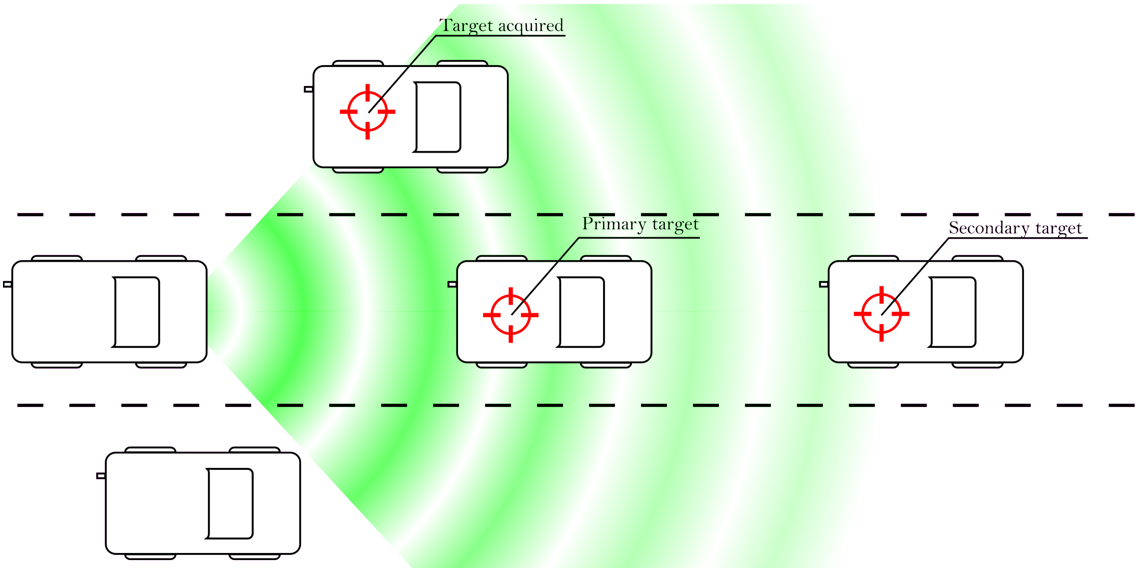
CHALMERS



Extended Adaptive Cruise Control based on multiple target information

Master of Science Thesis

JONAS KROOK
JAKOB LARSSON

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2013
Report No. EX062/2013

Extended Adaptive Cruise Control based on multiple target information
Jonas Krook and Jakob Larsson

Report No. EX062/2013
Department of Signals and Systems
Chalmers University of Technology
Gothenburg, 2013

SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-7721000

Abstract

Adaptive Cruise Control (ACC) is an improvement of regular cruise control that has been available in premium cars for over 10 years. With advanced sensing systems, based on for instance RADAR and camera sensors, ACC makes it possible for a vehicle to follow the preceding vehicle at a safe distance by automatically adjusting the velocity.

In this thesis, it is investigated whether more information from the environment may be used to improve the ACC. By using information from vehicles in adjacent lanes and vehicles ahead in the same lane, the ACC is extended with two new functions; Lateral prediction and Multi-target control. Lateral prediction is used to predict the future position of vehicles in the adjacent lanes. When a vehicle from an adjacent lane is predicted to join the same lane, the ACC equipped vehicle can change target in the control law faster. Multi-target control is introduced by letting the ACC use information from more than one preceding vehicle in the control law. With the extra information it is possible to earlier detect and react to traffic disturbances further ahead in the lane.

The extended ACC is evaluated with both simulations and in-vehicle tests and it is shown that it has advantages concerning driving comfort and safety over the standard ACC.

Key words: Adaptive Cruise Control, Multi-target control, Lateral prediction, Kalman filtering

Contents

Chapter 1

Introduction

Many vehicles today are equipped with sensors such as RADAR and camera. The output from these sensors is used by embedded electrical control units (ECU:s) in the vehicle to perceive the surrounding traffic situation. Based on this information, other ECU:s in the vehicle adapts the vehicle's speed by controlling the engine and the brakes. This opportunity is for instance used in active safety systems, which are able to control the vehicle speed according to traffic flow to avoid collisions.

1.1 Adaptive Cruise Control

The function that is able to control the vehicle speed according to traffic speed is commonly called Adaptive Cruise Control (ACC), and sometimes also called Intelligent Cruise Control (ICC). The ordinary cruise control, which the ACC is an extension of, is a function that is able to keep the vehicle at a certain set speed, usually determined by the driver. The set speed may also be determined automatically according to traffic regulations (Paine et al., 2007). The controller implementation of the cruise control is relatively simple, since the controller's objective is to regulate to zero difference between set speed and current speed.

The adaptive part of the ACC adds the possibility for the controller to keep a safety distance to the preceding vehicle (Kesting et al., 2007; Marsden et al., 2001). This kind of control requires knowledge about the preceding vehicle's speed and the distance to it. How this information is obtained may differ, but at least one of the quantities should be measured. Usually both are available to the ACC as measurements.

In this thesis the vehicle equipped with the ACC is referred to as Host, while the vehicle which is being followed is referred to as Target.

1.1.1 ACC with communication

As vehicles get more advanced, the number of possible ways of acquiring preceding vehicles' position and speed increases. By using communication links between vehicles, the information of position and speed can be relayed from Target to Host (Audi, 2010).

Additionally, data which has not been available earlier can be sent through the communication link. The preceding vehicle's requested acceleration, which is not directly measurable from Host, can be used as feed-forward contribution. Steering angle and yaw rate of the preceding vehicle can be used to get information on whether the vehicle stays in the same lane or not.

It is also possible to let the road infrastructure communicate with the vehicles, and, for instance, relay information of traffic disturbances further down the road.

1.2 Background

The work presented in this thesis is the result of a collaboration with Volvo Car Corporation (VCC), whose current ACC implementation is the basis in the following analysis and discussion. Some of the statements about ACC in the following text may apply only to this specific implementation, but the aim is to keep the presentation of this thesis work as general as possible.

The current ACC implementation has the following limitations:

- Normally, implementations of ACC:s utilize only the information of the closest preceding vehicle for controlling the speed. When the traffic flow is smooth this method of control behaves well, but in dense traffic where the speed varies much and vehicles change lanes, situations may arise which the ordinary ACC has trouble handling efficiently.

- If a vehicle moves into the same lane as Host, a human driver is often able to identify the lane change before the vehicle completes the maneuvre. Therefore, the Host driver may adjust the distance to the lane-changing vehicle in advance to provide sufficient space. However, often in ACC:s, lane-changing vehicles are not considered as the new Target until completely within the lane. If the distance to the lane-changing vehicle is smaller than the preset safety distance, the ACC may have to decelerate abruptly when the lane-change has been completed.

- In dense traffic, the speed of the traffic in front of Target limits the speed of Host. The inability of the ACC to adjust to lane changes and traffic speed further down the road means that the driver may perceive the ACC as fitful and that it is slow to react.

To extend the ACC capabilities, this thesis will evaluate the possibilities to consider more vehicles, in addition to the immediate preceding vehicle, when controlling the speed. The problem consists of recognizing vehicles which are about to change lanes and to let information about the vehicle in front of Target influence the ACC.

1.2.1 Lateral traffic disturbances

There are two different lateral disturbances that are important for the behaviour of the ACC:

- The first is the case where a vehicle in an adjacent lane drives into Host's lane and becomes Target. This is called a cut-in and causes the ACC to

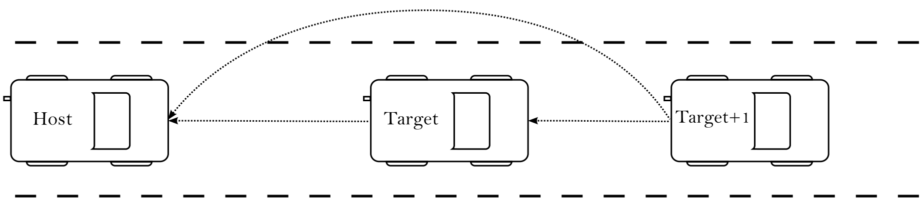


Figure 1.1: Overview of the multi-target ACC equipped vehicle, Host, which uses information about the two immediate preceding vehicles, Target and Target+1, in the control law.

brake if the distance to the cutting-in vehicle is small or if it is approached fast, i.e., it has a lower velocity than Host. Although the ACC is capable of braking and keeping the safety distance to the cutting-in vehicle, the braking does not begin until the cutting-in vehicle is entirely in Host's lane and selected as new Target by the ACC.

- The second lane change of interest is when Target leaves Host's lane, which is called a cut-out. In this case, the ACC should accelerate to either the set speed, or to the safety distance to the new Target. This acceleration should commence when the cut-out is apparent, but the ACC will not recognize the cut-out until Target has left the lane completely and no longer is selected as Target.

Predicting the lateral trajectories of the cutting in/out vehicles may improve the response of the ACC.

1.2.2 Longitudinal traffic disturbances

By using information about the vehicle ahead of Target, called Target+1 (see Figure 1.1), the ACC may react to longitudinal acceleration disturbances introduced by the traffic ahead earlier, and thus improve the responsiveness of the ACC. In certain traffic situations, Target mimics Target+1 and the behaviour of Target+1 can be used as a preview in the ACC.

1.3 Purpose

The purpose of this thesis is to investigate how lateral and longitudinal disturbances in Sections 1.2.1 and 1.2.2 can be compensated for, by using information from more vehicles than only Target, and what effects this has on the behaviour of the ACC, with respect to safety and comfort.

In the cut-in and cut-out situations the ACC may appear slow since a lane change often is apparent to the driver before the ACC takes the corresponding action. In this thesis it will be evaluated if the vehicles' future lateral position can be predicted and used as a mean to recognize possible lane changes before they are completed.

The goal of the lateral prediction is to make the ACC react faster in cut-in and cut-out situations, and to decrease the brake force applied in cut-in situations.

Furthermore, this thesis will study whether and how the information from Target+1 can be used to better anticipate the traffic ahead. The study consists of two parts, where the first part compares two linear controllers; one with information only about Target and one with information about both Target and Target+1. The goal is to evaluate whether the controller with Target+1 information responds quicker, and thus assessing whether the Target+1 information can be used to anticipate traffic ahead. The second part investigates whether the information of Target+1 successfully can be implemented in the existing Volvo ACC.

An ACC with only information of Target will be denoted single-target ACC, while an ACC with the additional information of Target+1 will be denoted multi-target ACC.

As more vehicles are equipped with ACC the chances of a situation where several vehicles with ACC drives after each other increases, commonly called a vehicle string. In such a case it is important that the ACC's exhibit a property called string stability. This means that inter-vehicle distance perturbations have to be attenuated by the vehicle string. It is desired that the string stability properties of the multi-target ACC are compared to the same properties of a single-target ACC.

1.4 Objectives

The objectives of this thesis for the lateral prediction are:

- **Construct a motion model** – In order to predict the lateral position of preceding vehicles, a model is needed that describes the expected lateral behaviour of a vehicle at a given time.

- **Estimate lateral motion** – An algorithm should be developed that uses the sensor measurements and the motion model to calculate an estimate of the current lateral motion of a vehicle.

- **Predict future lateral position** – By combining the estimated lateral motion and the motion model, a procedure has to be created to calculate a prediction of the lateral position of a vehicle t_H seconds ahead. The predicted lateral position should be used as a mean for detecting lane changing vehicles earlier.

- **Implementation** – The three above mentioned parts should be implemented in a VCC vehicle's hardware environment. The implementation should be tested in real traffic to evaluate how the prediction mitigates lateral traffic disturbances.

The objectives of this thesis for the multi-target ACC are:

- **Extend ACC** – A controller structure for using both Target and Target+1 as basis for speed control should be constructed. The controller should be able to calculate a desired acceleration based on sensor measurement input from both Target and Target+1.

- **Stability study and analysis of multi-target ACC** – The effect of Target+1 should not cause oscillations or unbounded errors. It should be determined whether the extended controller structure makes a string of vehicles stable.
- **Comparison of single-target and multi-target ACC** – The multi-target ACC should be compared to the single-target ACC to find out in which situation it is beneficial to use Target+1 information and in which situation the information will worsen the performance.
- **Implementation** – The controller structure for controlling with respect to Target+1 should be implemented and integrated with the VCC single-target ACC in the vehicle's hardware environment. The results from the comparison of a single-target and multi-target ACC should be used to guide the implementation of the multi-target ACC in order for it to attenuate longitudinal traffic disturbances.

1.5 Delimitations

The thesis has the following delimitations:

- The lateral prediction will only be evaluated on straight road segments, due to limitations in the current Target acquisition and Target tracking.
- Only longitudinal control will be investigated for the multi-target ACC.
- Experimental testing will only be made with VCC vehicles.

1.6 Outline

In Chapter 2, the lateral prediction is described. Theory for estimating and predicting future position is presented which is used to extend the Target selection algorithm in the VCC ACC with lateral prediction. The extended Target selection algorithm is evaluated by comparing it to the existing Target selection algorithm, using both recorded vehicle data and in-vehicle tests.

The multi-target ACC part of the thesis is divided into two sub parts. In Chapter 3, theory about spacing policies and string stability is given together with string stability analysis and an optimal control analysis, where multi-target linear control method is compared to single-target control. The result from the analysis is used in Chapter 4 to extend the existing VCC single-target ACC to a multi-target ACC. The developed multi-target ACC is designed and tuned in a simulation environment and then evaluated with in-vehicle tests.

Conclusions from the two main parts are presented in Chapter 5 together with an outline of future work.

Chapter 2

Lateral prediction

The ACC system relies on four main subsystems as described in Figure 2.1. These are the sensing modules, the Target selection algorithm, the controller and the actuators. The sensor modules and the actuators are shared among several other systems in the car, but the Target selection and the controller are only used in the ACC system. Two sensor modules are used by the ACC. One sensor module provides the Target selection with a list of detected vehicles in front of Host. The information which is sent from the sensor module to the Target selection includes relative distance, r, relative speed, $\dot{r}$, and relative angle, ϕ, as shown in Figure 2.2. The second sensor module provides Host's yaw rate, velocity and acceleration.

The information sent to the Target selection is used to designate vehicles as either Target or not. This is done by estimating the Host's current route, $f(p)$ in Figure 2.2. The closest vehicle in the lane is designated as Target and sent to the controller. If no such vehicle exists, the ACC maintains the set speed.

The controller uses the information from the sensor modules to estimate a longitudinal range and relative longitudinal velocity to Target. In Figure 2.2 the longitudinal range is represented by x. The controller also estimates Target's absolute acceleration. The estimations of range, range-rate and acceleration are used as input values to calculate a desired acceleration. The desired acceleration is sent to the actuators where the acceleration of Host is controlled.

The prediction of the lateral position, henceforth called lateral prediction, modifies the sensor values which are used in the Target selection, as can be seen in Figure 2.3. This will affect which vehicle that is selected, without any need to change the implementation of the Target selection.

This chapter first explains the theory behind the lateral prediction and how

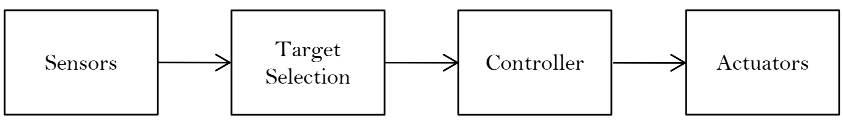


Figure 2.1: System overview of the ACC.

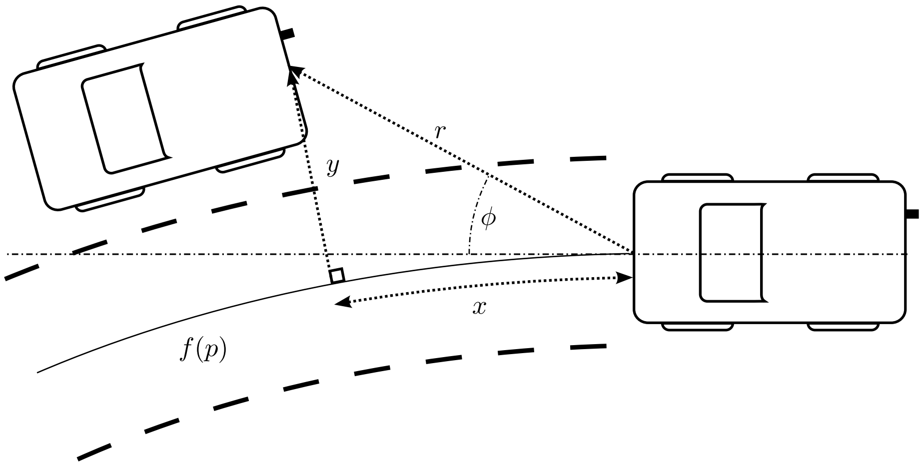


Figure 2.2: The relation of y, x, r, ϕ and $f(t)$.

Figure 2.3: System overview of the ACC with lateral prediction.

it is implemented. The Target selection and the lateral prediction are then combined and tested in simulations. The results from the simulations are used to extend the ACC in a real car with lateral prediction.

2.1 Target model

In order to accurately estimate vehicles' lateral position and velocity, a model for the vehicles' lateral driving dynamics is needed. A complete model of the lateral driving dynamics for a vehicle includes more information than the lateral position of the vehicle. An important part is its steering angle, which in general is not measurable by Host's sensor modules. However, when a detected vehicle changes lanes, a simple assumption is to expect its lateral acceleration to be constant. This can be expressed as

$$\dot{a}_y = 0,$$

where a_y is the lateral acceleration. Moreover

$$a_y = \dot{v}_y \quad \text{and} \quad v_y = \dot{y},$$

where v_y is the lateral velocity and y is the lateral position which is calculated from r and ϕ in Figure 2.2. Here, y is defined such that zero is the center of Host and positive values mean to the left of Host's center. The state space form of the lateral dynamics becomes

$$\dot{\gamma}(t) = \begin{bmatrix} \dot{y}(t) \\ \dot{v}_y(t) \\ \dot{a}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \gamma(t) = A\gamma(t) \tag{2.1}$$

$$z(t) = \begin{bmatrix} y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \gamma(t) = C\gamma(t), \tag{2.2}$$

where z is the measured lateral position.

By using a Kalman-filter the idea is to combine (2.1) and (2.2), to get an estimate of the lateral position that takes both the model and the measurements into account.

2.2 Kalman filter theory

In this section a basic introduction of Kalman filter theory for discrete time is presented. For further and deeper information about Kalman filters, refer to (Åström and Wittenmark, 1997).

A basic prerequisite for constructing a Kalman filter is a system in state-space form

$$x(k+1) = Ax(k) + Bu(k) + Nv_1(k) \tag{2.3}$$

$$y(k) = Cx(k) + Du(k) + v_2(k), \tag{2.4}$$

where x is the state of the system, y is the measurements, u is the external input, and v_1 and v_2 are model and measurement noise, respectively. In the ideal case all states are measurable and the model, as well as the measurements, are noiseless.

In a typical real world situation this ideal assumptions are not satisfied. The model is usually not an exact mathematical representation of the real world system and the measurements are in general affected by noise. Also, not all states are measurable. This is a problem because knowledge of the non-measurable states is desired. To address this problem an observer is used to estimate the state of the system. The state estimate, $\hat{x}$, has the same dynamics as x in (2.3), but without the noise term v_1. The state estimate will be calculated in each time step by correcting the a priori estimate, $\hat{x}(k, k-1)$, with the a priori estimate error, $y(k) - C\hat{x}(k|k-1) - Du(k)$.

In an ideal situation, $y(k)$ and $C\hat{x}(k|k-1) + Du(k)$ is equal, but because of the model and measurement noise they will differ. The difference, $y(k) - C\hat{x}(k|k-1) - Du(k)$, is used as a feedback term to improve the accuracy of the state estimate. Feedback with gain K results in the equation

$$\hat{x}(k+1|k) = A\hat{x}(k|k-1) + Bu(k) + K(y(k) - C\hat{x}(k|k-1) - Du(k)), \tag{2.5}$$

which is an observer of the system.

The objective of the observer is to produce an estimate $\hat{x}$ which is close to the true state x. This implies that the estimation error

$$\tilde{x}(k) = x(k) - \hat{x}(k),$$

should be small. From (2.3) and (2.5) the model of the estimation error can be written as

$$\tilde{x}(k+1|k) = (A - KC)\tilde{x}(k|k-1) + Nv_1(k) - Kv_2(k). \tag{2.6}$$

Thus, the choice of K affects the behaviour of the estimation error in two ways. $A - KC$ determines how fast old estimation errors, $\tilde{x}(k|k-1)$, vanishes, while $Kv_2(k)$ determines how much the measurement noise, v_2, affects the estimation error. K should be chosen such that $A - KC$ is stable with the eigenvalues in the stability region. At the same time K should not amplify the noise much. The ideal case would be to have all eigenvalues of $A - KC$ to be zero and K equal to zero. In that way all estimation errors vanishes after one time step, and the measurement noise would not affect the estimations. However, with such system the observer would not be necessary at all.

The existence of a K such that $A - KC$ is stable requires that (A, C) is detectable. If such K exists, the problem of how stability of $A - KC$ and sensitivity to measurement noise should be weighted depends on v_1 and v_2. If the variance of v_1 is much larger than v_2, it is likely that it is more important that estimation errors vanishes than that the measurement noise is attenuated.

