Robust cooperative adaptive cruise control design and implementation for connected vehicles

Mark Trudgen, Rebecca Miller and
Javad Mohammadpour Velni*

School of Electrical and Computer Engineering,
College of Engineering,
University of Georgia,
Athens, GA, USA
Email: mtrudgen@uga.edu
Email: zrogles@uga.edu
Email: javadm@uga.edu
*Corresponding author

Abstract: Cooperative adaptive cruise control (CACC) is a developing technology that achieves reduced inter-vehicle distance following in order to increase roadway capacity. CACC provides much lower headway values than adaptive cruise control (ACC) or human drivers can; however, this relies on real-time acceleration data, which inherently has inexact timing. In order to implement a string stable CACC platoon following scheme, meaning that disturbances are attenuated down the stream of the platoon of vehicles, while also overcoming the inherent challenges of wireless communication and uncertain internal model parameters, we design an $\mathcal{H}_\infty$ controller that is robust to all aforementioned uncertainties. We implement this controller on a laboratory-scale test bed with an anti-windup compensation scheme and particularly show that the controller design is able to account for communication shortcomings.

Keywords: cooperative adaptive cruise control; CACC; robust $\mathcal{H}_\infty$ control; anti windup; loop shaping; platoon following; wireless communication; cyber physical systems; CPSs.

Reference to this paper should be made as follows: Trudgen, M., Miller, R. and Velni, J.M. (2018) ‘Robust cooperative adaptive cruise control design and implementation for connected vehicles’, *Int. J. Automation and Control*, Vol. 12, No. 4, pp.469–494.

Biographical notes: Mark Trudgen received his BS and MS degrees in Mechanical Engineering and PhD in Engineering. He joined the University of Georgia as a Research Assistant in the Complex Systems Control Laboratory under the direction of Dr. Velni in August 2013. In January 2017, he joined the University of Georgia as a Lecturer in the School of Electrical and Computer Engineering. His current research is on low-order LPV model development and robust control design for automotive and manufacturing applications.

Rebecca Miller received her MS degree in Engineering and BS degree in Computer Science from the University of Georgia. Her areas of research include power line robotics and cooperative adaptive cruise control. Additionally, she spent several years designing control systems for a pilot project in the alternative fuels industry and worked in the photovoltaic industry designing and installing solar panel systems and accompanying monitoring systems. She has several published papers on these topics.

Copyright © 2018 Inderscience Enterprises Ltd.

Javad Mohammadpour Velni received his BS and MS degrees in Electrical Engineering and PhD in Mechanical Engineering. He joined University of Georgia as an Assistant Professor of Electrical Engineering in August 2012. He has published over 100 articles in international journals and conferences, served in the editorial boards of ASME and IEEE Conferences on Control Systems and edited two books on large-scale systems (2010) and LPV Systems Modeling and Control (2012). His current research is on secure control of cyber physical systems, coverage control of multi-agent systems, and data-driven model learning and control of complex distributed systems.

This paper is a revised and expanded version of a paper entitled 'Robust cooperative adaptive cruise control design for connected vehicles' presented at Proceedings of the ASME 2015 Dynamic Systems and Control Conference, Columbus, Ohio, USA, 28–30 October 2015.

1 Introduction

Traffic congestion is becoming a global issue such that increasing the capacity of existing roads has become very advantageous, especially in urban areas where congestion and traffic growth is the highest (Jones, 2013). Cooperative adaptive cruise control (CACC) is an advanced control technology that safely reduces the allowable *headway* time between vehicles (Ploeg et al., 2011). The smaller inter-vehicle following distances allow for higher traffic throughput without the additional high costs and delays associated with road construction projects. CACC can also provide traffic solutions to roads that are landlocked (Vander Werf et al., 2002). Connected cars are a modern cyber physical system (CPS) where the vehicles in a platoon form a connected system (Holdren et al., 2010). This technology has the potential to be implemented on any car.

CACC technology can provide significant traffic mitigation gains due to its ability to decrease the headway distance a significant amount compared to human drivers and other technologies. As a baseline to compare with human drivers, the *2010 Highway Capacity Manual* shows that human drivers achieved a headway of 1.1 seconds in maximum flow of a multi-lane highway at 60 mi/h (Vander Werf et al., 2002), whereas CACC has shown headway values on the order of half of human drivers (Ploeg et al., 2014a). CACC has also shown significant improvements over its predecessor adaptive cruise control (ACC).

ACC is a longitudinal vehicle following technology that employs radar (or lidar) sensors to measure the distance between two vehicles. An onboard adaptive cruise controller uses the relative distance and velocity to adjust the acceleration of the vehicle in order to position the vehicle appropriately (Xiao and Gao, 2010). ACC is an extension of conventional cruise control (CCC) in that CCC adjusts the acceleration of the car to regulate a desired user defined speed, whereas ACC accelerates (and decelerates) accordingly within a platoon of vehicles to maintain a desired headway (Ploeg et al., 2011). CACC extends the ACC technology by adding inter-vehicle wireless communication (de Bruin et al., 2004). This extension enables the designed controller to achieve smaller *headway* distances, which is critical for platoon technology to have a noticeable impact on traffic mitigation (Shladover et al., 2012). Juxtaposing the two technologies, we see the main drawback of the ACC technology: the smallest stable headway values are too large to have any significant impact on traffic mitigation (Vander

Werf et al., 2002). This justifies the need for CACC technology as it is able to achieve headway values that have significant impacts on traffic mitigation (Jones, 2013; Shladover et al., 2012).

The minimum achievable headway values are governed by an important metric commonly referred to as string stability (Ploeg et al., 2014a). The concept of string stability was first introduced in Levine and Athans (1966) and later extended in Peppard (1974), which led to the development of systems using the nearest neighbour as a measurement. String stability is essential to ensuring safety as it guarantees that all disturbances introduced in the string be attenuated as they propagate in the downstream direction (Peppard, 1974; Ploeg et al., 2011). Disturbances added to the string in any location should be mitigated. The presence of disturbances can create an increase fuel consumption, so-called *ghost traffic jams*, or even in extreme cases, an accident (Guo and Yue, 2014). Therefore, it is advantageous to use a controller design framework that can explicitly account for string stability in the controller synthesis. Designing for string stability prevents 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 (Klinge and Middleton, 2009); however, these methods lack the consideration of a measure of performance as seen in Naus et al. (2010) and Ploeg et al. (2014b), which give a frequency-domain approach for controller synthesis.

Platoon controller design is seen in the literature given the advancement of technology from CCC to present day CACC. A first-principle model of the vehicles motion is described in Godbole and Lygeros (1994) and Sheikholeslam and Desoer (1990) as a third-order nonlinear model. For the purposes of controller design, a common technique seen in the literature is to linearise this nonlinear model via the feedback linearisation method. Recent works have designed basic ACC and CACC controllers from the linearised model showing experimental validity (de Bruin et al., 2004; Milanés et al., 2014; Ploeg et al., 2011). The authors in Guo and Yue (2014) developed a sampled data approach to CACC design specifically to account for the presence of sensors and actuator failures. The authors in van Willigen et al. (2011) studied CACC in worst-case driving scenarios such as braking. Optimisation of fuel economy for platoons is a relevant issue, and as such model predictive control (MPC) has also gained attention (Li et al., 2011). An MPC-based CACC approach was designed for heavy duty vehicles, such as tractor trailer trucks (Turri et al., 2014), 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 system where the effects of sampling, hold, and network delays can be taken into account. An $\mathcal{H}_\infty$ formulation of network controlled problems is given in Seiler and Sengupta (2005). Still, other works have investigated time-varying communication delays and communication structures beyond the classical architecture as in Montanaro et al. (2014).

This paper builds from the authors preliminary work reported in Trudgen and Mohammadpour (2015), where a controller robust only to the internal model parameters was validated through simulations. We extend the $\mathcal{H}_\infty$ controller design framework so that it is robust to uncertainties in both the acceleration received from the wireless communication and the internal vehicle parameters using the induced energy-to-energy gain (or $\mathcal{H}_\infty$ norm) due to the presence of uncertainty (Skogestad and Postlethwaite, 2007). Robustness to any delays in wireless communication is critical in small headway applications; this is seen and validated through experimental tests done on a laboratory test bed. Given the inherent limitations of the laboratory test bed, we include an anti-windup controller (Azar and Serrano, 2015) that also optimises the $\mathcal{L}_2$-gain performance.

