Proceedings of the ASME 2015 Dynamic Systems and Control Conference
DSCC2015
October 28-30, 2015, Columbus, Ohio, USA

DSCC2015-9807

ROBUST COOPERATIVE ADAPTIVE CRUISE CONTROL DESIGN FOR CONNECTED VEHICLES

Mark Trudgen
Complex Systems Control Laboratory
College of Engineering
University of Georgia
Athens, Georgia 30602
Email: mtrudgen@uga.edu

Javad Mohammadpour
Complex Systems Control Laboratory
College of Engineering
University of Georgia
Athens, Georgia 30602
Email: javadm@uga.edu

ABSTRACT

In this paper, we design and validate a robust H_∞ controller for Cooperative Adaptive Cruise Control (CACC) in connected vehicles. CACC systems take advantage of onboard sensors and wireless technologies working together in order to achieve smaller inter-vehicle following distances, with the overall goal of increasing vehicle throughput on busy highways, and hence serving as a viable approach to reduce traffic congestion. A group of connected vehicles equipped with CACC technology must also ensure what is known as string stability. This requirement effectively dictates that disturbances should be attenuated as they propagate along the platoon of following vehicles. In order to guarantee string stability and to cope with the uncertainties seen in the vehicle model used for a model-based CACC, we propose to design and implement a robust H_∞ controller. Loop shaping design methodology is used in this paper to achieve desired tracking characteristics in the presence of competing string stability, robustness and performance requirements. We then employ model reduction techniques to reduce the order of the controller and finally implement the reduced-order controller on a simulation model demonstrating the robust properties of the closed-loop system.

NOMENCLATURE

- h Headway
- q_i Position of the i^{th} car
- ϕ Internal delay
- τ Time constant

1 INTRODUCTION

Connected vehicles are an example of a modern day cyber physical system (CPS) that through the use of Cooperative Adaptive Cruise Control (CACC) provide an innovative solution to the traffic congestion problem [1]. Traffic is becoming an increasing problem in today's world as congestion in many urban areas is growing at a much faster rate than the traditional means of traffic alleviation can assuage [2]. CACC is a technology that seeks to reduce traffic congestion by means of achieving higher traffic flow rates using advanced control systems to safely reduce the allowable *headway* time between vehicles [3]. A widespread advantage of CACC over traditional means of increasing traffic throughput, i.e., road construction, is that CACC has the potential to be implemented on any car in highway system without the additional high costs and time delays associated with road construction projects [4].

CACC technology is an extension of Adaptive Cruise Control (ACC), which in turn is an extension of conventional cruise control (CCC), a technology traditionally used to regulate a vehicle at a constant highway speed [5]. ACC extends the CCC technology by regulating the so-called *headway* distance between vehicles that are arranged together in a platoon [6]. ACC employs radar (or lidar) sensors to measure the relative velocity and displacement with the preceding vehicle, and a longitudinal control framework is then implemented to space the vehicles to an appropriate *headway* [5] by adjusting the acceleration and deceleration of the vehicle. CACC extends the ACC technology by adding wireless inter-vehicle communication [7]. This ex-

tension enables smaller *headway* distances, which is critical for platoon technology to have a noticeable impact on traffic mitigation [4, 8]. According to the *2010 Highway Capacity Manual*, a study observing human drivers showed that the maximum flow rate for a multilane highway (at 60 mi/h) equates to 1.1 seconds of *headway* [4]. Herein lies the main drawback of ACC technology, that is the smallest stable *headway* is larger than the average time-gap that human drivers naturally exhibit [2, 8], thus justifying the need for CACC technology. The vehicles that are virtually connected to each other through CACC technology must ensure an important metric called string stability [9]. This concept was first introduced in [10] and later extended in [11], which led to the development of systems using the nearest neighbor as a measurement. Essentially, string stability is a requirement that all disturbances introduced in the string be attenuated as they propagate in the upstream direction [6, 11]. String stability is essential to ensuring the safety and feasibility of the string [9]. Not only do any disturbances in position, velocity or acceleration create increased energy consumption, these disturbances must also be mitigated in order to prevent the so-called *ghost traffic jams* [6], or even in extreme cases, an accident [12]; hence, a control design formulation that can explicitly account for string stability inherently meets design objectives and exterminates the need for any ad-hoc *a posteriori* tuning to achieve string stability. This notion of string stability has been studied in several aspects such as Lyapunov stability, and input-output stability; however, these methods lack the consideration of a measure of performance as seen in [13, 14], which give a frequency-domain approach for controller synthesis.

