Title

Enhanced AHS Safety Through the Integration of Vehicle Control and Communication

Authors

Hedrick, J. K.
Uchanski, M.
Xu, Q.

Publication Date
2001-10-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Enhanced AHS Safety Through the Integration of Vehicle Control and Communication

J.K. Hedrick, M. Uchanski, Q. Xu
University of California, Berkeley

California PATH Research Report
UCB-ITS-PRR-2001-28

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

Final Report for MOU 388

October 2001

ISSN 1055-1425

CALIFORNIA PARTNERS FOR ADVANCED TRANSIT AND HIGHWAYS

Enhanced AHS Safety Through the Integration of Vehicle Control and Communication

Final Report

PATH MOU 388

J.K. Hedrick
M. Uchanski
Q. Xu

Mechanical Engineering Department
University of California at Berkeley
Berkeley, CA 94720

Abstract

A new architecture for a mobile ad-hoc communication network is developed, and simulation results show that the proposed MAC protocol can enhance transmission success probability from 0 to 97%.

Two novel applications of vehicle-to-vehicle networks are then developed and simulated. The first application is a “Cooperative Adaptive Cruise Control” which uses communicated information to improve on ordinary cruise control systems. Communicating a “virtual brake light” between vehicles is shown to decrease the accelerations required to maintain safe following distances in cut-in and hard braking situations. The second application is a “Cooperative Estimation” algorithm. The idea behind this algorithm is that each vehicle on the road is potentially a driving condition “sensor.” By combining data communicated from many vehicles on the roadway, the cooperative estimation algorithm is able to produce estimates of driving time and road condition that are significantly better than those that any one vehicle could produce on its own.

While developing the cooperative road condition estimation algorithm, it is found that the problem’s crux is at the vehicle level, so the friction estimation problem is then investigated at a vehicle/tire level through extensive experimentation. An exciting new type of “slip-based” road condition estimator is developed and experimentally shown to be able to distinguish between wet and dry roads using no dedicated road condition sensors sensors. In-depth literature review complements our own results and shows that “slip-based” estimators of the type developed may be very useful to help solve the AHS road condition estimation problem.

Keywords: Ad-hoc networks, Communication architecture, Cooperative adaptive cruise control, Cooperative estimation, Friction coefficient, Slip, Tires

Executive Summary

This final report presents research findings obtained under PATH MOU 388, a two-year project devoted to catalyzing AHS (Automated Highway System) deployment and improving AHS safety through the integration of communication and control.

One of the main factors separating true *intelligent highway systems* like the PATH concept from *collections of intelligent cruise control vehicles* is the presence of an inter-vehicle communication network to coordinate vehicle maneuvers. Unfortunately, automotive communication networks have been slow to appear on roadways, largely due to two problems: First, the networking technology to handle highly mobile networks has been slow to develop, and second, few applications have been developed that use these networks.

To address the network-technology problem, we conducted an extensive review of the state-of-the-art for mobile networks, a summary of which we present in this report. Based on results of this review, we developed a new communication protocol for vehicle-to-vehicle ad-hoc networks. Simulation results that we present here show that the new proposed MAC protocol can transmit with probability of over 97% the messages which cannot get through in the old protocol.

To help solve the application problem, we developed two new ways for vehicle-to-vehicle networks to improve highway safety and convenience. The first of these applications,“Cooperative Adaptive Cruise Control,” uses inter-vehicle communication to improve on adaptive cruise control systems. Simulation results in this report show that during cut-in and hard braking situations, a communicated “virtual brake light” significantly decreases the accelerations that the adaptive cruise control system needs to maintain safety distances. The second of these applications—a new concept, which we dub “cooperative estimation”—uses inter-vehicle communication to generate estimates of important driving quantities that are more useful than any vehicle on the highway could produce by itself. We apply the cooperative estimation concept to travel time estimation and road condition estimation, and simulation results show that inter-vehicle communication can significantly improve travel time and coefficient of friction estimates.

