SECONDA UNIVERSITÀ DEGLI STUDI DI NAPOLI

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Summer School SIDRA 2013

Design of an Adaptive Cruise Control System based on the Constant Time-Gap Policy

Advisor
Prof. Ing. Ciro Natale

Ph.D Student
Dott. Pasquale Cirillo

Contents

1 Introduction **1**
1.1 Contents 2

2 Adaptive Cruise Control System **3**
2.1 ACC Architecture 4
2.1.1 Individual vehicle stability 5
2.1.2 String stability 5

3 String Stability **7**
3.1 Autonomous Control 8
3.1.1 Autonomous Control with Constant Time-Gap Spacing Policy 8

4 Simulation **11**

Bibliography **20**

List of Figures

2.1 Adaptive Cruise Control scheme. 4
2.2 Adaptive Cruise Control architecture. 5
2.3 String of ACC vehicles. 6

4.1 Pole and zero map of $\hat{H}(s)$. 12
4.2 Vehicle model - Simulink block diagram. 13
4.3 ACC upper controller - Simulink block diagram. 13
4.4 Complete system - Simulink block diagram. 14
4.5 Vehicle response to a sinusoidal input. 15
4.6 Platoon vehicles positions. 16
4.7 Platoon vehicles velocities. 17
4.8 Vehicles spacing error. 17
4.9 Pole and zero map of $\hat{H}(s)$ - non string stable controller. 18
4.10 Platoon vehicles positions - non string stable controller. 18
4.11 Platoon vehicles velocities - non string stable controller. 19
4.12 Vehicles spacing error - non string stable controller. 19

Chapter 1

Introduction

In the last decades, a variety of driver assistance systems are being developed by automotive industries in order to reduce driver burden and, first of all, to reduce highway accidents. Examples of these assistance systems are:

- collision avoidance systems which automatically detect slower moving preceding vehicles and provide warning and brake assist to the driver
- adaptive cruise control (ACC) systems which are enhanced cruise control systems and enable preceding vehicles to be followed automatically at a safe distance
- driver condition monitoring systems which detect and provide warning for driver drowsiness, as well as for obstacles and pedestrians
- lane departure warning systems
- lane keeping systems which automate steering on straight roads

Many of these systems help to reduce the driver burden, but collision avoidance and adaptive cruise control systems can also help to reduce the resultant traffic congestion that tend to cause accidents.

The aim of this work is to design an ACC system using the Constant Time-Gap Spacing Policy in order to guarantee the string stability property in a platoon of ten vehicles.

1.1 Contents

In Chapter 2 an introduction to the Adaptive Cruise Control systems is presented focusing on the ACC system Architecture and on both the individual vehicle stability and string stability properties.
In Chapter 3 the string stability property is illustrate in details. In particular, the string stability property is demonstrated considering the Constant Time-Gap Spacing Policy.
In Chapter 4 an example of ACC system design is presented.

Chapter 2

Adaptive Cruise Control System

An adaptive cruise control (ACC) is an extension of the standard cruise control system and provides assistance to the driver in the task of longitudinal control of the vehicle during motorway driving. The system controls the accelerator, engine powertrain and vehicle brakes to maintain a desired spacing/time-gap to the vehicle ahead. It has been designed by vehicle manufacturers in order to reduce the resultant traffic congestion that tend to cause accidents and to increase the quality of the driving experience providing considerable reductions in the variation of acceleration compared to manual driving. This indicates a potential comfort gain for the driver and environmental benefits as explained in [1]. Moreover, the influence of the ACC systems on highway traffic is being studied by several research groups with the objective of designing new systems to promote smoother and higher traffic flow [2] [3] [4].

In particular, in the absence of preceding vehicles, the ACC vehicle travels at a user-set speed, much like a vehicle with a standard cruise control system. However, if a preceding vehicle is detected on the highway by the vehicle's radar, the ACC system determines whether or not the vehicle can continue to travel safely at the desired speed. If the preceding vehicle is too close or traveling too slowly, then the ACC system switches from speed control to spacing control. In spacing control, the ACC vehicle controls both the throttle and brakes so as to maintain a desired spacing from the preceding vehicle. However, the majority of ACC systems are designed only to operate at highway speeds and will often have a minimum velocity of around 30 km/h [5].