The covariance of the estimation error is expressed as

$$P = E[\tilde{x}(k)\tilde{x}(k)^T].$$

By minimizing P, an optimal filter with respect to estimation error covariance is acquired. The covariance of the estimation error is driven by the gaussian white noise v_1 and v_2. By (2.6), theorem 5.6 in Glad and Ljung (2000), and since $\tilde{x}(k|k-1)$ is independent of $v_1(k)$ and $v_2(k)$, the variance can be calculated as

$$\begin{aligned} P &= E[\tilde{x}(k|k-1)\tilde{x}(k|k-1)^T] \\ &= APA^T + R_1 - (APC^T + R_{12}) \\ &\quad \cdot (R_2 + CPC^T)^{-1} \\ &\quad \cdot (CPA^T + R_{12})^T, \end{aligned} \tag{2.7}$$

where $R_1 = E[v_1 v_1^T]$, $R_2 = E[v_2 v_2^T]$ and $R_{12} = E[v_1 v_2^T]$ are the variance of v_1 and v_2, and their cross variance, respectively. The optimal observer gain is calculated by $K = (APC^T + R_{12})(CPC^T + R_2)^{-1}$, and used in (2.5) it forms the Kalman filter (Glad and Ljung, 2000).

The state estimate available in each time step is completely calculated using information from previous time steps. This means that the new information in a time step is not used for the current time state estimate. To use the latest information available in a state estimate, the following update step is used:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \tilde{K}(y(k) - C\hat{x}(k|k-1) - Du(k)) \tag{2.8}$$

$$\hat{x}(k+1|k) = \hat{x}(k|k) + Bu(k), \tag{2.9}$$

where $\tilde{K} = PC^T(CPC^T + R_2)^{-1}$.

2.3 Target prediction

The Target selection is realized in a computer controlled system, which is a discrete time system. This necessitates a discrete time model of the lateral vehicle dynamics and a discrete time observer. With the sample time h the model (2.1)-(2.2) is discretized by using the second order Taylor polynomial:

$$\gamma(k+1) = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \gamma(k) = \Phi\gamma(k) \tag{2.10}$$

$$z(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \gamma(k) = C\gamma(k). \tag{2.11}$$

To calculate a discrete time state estimate, equations (2.8) and (2.9) are used in conjunction with (2.10) and (2.11) to form a discrete time Kalman filter for the estimation of the lateral position, velocity and acceleration:

$$\hat{\gamma}(k|k) = \hat{\gamma}(k|k-1) + \tilde{K}(z(k) - C\hat{\gamma}(k|k-1)) \tag{2.12}$$

$$\hat{\gamma}(k+1|k) = \Phi\hat{\gamma}(k|k), \tag{2.13}$$

where

$$\tilde{K}(k) = (\Phi P(k|k-1)C^T)(R_2 + CP(k|k-1)C^T)^{-1}, \tag{2.14}$$

and

$$\begin{aligned} P(k+1|k) &= \Phi P(k|k-1)\Phi^T + R_1 \\ &\quad - (\Phi P(k|k-1)C^T)(R_2 + CP(k|k-1)C^T)^{-1}(CP(k|k-1)A^T). \end{aligned} \tag{2.15}$$

The Kalman filter only estimates the current state $\hat{\gamma}(k|k)$. To make a prediction one step ahead, (2.10) is applied to calculate $\hat{\gamma}(k+1|k)$.

A predicted value n steps ahead is achieved by applying the model (2.10) n times

$$\begin{aligned} \hat{\gamma}(k+n|k) &= \Phi\hat{\gamma}(k+n-1|k) \\ &= \Phi^2\hat{\gamma}(k+n-2|k) \\ &\vdots \\ &= \Phi^{n-1}\hat{\gamma}(k+1|k) \\ &= \Phi^n\hat{\gamma}(k|k) \\ &= \begin{bmatrix} 1 & nh & \frac{1}{2}n^2h^2 \\ 0 & 1 & nh \\ 0 & 0 & 1 \end{bmatrix} \hat{\gamma}(k|k). \end{aligned} \tag{2.16}$$

Since n is the amount of time steps that are predicted and h is the sample time,

$$t_H = nh,$$

is the total time predicted, and will be referred to as the prediction horizon.

2.4 Tuning the filter

For the determination of $\tilde{K}$ in the Kalman filter, (2.12), knowledge of R_1, R_2 and R_{12} in (2.15) is required. In this particular case, the variance of the measurement noise, R_2, is scalar since only one state is measured. Thus, R_2 is simply the variance of the lateral position measurement noise.

The covariance of the unmodelled dynamics is defined by

$$R_1 = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \sigma_{13}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \sigma_{23}^2 \\ \sigma_{31}^2 & \sigma_{32}^2 & \sigma_{33}^2 \end{bmatrix}, \tag{2.17}$$

where the values on the diagonal are the variance of the lateral position, lateral velocity and lateral acceleration noises, respectively. The off-diagonal values are the covariance of the lateral position noise, lateral velocity noise and lateral acceleration noise. Since $\sigma_{ij}^2 = \sigma_{ji}^2$, there are six parameters to be determined for R_1.

Since R_1 and R_2 are unknown, they are considered as tuning parameters.

Analysis of logged sensor data shows that most lane changes takes between one and three seconds to complete. Thus, the prediction horizon, t_H, is constrained by these values. Larger horizons would not give any advantage since most lane changes are not possible to predict more than three seconds ahead.

The upper bound on the prediction horizon is also determined by the amount of noise amplification which a large time horizon introduces. A small amplification of noise on the lateral acceleration could have large effect on the prediction. In order to get a time horizon that is both significant for predicting lane changes accurately and which does not introduce too much noise amplification, the path taken in this thesis is to set the prediction horizon to two seconds; the average time of a lane change.

2.4.1 Variance effect on prediction

The variances σ_{11}^2, σ_{22}^2, σ_{33}^2 from (2.17), and R_2 compose the four design parameters of the Kalman filter, and they are used for tuning the lateral predictor. The tuning is done by testing different parameter sets on a range of scenarios.

If the variance of the model noise of the acceleration, σ_{33}^2, is low compared to the variance of the model noise of the position and velocity, σ_{11}^2 and σ_{22}^2, e.g.

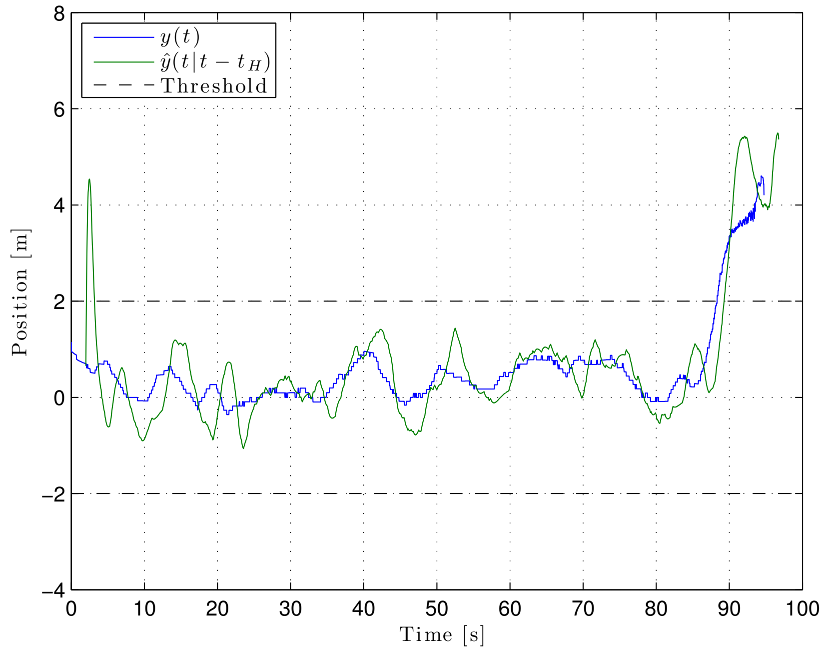
$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.05 \end{bmatrix}, \qquad R_2 = 500, \tag{2.18}$$

then the predicted acceleration is slowly varying, since the acceleration is assumed constant. In Figure 2.4 the effect of (2.18) on the prediction of lateral position is shown for a representative scenario. The blue line in 2.4a is the lateral position measured by the sensors at time t, i.e. $y(t)$. The green line, $\hat{y}(t|t - t_H)$, is the prediction of y at time t, given measurements up to time $t - t_H$. In 2.4b the prediction error can be seen. The predicted position errors are for the most part not larger than a meter, but at the lane change, which is initiated at 85 seconds, the assumption of low σ_{33}^2 leads to low responsiveness of the predictor output. As can be seen at the lane change at 89 s, the prediction crosses the threshold one second after the measurement. Since the prediction horizon is 2 seconds, the cut-out will be detected one second ahead when using lateral prediction.

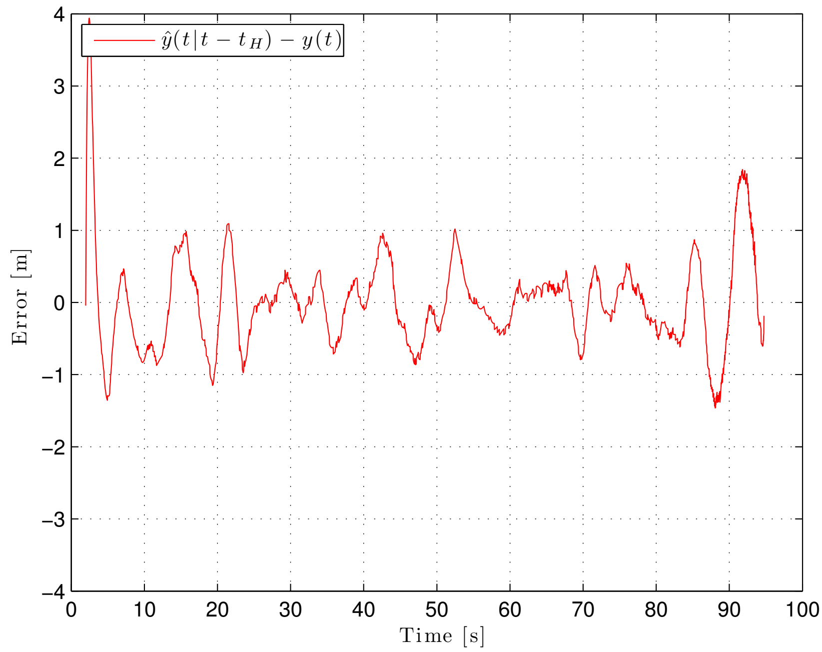
When σ_{33}^2 is high compared to σ_{11}^2 and σ_{22}^2, e.g.

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 500 \end{bmatrix}, \qquad R_2 = 500, \tag{2.19}$$

then the prediction error is very large over all time, and can be as much as 3 meters, but the responsiveness of the predictor is high at the moment of the

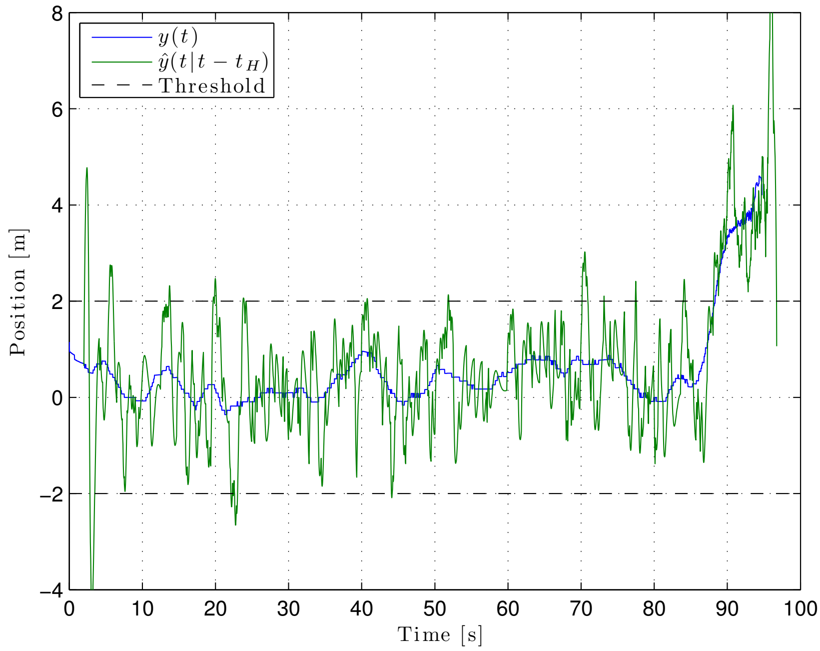


(a) Measured and predicted position.

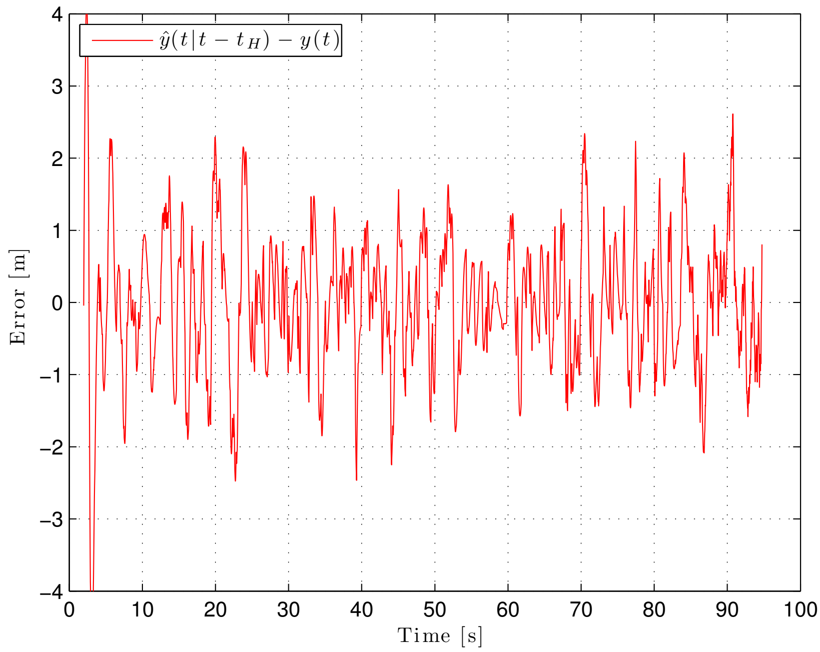


(b) Prediction error.

Figure 2.4: A cut-out situation where the assumed acceleration variance is too low, $R_1(3,3) = 0.05$. The estimated acceleration is then stiff to changes, which leads to smooth prediction and a delay of the prediction. $t_H = 2$.



(a) Measured and predicted position.



(b) Prediction error.

Figure 2.5: A cut-out situation where the assumed acceleration variance is too high, $R_1(3,3) = 500$. The estimated acceleration is then allowed to change a lot, which leads to noisy prediction and larger errors. $t_H = 2$.

lane change, as shown in Figure 2.5. The high variance of σ_{33}^2 also gives rise to large overshoots when the lane change is completed. The detected vehicle is predicted to a position almost two lanes away. At a cut-out, the overshoot does not introduce errors in the Target selection. On the contrary, an overshoot of the lateral position during a cut-in may cause errors if the detected vehicle is predicted across the Host's lane.

The effect of the measurement noise variance is the opposite of the process noise variance of the acceleration; low values of R_2 imply large errors and higher responsiveness of the predictor , while large values of R_2 means smaller errors but lower responsiveness of the predictor . Figures 2.6 and 2.7 shows what happens when the measurement variance is too low and too high, respectively.

The design parameters of the Kalman filter R_1 and R_2 have finally been chosen as

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \qquad R_2 = 500. \tag{2.20}$$

This means that the measurement noise variance is a lot larger than the variance of any of the modelling errors, and that the assumption of constant lateral acceleration is more uncertain than the assumptions on lateral position and velocity.

The chosen parameter configuration is a compromise to get acceptable prediction errors over time and to get a high responsiveness when the detected vehicle changes lanes. In Figure 2.8 it can be seen that the predicted and the measured lateral position both cross the threshold at the same point, i.e., the lane change will be detected two seconds in advance.

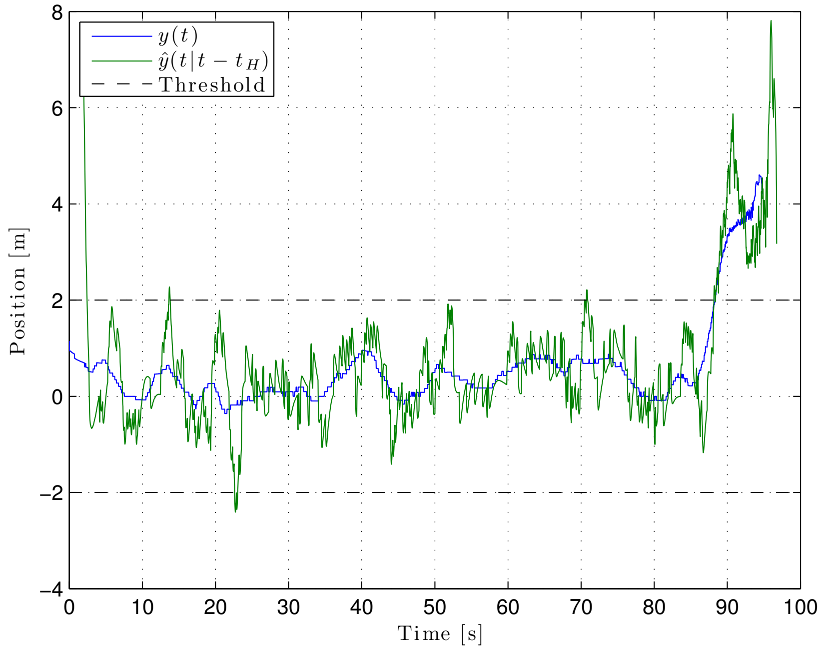
2.4.2 Limiting overshoot of predicted position

Shortly after 90 seconds in Figure 2.8 , the prediction error is more than 2 meters. This error appears directly after the cut-out and it is a consequence of the high estimated lateral velocity and acceleration in $\hat{\gamma}$ in (2.12) during the cut-out.

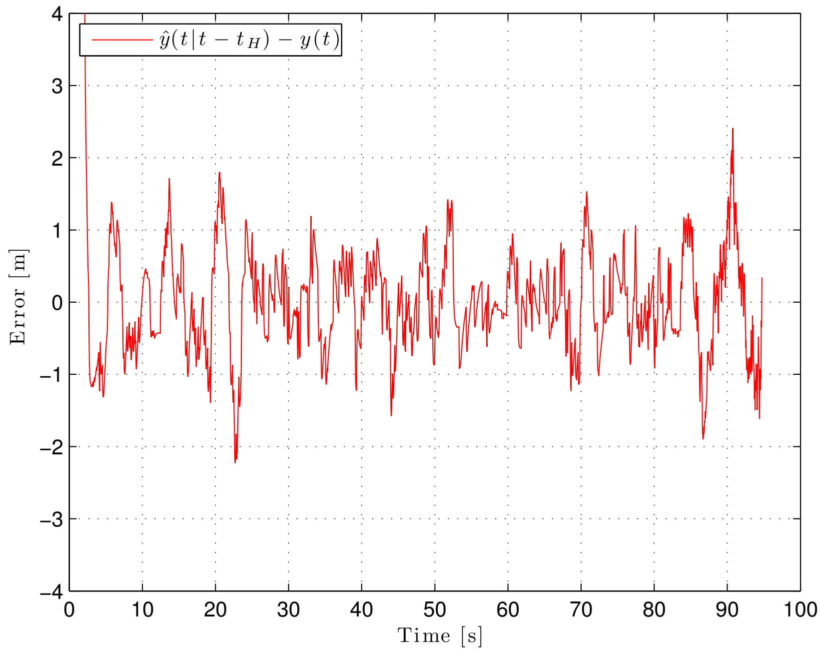
The assumption that the lateral acceleration is constant is not realistic, since with a constant non-zero acceleration, Target would soon leave the road. To address this issue the variance of the model noise for the acceleration could be increased. In that way the acceleration is expected to deviate much more from the model (2.10). Increasing the variance does decrease the total time of the overshoot, but not its magnitude. In addition it introduces high magnitude noise in regular driving as can be seen in Figure 2.5 at 70 seconds. After the initial velocity increase caused by driver steering, the acceleration should tend to zero during the cut-out or cut-in. This indicates that the model may need adjustments. Thus, the dynamics of the acceleration in (2.10) is changed to

$$\hat{a}(k+1) = \rho\hat{a}(k),$$

where $0 \leq \rho \leq 1$. The effect of setting ρ to 0.975 can be seen in Figure 2.9. The magnitude of errors decreases with almost a meter compared to when ρ is 1 (Figure 2.8). A side effect is that the responsiveness of the predictor drops as ρ decreases, since the estimated lateral acceleration is lower for decreasing ρ.

