Stability analysis for a string of automated guided vehicles

Stability analysis for a string of automated guided vehicles

J.J.C. van Schendel

DCT 2006.083

Author:
J.J.C. van Schendel
Student number:
0547207
DCT number:
DCT 2006.083
Supervisors:
Dr. Ir. A. Serrarens
Ir. J. van Baalen (Innovius Automotive Mechatronics BV)

Department of Mechanical Engineering
TU Eindhoven
September 2006

Summary

Adaptive Cruise Control (ACC) is a system which maintains a speed set by the driver (like a conventional cruise control) and keeps a certain distance to its predecessor. When detecting a predecessor, it measures the distance to and the relative speed of the vehicle in front. If necessary, the speed is corrected to obtain a safe following distance. ACC has intensively being researched for years, resulting into the application on exclusive cars.

For technical reasons the working range of the ACC is limited: it cannot utilize the full braking capabilities of a vehicle (about 25%). The driver remains responsible for driving the car and must be prepared to overrule the ACC system in case of emergency at all times.

In the future, vehicles might drive fully automated. A first step towards fully automation is the application of technical control systems, such as ACC, lane keeping and obstacle avoidance. Further enhancements and integration of these systems can lead to “platooning” of vehicles, which means that vehicles drive automatically according to a car-following method.
In this project the objective is: analyse a string of ten AGVs on platoon stability. As a preparation for the project a literature study was done, to gain knowledge of the working concept of ACC and the effects of driving in platoons.

To be able to do research to the stability of an AGV string, a car-following model is being determined. To do this, first a single vehicle is modelled and since all cars in the platoon have the same dynamics, the single vehicle model is copied ten times. To control this string, equal P(I)D controllers are applied to all vehicles, except the leading vehicle. These controllers try to keep the headway distance as constant as possible and the velocity between subsequent vehicles error small.

The string is subjected to three different scenarios: two situations are conducted from real life and reflect calm changes in the traffic flow and one scenario is conducted from merging traffic. Under these circumstances the performance and stability of the string are determined in Simulink supported by formulated requirements.
Besides this research on stability in a time domain fashion, a theoretical stability based on pole analysis is determined from a system linearization and state-variable description matrices. It appears that the linearized system is stable for both PD and PID control.

Finally the gain and phase margins are determined for different string lengths with help of Nyquist plots. This stability depends on the chosen input and output signals. If the right in- and outputs are chosen, the string results to be stable for string lengths up to eight vehicles and unstable for strings longer than eight vehicles (Nyquist).

Conclusion is that both theory and simulation stability result in a stable system for strings containing up to eight cars. For strings containing more than eight vehicles, Nyquist and simulation give different results.

Contents

Introduction

The subject of design and analysis of various longitudinal control models has been studied since several decennia. The motivation has particularly been the high number of rear-end collisions and traffic jams, which are desired to be reduced. For this purpose many different vehicle-follower controllers, non-linear vehicle dynamics and automated guide way transit systems have been modelled. These systems might lead to automated guided vehicles (AGVs) in the future.

The background of this study concentrates on the development of the so called City Dual Mode (CDM). CDM forms the ensemble of slim vehicles that drive fully automated on a separate low-cost infrastructure in combination with manual driving in city areas.

Study aspects

In this study research is done to the stability aspects of the longitudinal control for an AGV: adaptive cruise control (ACC). In earlier research, many different ACC controllers are used: linear, non-linear, proportional, constant or smooth acceleration etc [Hatipoğlu, 1996]. For the stability analysis in this project, a P(I)D controller is sufficient.

To do the research, a string with a different number of cars is considered at a cruise speed of about 72 km/h. The distance between the vehicles has to be as constant as possible while the correction of errors has to be managed in a comfortable way. The most important aspect of the study is to determine the stability of the string of ACC controlled automated guided vehicles, both in theory and in Simulink, during braking and accelerating situations.

Goal

Model and simulate the dynamical behaviour of a string of ten vehicles equipped with ACC in order to test the system stability.

Report

In this report a study is performed on the stability of a string of AGVs, as mentioned above. To understand the context of this project, the terms and applications of adaptive cruise control (ACC) and automated guided vehicles (AGV) are explained in Chapter 1. Next a car-following model is defined to describe the non-linear dynamics of a single car and the behaviour of a string is described in Chapter 2. This dynamic vehicle model is implemented in Simulink in Chapter 3. After determining and adding a good P(I)D controller in the simulink model, a controlled string is simulated. The theoretical model has to be linearized, in order to prove theoretical stability in Chapter 4. In this chapter, the Simulink stability is determined as well.

Finally the problem of this study is solved: is the vehicle string stable? And under which conditions is this true? The conclusion and an outlook towards the near future will close this report.

Chapter 1

Adaptive Cruise Control
&
Automated Guided Vehicles

In this chapter an extensive description of Adaptive Cruise Control and Automated Guided Vehicles is given, to understand the context of this research. With help of earlier studies, the working concept and the application of these two systems are explained, so that they can be used to determine a good car-following model later on in this report.

1.1 Adaptive Cruise Control (ACC)

Adaptive or Advanced cruise control (ACC) is also called Intelligent cruise control (ICC) or Adaptive Intelligent cruise control (AICC) [Van Mieghem, 2004]. This system is an extension of the already well known cruise control, which can be found in lots of modern cars. The extension exists of a control structure that can influence the speed, to maintain a preset following distance or time. A few sensors determine whether there is a predecessor and they can determine the distance to and the relative speed of that predecessor as well. The detection element consists of a LIDAR-sensor (Light Detection and Ranging), which uses laser technology, or a RADAR sensor (Radio Detection and Ranging), which uses radio technology. Sometimes the system consists of a combination of both techniques. The range of these sensors is between 100 and 150 metres [Osugi, 1999].

.

1.1.1 Working concept

Higashimata (2001) and Holzmann (1997) give a good description of the working principle of an ACC. An adaptive cruise control has two inputs: a desired speed and a following time with respect to the car in front, both given in by the driver. The ACC continuously tries to realise both the desired speed and the following time. The system realises this by adjusting the throttle or brake pedal in order to solve the errors. The ACC functions as a conventional cruise control if there are no cars in front of the vehicle. So the system maintains the desired speed. When the sensors (Figure 1.1) detect a predecessor that drives slower or the distance becomes too small, the system adjusts the speed by lowering the throttle, shift to a lower gear or by active braking. Modern ACC systems have a maximal deceleration of about 3.5 m/s^2 [Jones, 2001].

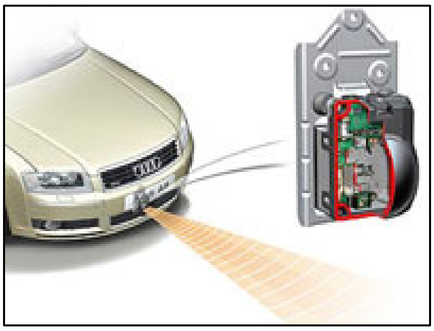


Figure 1.1: ACC sensors

When the desired braking force is larger than the maximum of the ACC system (25% of the vehicle's maximum), some signals are given to the driver that he should help the system by pressing the brake. The system can be shut down at any time, by pressing the brake or clutch or by pushing a button. The system can also be shut down temporarily, by pressing the throttle. In addition to some other modern technologies, an ACC does not communicate with other vehicles or road signs. That's why an ACC can be applied in all sorts of traffic [Van Mieghem].

An ACC replaces the driver in:

- Maintaining a desired constant speed.
- Decelerate when approaching a slower vehicle or when the headway distance is too small.
- Maintaining a desired constant distance to a predecessor.
- Accelerate to a desired speed when there are no obstacles ahead.

A driver should take over command when:

- The given acceleration or deceleration by the ACC does not require the demands, with respect to the speed of and the distance to the predecessor.
- Decreasing the speed during merging and in corners where predecessors are not seen by the sensors.
- The ACC controlled acceleration or deceleration is not comfortable.
- The ACC reacts on a predecessor that is not present ('ghost').

1.1.2 Application of ACC

ACC has a promising future, because of the expected positive effect on traffic safety. Due to its fast and adequate reaction, disturbances in the traffic flow are absorbed more easily. By experts ACC is seen as a first step towards fully automatic driving in the future.
However, the working range of an ACC is limited because of technical reasons: it cannot utilize the full braking capacity of a vehicle. This is why the driver remains responsible for driving the car and should be prepared to overrule the ACC system in case of emergency. According to Hoetink (2003), ACC systems are only effective at highways with little traffic. Else, safety can't be guaranteed. Nevertheless in some modern cars the ACC system is already integrated, for example in the more luxurious Mercedes S500 in Figure 1.2, where the ACC system is called DISTRONIC.

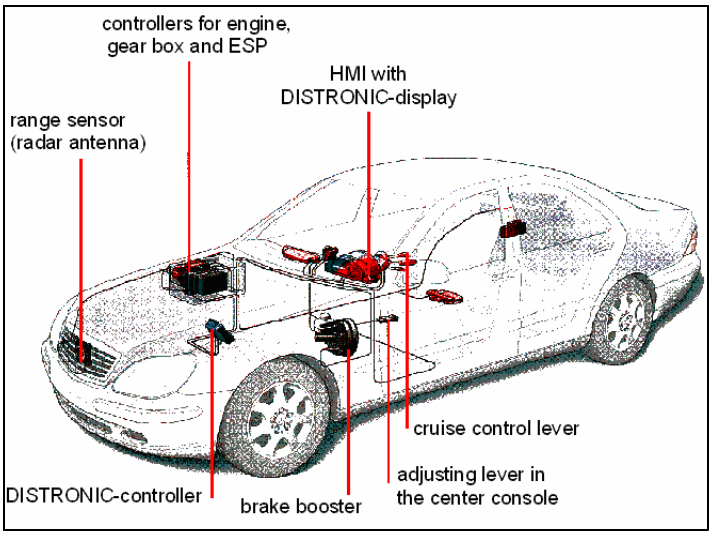

controllers for engine, gear box and ESP
HMI with DISTRONIC-display
range sensor (radar antenna)
cruise control lever
adjusting lever in the center console
DISTRONIC-controller
brake booster

Figure 1.2: Mercedes S500 DISTRONIC

1.2 Automated Guided Vehicle (AGV)

Automated guided vehicles are computer controlled, possibly unmanned, vehicles. They are being used mainly in production and storage environments since many years and help to reduce the costs of manufacturing and increase efficiency in manufacturing systems. However, there are some projects where (semi) AGVs are applied in public transport, for example the ParkShuttle shown in Figure 1.3, driving between the metro station Kralingsezoom and the Rivium.

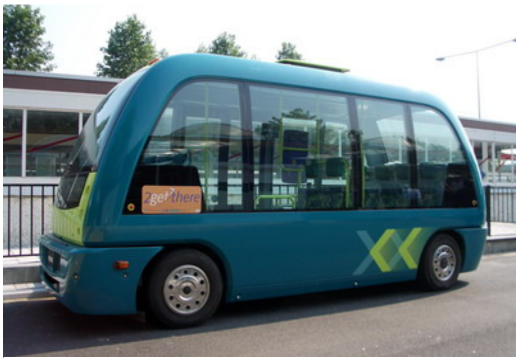


Figure 1.3: ParkShuttle

1.2.1 Working concept

The AGV mode is supposed to be entirely autonomous, which means the driver is not allowed to do anything. The navigation of AGVs is controlled with the help of vision/radar based sensors. These sensors are being used for different purposes. A few sensors locate the vehicle's position with respect to road signs. These signs can be all sorts of things, for example a painted line on the road or a magnet strip in the road [Weyns, 2005]. Using other sensors, the distance to and the relative speed of the preceding vehicle can be computed, resulting in a form of adaptive cruise control. Further, a smart system with a global positioning system (GPS) computes the global position of the car and computes which path the car follows from point A to point B [Slocum, 2001]. Steering is entirely managed by an electronic steered control.

1.2.2 The Cito Dual Mode

The Cito Dual Mode (CDM) transportation system is an example of an AGV and is designed by Small Advanced Mobility BV (SAM), Modesi BV and Innovius Automotive Mechatronics BV. The Cito (Figure 1.4) is a 1+1-person vehicle that can be driven both automatically and manually. In both drive-modes (dual mode) steering, braking and propulsion are fully managed electronically, which is called 'x-by-wire'. The difference is that in the manual mode the driver controls the vehicle by commanding the tasks.

