6}$$

Similarly, a fitness function for string stability as another goal is derived. As described in Section II, the maximum amplification rate A can be used as a measurement distance from string stability for a particular system. To estimate A, numerical approximations can be used for the transfer function in (2). If A is positive, the system is not stable and disturbances are amplified, while if $A \leq 0$ the system is string stable. However, the rate at which the system dissipates disturbances is not the focus of this work. Therefore, we employ the same concept as above to distinguish stable from unstable systems and use the following fitness function for string stability:

$$f_{stable} = \begin{cases} A; & A > 0 \\ 0; & otherwise. \end{cases} \tag{7}$$

To ensure that the automated system does not achieve string stability simply by leaving an excessive timegap to the lead vehicle, a fitness function to minimize the timegap τ is also used for the goal of low spacing. Here timegap τ is the inter-vehicle headway measured in time. While important for flow optimality, maintaining a low timegap is considered secondary to the the safety and string stability requirements. Therefore, the fitness function for timegap returns a large constant value for system configurations that are not both safe and string stable. This constant needs to be bigger than any possible value for the spacing, e.g., ∞, as this means that all systems that are not safe and stable get a worse fitness value for spacing than all systems that are safe and stable.

$$f_{spacing} = \begin{cases} \infty; & f_{safety} + f_{stable} > 0 \\ \tau; & otherwise. \end{cases} \quad (8)$$

Combining these three fitness functions yields the final fitness function that can be used for optimization with multi-objective techniques to find system parameter values that are safe, stable, and minimize the inter-vehicle spacing:

$$f_{final} = [f_{safety} \quad f_{stable} \quad f_{spacing}]. \quad (9)$$

IV. Numerical example and experiments

In this section, a numerical example is used to demonstrate the proposed method on the driving behavior of an automated driving system with path planning. We introduce experimentally-collected field data that is used as the lead vehicle driving profile, and introduce the example automated driving system, which is simulated to follow the lead vehicle. Flow optimal system parameter values are selected via the proposed optimization routine, and simulations are conducted to demonstrate the performance of the optimized system.

A. Experimental Data and String Stability Computation

The experimental data used for the lead vehicle speed profile is presented in the work by Gunter, *et al.* [15] and is collected from an instrumented vehicle that drives on a straight, flat track. The lead vehicle in the data drives at a constant 22 m/s (80 km/h) and conducts multiple, quick deceleration events before accelerating back up to 22 m/s after each event as seen in Figure 3. This dataset is selected since it is representative of the rapid disturbances that may be seen in typical traffic flow. More information on the experimental methods used to collect the data are presented in the article by Gunter, *et al.* [15].

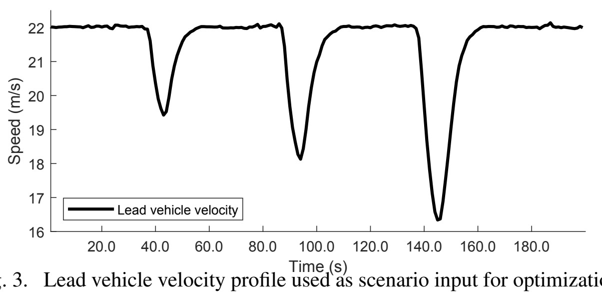


Fig. 3. Lead vehicle velocity profile used as scenario input for optimization.

The data in Figure 3 is used by the simulation software as the lead vehicle speed profile. During the optimization, a speed-to-speed transfer function is estimated using the MATLAB function `tfest()` receiving this lead vehicle speed profile and the simulated following vehicle trajectory as input. This function finds the best-fit transfer function between two timeseries datasets. The maximum amplification amplitude A is then computed using (2).

B. Example Automated Driving System

The proposed methodology to find flow optimal parameters is designed to work on a broad range of automated driving system that a vehicle manufacturer may have implemented on a vehicle. However, for the purpose of demonstration, we use an example automated driving system that is described below.

From the multitude of approaches in literature, we chose the following model-predictive control approach [17], [18] as basis for our implementation for two reasons: First, it is designed for the use as a highway pilot of characteristic of SAE level 4 systems [25]. Second, it models the aspects that influence velocity, acceleration and spacing explicitly, making the results in this article easier to interpret.

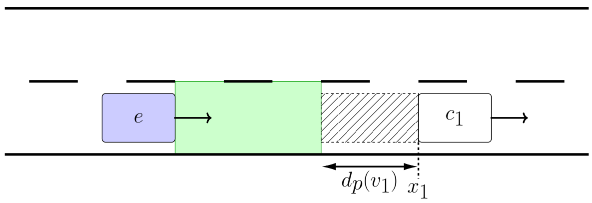


Fig. 4. Schematic simplified depiction of the predicted positions x_i and planned safety distance $d_p(v_i)$ at a specific prediction time t_k.