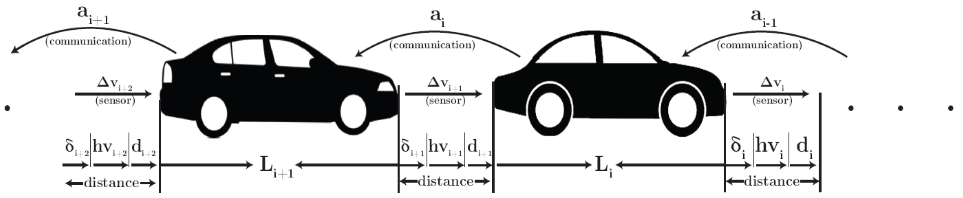
This paper is structured as follows: Section 2 describes the basics of CACC, starting with the nonlinear model seen in Godbole and Lygeros (1994) and Sheikholeslam and Desoer (1990) and arrives at a CACC design problem using the linearised model. In Section 3, we represent uncertainty in the CACC design framework and describe an anti-windup compensator design. Section 4 describes our laboratory-scale test bed that is used to appropriately model the CACC driving and quantifies the associated parameters. Section 5 presents simulation results for both our robust design and a nominal controller design, and these results are then validated using our test bed. Section 6 draws conclusions.

2 CACC technology

Figure 1 shows a typical string of vehicles equipped with CACC. The arrangement and design of various longitudinal ACC strategies has been studied in the literature (see, e.g., (Xiao and Gao, 2010) and references therein). In a CACC scheme, the lead car determines the trajectory 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. Position sensing is typically done via the use of radar (or lidar) technology (Ploeg et al., 2011). In a platoon, the distance between vehicles is broken into three 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 $\delta_i = 0$. More specifically, δ_i, the spacing policy, is given as (Guo and Yue, 2014).

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

Figure 1 A string of vehicles equipped with CACC technology



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 (Godbole and Lygeros, 1994; Sheikholeslam and Desoer, 1990).

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

where g_i is given as

$$g_i = \frac{1}{\tau_i m_i}, \tag{3}$$

where m_i represents the i^{th} vehicle's mass, and τ_i is the engine time-constant of the i^{th} vehicle. The above model is nonlinear due to the nonlinear function $f_i(v_i, a_i)$ described as

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

where $\frac{\tau_i A_i c_{di}}{2m_i}$ 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 linearise the above nonlinear system dynamics, the following control law is adopted (Godbole and Lygeros, 1994; Sheikholeslam and Desoer, 1990)

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

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 linearisation, 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 induced 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 (Guo and Yue, 2014)

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

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 (de Bruin et al., 2004; Lidström et al., 2012; Ploeg et al., 2011) in assuming a low-level linearising feedback controller. The system in (8) gives the linearisation for the i^{th} vehicle, and the overall system is hence a decentralised platoon.

2.1 Block diagram representation for connected vehicles

Assuming the linearised plant dynamics, we can cast the CACC design problem into a block diagram representation. 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 *desired* acceleration (that is used as an input to the linearising controller, see, e.g., (Guo and Yue, 2014)). Finally, d_i denotes an added static following distance, and L_i is the length of the i^{th} vehicle. In addition, $G_i(s)$ represents the system transfer function, and $H(s)$ describes the spacing policy given as

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

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

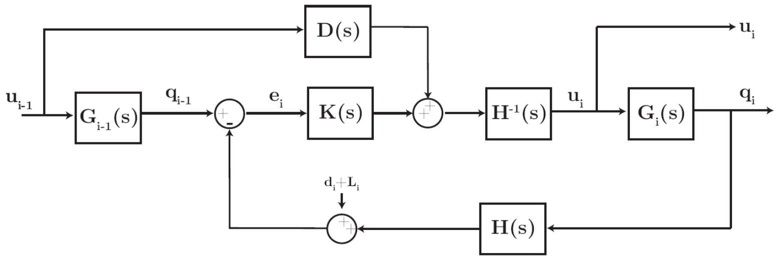
where τ_i is the engine time constant, ϕ_i is the internal time delay, and h is the headway. We note that the linearised model adequately describes the dynamics provided that linearising acceleration controller takes into account parameters given in (5) (Ploeg et al., 2011).

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 represented in the frequency domain as

$$D(s) = e^{-\theta s}, \tag{11}$$

where θ is the delay associated with the wireless communication (Ploeg et al., 2011, 2013). By adding a stabilising controller, $K(s)$, and using the communicated acceleration as a feedforward term, the block diagram given between $u_{i-1}(s)$ and $u_i(s)$ is shown in Figure 2. Without the loss of generality, $L_i = d_i = 0$ can be assumed.

Figure 2 CACC block diagram



2.2 String stability

From the configuration shown in Figure 2, we define the transfer function from u_{i-1} to u_i as

$$\Gamma_{CACC}(s) = \frac{1}{H(s)} \frac{D(s) + G_{i-1}(s)K_i(s)}{1 + G_i(s)K_i(s)}. \tag{12}$$

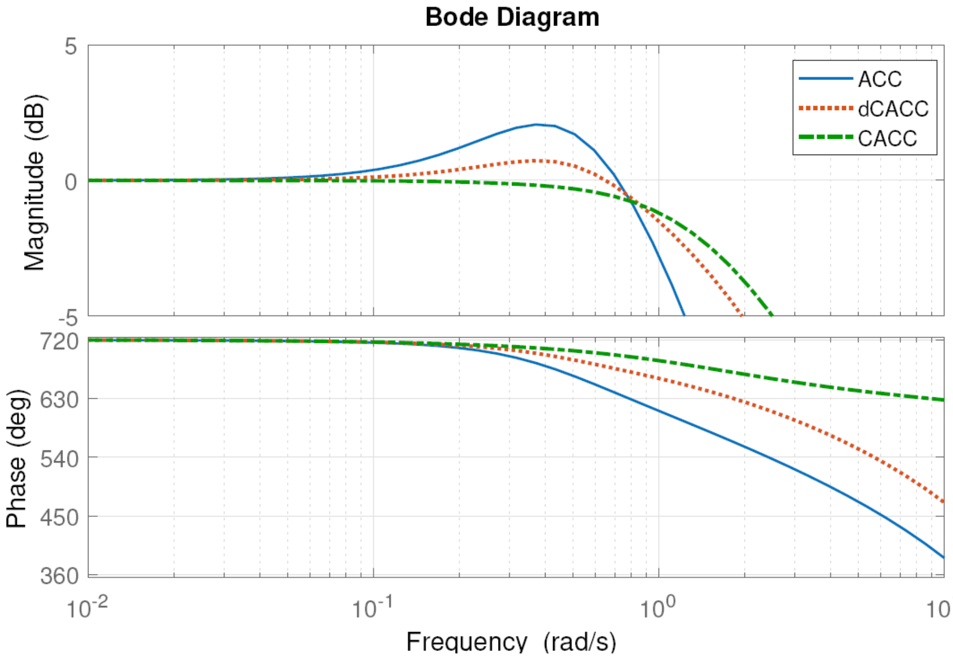
However, as in Ploeg et al. (2014a) we will assume homogeneity in the string such that $G_{i-1}(s) = G_i(s)$. A complete discussion of string stability is given in Ploeg et al. (2014b). With respect to (12) it suffices to limit the discussion of string stability to ensuring that

$$\left\| \Gamma_{CACC}(j\omega) \right\|_{\mathcal{H}_\infty} \leq 1, \tag{13}$$

where $\|\cdot\|_{\mathcal{H}_\infty}$ denotes the system's $\mathcal{H}_\infty$ norm, is met. We recall that the overall goal is to reduce the *headway* while remaining string stable. As noted in Shladover et al. (2012) for this technology to have a noticeable impact on traffic mitigation, a headway significantly smaller than 1.1 sec must be realised as this is the average headway achieved by human drivers (Jones, 2013).

From Figure 2, we see that setting $D(s) = 0$ would result in an ACC scheme. Other smart cruise control schemes exist; the authors in Ploeg et al. (2013) used an onboard estimator to estimate u_{i-1} as a way to gracefully transition between CACC and ACC schemes. To validate the need of CACC versus the previously mentioned ACC and so-called 'dCACC' schemes, we designed a proportional-derivative (PD) controller with a headway value of $h = 0.6$ sec and tested the three schemes. Figure 3 shows the corresponding Bode plots of the three schemes, again showing the need for CACC. It is seen from Figure 3 that the additional complexity needed in CACC implementation is justified given the ability to achieve lower headway values. It is also seen from Figure 3 that an onboard estimate of the previous vehicle's acceleration cannot guarantee string stability as effectively as CACC can.

Figure 3 Frequency response associated with ACC, dCACC and CACC for a headway of $h = 0.6$ sec (see online version for colours)



3 Robust CACC design

Now, to cast the CACC system into the robust $\mathcal{H}_\infty$ control design framework, we first introduce the sources of uncertainty and describe how to quantify them in this section.