From the discussion above, it is clear that an ACC system will have two modes of steady state operation (see Fig. 2.1):

- speed control

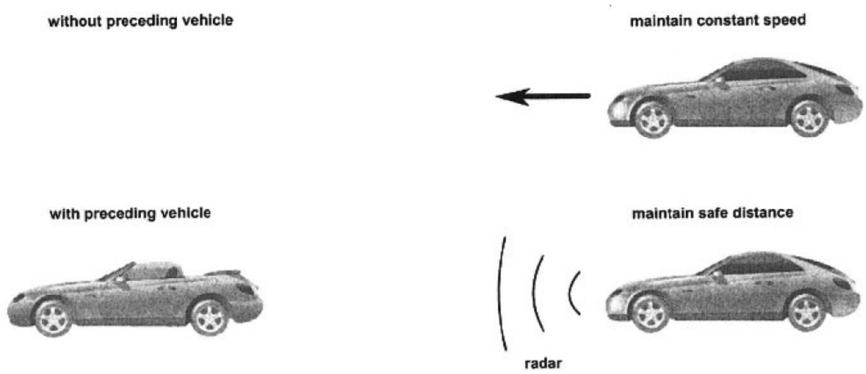

Figure 2.1: Adaptive Cruise Control scheme.

- vehicle following (i.e. spacing control)

In addition, the controller can perform a transitional maneuver to switch between the two modes illustrated above, including:

- ensuring smooth transition from speed control to vehicle following and vice-versa

- determining transition trajectories to ensure the vehicle reaches its desired steady state spacing or speed each time a new preceding vehicle is encountered, the current preceding vehicle makes an exit or a lane change, etc.

In the next chapter, the problem of vehicle following is presented.

2.1 ACC Architecture

The control architecture for an ACC system is typically designed to be hierarchical. Two levels are considered: an upper level controller determines the desired acceleration for each vehicle; a lower level controller determines the throttle and/or brake commands required to track the desired acceleration (see Fig. 2.2).

The upper level is designed in order to meet two performance specifications: the first specification is the individual stability of the vehicle, the second one is to guarantee the “string stability” in a platoon of vehicles. Individual vehicle stability and string stability definitions are illustrate in [6] and reported below.

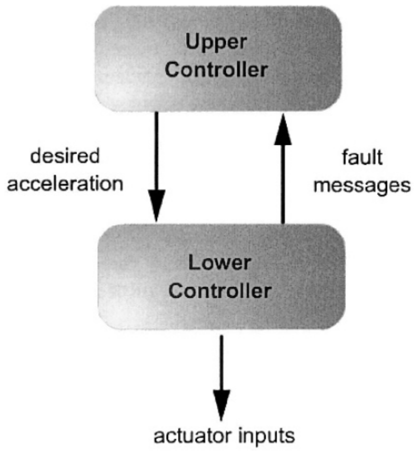

Figure 2.2: Adaptive Cruise Control architecture.

2.1.1 Individual vehicle stability

The vehicle following control law is said to provide individual vehicle stability if the spacing error of the ACC vehicle converges to zero when the preceding vehicle is operating at constant speed. If the preceding vehicle is accelerating or decelerating, then the spacing error is expected to be non-zero. Spacing error in this definition refers to the difference between the actual spacing from the preceding vehicle and the desired inter-vehicle spacing. In particular, considering a string of vehicles as shown in Fig. 2.3, where x_i is the position of the i-th vehicle measured from an inertial reference, the spacing error is defined as:

$$\delta_i = x_i - x_{i-1} + L_{des} \tag{2.1}$$

where L_{des} is the desired spacing and includes the preceding vehicle length l_{i-1}. Thus, the individual vehicle stability is provided if the following conditions are satisfied:

$$\ddot{x}_{i-1} \rightarrow 0 \Rightarrow \delta_i \rightarrow 0 \tag{2.2}$$

2.1.2 String stability

Since the spacing error is expected to be non-zero during acceleration/deceleration of the preceding vehicle, it is important to describe how the spacing error would propagate from vehicle to vehicle in a string of ACC vehicles that use the same spacing policy and control law. The string stability of a string of ACC vehicles refers to

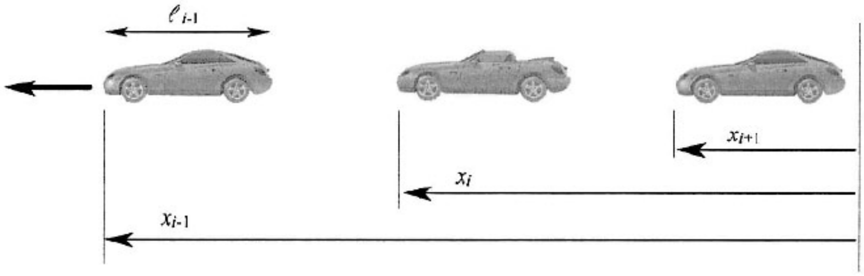

Figure 2.3: String of ACC vehicles.

a property in which spacing errors are guaranteed not to amplify as they propagate towards the tail of the string [7] [8].

