Department of Mechanical Engineering
Systems and Control
Dynamics & Control Group

CACC in vehicle platooning under absence of velocity and acceleration measurements

Master Thesis

Caner Yılmaz
ID: 1536117

Supervisor:
Dr. Ir. Erjen Lefeber

DC 2022.079

Eindhoven, September 14, 2022

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

14 September 2022 ………………………………………………………

Name

Caner YILMAZ ………………………………………………………

ID-number

1536117 ………………………………………………………

Signature

………………………………………………………

Submit the signed declaration to the student administration of your department.

[i] See: https://www.tue.nl/en/our-university/about-the-university/organization/integrity/scientific-integrity/
The Netherlands Code of Conduct for Scientific Integrity, endorsed by 6 umbrella organizations, including the VSNU, can be found here also. More information about scientific integrity is published on the websites of TU/e and VSNU

Abstract

Cooperative Adaptive Cruise Control (CACC) shows great promise in reducing human intervention in longitudinal driving, by incorporating both intervehicular communication and measurements obtained with on-board sensors into the control system. Despite its great potential, CACC is prone to communication impairment; and global measurements provided by the on-board dead reckoning systems (e.g. accelerometer, wheel encoder) are subject to cumulative errors and measurement noise, especially in strong weather conditions. In this thesis, an observer-based a-CACC framework is proposed to tackle the latter issue for a nonlinear vehicle model with the inclusion of counteracting forces such as friction and air drag. This is achieved by developing an observer for the global velocity and applied force measurements, which are then used to indirectly estimate global acceleration. Then, the conditions on whether the observer provides accurate estimation are assessed with a cascaded system approach. Finally, the state estimations are incorporated into CACC controller, which comprises of a PD action and a feed forward. The simulation results and mathematical work indicate that stable tracking dynamics can be achieved for a sufficiently large pool of initial estimations, provided that the vehicle is limited to either forward or backward motion. String stability is also achieved by appropriately increasing the desired time headway. Future work can be done on string stability, by automating the necessary time headway in response to initial intervehicular distance. The observability of the system during switching between "Drive" and "Reverse" gear of the vehicle is also open to further research.

Acknowledgements

First of all, I would like to express my gratitude to Dr. Ir. Erjen Lefeber for organizing this project within a short amount of time after I had completed my internship at TNO. His suggestions during our frequent meetings and his lecture notes on nonlinear control offered great guidance in successfully completing my thesis.

I also would like to thank my family for both financially and emotionally supporting me throughout my master study. Additionally, my father's teachings based on his PhD experience offered crucial insights on how to write quality academic reports. Especially with her support, my mom significantly reduced the time I had to allocate for housework, and focus on my thesis instead. In order to prevent burnout, my brother and my friends offered nice distractions from my study with online gaming and visiting fun places outside of my house and school.

Last but not least, I would like to thank you, as a reader, for taking interest in reading my thesis work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cooperative Adaptive Cruise Control (CACC)

History of vehicle control dates back to 1788, when James Watt and Matthew Boulton used a centrifugal governor in their steam engines to adjust the throttle depending on different loads received [1]. Similarly, in 1908, a governor was used in automobiles to maintain the speed of an engine [2]. In 1948, Ralph Teetor invented a feature ”Speedostat” that automatically regulated the speed of the vehicle; now labeled as Cruise Control (CC) in modern language [3].

Even though this invention has been important in making the first step into the domain of autonomous driving, CC was vulnerable to hazardous weather conditions [4]. Car accidents due to reckless human driving was (and still is) an ongoing issue. Hence, Adaptive Cruise Control (ACC) was proposed in which the vehicle control is based on a intervehicular spacing policy using relative position and velocity information obtained by on-board sensors such as radar or LiDAR [5].

ACC has later been extended into Cooperative Adaptive Cruise Control (CACC) [6]. In addition to on-board sensor data, information on the preceding vehicle's acceleration and certain mechanical properties is also communicated. The communication of acceleration information allows for the controller to achieve string stability; that is, the disturbances in traffic flow of a vehicle platoon does not amplify downstream. As string stability is quantified by acceleration information (which is not communicated in regular ACC), the intervehicular distance needed to be kept relatively large in order to account for sudden brakes of the predecessor vehicle.

1.2 Previous work on CACC

Most commonly in literature, CACC is studied with the assumption that the *desired acceleration* is communicated between the vehicles, which is labeled by u-CACC [6]. In u-CACC, variables depending on dynamical behavior of the predecessor vehicle also needs to be known. These variables may depend on manufacturing information, which may be classified; or it may not be in a universal format for other vehicles to process in their control algorithms.

As a solution to the drawbacks of u-CACC, a-CACC is proposed, in which *measured acceleration* is shared with the follower vehicle, instead of the desired acceleration [7]. Hence, the control algorithm can be applied without requiring knowledge on the dynamical behavior of the preceding vehicle.

Both u-CACC and a-CACC are prone to communication impairments and measurement noise. As such, degraded CACC is proposed for u-CACC (combined into name u-dCACC) [8], in which predecessor acceleration is estimated by a Singer model; a probability-based linear acceleration model used for tracking targets. Similarly, a Singer model is used in degraded a-CACC (a-dCACC) to estimate predecessor's acceleration measurement [9]. In addition, a-dCACC has a linear observer for unknown global acceleration. Several other solutions are proposed to combat temporary packet loss or disturbances [10].

String stability is usually defined with the ratio of $\mathcal{L}_2$ or $\mathcal{L}_\infty$ norms of consecutive vehicles' acceleration [11]. Although string stability is typically tested numerically, an analytical criterion to determine string stability, for a numerically evaluated minimum time headway is proposed in [9]; though, a linear vehicle model is employed. A controller achieving string stability is also designed for a nonlinear vehicle model in [12, 13], though it is assumed all measurements are available.

1.3 Problem formulation

After going through the previous work on tackling the challenges of CACC, we found that the permanent and simultaneous loss of both velocity and acceleration measurements was not studied, let alone for a nonlinear model with counteracting forces included. Hence, the research objective of this study is defined as

Design an observer-based CACC framework for a vehicle in 1-D platooning, whose velocity and acceleration measurements are unavailable, while also including the "nonlinear" effects caused by rolling resistance, damping and drag forces.

The research objective has been achieved through the following tasks:

- Design a preliminary global observer for the velocity without considering driveline dynamics, by treating the applied force as an input, instead of an unknown variable;

- Extend the preliminary observer into full-state by including the driveline dynamics and check for conditions on the observability of the full-state; and
- Create an observer-based CACC framework and test input-to-state stability (ISS) and string stability of the system.

One of the major difficulties of the research objective is the fact that feedback linearization cannot be applied to the motion model, because of the fact that global velocity measurements are not available.

1.4 Outline

In Chapter 2, preliminary knowledge on control theory and observer design is presented. In Chapter 3, a global observer is designed in order to compensate for absence of velocity measurement, without including driveline dynamics. In Chapter 4, the observer is extended to full state and conditions on the observability are presented. In Chapter 5, an observer-based control framework is proposed; along with simulation-based testing of input-to-state stability (ISS) and string stability. Finally, conclusion and recommendations to improve the CACC framework are given in Chapter 6.

Chapter 2

Preliminaries

In this chapter, the motion model of the vehicle is introduced and background information relevant to the thesis is given based on the existing literature.

2.1 Mathematical Model

The vehicle motion model is derived from the force equilibrium

$$m\dot{v}_i = F_i - (c_0 + c_1 v_i + c_2 v_i^2), \tag{2.1}$$

where v_i and F_i are velocity and thrust force of the vehicle i, respectively. The counteracting forces are composed of rolling resistance, damping and air drag. Each term of the second degree polynomial $(c_0 + c_1 v_i + c_2 v_i^2)$ models these forces in respective order; where the known coefficients c_0, c_1, c_2 are constant. It also holds that c_0 and c_2 are nonnegative.

The thrust force is driven by the following linear model, which we call "driveline dynamics" [14]:

$$\dot{F}_i = -\frac{1}{\tau}F_i + \frac{1}{\tau}\bar{u}_i. \tag{2.2}$$

In addition, position q_i measured from the rear bumper of the vehicle can be expressed by

$$\dot{q}_i = v_i. \tag{2.3}$$

By combining (2.1), (2.2) and (2.3), the motion model for agent i in 1-D platoon can be formulated as

$$\begin{aligned} \dot{q}_i &= v_i \\ \dot{v}_i &= \frac{1}{m}[F_i - (c_0 + c_1 v_i + c_2 v_i^2)] \\ \dot{F}_i &= -\frac{1}{\tau}F_i + \frac{1}{\tau}\bar{u}_i. \end{aligned} \tag{2.4}$$

The mathematical model is formulated under the assumption that the vehicle platoon is 1-D; that is, curvature and slope of the road are not taken into account. It is also assumed that the position measurement can be obtained with radar or GNSS. Otherwise, the system becomes unobservable. Therefore, the output of the system is taken as

$$y_i = q_i. \tag{2.5}$$

In some other literature, the vehicle model is transformed by a change of coordinates such that the acceleration a_i is part of the state vector $x = [q_i \ v_i \ a_i]$. However, designing an observer using the structure (2.4) is easier in the way that after estimating v_i and F_i, the acceleration estimate can be derived.

2.2 Observer design

2.2.1 Local observability

Consider the following system [15]:

$$\begin{aligned} \dot{x} &= f(x) + \Sigma_{i=1}^{m} g_i(x) u_i \\ y &= h(x) \end{aligned} \tag{2.6}$$

where x, y, u are the state, output and input of the system, respectively. Also, let W be the observability matrix defined as

$$W = \frac{\partial}{\partial x} \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ \frac{d^k y}{dt^k} \end{bmatrix} \tag{2.7}$$

where $k \geq n - 1$, with n denoting the relative degree of the system, which is the order of time derivative of output y when the input first explicitly appears. Then, the system is locally observable if $rank(W) = n$ and all inputs are equal to zero. If W is full rank and $k = n - 1$, then local observability holds for any input u, i.e., without assuming zero input.

2.2.2 Global observability with linearized error dynamics

In order to design a global observer for nonlinear systems, several strategies are proposed in literature. The most common method employed is to find a suitable change of coordinates for input and state such that we obtain a canonical form:

$$\begin{aligned} \dot{z} &= A_z z + \alpha(u_z, \eta) \\ \eta &= C_z z \end{aligned} \tag{2.8}$$

where z, u_z, η are the new coordinates for state, input and output, respectively. The input u_z is a function of available measurements and state estimate $\hat{z}$ only. The term

$\alpha(.)$ can be any function dependent on input and output only. Meanwhile, A_z and C_z are constant state and measurement matrices, respectively. Hence, a global observer can be designed such that the error dynamics is linear in z-coordinate errors.

In [16], a set of conditions is proposed to check whether global observer with linearized error dynamics exists for a class of nonlinear systems. First, the dynamical system is transformed into the form:

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 \\
\dot{\xi}_2 &= \xi_3 \\
&\vdots \\
\dot{\xi}_n &= \bar{f}(\xi_1, ..., \xi_{n-1}, u_\xi) \\
\eta &= \xi_1
\end{aligned} \tag{2.9}$$

In order to design a global observer for a second-order system (i.e. $n = 2$), it is sufficient that $\bar{f} = \dot{\xi}_2$ can be expressed in the form:

$$\dot{\xi}_2 = a(\xi_1, u_\xi) + b(\xi_1)\xi_2 + c(\xi_1)\frac{\xi_2^2}{2}. \tag{2.10}$$

For third order system ($n = 3$), there are two conditions that must be satisfied. The first one is that $\bar{f} = \dot{\xi}_3$ should be expressed in the form:

$$\dot{\xi}_3 = a(\xi_1, u_\xi) + b(\xi_1)\xi_2 + c(\xi_1)\frac{\xi_2^2}{2} + d(\xi_1)\frac{\xi_2^3}{3} + [\rho(\xi_1) + \sigma(\xi_1)\xi_2]\xi_3. \tag{2.11}$$

Given the form (2.11), the following partial differential equations (PDE) must be satisfied:

$$\begin{aligned}
\frac{d\sigma}{d\xi_1} &= \frac{3}{2}d + \frac{2}{3}\sigma^2, \\
\frac{d\rho}{d\xi_1} &= c + \rho\sigma.
\end{aligned} \tag{2.12}$$

If both conditions hold, then there exists a coordinate transformation for the third-order system, which leads to linearized error dynamics.

2.3 Cascaded system

Definition 2. A continuous function $f(x) : [0, \infty) \to [0, \infty)$ is a class $\mathcal{K}$ function if it is strictly increasing and $f(0) = 0$.

Definition 3. A function $f(x)$ is radially unbounded if $|x| \to \infty$ implies $f(x) \to \infty$.

Definition 4. A function $f(x)$ is Lipschitz continuous if $f(x)$ is continuous and for any x_1, x_2, it holds that $|f(x_1) - f(x_2)| \leq M|x_1 - x_2|$ for some constant $M \in [0, \infty)$.

Consider a cascaded system in the form [17, 18]:

$$\begin{aligned} \dot{x} &= f_1(x,t) + g(x,y,t)y \\ \dot{y} &= f_2(y,t) \end{aligned} \tag{2.13}$$

where $f_1(x,t)$ is continuously differentiable and $f_2(y,t), g(x,y,t)$ are Lipschitz continuous. This system could be treated as the state

$$\Sigma_1 : \dot{x} = f_1(x,t),$$

perturbed by the output

$$\Sigma_2 : \dot{y} = f_2(y,t).$$

The cascaded system is GUAS if the following assumptions are satisfied:

Assumption on Σ_1: The system Σ_1 is GUAS and there exists a Lyapunov function $V(x,t)$ such that the following inequalities are satisfied:

$$W(x) \leq V(x,t) \tag{2.14}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f_1(x,t) \leq 0 \tag{2.15}$$

$$\left|\frac{\partial V}{\partial x}\right| |x| \leq cV(x,t) \tag{2.16}$$

with $W(x)$ some positive-definite and radially unbounded function and c a positive constant.

Assumption on Σ_2: The system Σ_2 is GUAS and the integral inequality below holds:

$$\int_{t_0}^{\infty} |y(t, t_0, y(t_0)|dt \leq \kappa(|y(t_0)|) \tag{2.17}$$

$\forall t_0 \leq 0$ and for some class $\mathcal{K}$ function $\kappa(.)$.

Assumption on interconnection: The affine term $|g(x,t,u)|$ is upper-bounded by a continuous function linear in $|x|$:

$$|g(x,t,u)| \leq \theta_1(|y|) + \theta_2(|y|)|x|. \tag{2.18}$$

Chapter 3

Observer design without driveline dynamics

In this chapter, a preliminary observer design will be made for the following subsystem:

$$\begin{aligned} \dot{q}_i &= v_i \\ \dot{v}_i &= \frac{1}{m}[F_i - (c_0 - c_1 v_i - c_2 v_i^2)], \end{aligned} \tag{3.1}$$

where the applied force F_i is treated as an input to the subsystem. The objective is to design a global observer with linearized error dynamics. This is done by an appropriate coordinate transformation such that the subsystem is linear in new state and input coordinates. For visual simplicity, the subscript i is dropped in this chapter.