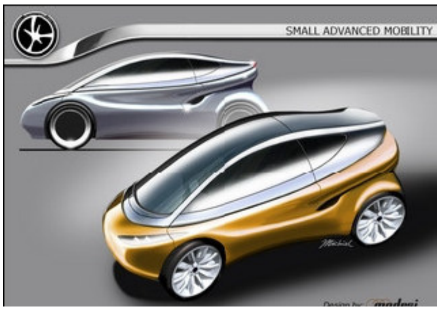


Figure 1.4: The Cito

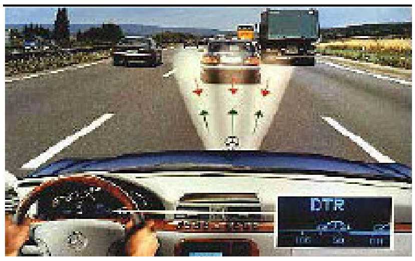


Figure 1.5: ACC system

The vehicle guidance is taken over by an electrical control system during the automatic navigated mode. In this mode, the vehicle keeps a constant distance to the car in front with the help of sensors. This longitudinal control is realised by adaptive cruise control (Figure 1.5).

Chapter 2

Car following model

As mentioned in Bengtsson (2001), the human driver behaviour has been studied since the beginning of the 1950s by Reushel (1950) and Pipes (1953). But during the 1990s the topic has grown considerably. The division of driver behaviour into separately studied parts has been a common theme, since a general driver model is very complex. For example, there are many separate models for describing steering control, longitudinal behaviour and safety behaviour.

This chapter concentrates on the longitudinal behaviour model of a platoon of AGVs, which is needed to determine the string stability. Firstly, a dynamic model of a single vehicle is given. This model is used to describe the string model with several vehicles. Because both models appear to be non-linear, both the model of the single car and the string have to be linearized to compute a linear string model.

2.1 Dynamic vehicle model

To describe the longitudinal motion of a string of vehicles, firstly the longitudinal motion of a single vehicle has to be described. This can be done with Newton's first law [Franklin, 2002]:

$$\sum F = M \cdot a . \qquad (2.1)$$

In this equation M is the mass of the vehicle, a is the acceleration and $\sum F$ is the sum of the forces acting on the vehicle. According to Sheikholeslam (1989), the sum of the forces is described by four forces: the air resistance force F_{air}, the rolling resistance force F_{roll}, the gravity force F_g and the driving force F_{drive}, which is the force that the engine will produce to achieve the acceleration. Because it's assumed that the road surface is horizontally, the force due to gravity equals zero:

$$F_g = M \cdot g \cdot \sin(\theta) , \qquad (2.2)$$

in which θ denotes the angle between the road surface and a horizontal plane (equal to zero) and g is the acceleration due to gravity.

Next, the dynamic model of a vehicle is formulated and showed in Figure 2.1:

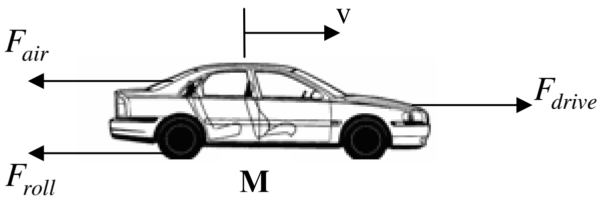


Figure 2.1: Dynamic vehicle model

The total equation of motion can be derived from Figure 2.1 and is described by:

$$F_{drive} - F_{roll} - F_{air} = M \cdot a\ , \tag{2.3}$$

in which [Baert, 2006],

$$F_{roll} = M \cdot g \cdot f_r \tag{2.4}$$

$$F_{air} = 0.5 \cdot C_d \cdot A_f \cdot \rho \cdot v^2\ , \tag{2.5}$$

with C_d: drag coefficient
A_f: frontal area
ρ: specific mass of air
v: velocity of the vehicle
f_r: rolling resistance coefficient

When adding equations (2.4) and (2.5) into equation (2.3), the acceleration of the vehicle is described by:

$$a = \frac{F_{drive} - M \cdot g \cdot f_r - 0.5 \cdot C_d \cdot A_f \cdot \rho \cdot v^2}{M}\ . \tag{2.6}$$

2.2 Platoon configuration

Like in earlier work in car-following driver modelling [Bengtsson], each car is able to percept the distance to and the relative speed of the preceding car with help of ACC sensors. Figure 2.2 shows the assumed platoon configuration for a platoon of three vehicles. This platoon is assumed to move in a straight line, without any merging traffic and the desired vehicle headway distance is 50 m.

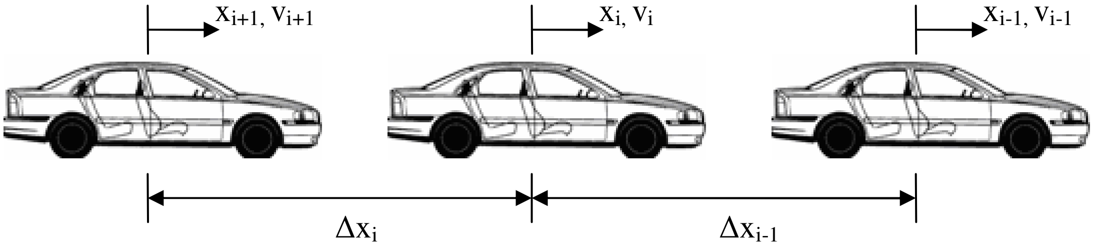


Figure 2.2: Platoon configuration

In Figure 2.2 is:
v_i: absolute velocity of vehicle *i*.
x_i: absolute position of vehicle *i*

$$\Delta x_i = x_i - x_{i+1} : \text{headway distance} \tag{2.7}$$

The absolute values are measured from a certain begin position *x(0)* at starting time *t(0)*. With these absolute values, the acceleration of each vehicle in the string can be calculated, as described in paragraph 2.1 (equation 2.6).

2.3 Theoretical linearization

Simulink is a computational tool capable of simulating the behaviour of a string of vehicles. Unfortunately the stability boundaries of the system are hard to determine using Simulink. So another method has to be used to determine the theoretical stability. The solution lies in the formulation of a state-space model. Because the system is non-linear (see equation 2.6), the state-space model has to be linearized to get a linear model from which the stability can be determined. This is done with help of the lecture notes of Kok (2002), first for a single vehicle and finally for a complete string of controlled vehicles in this paragraph. Before the models are linearized, the transfer function of the controller is determined and implemented in the model equations.

2.3.1 Controller

A transfer function should be determined to implement the characteristics of the controller into the system description. Because the controller is linear and works with different actions, gains and inputs, it can be written in the Laplace domain [Franklin]. The controller is a PID-type and its inputs are the position error and the velocity error, so the controller equation can be written as:

$$C = x_{error} \cdot P + \int x_{error} dt \cdot I + v_{error} \cdot D \,. \qquad (2.8)$$

To convert equation 2.8 into the Laplace domain, the following is determined in Franklin: an integral becomes $1/s$ in the Laplace domain and a differential becomes s. According to Qiu (2002) the resulting Laplace equation of the PID controller is:

$$C(s) = P + \frac{I}{s} + D \cdot s \,. \qquad (2.9)$$

This is equivalent with:

$$C(s) \cdot s = P \cdot s + I + D \cdot s^2 \quad . \qquad (2.10)$$

With help of equation 2.10, the transfer function of a PID controller can be computed:

$$C_{PID}(s) = \frac{D \cdot s^2 + P \cdot s + I}{s} . \qquad \text{(PID controller)} \qquad (2.11)$$

The transfer function of a PD controller is almost the same as the PID controller in equation 2.11, only I is equal to 0:

$$C_{PD}(s) = \frac{D \cdot s^2 + P \cdot s}{s} . \qquad \text{(PD controller)} \qquad (2.12)$$

These controller descriptions can be used, together with the vehicle descriptions in paragraphs 2.3.2 and 2.3.3, to determine the stability of controlled vehicles.

2.3.2 Single vehicle

To determine a state space model of a single vehicle, first a system state and a system input have to be formulated. Because all vehicles can percept the position to and the relative velocity of the preceding vehicle, these two signals are chosen to be the system state *q*. Moreover, the output of the controller is F_{drive}, so the system input u is equal to F_{drive} [Sheikholeslam, 1989]:

$$q = \begin{bmatrix} x & v \end{bmatrix}^T, \tag{2.13}$$

$$u = \begin{bmatrix} F_{drive} \end{bmatrix}. \tag{2.14}$$

The derivative of the system state q becomes:

$$\dot{q} = \begin{bmatrix} \dot{x} & \dot{v} \end{bmatrix}^T = \begin{bmatrix} v & a \end{bmatrix}^T. \tag{2.15}$$

To obtain a linear system description of a single vehicle, the equation of motion in formula (2.6) has to be linearized. The non-linearity in this equation is due to the air resistance force F_{air}, which is a function of the square of the velocity. So F_{air} has to be linearized about a steady state velocity: $v_{ss} = 20 m/s$. With this assumption, the linearization can be done [Kok, 2002]:

$$F_{air} = 0.5 \cdot C_d \cdot A_f \cdot \rho \cdot v^2, \tag{2.5}$$

$$F_{air,lin} = C_d \cdot A_f \cdot \rho \cdot v_{ss} \cdot v = \gamma \cdot v_{ss} \cdot v, \tag{2.16}$$

in which $\gamma = C_d \cdot A_f \cdot \rho$ is a constant term.
Together with formulas (2.6), (2.15) and (2.16) a system description can be determined:

$$\dot{q} = A \cdot q + B \cdot u + W, \tag{2.17}$$

$$\dot{q} = \begin{bmatrix} 0 & 1 \\ 0 & -\gamma \cdot v_{ss} / M \end{bmatrix} \cdot q + \begin{bmatrix} 0 \\ 1/M \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ -F_{roll} / M \end{bmatrix}. \tag{2.18}$$

The output *y* of the system is equal to the position and is described by:

$$y = [x] = C \cdot q + D \cdot u, \tag{2.19}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot q + [0] \cdot u. \tag{2.20}$$

With the state-variable description matrices *A*, *B*, *C* and *D* from equations (2.18) and (2.20) and the controller description (paragraph 2.3.1), a Nyquist plot can be made using Matlab.

2.3.3 String of PID controlled vehicles

The stability of a controlled single vehicle can be computed by Nyquist, but the stability of a string of PID controlled vehicles can be calculated by writing the string in the following state space equation [Franklin, 2002]:

$$\dot{q} = A \cdot q + W \;, \tag{2.21}$$

in which *W* is a constant disturbance matrix containing the constant terms that are independent of the system state. The stability can now be computed from matrix *A* (Chapter 4). Moreover, the state-variable description matrix *B* is not present, because applying the PID controllers the system input *u* can be written in terms of the system state *q*, so that *B* equals zero. This will be explained further on.

Before the state space model from equation 2.21 can be used, a system state of the string has to be determined. As in Liang (1998), this system state contains the position (x_i), the velocity (v_i) and the acceleration (a_i) of each vehicle in the string. In this project the maximal number of cars in the string is assumed to be ten vehicles. As an example, the number of cars is set to two cars, to keep the calculation clear. So the system state equals:

$$q = \begin{bmatrix} x_0 & x_1 & v_0 & v_1 & a_0 & a_1 \end{bmatrix}^T . \tag{2.22}$$

Than, the derivative of the system state is:

$$\dot{q} = \begin{bmatrix} v_0 & v_1 & a_0 & a_1 & \dot{a}_0 & \dot{a}_1 \end{bmatrix}^T , \tag{2.23}$$

in which $\dot{a}_0$ and $\dot{a}_1$ are the jerks of the two vehicles in the string. With the assumption that the acceleration of the leading car (a_0) is constant, the jerk (the derivative of the acceleration) of that leader equals zero. Together with the single vehicle model, a PID controller and equation 2.6, the acceleration is described by:

$$a_i = \frac{F_{drive} - M \cdot g \cdot f_r - 0.5 \cdot C_d \cdot A_f \cdot \rho \cdot v_i^2}{M} , \tag{2.6}$$

in which the driving force of a PID position controlled vehicle equals:

$$F_{drive} = P \cdot (x_{i-1} - x_i - 50) + I \cdot \int (x_{i-1} - x_i - 50) dt + D \cdot (v_{i-1} - v_i) . \tag{2.24}$$

So the jerk of the i-th vehicle is described by:

$$\dot{a}_i = \frac{P \cdot (v_{i-1} - v_i) + I \cdot (x_{i-1} - x_i - 50) + D \cdot (a_{i-1} - a_i) - C_d \cdot A_f \cdot \rho \cdot v_i \cdot a_i}{M} \tag{2.25}$$

The string description follows from a combination of equations 2.22, 2.23 and 2.25:

$$\dot{q}=\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{I}{M} & -\frac{I}{M} & \frac{P}{M} & \frac{-P-C_d\cdot A_f\cdot\rho\cdot a_{ss}}{M} & \frac{D}{M} & \frac{-D-C_d\cdot A_f\cdot\rho\cdot v_{ss}}{M} \end{bmatrix}\cdot q+\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{-I\cdot 50}{M} \end{bmatrix} \quad (2.26)$$

Remark: the fifth row in matrix *A* equals zero, because of the assumption that the jerk of the first vehicle is zero. The derivation of a PD controlled string is given in Appendix 5.