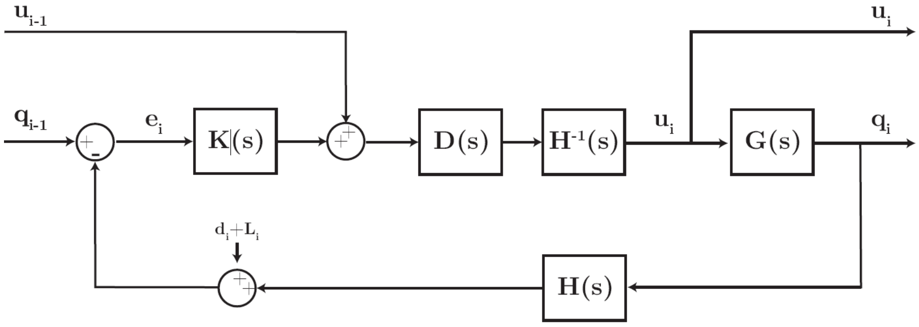
3.1 Sources of uncertainty

With respect to our CACC application, there are several reasons to incorporate robustness into a control design framework. We expect uncertainty in all our identified parameters, as the parameters are only approximately known. In our laboratory-scale test bed described in Section 4, τ will vary based on the battery voltage level and the wear on the motor windings. Also, linear models are only adequate for a small operating range, and original measurements taken to find parameters have inherent errors despite calibration. Since we consider a lumped parameter model, the exact value of τ for each motor is unknown. As we developed our model through system identification methods, at high frequencies the structure of the model can become unknown and uncertainties in parameters always arise. We can also expect some uncertainties within the controller (Skogestad and Postlethwaite, 2007).We also expect varying time delays due to the computations and the communication packet-handling protocol.

3.2 CACC block diagram reformulation

In order to account for the acceleration delay, due to the wireless communication, we reformulate the block diagram shown in Figure 2 to Figure 4. First, we introduce the same delay, $D(s)$, to the controller output as experienced by the communication delay. While delays are parasitic to controller design and should as a general rule be avoided, the means justify the end as this block diagram reformulation now allows us to account for communication delay in our controller design which is critical to CACC.

Figure 4 CACC block diagram reformulation



From Figure 3 we recall that only CACC is able to achieve string stability at low headway values. In typical communication schemes, data loss and an uneven communication time is to be expected. If there is not a dedicated microcontroller for communication, as in our laboratory test bed, this non-uniform delay is even more common. By designing a robust controller we are able to withstand the delays and minor packet losses without having to pull back the CACC scheme into either a dCACC or ACC scheme to maintain string stability with a low headway value. This is crucial as low headway values are needed to realise traffic congestion gains (Holdren et al., 2010; Shladover et al., 2012). The issue of implementation where the controller and the communication are synced to share the same delay timing is achieve via microcontroller programming. *In fact, the main novelty in this work is the introduction of robust control to account for varying parameters and a reformulation of Figure 2 to account for all uncertain parameters.*

3.3 Representing uncertainty in the CACC design framework

For the block diagram shown in Figure 4, the parameters ϕ, τ, and θ are assumed to have the nominal values of $\overline{\tau}$ = 0.14 sec., $\overline{\phi}$ = 0.018 sec., and $\overline{\theta}$ = 0.02 sec. for our test bed. We also consider a variation in the range of $\tau \in [0.10, 0.17]$, $\phi \in [0.018, 0.10]$, and $\theta \in [0.02, 0.10]$. To guarantee the closed-loop system stability in the presence of the model uncertainty associated with the above uncertain parameters, we first represent the lumped parameter *multiplicative uncertainty* as shown in Figure 5 and equation (14).

$$G_p(s) = G_0(s)\big(1 + W_p(s)\Delta_p(s)\big), \tag{14}$$

where $G_p(s)$ represents the perturbed model, $G_0(s)$ represents the nominal model, $\|\Delta_p\|_\infty \leq 1$, and W_p represents the lumped uncertainties transfer function that satisfies (Skogestad and Postlethwaite, 2007)

$$\left|\frac{G_p(j\omega) - G_0(j\omega)}{G_0(j\omega)}\right| \leq \left|W_p(j\omega)\right|, \tag{15}$$

for any frequency ω. We then let τ, ϕ and θ 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 6, where in (15), G_p(s) is taken as the perturbed plant and $G(s)$ is fixed as the plant designed at $\overline{\tau}, \overline{\phi}$, and $\overline{\theta}$. Then, a filter, $W_p(s)$, was fitted to the Bode plot according to (15). This results in the following high-pass filter

$$W_p(s) = \frac{2.85s}{s+7}. \tag{16}$$

Figure 5 Lumped parameter multiplicative uncertainty

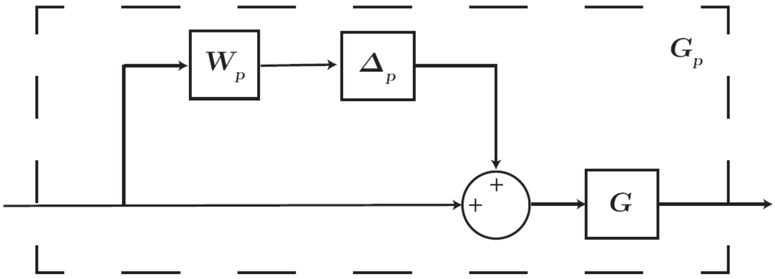


Figure 6 Bode plots to find the multiplicative uncertainty weight (see online version for colours)

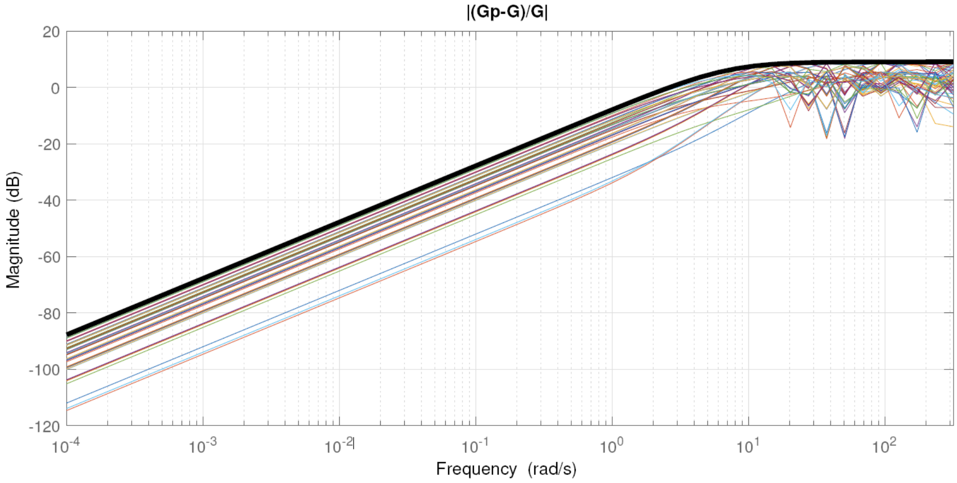
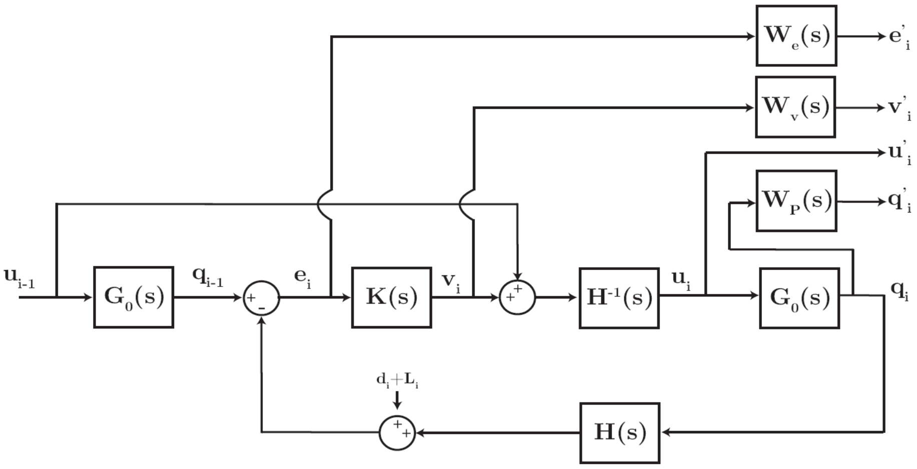


Figure 7 Configuration of the closed-loop control system



3.4 *Loop shaping for H_∞ control design*

Next, we use the *S*/*KS* loop shaping approach (Skogestad and Postlethwaite, 2007) to design a robust controller that can guarantee tracking with zero steady-state error and a low control effort for a headway value of 0.35 sec. The value $h = 0.35$ sec was chosen as it is significantly lower than the human comparison of $h = 1.1$ sec. This headway allows us to achieve both string stability and robustness. The corresponding block diagram in Figure 7 depicts how disturbances and noise signals affect the closed-loop system. Using this block diagram setup, the string stability requirement can be directly handled within the $\mathcal{H}_\infty$ framework. In standard loop shaping, weight W_e is tuned to penalise tracking error at low frequencies. The weight W_e is selected to be a low pass filter, tuned to eliminate the steady-state error. Weight W_v is tuned to penalise controller output, and is selected to be a high pass filter. The filters are tuned as

$$W_e(s) = 5 \cdot \frac{0.075}{s+0.9}, \tag{17}$$

$$W_v(s) = 0.4 \frac{s}{s+50}. \tag{18}$$

Next, we select the *desired* acceleration, u_i, as an exogenous output signal Ploeg et al. (2014a). 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

$$T_i(s) = \frac{u_i(s)}{u_{i-1}(s)}. \tag{19}$$

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. Note that as in (Skogestad and Postlethwaite, 2007), this formulation of W_p allows us to capture the variation in delays in our system as uncertainty inside W_p.