The autonomous driving system used in this article works as follows: first, in the planning the positions x_i and velocities v_i of surrounding cars c_i are estimated for each sample time step t_k over a short time horizon (see Figure 4). For safety distance planning, a timegap τ is used, which is one of the parameters to optimize. The safe operating envelope is bounded by the planned safety distance $d_p(v_i) = \tau \cdot v_i$ to the predicted positions x_i of other cars c_i. Within the bounds of this envelope, a trajectory is identified by quadratic optimization the following way:

$$J = \sum_{k=0}^{N} \theta (v_k - v_{des})^2 + \kappa a_k^2. \quad (10)$$

θ and κ along with τ, which determines the planning distance d_p, are the system parameters for which our approach finds optimal values. The objective is to keep the velocity at each time step v_k as close to the desired velocity v_{des} as possible over all time steps N of the prediction and planning horizon. The desired velocity v_{des} is determined by the driver in Level 1-2 systems, and by the speed limit in higher levels of autonomy. In the example presented in this article, a desired speed of $v_{des} = 33.33$ m/s= 120 km/h is used. Similarly, the acceleration at each time step a_k is kept as low as possible in each time step. While optimizing the trajectory one can not minimize both the acceleration and the difference of velocities at the same time. For example, assume a car in front of the automated vehicle is slowing down, there are two possible outcomes: either the system starts braking early with small braking acceleration ($\theta << \kappa$), which would minimize the acceleration in all time steps, but causes the velocity to deviate from the desired velocity for a larger number of time steps; or the system brakes later with a larger (negative) acceleration ($\theta >> \kappa$), which minimizes the difference of the velocities in all time steps, but causes a greater absolute value of the acceleration. This underlines the difficulty in finding satisfactory parameter values.

The tracking of the longitudinal trajectory is done by a PID controller, controlling the position of the gas and the brake pedal. It is tuned for a physical model of a sports car with a weight of 1,410 kg, length of 4.18 m, width of 1.83 m and height of 1.35 m. The model is provided by the widely used physical simulation software CarMaker by IPG Automotive [26], which also served as the simulation environment for this work. This simulation tool is capable of capturing the vehicle dynamic response and produces realistic driving behavior that mimics real-world testing.

These parameters (θ, κ, and τ) influence the path planning by motivating the planner to prefer trajectories of certain form. We allow the weights to be withing 1 to 1000 (see Table I), and the timegap to be between 0.0 s and 5.0 s. These boundaries are a rough over approximation, e.g. a driving system that keeps 5.0 s is not desirable at all. However, the desire for low spacing will ensure that a small value is chosen for this timegap.

Parameter	Lower Bound	Upper Bound
Velocity model parameter θ	1.0	1000.0
Acceleration model parameter κ	1.0	1000.0
Timegap model parameter τ	0.0	5.0

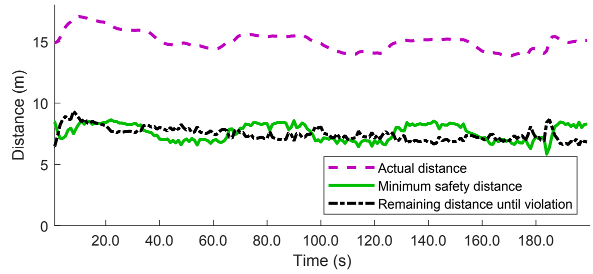
TABLE I

SYSTEM PARAMETERS FOR PATH PLANNING.

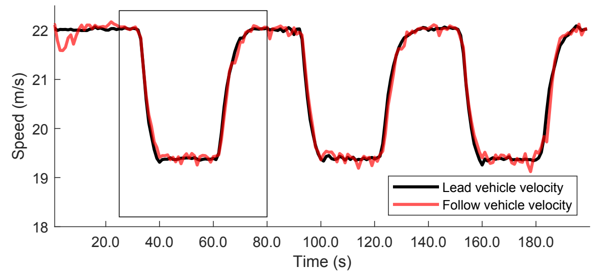
C. Results

For the optimization, the multi-objective genetic algorithm from the global optimization toolbox of MATLAB [27] is applied. Note that the technological aspect is not the focus of this work and other techniques could be used as well. The population size was set to 150 and the number of generations to 15, resulting in 2250 simulation executions. The experiments were executed multiple times to rule out randomization effects.