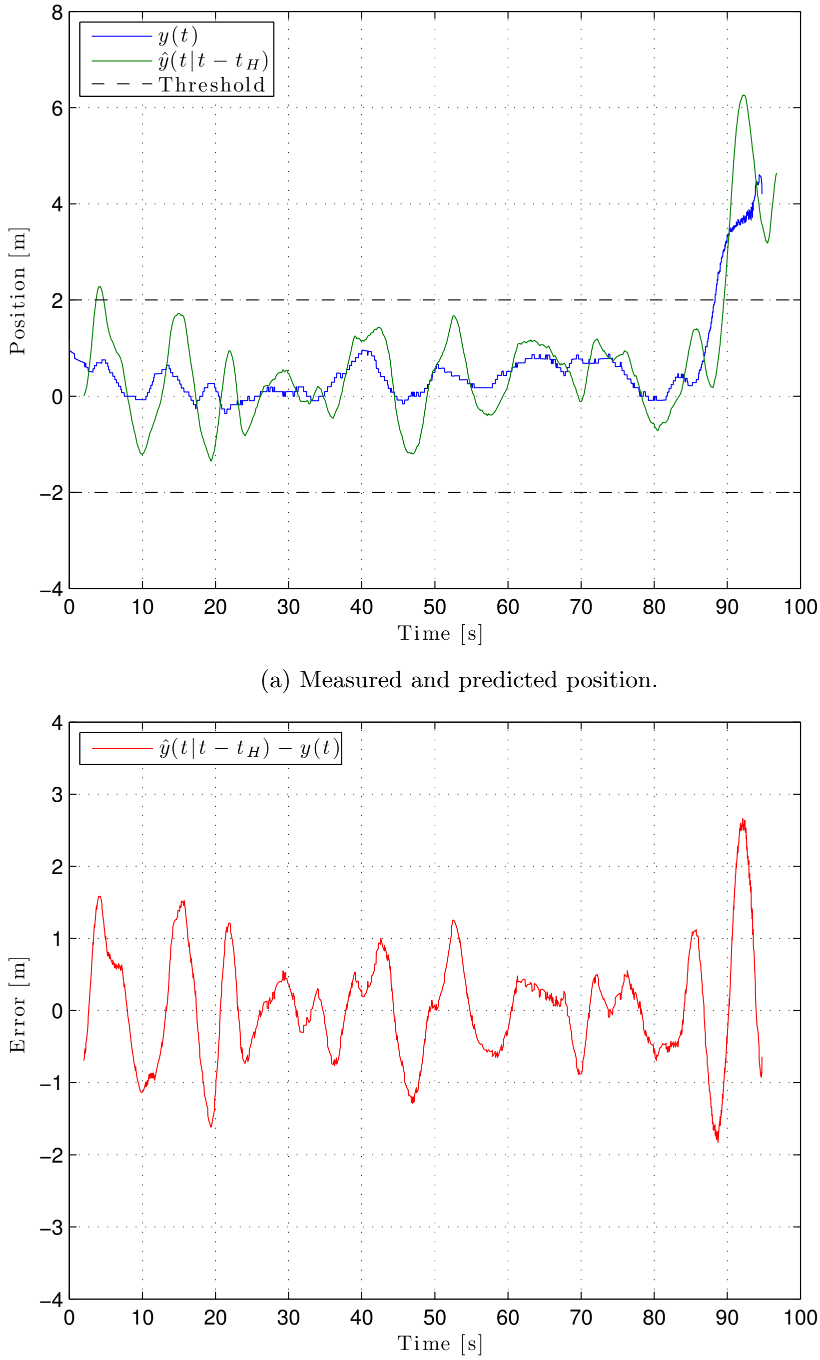


(a) Measured and predicted position.



(b) Prediction error.

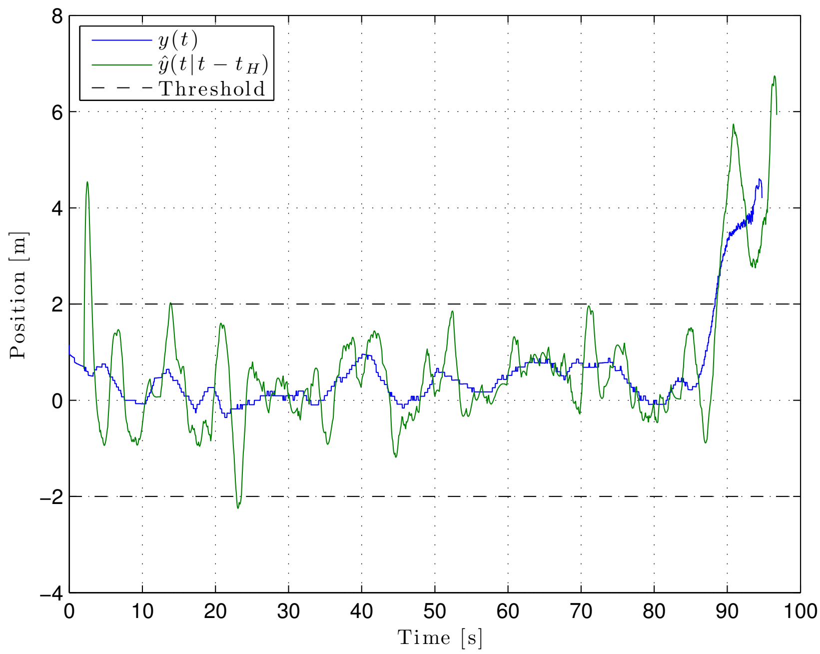
Figure 2.6: A cut-out situation where the assumed measurement variance is set too low, $R_2 = 5$. The prediction becomes noisy and has larger errors. $t_H = 2$.



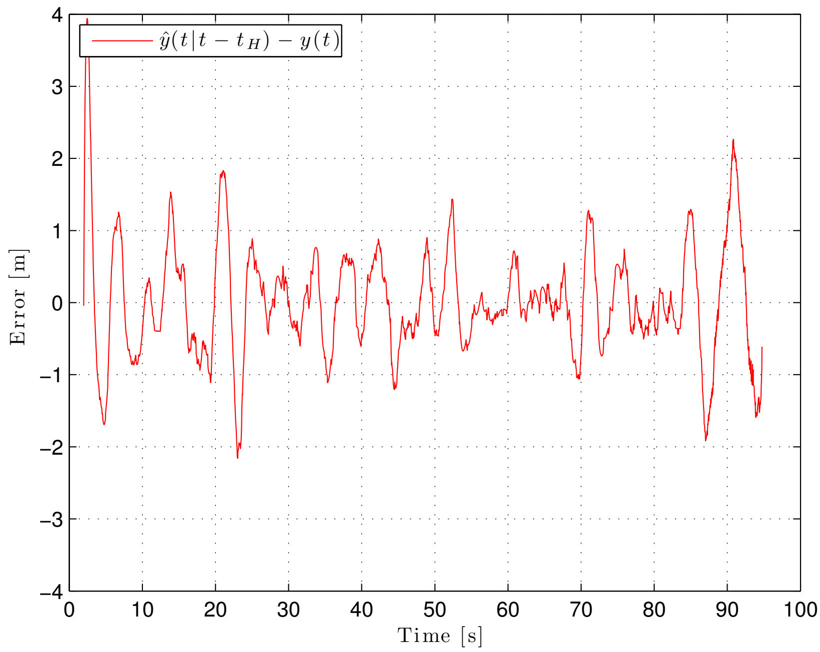
(a) Measured and predicted position.

(b) Prediction error.

Figure 2.7: A cut-out situation where the assumed measurement variance is set too high, $R_2 = 50000$. The prediction becomes smooth and delayed. $t_H = 2$.



(a) Measured and predicted position.



(b) Prediction error.

Figure 2.8: The prediction output in a cut-out situation. $t_H = 2$.

With the modified model, the cut-out is detected 1.2 seconds in advance if the prediction is used instead of the measurement.

2.5 Results

The simulation results show that with the selected tuning in (2.20), the cut-in and cut-out situations could in some cases be detected up to two seconds ahead of the standard Target selection module (i.e, without predictor). The smallest prediction times that are attained are half a second. Increasing the prediction horizon increases the swiftness of the Target selection, but at a prediction horizon larger than 2 seconds the effect begins to saturate.

When selecting a prediction horizon, the attained prediction time needs to be evaluated with respect to accuracy.

The accuracy is defined as

$$A \triangleq \frac{\int_{t=0}^{t} \delta(\hat{y}(t+t_H|t), t)}{\int_{t=0}^{t} 1}$$

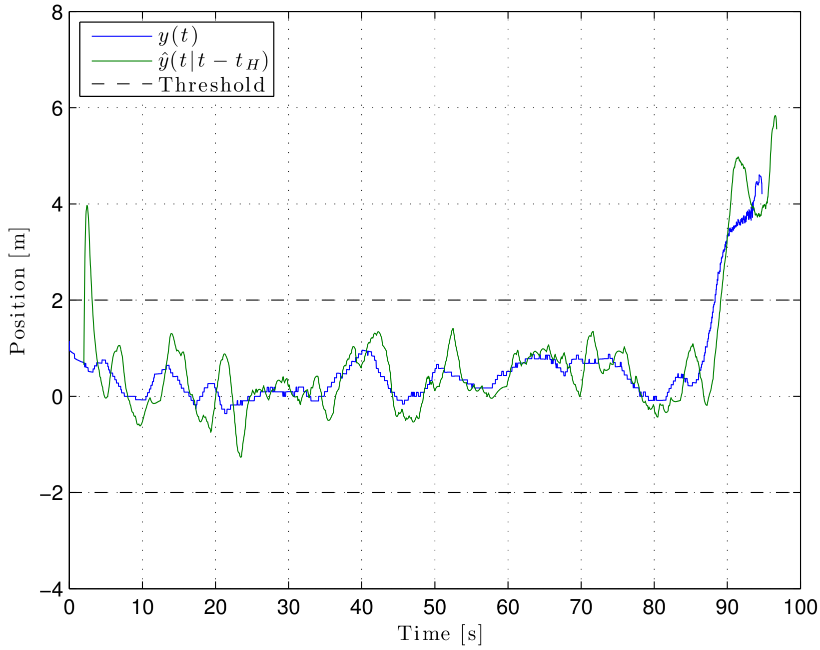
where

$$\delta(y,t) = \begin{cases} 1, & \text{if } y \text{ is in the correct lane at time } t \\ 0, & \text{otherwise.} \end{cases}$$

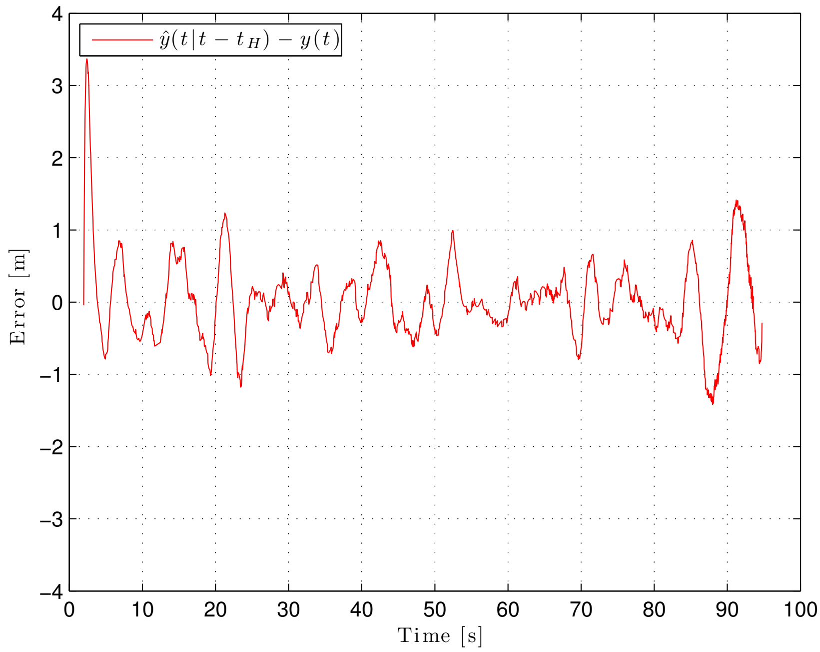
When no lane change is ongoing, the measured position determines the correct lane of Target. During the lane change, both lanes are considered correct. However, $\hat{y}(t+t_H|t)$ may only change lanes once during the lane change. All additional changes is considered as errors.

In Figure 2.10, the average attained prediction time, i.e., how much faster the lane change is detected with the lateral prediction than the standard Target selection, is plotted with respect to accuracy for four different parameter settings. Points to the upper right are desirable, which means high accuracy and high attained prediction time. From the figure, it is evident that the difference between the different parameters in the set $\rho \in \{1, 0.975, 0\}$ is small. However, for a given prediction horizon, lower values of ρ give higher accuracy and lower attained prediction time. The same effect may be acquired by lowering the prediction horizon. Increasing the measurement noise variance in the Kalman filter, e.g. setting $R_2 = 5000$, decreases the performance. This is mostly due to reduced responsiveness of the lateral predictor and not worse accuracy.

Results from tests conducted in vehicles are harder to evaluate than the results obtained from the simulations, but the improvement measured from log files are around one second with a prediction horizon of two seconds. The vehicle tests were camera recorded, and screen captured pictures show that without the predictor a vehicle is selected as Target when three wheels have crossed the lane markings, while the predictor lets the Target selection designate the vehicle as Target when only one wheel has crossed the lane markings. This can be seen in Figure 2.11. The first figure shows the position of the vehicle when the Target selection designates the vehicle as Target without lateral prediction. The second picture shows the same situation with lateral prediction. It is easy to see that lateral prediction makes a positive difference.



(a) Measured and predicted position.



(b) Prediction error.

Figure 2.9: The prediction output in a cut-out situation. $t_H = 2$ and $\rho = 0.975$. $t_H = 2$.

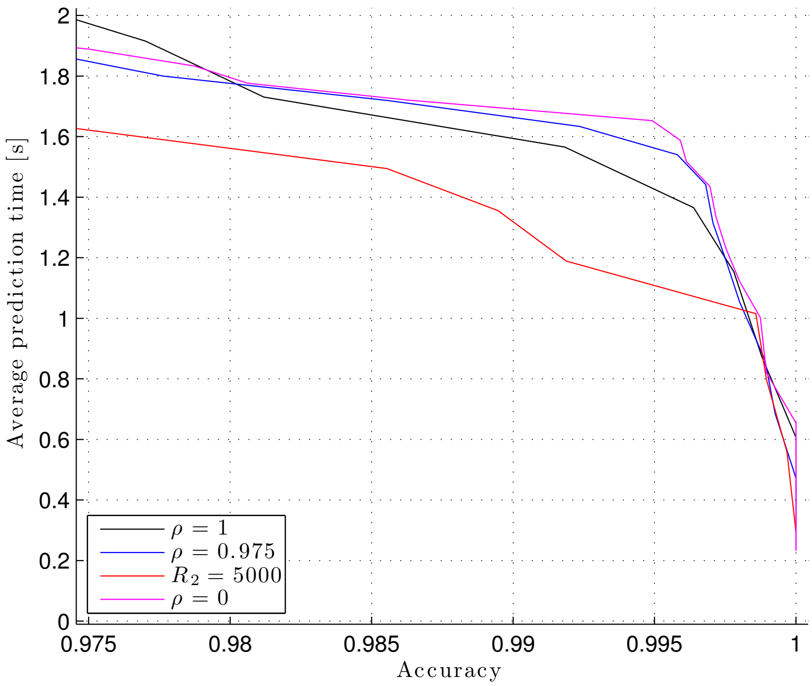


Figure 2.10: The average attained prediction time with respect to accuracy.

(a) Without the Kalman predictor.

(b) With the Kalman predictor.

Figure 2.11: The pictures depicts a cut-in situation from the left lane. The pictures are captured at the instant the Target selection designates the vehicle as Target. $t_H = 2$.

Chapter 3

Multi-target control

Typically, in today's vehicles the Adaptive Cruise Control only uses single-target control. This means that it only uses information about the immediate preceding vehicle, Target, in the control law. The information that is used in the control law is mainly the range, which is the distance between Host and Target, and the range-rate, which is the difference of velocity between Host and Target, i.e., the derivative of the range. The acceleration of Target is also often used in the control law.

A driver using ACC will in many situations experience that the vehicle reacts slowly when vehicles ahead accelerates or decelerates, which largely is due to time constants and delays in the propulsion and brake system of the vehicle. One way to improve the ACC would be to use a multi-target controller. Instead of only using information of Target, the ACC can also use information from the second closest preceding vehicle, Target+1, as shown in Figure 3.1. By using information from more than one vehicle ahead the goal is to react faster to traffic disturbances and thus decrease the range error and range-rate error, as well as the acceleration, compared to a corresponding single-target controller.

In this chapter the effects of introducing a multi-target controller is evaluated with respect to stability and performance, by comparing it with a corresponding single-target controller. In the analysis it is assumed that Target follows Target+1 in a well-defined way. This is of course not true in reality, since it can never be guaranteed that Target follows Target+1. In Chapter 4 the multi-target controller is modified to also handle situations when Target does not follow Target+1.

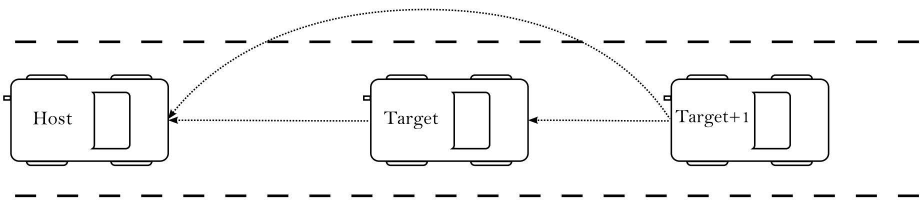


Figure 3.1: System overview of the multi-target ACC equipped vehicle, Host, which uses information about the two immediate preceding vehicles, Target and Target+1, in the control law.

3.1 Vehicle platooning

An emerging research area in the automotive industry is vehicle platooning, where several vehicles drive together in a string. The goal is to achieve small inter-vehicle distances and making the vehicles less sensitive to traffic shock-waves by letting the vehicles exchange information wirelessly.

There are several expected benefits of vehicle platooning. Traffic throughput on roads is expected to increase (Robinson et al., 2010). The aerodynamic drag is expected to reduce because of the small inter-vehicle distance, leading to lower fuel consumption. Since the vehicles in a platoon are assumed to be autonomously driven, the driver do not need to control the vehicle, but just supervise the autonomous driving systems, thus reducing the driving workload.

One key difference between vehicle platooning and a commercial available ACC is in what way the vehicles follow each other. In the platooning case the expected behaviour of the vehicles is known, i.e., the vehicles should follow each other. This is not the case in regular traffic, where Target+1 for example can drive faster than the set speed of Target.

Both the longitudinal control and the lateral control of the vehicles is important in vehicle platooning. However in this report the only problem considered will be the longitudinal control.

3.2 Spacing policies

An important property for both vehicle platooning and commercial ACC is the spacing policy, i.e., what range a vehicle should keep to the preceding vehicle. There are a few spacing policies suggested in literature. One of the most common is the constant time headway policy (CTH), where the desired range to the preceding vehicle is calculated from a desired time gap (Ferrara and Vecchio, 2006; Liang and Peng, 2000; Naus et al., 2010; Xiao and Gao, 2011). This can be expressed as:

$$x_d(t) = v(t)T_g, \tag{3.1}$$

where $x_d(t)$ is the desired range to the preceding vehicle, $v(t)$ is Host's velocity and T_g is the desired time gap, i.e., the time it takes for Host to reach Target, when the velocity is kept constant (Naus et al., 2010). A constant term, $x_{d,0}$, can be added in (3.1) in order to maintain a minimum range at low velocities:

$$x_d(t) = x_{d,0}(t) + v(t)T_g. \tag{3.2}$$

The CTH spacing policy is common in commercial ACC:s because the desired range in the control law will become larger when the velocity is increased, which gives a safer behaviour; if a larger range is kept between the vehicles it will be easier to react in time when, for instance, Target brakes.

Another way to choose the spacing is to let it be constant, i.e.,

$$x_d(t) = x_{d,0}(t). \tag{3.3}$$

The advantage of using constant spacing is that the range can be kept small independent of the velocity, which leads to less aerodynamic drag and larger traffic throughput (Naus et al., 2010; Peppard, 1974; Seiler et al., 2004). This spacing policy is however not used in ordinary traffic since the constant range between the vehicles will lead to dangerously small ranges at high velocities. But for vehicle platooning, where the vehicles are driven autonomously and have short reaction times, this spacing policy can be favourable (Robinson et al., 2010).

3.3 Stability

To get an understanding of what the differences are between single-target and multi-target control the properties of both are discussed with respect to local stability and string stability.

3.3.1 Local stability

The spacing error is defined as

$$\varepsilon(t) = x - x_d, \tag{3.4}$$

where x is the range between Host and Target, and x_d is the desired range according to the adopted spacing policy.

An essential property of the ACC is the stability of the underlying longitudinal dynamics control system. That is, the spacing error, $\varepsilon(t)$, and the velocity error $\dot{\varepsilon}(t)$ should tend to zero as $t \to \infty$:

$$\begin{matrix} |\varepsilon(t)| \to 0 \\ |\dot{\varepsilon}(t)| \to 0 \end{matrix}, \qquad t \to \infty. \tag{3.5}$$

Closed loop stability can be presented by resorting to classical stability results, e.g., for linear systems the Nyquist stability criterion, and for non-linear systems the theory of Lyapunov stability.

3.3.2 String stability

An important concept for vehicle platooning is the string stability. In literature, several ways to present string stability exists, but the underlying desire is to have a stability concept that concludes whether perturbations in the range or range-rate amplifies or attenuates downstream (i.e. further back in the platoon) (Swaroop, 1997). If the platoon is not string stable, traffic shockwaves will be amplified downstream. This will make it uncomfortable to ride the vehicle and can lead to collisions.

The stability concept also depends on whether the platoon are assumed to be homogeneous, i.e., all vehicles have identical dynamics and controllers, or if it is assumed to be heterogeneous, i.e., the dynamics and controllers can differ between the vehicles.

The most general case is the non-linear heterogeneous interconnected system, where the vehicles in the platoon can have non-linear dynamics and controllers. A rigorous definition has been developed by Swaroop and Hedrick (1996), and defines a platoon of vehicles as string stable if all vehicles, for bounded initial range and range-rate errors, have bounded range and range-rate errors over all time and for any number of vehicles.

For homogeneous linear interconnected systems, this simply becomes

$$\frac{||\varepsilon_i||_\infty}{||\varepsilon_{i-1}||_\infty} < 1 \ , \tag{3.6}$$

where i is the i:th vehicle in the platoon (Naus et al., 2010). This expression holds for ε being both range error and range-rate error. If this were not the case, with infinite number of vehicles the range and range-rate errors would grow to infinity.