3.1 Observability of the system

In order to design a global observer, the system must be locally observable first for all values of $x = [q\ v\ F]^T$. The observability matrix W is computed as

$$W = \frac{\partial}{\partial x}\begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \frac{\partial}{\partial x}\begin{bmatrix} q \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.2}$$

Since W is full rank, the subsystem (3.1) is locally observable everywhere.

Now, we define $\xi_1 = q$ and $\xi_2 = v$. Then, the subsystem takes the form of (2.10):

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \frac{1}{m}[F - (c_0 + c_1 \xi_2 + c_2 \xi_2^2)] \\ &= \underbrace{\frac{F - c_0}{m}}_{a(\xi_1),F} + \underbrace{\frac{-c_1}{m}}_{b(\xi_1)} \xi_2 + \underbrace{\frac{-2c_2}{m}}_{c(\xi_1)} \frac{\xi_2^2}{2} \end{aligned} \tag{3.3}$$

Hence, by [16], a suitable change of coordinates can be found such that a global observer can be designed with linearized error dynamics.

3.2 Coordinate transformation

The system (3.4) can be rewritten as

$$\begin{aligned} \dot{q} &= v \\ \dot{v} &= \frac{1}{m}\left[F - c_0 + \frac{c_1^2}{4c_2} - \left(\sqrt{c_2}v + \frac{c_1}{2\sqrt{c_2}}\right)^2\right]. \end{aligned} \tag{3.4}$$

Then, we apply the following transformation:

$$\begin{aligned} u &= F - c_0 + \frac{c_1^2}{4c_2} \\ \bar{v} &= \sqrt{c_2}v + \frac{c_1}{2\sqrt{c_2}}. \end{aligned} \tag{3.5}$$

The transformation in (3.5) leads to the following auxiliary system:

$$\begin{aligned} \dot{q} &= \frac{1}{\sqrt{c_2}}\bar{v} - \frac{c_1}{2c_2} \\ \dot{\bar{v}} &= \frac{\sqrt{c_2}}{m}\left[u - \bar{v}^2\right]. \end{aligned} \tag{3.6}$$

We apply a second transformation; this time, on state and output as follows:

$$\begin{aligned} z_1 &= \exp\left(\frac{c_2}{m}q\right) \\ z_2 &= \bar{v}\exp\left(\frac{c_2}{m}q\right) \\ \eta &= \exp\left(\frac{c_2}{m}q\right) = z_1. \end{aligned} \tag{3.7}$$

The time derivatives of new state coordinates are computed as

$$
\begin{aligned}
\dot{z}_1 &= \left(\frac{\sqrt{c_2}}{m}\bar{v} - \frac{c_1}{2m}\right)\exp\left(\frac{c_2}{m}q\right) \\
&= -\frac{c_1}{2m}z_1 + \frac{\sqrt{c_2}}{m}z_2 \\
\\
\dot{z}_2 &= \left(\dot{\bar{v}} + \frac{\sqrt{c_2}}{m}\bar{v}^2 - \frac{c_1}{2m}\bar{v}\right)\exp\left(\frac{c_2}{m}q\right) \\
&= \left(\frac{\sqrt{c_2}}{m}u - \frac{\sqrt{c_2}}{m}\bar{v}^2 + \frac{\sqrt{c_2}}{m}\bar{v}^2 - \frac{c_1}{2m}\bar{v}\right)\exp\left(\frac{c_2}{m}q\right) \\
&= \left(\frac{\sqrt{c_2}}{m}u - \frac{c_1}{2m}\bar{v}\right)\exp\left(\frac{c_2}{m}q\right) \\
&= -\frac{c_1}{2m}z_2 + \frac{\sqrt{c_2}}{m}u\eta.
\end{aligned}
\tag{3.8}
$$

This results in the following transformed dynamics ($z = [z_1 \quad z_2]^T$):

$$
\begin{aligned}
\dot{z} &= \underbrace{\begin{bmatrix} -\frac{c_1}{2m} & \frac{\sqrt{c_2}}{m} \\ 0 & -\frac{c_1}{2m} \end{bmatrix}}_{A} z + \underbrace{\begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m}\eta \end{bmatrix}}_{\alpha(\eta,u)} u \\
\eta &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} z
\end{aligned}
\tag{3.9}
$$

which is in the canonical form (2.8).

3.3 Observer design

An observer for the system in (3.9) can be formulated as follows (with negative feedback):

$$
\begin{aligned}
\dot{\hat{z}} &= \underbrace{\begin{bmatrix} -\frac{c_1}{2m} & \frac{\sqrt{c_2}}{m} \\ 0 & -\frac{c_1}{2m} \end{bmatrix}}_{A} \hat{z} + \underbrace{\begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m}\eta \end{bmatrix}}_{\alpha(\eta,u)} u + \underbrace{\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}}_{L} (\eta - \hat{\eta}) \\
\hat{\eta} &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \hat{z}
\end{aligned}
\tag{3.10}
$$

where the observer estimation is denoted by superscript $\hat{}$. Defining the observer error as $\tilde{z} = z - \hat{z}$, linearized error dynamics is obtained as

$$
\dot{\tilde{z}} = (A - LC)\tilde{z}. \tag{3.11}
$$

If all eigenvalues of $A - LC$ are in left-half plane (LHP), the error is guaranteed to converge to zero as time goes to infinity. Hence, the estimated state should converge to the true value as well. The characteristic polynomial of $A - LC$ is calculated as

$$
\begin{aligned}
(\lambda + \frac{c_1}{2m} + l_1)(\lambda + \frac{c_1}{2m}) + \frac{\sqrt{c_2}}{m} l_2 = 0 \\
\underbrace{1}_{a} \lambda^2 + \underbrace{\left(\frac{c_1}{m} + l_1\right)}_{b} \lambda + \underbrace{\left(\frac{c_1}{2m} l_1 + \frac{\sqrt{c_2}}{m} l_2 + \frac{c_1^2}{4m^2}\right)}_{c} = 0
\end{aligned}
\tag{3.12}
$$

According to Routh-Hurwitz stability criterion [15], the eigenvalues are strictly in LHP if and only if

$$
\begin{aligned}
\frac{b}{a} > 0 \\
\frac{c}{a} > 0.
\end{aligned}
\tag{3.13}
$$

This implies the error dynamics is asymptotically stable if and only if

$$
\begin{aligned}
l_1 &> -\frac{c_1}{m} \\
l_2 &> \frac{-c_1^2 - 2c_1 m l_1}{4\sqrt{c_2} m}.
\end{aligned}
\tag{3.14}
$$

It can be realized from (3.14) that a sufficient lower bound for l_2 would be

$$
\frac{-c_1^2 - 2c_1 m \inf(l_1)}{4\sqrt{c_2} m} = \frac{c_1^2}{4\sqrt{c_2} m}. \tag{3.15}
$$

After obtaining the state estimation, an inverse transformation can be made to the original coordinates as follows:

$$
\begin{aligned}
\hat{q} &= \frac{m}{c_2} \log(\hat{z}_1) \\
\hat{v} &= \frac{1}{\sqrt{c_2}} \frac{\hat{z}_2}{\hat{z}_1} - \frac{c_1}{2c_2} \\
\hat{y} &= \hat{q}.
\end{aligned}
\tag{3.16}
$$

By taking the time derivative of the inverse transformation in (3.16), the observer in original coordinates can be derived. For position,

$$
\begin{aligned}
\dot{\hat{q}} = \frac{m}{c_2} \frac{\dot{\hat{z}}_1}{\hat{z}_1} &= \frac{m}{c_2} \frac{\frac{c_2}{m} \hat{v} e^{\frac{c_2}{m}\hat{q}} + l_1 \left(e^{\frac{c_2}{m} q} - e^{\frac{c_2}{m}\hat{q}}\right)}{e^{\frac{c_2}{m}\hat{q}}} \\
&= \hat{v} + \frac{m}{c_2} l_1 \left(e^{\frac{c_2}{m}(q - \hat{q})} - 1\right).
\end{aligned}
\tag{3.17}
$$

For velocity,

$$
\begin{aligned}
\dot{\hat{v}} &= \frac{1}{\sqrt{c_2}} \frac{\dot{\hat{z}}_2 \hat{z}_1 - \dot{\hat{z}}_1 \hat{z}_2}{\hat{z}_1{}^2} = \frac{1}{\sqrt{c_2}} \frac{\dot{\hat{z}}_2 - \dot{\hat{z}}_1 \hat{\hat{v}}}{\hat{z}_1} \\
&= \frac{1}{\sqrt{c_2}} l_2 \left(e^{\frac{c_2}{m}(q-\hat{q})} - 1 \right) - \frac{\hat{\hat{v}}}{\sqrt{c_2}} l_1 \left(e^{\frac{c_2}{m}(q-\hat{q})} - 1 \right) - \frac{\hat{\hat{v}}^2}{m} + \frac{u}{m} e^{\frac{c_2}{m}(q-\hat{q})} \\
&= \frac{1}{\sqrt{c_2}} (l_2 - \hat{\hat{v}} l_1) \left(e^{\frac{c_2}{m}(q-\hat{q})} - 1 \right) + \frac{1}{m} \left(u e^{\frac{c_2}{m}(q-\hat{q})} - \hat{\hat{v}}^2 \right),
\end{aligned}
\tag{3.18}
$$

with

$$
\begin{aligned}
\hat{\hat{v}} &= \sqrt{c_2} \hat{v} + \frac{c_1}{2\sqrt{c_2}}, \\
u &= F - c_0 + \frac{c_1^2}{4c_2}.
\end{aligned}
$$

3.4 Verification with Lyapunov proof

In this section, results obtained on the stability of observer error dynamics in previous section are verified by means of Lyapunov methods. This verification serves as a basis for Chapter 4, where the observer is extended into full-state.

In order to verify whether the observer in Section 3.3 works fine without a control input ($u = 0$), let's take the following Lyapunov function candidate:

$$V = \tilde{z}^T P \tilde{z}, \tag{3.19}$$

where P is a square matrix to be determined. Taking the time derivative of V along solutions of (3.11) yields the following:

$$\dot{V} = \tilde{z}^T [(A - LC)^T P + P(A - LC)] \tilde{z}. \tag{3.20}$$

We declare an arbitrary positive definite matrix Q such that

$$\dot{V} = -\tilde{z}^T Q \tilde{z}. \tag{3.21}$$

This implies that

$$(A - LC)^T P + P(A - LC) = -Q. \tag{3.22}$$

Given a choice of Q, if $(A - LC)$ is Hurwitz, then P is a unique and positive definite matrix satisfying (3.22) [15].

We choose an arbitrary positive definite matrix Q first, say $Q = I$. This would automatically satisfy the conditions $\dot{V} \leq 0\ \forall \tilde{z}$; and $\dot{V} = 0 \iff \tilde{z} = 0$. Since $A - LC$ is also known in terms of l_1 and l_2, the goal is to compute P and find the range of values for l_1 and l_2 such that P remains positive definite. The P matrix can be symbolically represented by

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \tag{3.23}$$

with unknown scalars p_{11}, p_{12}, p_{21} and p_{22}. We also have

$$A - LC = \begin{bmatrix} -\frac{c_1}{2m} - l_1 & \frac{\sqrt{c_2}}{m} \\ -l_2 & -\frac{c_1}{2m} \end{bmatrix}. \tag{3.24}$$

Substituting (3.23) and (3.24) into (3.22) yields:

$$\begin{bmatrix} -\frac{c_1}{2m} - l_1 & -l_2 \\ \frac{\sqrt{c_2}}{m} & -\frac{c_1}{2m} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} -\frac{c_1}{2m} - l_1 & \frac{\sqrt{c_2}}{m} \\ -l_2 & -\frac{c_1}{2m} \end{bmatrix} = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.25}$$

The P matrix is computed by

$$P = P_0 \begin{bmatrix} c_1^2 + 2l_2^2m^2 + l_1c_1m + 2l_2\sqrt{c_2}m & c_1\sqrt{c_2} - l_2c_1m - 2l_1l_2m^2 \\ c_1\sqrt{c_2} - l_2c_1m - 2l_1l_2m^2 & 2c_2 + c_1^2 + 2c_1^2m^2 + 3l_1c_1m + 2l_2\sqrt{c_2}m \end{bmatrix} \tag{3.26}$$

with P_0 equal to

$$P_0 = \frac{m}{(c_1 + l_1m)(c_1^2 + 2l_1c_1m + 4l_2\sqrt{c_2}m)}. \tag{3.27}$$

Substituting the result into (3.19) yields,

$$\begin{aligned} V = P_0[&\tilde{z}_1^2(c_1^2 + 2l_2^2m^2 + l_1c_1m + 2l_2\sqrt{c_2}m) \\ &+ \tilde{z}_2^2(2c_2 + c_1^2 + 2l_1^2m^2 + 3l_1c_1m + 2l_2\sqrt{c_2}m) \\ &+ 2\tilde{z}_1\tilde{z}_2(c_1\sqrt{c_2} - l_2c_1m - 2l_1l_2m^2)] \end{aligned} \tag{3.28}$$

It can be seen that having $\tilde{z}_1 = \tilde{z}_2 = 0$ leads to $V = 0$. Now, assuming this does not hold, we find the range of observer gain values such that $V > 0$; and see if (3.14) still holds.

In order to do that, we will check the positive-definiteness of P. In order for P to be positive-definite, it should be symmetric and the following should be satisfied:

$$\begin{aligned} p_{11} &> 0 \\ det(P) = p_{11}p_{22} - p_{12}^2 &> 0 \end{aligned} \tag{3.29}$$

Since P is symmetric, it suffices to find the range of values of l_1 and l_2 such that,

$$\begin{aligned} m\frac{c_1(c_1 + l_1m) + 2ml_2(ml_2 + \sqrt{c_2})}{(c_1 + l_1m)(c_1^2 + 2l_1c_1m + 4l_2\sqrt{c_2}m)} &> 0 \\ m^2\frac{(c_1 + l_1m)^2 + (\sqrt{c_2} + l_2m)^2}{(c_1 + l_1m)^2(c_1^2 + 2l_1c_1m + 4l_2\sqrt{c_2}m)} &> 0 \end{aligned} \tag{3.30}$$

From the second inequality, we need to have $(c_1^2 + 2l_1c_1m + 4l_2\sqrt{c_2}m) > 0$. In other words,

$$l_2 > \frac{-c_1^2 - 2c_1 m l_1}{4\sqrt{c_2} m}. \tag{3.31}$$

After applying this necessary condition into the first inequality of (3.30) (and rearranging the terms), we obtain

$$m \frac{\frac{c_1^2}{2} + \frac{c_1^2}{8c_2}(c_1 + 2ml_1)^2}{(c_1 + l_1 m)(c_1^2 + 2l_1 c_1 m + 4l_2\sqrt{c_2} m)} > 0 \tag{3.32}$$

We had already established that $(c_1^2 + 2l_1 c_1 m + 4l_2\sqrt{c_2} m) > 0$ must hold. Since the numerator is positive, it is necessary to have $(c_1 + l_1 m) > 0$. That is,

$$l_1 > -\frac{c_1}{m}. \tag{3.33}$$

The inequalities (3.31) and (3.33) validate what we had found in (3.14).