Several approaches have been undertaken in designing a controller for a platoon of vehicles. The system model considered to describe the vehicular motion is usually a 3^{rd} order nonlinear model [15, 16], where subsequently the plant is linearized by the use of feedback linearization method. For the control design using the linearized model, several CACC experimental results have been reported, e.g., in [6, 7, 17]. These recent works show the promise in using CACC. Indeed, several aspects of CACC technologies have been studied. The authors in [12] developed a sampled data approach to CACC design in the presence of sensors and actuator failures and [18] studied strategies for worst case scenarios. Model predictive control (MPC) has also gained attention as a way to cast the CACC problem in a framework that can directly optimize fuel economy. A CACC MPC approach can be considered very useful for heavy duty vehicles, such as tractor trailer trucks as in [19], where smaller *headway* distances can be sacrificed for better fuel economies as traffic throughput may not be the primary objective as is the case with urban rush hour highway demands. CACC can also be viewed in light of the communication as a networked control problem where the effects of sampling, hold, and network delays can be taken into account. An H_∞ formulation of network controlled problems is given in [20]. Still, other works have investigated communication-based time-varying delays and communication structures beyond the classical architecture as in [21].

To the authors' best knowledge, no previous work has extended the CACC framework to include modeling uncertainties directly arising from the plant using a decentralized framework. Fundamentally, all system models exhibit a level of model uncertainty [22]. Indeed, in the experimental results of [6], it was noted that the parameters of the plant were found using a least squares averaging technique, and it is known that uncertainty comes from the parameters describing the linearized plant. In [23] time constant parameter variations were mentioned, but aside from ensuring LHP stable poles, a robust control design framework was not considered. Similarly, although packet loss and communication delays were considered in [21], no consideration was made with respect to parameter variations of a linearized plant. We consider in this paper a robust controller design, where an H_∞ controller is sought to be synthesized as the induced energy-to-energy gain (or H_∞ norm) is a natural norm to use in the presence of uncertainty [22], especially considering the literature available on $\mathcal{L}_2$ string stability [9, 13]. In our formulation, we choose to model the CACC problem in a decentralized manner similarly to [9]. While other formulations exist for centralized control such as [24], we choose the decentralized formulations as they have strong relevance to every day traffic applications where there is no set leader. A decentralized implementation also gives each driver in the string control over a range of *headway* values, which is desirable considering different driving abilities; however, an investigation of psychological aspects is not considered here, for which the reader is referred to references in [2].

This paper is structured as follows: Section 2 describes the platoon following technologies and how the simple nonlinear model governing vehicles is linearized. We also show simulation results comparing different adaptive cruise control technologies. Section 3 explains the design of a robust H_∞ controller, whose order will then be reduced while still retaining the desired robust properties. Section 4 illustrates the results of a 5-car simulation, and Section 5 draws conclusions.

2 Various Cruise Control Technologies

Design of various longitudinal adaptive cruise control strategies have been studied in the literature (i.e. [5] and references therein). Figure 1 shows a representative view of a typical string of vehicles equipped with cooperative adaptive cruise control (CACC), where the lead car of the string sets a trajectory to follow and communicates its acceleration a_0 only to the following vehicle. Alongside the communicated acceleration, the following vehicle is equipped with onboard sensors to measure the relative distance and velocity. This is typically done via the use of radar (or lidar) [6]. In considering a platoon, the distance between vehicles is broken into 3 segments: d_i is the desired static distance between vehicles, hv_i is the product of the minimum

headway required and the velocity of the i^{th} vehicle, and finally δ_i is an additional spacing parameter. The i^{th} vehicle is said to be in the correct positioning when $\delta_i = 0$.

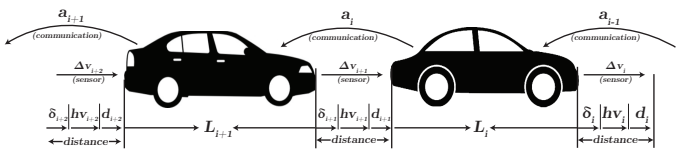

Figure 1: A string of vehicles equipped with cooperative adaptive cruise control technology.

More specifically, δ_i, the spacing policy, is given as [12]

$$\delta_i = q_{i-1} - q_i - L_i - hv_i - d_0, \tag{1}$$

where h is the time gap (headway), d_0 is a given minimum distance and L_i is the length of the i^{th} vehicle. The system dynamics can be represented as [15, 16]

$$\begin{aligned} \dot{\delta}_i &= v_{i-1} - v_i - h\dot{v}_i \\ \Delta\dot{v}_i &= a_{i-1} - a_i \\ \dot{a}_i &= f_i(v_i, a_i) + g_i(v_i)c_i, \end{aligned} \tag{2}$$

where $g_i(v_i)$ is given as

$$g_i(v_i) = \frac{1}{\tau_i m_i}. \tag{3}$$

Subsquently, the model is nonlinear due to the nonlinear function $f_i(v_i, a_i)$ which is described as

$$f_i(v_i, a_i) = -\frac{1}{\tau_i}\left[\dot{v}_i + \frac{\sigma A_i c_{di}}{2m_i} v_i^2 + \frac{d_{mi}}{m_i}\right] - \frac{\sigma A_i c_{di} v_i a_i}{m_i}, \tag{4}$$