According to Franklin, a Nyquist plot is needed to determine the gain and phase margins. This is not possible with the system description above, so another state space model has to be determined:

$$\dot{q}=A\cdot q+B\cdot u+W \quad (2.27)$$
$$y=C\cdot q+D\cdot u, \quad (2.28)$$

in which *q* is the system state, *u* the system's input and *y* the system's output. Because the acceleration, position and velocity of the leading vehicle are prescribed, these three signals are supposed to be the system's input (*u*). Moreover the system's output equals the velocity of the following vehicles. With these assumptions, the next states can be determined for a string containing a leader and two following vehicles:

$$q=\begin{bmatrix} x_1 & x_1-x_2 & v_1 & v_2 & a_1 & a_2 \end{bmatrix}, \quad (2.29)$$
$$u=\begin{bmatrix} x_0 & v_0 & a_0 \end{bmatrix}, \quad (2.30)$$
$$y=\begin{bmatrix} v_1 & v_2 \end{bmatrix}. \quad (2.31)$$

Together with equations 2.27 and 2.28 the following state-variable description matrices of a string containing three vehicles can be computed:

$$A=\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-I}{M} & 0 & \frac{-P-\gamma\cdot a_{ss}}{M} & 0 & \frac{-D-\gamma\cdot v_{ss}}{M} & 0 \\ \frac{I}{M} & \frac{I}{M} & \frac{P}{M} & \frac{-P-\gamma\cdot a_{ss}}{M} & \frac{D}{M} & \frac{-D-\gamma\cdot v_{ss}}{M} \end{bmatrix}, \quad (2.32)$$

in which a_{ss} and v_{ss} are the steady state acceleration (equal to 0 m/s^2) and the steady state velocity (equal to 20 m/s) respectively.

The other matrices are:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{I}{M} & \frac{P}{M} & \frac{D}{M} \\ 0 & 0 & 0 \end{bmatrix}, \qquad (2.33)$$

$$W = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{-50 \cdot I}{M} \\ \frac{-50 \cdot I}{M} \end{bmatrix}, \qquad (2.34)$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \qquad (2.35)$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (2.36)$$

With these *A*, *B*, *C* and *D* matrices, the system can be described and a Nyquist plot can be made in Matlab. In Chapter 4 the Nyquist plots and the resulting gain and phase margins will be discussed.

Chapter 3:

Matlab implementation

After modelling the single vehicle and the controlled string in the previous chapter, the platoon model can be implemented in Matlab Simulink. This is needed to simulate the behaviour of the string and determine the stability in Simulink.

3.1 Vehicle dynamics model

After making a dynamic vehicle model in Chapter 2, a Simulink model can be made. Because all vehicles in the string are assumed to have the same dynamics and control, all the Simulink components are the same.

Each vehicle model has two inputs [Liang, 1999], namely the position and the velocity. With the use of a controller between the vehicles, the errors of these inputs are kept as small as possible. Besides, the controller manages the needed acceleration, so that the ride is comfortable and the errors are damped as quickly as possible.

With help of the dynamic vehicle model, the next simulink model of the vehicle dynamics can be made (Figure 3.1):

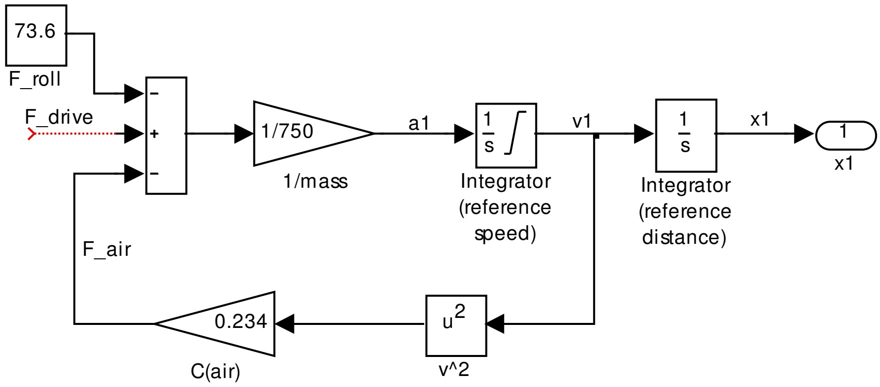


Figure 3.1: Simulink model of the vehicle dynamics

In this picture F_{drive} is the output of the controller and is denoted as *u* in Figure 3.2. Further, F_{roll} is a constant and F_{air} is a function of the square of the velocity (equation 2.5).

The integrators in Figure 3.1 both have an initial condition. This is needed to prescribe the beginning position. These conditions are given by: initial speed is 20 m/s and the initial headway distance is 50 m. By changing these conditions, the behaviour of the string can be determined.

Like Sheikholeslam did, here the acceleration of the leading vehicle of the string is prescribed, so that the behaviour of the predecessors can be determined. Whether the string is stable, is discussed in Chapter 4.

3.2 PD controller

To control a string of vehicles, a controller is needed to compute the driving forces of the vehicles. This controller calculates the needed force to eliminate the errors, with help of the inputs. Since the position is controlled, the P action of the controller acts on the position error and the D action acts on the derivative of the position error. This means that the D action acts on the velocity error, which can be returned directly from the vehicle dynamics [Franklin, 2002].

The PD controller model is shown in Figure 3.2:

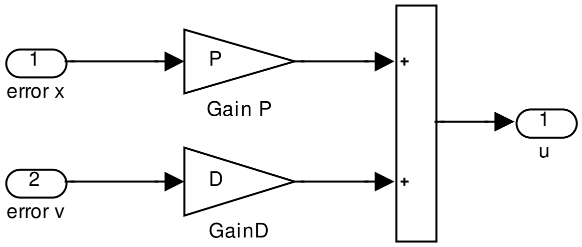


Figure 3.2: PD controller Simulink model

To compute the P and D gains, requirements are defined in Appendix 1. Together with these requirements, a good approach for computing the P, I and D gains is given by Zhong (2006) in Table 1:

Table 1: Effects of P, I and D action

Action	Rise time	Overshoot	Settling time	Steady state error
P	Shorter	Larger	No effect	Smaller
I	Shorter	Larger	Longer	No error
D	No effect	Smaller	Shorter	No effect

In this table the effects are given when the P, I or D gains are increased. So for example, when the D gain is raised, no effect is shown in the rise time.

The best way to determine the P and D gains is explained in Zhong: begin with only a P action. When the overshoot is at a reasonable value, a D action can be applied. If the damping is good enough, the right values are determined.

An approximation of the P gain can be calculated with help of some initial conditions and assumptions, which is done in Appendix 2. The eventual P and D values that are found with the trial and error method are given in Chapter 4.

3.3 PID controller

As in Chapter 4 will be shown, the PD controller has a steady state error. To eliminate this error, an I action is added. Because of the I action, an extra controllable signal is required. As in the PD controller of the previous section, the P action acts on the position error and the D action on the velocity error. So the I action acts on the integrated position error. This results in the next PID Simulink model (Figure 3.3):

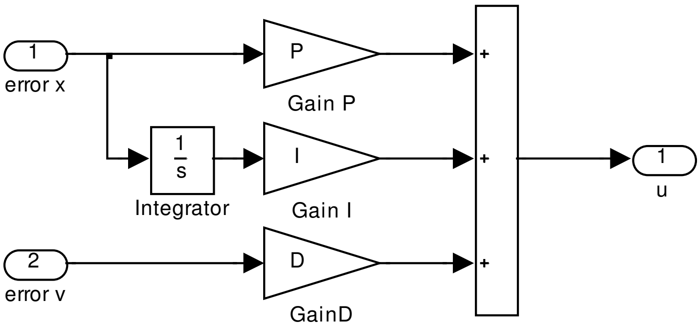


Figure 3.3: PID controller Simulink model

In this Simulink model the P, I and D are constant gains. The integrator has an initial condition such that the acceleration at starting time ($t = 0$) is equal to zero.

The computed P, I and D gains in Chapter 4 are found with help of Appendix 2, Table 1, the requirements mentioned in Appendix 1 and the trial and error method.

Chapter 4

Stability analysis

To check whether the used controller is stable, a stability analysis is done. Firstly, the definition of stability is explained. Secondly, the stability of the string is determined with Matlab Simulink. Next the theoretical stability is determined from system matrix *A*. Finally the gain and phase margins of different string lengths are computed with help of Nyquist plots.

4.1 Stability definition

To determine the stability of a (controlled) system, first a clear definition should be defined. For Nyquist plots this definition is given by Franklin: the curve should keep the point (-1,0) to its left. But when no Nyquist plot can be made, other definitions are valid. For example, when the system matrix *A* is given in an equation like 2.26, the stability can be determined from the poles of this matrix *A*. Namely, if the poles (eigenvalues) lie in the left half plane, the system is stable [Franklin, 2002]. So if the real part of the pole is negative, the system is stable. If the pole lies on the imaginary axis, the system is marginally stable. Else the system is unstable.

There is a way to determine the stability of the models with Simulink: the errors have to become smaller during simulation time (damping) and become constant. If so, the performance of the controller can be determined with help of some boundaries. Most of these boundaries are copied from earlier work [BMW, 2000] and given in Appendix 1. For example, the maximal steady state error of the final vehicle is 10 cm. Whether or not the string is stable (enough) depends on the defined boundaries. So, if stricter boundaries are desired, the controller has to be adapted. To restrict the scope of this project the requirements of a stable string are limited and are given in Appendix 1.

4.2 Model simulations (Simulink)

As mentioned in the previous paragraph, the approximate stability of the string can be determined in Simulink by applying some restrictions. Because two different controllers were designed, in this paragraph both PD and PID controller Simulink results are discussed. The stability is determined by simulating two small excursions and a more extreme one mentioned in Güvenç (2006):

- Accelerate from 20 m/s (72 km/h) to 27.8 m/s (100 km/h) in 15 seconds
- Decelerate from 20 m/s (72 km/h) to 13.9 m/s (50 km/h) in 15 seconds
- Changing the initial headway from 50 m to 40 m (headway time of 2 s at 20 m/s)

4.2.1 PD controller results

The PD controller consists of a proportional and derivative control. The proportional gain is calculated in Appendix 2: $P = 50$. With this P gain and the trial and error method, the following D gain is computed: $D = 700$. When simulating the small acceleration on the string of ten vehicles, the next result was obtained:

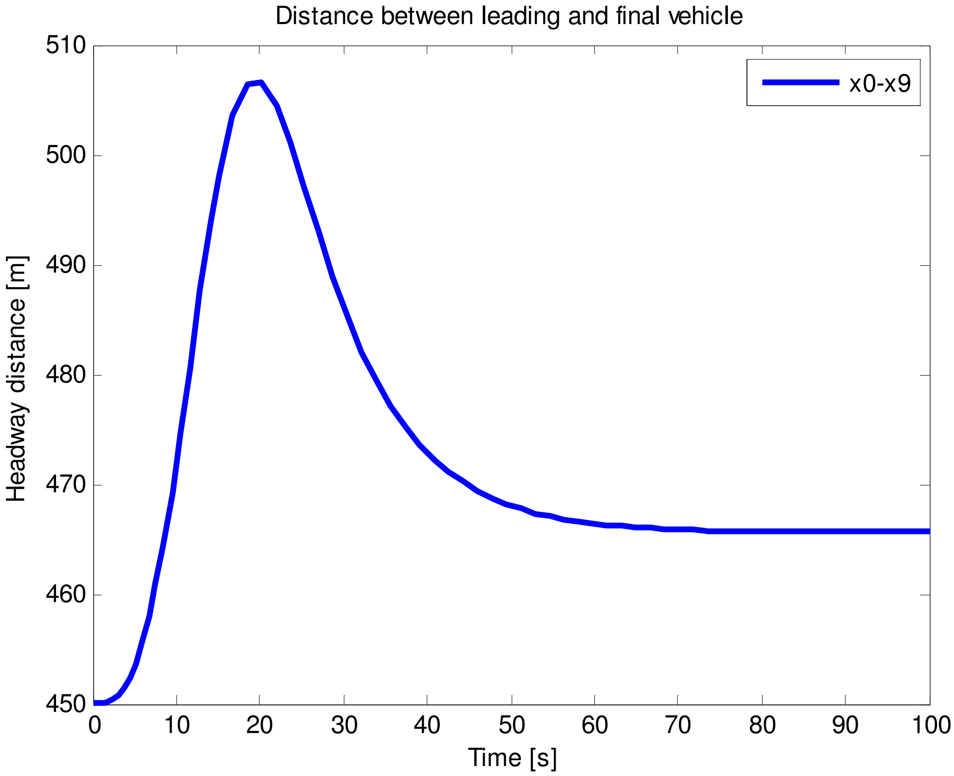


Figure 4.1: Headway distance between vehicle 0 and 9 during small acceleration

In Figure 4.1 the headway distance error between the leader of the string and the last vehicle is shown. Here the initial headway distance is 450 m ($9 \cdot 50$). Figure 4.2 shows that the error is stable, because the error is damped and becomes constant. However the performance isn't satisfying the conditions in Appendix 1: the steady state error is about 15 m and the maximal overshoot is 56 m. Moreover, the settling time is about 50 seconds.