Remark: The range of l_2 is $(-\infty,\ +\infty)$. However, only values in the subset $(\frac{c_1^2}{4\sqrt{c_2} m},\ +\infty)$ lead to stability in error dynamics for all values of $l_1 > -\frac{c_1}{m}$.

3.5 Summary

In this chapter, a global observer for the subsystem without the driveline dynamics is proposed. This is done by an appropriate state and input transformation, which results in the observer error dynamics being linear in transformed coordinates. Then, the stability of the error dynamics is verified by means of Lyapunov. This work forms the basis in extending the observer to full-state in the next chapter.

Chapter 4

Full-state observer design

Recall the nonlinear dynamics below:

$$\begin{aligned}
\dot{q}_i &= v_i \\
\dot{v}_i &= \frac{1}{m_i}[F_i - (c_0 + c_1 v_i + c_2 v_i^2)] \\
\dot{F}_i &= -\frac{1}{\tau_i}F_i + \frac{1}{\tau_i}\bar{u}_i,
\end{aligned} \tag{4.1}$$

where $\tau_i > 0$ is a positive time constant representing driveline dynamics of the agent; and $\bar{u}_i$, the control input yet to be determined. The constants c_0, c_1, c_2 are assumed to be known; and the velocity $v_i \geq 0$ in Drive gear. The goal is to design a global observer for velocity v_i and thrust force F_i. Once again, we assume that the position can be measured:

$$y_i = q_i. \tag{4.2}$$

For simplicity, the subscript i is dropped in this chapter, also.

4.1 Observability of the system

In order to check whether a global observer for a system can be designed, the system must be locally observable for all values of $x = [q\ v\ F]^T$. Computing the matrix (2.7) yields

$$\begin{aligned}
W &= \frac{\partial}{\partial x}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} \\
&= \frac{\partial}{\partial x_i}\begin{bmatrix} q \\ v \\ \frac{1}{m}[F - (c_0 + c_1 v + c_2 v^2)] \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{m}(c_1 + 2c_2 v) & \frac{1}{m} \end{bmatrix}.
\end{aligned} \tag{4.3}$$

Indeed, the observability matrix W is full rank. Hence, the system is **locally observable** everywhere. Now, we would like to find a change of state (z), input (u) and output (y) such that we obtain linearized dynamics:

$$\begin{aligned} \dot{z} &= f(z) + g(z)u \\ h &= h(z) \end{aligned} \tag{4.4}$$

where $f(z)$ is linear. In order to check whether such a form exists, we shall first transform the model (4.1) by $\xi_1 = q$, $\xi_2 = v$, $\xi_3 = \dot{v} = \frac{1}{m}[F - (c_0 + c_1 v + c_2 v^2)]$. Then,

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_3 \\ \dot{\xi}_3 &= \frac{\dot{F}}{m} - \frac{c_1}{m}\xi_3 - \frac{2c_2}{m}\xi_2\xi_3. \end{aligned} \tag{4.5}$$

Combining the fact that $F = \xi_3 m + c_0 + c_1\xi_2 + c_2\xi_2^2$; with (4.1) and (4.5), we obtain

$$\dot{\xi}_3 = \underbrace{\frac{\bar{u} - c_0}{m\tau}}_{a(\xi_1)} + \underbrace{\left(-\frac{c_1}{m\tau}\right)}_{b(\xi_1)}\xi_2 + \underbrace{\left(-\frac{2c_2}{m\tau}\right)}_{c(\xi_1)}\frac{\xi_2^2}{2} + \underbrace{0}_{d(\xi_1)}\frac{\xi_2^3}{3} + \left(\underbrace{\left(-\frac{c_1}{m} - \frac{1}{\tau}\right)}_{\rho(\xi_1)} + \underbrace{\left(-\frac{2c_2}{m}\right)}_{\sigma(\xi_1)}\xi_2\right)\xi_3 \tag{4.6}$$

which is of the form (2.11). The following PDEs must also be satisfied:

$$\begin{aligned} \frac{d\sigma}{d\xi_1} &= \frac{3}{2}d + \frac{2}{3}\sigma^2 = \frac{8c_2^2}{3m^2}, \\ \frac{d\rho}{d\xi_1} &= c + \rho\sigma = \frac{2c_1c_2}{m^2}. \end{aligned} \tag{4.7}$$

However, $\frac{d\sigma}{d\xi_1} = \frac{d\rho}{d\xi_1} = 0$. Hence, the PDEs are only satisfied for $c_2 = 0$, which implies a linear model by default anyway. Hence, an observer with linearized dynamics does not exist. However, we can still design a nonlinear observer directly.

4.2 Full-state observer design

When the driveline dynamics was not included (i.e., the force F was treated as an input), the expressions for position and velocity estimates were derived as follows:

$$\dot{\hat{q}}_{no,DL} = \hat{v} + \frac{m}{c_2}l_1\left(e^{\frac{c_2}{m}(q-\hat{q})} - 1\right) \tag{4.8a}$$

$$\dot{\hat{v}}_{no,DL} = \frac{1}{\sqrt{c_2}}(l_2 - \hat{\bar{v}}l_1)\left(e^{\frac{c_2}{m}(q-\hat{q})} - 1\right) + \frac{1}{m}\left(ue^{\frac{c_2}{m}(q-\hat{q})} - \hat{\bar{v}}^2\right) \tag{4.8b}$$

with

$$
\begin{aligned}
u &= F - c_0 + \frac{c_1^2}{4c_2} \\
\bar{v} &= \sqrt{c_2}v + \frac{c_1}{2\sqrt{c_2}} \\
l_1 &> -\frac{c_1}{m} \\
l_2 &> \frac{c_1}{4\sqrt{c_2}m}.
\end{aligned}
$$

Since the force is now also an unknown output, the term u needs to be replaced by $\hat{u}$. Hence, the full-state observer shall be formulated as follows:

$$
\begin{aligned}
\dot{\hat{q}} &= \hat{v} + \frac{m}{c_2}l_1\left(e^{\frac{c_2}{m}(q-\hat{q})} - 1\right) = \dot{\hat{q}}_{noDL} \\
\dot{\hat{\bar{v}}} &= \frac{1}{\sqrt{c_2}}(l_2 - \hat{\bar{v}}l_1)\left(e^{\frac{c_2}{m}(q-\hat{q})} - 1\right) + \frac{1}{m}\left(\hat{u}e^{\frac{c_2}{m}(q-\hat{q})} - \hat{\bar{v}}^2\right) \\
&= \dot{\hat{\bar{v}}}_{noDL} + \frac{1}{m}(u-\hat{u})e^{\frac{c_2}{m}(q-\hat{q})} \\
\dot{\hat{F}} &= f_3(q, \hat{q}, \hat{v}, \hat{F}, \bar{u}).
\end{aligned}
\tag{4.9}
$$

It is desired to keep the structure of position and velocity observers the same, while designing the observer function $f_3(q, \hat{q}, \hat{v}, \hat{F}, \bar{u})$. One could design an observer for F, by simply replacing the true values of F with its estimate $\hat{F}$:

$$
\dot{\hat{F}} = -\frac{1}{\tau}(\hat{F} - \bar{u}). \tag{4.10}
$$

Then, by defining the observer error for force as $\tilde{u} = u - \hat{u} = F - \hat{F}$, one could realize that we have the following stable error dynamics:

$$
\dot{\tilde{u}} = -\frac{1}{\tau}\tilde{u}, \tag{4.11}
$$

with analytical solution

$$
\tilde{u}(t) = \tilde{u}(0)e^{-\frac{t}{\tau}}. \tag{4.12}
$$

Hence, regardless of the controller we use, it is guaranteed that the force error is bounded and converges to zero. Since (4.12) is continuously differentiable everywhere, the force error dynamics is globally (uniformly) asymptotically stable (U-GAS). What is left to check is whether the error in velocity still converges to zero; and if not, find the conditions where the error dynamics remain uniformly asymptotically stable (UAS).

4.3 Stability of observer error dynamics

The full-state observer leads to the following error dynamics in matrix form as follows:

$$\begin{aligned}\dot{\tilde{z}} &= (A - LC)\tilde{z} + \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} e^{\frac{c_2}{m}q} \end{bmatrix} \tilde{u} \\ \dot{\tilde{u}} &= -\frac{1}{\tau}\tilde{u}.\end{aligned} \tag{4.13}$$

It should be noted that the affine term $\frac{\sqrt{c_2}}{m} e^{\frac{c_2}{m}q}$ is unbounded if $\dot{q} > 0$. Hence, the stability of the error dynamics shall be analysed under three cases depending on the time derivative of $q(t)$.

Case (i): $\dot{q}(t) \geq 0$

We retry putting the error dynamics into cascaded form by using the following error state instead:

$$\tilde{w} = e^{-\frac{c_2}{m}q}\tilde{z}. \tag{4.14}$$

Taking the time derivative of new error coordinate transformation, we obtain the following result:

$$\begin{aligned}\dot{\tilde{w}} &= e^{-\frac{c_2}{m}q}\dot{\tilde{z}} - \frac{c_2}{m}\dot{q}e^{-\frac{c_2}{m}q}\tilde{z} \\ &= (A - LC - \frac{c_2}{m}\dot{q})e^{-\frac{c_2}{m}q}\tilde{z} + \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} \end{bmatrix} \tilde{u} \\ &= (A - LC - \frac{c_2}{m}\dot{q})\tilde{w} + \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} \end{bmatrix} \tilde{u}.\end{aligned} \tag{4.15}$$

The dynamics for force error is kept the same. Hence, we obtain the following cascaded system:

$$\begin{aligned}\dot{\tilde{w}} &= \underbrace{\left(A - LC - \frac{c_2}{m}\dot{q}(t)\right)\tilde{w}}_{f_1(\tilde{w},t)} + \underbrace{\begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} \end{bmatrix}}_{g(\tilde{w},t)} \tilde{u} \\ \dot{\tilde{u}} &= -\frac{1}{\tau}\tilde{u} = f_2(\tilde{u}).\end{aligned} \tag{4.16}$$

Assuming first that $\Sigma_{1,w} : \dot{\tilde{w}} = f_1(\tilde{w}, t)$, let $V(t, \tilde{w}) = \tilde{w}^T P \tilde{w}$ be a Lyapunov candidate function with P a square scalar matrix to be determined. The time derivative along $\Sigma_{1,w}$ of the candidate function is computed as

$$\begin{aligned}
\dot{V}(t,\tilde{w}) &= \dot{\tilde{w}}^T P\tilde{w} + \tilde{w}^T P\dot{\tilde{w}} \\
&= \tilde{w}^T \left[\left(A - LC - \frac{c_2}{m}\dot{q}(t)\right) P + P\left(A - LC - \frac{c_2}{m}\dot{q}(t)\right)\right]\tilde{w} \\
&= \tilde{w}^T \left[(A - LC)^T P + P(A - LC) - \frac{2c_2}{m}\dot{q}(t)P\right]\tilde{w} \\
&= -\tilde{w}^T Q(t)\tilde{w}.
\end{aligned} \tag{4.17}$$

For $Q(t) = Q^T(t) > 0$, which guarantees $\dot{V}(t,\tilde{w})$ is negative semi-definite, the matrix P is a solution to the following Lyapunov equation derived from (4.17):

$$\left[(A - LC)^T P + P(A - LC) - \frac{2c_2}{m}\dot{q}(t)P\right] = -Q. \tag{4.18}$$

By choosing $Q(t) = \frac{2c_2}{m}\dot{q}(t)P + I$, (4.18) simplifies into

$$(A - LC)^T P + P(A - LC) = -I \tag{4.19}$$

which has a unique and positive-definite solution for P equal to the matrix obtained in (3.26-3.27), the case without driveline dynamics. This leads to the same Lyapunov function in (3.28), which is positive-definite. Since the Lyapunov function is also radially unbounded, the last step would be to check the condition that makes $Q(t)$ positive definite:

$$\frac{2c_2}{m}\dot{q}(t)P + I > 0 \Longrightarrow \dot{q}(t) > -\frac{m}{2c_2}P^{-1}. \tag{4.20}$$

Since P^{-1} is positive definite, $\dot{q}(t) \geq 0$ is a sufficient condition for $Q(t)$ to be positive definite. In addition, $f_1(\tilde{w},t)$ is linear in $\tilde{w}$; thus, the assumption on $\Sigma_{1,w}$ is satisfied.

The assumption on $\Sigma_2 : f_2(\tilde{u})$ is also satisfied, since the time integral (2.17) for any time $t_0 \geq 0$ is bounded by a function $\kappa(.)$, which is of class $\mathcal{K}$ w.r.t. $|\tilde{u}(t_0)|$:

$$\begin{aligned}
\int_{t_0}^{\infty} ||\tilde{u}(t)||dt &= \int_{t_0}^{\infty} |\tilde{u}(t_0)|e^{-\frac{t}{\tau}}dt \\
&= \frac{|\tilde{u}(t_0)|}{\tau}e^{-\frac{t_0}{\tau}} \\
&\leq \frac{|\tilde{u}(t_0)|}{\tau} = \kappa(|\tilde{u}(t_0)|).
\end{aligned} \tag{4.21}$$

Finally, the remaining step is to show that $g(\tilde{w},t)$ has an upper bound as follows:

$$\begin{aligned}
g(\tilde{w},t) &= \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} \end{bmatrix} \\
&\leq \underbrace{\left|\begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} \end{bmatrix}\right|}_{\theta_1} + \underbrace{\left|\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\right|}_{\theta_2} |\tilde{w}|.
\end{aligned} \tag{4.22}$$

Therefore, by [17], the cascaded system is UAS for all $\dot{q}(t) = v(t) \geq 0$.

From a practical perspective, this proof is sufficient, since a vehicle in a platoon cannot achieve negative velocity in Drive gear. However, for controlled system with controller composed of estimates from the observer, it is possible to achieve negative velocity (e.g. when the initial values of velocity and force are set to zero). Hence, the stability of the cascaded system is also tested for $\dot{q}(t) < 0$.

Case (ii): $\dot{q}(t) < 0$

Recall the pre-transformed cascaded system in (4.13). In Section 3.4, it is already shown that $\Sigma_1 : \dot{\tilde{z}} = f_1(\tilde{z}, t)$ is U-GAS. Since $f_1(\tilde{z}, t)$ is linear in $\tilde{z}$, the assumption on Σ_1 is satisfied. The assumption on Σ_2 also holds, as already proven in (4.21). The remaining step is to show upper bound on $|g(\tilde{w}, t)|$, when $\dot{q}(t) < 0$. If q is monotonically decreasing, the upper bound can be trivially found as

$$\begin{aligned} g(\tilde{w}, t) &= \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} e^{\frac{c_2}{m} q} \end{bmatrix} \\ &\leq \underbrace{\left| \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} e^{\frac{c_2}{m} q_0} \end{bmatrix} \right|}_{\theta_1} + \underbrace{\left| \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \right|}_{\theta_2} |\tilde{w}| \end{aligned} \tag{4.23}$$

with finite $q_0 = q(0)$.