where m_i represents the i^{th} vehicle's mass, τ_i is the engine time-constant of the i^{th} vehicle, $\frac{\tau_i A_i c_{di}}{2m_i}$ is the air resistance, d_{mi} is the mechanical drag, c_{di} is the drag coefficient and σ is the specific mass of the air. To linearize the above nonlinear system dynamics, the following control law is adopted [15, 16]

$$c_i = u_i m_i + \frac{\sigma A_i c_{di} v_i^2}{2} + d_{mi} + \tau_i \sigma A_i c_{di} v_i a_i, \tag{5}$$

where u_i is the new control input signal to be designed for the closed-loop system where $c_i < 0$ and $c_i \geq 0$ correspond to brake and throttle actions, respectively. Using (5) results in a feedback linearization, which combined with (2) gives

$$\dot{a}_i(t) = -\frac{a_i(t)}{\tau_i} + \frac{u_i(t)}{\tau_i}. \tag{6}$$

Since $a_{i-1}(t)$ is sent from the preceding vehicle, a communication delay θ_i is introduced so the acceleration arriving at the i^{th} vehicle is $a_{i-1}(t-\theta_i)$. Writing the CACC model in the state-space form gives [12]

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_{i1} u_i(t) + B_{i2} w_i(t - \theta_i) \\ y_i(t) &= \left[x_i^T(t), w_i(t)\right]^T, \end{aligned} \tag{7}$$

where θ_i is the communication delay, $x_i = [\delta_i, \Delta v_i, a_i]^T$ is the state vector, $w_i(t) = a_{i-1}(t)$ and $y_i(t) = [\delta_i, \Delta v_i, a_i, w_i]^T$ is the output vector, and additionally,

$$A_i = \begin{bmatrix} 0 & 1 & -h \\ 0 & 0 & -1 \\ 0 & 0 & -1/\tau_i \end{bmatrix}, \quad B_{i1} = \begin{bmatrix} 0 \\ 0 \\ 1/\tau_i \end{bmatrix}, \quad B_{i2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{8}$$

We follow [6, 7, 23] in assuming a low-level linearizing feedback controller. The system in (8) gives the linearization for the i^{th} vehicle, and the overall system is hence a decentralized platoon.

2.1 Adaptive Cruise Control

By setting a_{i-1} to zero in (2), the CACC model reduces to the ACC model, and the same feedback linearizing controller given in (5) can be used to achieve a linear model. Next, by using the setup proposed in [6], the corresponding block diagram is given in Figure 2.

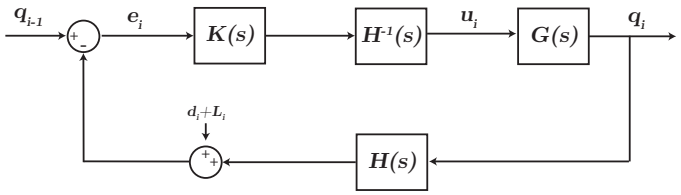

Figure 2: Adaptive cruise control block diagram.

For the i^{th} vehicle, we use the following notation: q_{i-1} denotes the preceding vehicle's position, q_i denotes the local position, e_i is the error signal inputted into the controller $K(s)$ and u_i is the so-called *desired* acceleration (that is used as an input to the linearizing controller, see, e.g., [12]). Finally, d_i denotes an added static following distance, and L_i is the length of the i^{th} vehicle. Without the loss of generality, $L_i = d_i = 0$ is assumed. In

addition, $G(s)$ represents the system transfer function, and $H(s)$ describes the spacing policy given as

$$G(s) = \frac{q_i(s)}{u_i(s)} = \frac{1}{s^2(\tau_i s + 1)} e^{-\phi_i s} \tag{9}$$

$$H(s) = hs + 1, \tag{10}$$

where τ_i is the engine time constant and the nominal value is taken as $\bar{\tau}$ = 0.1 sec. and $\bar{\phi}$ = 0.2 sec is an associated nominal internal delay and h represents the designed *headway* value [6, 25]. We built a simulation model in MATLAB/SIMULINK that was composed of 5 cars using a simple stabilizing controller is given by .

$$K(s) = K_D s + K_P, \tag{11}$$

where K_D = 0.7 and K_P = 0.2 [6].

The *headway* time, h, is set to 0.6 sec. Using this headway value, we do not achieve string stability in the ACC case. This headway value is chosen to illustrate that even lower headway values can be achieved with communication, thus justifying additional model complexity required. An inherent goal is to reduce the headway as this correlates to a better traffic mitigation.

2.2 Degraded Cooperative Adaptive Cruise Control

As a bridge between ACC and CACC, the authors in [25] propose the use of an onboard observer that uses local measurements to estimate the accerlation of the previous vehicle. This can be used when, e.g., a communication link experiences packet losses and before resorting to an ACC scheme [25]. A block diagram of the degraded Cooperative Adaptive Cruise Control (dCACC) case is shown in Figure 3, where $T(s)$ is a Kalman estimator and $T_{aa}(s)$ is a smoothing filter. The boxed section in Figure 3 is used to denote the estimation scheme. It is noted that this is an onboard estimation scheme implemented in the i^{th} vehicle. Using (11) again, we see that the dCACC has improved damping compared to the ACC case, but still not being able to achieve string stability for low headway values.