This slow damping and these large errors lead to the dynamical behaviour of the vehicles in the string in Figure 4.2. As can be seen, the headway distance between the cars (Time [s] is 20) is much too small, the overshoot is very large and the settling time is too long.

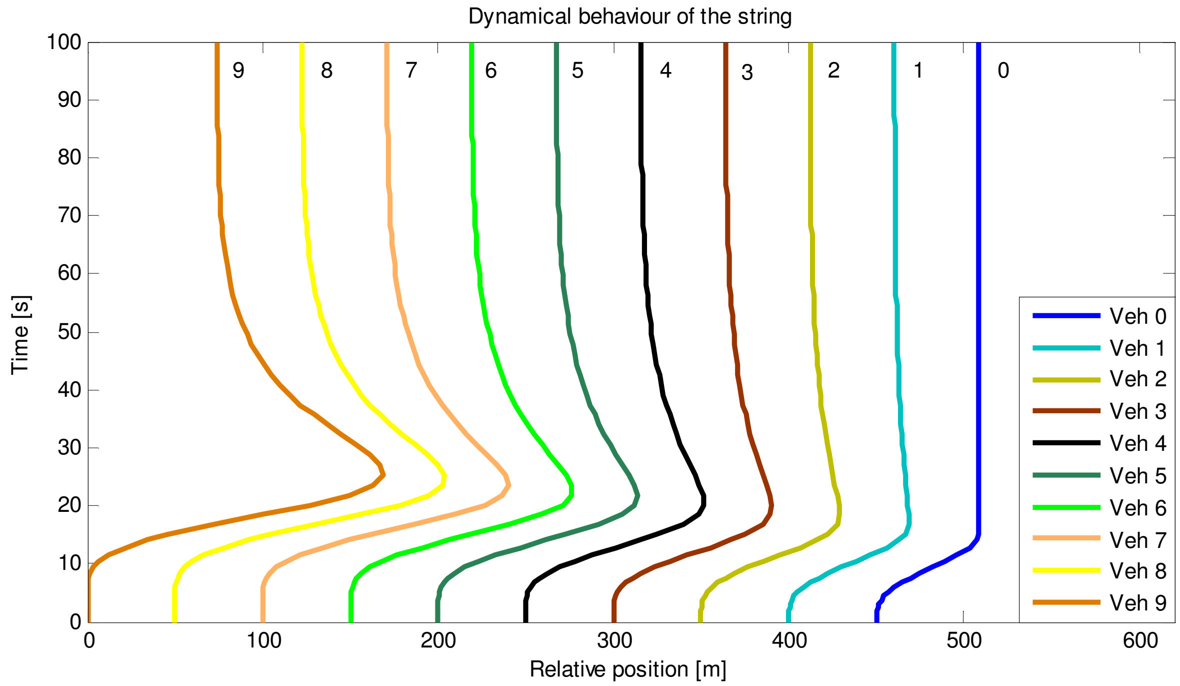


Figure 4.2: Dynamical behaviour of the vehicles in the string

Although the acceleration behaviour of this PD position controlled string is stable (Figure A4.1 in Appendix 4), a PD controller is desired that decreases the overshoot and the settling time of the position error. The only way to do this is increasing both the P and the D gain. With the trial and error method the following P and D gains are found: $P = 650$ and $D = 1720$. With this controller the position error during the small acceleration is shown in Figure 4.3:

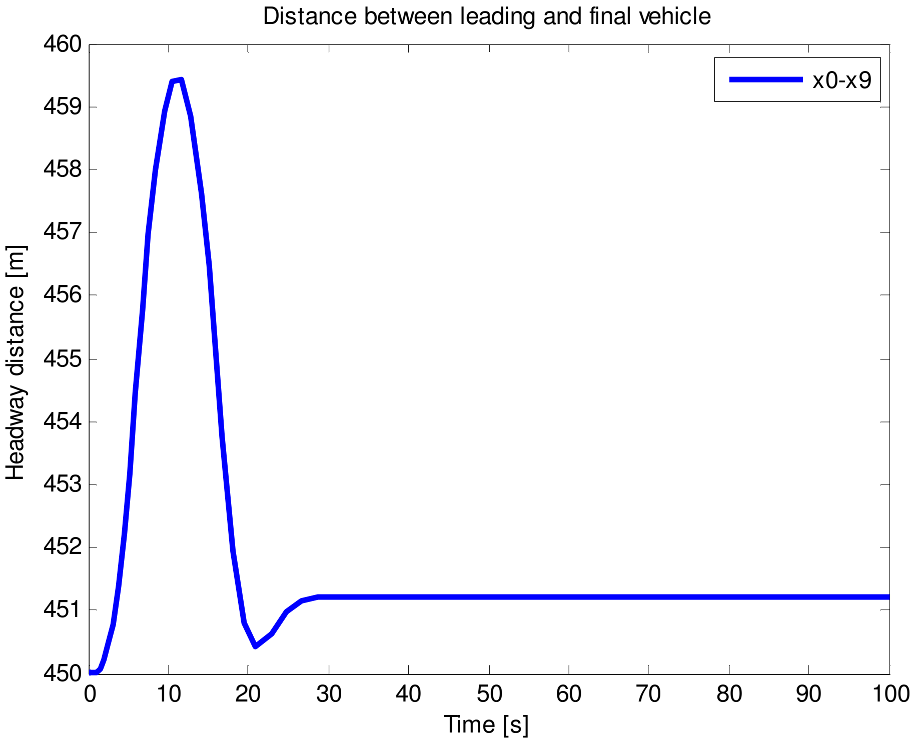


Figure 4.3: Headway distance with new controller during small acceleration

In this figure, a stable system is shown where both overshoot and settling time are intensively decreased: the maximal overshoot is 9.5 m and the settling time is about 30 seconds. These two results satisfy the performance requirements in Appendix 1. The only problem is the steady-state error, which is about 1.2 m and still much too large according to the requirement (maximum is 10 cm). To minimize this steady state error an I action is applied in paragraph 4.2.2.

During the deceleration circumstance, the following position error between the leading and final vehicle is found:

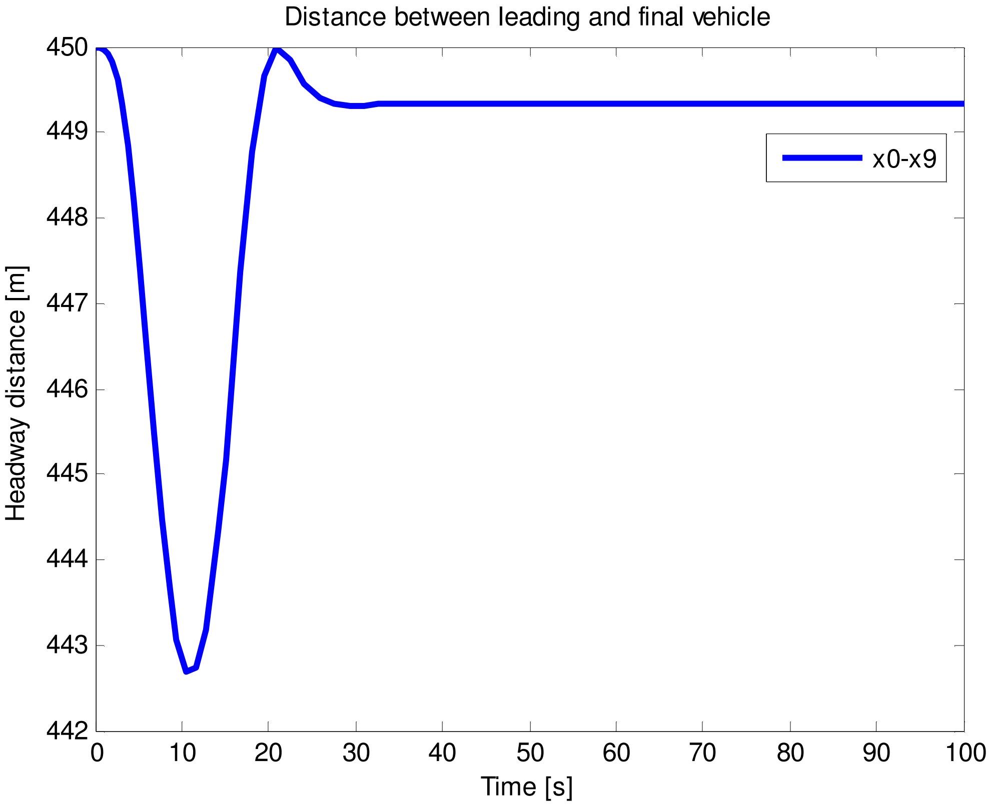


Figure 4.4: Headway distance during small deceleration

In Figure 4.4 the maximal overshoot is about 7.4 m and the settling time is 32 seconds. Just like the result during a small acceleration, this result satisfies the requirements. But like in Figure 4.3, the steady state error of 75 cm exceeds the maximum of 10 cm, so both situations require an I action.

4.2.2 PID controller results

As mentioned in Table 1, the way to eliminate a steady state error is applying an I action. Because the overshoot and settling time of the previous PD controller satisfy the requirements, these two gains are kept the same. The needed I action is determined with the trial and error method and results in the following gains: $P = 650$, $I = 9.4$ and $D = 1720$.

To test the stability of this controller, the excursions mentioned in paragraph 4.2 are simulated in Simulink.

First the small acceleration to speed up from a velocity of 20 m/s (72 km/h) to 27.8 m/s (100 km/h) is simulated. The position error between the leader and the final vehicle during this situation is depicted in Figure 4.5:

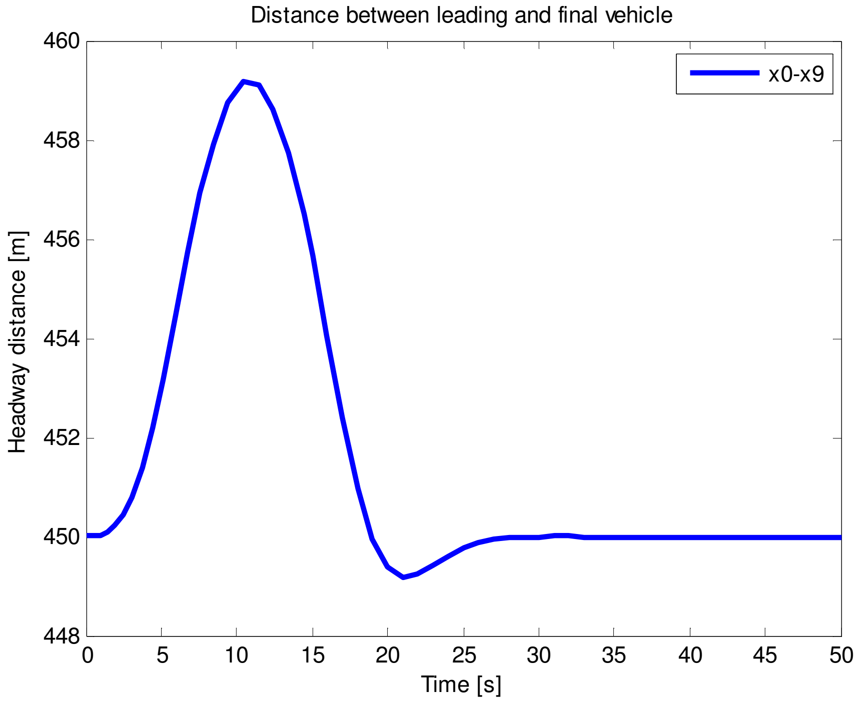


Figure 4.5: Headway distance error of the PID controlled string

In comparison with Figure 4.3, both overshoot and settling time are the same. The difference is the steady state error, which is decreased to less than 1 cm. Also the acceleration of the final vehicle in Figure 4.6 satisfies the requirements: the maximal acceleration is 1.2 m/s^2 and the settling time is about 35 seconds. Both Figure 4.5 and Figure 4.6 show a stable system: the error tends towards a constant value.