Case (iii): Mix of $\dot{q}(t) < 0$ **and** $\dot{q}(t) \geq 0$

Now, suppose that decrease or increase in q is not monotonic, i.e. $\exists t_1 < t$ such that $\dot{q}(t_1) = 0$ and $sign(\dot{q}(t_1^-)) = -sign(\dot{q}(t_1^+)) \neq 0$. Since q is continuously differentiable, a local extremum is created at time t_1, which is bounded. If there are similar instances at time $T = t_1, t_2, ..., t_n \in [0, t)$, multiple local extrema are created accordingly. If T is a finite set and $\dot{q}(t) < 0$ for $t > t_n$, the upper bound of $|g(\tilde{w}, t)|$ can be expressed as

$$\begin{aligned} g(\tilde{w}, t) &= \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} e^{\frac{c_2}{m} q} \end{bmatrix} \\ &\leq \underbrace{\left| \begin{bmatrix} 0 \\ \frac{\sqrt{c_2}}{m} e^{\frac{c_2}{m} q_{max}} \end{bmatrix} \right|}_{\theta_1} + \underbrace{\left| \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \right|}_{\theta_2} |\tilde{w}| \end{aligned} \tag{4.24}$$

where q_{max} is a finite value expressed by

$$q_{max} = \max \left(q_0, \bigcup_{t_i \in T}^{n} q(t_i) \right). \tag{4.25}$$

However, if either T is not finite or $\dot{q}(t) > 0$ for all $t > t_n$, we do not have a mathematical proof on whether the observer error dynamics still tend to zero. The case of having $\dot{q}(t) = v(t)$ in both positive and negative regions is further explored through simulations in Sections 4.3 and 5.3.

4.4 Simulation test

The simulation test is conducted in MATLAB, with the set values given in Table 4.1.

Table 4.1: Initial state and state estimation values

m	True value	Estimate #1	Estimate #2	Estimate #3
q_0 [m]	0	0	0	0
v_0 [m/s]	20	10	-5	25
F_0 [N]	4000	2500	4500	6000

The vehicle parameters are computed/assigned mostly based on Toyota Prius [19]; and the motivation behind the choice of the numerical values of these parameters can be found in Appendix A. Along with observer gains, the constant parameters are given in Table 4.2.

Table 4.2: Vehicle parameters and observer gains

Vehicle parameters	
m	1400 [kg]
τ	0.1
c_0	144.207 [N]
c_1	4 [Ns/m]
c_2	0.3803 [Ns$^2/m$]
Observer gains	
l_1	10
l_2	1000

The simulation (the code which is given in Appendix B) is done for 300 seconds, with timestep $\Delta t = 0.01s$, i.e., position measurements are assumed to be obtained at 100 Hz. The true and estimated state trajectories plotted in Figure 4.1, indicate that the observer error converges to zero for all states. The convergence rate of velocity estimation (to true trajectory) is slower, compared to position and force. This is likely because velocity observer is subject to tuning of both l_1 and l_2. It is also worth noting that the observer is working fine even when the velocity became negative, giving us more confidence that the observer may be working globally.

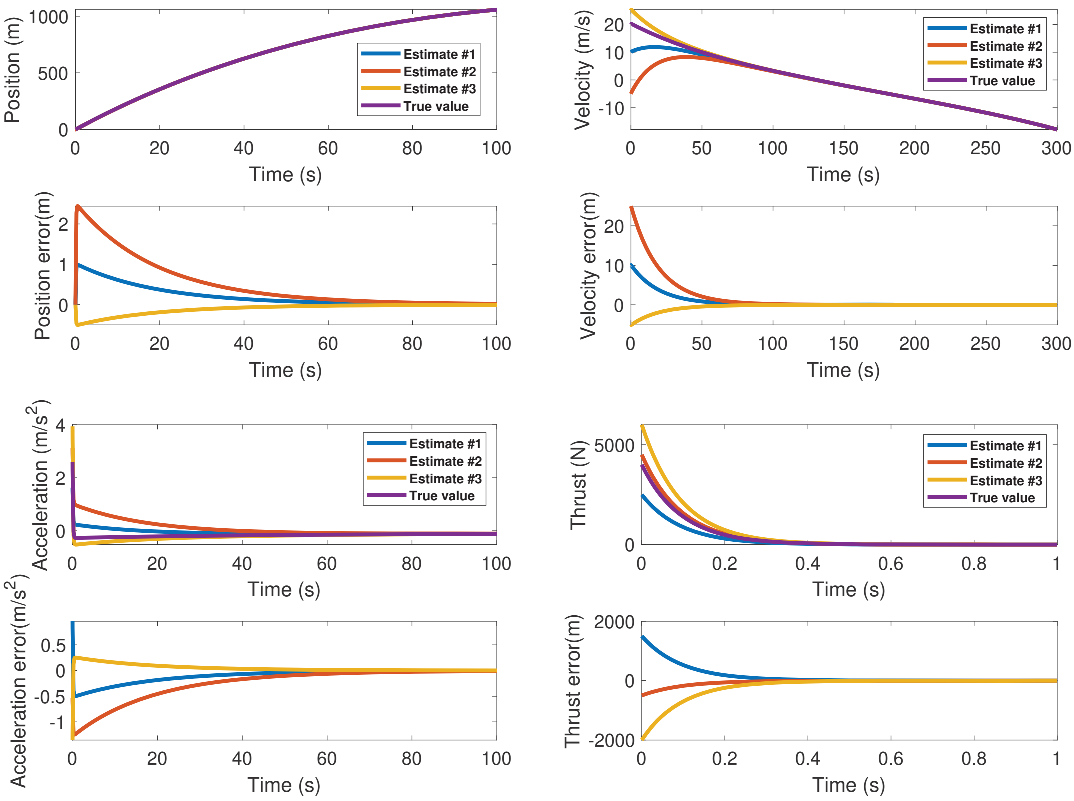

Figure 4.1: State trajectories and estimation errors pertaining to Tables 4.1 and 4.2

4.5 Summary

First, the local and global observability of the full system is assessed. Then, it is found out that the system does not have a coordinate transformation, which leads to linearized error dynamics. Hence, the full-state observer is designed directly by substitution of true force value F, with its estimated version $\hat{F}$. The stability check according to [17] shows that the full-state observer works globally, if the vehicle is restricted to forward or backward movement only. Finally, the mathematical conclusion is verified by a simulation in MATLAB.

In the next chapter, an observer-based degraded CACC framework is proposed.

Chapter 5

Observer-based CACC and simulations

This chapter is dedicated to design of the controller for the vehicle subject to absence of global velocity and acceleration measurements, the latter of which depends on both velocity and force terms. In CACC, a spacing error function is constructed. This function denotes the error in the desired intervehicular distance the vehicle tries to achieve with the predecessor. In CACC, there are two main objectives that must be achieved:

- The system achieves stable tracking dynamics; global velocity, acceleration and control input remains bounded when incoming signal on predecessor information is bounded. Additionally, the tracking error dynamics is 0-GAS i.e., $a_{i-1} = 0$ implies the spacing error globally asymptotically converges to zero.
- String stability is achieved; the energy of the disturbances in traffic flow does not grow in the direction of platoon upstream. Mathematically, this is equivalent to $||a_i||_{\mathcal{L}_2} \leq ||a_{i-1}||_{\mathcal{L}_2}$ for any follower vehicle i [11].

The design of CACC will follow a similar format as a-CACC; except now we have a nonlinear vehicle model. In [9], it is observed that u-CACC and observer-based a-CACC has similar performance in achieving the two objectives above. a-CACC also removes the necessity to require information on τ_{i-1} and u_{i-1}.

5.1 CACC with perfect measurements

We start by defining the spacing error e_s by [7]:

$$e_s = q_{i-1} - q_i - (L_i + h_i v_i) \tag{5.1}$$

where subscript $(i-1)$ denotes predecessor vehicle, while $(L_i + h_i v_i)$ is the desired intervehicular distance. The length of the vehicle is denoted by L_i, while $h_i \in \mathbf{R}^+$ is the desired time headway of the vehicle. It is assumed that the vehicle i processes predecessor information with the following model below:

$$\begin{aligned} \dot{q}_{i-1} &= v_{i-1} \\ \dot{v}_{i-1} &= a_{i-1}. \end{aligned} \tag{5.2}$$

The first and second time derivatives of spacing error are computed as follows:

$$\begin{aligned} \dot{e_s} &= v_{i-1} - v_i - h_i \dot{v}_i \\ &= v_{i-1} - \left(1 - \frac{h_i c_1}{m_i}\right) v_i + \frac{h_i c_2}{m_i} v_i^2 - \frac{h_i}{m_i}(F_i - c_0) \end{aligned} \tag{5.3}$$

$$\begin{aligned} \ddot{e_s} &= a_{i-1} - \frac{h_i}{m_i}\dot{F}_i - \left(1 - \frac{c_1 h_i}{m_i} - \frac{2h_i c_2}{m_i} v_i\right) \dot{v}_i \\ &= a_{i-1} + \frac{h_i}{m_i \tau_i}(F_i - \bar{u}_i) - \left(1 - \frac{c_1 h_i}{m_i} - \frac{2h_i c_2}{m_i} v_i\right) \dot{v}_i. \end{aligned} \tag{5.4}$$

As the spacing error to converge to zero, the relative velocity $e_v = v_{i-1} - v_i$ should also converge to zero (otherwise, the platoon may not remain in equilibrium post-convergence). Hence, we shall add e_v as another error coordinate. Its time derivative is as follows:

$$\dot{e_v} = a_{i-1} - \dot{v}_i = a_{i-1} - \frac{1}{m_i}\left(F_i - c_0 - c_1 v_i - c_2 v_i^2\right). \tag{5.5}$$

The tracking error state vector is denoted by $e_c = [e_{s1} \quad e_{s2} \quad e_v]^T$, where

$$\begin{aligned} e_{s1} &= e_s \\ e_{s2} &= \dot{e_s} = \dot{e}_{s1}. \end{aligned} \tag{5.6}$$

Combining (5.3), (5.5) and (5.6); $\dot{v}_i$ can be rewritten as

$$\dot{v}_i = \frac{e_v - e_{s2}}{h_i}. \tag{5.7}$$

The controller employed is a combination of a PD-controller and a feed forward as follows:

$$\bar{u}_i = \frac{m_i \tau_i}{h_i}\left[K_p e_{s1} + K_d e_{s2} - \left(1 - \frac{c_1 h_i}{m_i} - \frac{2h_i c_2}{m_i} v_i\right) \dot{v}_i + a_{i-1}\right] + F_i \tag{5.8}$$

where K_p and K_d are constant proportional and derivative gains of the controller, respectively. The goal behind the design of (5.8) is to obtain a linear form for the tracking error dynamics. Note that we will need to replace v_i with the observer estimate $\hat{v}_i$, when combining the tracking and observer error dynamics. However, for this step, the goal is to first ensure stability when measurements are available.

After implementing the controller (5.8), the tracking error dynamics is obtained as

$$\begin{bmatrix} \dot{e_{s1}} \\ \dot{e_{s2}} \\ \dot{e_v} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -K_p & -K_d & 0 \\ 0 & \frac{1}{h_i} & -\frac{1}{h_i} \end{bmatrix}}_{A_c} \begin{bmatrix} e_{s1} \\ e_{s2} \\ e_v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} a_{i-1} \tag{5.9}$$

Here, a_{i-1} can be treated as an input to the tracking error dynamics. Hence, in order to achieve asymptotic stability, the eigenvalues of A_c must be in LHP. The eigenvalues are computed as follows:

$$\left\{ -\frac{1}{h_i}, \frac{-K_d \pm \sqrt{K_d^2 - 4K_p}}{2} \right\}. \tag{5.10}$$

For asymptotic stability, it is sufficient to have $K_p > 0$ and $K_d > 0$.

In order to show ISS with respect to a_{i-1}, first assume that $a_{i-1} = 0$. We will take the Lyapunov candidate function:

$$V_1 = e_c^T P e_c \tag{5.11}$$

where P is a square matrix to be determined; and $e_c = [e_{s1}\ e_{s2}\ e_v]^2$ is the tracking error state vector. Taking the time derivative of V yields the following:

$$\dot{V_1} = e_c^T [A_c^T P + P A_c] e_c = -e_c^T Q e_c \tag{5.12}$$

for some matrix Q. This implies that

$$A_c^T P + P A_c = -Q. \tag{5.13}$$

We shall choose an arbitrary positive definite matrix Q first, say $Q = I$. This would automatically satisfy the conditions $\dot{V_1} \le 0\ \forall e_c$; and $\dot{V_1} = 0 \iff e_c = 0$. Then, the goal is to verify the range of values for K_p and K_d such that V_1 remains positive definite. Solving (5.11) and (5.13) simultaneously yields the Lyapunov function:

$$\begin{aligned} V_1 = [&K_d K_p (e_{s2} + e_v)^2 + K_d K_p^2 h_i (e_{s1} - h_i e_v)^2 + K_d^2 K_p h_i^2 e_v^2 \\ &+ (K_p h_i^2 + K_d h_i + 1)(K_d e_{s1} + e_{s2})^2 + (2K_p + K_p^2 h_i^2) e_{s2}^2 \\ &+ (K_p^3 h_i^2 + K_p^2 h_i^2 + 2K_p^2 + K_p + K_d K_p h_i + K_d K_p^2 h_i) e_{s1}^2] \\ &/\ [2K_p K_d (K_p h_i^2 + K_d h_i + 1)] \end{aligned} \tag{5.14}$$

which purely consists of quadratic terms. Hence, in order for V_1 to be positive definite for all values of e_c, the leading coefficient of each quadratic term must be positive. The leading coefficients of first and fourth quadratic term are positive iff

$$
\begin{aligned}
(K_p h_i^2 + K_d h_i + 1)^{-1} &> 0 \\
K_p h_i (K_p h_i^2 + K_d h_i + 1)^{-1} &> 0 \\
K_d h_i^2 (K_p h_i^2 + K_d h_i + 1)^{-1} &> 0 \\
(K_p K_d)^{-1} &> 0.
\end{aligned}
\tag{5.15}
$$

The inequalities in (5.15) are satisfied iff $K_p, K_d, h_i > 0$. Under this condition, it is clear that the remaining quadratic terms have positive leading coefficients. Consequently, V_1 is positive definite $\forall e_c$; and is equal to zero iff $e_c = 0$. Hence, the tracking error dynamics is 0-GAS for $K_p, K_d, h_i > 0$. The system (5.9) is also in LTI form; therefore, by [20], it can is concluded that the tracking error dynamics is ISS with respect to a_{i-1}. In other words, bounded predecessor acceleration measurement leads to bounded tracking error state e_c.