2.3 Cooperative Adaptive Cruise Control

Next, by introducing a dedicated short range communication (DRSC) protocol between vehicles, the leading vehicle's acceleration can be communicated to the following vehicle. As this signal is transmitted through communication channel, there is a delay; hence,

$$D(s) = e^{-\theta s}, \tag{12}$$

where θ = 0.02 sec. is chosen as in [6, 25]. The implemented model in MATLAB/SIMULINK is modified to now include these communication delays as shown in Figure 4.

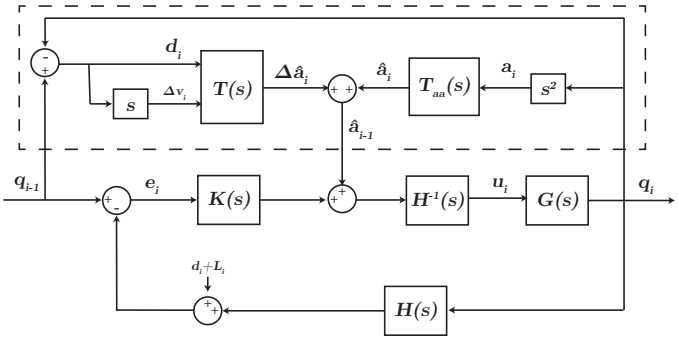

Figure 3: dCACC Block Diagram.

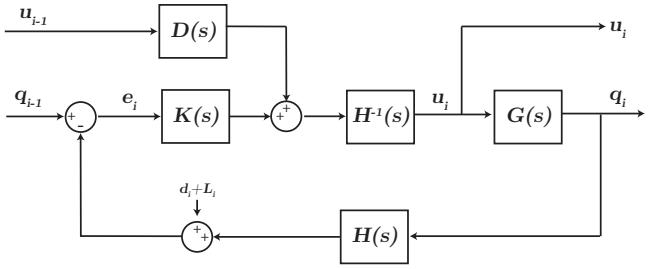

Figure 4: CACC block diagram.

In the case of a CACC scheme with a stabilizing controller, the block diagram is shown in Figure 4 and the controller $K(s)$ is the same as in (11) [6]. The communicated acceleration is used as a feedforward term. Using the same *headway* value used in the previous cases, the CACC scheme does achieve string stability. Although the error is non-zero, it does not increase along the string.

String Stability

We denote the transfer function from q_{i-1} to q_i as $\Gamma_{CACC}(s)$ given by

$$\Gamma_{CACC}(s) = \frac{1}{H(s)} \frac{G(s)K(s) + D(s)}{1 + G(s)K(s)}. \tag{13}$$

$D(s)$ represents the delay associated with either the dCACC case or the CACC case. Setting $D(s)$ = 0 yields the ACC case. Figure 5 shows the Bode plots corresponding to the three platoon control approaches described before for h = 0.6 sec. For string stability $||\Gamma(j\omega)|| < 1$ needs to be achieved for any ω, which physically implies that the position of the vehicle q_i remains behind the preceding vehicle q_{i-1}. From Figure 5, it is observed that only the CACC system satisfies this requirement. As noted in [8] for this technology to have a noticeable impact on traffic mitigation, a *headway* significantly smaller than 1.1 sec. already seen in the

naturalistic driving must be achieved [2], and the dCACC and ACC cases do not even achieve the naturalistic driving *headway* value.

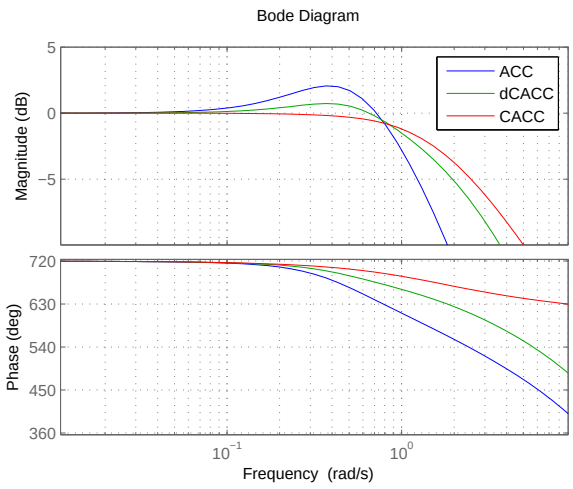

Figure 5: Frequency response associated with ACC, dCACC and CACC.

3 Robust CACC Design

In this section, we discuss the design of a robust CACC system in the framework of robust H_∞ control. To this purpose, we first introduce the sources of uncertainty and describe how to quantify them.