Instead of requiring that the errors should be bounded, some definitions use L_2 norms as the measure to be attenuated (Klinge and Middleton, 2009; Rapaport and Astolfi, 2004). This means that low steady state errors are penalized, which may be counter productive. Since no allowed steady state errors means stiff control on range, the L_2 norms may lead to high accelerations and high range-rate errors.

In the remainder of this thesis the string stability definition in (3.6) is used with respect to range and range-rate error.

3.4 Spacing policy effect on stability

The spacing policy's effect on stability has been extensively studied in literature. Depending on the spacing policy and the set of sensors available, different results hold. In this section the fundamental results on string stability are presented.

3.4.1 Vehicle dynamics

The output of the ACC, the desired acceleration, is sent to the vehicle's low-level control system, where it is used to calculate appropiate requests for the vehicle's propulsion and brake system. The relation between the acceleration of the vehicle, $A(s)$, and the desired acceleration, $U(s)$, can approximately be described by

$$\frac{A(s)}{U(s)} = \frac{1}{1+sT}e^{-s\tau}, \tag{3.7}$$

where T is the time constant and τ is the time delay of the vehicle's propulsion and brake system.

The relation between a vehicle's position, $P(s)$, and acceleration is

$$P(s) = \frac{1}{s^2}U(s). \tag{3.8}$$

Combining (3.7) and (3.8) gives the following transfer function from desired acceleration to actual position of the vehicle

$$G(s) = \frac{1}{s^2}\frac{1}{1+sT}e^{-s\tau}. \tag{3.9}$$

Model (3.9) approximately describes the vehicle dynamics and will be sufficient in this thesis, since τ includes all system delays and T is the time constant

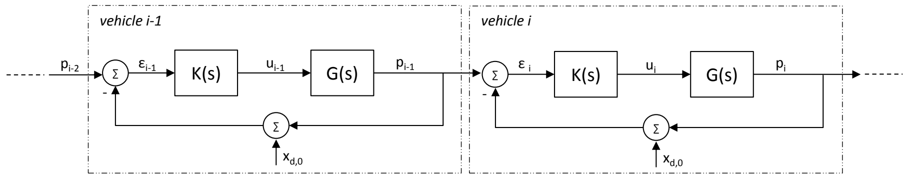


Figure 3.2: Range error dynamics between vehicles in a homogeneous platoon of vehicles using PD-controllers. $G(s)$ is the vehicle dynamics and $K(s)$ is the controller. Inputs to the controller are the range and the range-rate error to the preceding vehicle.

of the vehicle actuators. Since the actuators are regulated by closed loop controllers, the time constant T is chosen such that all vehicle dynamics is accounted for. The model (3.9) could be improved by using different values on T and τ depending on whether it is the propulsion or the brake system that currently is being used, and also introduce a delay when switching between these systems.

The vehicle dynamics will have an impact on the string stability. Larger time constants and time delays means that shorter reaction times are needed to avoid collisions. By the same reason the stability margins of the closed loop system of any vehicle in the vehicle string are decreased.

3.4.2 Constant spacing

It has been shown that a platoon, consisting of vehicles described by (3.9), never can be string stable if the vehicles employ constant spacing policy and only have information of Target (Naus et al., 2010; Seiler et al., 2004; Sheikholeslam and Desoer, 1993). With range and range-rate error to Target available as system inputs for each vehicle, a simple PD-controller will give the following control signal

$$U_i(s) = K(s)E_i(s), \tag{3.10}$$

where $E_i(s)$ is the range error between vehicle i and the preceding vehicle, and

$$K(s) = K_p + K_v s, \tag{3.11}$$

where K_p is the proportional gain and K_v is the derivative gain. The range error dynamics between the vehicles, shown in Figure 3.2, can then be written as

$$\Gamma_1 = \tfrac{E_1}{P_0} = \frac{1}{1+G(s)K(s)} = \frac{s^3T+s^2}{s^3T+s^2+e^{-s\tau}(sK_v+K_p)} \tag{3.12}$$

$$\begin{aligned}\Gamma_i = \tfrac{E_i}{E_{i-1}} &= \frac{G(s)K(s)}{1+G(s)K(s)} \\ &= \frac{e^{-s\tau}(sK_v+K_p)}{s^3T+s^2+e^{-s\tau}(sK_v+K_p)}, \text{ for } i>1,\end{aligned} \tag{3.13}$$

where Γ_1 describes the dynamics of the first range error, and Γ_i describes the dynamics of the consecutive range errors.

In the following analysis, the time delay, τ, is assumed to be zero to be able to perform an initial linear analysis. The influence of the time delay will be investigated later on in Chapter 4. The time constant, T, is assumed to be 0.5.

String stability will only be achieved when Γ_i is smaller than 0 dB over all frequencies. A bode plot of (3.12) and (3.13) is shown in Figure 3.3 where it can be seen that Γ_i is not below 0 dB for all frequencies. Although Figure 3.3 only shows one set of control parameters, it can be proven that it does not exist any set of control parameters that will attenuate disturbances at all frequencies for Γ_i. This can be proven with the Bode sensitivity integral as done in Seiler et al. (2004).

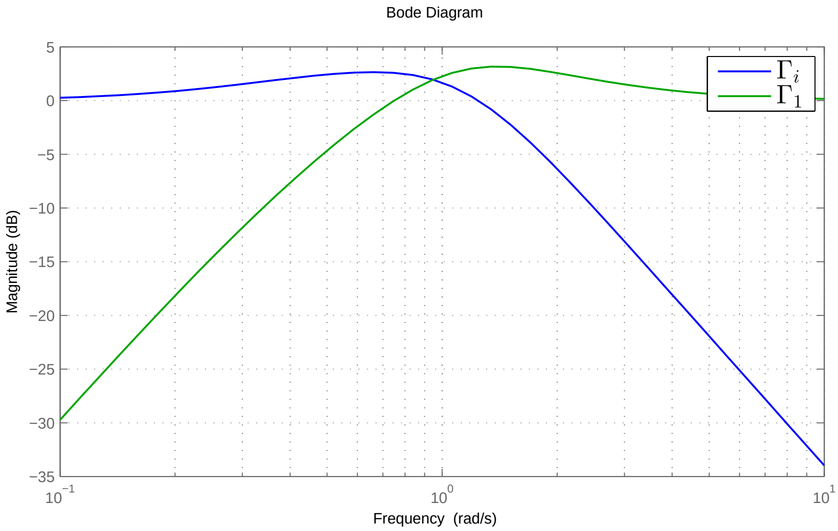


Figure 3.3: Example of non string stable platoon when constant spacing policy is used and only information regarding the preceding vehicle is available. $K_p = 0.3$ and $K_v = 0.9$.

On the other hand, with information of the lead vehicle (the first vehicle in the platoon) available to all vehicles, it is possible to make the platoon string stable (Naus et al., 2010; Seiler et al., 2004; Sheikholeslam and Desoer, 1996).

The control signal (3.11) is extended with a term depending on the range error between vehicle i and the lead vehicle, $E^l_{p,i}(s)$, in the following way:

$$U_i(s) = K(s)E_{p,i}(s) + K^l(s)E^l_{p,i}(s), \tag{3.14}$$

where

$$K^l(s) = K^l_p + K^l_v s. \tag{3.15}$$

The range error dynamics between the first and second vehicle will be the same as in (3.12) and for the following pair of vehicles it will be as shown in Figure 3.4. This can be written as

$$\Gamma_i = \frac{E_i}{E_{i-1}} = \frac{e^{-s\tau}(sK_v + K_p)}{s^3T + s^2 + se^{-s\tau}(K_v + K^l_v) + e^{-s\tau}(K_p + K^l_p)}, \text{ for } i > 1, \tag{3.16}$$

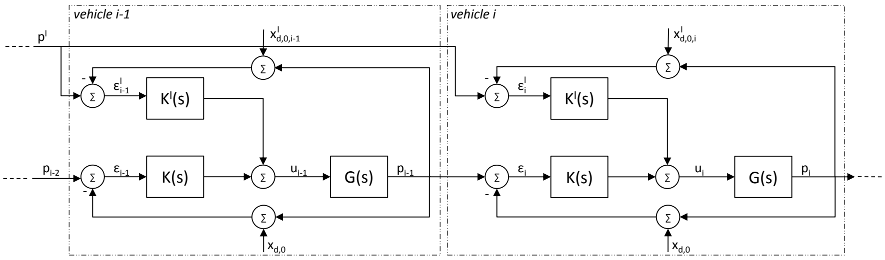


Figure 3.4: Range error dynamics between vehicles in a homogeneous platoon of vehicles using PD-controllers. $G(s)$ is the vehicle dynamics and $K(s)$ and $K^l(s)$ are the controllers. Input to $K(s)$ is the range error to the preceding vehicle, ε, and input to $K^l(s)$ is the range error to the lead vehicle ε^l. The desired range to the lead vehicle, $x^l_{d,0,i}$ and $x^l_{d,0,i-1}$, depend on how far back in the platoon the vehicles are.

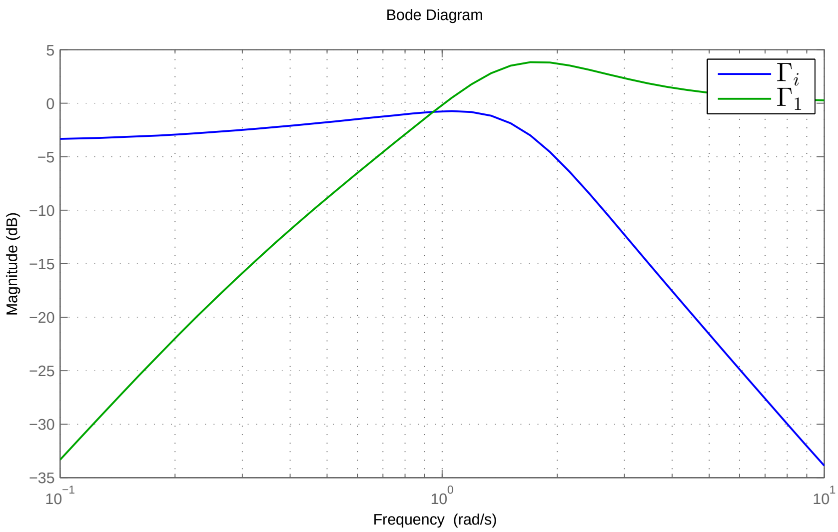


Figure 3.5: Example of string stable platoon when constant spacing policy is used and information regarding the preceding vehicle and the lead vehicle are available. $K_p = 0.3$, $K_v = 0.9$, $K^l_p = 0.15$ and $K^l_v = 0.45$.

which for certain set of parameters can be made string stable, see Figure 3.5.

Peppard (1974) has shown that with information from both the preceding vehicle and the following vehicle it is possible to achieve string stability. However, as mentioned in the paper, the effects on driving behaviour when taking rear vehicles into consideration in the controller are probably not wanted from a driver's comfort point of view. If for instance a vehicle approaches from the rear it will push Host forward towards Target. If the driver of Host has not noticed the vehicle behind, this action will come as a surprise.

Additionally, Shaw and Hedrick (2007) have shown that heterogeneous platoons using only leader following are stable, but this approach to string stable platoons does not consider vehicle separation. If enforced correctly, the strategy could be effective, but, if some error occur, for instance if one vehicle in the platoon receives an erroneous range measurement, the vehicles will not be able to avoid collisions with each other.

3.4.3 Constant Time Headway spacing

If the CTH spacing policy is used, the string stability conditions are easier to fulfill. Naus et al. (2010) have provided a constraint on the headway time T_g, in (3.1), which guarantees string stability for a given controller, implying that the constant time headway policy may have a positive effect on string stability. If the linear feedback controller has a cut-off frequency of ω_K Hz, then the constraint on T_g for guaranteeing string stability is

$$T_g \geq \sqrt{2}/\omega_K.$$

However it should be noted that, in the paper by Naus et al. (2010), the constraint on T_g is calculated by assuming an ideal vehicle in the form of a double integrator. Although this is not an accurate assumption, the results give an indication on how string stability is affected by the constant time headway spacing policy.

Xiao and Gao (2011) have developed another constraint on the headway time. The constraint is that a headway time that is twice as large as the total of the system's time lag and time delays makes the platoon string stable. The derivation of the constraint is only valid for homogeneous platoons with linear dynamics, but simulation results suggest that the same constraint applies for heterogeneous platoons as well (Xiao and Gao, 2011). A similar result is obtained by Zhou and Peng (2005).

The previous mentioned articles have only dealt with linear systems. With non-linear systems it is complicated to analyse the string stability. Since string stability means that a spacing error should decrease through the platoon, the amplification of the spacing errors should be less than unity. For linear systems the amplification is usually easy to calculate for all frequencies. In the non-linear case, however, the amplification of spacing errors can in many cases only be determined to be within a range of values. This range can be determined by finding a Lyapunov function for the given system that are bounded from below and above by two different non-decreasing functions (Teel, 1996). There is no guarantee that such functions exist for a given system. If the amplification can be determined to be in a range of values that are all less than unity, then the interconnected system is string stable.

3.5 Multi-target control string stability

In the above mentioned papers, only information about Target and the lead vehicle are available in the control law of Host. Next it is investigated if string stability can be achieved when information of more than one preceding vehicle is available and constant spacing policy is used. For simplicity it is assumed that only information about Target and Target+1 are available, but it is possible to use more vehicles ahead in the control law. The control law (3.11) extended with an additional part that acts on the range error to Target+1, $E'_{p,i}(s)$, is given by:

$$U_i(s) = K(s)E_{p,i}(s) + K'(s)E'_{p,i}(s), \tag{3.17}$$

where

$$K'(s) = K'_p + K'_v s. \tag{3.18}$$

$E'_{p,i}(s)$ can for example be obtained via wireless communication between Host and Target+1 or via on-board sensors on Host.

The transfer functions for the error dynamics for the first two errors then becomes

$$G_1(s) = \tfrac{E_{p,1}(s)}{P_0(s)} = \frac{1}{1 + H(s)K(s)} = \frac{s^2}{s^2 + sK_v + K_p} \tag{3.19}$$

$$G_2(s) = \tfrac{E_{p,2}(s)}{E_{p,1}(s)} = \frac{s(K_v - K'_v) + K_p - K'_p}{s^2 + s(K_v + K'_v) + K_p + K'_p}. \tag{3.20}$$

Since each multi-target vehicle uses information from Target and Target+1, the order of the transfer function for the error dynamics will increase for each vehicle further back in the platoon.

$K(s)$ and $K'(s)$ can easily be chosen such that $||G_2||_\infty < 1$. The third vehicle has information of both Target and lead vehicle. In Seiler et al. (2004) it is shown that if the vehicles in a platoon has information about both Target and lead vehicle the whole platoon can be made string stable. However, for the multi-target control case only the third vehicle has information about the lead vehicle, and thus the string stability is only valid for the second error G_2, i.e., the error between the second and the third vehicle. The error dynamics for vehicles further back in the platoon will not be as favourable. The error dynamics between two consecutive pair of vehicles, where all vehicles uses multi-target control, will be:

$$\begin{aligned} E_{p,i}(s) = & \frac{K_v s + K_p}{s^2 + s(K_v + K'_v) + K_p + K'_p} E_{p,i-1}(s) + \\ & \frac{K'_v s + K'_p}{s^2 + s(K_v + K'_v) + K_p + K'_p} E_{p,i-2}(s). \end{aligned} \tag{3.21}$$

Since the complexity of the transfer functions for the error dynamics increases further back in the platoon, no simple conclusion whether the vehicles are string stable has been found. To get an idea about the behaviour, a platoon containing 10 vehicles has been simulated with different values on the control parameters in (3.17). From the simulations one can infer that multi-target control is not string stable by default. Figure 3.6 shows the best found set of control parameters for the multi-target controller, and Figure 3.7 shows the corresponding simulation for a single-target controller. As can be clearly seen, range errors increase in the end of the platoon, but it is also evident that the multi-target controller has better performance than the single-target controller. The errors become smaller, and for the first few vehicles the errors are actually decreasing. A corresponding controller with leader information can be made string stable as can be seen in Figure 3.8.

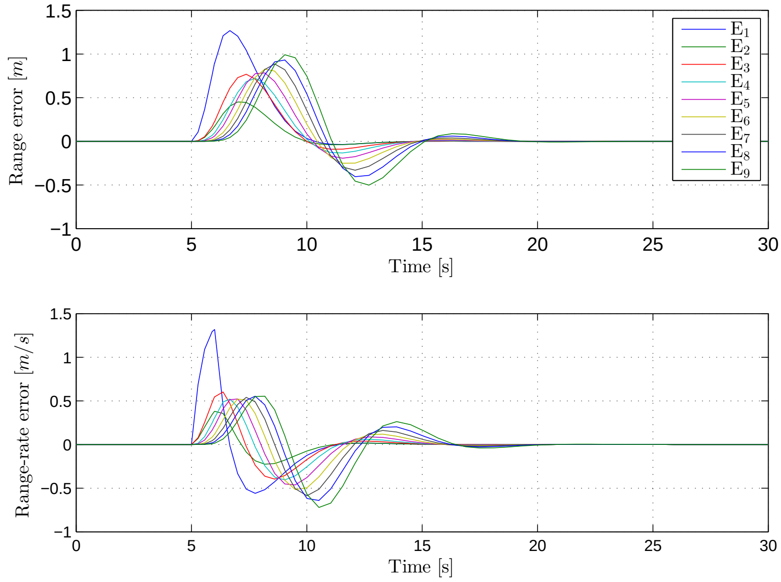


Figure 3.6: Ten vehicle platoon where the vehicles uses multi-target control. $K_p = 1$, $K_d = 1.5$, $K_p' = 0.5$, $K_v' = 0.75$.

3.6 Multi-target control performance

In Section 3.5 it was investigated whether a platoon of vehicles adapting multi-target control and constant spacing policy can be made string stable. No proof of string stability where found, but the results indicate that it can be advantageous to use multi-target control instead of single-target control. It was assumed that Target always follows Target+1, which is not true in regular traffic. However, if the range between Target and Target+1 is small it is likely that their driving behaviour will be linked to some extent.

Target and Target+1 are strongly linked if Target follows Target+1 strictly and tries to keep the range and range-rate errors small. Conversely, they are weakly linked if Target allows large range and range-rate errors to Target+1.

It is necessary to distinguish the situation where Target+1 accelerates from the situation where Target+1 decelerates. When Target+1 accelerates to a higher velocity than Target, Target can either accelerate and continue to follow Target+1, remain at the same velocity, or decrease the velocity. When Tar-

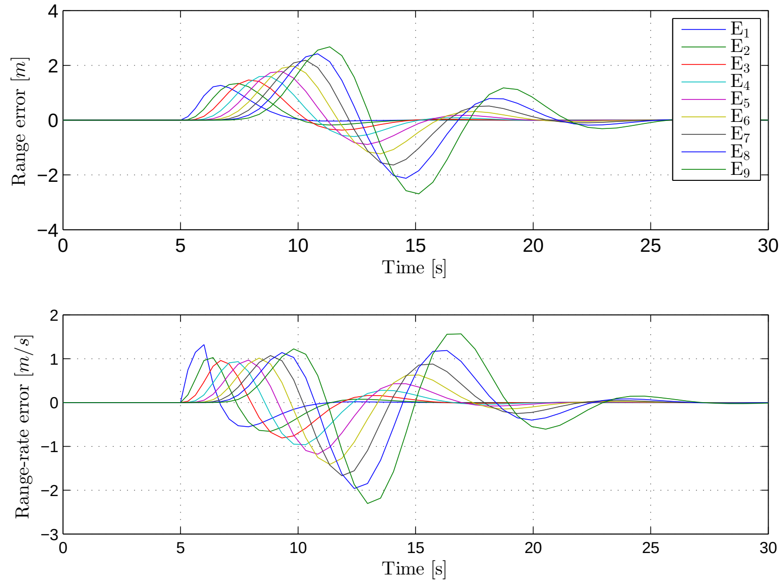


Figure 3.7: Ten vehicle platoon where the vehicles uses single-target control. $K_p = 1$, $K_d = 1.5$.

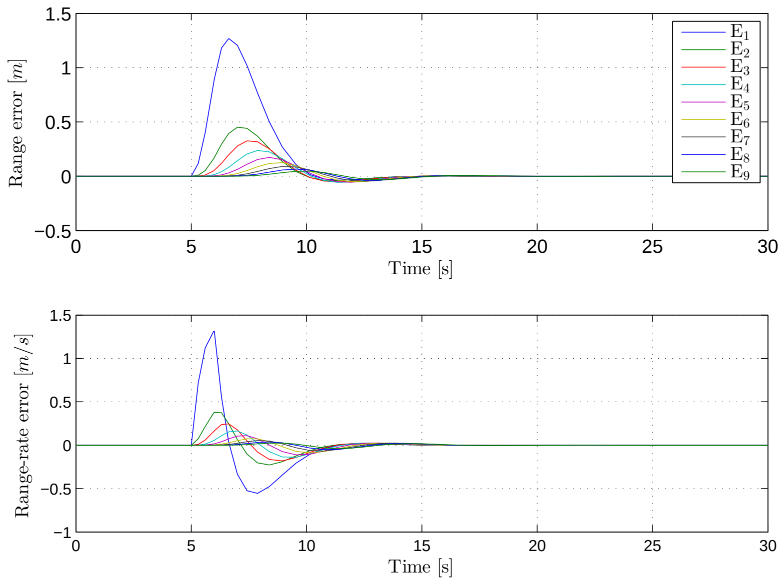


Figure 3.8: Ten vehicle platoon where the vehicles uses leader information in the control law. $K_p = 1$, $K_d = 1.5$, $K_p^l = 0.5$, $K_v^l = 0.75$.

get+1 decelerates to a lower velocity than Target, Target must also decelerate, otherwise a collision will occur.