Chapter 3

String Stability

As far as the upper level controller is concerned, the plant model used for control design is:

$$\ddot{x}_i = u \tag{3.1}$$

So, the acceleration of the car is assumed to be the control input. However, due to the finite bandwidth associated with the lower level controller, each car is actually expected to track its desired acceleration imperfectly. Therefore, a first order lag in the lower level controller performance is introduced:

$$\ddot{x}_i = \frac{1}{\tau s + 1}\ddot{x}_{i\_des} = \frac{1}{\tau s + 1}u_i \tag{3.2}$$

The mathematical definition of string stability is introduced below.
Let δ_i and δ_{i-1} be the spacing error of consecutive ACC vehicles in a string and let $\hat{H}(s)$ be the transfer function relating the spacing errors of consecutive vehicles defined as shown in equation (3.3).

$$\hat{H}(s) = \frac{\delta_i}{\delta_{i-1}} \tag{3.3}$$

The system is string stable if the conditions 3.4 and 3.5 are satisfied, where $h(t)$ is the impulse response function corresponding to $\hat{H}(s)$.

$$\left\|\hat{H}(s)\right\|_\infty \leq 1 \tag{3.4}$$

$$h(t) > 0, \forall\ t \geq 0 \tag{3.5}$$

Equation (3.4) ensures that $\|\delta_i\|_2 \leq \|\delta_{i-1}\|_2$, which means that the energy in the spac-

ing error signal decreases as the spacing error propagates towards the tail of the string. Moreover, equation (3.5) ensures that the steady state spacing errors of the vehicles in the string have the same sign.
Note that positive spacing error implies that a vehicle is closer than desired while a negative spacing error implies that it is further apart than desired. So, if the steady state value of δ_i is positive while the steady state value of δ_{i-1} is negative, then this might be dangerous even though in terms of magnitude δ_i might be smaller than δ_{i-1}. The condition that the impulse response have to be positive ensures that steady state values of δ_i and δ_{i-1} have the same sign.

Moreover, it should be noted that equations (3.4) and (3.5) imply equation (3.6) (for more details see Section 7.5 in [6]).

$$||\delta_i||_\infty \leq ||\delta_{i-1}||_\infty \tag{3.6}$$

3.1 Autonomous Control

An ACC system is an autonomous controller because only utilizes on board sensors like radar and does not depend on inter-vehicle communication or any other form of cooperation from other vehicles on the highway. This implies that the only variables available as feedback measurements for the upper controller are inter-vehicle spacing, relative velocity and the ACC vehicle's own velocity. For this reason, a constant spacing policy is unsuitable for autonomous control applications as shown in Section 6.5 of [6]. Instead, a constant time-gap policy can be considered.

3.1.1 Autonomous Control with Constant Time-Gap Spacing Policy

The constant time-gap (CTG) spacing policy is the most common spacing policy used in ACC systems by researchers as well as automotive manufacturers and it can ensure both individual vehicle stability and string stability. In the CTG spacing policy, the desired inter-vehicle spacing is not constant but varies linearly with velocity as in equation (3.7).

$$L_{des} = l_{i-1} + h\dot{x}_i \tag{3.7}$$

The constant parameter h is referred to as the time-gap. The spacing error varies with the velocity and it is defined as in equation (3.8) with $\epsilon_i = x_i - x_{i-1} + l_{i-1}$.

$$\delta_i = \epsilon_i + h\dot{x}_i \tag{3.8}$$

The corresponding longitudinal controller associated with the above spacing policy is introduced by Iannou and Chien [9] and can be represented as in equation (3.9).

$$\ddot{x}_{i\_des} = -\frac{1}{h}\left(\dot{\epsilon}_i + \lambda\delta_i\right) \tag{3.9}$$

It can be easily shown that the above control law provides individual vehicle stability [9] if $\ddot{x}_{i-1} \rightarrow 0$, then $\delta_i \rightarrow 0$. It has also been shown that the above controller guarantees string stability [7] for

$$h \geq 2\tau \tag{3.10}$$

where τ is the lumped lag associated with the actuators, engine and driveline described as in equation (3.2). Hence, this spacing policy (and controller) satisfies the required criterion in equation (3.10) as long as a sufficiently large headway time h is maintained.