3.5 H_∞ robust controller design for connected vehicles

After selecting the loop shaping weights, we use MATLAB to represent the system interconnection shown in Figure 7 into the linear fractional transformation (LFT) form. This is done by using the MATLAB command *sconnect*. Next, we express the closed-loop system as

$$z(s) = N(s)w(s), \tag{20}$$

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 (Skogestad and Postlethwaite, 2007). Now, by imposing the following requirement that

$$\|N(j\omega)\|_\infty \leq 1, \tag{21}$$

string stability would be guaranteed. Next, the robust control design problem is solved by invoking the MATLAB command *hinflmi*. The controller obtained matches the model order of the plant, and using model order reduction methods we reduce the order of the controller to fourth order. A comparison of the Bode plots of the reduced-order versus full order controller shows a good approximation over all frequencies. Next, we analyse the closed-loop properties of the system. Figure 8 illustrates that string stability requirement is met. We see from Figure 9 that we also achieve closed-loop robust stability. This condition is given as $||W_p(j\omega)T(j\omega)|| \leq 1$ for any ω.

Figure 8 String stability of the closed-loop system (see online version for colours)

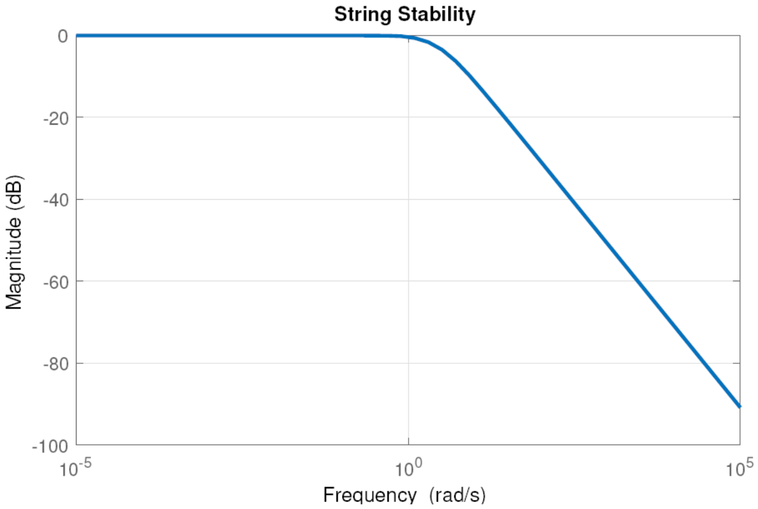


Figure 9 Robust stability of the closed-loop system (see online version for colours)

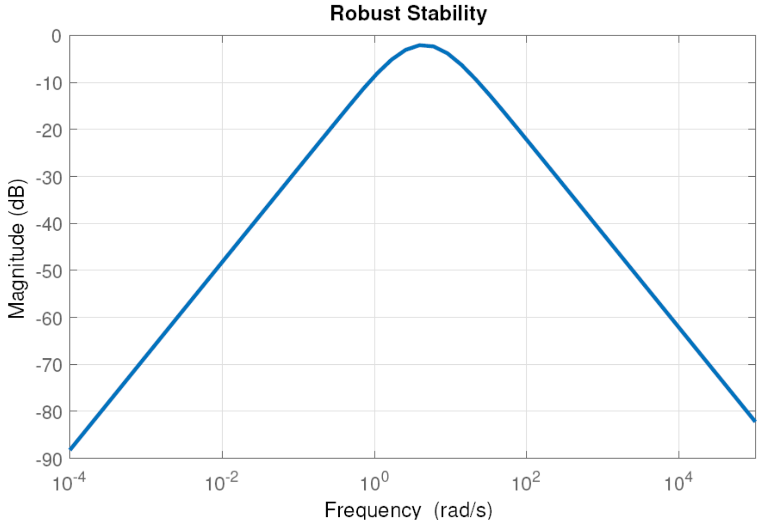


Figure 10 shows the welcomed result that we have also achieved a level of robust performance in our design.

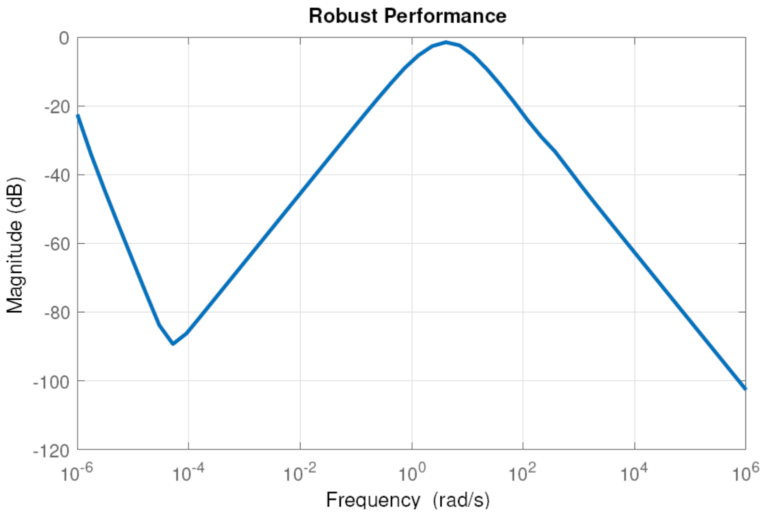
Interestingly, we notice from Figures 2 and 4 that the relationship

$$\frac{u_i(s)}{u_{i-1}(s)} = \frac{e_i(s)}{e_{i-1}(s)}, \tag{22}$$

holds true. For our reformulated block diagram in Figure 4 we rederive

$$\frac{e_i(s)}{e_{i-1}(s)} = \frac{1}{H(s)} \frac{D(s)+D(s)G(s)K(s)}{1+D(s)G(s)K(s)}. \tag{23}$$

Figure 10 Robust performance of the closed-loop system (see online version for colours)



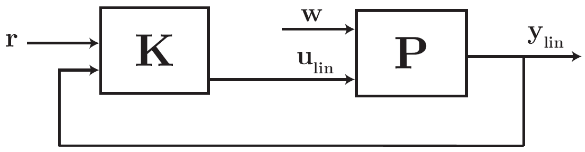
3.6 Anti-windup compensator design

To account for the actuator saturation that exists in our experimental test bed due to the hard limits on actuators, we use the procedure given in Wada and Saeki (1999) to augment our CACC system with an anti-windup controller. First, we consider the CACC block diagram of Figure 7 as a general feedback system without saturation as

$$y_{lin} = \mathbf{P}\begin{bmatrix} w \\ u_{lin} \end{bmatrix}, \quad u_{lin} = \mathbf{K}\begin{bmatrix} r \\ y_{lin} \end{bmatrix}, \tag{24}$$

where $u_{lin} \in \mathcal{R}^{n_u}$, $y_{lin} \in \mathcal{R}^{n_y}$, $r \in \mathcal{R}^{n_r}$, and $w \in \mathcal{R}^{n_w}$. The block diagram of the general feedback system without saturation is shown in Figure 11.