To be able to compare the performance between a single-target and multi-target controller, both will be designed to be optimal using the Linear-quadratic regulator (LQR) algorithm described in Glad and Ljung (2000). The single-target controller only uses Target information in the control law, while the multi-target controller additionally uses Target+1 information. The knowledge that Target and Target+1 is linked is used in the design of the multi-target controller, and it is assumed that Target uses an optimal single-target controller. Since this is not always true, the Host's multi-target controller must be designed to be able to handle the situation where the link between Target and Target+1 is weak.

The two different Host controllers are evaluated and compared against each other both in the situation where Target and Target+1 are strongly linked, and in the situation where Target and Target+1 are weakly linked.

The vehicle dynamics are assumed to be equal for all vehicles, which means that all differences between simulations can be attributed to differences in the control algorithms.

3.6.1 Error model of Host

Host, using multi-target control, has knowledge of range and range-rate to both Target and Target+1. Host also has knowledge of the ego acceleration and the accelerations of Target and Target+1. Target, which uses single-target control, can only use the range and range-rate to Target+1, the ego acceleration and the acceleration of Target+1 in the control law. The following error model of Host, using multi-target control, is used in the LQR-design:

$$\dot{x}_1(t) = x_3(t) \tag{3.22a}$$

$$\dot{x}_2(t) = x_4(t) \tag{3.22b}$$

$$\dot{x}_3(t) = x_6(t) - x_5(t) \tag{3.22c}$$

$$\dot{x}_4(t) = x_7(t) - x_5(t) \tag{3.22d}$$

$$T\dot{x}_5(t) = u_H(t-\tau) - x_5(t) \tag{3.22e}$$

$$T\dot{x}_6(t) = u_T(t-\tau) - x_6(t) \tag{3.22f}$$

$$T\dot{x}_7(t) = u_{T+1}(t-\tau) - x_7(t) \tag{3.22g}$$

where

- $x_1(t)$ = Range between Host and Target
- $x_2(t)$ = Range between Host and Target+1
- $x_3(t)$ = Range-rate between Host and Target
- $x_4(t)$ = Range-rate between Host and Target+1
- $x_5(t)$ = Host acceleration
- $x_6(t)$ = Target acceleration
- $x_7(t)$ = Target+1 acceleration

and $u_H(t-\tau)$,$u_T(t-\tau)$ and $u_{T+1}(t-\tau)$ are the control signals for Host, Target and Target+1 respectively;

$$\begin{aligned}
u_H(t) &= \alpha_1^h x_1(t) + \alpha_2^h x_2(t) + \alpha_3^h x_3(t) + \alpha_4^h x_4(t) + \\
&\quad \alpha_5^h x_5(t) + \alpha_6^h x_6(t) + \alpha_7^h x_7(t) && (3.23)\\
u_T(t) &= \alpha_1^t (x_2(t) - x_1(t)) + \alpha_2^t (x_4(t) - x_3(t)) + \\
&\quad \alpha_3^t x_6(t) + \alpha_4^t x_7(t) && (3.24)\\
u_{T+1}(t) &= w(t) && (3.25)
\end{aligned}$$

The control signal of Target+1 is modelled as noise, $w(t)$, since its driving behaviour is unknown.

Host, using single-target control, can only use states that includes itself and Target as feedback. That means that α_2^h, α_4^h and α_7^h are zero. The remaining parameters, $(\alpha_1^h, \alpha_3^h, \alpha_5^h, \alpha_6^h)$, are equal to the Target parameters, $(\alpha_1^t, \alpha_2^t, \alpha_3^t, \alpha_4^t)$, since both are assumed to be using optimal single-target control.

The transfer function from the range error between Target and Target+1, $x_2(t) - x_1(t)$, to the range error between Host and Target, can be written as:

$$G_1(s) = \frac{X_1(s)}{X_2(s) - X_1(s)}, \qquad (3.26)$$

and the transfer function from the acceleration of Target to the acceleration of Host can be written as:

$$G_2(s) = \frac{X_5(s)}{X_6(s)}. \qquad (3.27)$$

3.6.2 Error model of Target

Target, which uses single-target control, only has information about the range and range-rate to Target+1, the ego acceleration and the acceleration of Target+1. The error-model for Target becomes:

$$\begin{aligned}
\dot{x}_1^*(t) &= x_3^*(t) && (3.28a)\\
\dot{x}_3^*(t) &= x_6^*(t) - x_5^*(t) && (3.28b)\\
T\dot{x}_5^*(t) &= u_T(t-\tau) - x_5^*(t) && (3.28c)\\
T\dot{x}_6^*(t) &= u_{T+1}(t-\tau) - x_6^*(t), && (3.28d)
\end{aligned}$$

where

$$\begin{aligned}
x_1^*(t) &= x_2(t) - x_1(t)\\
x_3^*(t) &= x_4(t) - x_3(t)\\
x_5^*(t) &= x_6(t)\\
x_6^*(t) &= x_7(t).
\end{aligned}$$

The control law for Target, (3.24), can be re-written as:

$$u_T(t) = \alpha_1^t x_1^*(t) + \alpha_2^t x_3^*(t) + \alpha_3^t x_5^*(t) + \alpha_4^t x_6^*(t). \qquad (3.29)$$

3.6.3 LQR synthesis

The control laws of the vehicles are determined by minimizing the following cost function:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt, \tag{3.30}$$

where x is the states in the dynamic models of the vehicles described in Section 3.6.1-3.6.2, and u is the vehicle's control signal. Q and R are the weighting matrices of the states and control signal, respectively. Larger weights on a state or control signal means that the control law will be chosen such that the particular state or signal will be as close to zero as possible at all times. By using the same weight function for both controllers, the controllers will be comparable.

LQR design is used to calculate the optimal state feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$ for both Host and Target. First the state feedback of Target in (3.29), $u_T(t) = \mathbf{K}^t \mathbf{x}^*(t)$, is calculated. The control gains α_1^t through α_4^t for Target are calculated using the model (3.28).

The calculation of the control parameters of the multi-target Host is based on (3.22). LQR is used to acquire $\mathbf{K}^h$ which contains the parameters α_1^h through α_7^h in (3.23).

In the following analysis it is assumed that $T = 0.5$. Q and R are chosen as:

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{R} = 1. \tag{3.31}$$

Since the accelerations of Target and Target+1 (x_6 and x_7) are not controllable from Host's point of view, the weighting of these states are not important when calculating $\mathbf{K}^h$. The weight on the range to Target+1, $q2$, will affect how large the gains on the Target+1 states will be (i.e. α_2^h, α_4^h and α_7^h). The analysis and comparison of the two types of controllers are concerned only with their differences and whether a multi-target controller can be made better than the corresponding single-target controller, hence the specific values of the weights are not important. However, Target's control law $u_T(t)$ will be varied after $u_H(t)$ is obtained, which means that the weight $q2$ needs to be adjustable. A too high value of $q2$ will make Host follow Target+1 and collide with Target in some situations, while a too low $q2$ will make the effect of Target+1 in Host's control law minimal. The former case is bad when Target is not following Target+1, while the latter is bad when Target is following Target+1.

By comparing the performance of the single-target and multi-target controller for different values of $q2$ when the link is strong and weak, appropriate values for α_4^h and α_7^h can be obtained for different scenarios. This will then be the basis for choosing parameters in the next chapter.

3.6.4 Strong link

The multi-target controller is evaluated for two different Target control parameter configurations. The first configuration is chosen such that the link between Target and Target+1 is strong. The control parameters for Target is determined with LQR, where Q is the identity matrix and R is unity. This will penalize all states equally when minimizing the loss function.

The control parameters for Host with a single-target controller equals the control parameters for Target. The control parameters for Host with a multi-target controller is determined with LQR for a set of different values of $q2$ in (3.31). Since all vehicles will follow each other closely, this can be seen as a mini-platoon.

Figure 3.9 shows a bode plot of G_1, which is the amplification of Host's range errors. The dashed line represents Host with single-target control, and the solid lines represent Host with multi-target control. As expected from the papers mentioned in Section 3.4.2, when leader information is available, i.e. Target+1, the platoon is string stable. This can be seen in figure 3.9 by noting that the magnitude for multi-target control is always less than 0 dB.

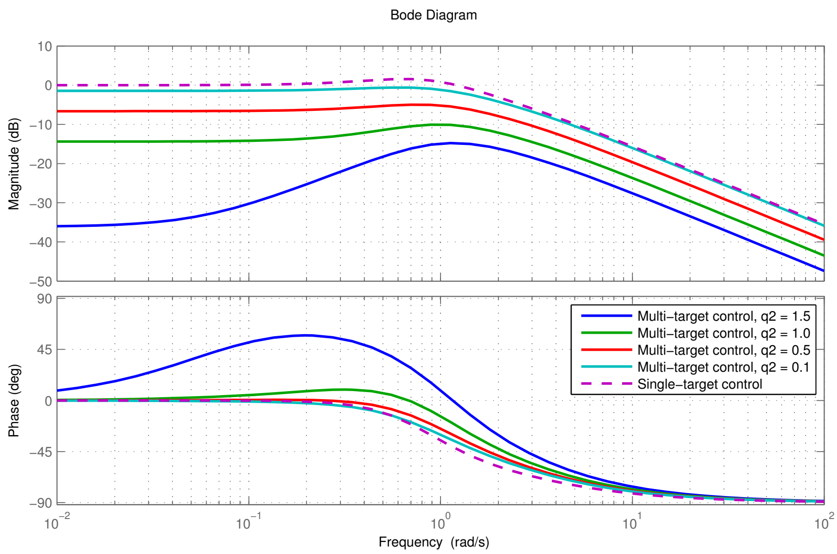


Figure 3.9: Bode plot of range error between Host and Target, when the link between Target and Target+1 is strong.

Figure 3.10 shows a bode plot of G_2; the amplification from Target acceleration to Host acceleration. From the figure it is evident that the maximal amplification of acceleration for Host with multi-target control is less than for Host with single-target control.

Figure 3.11 shows a step response of the system where Target+1 requests an acceleration of 3 m/s^2 during 3 seconds, i.e.:

$$u_{T+1}(t) = \begin{cases} 3, & \text{if } 3 < t \leq 6 \\ 0, & \text{otherwise.} \end{cases} \tag{3.32}$$

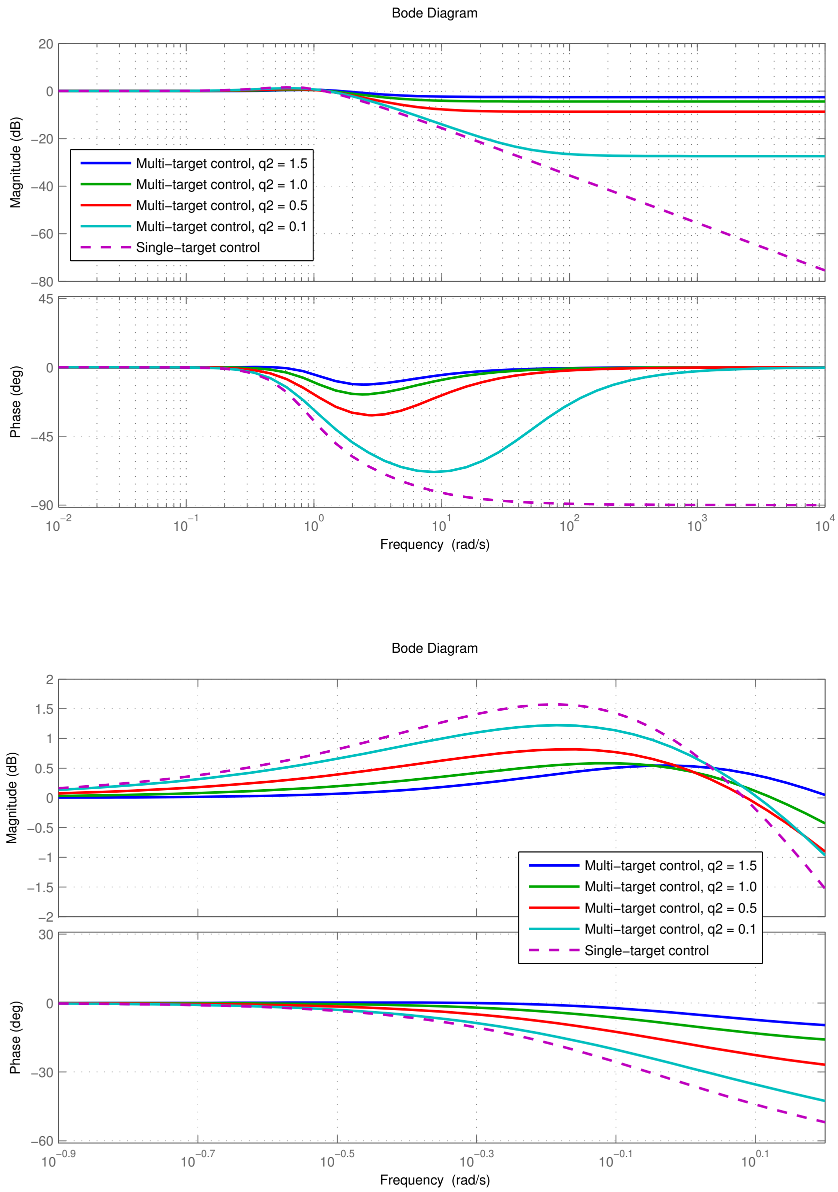


Figure 3.10: Bode plot of acceleration of Host, when the link between Target and Target+1 is strong. The bottom subplot shows a zoomed view.

As can be seen, a higher $q2$ causes the range-error and range-rate to be smaller.

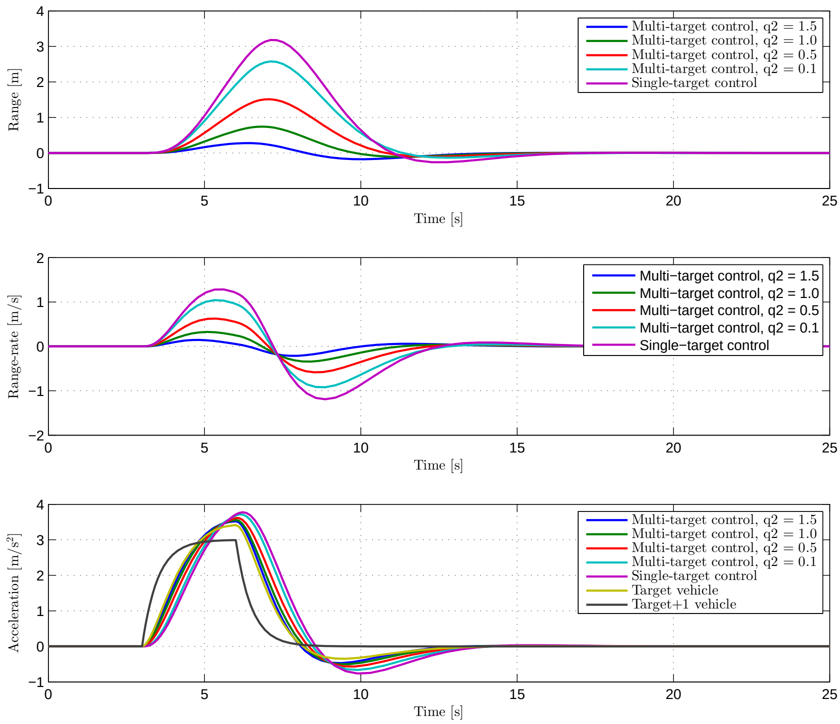


Figure 3.11: Pulse response when Target+1 accelerates 3 m/s^2 for three seconds, and the link between Target and Target+1 is strong.

3.6.5 Weak link

The second configuration is chosen such that the link between Target and Target+1 is weak. This behaviour is acquired by heavily penalizing the control signal in the LQR loss function, where Q is the identity matrix and $R = 100$. When Target+1 accelerates away, Target increases its speed, but with a lot less acceleration than Target+1.

The control parameters have been recalculated for Target, but the control parameters for Host with single-target and multi-target control have not, and are the same as when there was a strong link between Target and Target+1.

Figure 3.12 shows a bode plot of G_1, which is the amplification of range errors. Since the different Host controllers are much stiffer than the Target controller, each one is attenuating range errors.

Figure 3.13 shows a bode plot of G_2; the amplification from Target acceleration to Host acceleration. As opposed to the strong link, the acceleration amplification for Host with multi-target control is higher than for Host with

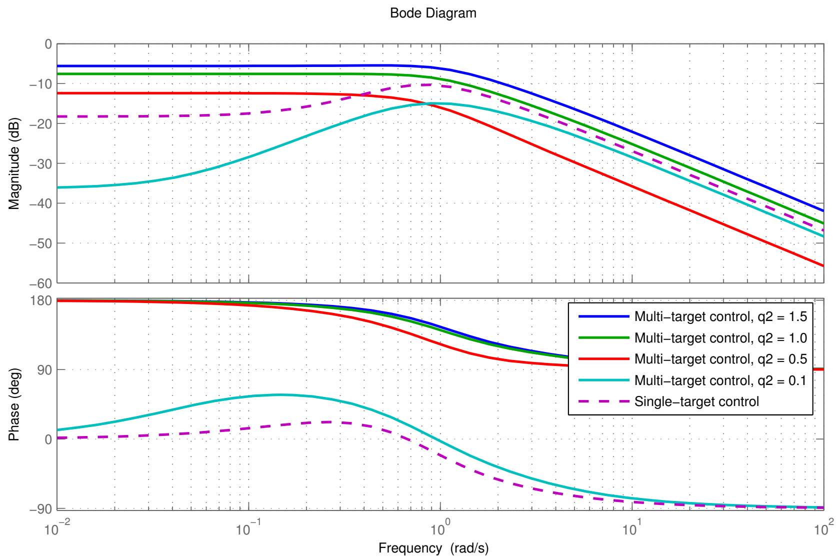


Figure 3.12: Bode plot of range error between Host and Target, when the link between Target and Target+1 is weak.

single-target control. This is a natural consequence of the less stiff Target controller.

Figure 3.14 shows a step response of the system where Target+1 requests an acceleration of 3 m/s^2 during 3 seconds. In this figure the downsides of the multi-target controller comes forth. When $q2$ is high the Host controller is affected too much by the range and range-rate errors to Target+1, and the range error to Target becomes negative. This means that the range to Target is smaller than desired, and if it gets too low, Host and Target may collide.

When the link between Target and Target+1 is weak, the behaviour will become worse if the gains on the Target+1 information are too large. This fact is used in the next chapter when a multi-target controller is synthesized for the ACC case.

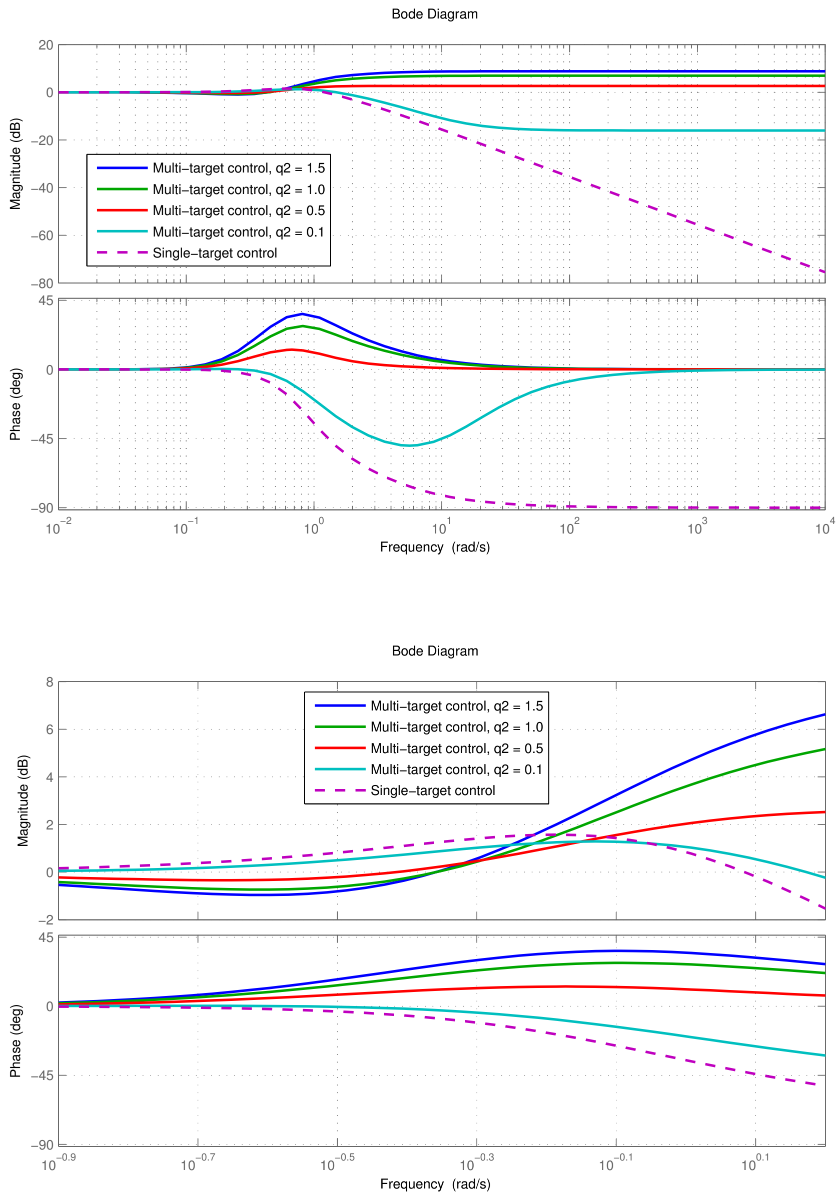


Figure 3.13: Bode plot of acceleration of Host, when the link between Target and Target+1 is weak. The bottom subplot shows a zoomed view.