5.2 Degradation of CACC

Since global velocity and force measurements are not available, these variables are generated by their estimates. Then, the controller (5.8) becomes:

$$
\hat{\bar{u}}_i = \frac{m_i \tau_i}{h_i} \left[K_p \hat{e}_{s1} + K_d \hat{e}_{s2} - \left(1 - \frac{c_1 h_i}{m_i} - \frac{2 h_i c_2}{m_i} \hat{v}_i \right) \dot{\hat{v}}_i + a_{i-1} \right] + \hat{F}_i \tag{5.16}
$$

where $\dot{\hat{v}}_i = \hat{a}_i$ is the acceleration estimate obtained from the full-state observer. Meanwhile, estimates of tracking errors are generated by

$$
\begin{aligned}
\hat{e}_{s1} &= q_{i-1} - \hat{q}_i - L_i - h_i \hat{v}_i \\
\hat{e}_{s2} &= v_{i-1} - \dot{\hat{q}}_i - h_i \dot{\hat{v}}_i = \dot{\hat{e}}_{s1} \\
\hat{e}_v &= v_{i-1} - \dot{\hat{q}}_i.
\end{aligned}
\tag{5.17}
$$

The controller can be rewritten as

$$
\begin{aligned}
\hat{\bar{u}}_i &= \frac{m_i \tau_i}{h_i} \left[K_p \hat{e}_{s1} + K_d \hat{e}_{s2} + a_{i-1} \right] - \left(\frac{m_i \tau_i}{h_i} - c_1 \tau_i - 2 c_2 \tau_i \hat{v}_i \right) \dot{\hat{v}}_i + \hat{F}_i \\
&= \bar{u}_i + (\hat{F}_i - F_i) - \frac{m_i \tau_i}{h_i} \left[K_p (e_{s1} - \hat{e}_{s1}) + K_d (e_{s2} - \hat{e}_{s2}) \right] \\
&+ \left(\frac{m_i \tau_i}{h_i} - c_1 \tau_i \right) (\dot{v}_i - \dot{\hat{v}}_i) - 2 c_2 \tau_i (v_i \dot{v}_i - \hat{v}_i \dot{\hat{v}}_i) \\
&= \bar{u}_i + \gamma(.).
\end{aligned}
\tag{5.18}
$$

Under the assumption $v_i \geq 0$, we had established the observer error decays to zero uniform asymptotically. Hence, the function $\gamma(.)$ composed of additional terms due to degradation of CACC, also decays to zero as time $t \to \infty$. Since $\gamma(.)$ is continuous, it remains bounded for bounded predecessor information. Summation of two bounded and continuous functions is bounded, the new input $\hat{\bar{u}}_i$ remains bounded. Whether this new input leads to asymptotic convergence of spacing error to zero, is tested with MATLAB in the next section.

5.3 Simulation tests

5.3.1 Observability of the system in a mix of forward and backward motion

This subsection provides a counterexample to the observability of the system (2.4) pertaining to Case (iii) of Section 4.3. For this counterexample, it is assumed the predecessor vehicle $i-1$ has the same mathematical model (2.4) as the follower vehicle i; and has the following input:

$$\bar{u}_{i-1} = \begin{cases} 2000 & t \in [10, 30) \\ -2000 & t \in [50, 70) \\ 0 & otherwise \end{cases} \; [N]. \tag{5.19}$$

Without counteracting forces, this input translates into acceleration and deceleration of about $a_{i-1} \approx 1.4\ m/s^2$, which is a reasonable value. It is assumed that vehicles are initially in steady state; that is, they travel at same initial velocity of 20 m/s. The state estimate of vehicle i can be an arbitrary real column vector. In this regard, the simulation is conducted with these initial conditions (with same SI units as in Table 4.1):

$$x_{i-1}(0) = [70 \quad 20 \quad 0]^T \quad x_i(0) = [0 \quad 20 \quad 0]^T \quad \hat{x}_i(0) = [0 \quad 30 \quad 50]^T$$

In order to have a starting point, desired time headway for the follower vehicle is taken as $h = 1.75\ s$, in accordance with the string stability analysis of [9]. The observer gains $l_1 = l_2 = 10$ are assigned based on the lower bounds in (3.14). The values of the remaining constants can be found in Appendix A. The simulation (the code which is given in Appendix C.1) is conducted for 300 seconds, with sampling time $\Delta t = 0.01s$. For consistency, the true value is also computed with the same Δt (instead of using ode45, which has varying timestep).

In Figure 5.1, it can be seen that the observer struggles to drive the velocity estimate to its true value. In its corresponding error plot, we see that the offset between estimate and true value starts increasing at around $t = 70s$, where the vehicle has zero velocity. For acceleration, it is less obvious. Within the first 100 seconds, the acceleration estimation seems to be following the true value quite nicely; and the acceleration seems to converge to zero by then. However, extension of the simulation to 300 seconds shows that the acceleration error does not stay at zero. This is supported by the visible offset in acceleration curves starting from $t = 200\ s$.

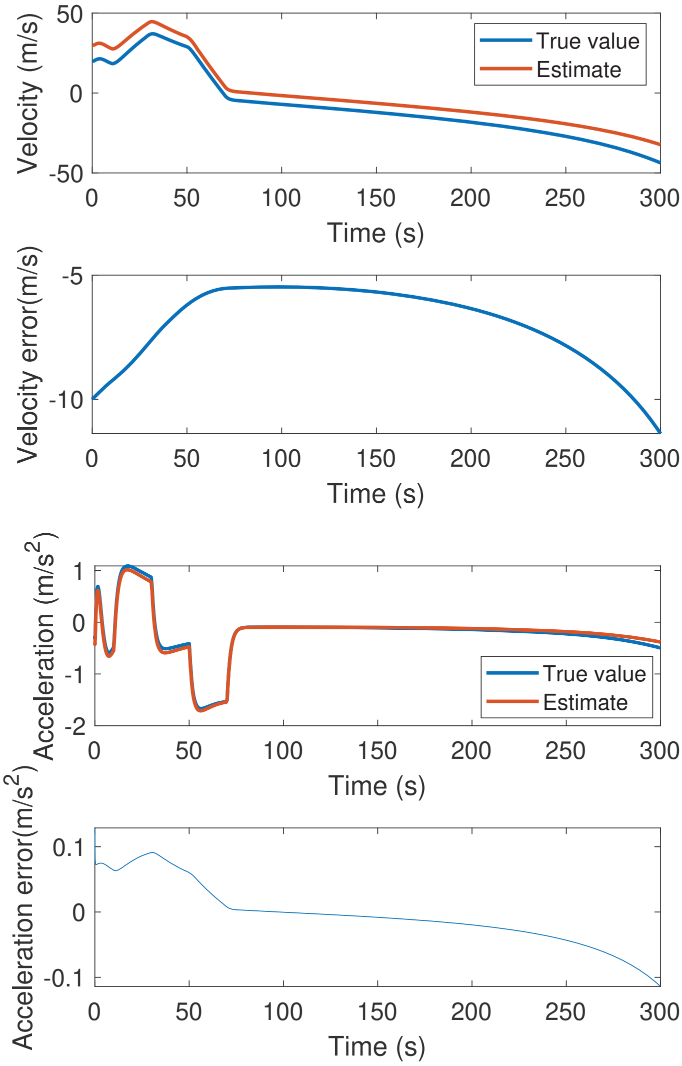

Figure 5.1: Counterexample to observability of Case (iii)

In regards to Section 4.4, we had confidence that the observer could work globally. However, this simulation gave us unexpected results. Along the mathematical proof in Section 4.3, it is guaranteed that the observer error converges to zero asymptotically if and only if the vehicle is restricted to moving forward or backward.

5.3.2 Stability of tracking dynamics

In previous subsection, it was found that a mix of forward an backward motion lead to instability in observer error dynamics, which subsequently led to spacing error to also grow. Therefore, the input of the leader vehicle should be revised such that in steady state, the thrust force dominates the rolling resistance and counteracting drag forces. The new input

(5.20) for the leader below is designed to test the performance of the follower vehicle's controller against sudden thrust or braking of the leader, while still keeping the vehicle platoon in forward motion. This input is

$$\bar{u}_{i-1} = \begin{cases} 4000 & t \in (10, 30] \\ -2000 & t \in (50, 70] \\ 1000 & otherwise \end{cases} [N]. \tag{5.20}$$

The simulation is conducted with the same initial true state values as Section 5.3.1:

$$x_{i-1}(0) = [70 \quad 20 \quad 0]^T \quad x_i(0) = [0 \quad 20 \quad 0]^T.$$

Instead of assigning a single initial value for follower state estimate, we create vectors for initial estimates of velocity and force as follows:

$$\hat{v}_i(0) = -50, \dots, -1, 0, 1, \dots, 50 \ [m/s]$$
$$\hat{F}_i(0) = -5000, \dots, -100, 0, 100, \dots 5000 \ [N]$$

Since measurement on global position is available, it is assumed $q_i(0) = \hat{q}_i(0) = 0$. In order to reduce simulation time, the timestep is increased to $\Delta t = 0.1\ s$. The time headway and observer gains are kept the same (i.e. $h_i = 1.75\ s$ and $l_1 = l_2 = 10$). The vehicle parameters and control values can be found in Appendix A, while the simulation code can be found in Appendix C.2.

Mesh surfaces in Figures 5.2 and 5.3 denote the spacing error e_{s1} and its rate e_{s2}, as a function of initial velocity and force estimates. It is observed that for any initial estimate, the spacing error and its rate converge to zero as time goes to infinity. In other words, when the spacing error converges to zero, it stay zero. Even at time instances of 50 and 100 seconds (i.e., soon after predecessor's gas and braking signals, respectively), a considerable amount of reduction in spacing error occurs. Based on the simulations, we may conclude that the observer-based controller leads to stable tracking dynamics for a sufficiently large pool of initial state estimation.

5.3.3 String stability

For string stability analysis, the robustness of CACC controller is tested with a harmonic signal for predecessor acceleration along the linear system (5.2):

$$a_{i-1} = \cos(\omega t) \tag{5.21}$$

where a_{i-1} varies with the frequency of the signal, ω. In order to avoid an observability issue similar to the one in Section 5.3.1, a cosine wave is preferred over more commonly used sine wave. The driveline dynamics present in follower vehicle's model (2.4) acts as a low-pass filter; hence, it is found to be sufficient that the frequency of (5.21) is kept at low values, say $\omega \in [0, 0.2]$ Hz. Initially, the vehicles start at steady state; they have constant

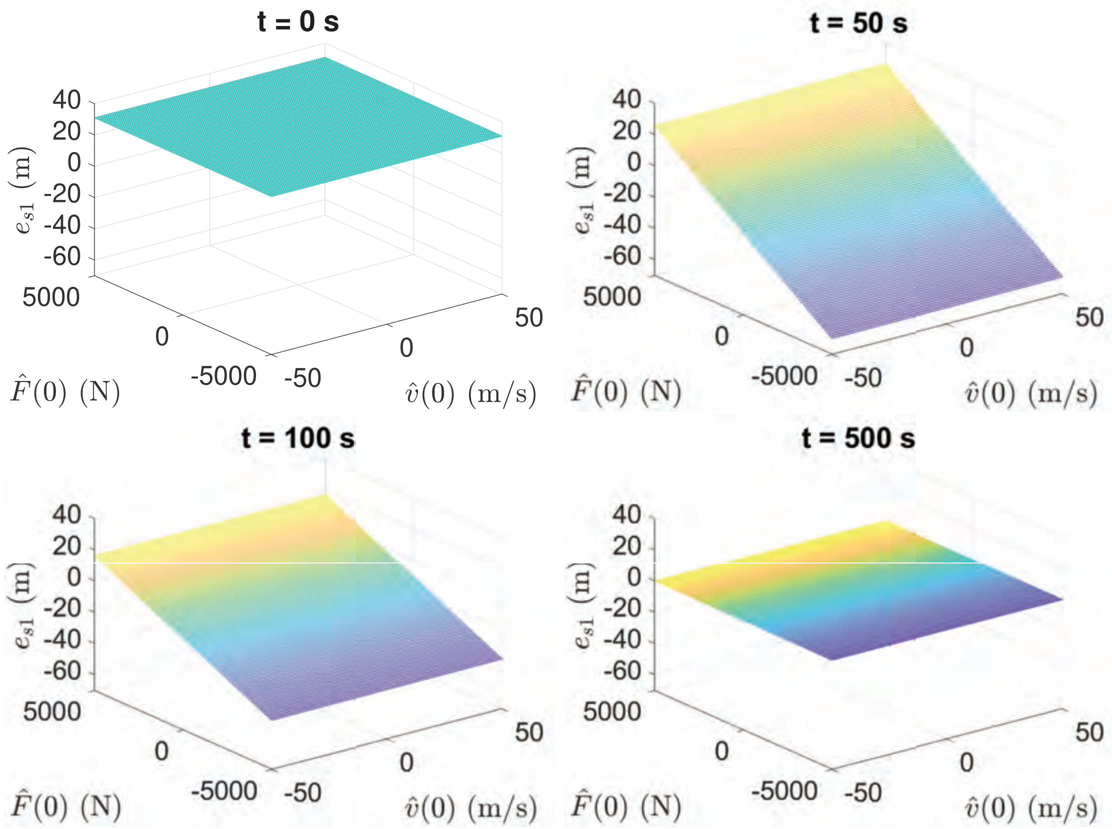

Figure 5.2: Spacing error at times t = 0, 50, 100, 500 s

initial velocity of 20 m/s at time $t_0 = 0$.

Recall that for string stability, it is required that $||a_i||_{\mathcal{L}_2} \leq ||a_{i-1}||_{\mathcal{L}_2}$ for any follower vehicle i. We go with a numerical approach and define the ratio of the accelerations as

$$|\Gamma(j\omega)|_{\mathcal{L}_2} = \frac{||a_i(j\omega)||_{\mathcal{L}_2}}{||a_{i-1}(j\omega)||_{\mathcal{L}_2}} = \frac{\sqrt{\sum_k \left|a_i^{(k)}(j\omega)\right|^2}}{\sqrt{\sum_k \left|a_{i-1}^{(k)}(j\omega)\right|^2}} \tag{5.22}$$

where superscript (k) stands for k-th output in discrete-time. In MATLAB, $\mathcal{L}_2$ norm can be computed with the command norm(X,2); with X replaced by the vector of all values of acceleration output. For string stability, it is aimed that

$$|\Gamma(j\omega)|_{\mathcal{L}_\infty} = \max_{\omega\in[0,\infty)}|\Gamma(j\omega)|_{\mathcal{L}_2} \leq 1. \tag{5.23}$$

The simulations are conducted in two steps: Observer tuning and determining minimum time headway. Due to CACC controller (5.16) being composed of observer estimations, it is hypothesized that the choice of observer gains have a significant effect on the string

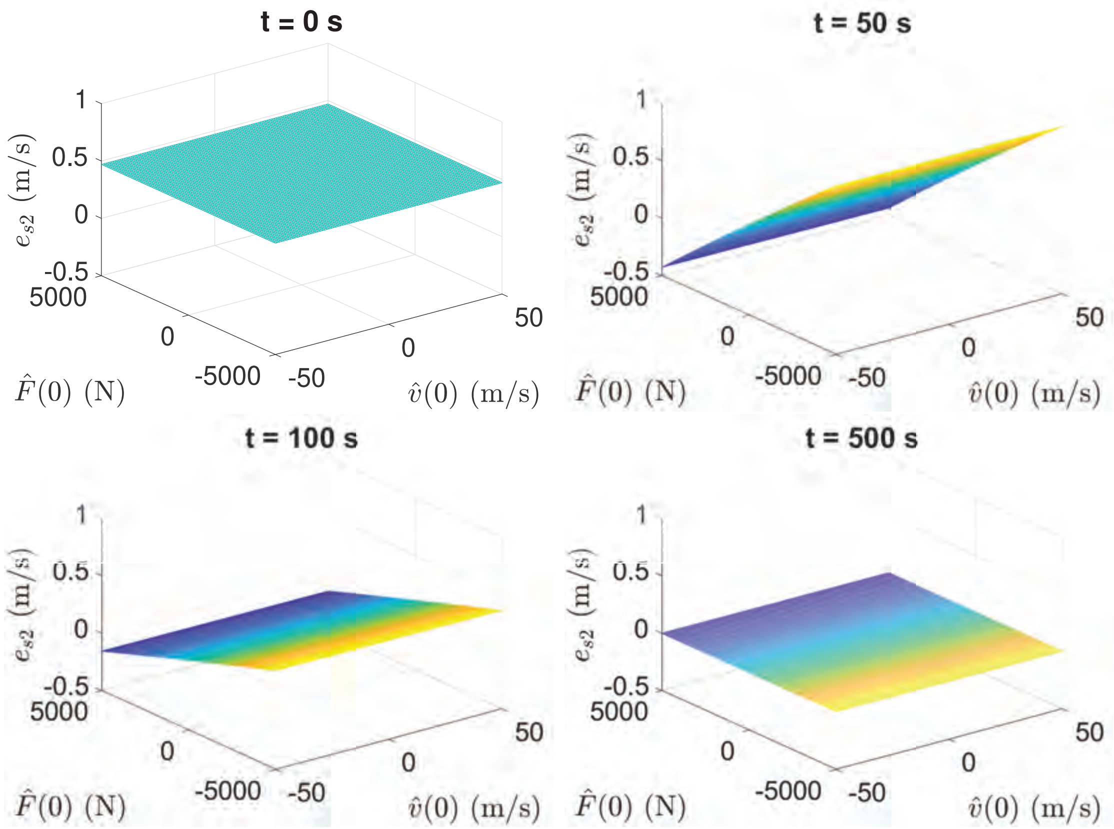

Figure 5.3: Spacing error rate at times t = 0, 50, 100, 500 s

stability. Then, with an appropriate choice of observer gains, a minimum required time headway can be computed numerically. The MATLAB simulation codes for both steps can be found in Appendix D.1 and D.2, in respective order.