Figure 11 General feedback system



With respect to Figure 11 we partition the plant, $\mathbf{P}(s)$, as

$$\mathbf{P} \triangleq [P_1, P_2] = \left[\begin{array}{c|cc} A_p & B_{p1} & B_{p2} \\ \hline C_p & D_{c1} & 0 \end{array}\right]. \tag{25}$$

Likewise, with respect to Figure 11 we partition the controller, $\mathbf{K}(s)$, as

$$\mathbf{K} \triangleq [K_1, K_2] = \left[\begin{array}{c|cc} A_c & B_{c1} & B_{c2} \\ \hline C_c & D_{c1} & D_{c2} \end{array}\right]. \tag{26}$$

We choose to use the formulation in Wada and Saeki (1999) because it is assumed that $K(s)$ has already been designed to guarantee closed-loop stability of (27) and to achieve desired performance metrics. Next, we add an anti-windup compensation scheme as

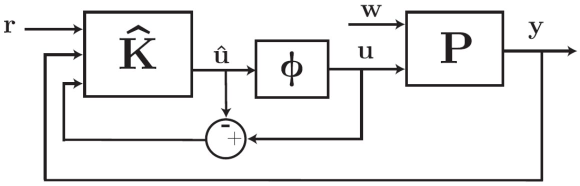
$$y = \mathbf{P}\begin{bmatrix} w \\ u \end{bmatrix}, \quad u = \phi(\hat{u}), \quad \hat{u} = \hat{\mathbf{K}}\begin{bmatrix} r \\ y \\ u = \hat{u} \end{bmatrix}, \tag{27}$$

where $u,\ \hat{u} \in \mathcal{R}^{n_u}$, $y_{lin} \in \mathcal{R}^{n_y}$. Now, ϕ is given as the saturation function and ψ denotes a deadzone function, both are defined as

$$\phi(v) \triangleq \begin{cases} a \cdot \operatorname{sgn}(v), & |v| > a \\ v, & |v| \le a \end{cases}, \quad \psi \triangleq v - \phi(v), \tag{28}$$

where a is used to characterise the saturation. Using (27) and (28), Figure 12 illustrates a general feedback system with saturation.

Figure 12 General anti-windup system



Now the controller, $\mathbf{K}(s)$, is augmented with feedback from the saturation block, and the augmented controller, $\hat{\mathbf{K}}(s)$, is defined as

$$\hat{\mathbf{K}} \triangleq [K_1, K_2, K_3] = \left[\begin{array}{c|ccc} A_c & B_{c1} & B_{c2} & \Lambda_1 \\ \hline C_c & D_{c1} & D_{c2} & \Lambda_2 \end{array}\right]. \tag{29}$$

The transfer functions of K_1 and K_2 are the same as (29) and Λ_1 and Λ_2 are constant matrices introduced for anti-windup compensation. To solve for Λ_1 and Λ_2, the authors in Wada and Saeki (1999) used a linear matrix inequality (LMI) based formulation to guarantee that

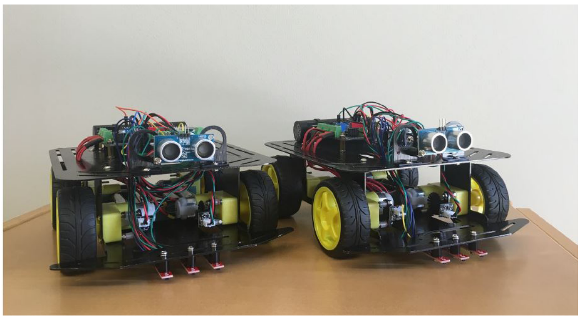
$$\|y - y_{in}\|_{\mathcal{L}_2} \le \|\psi(U_{lin})\|_{\mathcal{L}_2}, \tag{30}$$

where y represents the output of the plant without the saturation, y_{in} represents the output of the plant with the saturation, (u_{lin}) represents a system between the linear and nonlinear systems, and $\|\cdot\|_{\mathcal{L}_2}$ denotes the vector's two-norm. We solve the LMI condition to determine Λ_1 and Λ_2 and augment our CACC system with the anti-windup compensator.

4 Our laboratory-scale CACC test bed

In considering the selection for our test bed, we see from (9) that a simple first-order ordinary differential equation can accurately represent the dynamics of the CACC problem. Indeed, Ploeg et al. (2014a) verifies that the measured step response of an acceleration controlled test vehicle is well modelled by (9). Using this knowledge we built a laboratory test bed model. Figure 13 shows a picture of our test bed.

Figure 13 CACC test bed (see online version for colours)



4.1 Test bed hardware

For our laboratory test bed, we modified a DFRobot Baron 4WD Mobile platform to meet our needs. Various technologies and sensors were added to realise the CACC concepts:

- *Drivetrain:* the onboard DC motors are rated at 6 V with a no-load current of 71 mA and a stall current of 470 mA. The gear ratio is 1.120 and the torque is 1.92 kg•cm. Through testing, the maximum voltage supplied to the motors from the microcontroller is found to be 4.1 V. The lowest voltage required to turn the motors under the weight of the vehicle is 1.5 V. The velocities corresponding to the maximum and minimum voltages are 0.4 m/s and 0.14 m/s, respectively. Although scaled, these motor nonlinearities pose the same control related conceptual issues as that of a full scale vehicle.
- *Microcontroller:* the Romeo V2.2 (R3) is an expanded microcontroller printed circuit board and it is programmable using the Arduino IDE software.
- *Communication:* we chose to use XBee series 1 (S1) RF modules to communicate with each other. The XBee's implement the IEEE 802.15.4 protocol within the 2.4 GHz frequency band to communicate. The radios transmit data at a rate of 250,000 bits per second (b/s), and testing proved a serial interface data rate of 57,600 b/s to be the fastest reliable speed.
- *Wheel encoders:* the wheel encoders use a non-contact, optical method to track the rotation of the wheels. We doubled the resolution of the original disk to a final resolution of 0.5 cm.

- *Proximity sensor:* to detect the distance between the vehicles we used the Paralax PING Ultrasonic proximity sensor. This sensor works by emitting a 40 kilohertz ultrasonic burst and providing an output pulse that corresponds to the time it takes to receive an echo.
- *Accelerometer:* an ADXL335 triple-Axis accelerometer was installed to measure onboard acceleration data.
- *Infrared line following sensors:* to mitigate the problem of longitudinal control, we installed infrared line following sensors.

4.2 System identification

For the purpose of creating a CACC test bed that is modelled by (9) and (11), we perform system identification to find the internal time delay ϕ, the nominal communication delay θ, and the time constant τ.

4.2.1 Step response

To identify our test bed parameters we applied a step function input. To do this, we exploit the well known relation that voltage and velocity are related through a constant such that $Velocity = \alpha \bullet Voltage$ where α is given in $\frac{m}{V \cdot s}$. We then command the onboard microcontroller to a maximum step of 4.1 V. Figure 14 shows a plot of the corresponding interpolated acceleration data.

Figure 14 CACC acceleration data collected from our test bed

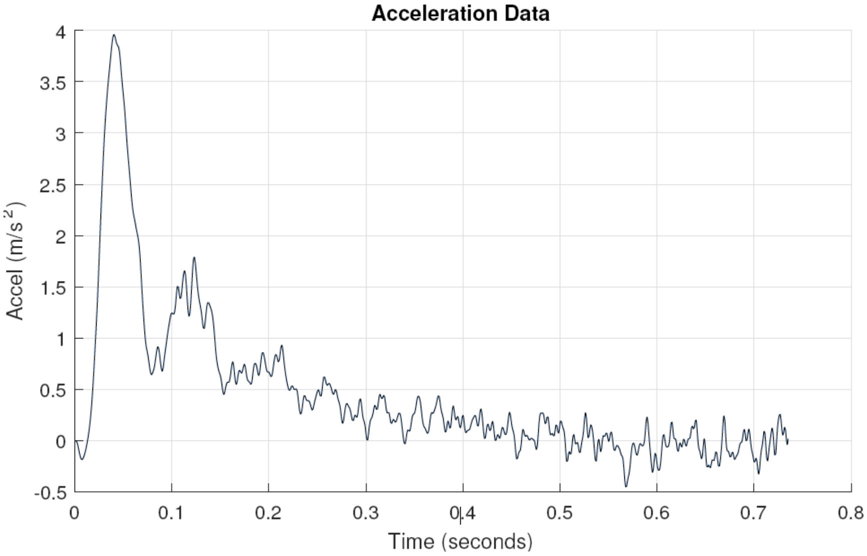


Figure 15 CACC test bed step response (see online version for colours)