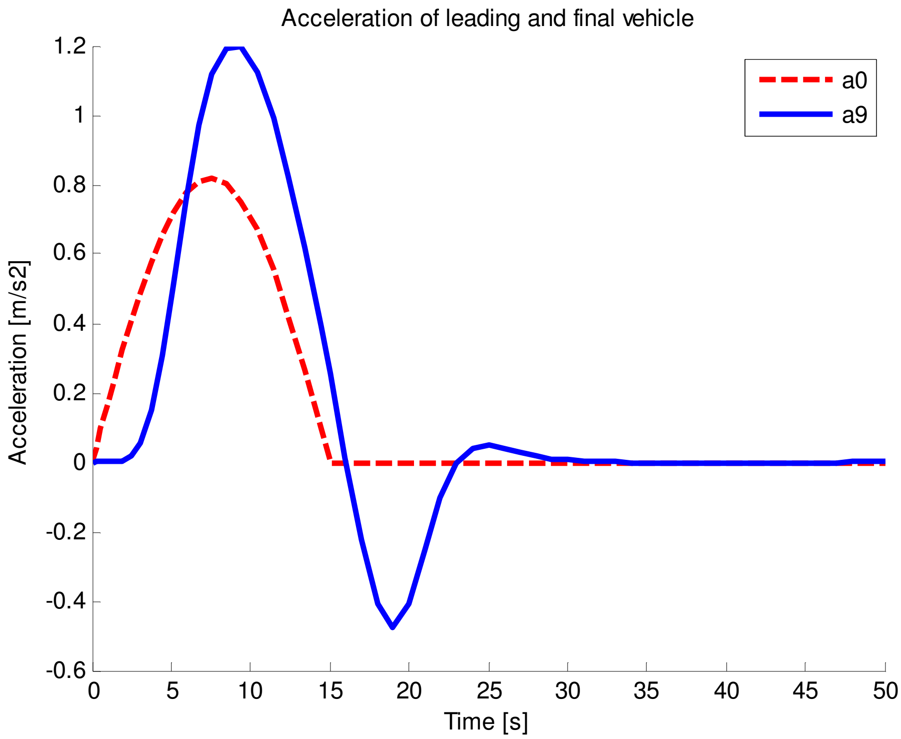


Figure 4.6: Acceleration error of the PID controlled string

With this PID controller, the resulting dynamic behaviour of all vehicles in the string during a small acceleration is depicted in Figure 4.7:

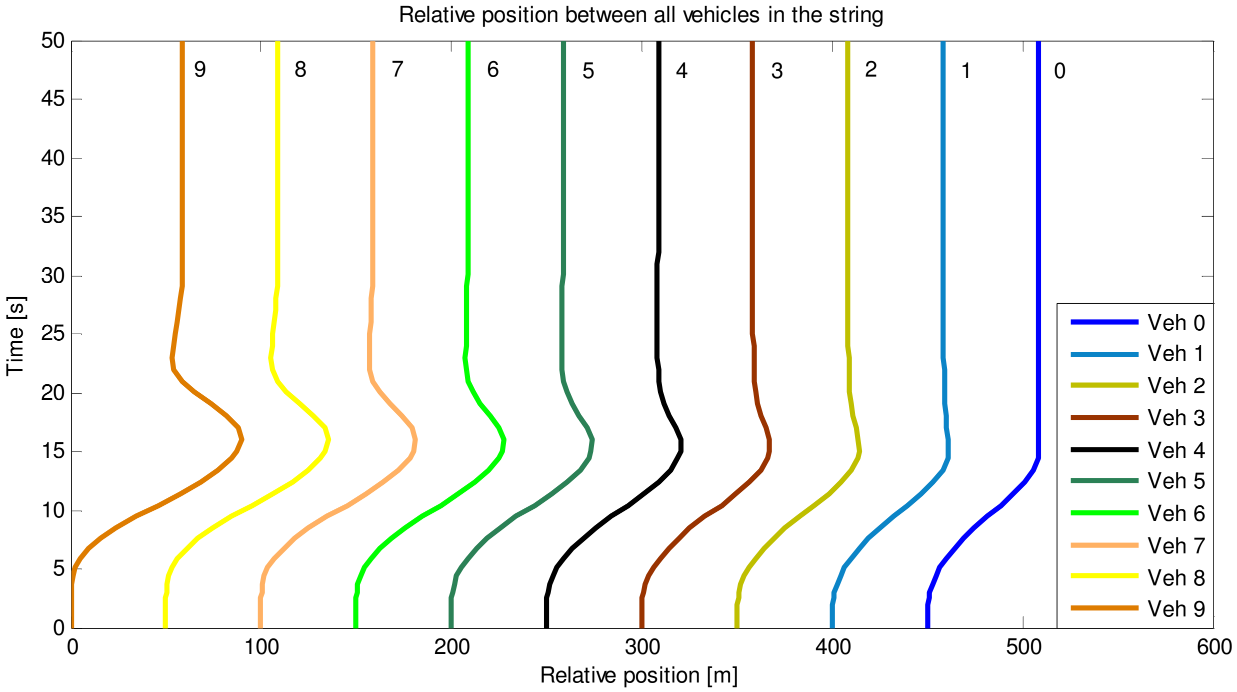


Figure 4.7: Dynamic behaviour of all vehicles in the string

In this figure the relative position between the vehicles is shown as a function of the time. As can be seen the overshoot increases as the string gets longer.

The next step is simulating the other small excursion: decelerating from 20 m/s (72 km/h) to 13.9 m/s (50 km/h) in 15 seconds. The resulting headway error between the leading and the final car during this situation is shown in Figure 4.8:

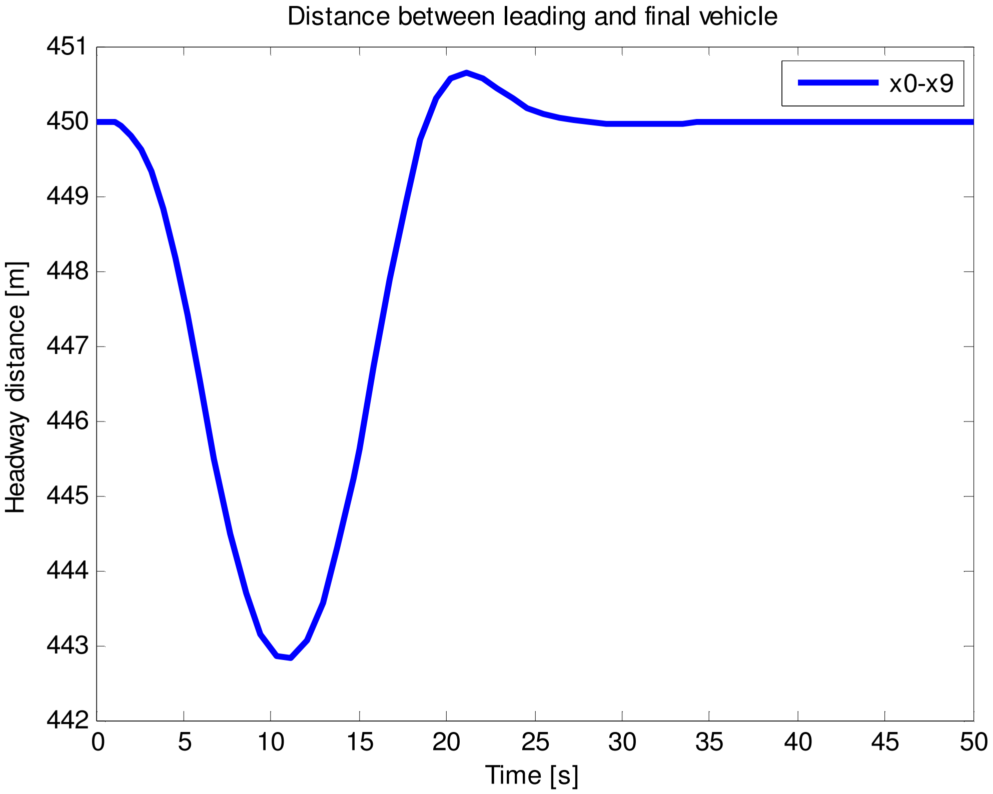


Figure 4.8: Headway distance error during small deceleration

This result is stable and satisfies the requirements: the overshoot (7.2 m) is less than 10 m, the settling time (35 s) is shorter than 40 seconds and the steady state error (2 cm) is smaller than 10 cm. The resulting decelerating of the leading and final vehicle is shown in Figure 4.9.

Here the maximal deceleration of 0.94 m/s^2 is less than the system maximum of 2 m/s^2, the maximal acceleration of 0.38 m/s^2 is less than the system maximum of 1.2 m/s^2 and the resulting settling time of the final vehicle's acceleration of 33 seconds is shorter than the maximum of 40 seconds. This results in a comfortable and stable ride.

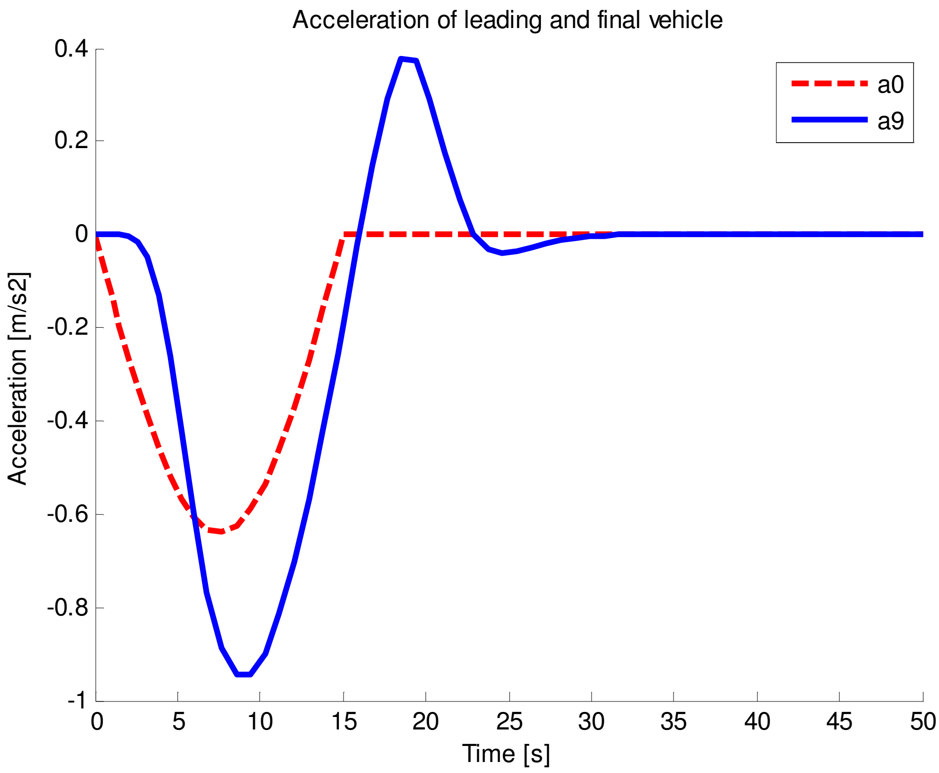


Figure 4.9: Acceleration error during small deceleration

The previous two excursions are very common in real traffic. But to check whether the string is stable during a more extreme excursion, the next situation is simulated: an initial headway distance of 40 m instead of 50 m (which is achieved by changing the initial value in the integrator), while the desired headway distance remains 50 m. So the following vehicle should decelerate and enlarge his distance to the vehicle in front from 40 m to 50 m. The position error during this circumstance is pictured in Figure 4.10:

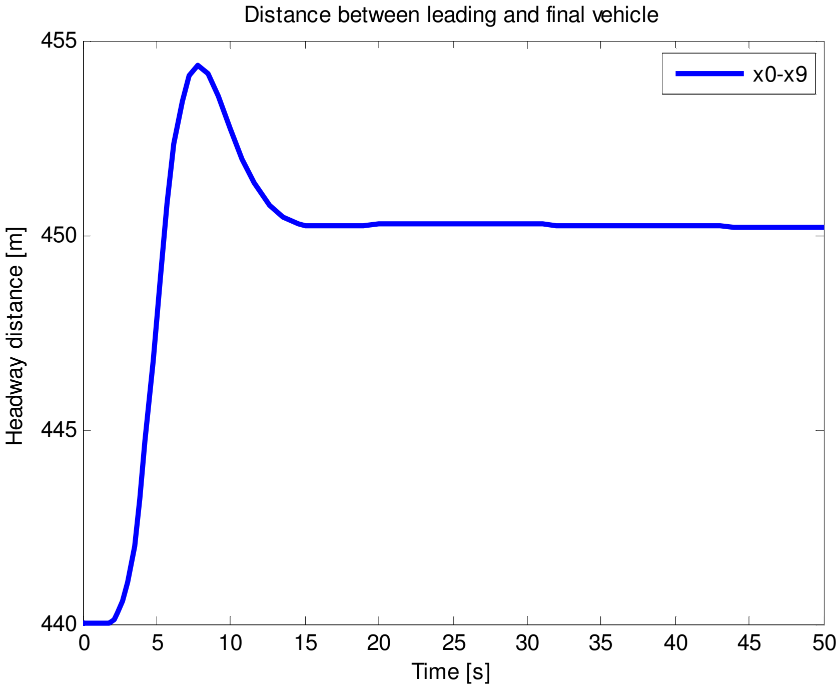


Figure 4.10: Headway error during change in initial headway

In the figure can be seen that the overshoot satisfies the requirement: the overshoot is about 4.5 m, which is less than the required 10 m. Unfortunately the settling time is very long: at time is 50 seconds, the error is still 20 cm. At time is 90 seconds this error is less than 10 cm. So the position behaviour of the final vehicle in the string is stable but does not satisfy all requirements. When the acceleration error is plotted (Figure 4.11), it's obvious that the designed controller does not satisfy the acceleration requirements in Appendix 1.

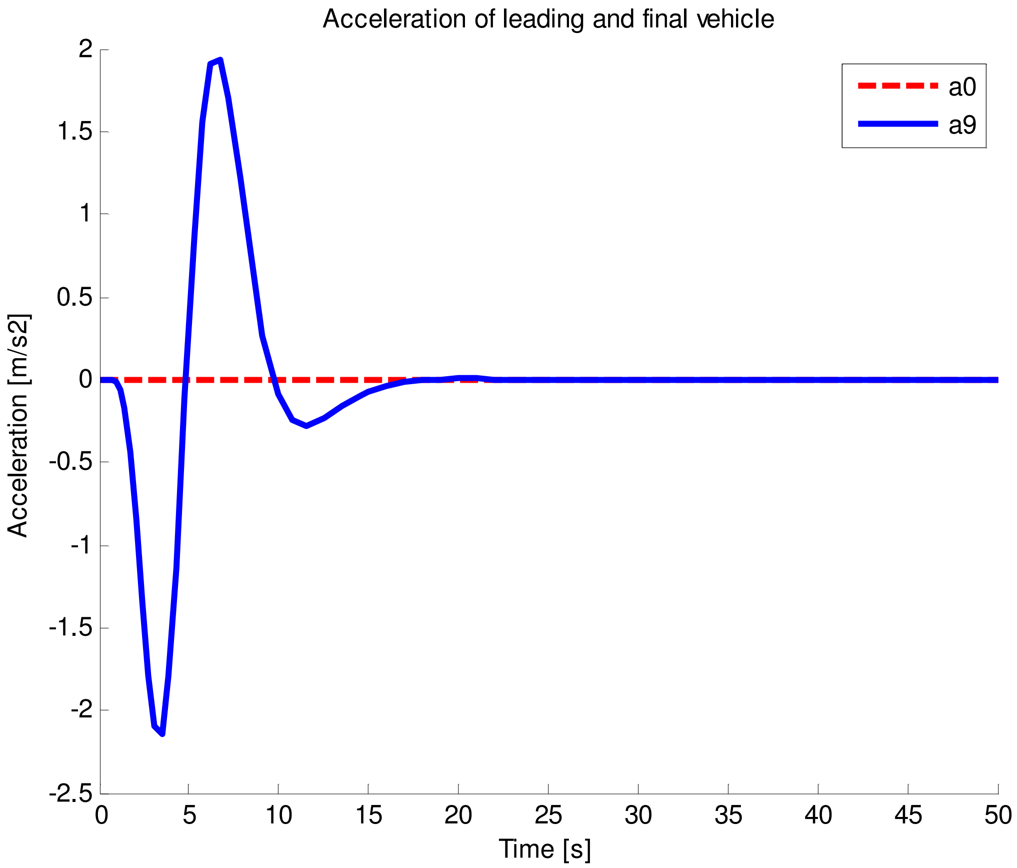


Figure 4.11: Acceleration error during a change in headway distance

The figure shows that the maximal acceleration of the final car in the string is about 2 m/s^2, while a maximum of 1.2 m/s^2 is allowed. Also the maximal deceleration of about 2.2 m/s^2 exceeds the maximum of 2 m/s^2.

Conclusion: the PID controller almost satisfies all requirements during a small headway distance change.

4.3 Theoretical stability

The stability of the PID controlled string in paragraph 2.3.3 can be computed theoretically. As described in Franklin, the stability can be determined from the state-variable description matrix *A*: if the poles of matrix *A* lie in the left half plane, the system is stable. If the poles lie on the imaginary axis the system is said to be marginally stable and if the poles lie in right half plane the system is unstable. With the state-variable description matrix *A* of two vehicles from paragraph 2.3.3 (equation 2.26) and Matlab, the poles (eigenvalues) can be computed.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{I}{M} & -\frac{I}{M} & \frac{P}{M} & \frac{-P - C_d \cdot A \cdot \rho \cdot a_{ss}}{M} & \frac{D}{M} & \frac{-D - C_d \cdot A \cdot \rho \cdot v_{ss}}{M} \end{bmatrix} \tag{4.1}$$

After implementing matrix *A* and all values of the variables (Appendix 6) and $P = 650$, $I = 9.4$ and $D = 1720$ in Matlab, the poles are computed with the command '*eig*'. The resulting poles are:

$$\lambda_1 = 0,\ \lambda_2 = 0,\ \lambda_3 = 0,\ \lambda_4 = -1.84,\ \lambda_5 = -0.45,\ \lambda_6 = -0.02$$

The first three poles are 0, because the jerk of the first car is assumed to be zero and is a function of x_0, v_0 and a_0. The other eigenvalues only have a negative real part, so all lie in the left half plane, which means that they are all stable and so is this system. When the number of vehicles in the string is raised to a number higher than eight vehicles (nine or ten), all poles (except for the three poles equal to zero) stay in the left half plane and the string remains stable. In the next paragraph the gain and phase margins of the stable system are discussed.

4.3.1 Gain and phase margins (Nyquist)

After the conclusion that the controlled string is stable, the gain and phase margins are computed with help of Nyquist plots to determine the stability margin in the system. These plots can be made with the matrices determined in paragraph 2.3.3 and Matlab (functions '*ss2tf*', '*tf*' and '*nyquist*'). With matrices *A* (2.32), *B* (2.33), *C* (2.35) and *D* (2.36), a system transfer function is made in Matlab. When this is done for the string with ten vehicles, Nyquist plots can be made with different input and output signals.

First the velocity of the leader is chosen to be the input signal and the velocity of the second follower is the output signal. This results in the next Nyquist plot:

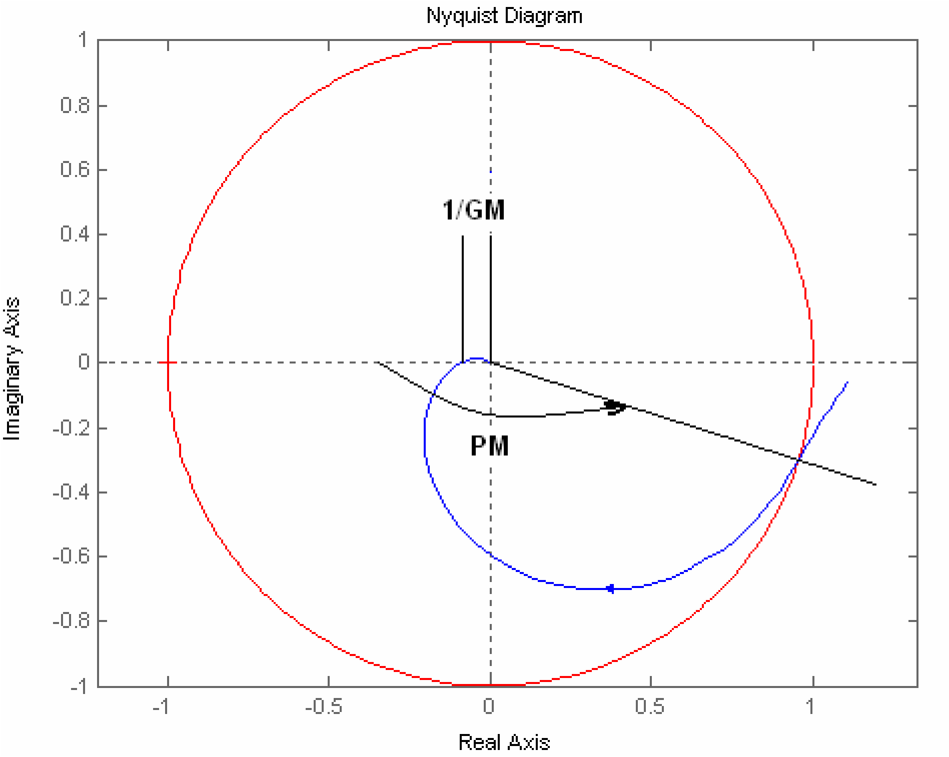


Figure 4.12: Nyquist plot with input v0 and output v2

In Figure 4.12 the gain (GM) and phase (PM) margins are depicted with the circle of radius 1. The value of the GM equals 10.6 and PM is 160°. For a stable system PM is positive and GM is greater than 1 [Franklin], which means that the system with input v0 and output v2 is stable. The same can be done with the velocity of the seventh follower (v7). The resulting Nyquist plot is given in Figure 4.13, where the plot keeps (-1,0) to it's left and GM equals 1.1 and PM equals 21°. So when the output is chosen to be the velocity of the eighth vehicle (v7), the system is still stable.

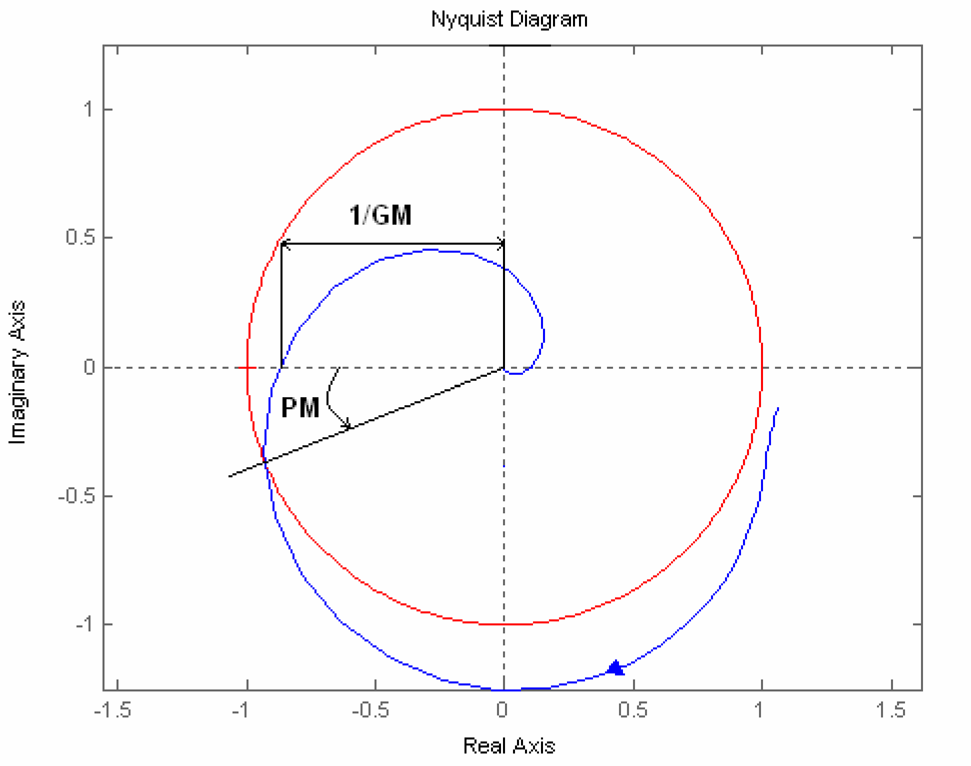


Figure 4.13: Nyquist plot with input v0 and output v7

Finally, the gain and phase margins of the total string with ten vehicles are determined. The result is shown in figure 4.14, in which can be seen that the Nyquist plot keeps the point (-1,0) to it's right. Concluded is that the string with 10 vehicles (or 9, as shown in Appendix A4.2) is unstable.