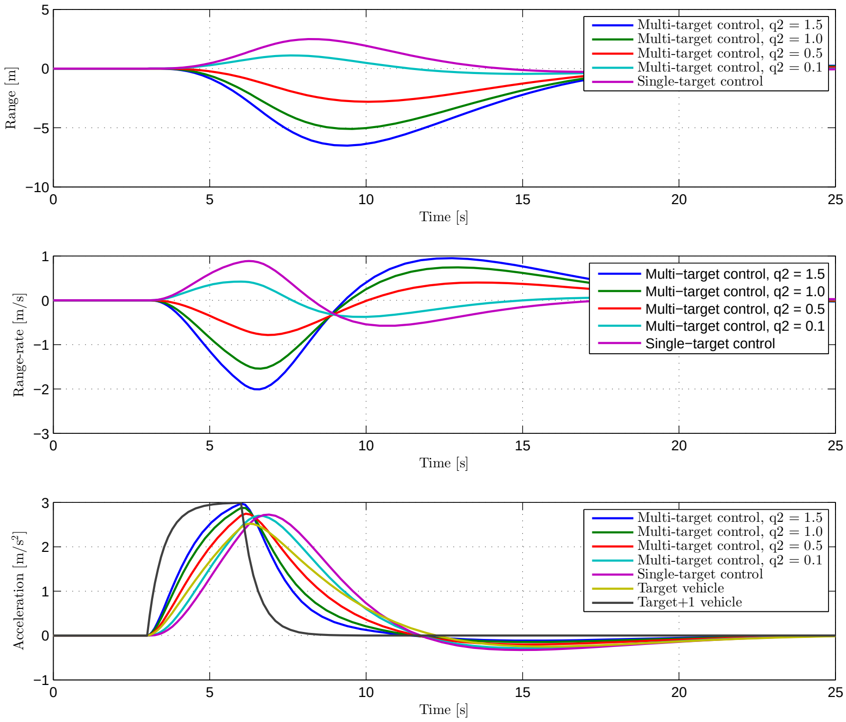


Figure 3.14: Pulse response when Target+1 accelerates 3 $\mathrm{m/s^2}$ for three seconds, and the link between Target and Target+1 is weak.

Chapter 4

Case study: Multi-target ACC

In the previous chapter it was shown that an optimal multi-target ACC can be used in vehicles to decrease range error, range-rate error and acceleration in certain situations. This knowledge is used in this chapter to expand the existing single-target ACC used in vehicles by VCC to a multi-target ACC. Since there already exists a single-target ACC in the vehicles, which must be used as a base, the analysis in the previous section will only give a hint of how the existing ACC should be modified and tuned to achieve a well behaving multi-target ACC.

4.1 The existing single-target ACC

The existing ACC from VCC is based on a non-linear single-target control law that uses the range, range-rate and acceleration of Target to calculate a desired acceleration,

$$u(t) = f(x_1(t), x_3(t), x_6(t)), \tag{4.1}$$

where the state variables are defined according to (3.22), i.e., $x_1(t)$ and $x_3(t)$ are the range and range-rate between Host and Target and $x_6(t)$ is the acceleration of Target.

The exact structure of $f(x)$ can not be disclosed in this report due to intellectual property reasons and in the remaining part of the chapter it is modelled as a black box that will not be changed. The reason for not changing the single-target ACC is that it has already been tested extensively by VCC and is known to have a comfortable behaviour at the same time as the range and range-rate errors are kept small. Often, Host will only have one vehicle close ahead, i.e, no Target+1 will be present. In these situations it is important that it exists a robust single-target controller to fall back to.

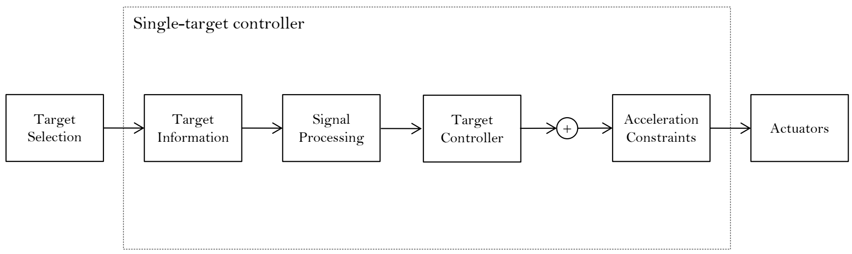
It is known that the single-target controller will force the range and range-rate error to Target to go to zero in finite time.

Velocity dependent spacing is used in the single-target ACC to attain a safer and comfortable behaviour, see Section 3.2.

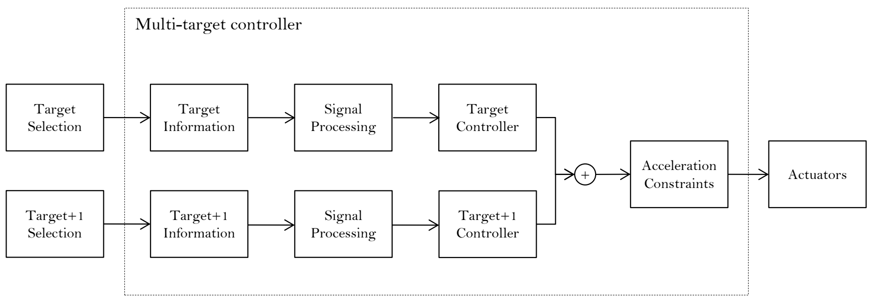
To further improve the comfort a rate-limiter is used to limit the rate of change of the acceleration. By limiting the first derivative of the desired acceleration the output is not allowed to change faster than a specified limit.

4.2 The multi-target ACC

The single-target ACC is extended to a multi-target ACC by adding a separate controller that acts on the Target+1 information, see Figure 4.1. The desired accelerations from the Target controller and the Target+1 controller are added together to calculate a combined desired acceleration. In Section 4.3 the multi-target controller is further extended by letting these parts interact with each other.



(a) Single-target ACC.



(b) Multi-target ACC.

Figure 4.1: In (a) the system overview of the already existing single-target ACC is shown. In (b) the single-target controller has been extended to a multi-target controller where Target+1 information is used in addition to the Target information.

An important requirement on the multi-target controller is that it should never request unnecessary accelerations leading to unsafe situations. If for example Target+1 accelerates and Target does not follow, it is crucial that Host does not start to accelerate, since this unnecessary acceleration could lead to a unsafe situation where the range to Target becomes too short.

4.2.1 Target+1 information

It is assumed that range and range-rate between Host and Target+1 are available, either via communication or through on-board sensors on Host. The range between Host and Target+1 will not be used in Host's control law, since it is not known at what range Target desires to follow Target+1. It could be assumed that the desired range between Target and Target+1 is the same as the desired range between Host and Target, but this is not always true, since the preferred range to the preceding vehicle will vary from driver to driver. Another aspect is that the length of the Target vehicle can vary largely. The length of a truck could be many times larger than the length of car.

There are some special cases when it could be beneficial to also use the range to Target+1 in the control law. If the range between Target and Target+1 is very small, safety could be improved by letting Host keep a larger range to Target. In case Target would collide with Target+1, Host would be at a safe range and a second collision could be avoided.

4.2.2 Control law

The non-linear single-target controller (4.1) is extended to a multi-target controller with the following control law:

$$u(t) = \underbrace{f(x_1(t), x_3(t), x_6(t))}_{\text{Target controller}} + \underbrace{\alpha_1 x_4 + \alpha_2 x_7}_{\text{Target+1 controller}}, \tag{4.2}$$

where x_4 is the range-rate between Host and Target+1, x_7 is the acceleration of Target+1 and α_1 and α_2 are control parameters used to tune the multi-target controller.

4.3 Controller synthesis

To achieve a well-behaving multi-target ACC, the control law (4.2) needs to be tuned and modified. A simulation environment of three vehicles (Host, Target and Target+1) is used in the controller synthesis to evaluate different tunings and modifications. The vehicles are modelled with the dynamics in (3.9) with $T = 0.5$ and $\tau = 0.3$.

The performance of the multi-target ACC is evaluated by comparing it with the corresponding single-target ACC. The multi-target ACC will be used in regular vehicles in regular traffic, hence it should be tuned and modified for real-life situations. To be able to do this, six different driving scenarios have been identified, which are used to tune and modify the multi-target ACC.

4.3.1 Driving scenarios

The identified driving scenarios aim to cover both situations where multi-target control can be beneficial, as well as situations where it can lead to problems. The six identified driving scenarios are:

1. *Target+1 accelerates and Target follows.* In Section 3.6 it was shown that it is possible to achieve smaller range error, range-rate error and

acceleration if multi-target control is used instead of single-target control when Target+1 accelerates and Target follows. With multi-target control, the reaction will be faster due to the extra information from Target+1.

2. *Target+1 accelerates and Target does not follow.* In Section 3.6 it was shown that if the link between Target and Target+1 is weak or non existing, i.e., Target does not follow Target+1, multi-target control could have a worse behaviour than single-target control. The Target+1 information will in this scenario only worsen the behaviour.

3. *Target+1 decelerates which leads to Target deceleration.* The same reasoning as for driving scenario 1 holds; the extra information from Target+1 will give multi-target control a faster response than with single-target control. In fact, Target must always decelerate if Target+1 decelerates, otherwise a collision will occur. If Target+1 decelerates at the same time as Target keeps constant velocity, Host should decelerate to avoid a multi-vehicle collision.

4. *Target accelerates and changes lanes, which makes Target+1 the new Target.* With a single-target controller, Host will follow Target and increase the velocity until Target leaves the lane. Then Host will have to decelerate since Target+1 drives with a lower velocity. With the extra information from Target+1, it should be possible to lower these accelerations.

5. *Target has an initial range error to Target+1 and catches up.* When Target catches up with Target+1 it has to decelerate to get the same velocity as Target+1. Host, that follows Target, has to do the same thing if it is using a single-target controller. If Host is using multi-target control it should be possible to decrease the acceleration earlier on due to the Target+1 information.

6. *Stop & Go driving.* The Stop & Go driving scenario is meant to resemble queue driving. Target follows Target+1 which alternates between accelerating and decelerating, i.e., a combination of driving scenarios 1 and 3 with fast changes in-between.

In driving scenario 1-4 and 6 it is assumed that the initial range and range-rate error for the vehicles are equal to zero.

In the following controller synthesis a large number of simulations have been done with different control parameters, velocities and accelerations. The simulations that are presented show the typical behaviour with the best found control parameters aiming to reduce range error, range-rate error and acceleration of Host.

To be able to handle all kind of velocities and accelerations of Target and Target+1, one set of control parameters is not sufficient, instead gain scheduling is introduced in Subsection 4.3.5.

The figures in the controller synthesis contain 4 sub-plots; range errors, range-rate, accelerations and desired accelerations. In the second sub-plot the range-rate from Host to Target is shown with solid lines. The range-rate from Host to Target+1, for the multi-target ACC, is shown with a dashed green line. In the fourth sub-plot the contribution to the desired acceleration from the Target controller and the Target+1 controller of Host are shown separately.

4.3.2 Contribution from the Target+1 controller

The Target+1 controller in the control law (4.2) consists of two parts. One part that acts on the range-rate to Target+1, x_4, and one part that acts on the acceleration of Target+1, x_7. To see how each of these two parts affect the behaviour of the multi-target controller, they are simulated separately. First only the range-rate is used in the Target+1 controller in (4.2), i.e., $\alpha_2 = 0$. Driving scenario 3, *Target+1 decelerates which leads to Target deceleration*, can be seen in Figure 4.2 . The multi-target ACC will not react faster than the single-target controller since the acceleration of Target+1 is not used. However, the range-error, range-rate error and acceleration will in total be smaller since the range-rate error to Target+1 will give a contribution earlier than the range-rate error to Target. When Target+1 stops decelerating a range-rate error between Host and Target+1 occurs due to the dynamics and the rate-limiter of Host, i.e., Target+1 drives faster than Host. This range-rate error will lower the deceleration of Host with multi-target ACC, which in the end leads to smaller errors to Target.

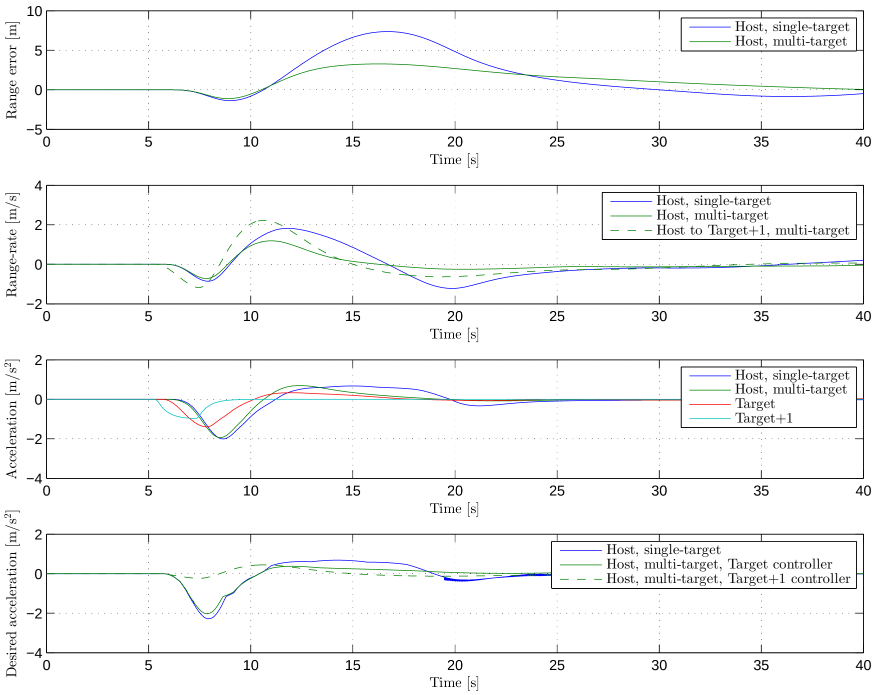


Figure 4.2: Driving scenario 3, $\alpha_1 = 0.2$ and $\alpha_2 = 0$. The initial velocity of the vehicles are 30 km/h. After 5 seconds Target+1 decelerates 1 m/s^2 during 2 seconds.

If instead only the acceleration of Target+1 is used in the Target+1 controller in (4.2), i.e., $\alpha_1 = 0$, the response will be faster if multi-target ACC is used instead of single-target ACC. This will lead to a smaller deceleration, as can be seen in Figure 4.3, since the multi-target ACC will start the deceleration earlier, i.e., the deceleration will be smaller but will take place during a longer period of time. The deceleration of the multi-target ACC is, with the selected tuning, 25 % lower compared to the single-target ACC.

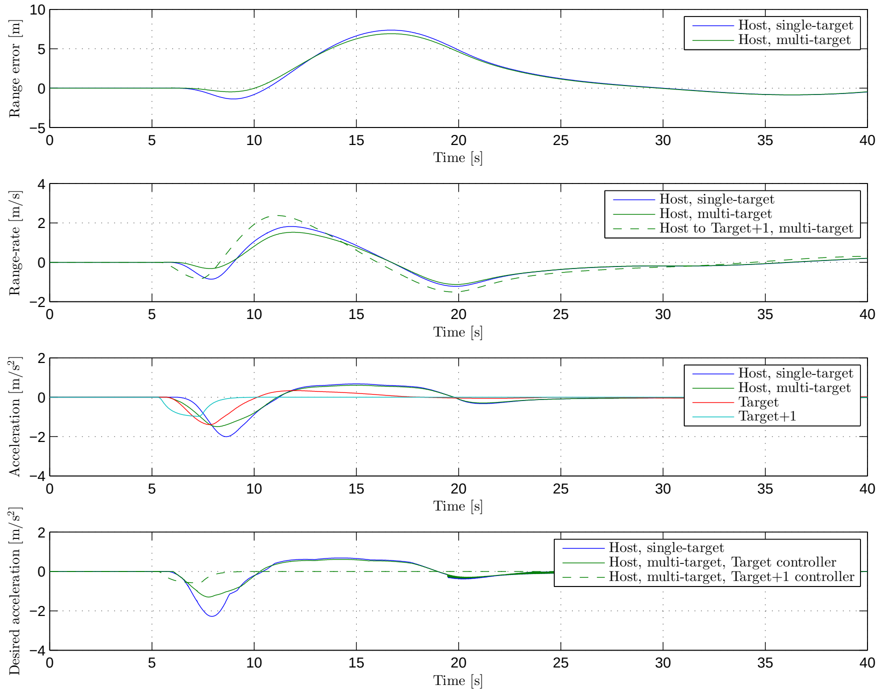


Figure 4.3: Driving scenario 3, $\alpha_1 = 0$ and $\alpha_2 = 0.6$. The initial velocity of the vehicles are 30 km/h. After 5 seconds Target+1 decelerates 1 m/s^2 during 2 seconds.

Using both the range-rate error to Target+1 and the acceleration of Target+1 in (4.2) leads to smaller range and range-rate errors at the same time as the size of the deceleration will be smaller, as can be seen in Figure 4.4. The acceleration is lowered by 22 % at the same time as the range and range-rate error is significantly smaller than those of the corresponding single-target ACC. Since the range error to Target is small, the multi-target ACC can have a low acceleration after Target+1 has stopped decelerating. Compare the acceleration of 'Host, multi-target' at 15 s in Figures 4.3 and 4.4.

4.3.3 Limiting Host's Target+1 controller output

If the control law in (4.2) is used for driving scenario 2, i.e *Target+1 accelerates and Target does not follow*, the behaviour of Host will be very bad due to the Target+1 contribution of the controller. The range-rate error between Host and

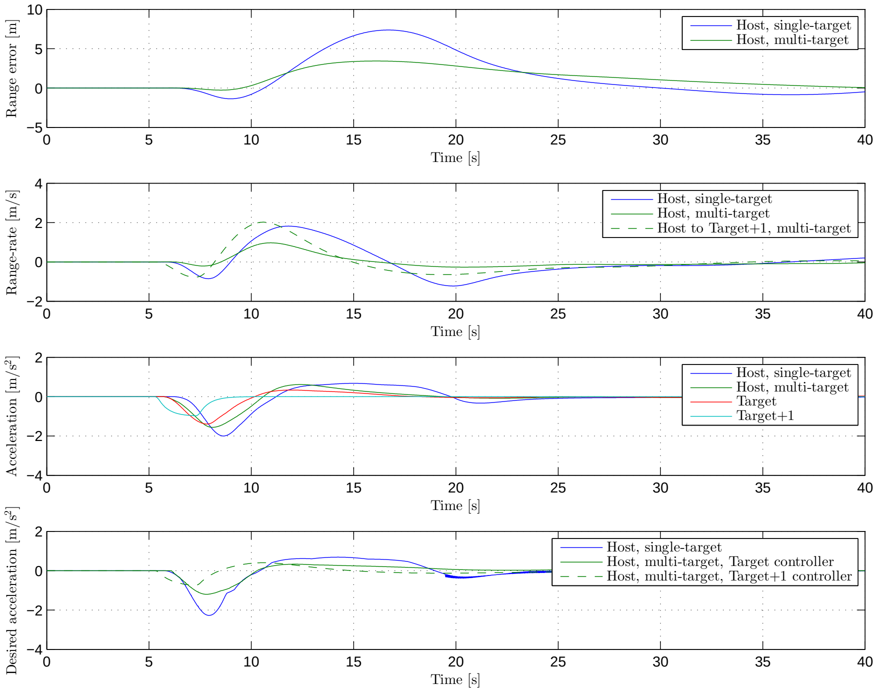


Figure 4.4: Driving scenario 3, $\alpha_1 = 0.2$ and $\alpha_2 = 0.6$. The initial velocity of the vehicles are 30 km/h. After 5 seconds Target+1 decelerates 1 m/s^2 during 2 seconds.

Target+1, x_4, and the acceleration of Target+1, x_7, in (4.2) will become large when Target+1 accelerates, as can be seen in Figure 4.5. After the acceleration of Target+1 is finished there will still be a range-rate error between Host and Target+1, which will require Host to accelerate .

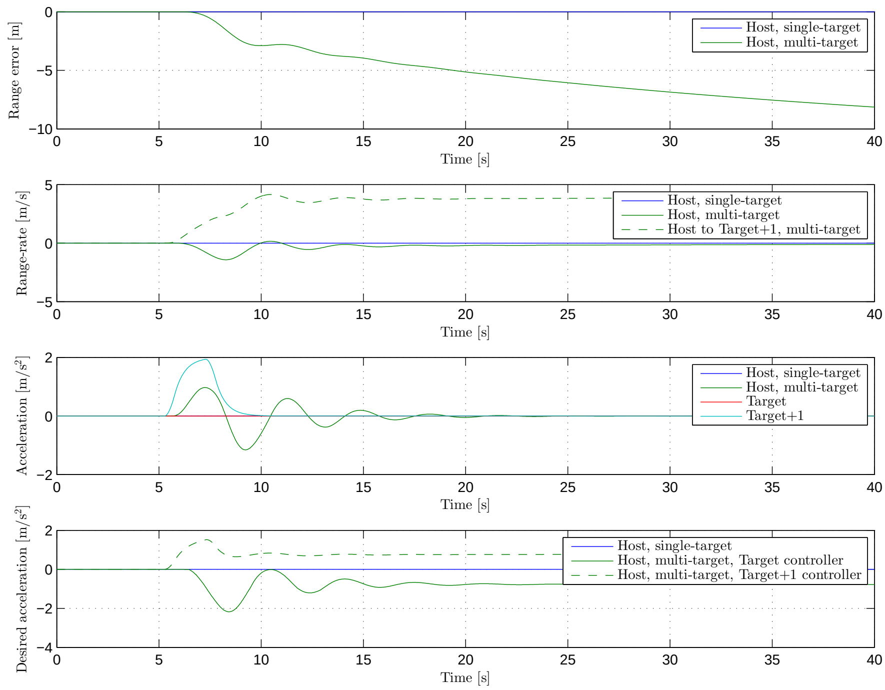


Figure 4.5: Driving scenario 2, $\alpha_1 = 0.2$ and $\alpha_2 = 0.6$. The initial velocity of the vehicles are 30 km/h. After 5 seconds Target+1 accelerates 2 m/s^2 during 2 seconds.