Moreover, considering the control law in equation (3.9), it can be shown that the spacing errors of successive vehicle δ_i and δ_{i-1} are independent of each other. In fact, differentiating the equation (3.8), we obtain:

$$\dot{\delta}_i = \dot{\epsilon}_i + h\ddot{x}_i \tag{3.11}$$

Substituting for $\ddot{x}_i$ from equation (3.9) into equation (3.11) and assuming $\ddot{x}_i = \ddot{x}_{i\_des}$, the error dynamics for δ_i are obtained as

$$\dot{\delta}_i = -\lambda\delta_i \tag{3.12}$$

Thus, δ_i is independent of δ_{i-1} and it is expected to converge to zero as long as $\lambda > 0$. However, the result is true if any desired acceleration can be instantaneously obtained by the vehicle (i.e. if the time constant τ associated with the lower level controller performance is assumed zero).

The string stability of the system can be analyzed by looking at the transfer function in equation (3.13) and checking if its magnitude is always less than 1.

$$\frac{\delta_i}{\delta_{i-1}} = \frac{s + \lambda}{h\tau s^3 + hs^2 + (1 + \lambda h)s + \lambda} \tag{3.13}$$

Further, if equation (3.10) is satisfied, then it is guaranteed that a value of λ such that $\left\|\hat{H}(s)\right\|_\infty \leq 1$ exists. Thus the condition expressed in equation (3.10) is both necessary and sufficient. For the string stability details of the CTG spacing policy refer to Section 6.6.1 of [6].

Chapter 4

Simulation

In this chapter, an ACC system design will be shown considering the string stability problem. In particular, in order to guarantee the stability of a string of ten vehicles, the CTG spacing policy (illustrated in Section 3.1.1) is considered.

Now, consider the vehicle model reported in equation (4.1) and the control input shown in equation (3.9).

$$H(s) = \frac{1}{s^2(1 + 2s)} \tag{4.1}$$

The string stability can be achieved choosing h and λ expressed as in equation (3.13) in order to satisfy the conditions shown in equations (3.4) and (3.10). However, there are no results available that provide a direct design procedure for ensuring that the impulse response $h(t)$ is non-negative. The results in [10] provide indirect design tips for the same. Two necessary conditions that must be satisfied by the transfer function $\hat{H}(s)$ in order for the impulse response to be non-negative are:

- The dominant poles of the system should not be a complex conjugate pair
- There should not be any zeros of the system that are completely to the right of all poles of the closed-loop system

So, given that the lumped lag of the system (which is the time constant of the lag in tracking any desired acceleration command and that take into account the accumulating brake or engine actuation lags and the sensor signal processing lags) in equation (4.1) is $\tau = 2$, the following parameters can be considered:

$$\begin{aligned} h &= 5 \\ \lambda &= 3 \end{aligned} \tag{4.2}$$

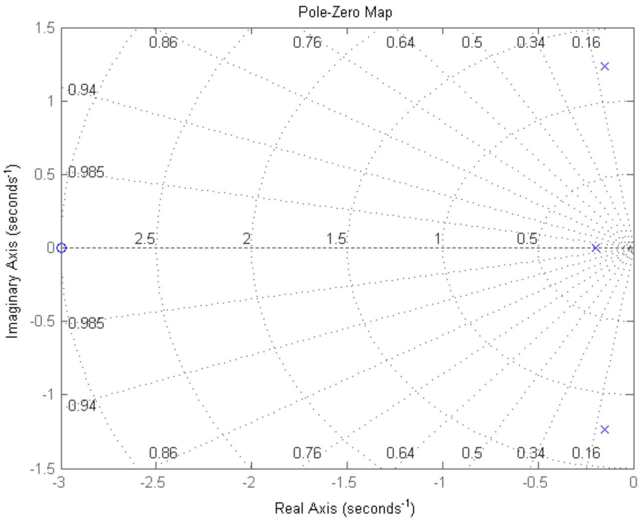

Figure 4.1: Pole and zero map of $\hat{H}(s)$.

This choice allows to obtain a $\left\|\hat{H}(s)\right\|_{\infty} = 1$. Moreover, the two conditions for the non-negativity of $h(t)$ are satisfied as shown in Figure 4.1. The poles and zeros of the transfer function $\hat{H}(s)$ are reported below:

```
Poles =
  -0.1526 + 1.2318i
  -0.1526 - 1.2318i
  -0.1947


Zeros =
    -3
```

The simulations are carried out in Matlab/Simulink. The vehicle and the ACC upper controller are implemented as Simulink blocks as reported in Figure 4.2 and Figure 4.3 respectively. Figure 4.4 shows, instead, the complete Simulink block diagram.

With reference to the complete system scheme shown in Figure 4.4, in order to simulate a periodical acceleration/deceleration of $0.3g$ to the leader vehicle, a sinusoidal signal of $0.6g$ of amplitude, zero bias and frequency of 0.5 Hz is imposed as its desired acceleration. Due to the model lag $\tau = 2$, the vehicle response to the input described above is as in Figure 4.5.