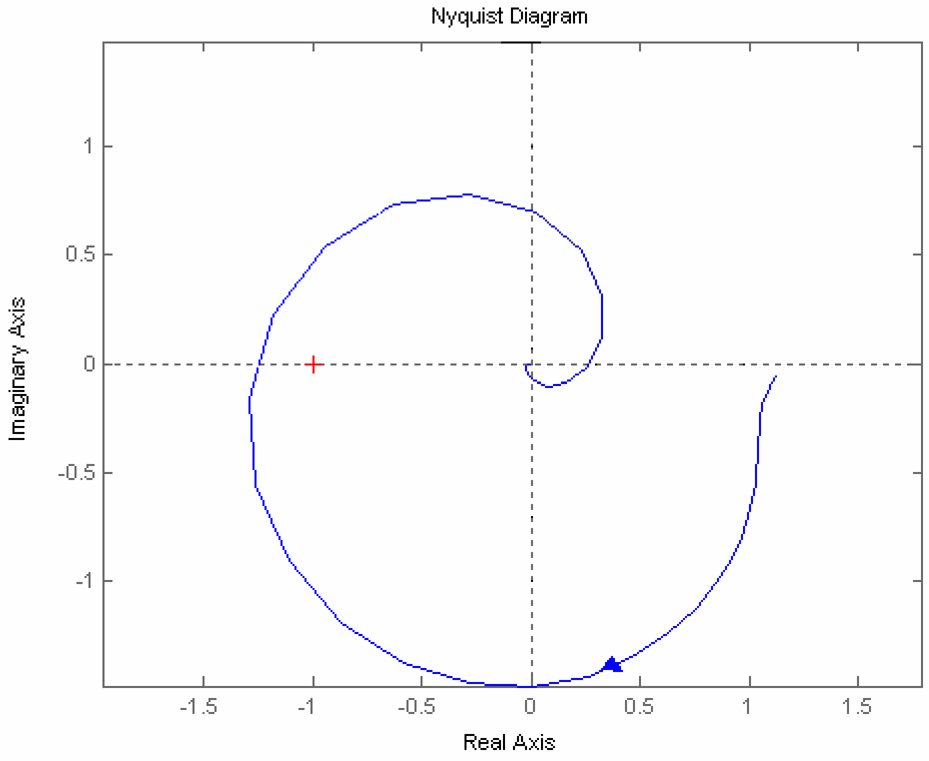


Figure 4.14: Nyquist plot with input v0 and output v9

Chapter 5

Conclusions and Future work

A stability analyses for a string of Automated Guided Vehicles is performed in this report. From the Simulink results using the controllers can be concluded that the performance of the PID controlled string satisfied the system requirements under normal conditions, which were small accelerations without any merging traffic. Certain manoeuvres resulted in a slower and less accurate performance. This happens only when vehicles merge into the string.
From the theoretically determined stability with Nyquist plots in Matlab can be concluded that certain string lengths resulted in Nyquist instability: the maximal length of a stable string is eight vehicles. When the string gets longer, the Nyquist plots show instability, while the theoretically determined eigenvalues remain stable. This means that theory and simulation did give the same stability result, only the Nyquist plots become instable for strings longer than eight vehicles. This might be caused by the non-linear term that can have a stabilising effect on the Simulink model but is not shown in the Nyquist plot.
Another reason for the Nyquist instability might be the effect of the input signals (u), which is defined in state-variable matrix B, on the state-space model, because this matrix B is only used in the Nyquist definition.

This research reviewed a PD and a PID controlled string. Because of the Nyquist instability, the stability of strings with multiple vehicles (more than eight) should be researched more accurate, so that the controllers can be adjusted to the length of the string to keep stability at any time.
Although the performance seemed to be good enough for small excursions, it will certainly not be suitable for more extreme excursions, such as emergency braking and merging traffic, which asks for further research to more extreme control strategies. In this project is also assumed that the vehicles can only determine the distance to and the relative velocity of the preceding vehicle. With the modern communications systems of today, all vehicles can communicate with each other and the controllers can become more sophisticated than the ones used in this report.

In the future, one of the important issues that needs to be addressed, is the conditions under which string instability occurs. Also the requirements mentioned in Appendix 1 should be examined, because modern systems are more accurate, faster and have a bigger range. Moreover, the possibilities of other control algorithms should be explored and designed to improve the following behaviour of the vehicles in the string. These control algorithms may be robust to communication delay and to changes in manoeuvres. Another possibility is that the controller can be switched with other controllers during certain circumstances. This might lead to good following behaviour under any circumstance.

Appendices

Appendix 1:

Requirements for the controllers

Some requirements for the controller are defined with help of BMW (2000):

For a comfortable ride the next accelerations are assumed:

- The maximal acceleration is 1.2 m/s^2
- The maximal deceleration is 2.0 m/s^2
- The maximal deceleration during an emergency stop is 5 m/s^2

Some goals have to be made with respect to overshoot, steady state error and settling time.

- The maximal steady state error of the position of the final vehicle is 10 cm.
- The maximal position overshoot of the final vehicle is set to 10 m.
- The maximal settling time of the final vehicle is 40 seconds.

Furthermore [BMW]:

- The vehicles should not crash, so the headway distance is bigger than 0 m at all time.
- The minimal initial distance between two cars is 20 m, which is equal to 1 s at 20 m/s.
- The maximal initial distance between two cars is 80 m, which is equal to 4 s at 20 m/s.
- The initial speed of all cars is set to 20 m/s.
- The desired distance between two cars is 50 m, which is equal to a headway time of 2.5 s at 20 m/s.

Appendix 2:

Calculation of P gain

The P action in the PD and PID controllers can be calculated with help of the maximal comfortable deceleration of 2 m/s^2 and the minimal distance between two cars of 20 m (equals 1 second at 20 m/s) [BMW]. Furthermore, the two following cars are assumed to have the same initial speed of 20 m/s.
Due to these assumptions and the construction of the controller (see Figure 3.3), the derivative part of the controller is equal to zero. So the approximation of the P action can be calculated:

$$\begin{aligned} \Delta x &= x_{i-1} - x_i = 20 \\ x_{error} &= x_{i-1} - x_i - 50 = -30 \end{aligned} \tag{A.1}$$

$$\begin{aligned} a &= \frac{F_{drive} - F_{roll} - F_{air}}{M} \\ &= \frac{P \cdot |x_{error}| + I \cdot \int x_{error} dt + D \cdot v_{error} - M \cdot g \cdot f_r - 0.5 \cdot C_d \cdot A \cdot \rho \cdot v_i^2}{M}, \end{aligned} \tag{A.2}$$

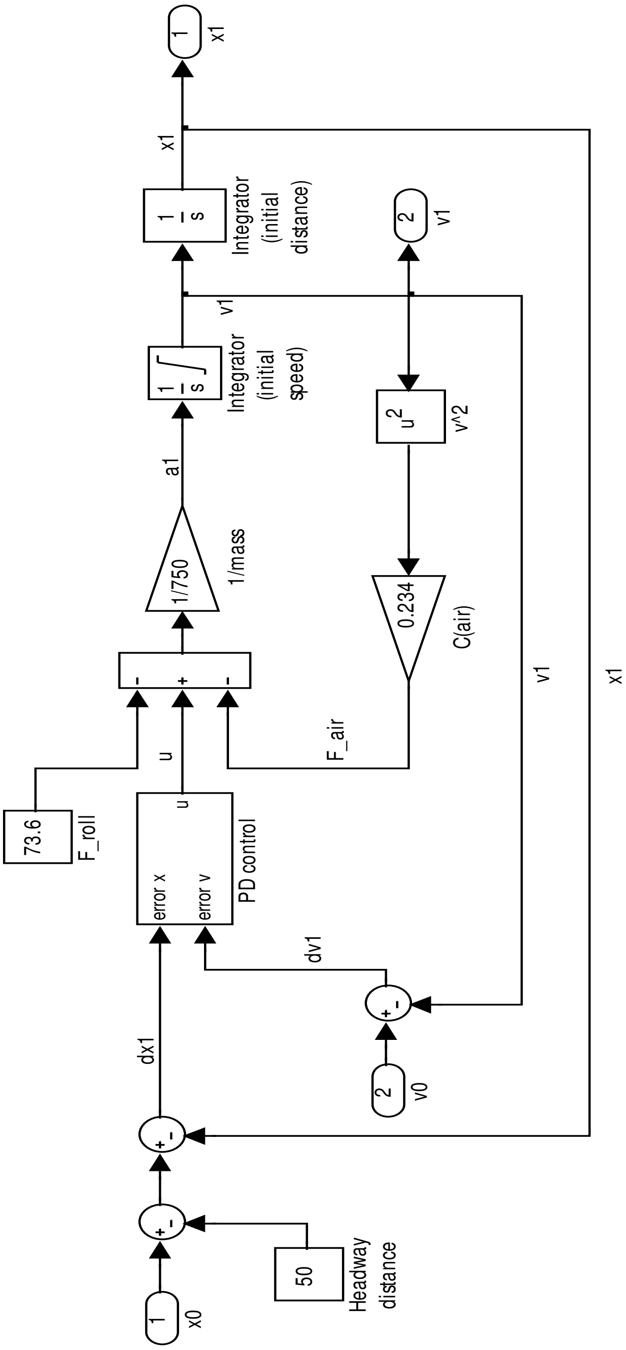
in which $I \cdot \int x_{error} dt = 167.2$ at $t = 0$, due to the initial condition that the acceleration at t = 0 equals 0. So, the initial value of the I action is equal to sum of the rolling resistance force and the air resistance force, which equals 167.2. With this condition the next P gain can be computed:

$$2 = \frac{P \cdot 30}{750} \rightarrow P = 50 \tag{A.3}$$

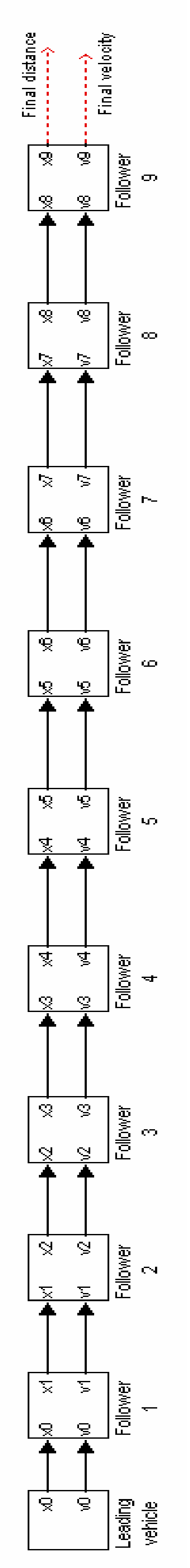
Appendix 3:

Complete Simulink model

One controlled vehicle:



Controlled string:



Appendix 4:

Simulink results

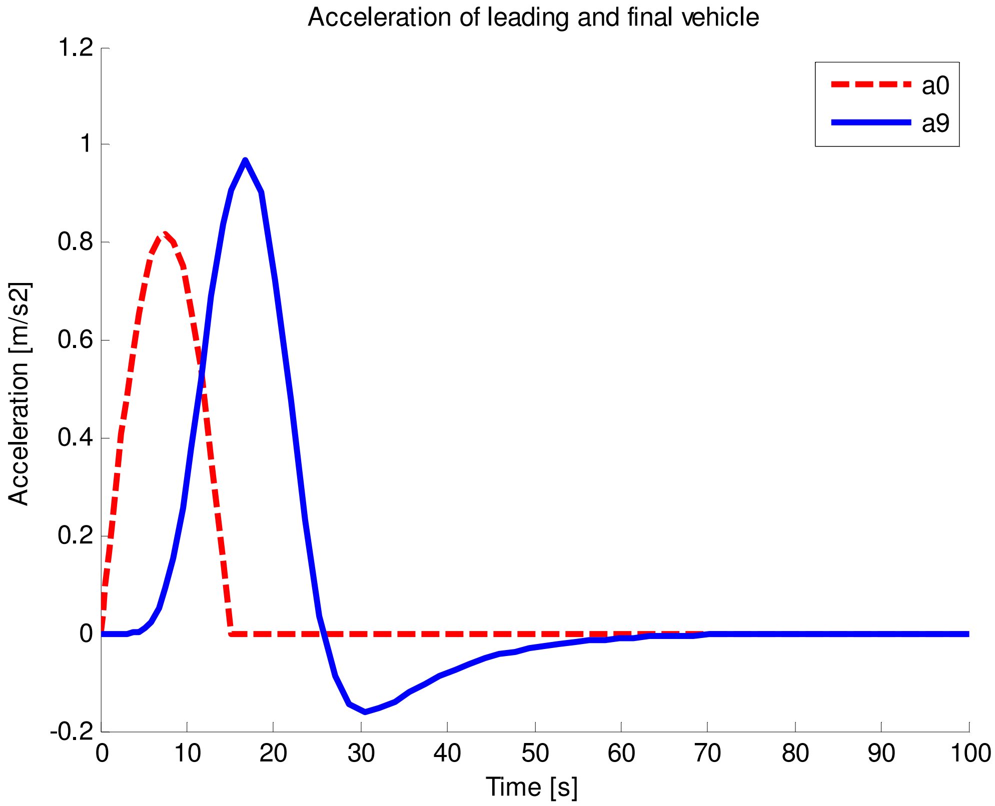


Figure A4.1: Acceleration of leading and final vehicle during small acceleration

Figure A4.1 is the acceleration result of the first and final vehicle in the string during the small acceleration manoeuvre (from 20 m/s to 27.8 m/s).

As depicted, the acceleration is stable because it tends towards a constant value of 0 m/s^2. Moreover, it satisfies the requirements in Appendix 1.

Matlab Nyquist results:

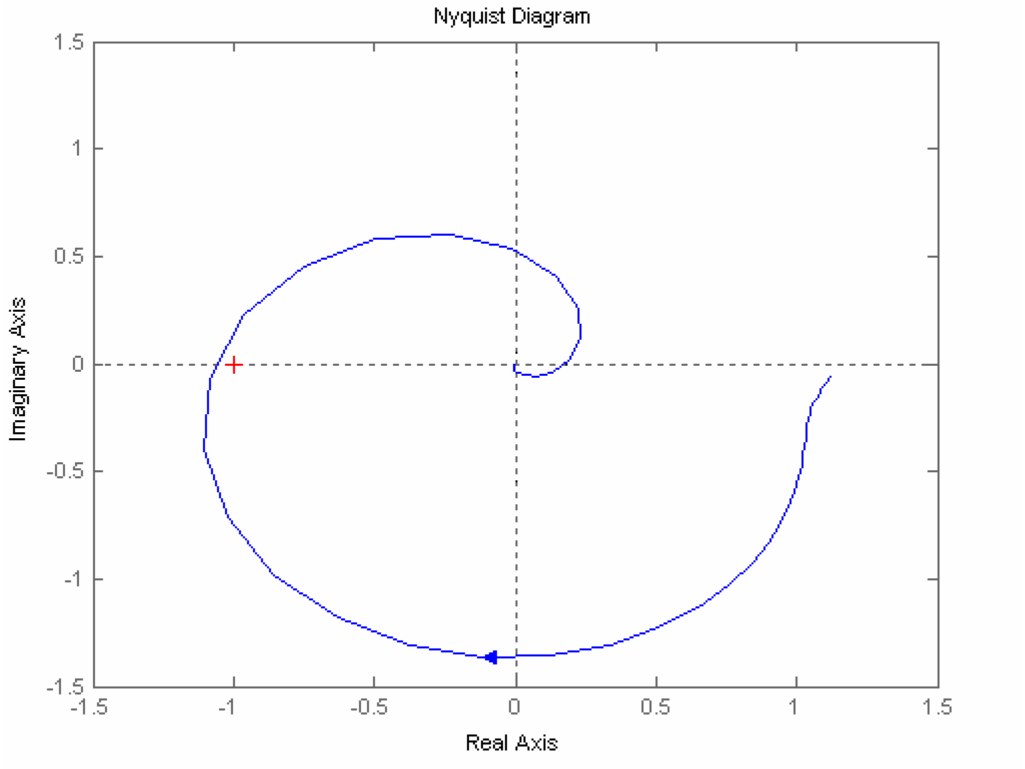


Figure A4.2: Nyquist plot with input v0 and output v8

In this figure is shown that the string with 9 vehicles is unstable ((-1,0) right from plot).

Appendix 5:

PD controlled string linearization

By writing the PD controlled string in the following system description, the stability can be computed from matrix *A*:

$$\dot{q} = A \cdot q + W \tag{2.21}$$

Where *W* is a constant disturbance matrix.
As in paragraph 2.3.3 is described, first of all a system state of the string should be determined. The system state for the string is the same as the one of the single vehicle, only the string state contains the position and the velocity of each vehicle in the string. In this project the maximal length of the string is assumed to be ten vehicles. To keep the calculation clear, here the string length is set to three vehicles. So the system state equals:

$$q = \begin{bmatrix} x_0 & x_1 & x_2 & v_0 & v_1 & v_2 \end{bmatrix}^T \tag{A.5}$$

In appendix 3 the Simulink model of a single vehicle is shown. Together with this model, Figure 3.2 and paragraph 2.3.3, the equation of motion of all vehicles can be determined:

$$a_i = \frac{P \cdot (x_{i-1} - x_i - 50) + D \cdot (v_{i-1} - v_i) - F_{roll} - F_{air}}{M} \tag{A.6}$$

The derivative of the system state is:

$$\dot{q} = \begin{bmatrix} v_0 & v_1 & v_2 & a_0 & a_1 & a_2 \end{bmatrix}^T \tag{A.7}$$

Together with the assumption that the acceleration of the first vehicle is zero, the resulting state space model of a PD controlled string is:

$$\dot{q} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{P}{M} & -\frac{P}{M} & 0 & \frac{D}{M} & \frac{-D - C_d \cdot A \cdot \rho \cdot v_{ss}}{M} & 0 \\ 0 & \frac{P}{M} & -\frac{P}{M} & 0 & \frac{D}{M} & \frac{-D - C_d \cdot A \cdot \rho \cdot v_{ss}}{M} \end{bmatrix} \cdot q + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{-P \cdot 50 - F_{roll}}{M} \\ \frac{-P \cdot 50 - F_{roll}}{M} \end{bmatrix}$$

Appendix 6:

Nomenclature, parameters and values

Symbol	Description	Value	Dimension
x_i	Position of the i'th vehicle	-	m
v_i	Velocity of the i'th vehicle	-	m/s
a_i	Acceleration of the i'th vehicle	-	m/s^2
$\dot{a}_i$	Jerk of the i'th vehicle	-	m/s^3
M	Vehicle mass	750	kg
C_d	Drag coefficient	0.3	m
A_f	Frontal area of the vehicle	1.3	m^2
ρ	Specific mass of air	1.2	kg/m^3
f_r	Rolling resistance coefficient	0.01	-
g	Gravity acceleration	9.81	m/s^2
θ	Angle between the road surface and a horizontal plane	0	°
γ	Constant	$C_d \cdot A_f \cdot \rho$	kg
F_{air}	Air resistance force	$0.5 \cdot C_d \cdot A_f \cdot \rho \cdot v^2$	N
F_{roll}	Rolling resistance force	$M \cdot g \cdot f_r$	N
F_g	Gravity force	$M \cdot g \cdot \sin(\theta)$	N
F_{drive}	Driving force	-	N
Δx_i	Headway distance	-	m
x_{error}	Position error	-	m
v_{error}	Velocity error	-	m/s
v_{ss}	Steady state velocity	20	m/s
a_{ss}	Steady state acceleration	0	m/s^2
P	Proportional gain	650	-
I	Integral gain	9.4	-
D	Derivative gain	1720	-
C	Controller	-	-
s	Laplace complex variable	-	-
u	System input	-	N
q	System state	-	-
A	State-variable description matrix	-	-
B	State-variable description matrix	-	-
C	State-variable description matrix	-	-
D	State-variable description matrix	-	-
W	Disturbance matrix	-	-

Bibliography

Ashley, S.; "Smart cars and automated highways", The American Society of Mechanical Engineers (1998).

Baert, R.S.G., I.J.M. Besselink, C.C.M. Luijten, et al.; "Introduction to automotive technology", Eindhoven University of Technology, Department of Mechanical Engineering (2006).

Bengtsson, J.; "Adaptive cruise control and driver modelling", Department of Automatic Control, Lund Institute of Technology (2001).

BMW; "ACC – Active Cruise Control Seminar", BMW AG, München (2000)

Franklin, G.F., J.D. Powell, A. Emami-Naeini; "Feedback control of dynamic systems", fourth edition (2002).

Güvenç, B.A., E. Kural; "Adaptive cruise control simulator. A low-cost multiple-driver-in-the-loop simulator", Istanbul Technical University, Department of Mechanical Engineering (2006).

Hatipoğlu, C., Ü. Özgüner, M. Sommerville; "Longitudinal headway control of autonomous vehicles", Department of Electrical Engineering, Ohio State University (1996).

Higashimata A., K. Adachi, T. Hashizume, S. Tange; "Design of a headway distance control system for ACC", JSAE Review 22 (2001), pp. 15 – 22.

Holzmann H., Ch. Halfmann, S. Germann, M. Wurtenberger, R. Isermann; "Longitudinal and lateral control and supervision of autonomous intelligent vehicles", Control Eng. Practice Vol. 5 No. 11 (1997), pp. 1599-1605.

Jones, W.D.; "Keeping cars from crashing", IEEE Spectrum Volume 38, Issue 9 (September 2001), pp. 40 – 45.

Kok, J.J.; "Regelen van multivariabele systemen", Eindhoven University of Technology, Department of Mechanical Engineering (2002).

Liang, C-Y., H. Peng; "Optimal adaptive cruise control with guaranteed string stability", Vehicle System Dynamics (1999), pp. 313 – 330.

Osugi, K., K. Miyauchi, N. Furui, H. Miyakoshi; "Development of the scanning laser radar for ACC system", JSAE Review 20 (1999), pp. 549 – 554.

Pipes, L. A.; "An operational analysis of traffic dynamics", Journal of Applied Physics, Vol. 24, No.3 (1953), pp. 274 – 287.

Qui, H., Q. Zhang; "Feedforward-plus-proportional-integral-derivative controller for an off-road vehicle electrohydraulic steering system", Department of Agricultural Engineering, University of Illinois at Urbana-Champaign (2002).

Reuschel, A.; “Vehicle movements in a platoon with uniform acceleration or deceleration of the lead vehicle”, Zeitschrift des Oesterreichischen Ingenieur- und Architekten-Vereines, No.95 (1950), pp. 59 – 62 and 73 – 77.

Sheikholeslam, S., C.A. Desoer; “Longitudinal control of a platoon of vehicles. I: Linear Model”, Institute of Transportation Studies, University of California at Berkeley (1989).

Slocum, A.H., S. Awtar; “Simplified automated material handling system: magnetic wheels based overhead transportation concept”, MIT Confidential (2001).

Van Mieghem, R.; “Adaptive cruise control systemen in gemengd verkeer”, Civil Technology & Management, University of Twente (2004).

Weyns, D., M. Schumacher, A. Ricci, M. Viroli, T. Holvoet; “Environments for multiagent systems”, Report AgentLink, Technical Forum Group Ljubljana (February 2005).

Zhong, J.; “PID controller tuning: a short tutorial”, Mechanical Engineering, Purdue University (2006).