To avoid this unnecessary and unsafe behaviour, the maximum Target+1 contribution to the acceleration is limited. The maximum contribution from Target+1 is linearly limited with respect to the contribution from Target in (4.2), i.e., a dependency between the two parts of the multi-target ACC is introduced. If the contribution from Target is large, then the contribution from Target+1 is also allowed to be large. The upper limit $u_{\text{max,Target+1 controller}}$ is calculated as

$$u_{\text{max,Target+1 controller}} = \alpha_{lim} u_{\text{req,Target controller}}, \tag{4.3}$$

where α_{lim} decides how large the contribution from Target+1 is allowed to be for a given contribution from Target, $u_{\text{req,Target controller}}$. The limit function, with $\alpha_{lim} = 0.15$, can be seen in Figure 4.6. The reason for selecting α_{lim} to be smaller than one is to make the contribution from Target+1 work better together with the contribution from Target. The already existing Target controller, which can not be changed, is tuned to be more aggressive when Target decelerates compared to when Target accelerates. By using α_{lim} less than one this behaviour is also achieved for the Target+1 controller of Host.

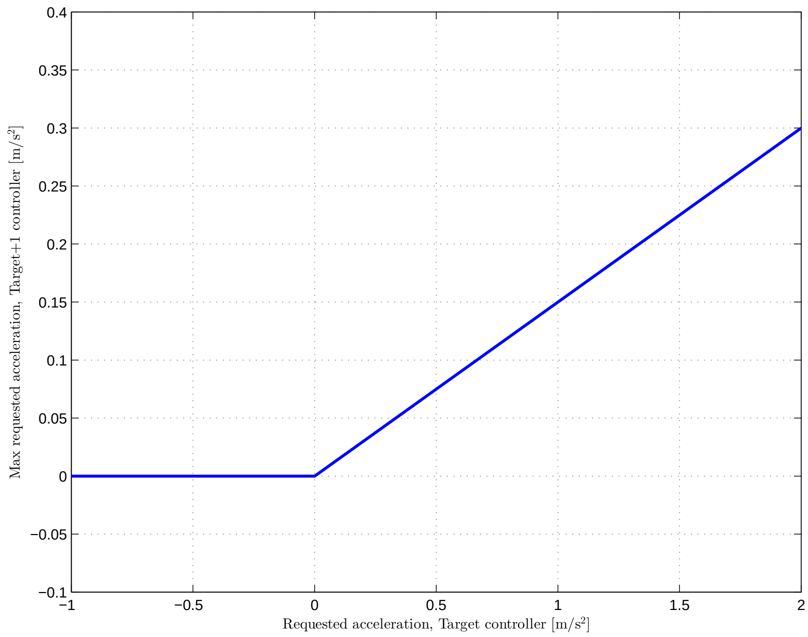


Figure 4.6: Limit function for the Target+1 controller of Host with $\alpha_{lim} = 0.15$.

When Target+1 decelerates, Target must decelerate to avoid a collision. Therefore no lower limit is required for the requested acceleration of the Target+1 controller.

Driving scenario 1, *Target+1 accelerates and Target follows*, can be seen without the limit in Figure 4.7 and with the limit implemented in Figure 4.8. Without the limit the multi-target ACC will respond very fast when Target+1 accelerates, but the limit must be used to handle a situation when Target does not follow Target+1. With the limit implemented the response will not be faster for the multi-target ACC compared to the corresponding single-target ACC. Due to the calm behaviour of the Target controller the acceleration will be slightly higher, but the range and range-rate error will be smaller. Whether this behaviour is desirable or not needs to be investigated with extensively in-vehicle tests.

4.3.4 Time-gap dependency

If the range from Target to Target+1 is large it is obvious that Target does not follow Target+1. In these cases the Target+1 controller of Host will only make the behaviour of the multi-target ACC worse. A weight function is inserted in the multi-target ACC to solve this problem. The output from the Target+1 controller is modified with respect to the time-gap between Target and Target+1.

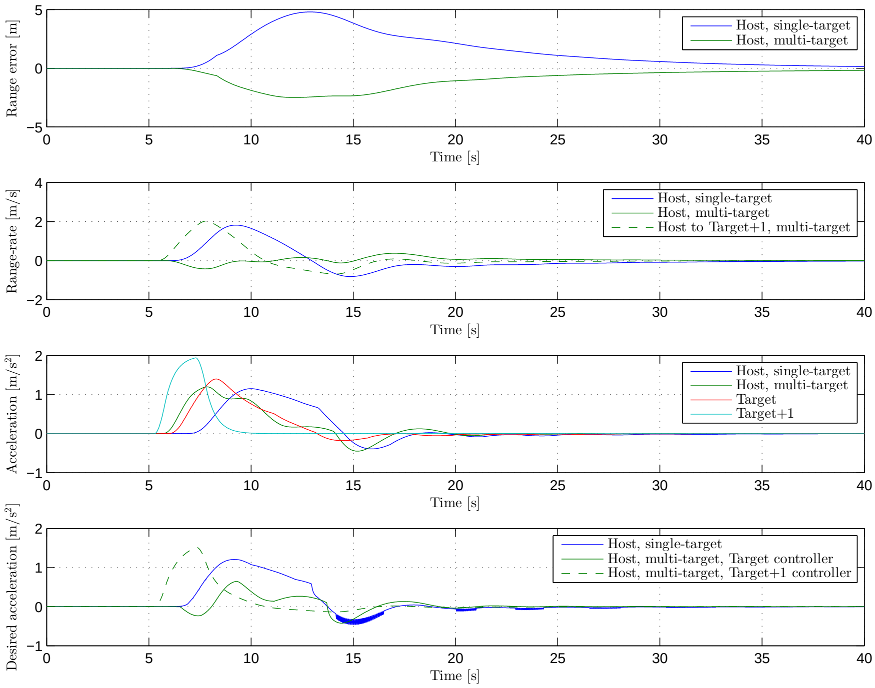


Figure 4.7: Driving scenario 1, $\alpha_1 = 0.2$ and $\alpha_2 = 0.6$. The initial velocity of the vehicles are 30 km/h. After 5 seconds Target+1 accelerates 2 m/s^2 during 2 seconds.

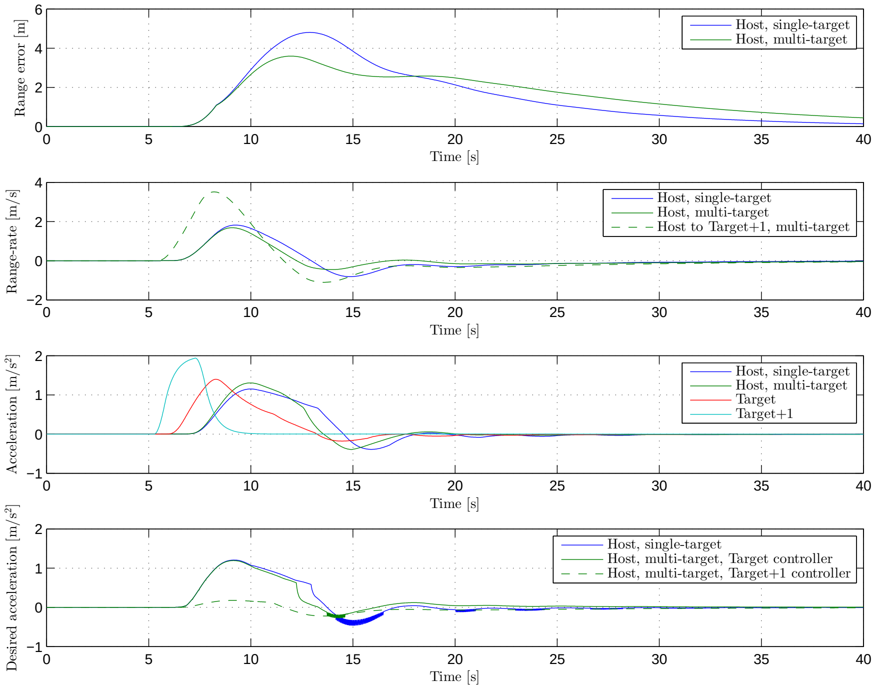


Figure 4.8: Driving scenario 1, $\alpha_1 = 0.2$ and $\alpha_2 = 0.6$ and Target+1 controller limited. The initial velocity of the vehicles are 30 km/h. After 5 seconds Target+1 accelerates 2 m/s^2 during 2 seconds.

The time-gap, T'_g, between Target and Target+1 is expressed as:

$$T'_g = \frac{x_2 - x_1}{v}, \tag{4.4}$$

where $x_2 - x_1$ is the range between Target and Target+1 and v is the velocity of Host. When the time-gap is small the entire output of Host's Target+1 controller will be used, and when the time-gap is large the output will not be used at all. A piecewise linear weight function is used to get a smooth transition, see Figure 4.9. Time-gap T1 is where the transition starts and time-gap T2 is where the transition ends. At time-gaps up to T1 it is assumed that Target follows Target+1 and at time-gaps larger than T2 it is assumed that Target drives completely independent of Target+1. A larger T2 will give an improved performance when Target+1 follows Target well, but a worse behaviour when this not is true.

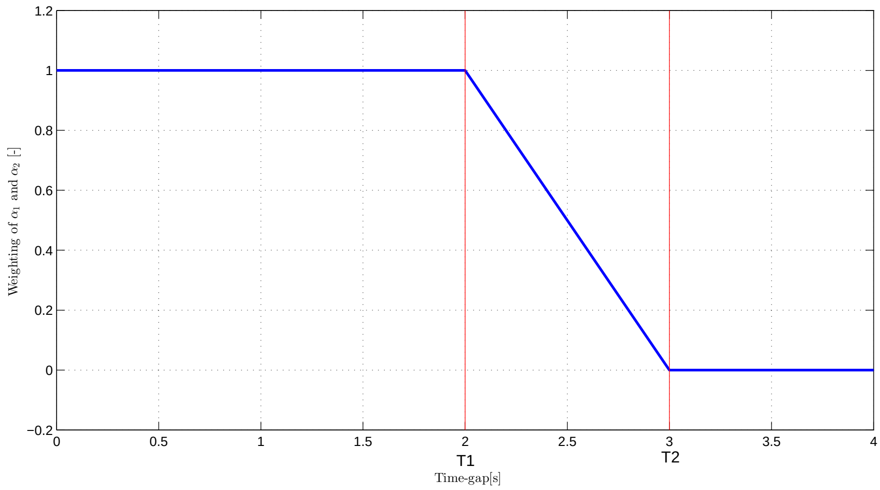


Figure 4.9: Weight function of α_1 and α_2 with respect to time-gap between Target and Target+1. T1 is the start of the transition and T2 is end of the transition.

4.3.5 Velocity dependent gains

The Target controller (4.1) of Host is gain scheduled to have a comfortable and robust behaviour for all different kinds of driving scenarios, i.e., there is a good balance between keeping the accelerations small at the same time as range and range-rate errors are attenuated. The gains depend on many different factors, e.g., velocity of Host and headway time to Target. In Section 4.3.3 the contribution from Target+1 in (4.2) was made dependent on the contribution from Target. To further improve the multi-target ACC, velocity dependent gain scheduling is inserted for α_2 and α_5. By simulating the different driving scenarios at different velocities and evaluating different gains, a set of gains has been obtained, see Figure 4.10. This set of gains makes the Target+1 controller of Host work well together with the Target controller. With in-vehicle tests it should be possible to fine-tune these gains.

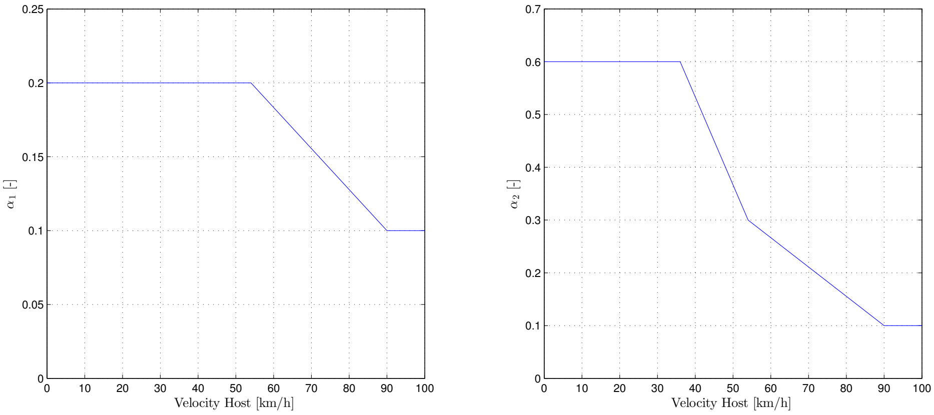


Figure 4.10: Weighting of α_1 and α_2 with respect to time-gap between Target and Target+1.

4.3.6 Driving scenario 4 - Target accelerates and changes lanes

Next it is evaluated how the multi-target ACC behaves in driving scenario 4, *Target accelerates and changes lanes, which makes Target+1 the new Target* . With a single-target ACC, Host will follow Target and increase the velocity until Target leaves the lane, then Host will have to decelerate since Target+1 drives with a lower velocity. The multi-target ACC will have a similar behaviour, but due to the Target+1 controller the acceleration will be smaller since Target+1 drives with constant velocity.

In Figure 4.11 Target starts to accelerate after 5 seconds. Host with multi-target ACC will accelerate and follow Target which will introduce a range-rate error to Target+1. This range-rate error will result in a negative contribution from the Target+1 controller leading to a smaller total acceleration. When Target leaves the lane, at $t = 10$ s, Host has to decelerate. Since the range-rate error is smaller than for the corresponding single-target ACC, the deceleration will also be smaller.

4.3.7 Driving scenario 5 - Target has an initial range-error to Target+1 and catches up

Figure 4.12 show a catch-up scenario where Host starts with zero range and range-rate error to Target, and Target starts with 25 meters range error to Target+1. Target accelerates due to the range error which makes Host accelerate as well . This will introduce a range-rate error to Target+1 which will lead to a negative contribution from the Target+1 controller of Host with multi-target ACC. The total requested acceleration will thus become smaller for the multi-target ACC compared to the corresponding single-target ACC.

When Target is getting closer to Target+1 it has to decelerate to get the same velocity as Target+1. The behaviour of Host is similar, but if a multi-target ACC is used, this acceleration is smaller since the acceleration used in

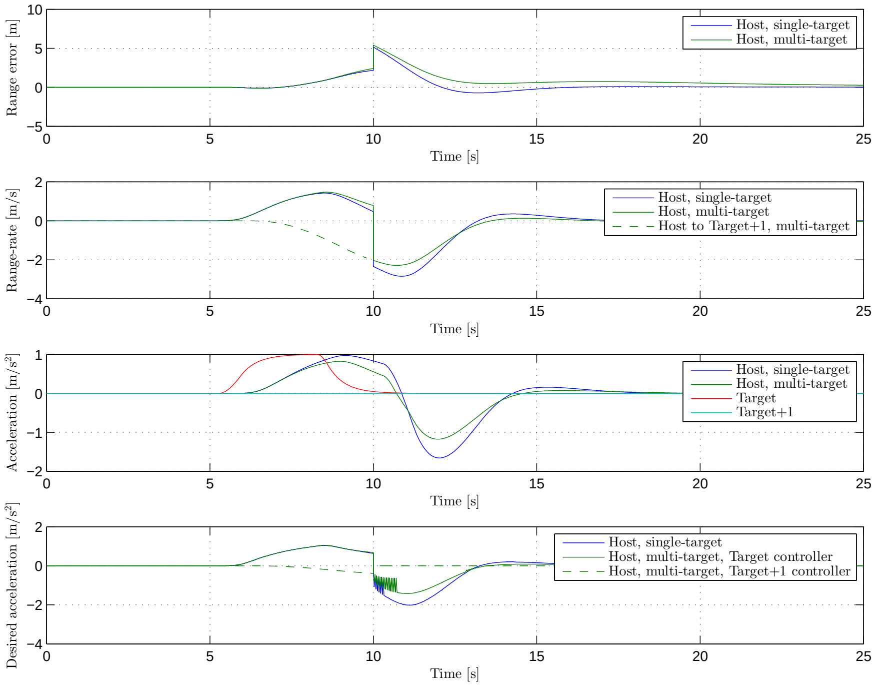


Figure 4.11: Driving scenario 4. The initial velocity of the vehicles are 50 km/h. Target starts accelerating after 5 seconds and leaves the lane after 10 seconds.

the catch-up was smaller.

4.3.8 Driving scenario 6 - Stop & Go driving

The Stop & Go driving scenario is used to resemble queue driving. This has been simulated in Figure 4.13 where Target+1 alternates between accelerating and decelerating 1 m/s^2.

The multi-target controller has a superior behaviour compared to the single-target controller. The extra information from Target+1 decreases both the range and range-rate error as well as the acceleration. Also, the range error is kept at a more stable level.

4.4 In-vehicle tests

The multi-target ACC from the previous section is further evaluated through implementing it in a test vehicle and evaluating it on a test track.

4.4.1 Current Target information filtering

The available Target information in the test vehicle are the range and range-rate. The acceleration for Target is calculated from the range and range-rate with a Kalman filter. The acceleration is also low-pass filtered with a range dependent time constant. The time constant is large at high range and low at short range. At short ranges the feed-forward gain of the acceleration must be quite large, since small range errors is desirable. At larger ranges (larger time-gaps), errors in range and range-rate will not be as noticeable and they will not matter as much. In those cases it is better to increase the comfort and lower the change of acceleration through the use of heavier low-pass filtering.

4.4.2 Fusing and filtering Target+1 data

Due to intellectual property issues it can not be disclosed in this report how the Target+1 information is obtained.

The Target+1 information is not always available and contains a lot of noise. The transition between single-target control and multi-target control must be bump-less to achieve a comfortable behaviour of the vehicle.

The information available to Host of Target+1 are range, range-rate and range acceleration. To get the absolute acceleration of Target+1 a Kalman filter is used as an observer.

The available signals which can be used to estimate the absolute Target+1 acceleration are, in addition to the information about Target+1, Host speed and acceleration. These signals are filtered in a Kalman filter. The accelerations are

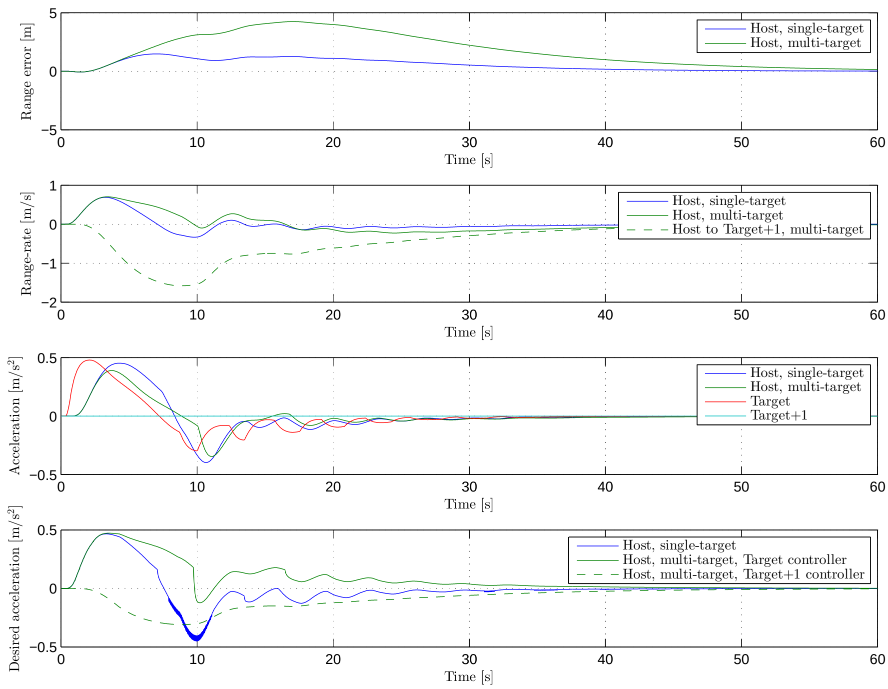


Figure 4.12: Driving scenario 5. The initial velocity of the vehicles are 50 km/h. Target starts with 25 meters range error to Target+1. Host has no range error to Target.