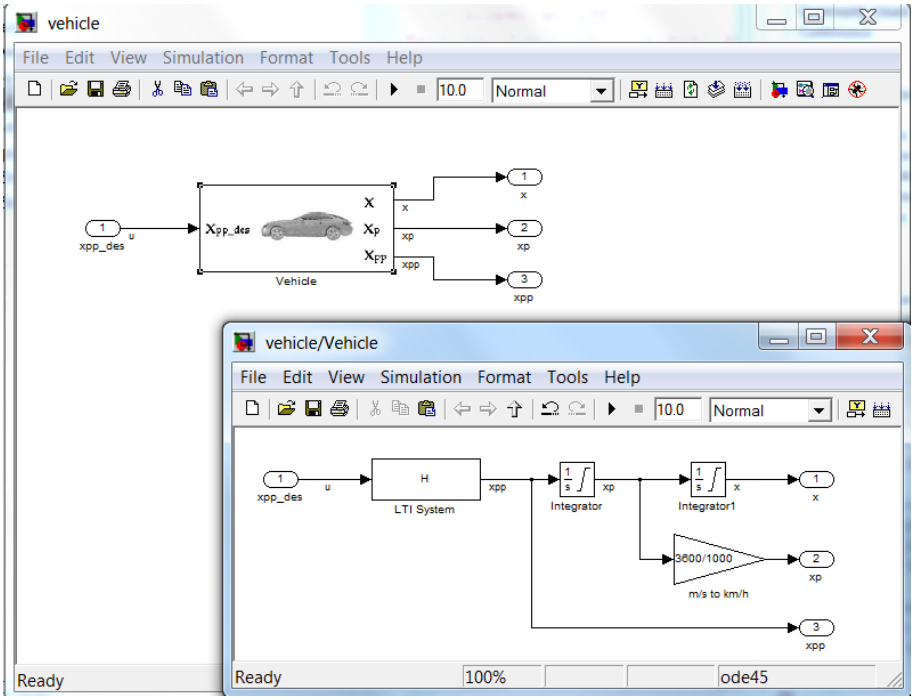

Figure 4.2: Vehicle model - Simulink block diagram.

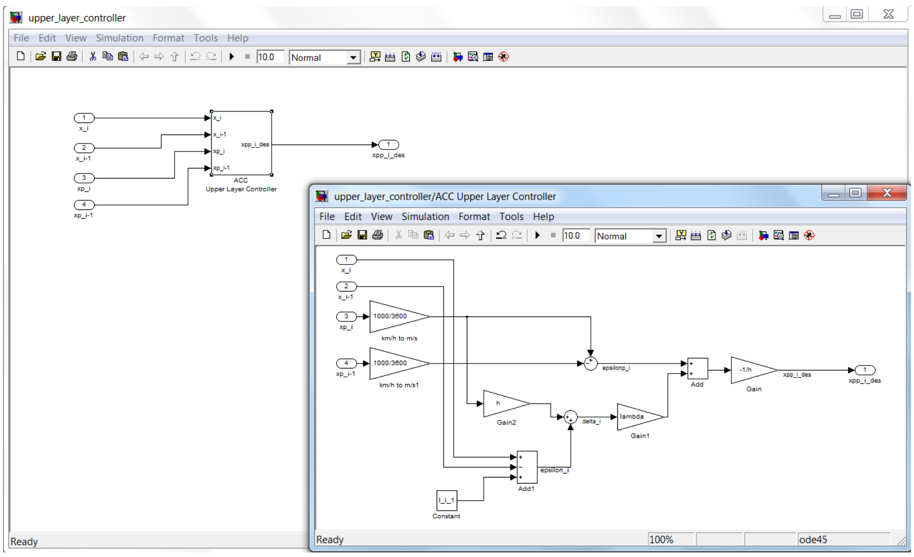

Figure 4.3: ACC upper controller - Simulink block diagram.