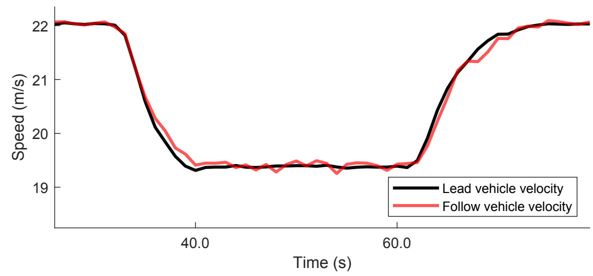
A new dataset that is not used during optimization and, thus, unseen by the system (see black curve in Figure 5b) is then used as a lead vehicle speed profile in CarMaker. The optimized system is simulated to drive behind the lead vehicle using the flow optimal system parameter values identified in the optimization ($\theta = 777.211$, $\kappa = 255.891$, $\tau = 0.684$) with the best fitness values of $[0, 0, 0.684]$. The resulting simulated following vehicle speed profile under the flow optimal parameter values shown in Figure 5 is plotted in a solid red line. For the duration of the scenario, the the system remains safe, in that the remaining distance until violation never becomes negative as seen in Figure 5a. This means that the safety distance is not violated. In Figure 5a we see that the spacing maintained by the vehicle is roughly twice the minimum safety distance which could lead to the assumption that it would be possible to further reduce the spacing to increase throughput. However, the maintained following distance in this example (roughly 15m at 80 km/h) is the minimum distance achievable to also be string stable. Additionally, there may be other circumstances for which this following behavior is required to maintain safety.



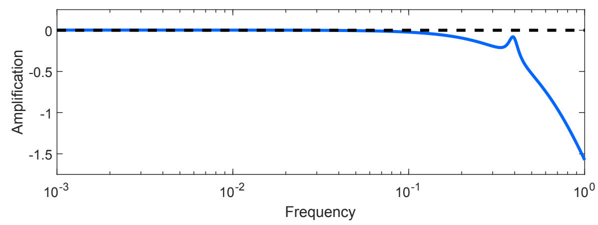
(a) Distances during the scenario.



(b) Velocities during the scenario.



(c) Zoom in from 25 s to 80 s.



(d) String stable Bode plot for the estimated speed-to-speed transfer function of the designed following vehicle.

Fig. 5. Results of experiment for the system configuration with parameter values $\theta = 777.211$, $\kappa = 255.891$, $\tau = 0.684$.

The system remains string stable, in that the perturbation is not amplified as seen in Figure 5c, where no systematic overshoot or undershoot is observed in the following vehicle's speed profile compared to the lead vehicle. The string stability of the resulting system is also seen in Figure 5d where the Bode plot of the speed-to-speed transfer function of the following vehicle is plotted. This shows that at all frequencies, the designed automated following system is able to either leave unchanged, or decrease the amplitude of the disturbance making it string stable. Since this system also minimizes the planned minimum following distance while remaining string stable (roughly 0.7 s headway in this case), the system is considered to be flow optimal.

The results plotted in Figure 5 only guarantee string stability for the specific scenario being shown in this figure. However, by constructing the Bode plot for the estimated transfer function of the following vehicle behavior, we see that the resulting system dissipates all frequencies of disturbances, and therefore is string stable.

V. Conclusions

In conclusion, the proposed methodology for selecting flow optimal and safe automated driving system parameters is a practical way to design string stable automated driving systems for real-world applications that minimize following distance while maintaining a safe following distance. This can be done within the framework of the existing controllers that are particular to each vehicle manufacturer. The example presented in this article shows one such system, and the simulation results with the calibrated system parameters show that the system maintains a safe following distance and does not amplify disturbances on a testing dataset that was not used for training, even with a following timegap as low as 0.7s. However, it is important to note that the example only represents one test drive, and therefore string stability assessments can only be made for the range of disturbance frequencies observed during this drive. Additionally, testing safety in one scenario does not imply safety in all scenarios. Furthermore, the current optimization does not take into account aspects such as rider comfort, which could be included by extending the fitness function to consider this.

The proposed methodology is a practical approach to identifying autonomous driving system parameters that do not only achieve a desired vehicle-level performance (e.g., safety), but also take the overall traffic flow into account by considering string stability and traffic density. This methodology is intended to be straightforward and controller agnostic, meaning that it can easily be implemented to design automated driving systems without specific knowledge of the underlying controller dynamics. This makes the proposed methodology particularly applicable in practice.

VI. Acknowledgements

The authors would like to acknowledge Dr. Daniel Work and George Gunter for providing the experimental data used in simulation experiments for this article.