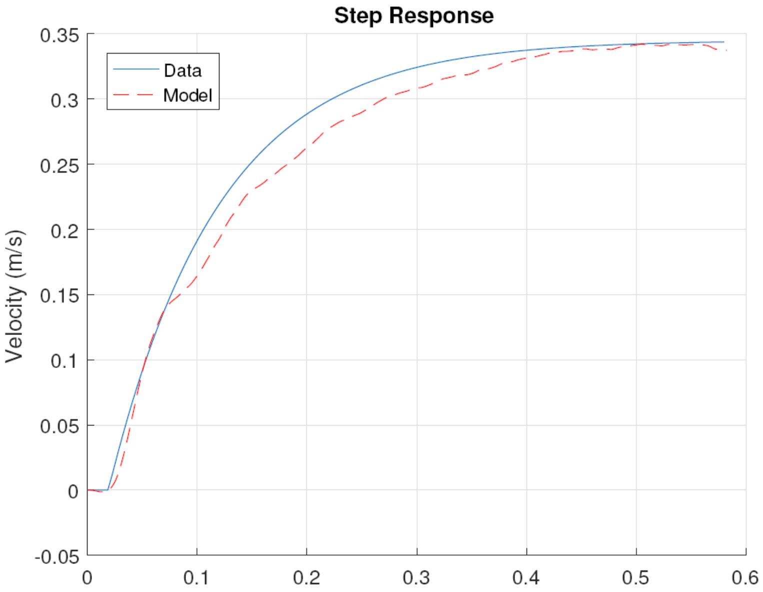


Figure 15 gives the corresponding velocity response. Through experimental testing we find $\alpha = 10.32 \frac{m}{V \cdot s}$. Next, we fit the model with the unknown parameters to the data. Writing a transfer function from desired acceleration to position as in (9) yields

$$G_{sys}(s) = \frac{q_i(s)}{u_i(s)} = \frac{0.0963}{s^2(0.14s+1)} e^{-0.018s}. \tag{31}$$

4.2.2 Estimating the communication delay

To calculate the communication delay, first we establish the best and worst case scenarios. The propagation delay, i.e., the time the data takes to travel through the air, is on the order of nanoseconds and is negligible in this case. In the best case scenario, the carrier channel is clear to send and the total communication delay is 1.17 ms. In the worst case scenario, we consider packet delivery is successful but takes the longest amount of time for channel assessment prior to transmission. In this scenario, the communication delay is 9.44 ms. Therefore, the communication delay is the inclusive set of $\in$ = [1.17 ms, 9.44 ms].

5 Simulation and experimental results

We first created a five-car platoon simulation model using the reduced-order robust controller. We modelled the cars we have in our laboratory test bed in MATLAB/Simulink taking note to also include the corresponding saturations and Coloumb friction characteristics.

5.1 Simulation results

First, we utilised our simulation model to examine the closed-loop performance with the reduced-order robust controller and nominal plant parameters. Figure 16 shows the corresponding error responses, and Figure 17 shows the corresponding velocity responses.

Figure 16 Error response of the five-car simulation model using nominal system parameters (see online version for colours)

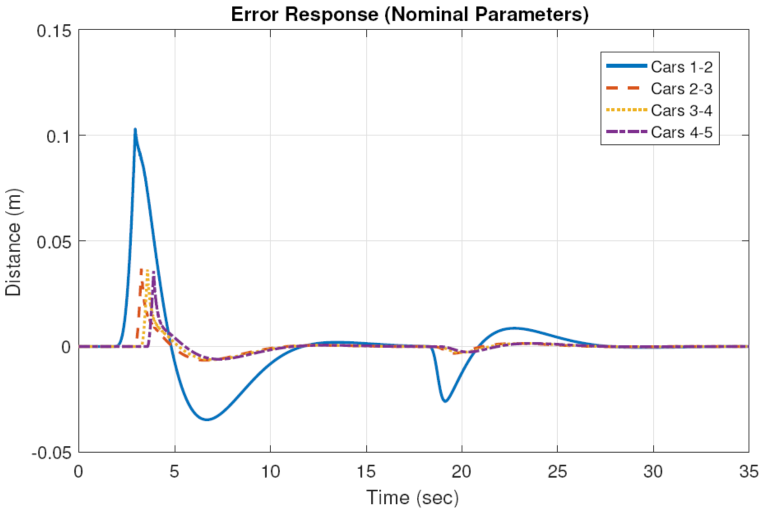
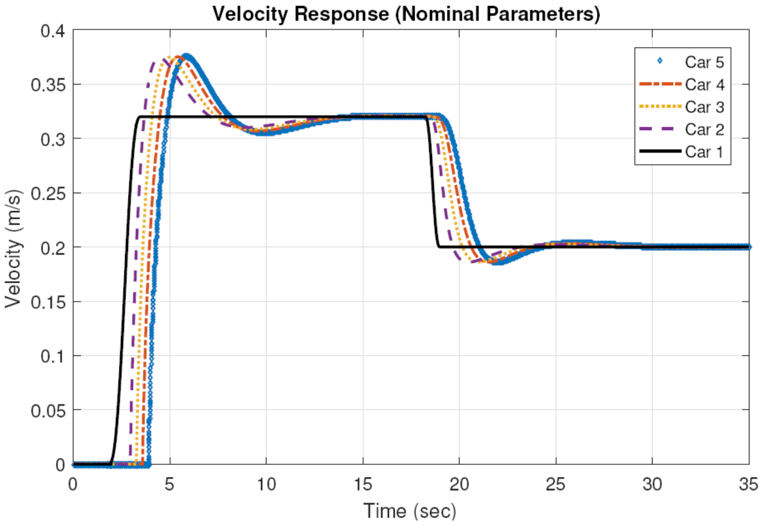


Figure 17 Velocity response of the five-car simulation model using nominal system parameters (see online version for colours)



In Figure 16, a positive error value indicates conservative behaviour, whereas, a negative value indicates that the following car is too close. From Figure 17, we observe that all cars start from rest, and the lead car follows a smooth trajectory. The following cars are tuned to allow overshoot in lieu of a faster settling time. After the lead car has reached 0.32 m/s, it undergoes a sharp deceleration. This is reflected by the error going negative, but the controller quickly reestablishes a zero steady-state tracking error (see, Figure 16). Next, we perturb both ϕ and θ to their respective maximum values. The resulting simulation outputs are given in Figures 18 and 19.

Figure 18 Error response for the five-car simulation model using perturbed system parameters (see online version for colours)

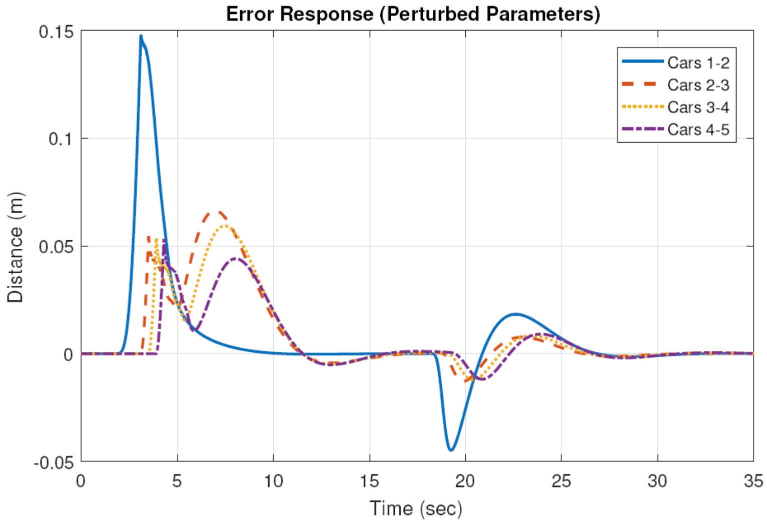
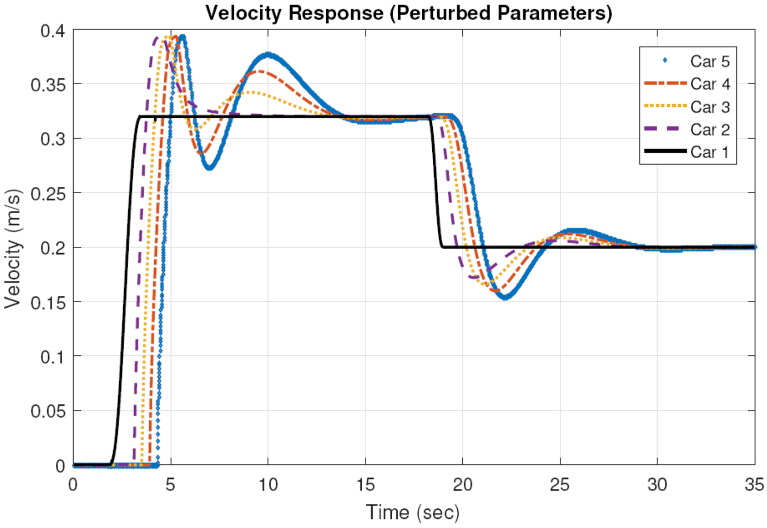


Figure 19 Velocity response for the five-car simulation model using perturbed system parameters (see online version for colours)



We see from the comparison of Figures 16 and 17 with Figures 18 and 19 that added delays cause an increase in error, but that the robust controller is still able to provide the steady-state tracking over the region of parameter perturbations.