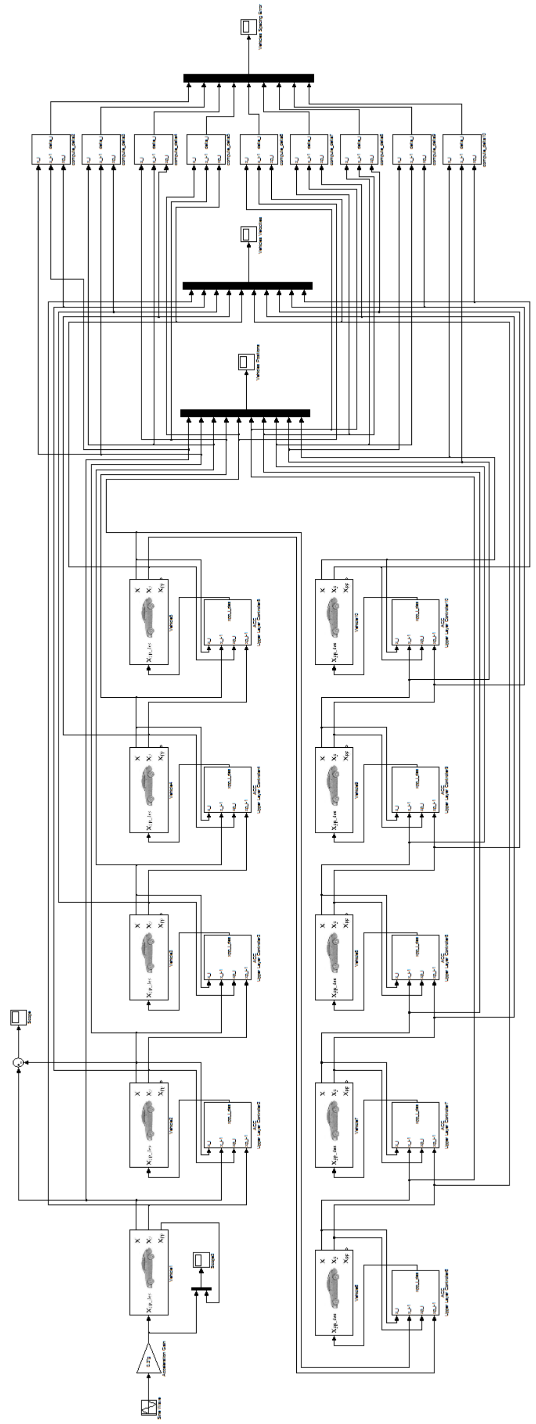

Figure 4.4: Complete system - Simulink block diagram.

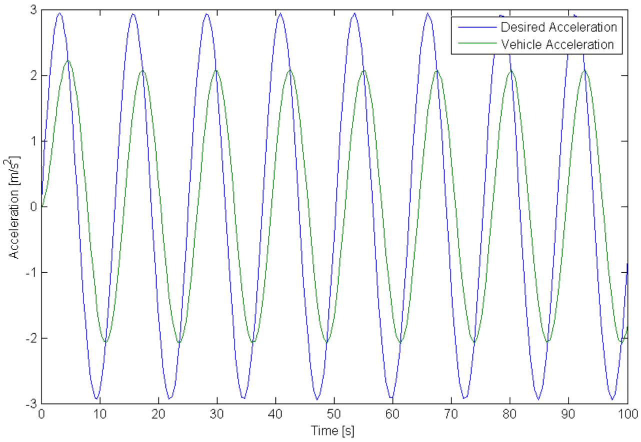

Figure 4.5: Vehicle response to a sinusoidal input.

An analysis of the designed ACC system is reported below. Considering for the i-th vehicle a length $l_i = 3\,\text{m}$, an initial velocity $\dot{x}_i = 40\,\text{km/h}$, a maximum and minimum velocity $\dot{x}_{i_{MAX}} = 130\,\text{km/h}$ and $\dot{x}_{i_{min}} = 0\,\text{km/h}$ respectively and an initial position $x_{i_0} = 100 - 10 \cdot i\,\text{m}$ (such that $x_{1_0} = 90\,\text{m}$, $x_{2_0} = 80\,\text{m}$, ..., $x_{10_0} = 0\,\text{m}$), the position and velocity of the vehicles in the platoon are as reported in Figure 4.6 and Figure 4.7 respectively. The string stability can be verified showing the vehicles spacing error δ_i as reported in Figure 4.8: in fact, the spacing error converges moving along the tail of the platoon.

For completeness, an example of non string stable controller is shown. Consider the following parameters:

$$\begin{aligned} h &= 2 \\ \lambda &= 3 \end{aligned} \tag{4.3}$$

This value of time-gap h does not satisfies the condition in equation (3.10) and, thus, the string stability property is not insured. In fact, this choice allow to obtain a $\left\|\hat{H}(s)\right\|_\infty = 7.0015$, opposing the condition in equation (3.4). The pole-zero map of the resultant $\hat{H}(s)$ is reported in Figure 4.9 and its poles and zeros are reported below (note that the necessary conditions for the non-negativity of the $h(t)$ is, anyway, satisfied):

```
Poles =
  -0.0322 + 1.3118i
```