3.1 Sources of Uncertainty

There are several reasons to incorporate robustness into a control design framework as there usually exist several sources of uncertainty within any dynamic system. There are always parameters that are only approximately known or are modestly in error. Also, linear models may only be adequate for a small operating range, and original measurements taken to find parameters have inherent errors despite calibration. If the model is obtained through system identification methods, at high frequencies the structure of the model can become unknown and uncertainties in parameters always arise. Finally, there might be uncertainties within the controller [22]. There are several different approaches to model uncertainties, which could be classified under structured or unstructured uncertainties [22].

With respect to CCAC applications, the authors in [6] note that "the parameters were estimated using a least-squares method." Several other authors have noted that alongside parameter variations seen in portion of (2) associated with the i^{th} vehicle parameters, the use of radar (or lidar) and the DRSC band gives other sources of uncertainties [5]. Indeed, since all CACC systems run on onboard processors, albeit real-time systems, there is still a non-uniform processing time that adds to the potential time delays resulting in uncertainties in the plant.

3.2 Representing Uncertainty

For the plant given in (9), the parameters ϕ and τ are assumed to have the nominal values of $\bar{\phi} = 0.2$ sec. and $\bar{\tau} = 0.1$ sec., where we consider a variation with $\phi \in [0.05, 0.5]$ and $\tau \in [0.02, 0.2]$. To guarantee the closed-loop system stability in the presence of the model uncertainty associated with ϕ and τ we first represent the lumped parameter *multiplicative uncertainty* as shown in Figure 6 and equation (14).

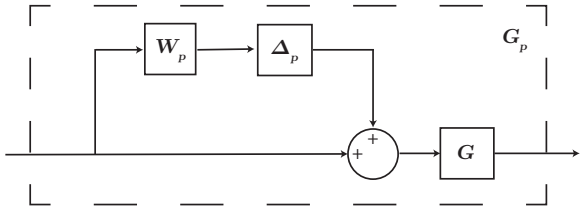

Figure 6: Lumped parameter multiplicative uncertainty.

$$G_p(s) = G(s)(1 + W_p(s)\Delta_p(s)), \tag{14}$$

where $G_p(s)$ represents the perturbed model, $G(s)$ represents the nominal model, $\|\Delta_p\|_\infty \leq 1$, and W_p represents the lumped uncertainties transfer function that satisfies [22]

$$\left| \frac{G_p(j\omega) - G(j\omega)}{G(j\omega)} \right| \leq |W_p(j\omega)|, \tag{15}$$

for any frequency ω. We then let ϕ and τ vary over each respective parameter set. Using a fine grid, we plotted the left hand side of (15) on a Bode plot shown in Figure 7, where in (15) $G_p(s)$ is taken as the perturbed plant and $G(s)$ is fixed as the plant at $\bar{\phi}$ and $\bar{\tau}$. Then, a high pass filter, $W_p(s)$ was fitted to the Bode plot according to (15). This results in the following high-pass filter

$$W_p(s) = \frac{6s + 0.003}{s + 14}. \tag{16}$$

3.3 Loop Shaping for H_∞ Control Design

Next, we use the loop shaping approach [22] to design a controller that can guarantee tracking with zero steady-state error and a low control effort. The corresponding block diagram in Figure 8 depicts how disturbances and noise signals affect the closed-loop system. Using this block diagram setup, as in [9], the string stability requirement can be directly handled within the H_∞ framework. In standard loop shaping, weight W_e shown in Figure 8 is tuned to penalize tracking error at low frequencies. The weight W_e is selected to be a low pass filter, tuned to eliminate the steady-state error, as

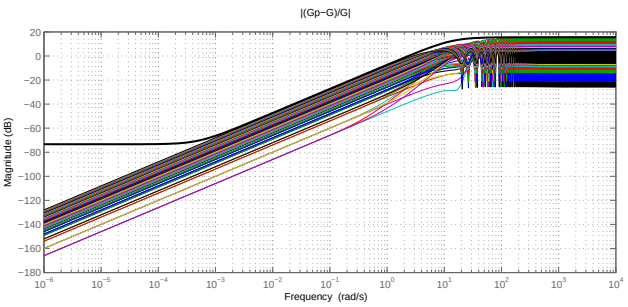

Figure 7: Bode plots to find the multiplicative uncertainty weight.

$$W_e(s) = \frac{0.028}{s+0.02}. \tag{17}$$

Next, we select the *desired* acceleration, u_i, as an exogenous output signal [9]. Writing the transfer function between the exogenous input, i.e., the previous vehicle's acceleration u_{i-1}, and the *desired* acceleration u_i yields,

$$T_i(s) = \frac{u_i(s)}{u_{i-1}(s)}. \tag{18}$$

If $||T_i(j\omega)|| \leq 1$ for any ω, we have achieved string stability. The weight W_p is a high pass filter used to model the multiplicative uncertainties as discussed in the previous section.