5.2 Experimental results

Next, using our test bed, we programmed the lead car to follow the smooth trajectory of the five-car simulation. We discretise the blocks in Figure 4 with a 20 msec sampling time and implement on the laboratory test bed.

5.2.1 Robust Controller

Figures 20 and 21 show the test bed results using the robust controller with nominal parameters.

Figures 20 and 21 show a comparison to Figures 16 and 17. Similarly, we perturbed both ϕ and θ to their respective maximum values and reran the experiments; the results are given in Figures 22 and 23.

Figure 20 Error response of the experimental test bed using nominal parameters (see online version for colours)

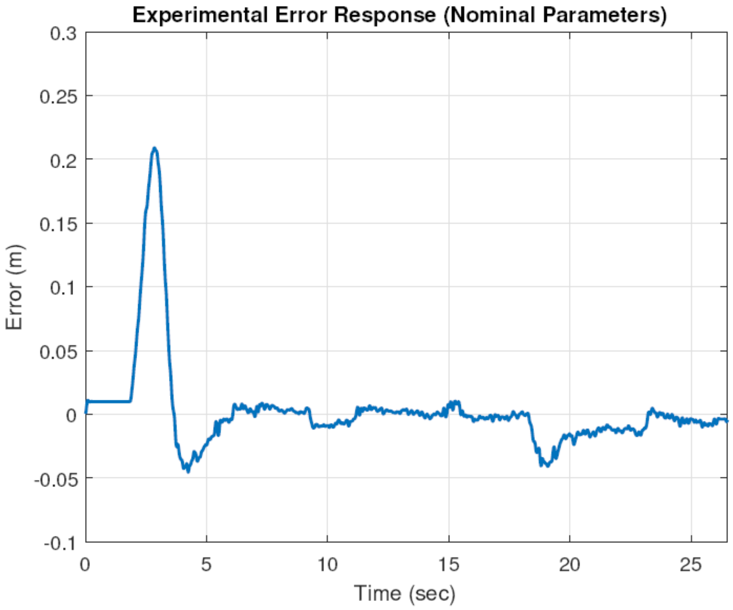


Figure 21 Velocity response of the experimental test bed using nominal parameters (see online version for colours)

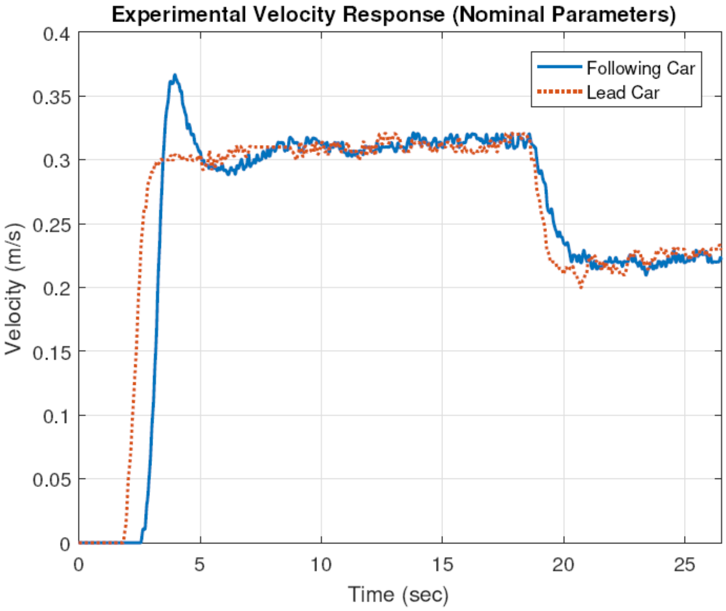


Figure 22 Error response of the experimental test bed using perturbed parameters (see online version for colours)

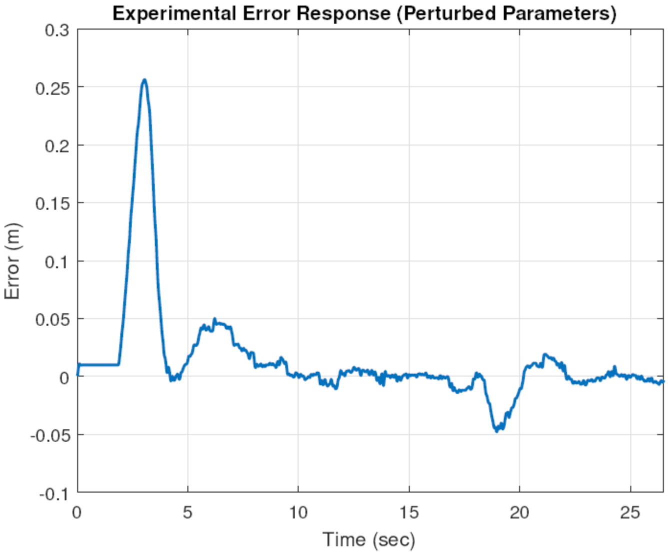
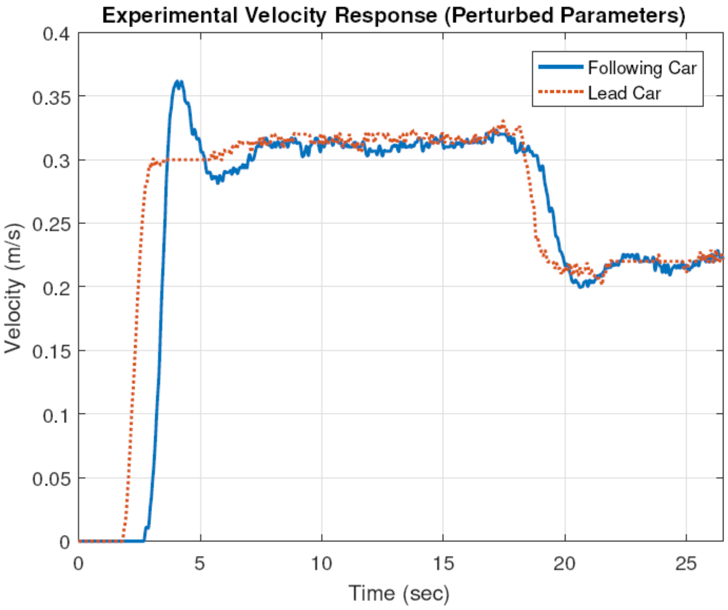


Figure 23 Velocity response of the experimental test bed using perturbed parameters (see online version for colours)



We observe that Figures 22 and 23 also show a good comparison with the results in Figures 18 and 19 in that the robust controller still maintains a high level of performance despite the parameter variation. This is critical in the CACC application since we see that despite the parameter variations, the robust controller allows for significant headway impact (h = 0.35 sec. vs. h = 1.1 sec.) while maintaining string stability. We note that all the experimental testing show an increase of ~0.1 m on the initial acceleration when compared to the simulation results. This consistent discrepancy is due to the initialisation needed in the onboard microcontroller.

5.2.2 Nominal controller

Next, we compare our results to a nominal $\mathcal{H}_\infty$ controller designed around the nominal parameters: $\overline{\tau}, \overline{\phi}$ and $\overline{\theta}$. For the experimental case of a nominal controller using nominal parameters we achieve performance gains over the robust controller design shown in Figures 20 and 21. However, when the nominal $\mathcal{H}_\infty$ controller is experimentally tested using the perturbed case, we observe unacceptably large overshoots illustrated in Figures 24 and 25.

Comparison of Figure 24 with Figure 22 shows that the nominal controller vastly overshoots five times worst than the robust controller, and hence disturbances are magnified instead of attenuated. Table 1 gives a comparison of the root mean square (RMS) values of the deceleration event. We see the nominal controller's RMS value of error doubles, but that the robust controller remains small. This validates the need for the (proposed) robust controller design, given the overall need of minimising headway distance in traffic situations.