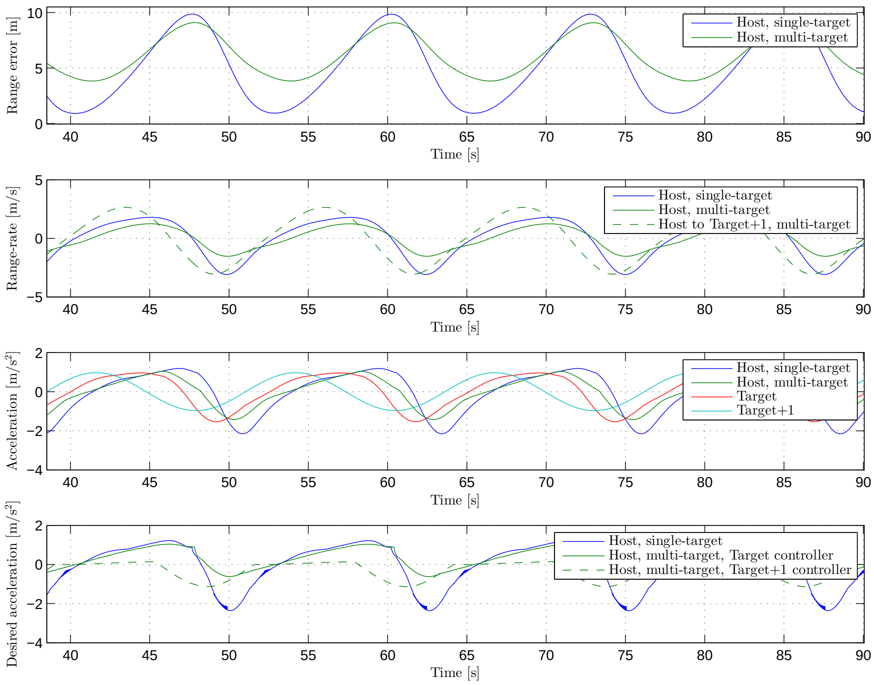


Figure 4.13: Driving scenario 6. The initial velocity of the vehicles are 15 km/h. Target+1 alternates between accelerating and decelerating 1 m/s^2.

assumed to be constant, which leads to the following discrete state space model:

$$x^\dagger(t+1) = \begin{bmatrix} x_1^\dagger \\ x_2^\dagger \\ x_3^\dagger \\ x_4^\dagger \\ x_5^\dagger \end{bmatrix} = \begin{bmatrix} 1 & h & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -h & -h^2/2 & 1 & h & h^2/2 \\ 0 & 0 & 0 & 1 & h \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x^\dagger(t) \tag{4.5}$$

$$y^\dagger = \begin{bmatrix} y_1^\dagger \\ y_2^\dagger \\ y_3^\dagger \\ y_4^\dagger \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix} x^\dagger(t), \tag{4.6}$$

where

- $x_1^\dagger$ = Host speed
- $x_2^\dagger$ = Host acceleration
- $x_3^\dagger$ = Range to Target+1
- $x_4^\dagger$ = Target+1 speed
- $x_5^\dagger$ = Target+1 acceleration
- $y_1^\dagger$ = Host speed
- $y_2^\dagger$ = Target+1 range
- $y_3^\dagger$ = Target+1 range rate
- $y_4^\dagger$ = Target+1 range acceleration

and h is the sample time. The covariances for the measured signals in R_2 are set high relative to R_1 to get smooth filtered signals. The range-rate to Target+1 is then obtained simply by subtracting the speed of Host from the Target+1 speed.

The filter is in use when Target+1 information is available. Which vehicle that is reported as Target+1 depends on the traffic situation, and the Target+1 can suddenly disappear, appear or change to another vehicle. This means instantaneous changes of range, range rate and range acceleration to Target+1. As the Kalman filter acts in many ways as a low pass filter, the sudden changes in the measured signals give rise to transients which in certain conditions may cause undesired actuation requests.

For example, if the designated Target+1 is a vehicle at a range of 60 meters, and it is then changed to a vehicle at a range of 20 meters, then a transient in the range would occur. Since the range-rate is the derivative of range, the Target+1 change would lead to a large negative range-rate. Large negative range-rates makes the ACC brake, which may be undesirable if the new Target+1 is speeding away.

To prevent such behaviour the Kalman filter is surrounded by logic that handles the abrupt changes in the input signals and makes sure that the transitions between different Target+1 are smooth.

The transients that can arise in the filter when Target+1 is changed or lost are avoided by a reset signal that sets the state of the Kalman filter to the current measured Host speed and Target range, while the remaining states are all set to zero. This means that the filter completely loses any knowledge about past states, and thus the transients are completely avoided.

This solution for avoiding false transients introduces another problem. The output from the filter will be completely discontinuous at changes or loss of Target+1. A step input to the controller will produce a step in the desired acceleration, producing a negative effect on comfort. To prevent the step changes in filter output, a function is introduced which smooths the output. The smoothing function remembers the state immediately before the Target+1 change and uses a ramp function to weigh between the remembered value and the current output of the filter, to get a smooth transition.

4.4.3 Results from the test track

Three vehicles were used in the testing. The first vehicle, Target+1, uses a regular cruise control and varies the set speed between 50 km/h and 30 km/h. The second vehicle, Target, follows the first vehicle with a single-target ACC. The third vehicle, Host, uses a multi-target ACC to follow the second vehicle. In Appendix A more information about the test environment is given. In Figure 4.14 the behaviour of Host can be seen when single-target ACC is used. In Figure 4.15 multi-target ACC is used. By comparing Figure 4.14 to Figure 4.15, it can be seen that the Target+1 part of the controller gives the controller a faster response compared to if only the Target part would have been used. The range-rate error is also smaller when multi-target ACC is used.

It can be seen that the multi-target ACC behaves well also when the Target+1 information is lost (when the Range goes towards 150) and that this does not cause any bumps in the desired acceleration.

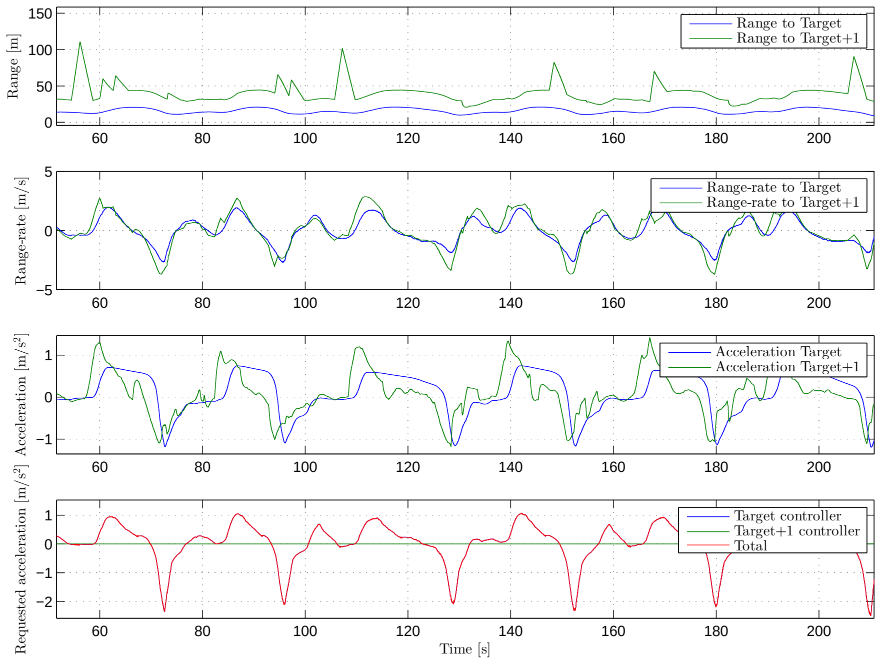


Figure 4.14: In-vehicle test where the Host vehicle uses single-target ACC.

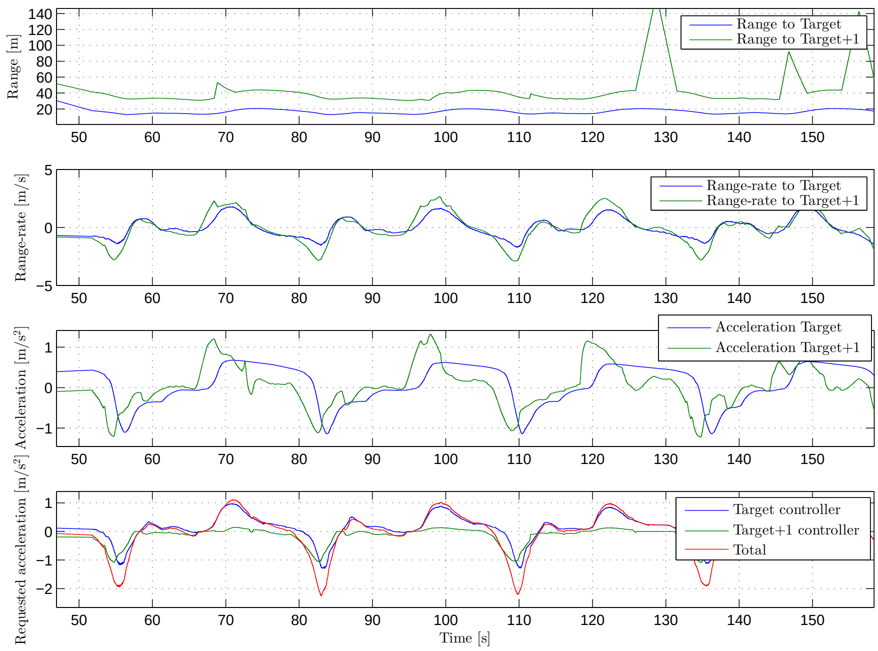


Figure 4.15: In-vehicle test where the Host vehicle uses multi-target ACC.

Chapter 5

Conclusions and future work

In this thesis two different subjects have been discussed; lateral prediction and multi-target control. This chapter presents conclusions for the different parts together with proposals of future work.

5.1 Conclusions

The results in the thesis clearly shows that it is possible to predict the lateral position of detected vehicles a few seconds ahead with a Kalman filter and a predictor. The used Target model with constant acceleration corresponds well to reality at cut-ins and cut-outs. With the modification of the acceleration dynamics in Section 2.4.2, the errors of the predictor can be reduced. Even though the prediction is simple, the errors can be kept at a low level and at the same time it is able to speed up the Target selection.

In Chapter 3 it is shown that a multi-target controller performs better than a single-target controller in many traffic situations. Although the multi-target controller in itself does not make a vehicle string stable, it decreases the error propagation gain, and up to a number of vehicles, the maximal leader error is not exceeded by the followers.

The optimal multi-target controller designed in Chapter 3 performs better than the optimal single-target controller when Target and Target+1 are strongly linked. In the other case, when the link between Target and Target+1 is weak, the behaviour can in some situations be worse than the single-target controller. This knowledge needs to be considered when multi-target control is used in an ACC application.

Multi-target ACC in traffic is beneficial over single-target ACC since the ACC has the possibility to react earlier to changes in traffic speed. This increases both comfort and safety since the reaction time of the ACC is decreased. Great care has to be put in the tuning and the surrounding logic to avoid bad behaviour when the Target vehicle does not follow Target+1. By limiting Host's Target+1 controller in certain situations, unnecessary accelerations can be avoided. Also, making the controller dependent on the time-gap between Target and Target+1 will improve the behaviour. The evaluations of the in-vehicle tests show that the multi-target ACC shows tendencies to improve the behaviour, and the logic for falling back to single-target control works well without causing noticeable effects.

5.2 Future work

The lateral prediction is sensitive to large variance noise with low frequency. A better knowledge on where other vehicles are located relative to the current lane would allow for increased prediction horizon without introducing larger errors. For instance a bicycle model where vehicles are modelled as objects with lateral and longitudinal position, absolute velocity, absolute acceleration, heading angle and curvature could be used. The prediction of lane changes would be two consecutive circular arcs with opposite sign in curvature. This could allow for a more accurate prediction of the lateral position.

In Chapter 2, (2.10) is used iteratively to get a state prediction of the future. While this method of prediction is true to the assumed model, this is not an accurate assumption of what the future state will look like in the real world. A typical lane change does not consist of constant lateral acceleration in two seconds. It is more accurate to assume that the lateral acceleration is constant the first second, and then constant but with opposite sign for the last second. The assumed model of the lateral vehicle dynamics predicts with constant acceleration for two seconds. As a consequence, an error in the estimated state will not be corrected by the predictor, instead the predictor will act as a lever and amplify errors.

Many errors in the lateral prediction occurs close in time to cut-ins and cut-outs. Since vehicles rarely change to a lane and then immediately back again, a timer could be used to inhibit the effect of the lateral prediction. That is, immediately after a lane change the predictor is not allowed to predict the lateral position across the lane boundary for a preset period of time.

Another way to decrease the effect of overshoot is to reset the lateral acceleration in cut-in situations. When a detected vehicle moves from outside the lane to inside the lane, the lateral acceleration is set to zero. A drawback of this method is that a vehicle passing through the lane is in the lane a longer period of time. It is desirable that vehicles just passing through the lane are selected as Target for as short time as possible in order to limit the control effort wasted on the passing vehicle. In Figure 5.1 a state machine can be seen which keeps track of the type of lane change. The event *WithinLane* is a flag set by the Target selection which is true if a vehicle is in the same lane. t is the time period the vehicle has been in the current lane and T is a threshold value. When entering *Within Lane Left* the lateral acceleration is set to $\max(a_y, 0)$. Conversely, when entering the state *Within Lane Right* the lateral acceleration is set to $\min(a_y, 0)$.

Since typical lane changes takes approximately 2 seconds, a comparison could be done of how easy it would be for Target to change lanes compared to staying in the current lane. Instead of directly predicting the lateral position, the lateral

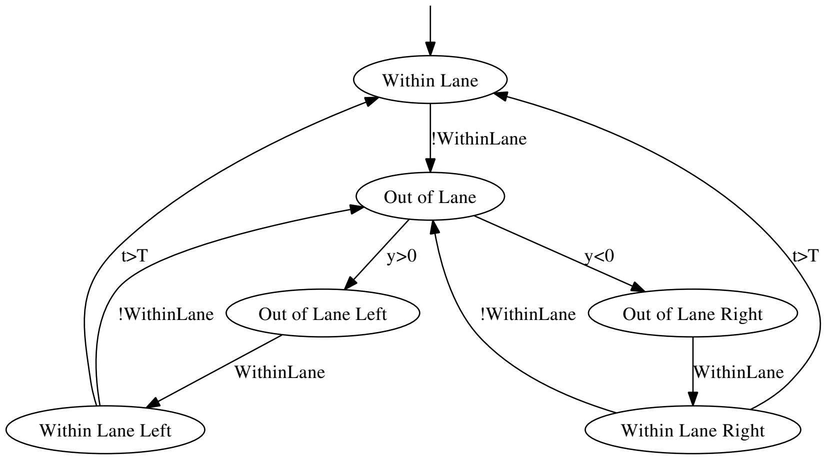


Figure 5.1: The state machine governing the overshoot prediction logic. When entering the state *Within Lane Left* or *Within Lane Right* the acceleration is reset to zero.

acceleration required to stay in the current lane the coming two seconds could be compared to the lateral acceleration required to make a lane change within two seconds. By assuming that drivers want to be subject to as low lateral forces as possible, the lower of the two required lateral accelerations is selected as the most probable manoeuvre. Based on the current lateral position and the required lateral accelerations, a Target.

In normal straight driving the lateral acceleration required to stay in lane would be low, while the lateral acceleration required to change lanes would be large. When a lane change is initiated the lateral acceleration increases, and therefore the lateral acceleration required to stay in the lane increases. On the contrary, the lateral acceleration required to change lanes decreases. At a point the lateral acceleration required to change lanes will be lower than the lateral acceleration required to stay in the lane. If this point in time is ahead of the lane change, this method could be used instead of the predictor.

The multi-target controller could use a state machine to detect different driving scenarios. This would allow for a gain scheduling that depends on the current traffic situation.

To better be able to synthesize and analyse the multi-target controller, a better simulation model would be beneficial. The vehicle model (3.9) is not a perfect model of the dynamics of the vehicle. Depending on whether the vehicle accelerates or decelerates, the time-constant and the delay will be different. By introducing this in the simulation environment more accurate simulations can be achieved.

More in-vehicle tests are needed, both for lateral prediction and multi-target control, to be able to evaluate the performance and tuning, but also to find situations that are not handled correctly.

Chapter 6

Bibliography

Audi (2010), 'New ideas for urban mobility – Audi improves efficiency and safety in road traffic'. Accessed 2012-05-20.
URL: *http://www.audiusanews.com/newsrelease.do?id=1811*

Ferrara, A. and Vecchio, C. (2006), Sliding Mode Control for Automatic Driving of a Platoon of Vehicles, Proceedings of the 2006 International Workshop on Variable Structure Systems, Alghero, Italy, pp. 262 – 267.

Glad, T. and Ljung, L. (2000), *Control Theory: Multivariable and Nonlinear Methods*, Taylor and Francis, London.

Kesting, A., Treiber, M., Schonhof, M., Kranke, F. and Helbing, D. (2007), 'Jam-avoiding Adaptive Cruise Control (ACC) and Its Impact on Traffic Dynamics', *Traffic and Granular Flow, Part VI* pp. 633 – 643.

Klinge, S. and Middleton, R. H. (2009), String Stability Analysis of Homogeneous Linear Unidirectionally Connected Systems with nonzero Initial Conditions, IET Irish Signals and Systems Conference.

Liang, C.-Y. and Peng, H. (2000), 'String Stability Analysis of Adaptive Cruise Control Vehicles', *JSME International Journal* **43**(3), 671 – 677.

Marsden, G., McDonald, M. and Brackstone, M. (2001), 'Towards an Understanding of Adaptive Cruise Control', *Transportation Research Part C* **9**, 33 – 51.

Naus, G. J. L., Vugts, R. P. A., Ploeg, J., van de Molengraft, M. R. J. G. and Steinbuch, M. (2010), 'String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach', *IEEE Transactions on Vehicular Technology* **59**(9), 4268 – 4279.

Paine, M., Paine, D., Griffiths, M. and Germanos, G. (2007), In-vehicle Intelligent Speed Advisory Systems, Proceedings of the 20th International Conference on the Enhanced Safety of Vehicles, Lyon, France.

Peppard, L. E. (1974), 'String Stability of Relative-Motion PID Vehicle Control Systems', *IEEE Transactions on Automatic Control* **19**(5), 579 – 581.

Rapaport, A. and Astolfi, A. (2004), 'A Remark on the Stability of Interconnected Nonlinear Systems', *IEEE Transactions on Automatic Control* **49**(1), 120 – 124.

Robinson, T., Chan, E. and Coelingh, E. (2010), Operating Platoons on Public Motorways: An Introduction to the SARTRE Platooning Programme, 17th World Congress on Intelligent Transport Systems, Busan, Korea.

Seiler, P., Pant, A., and Hedrick, K. (2004), 'Disturbance Propagation in Vehicle Strings', *IEEE Transactions on Automatic Control* **49**(10), 1835 – 1841.

Shaw, E. and Hedrick, K. (2007), Controller Design for String Stable Heterogeneous Vehicle Strings, Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, pp. 2868 – 2875.

Sheikholeslam, S. and Desoer, C. A. (1993), 'Longitudinal Control of a Platoon of Vehicles with no Communication of Lead Vehicle Information: A System Level Study', *IEEE Transactions on Vehicular Technology* **42**(4), 546 – 554.

Sheikholeslam, S. and Desoer, C. A. (1996), Longitudinal Control of a Platoon of Vehicles, American Control Conference, pp. 291 – 296.

Swaroop, D. (1997), String Stability of Interconnected Systems: An Application to Platooning in Automated Highway Systems, PhD thesis, University of California, Berkeley.

Swaroop, D. and Hedrick, J. K. (1996), 'String Stability of Interconnected Systems', *IEEE Transactions on Automatic Control* **41**(3), 349 – 357.

Teel, A. R. (1996), 'A Nonlinear Small Gain Theorem for the Analysis of Control Systems with Saturation', *IEEE Transactions on Automatic Control* **41**(9), 1256 – 1270.

Xiao, L. and Gao, F. (2011), 'Practical String Stability of Platoon of Adaptive Cruise Control Vehicles', *IEEE Transactions on Intelligent Transportation Systems* **12**(4), 1184 – 1194.

Zhou, J. and Peng, H. (2005), 'Range Policy of Adaptive Cruise Control Vehicles for Improved Flow Stability and String Stability', *IEEE Transactions on Intelligent Transportation Systems* **6**(2), 229 – 237.

Åström, K. and Wittenmark, B. (1997), *Computer-Controlled Systems: Theory and Design*, third edn, Prentice Hall, Upper Saddle River.

Appendix A

Test environment

VCC has special equipped vehicles which are designated as test cars. These cars are ordinary, fully outfitted cars with a prototyping system in the trunk. The prototyping system's main task is to function as a replacement of a control unit in the car. The functionality which should execute in the control unit is modelled and designed in Simulink models. By compiling the models to low level machine code, the functions can be tested in real-time in the car.

A.1 Sensors

The sensors used by the ACC are the forward looking radar and the forward looking camera. These sensor modules are supplied by third party manufacturers and communicates with control units in the car via CAN. On CAN, the sensors communicate the angle, range and range-rate to all detected objects. All units which are connected to CAN may therefore access the information of detected objects.

A.2 Rapid prototyping

The prototyping system in the trunk of the car is called Autobox which is a real-time system with much higher processing capacity than the control units in the vehicle. The Autobox is connected to CAN and therefore has access to all information from the sensors.

The Autobox has been used to test the lateral predictor and the extended functionality of the ACC. The Autobox is loaded with low level machine code, which means that in order to test implementations and parameters the functions under test are needed to be compiled by a C++ compiler. All functionality has been implemented in Simulink.

For the purpose of the thesis, a Simulink model of the Target selection and the ACC was provided by VCC. This model was modified and extended to house the lateral prediction and the multi-target ACC. From the extended model, new C++ code was generated and compiled to low level machine code, and downloaded to the Autobox.

A.3 Logging data

The Autobox has, in addition to connections to the CAN, the possibility to communicate through TCP via Ethernet connections. It is possible to monitor all variables and signals which are used in the original Simulink model in real time. It is also possible to change relevant parameters to see what effect they have on performance. All relevant signals were recorded and saved for further analysis.

Both the input and output from the tested functionality were logged. The recorded input was used as input for the test environments, and the resulting simulated output was validated by comparing it to the recorded output.