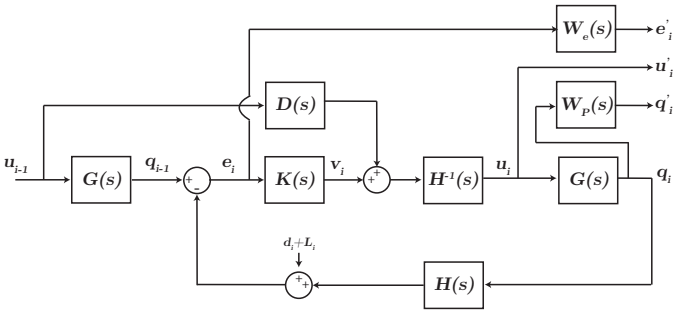

Figure 8: Configuration of the closed-loop control system.

3.4 H_∞ Robust Control Design for Connected Vehicles

After selecting the loop shaping weights, we use MATLAB to represent the system interconnection shown in Figure 8 into the linear fractional transformation (LFT) form. This is done by using the MATLAB command *sconnect*. Next, we express the closed-loop system as

$$z(s) = N(s) * w(s), \tag{19}$$

where z represents the vector containing controlled output signals, $N(s)$ describes the closed-loop system transfer function matrix and w represents the exogenous input signals [22]. In formulating the closed-loop system, the delays associated with (9) and (12) are approximated by using a 3^{rd} order Padé approximation. Now, by imposing the requirement that,

$$||N(j\omega)||_\infty \leq 1, \tag{20}$$

string stability will be achieved. Next, the robust control design problem is solved by invoking the MATLAB command *hinflmi*. A 13^{th} order controller is synthesized to satisfy (20). We finally use model order reduction methods to reduce the order of the controller. First, a Gramian-based balancing of state-space realization is performed to isolate states with negligible contribution to the input/output response. This results in an 8^{th} order controller. We further reduce the controller to 6^{th} order by using a balanced truncation model order reduction. Comparing the Bode plot of the 13^{th} order system with the 6^{th} order system shows a good approximation over all frequencies while also satisfying the requirement in (20).

4 Simulation Results and Discussion

Using the reduced-order controller designed in the previous section, we perform a 5-car simulation with the nominal values of $\bar{\phi}$ = 0.2 sec and $\bar{\tau}$ = 0.1 sec. The results are shown in Figures 9 and 10 illustrating the string stable behavior, along with the desired tracking performance. For the simulation we follow the same smooth velocity step as in the previous section, where the lead car decreases velocity from 60 kph to 40 kph. Figure 10 shows a low value of the error in the response, also demonstrating that after the first following car in the string, the error becomes negligible all together.

Next, by inspecting the block diagram given in Figure 8, we write the sensitivity and complementary sensitivity functions as

$$S(s) = \frac{G(s)(1-D(s))}{1+G(s)K(s)}, \tag{21}$$

$$T(s) = \frac{H^{-1}(s)(G(s)K(s)+D(s))}{1+G(s)K(s)}. \tag{22}$$

Figure 11 shows the corresponding Bode plots, which illustrate that string stability is achieved according to (18) as the complementary sensitivity transfer function $T(s)$ is always less than 1 at all frequencies. Additionally, Figure 12 shows the corresponding robust stability margin illustrating that in the given design, robust stability is achieved. This can be seen from Figure 12 since $||W_p(j\omega) * T(j\omega)|| \leq 1$ for any ω.

Next, using the reduced-order robust controller we perform

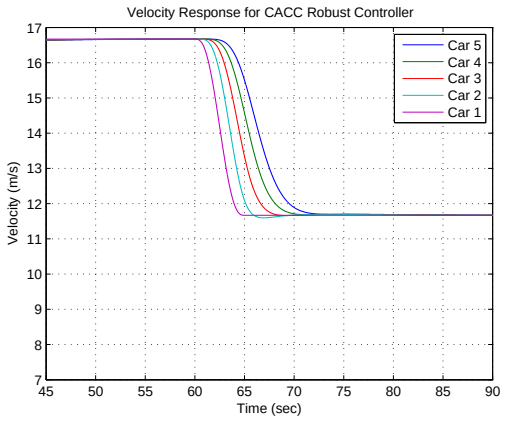

Figure 9: Velocity simulation using the designed robust controller.

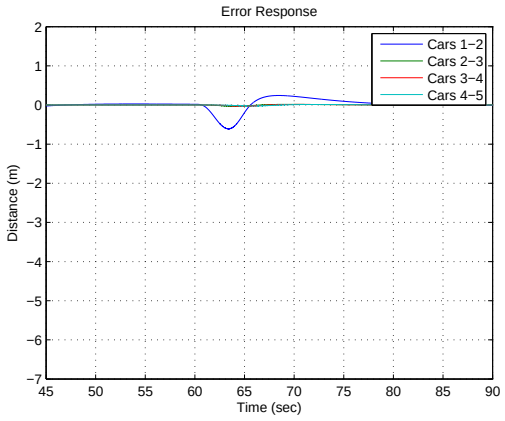

Figure 10: Error responses using the designed robust controller.