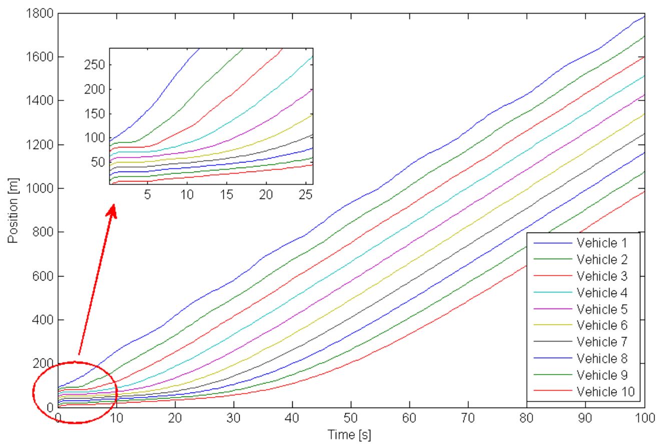

Figure 4.6: Platoon vehicles positions.

```
  -0.0322 - 1.3118i
  -0.4356

Zeros =
    -3
```

The non string stability of the considered controller can be analyzed showing the positions, velocities and, specially, the spacing errors δ_i of the vehicles in the platoon, reported in Figure 4.10, Figure 4.11 and Figure 4.12 respectively. Note that the spacing errors do not converge to a steady state, so the system is not string stable. Moreover, the spacing error δ_i changes sign. This rudely means that, during the advancement of the vehicles, the vehicle i-th collides with the vehicle i-1-th.

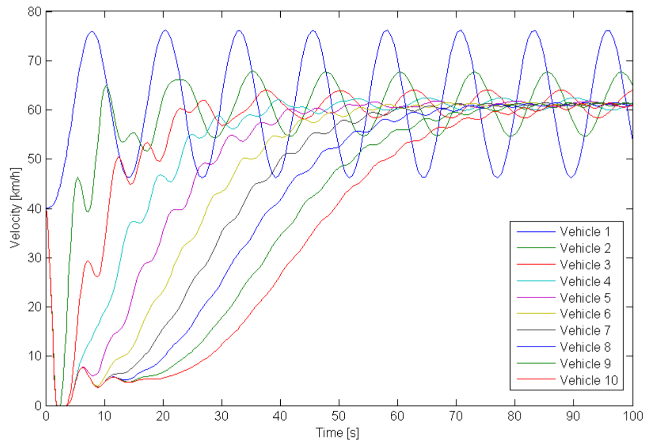

Figure 4.7: Platoon vehicles velocities.

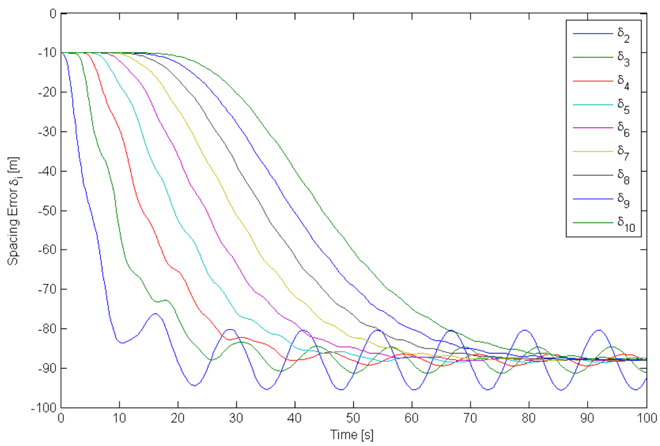

Figure 4.8: Vehicles spacing error.

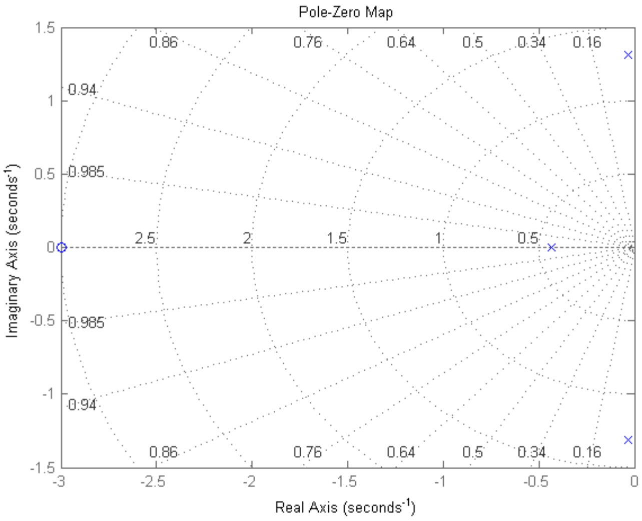

Figure 4.9: Pole and zero map of $\hat{H}(s)$ - non string stable controller.