References

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4):6–22, 2014.

[2] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, R. Haulcy, H. Pohlmann, F. Wu, B. Piccoli, B. Seibold, J. Sprinkle, and D. B. Work. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205 – 221, 2018.

[3] R. E. Stern, Y. Chen, M. Churchill, F. Wu, M. L. Delle Monache, B. Piccoli, B. Seibold, J. Sprinkle, and D. B. Work. Quantifying air quality benefits resulting from few autonomous vehicles stabilizing traffic. *Transportation Research Part D: Transport and Environment*, 67:351–365, 2019.

[4] W. Levine and M. Athans. On the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, 11(3):355–361, 1966.

[5] P. Ioannou, Z. Xu, S. Eckert, D. Clemons, and T. Sieja. Intelligent cruise control: theory and experiment. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 1885–1890, 1993.

[6] S. E. Shladover. Review of the state of development of advanced vehicle control systems (avcs). *Vehicle System Dynamics*, 24(6-7):551–595, 1995.

[7] D. Swaroop and J.K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.

[8] R. Rajamani, S. B. Choi, B. K. Law, J. K. Hedrick, R. Prohaska, and P. Kretz. Design and experimental implementation of control for a platoon of automated vehicles. *AMSE Journal of Dynamic Systems, Measurement, and Control*, 122(3):470–476, 1998.

[9] C.-Y. Liang and H. Peng. Optimal adaptive cruise control with guaranteed string stability. *Vehicle System Dynamics*, 32(4-5):313–330, 1999.

[10] A. Alam, B. Besselink, V. Turri, J. Martensson, and K. H. Johansson. Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency. *IEEE Control Systems*, 35(6):34–56, 2015.

[11] B. Besselink and K. H. Johansson. String stability and a delay-based spacing policy for vehicle platoons subject to disturbances. *IEEE Transactions on Automatic Control*, 2017.

[12] J. Monteil, M. Bouroche, and D. J. Leith. $\mathcal{L}_2$ and $\mathcal{L}_\infty$ stability analysis of heterogeneous traffic with application to parameter optimization for the control of automated vehicles. *IEEE Transactions on Control Systems Technology*, 2018.

[13] V. Milanés and S. E. Shladover. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transportation Research Part C: Emerging Technologies*, 48:285–300, 2014.

[14] G. Gunter, D. Gloudemans, R. E. Stern, S. McQuade, R. Bhadani, M. Bunting, M. L. Delle Monache, R. Lysecky, B. Seibold, J. Sprinkle, B. Piccoli, and D. B. Work. Are commercially implemented adaptive cruise control systems string stable? *Submitted to IEEE Transactions on Control Systems Technology*, 2019, under review.

[15] G. Gunter, C. Janssen, W. Barbour, R. E. Stern, and D. B. Work. Model based string stability of adaptive cruise control systems using field data. *Submitted to IEEE Transactions on Intelligent Vehicles*, 2019, under review.

[16] S. S. Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.

[17] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson. Longitudinal and lateral control for automated lane change maneuvers. In *2015 American Control Conference (ACC)*, pages 1399–1404. IEEE, 2015.

[18] J. Nilsson, J. Fredriksson, and E. Coelingh. Trajectory planning with miscellaneous safety critical zones. *IFAC-PapersOnLine*, 50(1):9083–9088, 2017.

[19] R. E. Wilson and J. A. Ward. Car-following models: fifty years of linear stability analysis–a mathematical perspective. *Transportation Planning and Technology*, 34(1):3–18, 2011.

[20] J. P. Hespanha. *Linear systems theory*. Princeton University Press, 2018.

[21] M. Wang. Infrastructure assisted adaptive driving to stabilise heterogeneous vehicle strings. *Transportation Research Part C: Emerging Technologies*, 91:276–295, 2018.

[22] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature*, pages 849–858. Springer, 2000.

[23] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf, and T. Nipkow. Formalising and monitoring traffic rules for autonomous vehicles in isabelle/hol. In *International Conference on Integrated Formal Methods*, pages 50–66. Springer, 2017.

[24] Economic Commission for Europe - Inland Transport Committee. Convention on road traffic, 1968. Vienna.

[25] Taxonomy SAE. Definitions for terms related to on-road motor vehicle automated driving systems. *J3016, SAE International Standard*, 2014.

[26] CarMaker. *V7.1*. IPG Automotive GmbH, Karlsruhe, Germany, 2019.

[27] MATLAB Global Optimization Toolbox. The MathWorks Inc., Natick, MA, USA, 3.4.4 2018a.