Step 1: Observer tuning

In this step, the effects of observer tuning on the string stability is studied. This is tested by assigning a sufficiently large pool of values for observer gains l_1 and l_2. The range of values used are

$$\begin{aligned} l_1 &= 1, 2, \ldots, 20 \\ l_2 &= 1, 201, \ldots, 4001. \end{aligned}$$

The simulation is tested with the following initial conditions:

$$x_{i-1}(0) = [70 \quad 20 \quad 0]^T \quad x_i(0) = [0 \quad 20 \quad 0]^T \quad \hat{x}_i(0) = [0 \quad 10 \quad 500]^T.$$

The sampling time is set to $\Delta t = 0.01$ s and the simulation is run for 100 seconds with the leader acceleration (5.21) fixed at a low frequency of $\omega = 0.01$ Hz, since it is expected $|\Gamma|_{\mathcal{L}_2}$ in (5.23) will be larger at lower frequencies. The simulation is done with two distinct time headway values.

Figure 5.4 illustrates the effect of tuning the observer gains, on the value of $|\Gamma|_{\mathcal{L}_2}$. For low time-headway ($h = 1.75$ s), the minimum acceleration ratio seems to be located at around $(l_1, l_2) = (20, 600)$. However, the gradient in $|\Gamma|_{\mathcal{L}_2}$ is significant around that point, which may imply that the location of absolute minimum was mostly determined by the specific numerical values of initial conditions. For both time headway instances, choosing high l_1 with low l_2 seems to be undesirable due to large $|\Gamma|_{\mathcal{L}_2}$ value. Since it is aimed to achieve string stability by gradually increasing time headway, it is preferred that the gradient of $|\Gamma|_{\mathcal{L}_2}$ around the chosen observer gain coordinates is low (i.e. the mesh surface remains relatively flat at both time headway instances). In this regard, it is best to have l_1 in mid-range value, while l_2 is within high-end range. However, no definitive correlation that is invariant of initial conditions could be made between observer gains and string stability.

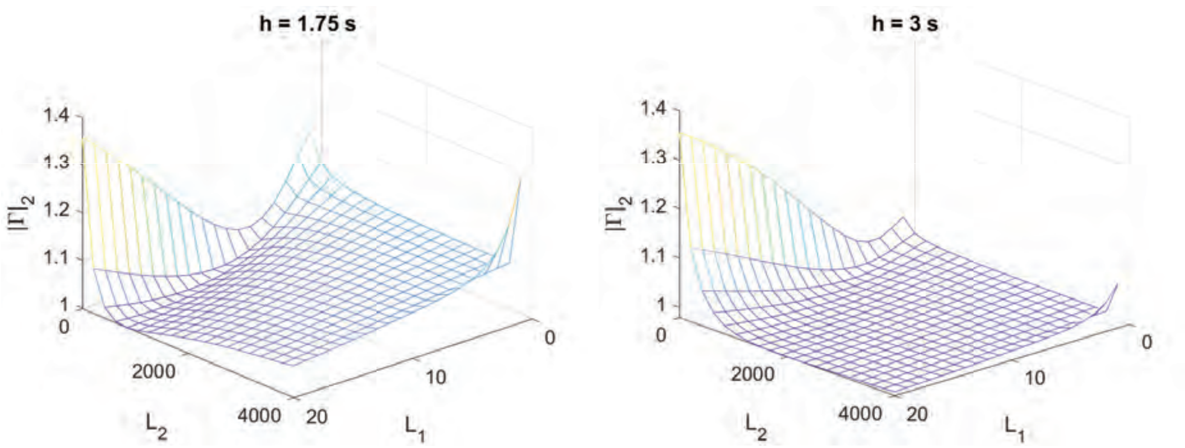

Figure 5.4: $|\Gamma|_{\mathcal{L}_2}$ for a large pool of observer gains at two time headway instances

Step 2: Determining minimum time headway

In this step, the minimum time headway required to respond to the harmonic signal (5.21) is numerically computed. In light of conclusions made in Step 1, the observer gains are now fixed at

$$l_1 = 10, \qquad l_2 = 3000.$$

Similar to Step 1, $|\Gamma|_{\mathcal{L}_2}$ is computed through MATLAB simulation. This time, though, we consider all frequencies $\omega \in [0, 0.2]$ Hz. The simulation is run repeatedly by manually increasing the time headway with small increments. It is found that string stability is achieved for $h \geq 2.44\ s$. The left plot of Figure 5.5 shows that the acceleration norm curve of the leader is always above the follower, implying that the two-vehicle platoon is string stable. The right plot indicates that maximum value of $|\Gamma|_{\mathcal{L}_2}$ is achieved at about 0.015

Hz. The mean value of each period of the curve also decreases with increased frequency, which validates our hypothesis that the platoon is more likely to stay string stable to higher frequency harmonic signals.

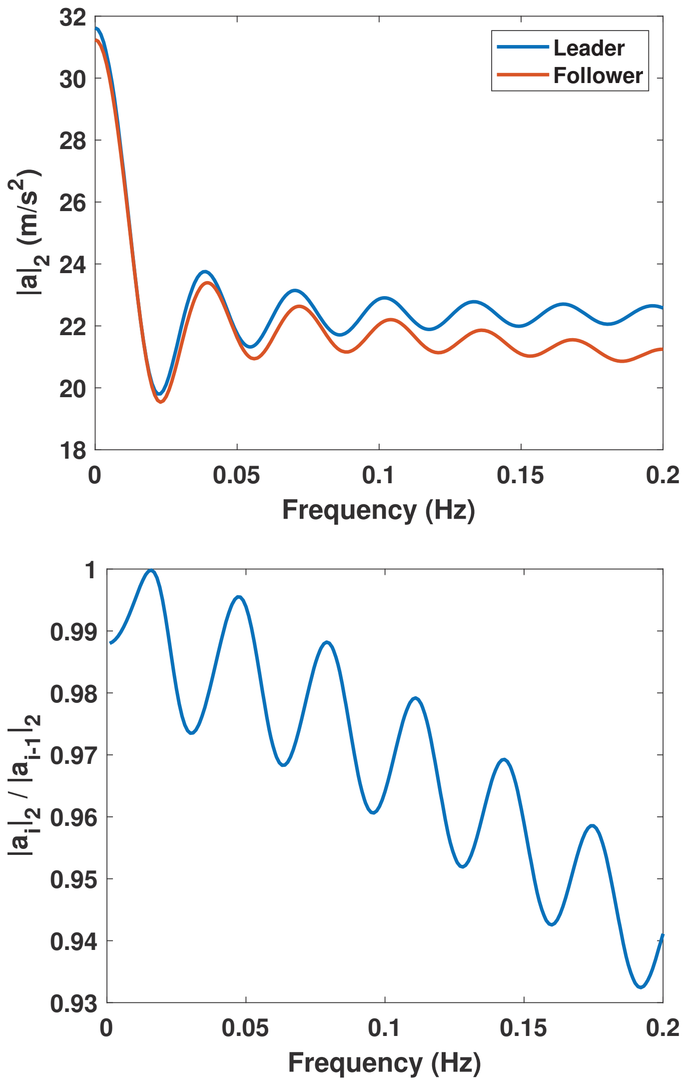

Figure 5.5: Acceleration norms and norm ratio of two vehicles at $h = 2.44\ s$

Chapter 6

Conclusion and future work

6.1 Conclusion

In [9], an observer-based a-CACC framework is proposed for a vehicle (with linear model) subject to loss of global and predecessor acceleration measurements. In this thesis, we provide an extension to the existing a-dCACC by including the effects of counteracting forces such as rolling resistance, damping and drag forces; thus leading to a nonlinear vehicle model. In addition to global acceleration measurement, the vehicle is also subject to loss of global velocity measurement, due to inherent measurement noise of wheel encoder or LiDAR's inefficiency in heavy weather conditions.

It is found that if the applied force of the vehicle is controlled directly, by considering force as a controllable input, a global observer can be designed with an appropriate change of coordinates to input and state. However, the incorporation of linear driveline dynamics into the model makes it quite difficult to design a global observer, since a suitable change of coordinates to obtain linearized error dynamics does not exist for the full-order model. This brings speculation as to whether the full-order model is globally observable because the existing criteria [21, 22] on global observability of the polynomial systems lead to inconclusive result, due to everlasting increase in power of the velocity term with each order of Lie derivative of output (i.e., position q). Even so, we managed to successfully design an observer for strictly forward and backward motion of the vehicle, by mathematically proving stability with a cascaded system approach.

Finally, an a-CACC framework is proposed, in which the controller uses the velocity and force estimates obtained from the observer. Although a complete mathematical proof has not been provided, the stability of tracking dynamics is shown through repeated simulations. It is found that even for large initial estimation errors, the spacing error decays to zero, giving confidence that a mathematical proof can be made if we assume true velocity to be bounded. The stability analysis was also conducted numerically. In the end, string stability could be achieved by gradually increasing the time headway, a value which depends on the initial spacing error. On the other hand, no definitive correlation could be made between tuning the observer and achieving string stability.

6.2 Future Work

The main improvement we recommend is about the vehicle model. Currently, it takes rolling resistance and friction to be constant. In addition, the rolling resistance and drag forces always act in same direction, regardless of the direction vehicle is moving. This may be the root of the global observability problem. A solution might be to treat c_0, c_1, c_2 as dependent variables and include them in the space vector. A simpler, though may not be the most accurate, option could be to modify the model (2.4) such that

$$\begin{aligned}
\dot{q}_i &= v_i \\
\dot{v}_i &= \frac{1}{m_i}[F_i - (c_0 sign(v_i) + c_1 v_i + c_2 v_i |v_i|)] \\
\dot{F}_i &= -\frac{1}{\tau_i} F_i + \frac{1}{\tau_i} \bar{u}_i.
\end{aligned} \tag{6.1}$$

This small change would lead to counteracting forces always acting in opposite direct of vehicle's movement. In this way, there is a higher likelihood that the system becomes globally observable even in the case, where the vehicle switches between forward and backward motion.

Although we know that the string stability is dependent on the time headway, the time headway is not automatically adjusted to accommodate for string stability. Hence, the automation of observer tuning and time headway adjustment in a nonlinear system can be one of the future topics. In addition, the effects of communication delay and predecessor signal noise can be investigated.

Bibliography

[1] Peter Kingsford. James Watt. https://www.britannica.com/biography/James-Watt. 1

[2] The 1908 Peerless Motors and Trans Mission. https://www.scientificamerican.com/article/the-1908-peerless-motors-and-trans/, Nov 1907. 1

[3] Ralph Teetor. Speed control device for resisting operation of the accelerator, 1950. US Patent 2,519,859. 1

[4] Explained: the different kinds of cruise control and how they work. https://www.shropshirestar.com/news/motors/features/2019/01/29/explained-the-different-kinds-of-cruise-control-and-how-they-work/. 1

[5] Greg Marsden, Mike McDonald, and Mark Brackstone. Towards an understanding of adaptive cruise control. *Transportation Research Part C: Emerging Technologies*, 9(1):33–51, 2001. 1

[6] Jeroen Ploeg, Bart Scheepers, Ellen van Nunen, Nathan van de Wouw, and Henk Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 5-7 October 2011, Washington D.C.*, pages 260–265, United States, 2011. Institute of Electrical and Electronics Engineers. 1

[7] Erjen Lefeber, Jeroen Ploeg, and Henk Nijmeijer. Cooperative adaptive cruise control of heterogeneous vehicle platoons. *IFAC-PapersOnLine*, 53:15217–15222, 01 2020. 2, 25

[8] Jeroen Ploeg, Elham Semsar-Kazerooni, Guido Lijster, Nathan van de Wouw, and Henk Nijmeijer. Graceful degradation of CACC performance subject to unreliable wireless communication. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1210–1216, 2013. 2

[9] Max Bolderman, Erjen Lefeber, Jeroen Ploeg, and Henk Nijmeijer. *Observer-based control for string stable CACC within heterogeneous vehicle platoons*. Number 2020.055 in DC. Technische Universiteit Eindhoven, May 2020. 2, 25, 29, 37

[10] Hao Wang, Yanyan Qin, Wei Wang, and Jun Chen. Stability of CACC-manual heterogeneous vehicular flow with partial CACC performance degrading. *Transportmetrica B: Transport Dynamics*, 7:1–26, 09 2018. 2

[11] Shuo Feng, Yi Zhang, Shengbo Eben Li, Zhong Cao, Henry X. Liu, and Li Li. String stability for vehicular platoon control: Definitions and analysis methods. *Annual Reviews in Control*, 47:81–97, 2019. 2, 25

[12] Swaroop Darbha and J. Karl Hedrick. String stability of interconected systems. volume 41, pages 1806 – 1810 vol.3, 07 1995. 2

[13] Zhiyuan Li, Huawei Liang, Pan Zhao, Shaobo Wang, and Hui Zhu. Efficent lane change path planning based on quintic spline for autonomous vehicles. In *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 338–344, 2020. 2

[14] Shahab Sheikholeslam and Charles A. Desoer. Longitudinal control of a platoon of vehicles. In *1990 American Control Conference*, pages 291–296, 1990. 5

[15] Hassan Kamal Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002. 6, 12, 13