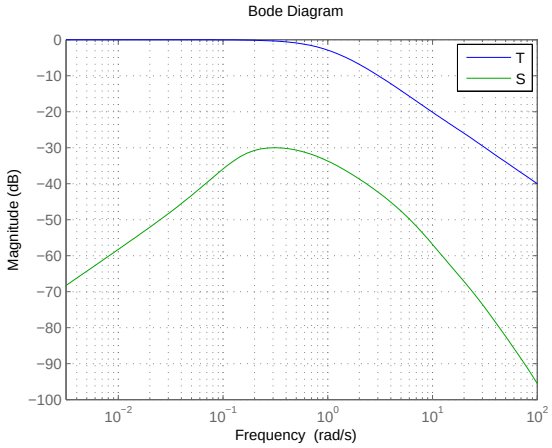

Figure 11: Frequency response of sensitivity and complementary sensitivity transfer functions.

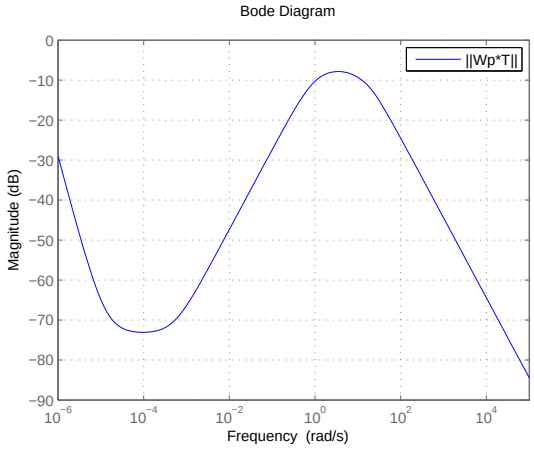

Figure 12: Plot showing the robust stability condition.

the 5-car simulations for the parameter values of $\phi = 0.5$ sec. and $\tau = 0.2$ sec. We then also perform the same 5-car simulation with the same perturbed parameter values for a standard (non-robust) H_∞ controller designed in [9]. Figure 13 shows the velocity response of the robust controller, demonstrating that the brief undershoot is quickly damped out. Figure 14 shows that the non-robust controller experiences several oscillations before reaching steady state. For both controllers, only the response of the first following car is considered non-trivial (similar to Figure 10), and a comparison of the error response between the two controllers is given in Figure 15. Comparing the two sets of simulations, i.e., the proposed robust design vs. the non-robust one, shows that the robust controller provides a much better performance over the region of parameter perturbation.

5 Conclusions

In this paper, we have provided some new results on the design and validation of a robust H_∞ controller for cooperative adaptive cruise control (CACC) of connected vehicles. The proposed design framework can account for the uncertainties in the vehicle model used for the CACC design to ensure string stability. The control design process includes: (i) quantifying the effect of uncertainties on the plant model, and (ii) employing the mixed-sensitivity, loop shaping-based H_∞ control design. Simulation results demonstrate that the robust controller can improve string stability and tracking performance – compared to non-robust designs in the literature – over the region of parameter perturbations.

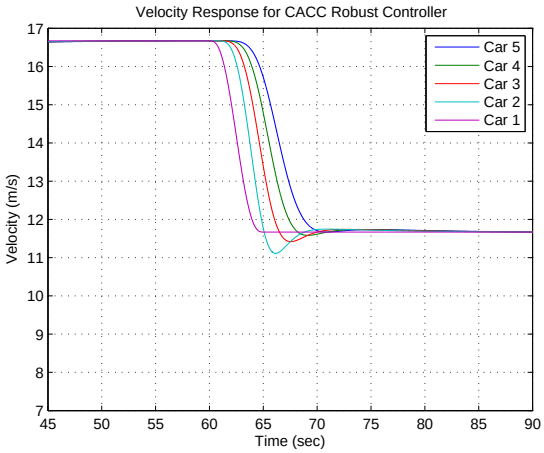

Figure 13: Velocity profiles for perturbed 5-car simulations using the proposed robust controller.

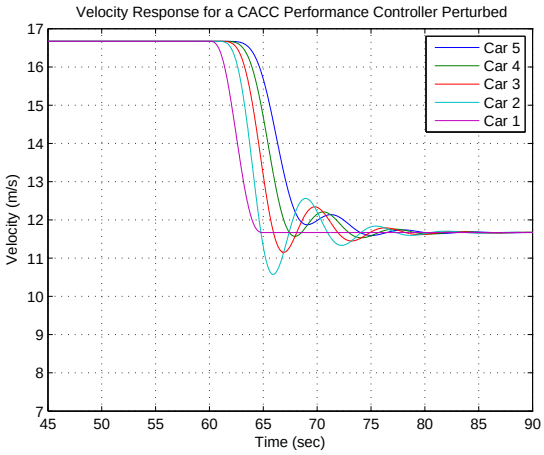

Figure 14: Velocity profiles for perturbed 5-car simulations using the (non-robust) controller proposed in [9].