Estimating road condition is of particular importance for AHS, so in addition to developing the cooperative road condition estimator, we also pursued the road friction estimation problem through experimental investigations at the vehicle level. The fruit of these investigations is a new road condition estimator that works during braking and avoids using dedicated road condition sensors. Results from our own on-vehicle extensive testing, combined with results from a handful of new papers in the literature, indicate that road condition estimators similar to ours have strong potential to deliver enhanced highway safety safety at minimal added cost.

Contents

List of Figures

Chapter 1

Introduction

This final report presents research findings obtained under PATH MOU 388, a two-year project devoted to catalyzing AHS (Automated Highway System) deployment and improving AHS safety through the integration of communication and control.

One of the main factors separating true *intelligent highway systems* like the PATH concept from *collections of intelligent cruise control vehicles* is the presence of an inter-vehicle communication network to coordinate vehicle maneuvers. Unfortunately, automotive communication networks have been slow to appear on roadways, largely due to two problems: First, the networking technology to handle highly mobile networks has been slow to develop, and second, few applications have been developed that use these networks. Research under MOU 388 has been aimed at solving these two problems.

In Chapter 2, we address the network design problem, first with a state-of-the-art survey, and then with an original "ad-hoc" network design and simulations.

In Chapters 3 and 4, we address the problem of lack of applications by developing two systems that use inter-vehicle networks to improve highway safety and efficiency. The first of these applications—"Cooperative Adaptive Cruise Control," which is described in Chapter 3—-uses inter-vehicle communication to improve on adaptive cruise control systems. Simulation results in this chapter show that during cut-in and hard braking situations, a communicated "virtual brake light" significantly decreases the accelerations that the adaptive cruise control system needs to maintain safety distances.

The second of these applications—a new concept, which we dub "cooperative estimation" and describe in Chapter 4—uses inter-vehicle communication to generate estimates of important driving quantities that are more useful than any vehicle on the highway could produce by itself. We apply the cooperative estimation concept to travel time estimation and road condition estimation, and simulation results show that inter-vehicle communication can significantly improve travel time and coefficient of friction estimates.

Estimating road condition is of particular importance for AHS, and it turns out to be a much more in-depth problem than our high-level statistical treatment of it in Chapter 4, "Cooperative Estimation," would indicate. In Chapter 5, "Slip-based Road Condition Estimation," we recount our experimental investigations into the quite involved vehicle-level road condition estimation problem. We follow a so-called "slip-based" approach to road condition estimation, and the fruit of our investigations is a new road identifier that works during braking and requires no dedicated road condition sensors. Results from our own on-vehicle extensive testing, combined with results from a handful of new papers in the literature which we review in this chapter, indicate that road condition estimators similar to ours have strong potential to deliver enhanced highway safety safety at minimal added cost.

Finally, in Chapter 6, we summarize the main findings of our work, and make recommendations for future directions of research.

Chapter 2

Research on Network

The increasing trend from centralized control algorithms towards decentralized, distributed real-time control implementation has greatly stimulated the research in the field of communication networked control system. A lot of control applications require the controllers, actuators, and sensors to be modulated into nodes and be distributed in space. A communication network, wired or wireless, is necessary in transferring signals between nodes. Using a network for control has the potential of making the system modular, enhancing robustness and simplifying wiring for some applications. One of the main features of the communication network is the shared communication resource by many nodes. This feature is essential for the efficient use of the communication channel, however it also makes data delay and data loss unavoidable. The randomly distributed time-various delay and unpredictable data loss deteriorate the performance of the conventionally designed controllers and result in instability. Therefore we are facing a two-folded problem. On one side, although existing network technique is successfully used in data transfer applications such as in Internet, a lot of work still need to be done to make it suitable to the real-time application as in control systems. On the other side, we can also extend the conventional theory and develop new control methodology to deal with the network-induced data delay and loss.

In Part 2, we summarize the work done in the communication architecture part of MOU388. First in section 2.1 is a survey of the previous research work in the field of networked control system. The scope of this survey is beyond the field of vehicle communication network and control. However it provides us with the insight in the field of networked control system both in depth and in width. The approaches suggested by various researchers give us hints and problem to work in our own research in the similar work in MOU388. No other comparable large scale literature review has been found in publication yet.

Then in section 2.2 we describe the network design and analysis work we have done in MOU388, which includes the architecture design ,the Medium Access Protocol design and network simulation using Network Simulator software.