[16] Arthur Krener and Witold Respondek. Nonlinear observer with linearizable error dynamics. *SIAM Journal on Control and Optimization*, 23, 04 1985. 7, 10

[17] Elena Panteley and Antonio Loria. On global uniform asymptotic stability of nonlinear time-varying systems in cascade. *Systems & Control Letters*, 33(2):131–138, 1998. 8, 22, 24

[18] Elena Panteley, Erjen Lefeber, Antonio Loria, and Henk Nijmeijer. Exponential tracking control of a mobile car using a cascaded approach. *IFAC Proceedings Volumes*, 31(27):201–206, 1998. IFAC Workshop on Motion Control (MC'98), Grenoble, France, 21-23 September. 8

[19] 2022 Prius full specs. `https://www.toyota.com/prius/2022/features/mpg_other_price/1221/1223/1263`. 23, 43

[20] Andrii Mironchenko and Fabian Wirth. A note on input-to-state stability of linear and bilinear infinite-dimensional systems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 495–500, 2015. 28

[21] Yu Kawano and Toshiyuki Ohtsuka. Global observability of polynomial systems. In *Proceedings of SICE Annual Conference 2010*, pages 2038–2041, 2010. 37

[22] Daniel Gerbet and Klaus Röbenack. On global and local observability of nonlinear polynomial systems: a decidable criterion. *at - Automatisierungstechnik*, 68:395–409, 06 2020. 37

[23] Low rolling resistance tires. http://righg.raabassociates.org/Articles/CGR%20on%20tire%20rolling%20resistance%20FINAL.pdf. 43

Appendix A

Vehicle parameters

The specs of Toyota Prius [19] are taken as reference for assigning the vehicle paramters c_0, c_1 and c_2. The rolling resistance c_0 can be computed by,

$$c_0 = C_{rr}N = C_{rr}mg \tag{A.1}$$

where $N = mg$ is the normal force acting on the vehicle, with vehicle mass $m = 1400\ kg$ and gravitational acceleration $g = 9.81\ m/s^2$. Toyota Prius uses low rolling resistance tires, so the average rolling resistance coefficient is taken from the catalogue [23]:

$$C_{rr} = 0.0105. \tag{A.2}$$

Substituting the numerical values, we obtain

$$c_0 = 144.207\ N. \tag{A.3}$$

The damping coefficient was not available explicitly in the vehicle specs. Hence, an acceptable value for damping coefficient is assigned as,

$$c_1 = 4\ Nm/s. \tag{A.4}$$

The drag force D (with back pressure neglected) is computed by

$$D = \frac{1}{2}\rho S C_D v^2 = c_2 v^2 \tag{A.5}$$

with air density $\rho = 1.225\ kg/m^3$ at sea level and drag coefficient $C_D = 0.24$. Maximum velocity of Toyota Prius is 180 km/h (or 50 m/s), which is approximately 0.15 times the speed of sound (Mach) at sea level. Below Mach 0.3, the change in air flow density is less than 5%; thus, the air flow can be assumed to be incompressible (i.e. air density ρ can be assumed constant). The drag coefficient is solely dependent on vehicle geometry, so it is also constant.

The surface area S is taken to be the front cross-sectional area of the vehicle. That is,

$$S = (car\ width) \times (car\ height) = 1.760m \times 1.470m \tag{A.6}$$

Substitution of the numerical values leads to

$$D = 0.3803v^2 \Rightarrow c_2 = 0.3803 \; Ns^2/m^2. \tag{A.7}$$

The remaining constants are set to the following values:

$$\begin{aligned} \tau &= 0.1 \\ L &= 4 \; m \\ K_p &= 0.2 \\ K_d &= 0.7. \end{aligned} \tag{A.8}$$

Unless stated otherwise, the platoon is assumed to be homogeneous with same c_0, c_1, c_2 coefficients.

Appendix B

Observer test

This appendix contains the MATLAB code for testing the observer in Section 4.4.

```
    clear all; close all; clc;

INIT = [0 20 4000];
INITH = [0 10 2500; 0 -5 4500; 0 25 6000];
rho = 1.225; Cd = 0.24; S = 1.470*1.760; nu = 0.7; g =9.81;
q(1) = INIT(1); v(1) = INIT(2); F(1) = INIT(3);
qh(1,:) = INITH(:,1).'; vh(1,:) = INITH(:,2).'; Fh(1,:) = INITH(:,3).';
m = 1400; tau = 0.1; c0 = 0.0105*m*9.81; c1 = 4; c2 = .5*rho*S*Cd;

dt = 0.01;
t = 0:dt:300;
L1 = 10;
L2 = 1000;

for i = 1:length(t)-1

    vtil = sqrt(c2)*v(i) + c1/(2*sqrt(c2));

    qdot(i) = v(i);
    vdot(i) = (1/m)*(F(i)-c0-c1*v(i)-c2*v(i)^2);
    Fdot(i) = (1/tau)*(-F(i));

 for j = 1:3
     vhtil = sqrt(c2)*vh(i,j) + c1/(2*sqrt(c2));
    qhdot(i,j) = vh(i,j) + (m/c2)*L1*(exp(c2*(q(i)-qh(i,j))/m)-1);
    vhdot(i,j) = (1/sqrt(c2))*(L2-vhtil*L1)*(exp(c2*(q(i)-qh(i,j))/m)-1)...
        + (1/m)*((exp(c2*(q(i)-qh(i,j))/m)*(Fh(i,j)-c0+c1^2/(4*c2)))...
        -vhtil^2);
    Fhdot(i,j) = (1/tau)*(-Fh(i,j));

    q(i+1) = q(i) + qdot(i)*dt;
    v(i+1) = v(i) + vdot(i)*dt;
    F(i+1) = F(i) + Fdot(i)*dt;

```

```
    qh(i+1,j) = qh(i,j) + qhdot(i,j)*dt;
    vh(i+1,j) = vh(i,j) + vhdot(i,j)*dt;
    Fh(i+1,j) = Fh(i,j) + Fhdot(i,j)*dt;

end
end

eq = [q; q; q]-qh.'; ev = [v; v; v]-vh.'; eF = [F; F; F]-Fh.';
%%

figure(1)

subplot(2,1,1)
plot(t,qh,'-','LineWidth',3)
hold on
plot(t,q,'-','LineWidth',3)
xlabel('Time (s)')
ylabel('Position (m)','FontSize',12)
legend('Estimate #1','Estimate #2','Estimate #3','True value')
set(gca,'fontsize', 14)
xlim([0 100])
subplot(2,1,2)
plot(t,eq,'-','LineWidth',3)
xlabel('Time (s)','FontSize',12)
ylabel('Position error(m)','FontSize',12)
set(gca,'fontsize', 14)
xlim([0 100])
figure(2)

subplot(2,1,1)
plot(t,vh,'-','LineWidth',3)
hold on
plot(t,v,'-','LineWidth',3)
xlabel('Time (s)','FontSize',16)
ylabel('Velocity (m/s)','FontSize',16)
legend('Estimate #1','Estimate #2','Estimate #3','True value')
set(gca,'fontsize', 14)

subplot(2,1,2)
plot(t,ev,'-','LineWidth',3)
xlabel('Time (s)','FontSize',12)
ylabel('Velocity error(m)','FontSize',12)
set(gca,'fontsize', 14)
figure(3)

subplot(2,1,1)
plot(t,Fh,'-','LineWidth',3)
hold on
plot(t,F,'-','LineWidth',3)
xlabel('Time (s)','FontSize',12)
ylabel('Thrust (N)','FontSize',12)
```

```
legend('Estimate #1','Estimate #2','Estimate #3','True value','FontSize',13)
xlim([0 1])
set(gca,'fontsize', 14)
subplot(2,1,2)
plot(t,eF,'LineWidth',3)
xlabel('Time (s)','FontSize',14)
ylabel('Thrust error(m)','FontSize',14)
xlim([0 1])
set(gca,'fontsize', 14)

figure(4)
subplot(2,1,1)
plot(t(1:end-1),vhdot,'LineWidth',3)
hold on
plot(t(1:end-1),vdot,'LineWidth',3)
xlabel('Time (s)','FontSize',12)
ylabel('Acceleration (m/s^2)','FontSize',14)
legend('Estimate #1','Estimate #2','Estimate #3','True value','FontSize',13)
xlim([0 100])
set(gca,'fontsize', 14)

subplot(2,1,2)
plot(t(1:end-1),[vdot;vdot;vdot].'-vhdot,'LineWidth',3)
xlabel('Time (s)','FontSize',12)
ylabel('Acceleration error(m/s^2)','FontSize',12)
xlim([0 100])
set(gca,'fontsize', 14)
```

Appendix C

Tracking dynamics

This appendix contains the MATLAB code which can be used for obtaining state trajectories and spacing error curves or surfaces for a two-vehicle platoon. In particular, the first code is used in Section 5.3.1, while the latter is used in Section 5.3.2.

C.1 State trajectory and estimation errors

```
    clear all; close all; clc;
Crr = 0.0105;
rho = 1.225; Cd = 0.24; S = 1.470*1.760; g =9.81;
m = 1400; tau = .1; c0 = Crr*m*g; c1 = 4; c2 = 0.5*rho*Cd*S;

INIT = [0 20 0];
INITH = [0 30 50];
INITL = [70 20 0];

q(1) = INIT(1); v(1) = INIT(2); F(1) = INIT(3);
qh(1) = INITH(1); vh(1) = INITH(2); Fh(1) = INITH(3);
ql(1) = INITL(1); vl(1) = INITL(2); Fl(1) = INITL(3);

L1 = 10; L2 = 10; dt = 0.01;
t = 0:dt:300; h = 1.75; Kp = .2; Kd = .7; L = 4;

for i = 1:length(t)-1
    %Leader Signal
    if t(i) > 10 & t(i) <= 30
            ul(i) = 2000;
            %ul(i) = 4000;
    else if t(i) > 50 & t(i) <= 70
            ul(i) = -2000;
        else
            ul(i) = 0;
            %ul(i) = 1000;
        end
    end
```

```

    vtil = sqrt(c2)*v(i) + c1/(2*sqrt(c2));
    vhtil = sqrt(c2)*vh(i) + c1/(2*sqrt(c2));
    vhtill = sqrt(c2)*vl(i) + c1/(2*sqrt(c2));

    qldot(i) = vl(i);
    vldot(i) = (1/m)*(Fl(i)-c0-c1*vl(i)-c2*vl(i)^2);
    Fldot(i) = (1/tau)*(-Fl(i)+ul(i));

    qdot(i) = v(i);
    vdot(i) = (1/m)*(F(i)-c0-c1*v(i)-c2*v(i)^2);

    qhdot(i) = vh(i) + (m/c2)*L1*(exp(c2*(q(i)-qh(i))/m)-1);
    vhdot(i) = (1/sqrt(c2))*(L2-vhtil*L1)*(exp(c2*(q(i)-qh(i))/m)-1)...
        + (1/m)*(((Fh(i)-c0+c1^2/(4*c2))*(exp(c2*(q(i)-qh(i))/m))-vhtil^2);

    e1(i) = ql(i) - q(i) - h*v(i) - L;
    e2(i) = vl(i) - v(i) - h*vdot(i);

    e1h(i) = ql(i) - qh(i) - h*vh(i) - L;
    e2h(i) = vl(i) - qhdot(i) - h*vhdot(i);

    U(i) =  Fh(i)+(m*tau/h)*(Kp*e1h(i)+Kd*e2h(i) -...
        (1-c1*h/m - 2*h*c2*vh(i)/m)*vhdot(i) + vldot(i));

    Fdot(i) = (1/tau)*(-F(i)+U(i));
    Fhdot(i) = (1/tau)*(-Fh(i)+U(i));

    q(i+1) = q(i) + qdot(i)*dt;
    v(i+1) = v(i) + vdot(i)*dt;
    F(i+1) = F(i) + Fdot(i)*dt;

    qh(i+1) = qh(i) + qhdot(i)*dt;
    vh(i+1) = vh(i) + vhdot(i)*dt;
    Fh(i+1) = Fh(i) + Fhdot(i)*dt;

    ql(i+1) = ql(i) + qldot(i)*dt;
    vl(i+1) = vl(i) + vldot(i)*dt;
    Fl(i+1) = Fl(i) + Fldot(i)*dt;
end
%%
figure(1)

subplot(2,1,1)
plot(t,q,'Linewidth',2)
hold on
plot(t,qh,'Linewidth',2)
hold on
plot(t,ql,'Linewidth',2)
xlabel('Time (s)')
ylabel('Position (m)')
```

```
legend('True value','Estimate','Leader')

subplot(2,1,2)
plot(t,q-qh,'Linewidth',2)
xlabel('Time (s)')
ylabel('Position error(m)')

figure(2)

subplot(2,1,1)
plot(t,v,'Linewidth',2)
hold on
plot(t,vh,'Linewidth',2)
set(gca,'Fontsize', 14)
xlabel('Time (s)')
ylabel('Velocity (m/s)')
legend('True value','Estimate')

subplot(2,1,2)
plot(t,v-vh,'Linewidth',2)
set(gca,'Fontsize', 14)
xlabel('Time (s)')
ylabel('Velocity error(m/s)')

figure(3)

subplot(2,1,1)
plot(t,F,'Linewidth',2)
hold on
plot(t,Fh,'Linewidth',2)
set(gca,'Fontsize', 14)
xlabel('Time (s)')
ylabel('Thrust (N)')
legend('True value','Estimate')

subplot(2,1,2)
plot(t,F-Fh,'Linewidth',2)
xlabel('Time (s)')
ylabel('Thrust error(N)')

figure(4)

subplot(2,1,1)
plot(t(1:end-1),vdot,'Linewidth',2)
hold on
plot(t(1:end-1),vhdot,'Linewidth',2)
set(gca,'Fontsize', 14)
xlabel('Time (s)')
ylabel('Acceleration (m/s^2)')
legend('True value','Estimate')
```

```
subplot(2,1,2)
plot(t(1:end-1),vdot-vhdot)
set(gca,'Fontsize', 14)
xlabel('Time (s)')
ylabel('Acceleration error(m/s^2)')

figure(5)
plot(t,[0 ul],'Linewidth',2)
hold on
plot(t,[0 U],'Linewidth',2)
legend('Leader','Follower')
xlabel('Time (s)')
ylabel('Control Input (N)')

figure(6)

subplot(2,1,1)
plot(t(1:end-1),e1,'Linewidth',2)
xlabel('Time (s)')
ylabel('Spacing error, e_{s1} (m)')

subplot(2,1,2)
plot(t(1:end-1),e2,'Linewidth',2)
xlabel('Time (s)')
ylabel('Spacing error rate, e_{s2}(m/s)')
```