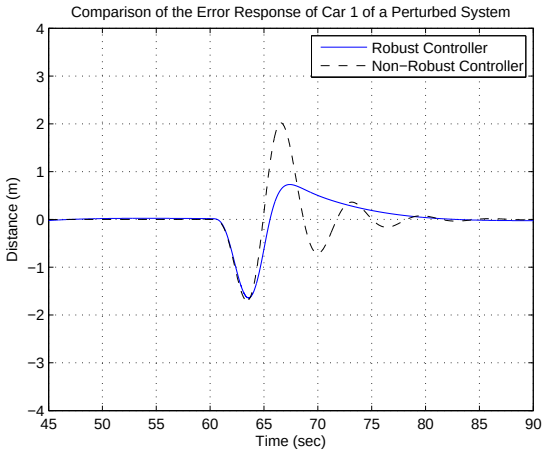

Figure 15: Tracking error profiles for perturbed 5-car simulations for the proposed robust controller and the H_∞ controller designed in [9].

References

[1] J. Holdren, E. Lander, and H. Varmus, "Report to the president and congress designing a digital future: Federally funded research and development in networking and information technology executive office of the president," President's Council of Advisors on Science and Technology, Tech. Rep., December 2010.

[2] S. Jones, "Cooperative adaptive cruise control: Human factors analysis," no. FHWA-HRT-13-045, 2013.

[3] "Focus on congestion relief," Federal Highway Administration 2012, http://www.fhwa.dot.gov/congestion/, accessed on 2015-01-06.

[4] J. Vander Werf, S. E. Shladover, M. A. Miller, and N. Kourjanskaia, "Effects of adaptive cruise control systems on highway traffic flow capacity," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1800, no. 1, pp. 78–84, 2002.

[5] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," *Vehicle System Dynamics*, vol. 48, no. 10, pp. 1167–1192, 2010.

[6] J. Ploeg, B. T. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 260–265.

[7] D. de Bruin, J. Kroon, R. van Klaveren, and M. Nelisse, "Design and test of a cooperative adaptive cruise control system," in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 392–396.

[8] S. E. Shladover, D. Su, and X.-Y. Lu, "Impacts of cooperative adaptive cruise control on freeway traffic flow," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2324, no. 1, pp. 63–70, 2012.

[9] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller synthesis for string stability of vehicle platoons," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 2, pp. 854–865, 2014.

[10] W. S. Levine and M. Athans, "On the optimal error regulation of a string of moving vehicles," *IEEE Transactions on Automatic Control*, 1966.

[11] L. Peppard, "String stability of relative-motion PID vehicle control systems," *Automatic Control, IEEE Transactions on*, vol. 19, no. 5, pp. 579–581, 1974.

[12] G. Guo and W. Yue, "Sampled-data cooperative adaptive

cruise control of vehicles with sensor failures," 2014.

[13] J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Lp string stability of cascaded systems: Application to vehicle platooning," *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 2, pp. 786–793, 2014.

[14] G. J. Naus, R. P. Vugts, J. Ploeg, M. Van de Molengraft, and M. Steinbuch, "String-stable CACC design and experimental validation: A frequency-domain approach," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 9, pp. 4268–4279, 2010.

[15] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in *American Control Conference, 1990*. IEEE, 1990, pp. 291–296.

[16] D. N. Godbole and J. Lygeros, "Longitudinal control of the lead car of a platoon," *Vehicular Technology, IEEE Transactions on*, vol. 43, no. 4, pp. 1125–1135, 1994.

[17] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 1, pp. 296–305, 2014.

[18] W. van Willigen, M. Schut, and L. Kester, "Evaluating adaptive cruise control strategies in worst-case scenarios," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, p. 1910.

[19] V. Turri, B. Besselink, J. Må rtensson, and K. H. Johansson, "Fuel-efficient heavy-duty vehicle platooning by look-ahead control," *Decision and Control (CDC), IEEE Conference on*, pp. 654–660, 2014.

[20] P. Seiler and R. Sengupta, "An H_∞ approach to networked control," *Automatic Control, IEEE Transactions on*, vol. 50, no. 3, pp. 356–364, 2005.

[21] U. Montanaro, M. Tufo, G. Fiengo, M. di Bernardo, and S. Santini, "On convergence and robustness of the extended cooperative cruise control," *Decision and Control (CDC), IEEE Conference on*, pp. 4083–4088, 2014.

[22] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. Wiley New York, 2007, vol. 2.

[23] K. Lidström, K. Sjoberg, U. Holmberg, J. Andersson, F. Bergh, M. Bjade, and S. Mak, "A modular CACC system integration and design," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1050–1061, 2012.

[24] J. P. Maschuw, G. C. Keßler, and D. Abel, "LMI-based control of vehicle platoons for robust longitudinal guidance," *radio communication*, vol. 11, p. 1, 2008.

[25] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of CACC performance subject to unreliable wireless communication," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). The Hague, The Netherlands: IEEE*, 2013.