Figure 24 Error response of the experimental test bed with the nominal controller and perturbed parameters (see online version for colours)

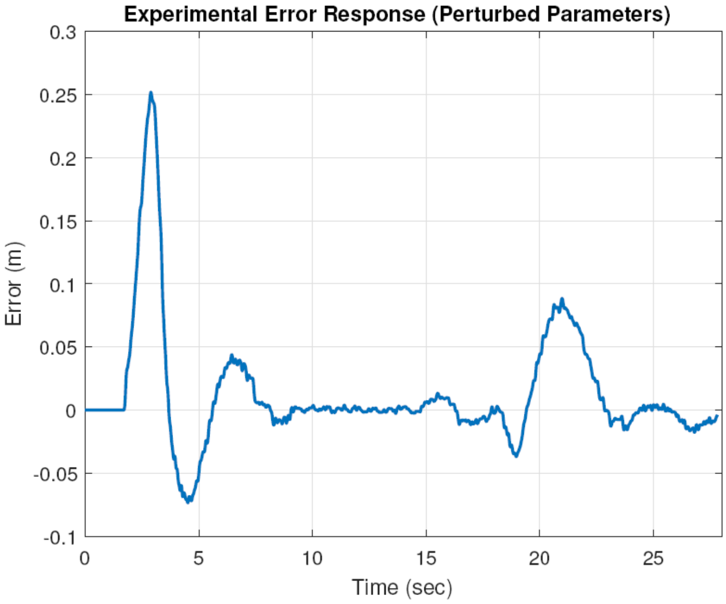


Figure 25 Velocity response of the experimental test bed with the nominal controller and perturbed parameters (see online version for colours)

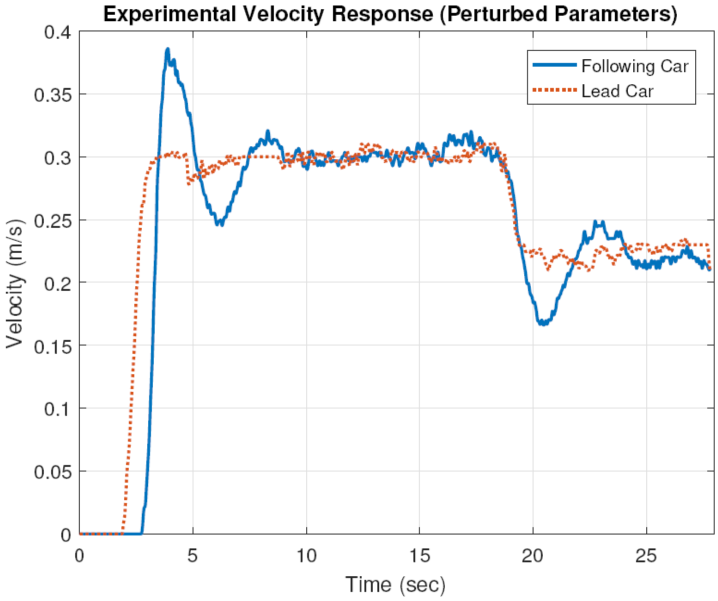


Table 1 RMS value comparisons for experimental data where *n* represents nominal and *p* represents perturbed.

Robust (n)	*Robust (p)*	*Nominal (n)*	*Nominal (p)*
0.0113	0.0117	0.0104	0.0250

6 Conclusions

In this paper, a CACC platoon following scheme is designed so that acceleration delays sourcing from the wireless communication can be included in the controller design. Furthermore, the delays and parameter uncertainties seen inside the CACC model are addressed by employing the mixed-sensitivity, loop shaping-based $\mathcal{H}_\infty$ control design to guarantee string stability. A laboratory test-bed design and application is included, for which we describe the quantification of the uncertainties in the plant model. Simulation results substantiate that the robust controller achieves string stability and tracking performance over the region of parameter perturbations for a low headway value of h = 0.35 sec. This is validated using the experimental test-bed, and the experimental results for a nominal $\mathcal{H}_\infty$ controller are run showing the need for a robust controller design.

Acknowledgements

This research was supported in part by a grant from the University of Georgia Research Foundation, Inc.

References

Azar, A.T. and Serrano, F.E. (2015) 'Design and modeling of anti wind up PID controllers, in *Complex System Modelling and Control Through Intelligent Soft Computations*, pp.1–44, Springer, Springer International Publishing.

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

Godbole, D.N. and Lygeros, J. (1994) 'Longitudinal control of the lead car of a platoon', *IEEE Transactions on Vehicular Technology*, November, Vol. 43, No. 4, pp.1125–1135.

Guo, G. and Yue, W. (2014) 'Sampled-data cooperative adaptive cruise control of vehicles with sensor failures', *IEEE Transactions on Intelligent Transportation Systems*, December, Vol. 15, No. 6, pp.2404–2418, IEEE.

Holdren, J., Lander, E. and Varmus, H. (2010) *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*, December, Technical report, President's Council of Advisors on Science and Technology.

Jones, S. (2013) *Cooperative Adaptive Cruise Control: Human Factors Analysis*, pp.FHWA-HRT-13-045 [online] https://trid.trb.org/view.aspx?id=1266088.

Klinge, S. and Middleton, R.H. (2009) 'String stability analysis of homogeneous linear unidirectionally connected systems with nonzero initial conditions' [online] http://digital-library.theiet.org/content/conferences/10.1049/cp.2009.1694.

Levine, W.S. and Athans, M. (1966) 'On the optimal error regulation of a string of moving vehicles', *IEEE Transactions on Automatic Control*, July, Vol. 11, No. 3, Ac-11, pp.355–356.

Li, S., Li, K., Rajamani, R. and Wang, J. (2011) 'Model predictive multi-objective vehicular adaptive cruise control', *IEEE Transactions on Control Systems Technology*, Vol. 19, No. 3, pp.556–566.

Lidström, K., Sjoberg, K., Holmberg, U., Andersson, J., Bergh, F., Bjade, M. and Mak, S. (2012) 'A modular CACC system integration and design', *IEEE Transactions on Intelligent Transportation Systems*, September, Vol. 13, No. 3, pp.1050–1061, IEEE.

Milanés, V., Shladover, S.E., Spring, J., Nowakowski, C., Kawazoe, H. and Nakamura, M. (2014) 'Cooperative adaptive cruise control in real traffic situations', *IEEE Transactions on Intelligent Transportation Systems*, February, Vol. 15, No. 1, pp.296–305, IEEE.

Montanaro, U., Tufo, M., Fiengo, G., di Bernardo, M. and Santini, S. (2014) 'On convergence and robustness of the extended cooperative cruise control', *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp.4083–4088, IEEE.

Naus, G.J., Vugts, R.P., Ploeg, J., Van de Molengraft, M. and Steinbuch, M. (2010) 'Stringstable CACC design and experimental validation: a frequency-domain approach', *IEEE Transactions on Vehicular Technology*, November, Vol. 59, No. 9, pp.4268–4279, IEEE.

Peppard, L. (1974) 'String stability of relative-motion PID vehicle control systems', *IEEE Transactions on Automatic Control*, October, Vol. 19, No. 5, pp.579–581, IEEE.

Ploeg, J., Scheepers, B.T., van Nunen, E., van de Wouw, N. and Nijmeijer, H. (2011) 'Design and experimental evaluation of cooperative adaptive cruise control', *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, pp.260–265.

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

Ploeg, J., Shukla, D.P., van de Wouw, N. and Nijmeijer, H. (2014a) 'Controller synthesis for string stability of vehicle platoons', *IEEE Transactions on Intelligent Transportation Systems*, April, Vol. 15, No. 2, pp.854–865, IEEE.

Ploeg, J., van de Wouw, N. and Nijmeijer, H. (2014b) 'Lp string stability of cascaded systems: application to vehicle platooning', *IEEE Transactions on Control Systems Technology*, March, Vol. 22, No. 2, pp.786–793.

Seiler, P. and Sengupta, R. (2005) 'An H_∞ approach to networked control', *IEEE Transactions on Automatic Control*, March, Vol. 50, No. 3, pp.356–364, IEEE.

Sheikholeslam, S. and Desoer, C.A. (1990) 'Longitudinal control of a platoon of vehicles', *American Control Conference*, pp.291–296, IEEE.

Shladover, S.E., Su, D. and Lu, X-Y. (2012) '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.

Skogestad, S. and Postlethwaite, I. (2007) *Multivariable Feedback Control: Analysis and Design*, Vol. 2, Wiley, New York.

Trudgen, M. and Mohammadpour, J. (2015) 'Robust cooperative adaptive cruise control design for connected vehicles', *ASME 2015 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, pp.V001T17A004–V001T17A004.

Turri, V., Besselink, B., Mårtensson, J. and Johansson, K.H. (2014) 'Fuel-efficient heavy-duty vehicle platooning by look-ahead control', *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp.654–660, IEEE.

van Willigen, W., Schut, M. and Kester, L. (2011) 'Evaluating adaptive cruise control strategies in worst-case scenarios', *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp.1910–1915, IEEE.

Vander Werf, J., Shladover, S.E., Miller, M.A. and Kourjanskaia, N. (2002) '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.

Wada, N. and Saeki, M. (1999) 'Design of a static anti-windup compensator which guarantees robust stability', *Transactions of the Institute of Systems, Control and Instrumentation Engineers*, Vol. 12, No. 11, pp.664–670.

Xiao, L. and Gao, F. (2010) 'A comprehensive review of the development of adaptive cruise control systems', *International Journal of Vehicle Mechanics and Mobility*, Vol. 48, No. 10, pp.1167–1192.