C.2 Spacing error for a pool of initial estimates

```
clear all; close all; clc;
Crr = 0.0105;
rho = 1.225; Cd = 0.24; S = 1.470*1.760; nu = 0.7; g = 9.81;
m = 1400; tau = .1; c0 = Crr*m*g; c1 = 4; c2 = 0.5*rho*Cd*S;

INIT = [0 20 0];
qh0 = 0; vh0 = -50:1:50; Fh0 = -5000:100:5000; %INITH
INITL = [70 20 0];

q0 = INIT(1); v0 = INIT(2); F0 = INIT(3);
ql(1) = INITL(1); vl(1) = INITL(2); Fl(1) = INITL(3);

L1 = 10; L2 = 10; dt = 0.1;
t = 0:dt:500; h = 1.75; Kp = .2; Kd = .7; L = 4;
%%
for k = 1:length(Fh0)
for j = 1:length(vh0)
for i = 1:length(t)-1

    if i == 1
        vh(i,j,k) = vh0(j); Fh(i,j,k) = Fh0(k); qh(i,j,k) = qh0;
```

```
        q(i,j,k) = q0; v(i,j,k) = v0; F(i,j,k) = F0;
    end

    if t(i) > 10 & t(i) <= 30
            ul(i) = 4000;
    else if t(i) > 50 & t(i) <= 70
            ul(i) = -2000;
        else
            ul(i) = 1000;
        end
    end
% ul(i) = 2000;
    vtil = sqrt(c2)*v(i,j,k) + c1/(2*sqrt(c2));
    vhtil = sqrt(c2)*vh(i,j,k) + c1/(2*sqrt(c2));
    vhtill = sqrt(c2)*vl(i) + c1/(2*sqrt(c2));

    qldot(i) = vl(i);
    vldot(i) = (1/m)*(Fl(i)-c0-c1*vl(i)-c2*vl(i)^2);
    Fldot(i) = (1/tau)*(-Fl(i)+ul(i));

    qdot(i,j,k) = v(i,j,k);
    vdot(i,j,k) = (1/m)*(F(i,j,k)-c0-c1*v(i,j,k)-c2*v(i,j,k)^2);

    qhdot(i,j,k) = vh(i,j,k) + (m/c2)*L1*(exp(c2*(q(i,j,k)-qh(i,j,k))/m)-1);
    vhdot(i,j,k) = (1/sqrt(c2))*(L2-vhtil*L1)*(exp(c2*(q(i,j,k)-qh(i,j,k))/m
        )-1)...
        + (1/m)*(((Fh(i,j,k)-c0+c1^2/(4*c2))*(exp(c2*(q(i,j,k)-qh(i,j,k))/m)
            ))-vhtil^2);

    e1(i,j,k) = ql(i) - q(i,j,k) - h*v(i,j,k) - L;
    e2(i,j,k) = vl(i) - v(i,j,k) - h*vdot(i,j,k);

    e1h(i,j,k) = ql(i) - qh(i,j,k) - h*vh(i,j,k) - L;
    e2h(i,j,k) = vl(i) - qhdot(i,j,k) - h*vhdot(i,j,k);

    U(i,j,k) =  Fh(i,j,k)+(m*tau/h)*(Kp*e1h(i,j,k)+Kd*e2h(i,j,k) -...
        (1-c1*h/m - 2*h*c2*vh(i,j,k)/m)*vhdot(i,j,k) + vldot(i));

    Fdot(i,j,k) = (1/tau)*(-F(i,j,k)+U(i,j,k));
    Fhdot(i,j,k) = (1/tau)*(-Fh(i,j,k)+U(i,j,k));

    q(i+1,j,k) = q(i,j,k) + qdot(i,j,k)*dt;
    v(i+1,j,k) = v(i,j,k) + vdot(i,j,k)*dt;
    F(i+1,j,k) = F(i,j,k) + Fdot(i,j,k)*dt;

    qh(i+1,j,k) = qh(i,j,k) + qhdot(i,j,k)*dt;
    vh(i+1,j,k) = vh(i,j,k) + vhdot(i,j,k)*dt;
    Fh(i+1,j,k) = Fh(i,j,k) + Fhdot(i,j,k)*dt;

    ql(i+1) = ql(i) + qldot(i)*dt;
    vl(i+1) = vl(i) + vldot(i)*dt;
```

```
    Fl(i+1) = Fl(i) + Fldot(i)*dt;
end
end
end

%%

E1(:,:) = e1(3001,:,:); %First coordinate of e1 determines k-th time

figure(1)

[Vh0,FFh0] = meshgrid(vh0,Fh0);
Z1 = griddata(vh0,Fh0,E1,Vh0,FFh0);
mesh(Vh0,FFh0,Z1)
set(gca,'FontSize',20)
xlabel('$$\hat{v}(0)$$ (m/s)','Interpreter','Latex', 'FontSize', 22)
ylabel('$$\hat{F}(0)$$ (N)','Interpreter','Latex', 'FontSize', 22)
zlabel('$$e_{s1}$$ (m)','Interpreter','Latex', 'FontSize', 22)

E2(:,:) = e2(end,:,:); %First coordinate of e2 determines k-th time

figure(2)

[Vh0,FFh0] = meshgrid(vh0,Fh0);
Z2 = griddata(vh0,Fh0,E2,Vh0,FFh0);
mesh(Vh0,FFh0,Z2)
set(gca,'FontSize',20)
xlabel('$$\hat{v}(0)$$ (m/s)','Interpreter','Latex', 'FontSize', 22)
ylabel('$$\hat{F}(0)$$ (N)','Interpreter','Latex', 'FontSize', 22)
zlabel('$$e_{s2}$$ (m/s)','Interpreter','Latex', 'FontSize', 22)
%zlim([-.5 1])
```

Appendix D

String stability

This appendix contains the MATLAB codes used for string stability analysis in Section 5.3.3.

D.1 Step 1: Observer tuning

```
    clear all; close all; clc;

%Initial conditions [q0 v0 F0]
INIT = [0 20 0]; %Vehicle i
INITH = [0 10 500]; %Vehicle i estimate
INITL = [70 20 0]; %Vehicle i-1

%Constants
rho = 1.225; Cd = 0.24; S = 1.470*1.760; g =9.81;

m = 1400; tau = 0.1; c0 = 0.0105*m*g; c1 = 4; c2 = .5*rho*Cd*S;
L1 = linspace(1,20,20); L2 = linspace(1,4001,20); dt = 0.1;
t = 0:dt:100; h = 3; Kp = .2; Kd = .7; L = 4; w = .01;

q(1,1:length(L1),1:length(L2)) = INIT(1); v(1,1:length(L1),1:length(L2)) =
    INIT(2); F(1,1:length(L1),1:length(L2)) = INIT(3);
qh(1,1:length(L1),1:length(L2)) = INITH(1); vh(1,1:length(L1),1:length(L2))
    = INITH(2); Fh(1,1:length(L1),1:length(L2)) = INITH(3);
ql(1,1:length(L1),1:length(L2)) = INITL(1); vl(1,1:length(L1),1:length(L2))
    = INITL(2); Fl(1,1:length(L1),1:length(L2)) = INITL(3);
%%
for k = 1:length(L2)
for j = 1:length(L1)
for i = 1:length(t)-1

    vtil = sqrt(c2)*v(i,j,k) + c1/(2*sqrt(c2));
    vhtil = sqrt(c2)*vh(i,j,k) + c1/(2*sqrt(c2));
    vhtill = sqrt(c2)*vl(i,j,k) + c1/(2*sqrt(c2));

```

```
    if vhtil == NaN | vtil == NaN
        break
    end

    qldot(i,j,k) = vl(i,j,k);
    vldot(i,j,k) = cos(w*t(i));

    qdot(i,j,k) = v(i,j,k);
    vdot(i,j,k) = (1/m)*(F(i,j,k)-c0-c1*v(i,j,k)-c2*v(i,j,k)^2);

    qhdot(i,j,k) = vh(i,j,k) + (m/c2)*L1(j)*(exp(c2*(q(i,j,k)-qh(i,j,k))/m)
        -1);
    vhdot(i,j,k) = (1/sqrt(c2))*(L2(k)-vhtil*L1(j))*(exp(c2*(q(i,j,k)-qh(i,j
        ,k))/m)-1)...
        + (1/m)*(((Fh(i,j,k)-c0+c1^2/(4*c2))*(exp(c2*(q(i,j,k)-qh(i,j,k))/m)
            ))-vhtil^2);

    e1(i,j,k) = ql(i,j,k) - q(i,j,k) - h*v(i,j,k) - L;
    e2(i,j,k) = vl(i,j,k) - v(i,j,k) - h*vdot(i,j,k);

    e1h(i,j,k) = ql(i,j,k) - qh(i,j,k) - h*vh(i,j,k) - L;
    e2h(i,j,k) = vl(i,j,k) - qhdot(i,j,k) - h*vhdot(i,j,k);

    U(i,j,k) =   Fh(i,j,k)+(m*tau/h)*(Kp*e1h(i,j,k)+Kd*e2h(i,j,k) -...
        (1-c1*h/m - 2*h*c2*vh(i,j,k)/m)*vhdot(i,j,k) + vldot(i,j,k));

    Fdot(i,j,k) = (1/tau)*(-F(i,j,k)+U(i,j,k));
    Fhdot(i,j,k) = (1/tau)*(-Fh(i,j,k)+U(i,j,k));

    q(i+1,j,k) = q(i,j,k) + qdot(i,j,k)*dt;
    v(i+1,j,k) = v(i,j,k) + vdot(i,j,k)*dt;
    F(i+1,j,k) = F(i,j,k) + Fdot(i,j,k)*dt;

    qh(i+1,j,k) = qh(i,j,k) + qhdot(i,j,k)*dt;
    vh(i+1,j,k) = vh(i,j,k) + vhdot(i,j,k)*dt;
    Fh(i+1,j,k) = Fh(i,j,k) + Fhdot(i,j,k)*dt;

    ql(i+1,j,k) = ql(i,j,k) + qldot(i,j,k)*dt;
    vl(i+1,j,k) = vl(i,j,k) + vldot(i,j,k)*dt;

end
end
end

%%
for k = 1:length(L2)
for j = 1:length(L1)

l2norm(j,k) = norm(vdot(:,j,k),2)/norm(vldot(:,j,k),2);

end
```

```
end

[LL1,LL2] = meshgrid(L1,L2);
Z = griddata(L1,L2,l2norm,LL1,LL2);
mesh(LL1,LL2,Z)
set(gca,'fontsize',14)
xlabel('L_1')
ylabel('L_2')
zlabel('|\Gamma|_2')
%ylim([0 6001])
%zlim([0 2])
```

D.2 Step 2: Determining minimum time headway

```
    clear all; close all; clc;

%Initial conditions [q0 v0 F0]
INIT = [0 20 0]; %Vehicle i
INITH = [0 10 500]; %Vehicle i estimate
INITL = [70 20 0]; %Vehicle i-1

%Constants
rho = 1.225; Cd = 0.24; S = 1.470*1.760; g =9.81;

m = 1400; tau = 0.1; c0 = 0.0105*m*g; c1 = 4; c2 = .5*rho*Cd*S;
L1 = 10; L2 = 3000; dt = 0.01;
t = 0:dt:100; h = 2.44; Kp = .2; Kd = .7; L = 4;

w = 0:0.01:2; %frequency range

q(1,1:length(h),1:length(w)) = INIT(1); ...
    v(1,1:length(h),1:length(w)) = INIT(2); ...
    F(1,1:length(h),1:length(w)) = INIT(3);
qh(1,1:length(h),1:length(w)) = INITH(1); ...
    vh(1,1:length(h),1:length(w)) = INITH(2); ...
    Fh(1,1:length(h),1:length(w)) = INITH(3);
ql(1,1:length(h),1:length(w)) = INITL(1); ...
    vl(1,1:length(h),1:length(w)) = INITL(2); ...
    Fl(1,1:length(h),1:length(w)) = INITL(3);

for k = 1:length(w)
for j = 1:length(h)
for i = 1:length(t)-1

    vtil = sqrt(c2)*v(i,j,k) + c1/(2*sqrt(c2));
    vhtil = sqrt(c2)*vh(i,j,k) + c1/(2*sqrt(c2));
    vhtill = sqrt(c2)*vl(i,j,k) + c1/(2*sqrt(c2));

    if vhtil == NaN | vtil == NaN
```

```
            break
        end

        qldot(i,k) = vl(i,k);
        vldot(i,k) = cos(w(k)*t(i));

        qdot(i,j,k) = v(i,j,k);
        vdot(i,j,k) = (1/m)*(F(i,j,k)-c0-c1*v(i,j,k)-c2*v(i,j,k)^2);

        qhdot(i,j,k) = vh(i,j,k) +(m/c2)*L1*(exp(c2*(q(i,j,k)-qh(i,j,k))/m)-1);
        vhdot(i,j,k) = (1/sqrt(c2))*(L2-vhtil*L1)*(exp(c2*(q(i,j,k) -...
            qh(i,j,k))/m)-1)+ (1/m)*(((Fh(i,j,k)-c0+c1^2/(4*c2))*...
            (exp(c2*(q(i,j,k)-qh(i,j,k))/m))-vhtil^2);

        e1(i,j,k) = ql(i,j,k) - q(i,j,k) - h(j)*v(i,j,k) - L;
        e2(i,j,k) = vl(i,j,k) - v(i,j,k) - h(j)*vdot(i,j,k);

        e1h(i,j,k) = ql(i,j,k) - qh(i,j,k) - h(j)*vh(i,j,k) - L;
        e2h(i,j,k) = vl(i,j,k) - qhdot(i,j,k) - h(j)*vhdot(i,j,k);

        U(i,j,k) =   Fh(i,j,k)+(m*tau/h(j))*(Kp*e1h(i,j,k)+Kd*e2h(i,j,k) -...
            (1-c1*h(j)/m - 2*h(j)*c2*vh(i,j,k)/m)*vhdot(i,j,k) + vldot(i,k));

        Fdot(i,j,k) = (1/tau)*(-F(i,j,k)+U(i,j,k));
        Fhdot(i,j,k) = (1/tau)*(-Fh(i,j,k)+U(i,j,k));

        q(i+1,j,k) = q(i,j,k) + qdot(i,j,k)*dt;
        v(i+1,j,k) = v(i,j,k) + vdot(i,j,k)*dt;
        F(i+1,j,k) = F(i,j,k) + Fdot(i,j,k)*dt;

        qh(i+1,j,k) = qh(i,j,k) + qhdot(i,j,k)*dt;
        vh(i+1,j,k) = vh(i,j,k) + vhdot(i,j,k)*dt;
        Fh(i+1,j,k) = Fh(i,j,k) + Fhdot(i,j,k)*dt;

        ql(i+1,k) = ql(i,k) + qldot(i,k)*dt;
        vl(i+1,k) = vl(i,k) + vldot(i,k)*dt;
        %Fl(i+1,j,k) = Fl(i,j,k) + Fldot(i,j,k)*dt;


end
end
end


for k = 1:length(w)
        Vlw(k) = norm(vldot(:,k),2);
for j = 1:length(h)
        Vw(j,k) = norm(vdot(:,j,k),2);
l2norm(j,k) = Vw(j,k)/Vlw(k);

end
```

```
end

Lmax = max(l2norm) %|Gamma|_inf —— must not exceed 1

%% Plots
figure(1)
plot(w(2:end),l2norm(2:end),'LineWidth',2)
xlabel('Frequency (Hz)')
ylabel('|a_i|_2 / |a_{i-1}|_2')

figure(2)
plot(w,Vlw,'LineWidth',2)
hold on
plot(w,Vw(1,:),'LineWidth',2)
xlabel('Frequency (Hz)')
ylabel('|a|_2 (m/s^2)')
legend('Leader','Follower')
```