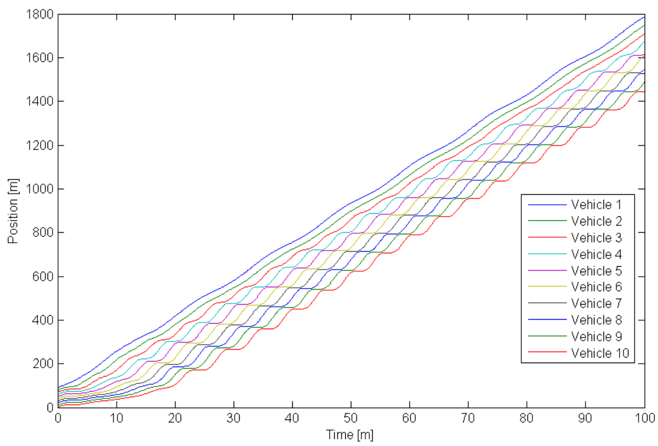

Figure 4.10: Platoon vehicles positions - non string stable controller.

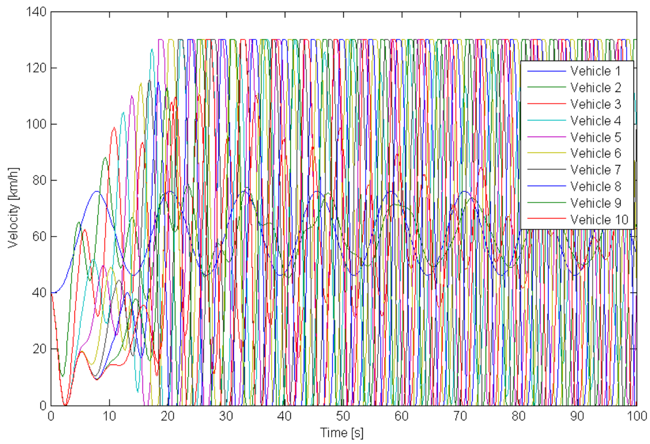

Figure 4.11: Platoon vehicles velocities - non string stable controller.

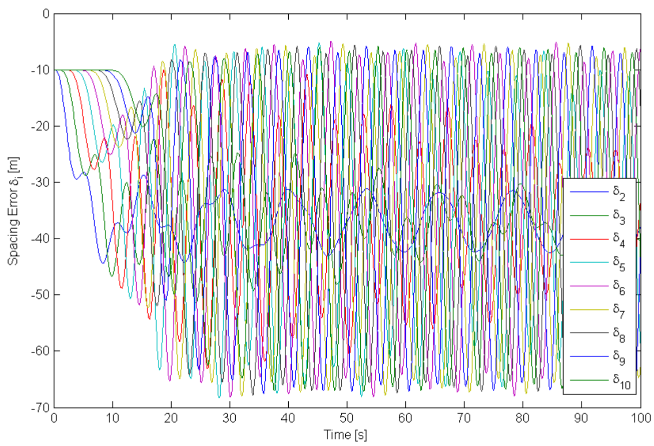

Figure 4.12: Vehicles spacing error - non string stable controller.

Bibliography

[1] Greg Marsden, Mike McDonald, and Mark Brackstone. Towards an understanding of adaptive cruise control. In *Transportation Research Part C 9, 33-51*, 2001.

[2] C. Liang and H. Peng. Design and simulations of a traffic-friendly adaptive cruise control algorithm. In *International Mechanical Engineering Congress and Exposition*, 1998.

[3] D. Swaroop and K. R. Rajagopal. Intelligent cruise control systems and traffic flow stability. In *Transportation Research Part C: Emerging Technologies, Vol. 7, No. 6, pp. 329-352*, 1999.

[4] D. Swaroop and R. Huandra. Intelligent cruise control system design based on a traffic flow specification. In *Vehicle system dynamics. Vol. 30, No. 5, pp. 319-344*, 1998.

[5] P. Shakouri and A. Ordys. Application of the state-dependent nonlinear model predictive control in adaptive cruise control system. In *14th international IEEE conference on intelligent transportation systems (itsc), pp. 686 -691*, 2011.

[6] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, 2006.

[7] D. Swaroop. String stability of interconnected systems: An application to platooning in automated highway systems. In *Ph.D. Dissertation*, 1995.

[8] D. Swaroop and J. K. Hedrick. String stability of interconnected dynamic systems. In *IEEE Transactions on Automatic Control*, 1996.

[9] P. A. Ioannou and C. C. Chien. Autonomous intelligent cruise control. In *IEEE Transactions on Vehicular Technology, Vol. 42, NO. 4, pp. 657-672*, 1993.

[10] D. Swaroop. On the synthesis of controllers for continuous time lti systems that achieve a non-negative impulse response. In *Automatica*, 2003.