2.1 Literature Review

In the literature, three kinds of networks have been considered for synchronous control, i.e., synchronous, asynchronous, and isochronous. The synchronous network delivers control information at constant rate at regular time intervals. The asynchronous network delivers information reliably with variable delay. TCP or the IEEE802.11 ad-hoc mode are the most common examples. In this kind of networks the transmission build queues for the data: new data is queued in behind the old data in a FIFO sense. The queues make the delay correlated. The isochronous network delivers data at a constant rate and discards the undelivered old data.

The synchronous network is an ideal case and does not exist in real application. The work in [9] [27] [35] [36] [37] [38] [39] study the problems related to asynchronous network. Isochronous network is studied in [45] and [46].

In the past a few years researchers in both the communication network and control theory attributed to the progress. The proposed design methodology range from new communication protocol, state estimation to μ-synthesis. The application fields include vehicle control, manufacturing, aircraft control, and telerobotics. Here we give a survey of the papers we found in the major control and communication periodicals. The first part of the survey— 2.1.1 is about the research from the viewpoint of communication network, such as the requirement for the network in the real-time control applications, the comparison of suitability for control applications of existing communication protocols, and the new protocol proposed specifically for real-time control. Subsection 2.1.2 consists of the research on the control side, such as the stability condition, controller design and state estimators. Finally in 2.1.3 we summarize the survey.

2.1.1 Communication network for control

Unlike the commonly used data transfer network, the network in control system has its specialty. One basic reason for this specialty is the real-time requirement of the control network, in contrast to the non-time critical applications in the general data networks such as browsing web or sending are receiving emails. Several papers study the requirement of the control system to the communication network, the applicability of existing network techniques, and the possible new methods to overcome the shortcoming of old ones. We describe these studies in this subsection.

Requirements of the communication network

Several papers discuss this problem from different point of view. Paper [34] qualitatively gives a presentation on how the control network is special in the sense of protocol, interoperatability, command status, topology, addressing, security, management and monitoring. It also gives a survey of the existing "standards" proposed by different industrial groups. In other papers [26] [27] [45] [47] the requirements of the control network are also touched upon. Summarized, some of the most general requirements for communication network used in control system are:

1. Unlike data network, which uses large data packets, and relatively infrequent bursty transmissions to send large data files, a control network must be able to send countless small data packets frequently to many nodes. [27] [34]

2. A message should be sent reliably (very low loss rate) with bounded delay. [27]

3. Queues are not desirable. If the new sampling data is available, the old data that has not been transmitted can be discarded. [45]

4. Protocol must add small overhead to the network performance. [26]

Comparison of the existing protocols

Several research groups compared the suitability of the currently existing protocols in the control network. Here we first introduce the concepts of the different protocols, then the comparison methodology and the results.

The protocol we discuss here refers to the mechanism of the network to control the access of individual node to the medium, i.e. the way to determine which node could transmit the data at any instant. The purpose for implementing such protocol is to mitigate the collision of the data packets sent by different nodes and to deal with the collisions when the collisions do happen. The protocol can be crucial in determining the data delay, data loss and hardware requirement of the network. Several protocols are proposed to meet the requirement for control network. Those have been studied include, Ethernet (IEEE 802.3: CSMA/CD), Token Bus (IEEE 802.4), Token Ring (IEEE 802.5), and Controller Area Network (CAN: CSMA/AMP). The former three are introduced in both [27] and [47], and can also be found in many common computer network texts. Additionally, Tilbury, et al introduced and studied CAN in [27]. Some discussion of the token passing protocols can also be found in [26].

1. *CSMA/CD (Carrier Sense Multiple Access/ Collision Detection* is specified in IEEE 802.3 standard. The most common implementation of CSMA/CD is Ethernet, which is widely used in Local Area Network (LAN). In CSMA/CD, whenever a node wants to transmit, it first listens to the network for a carrier. If the network is idle, namely no other node is transmitting, the node start transmitting right away. When two or more nodes sense the idle network at the same time they transmit simultaneously, and collision will happen. The node listens to the network at the same time of transmitting. When it detects collision, it stops transmitting immediately, and the collided messages are discarded. The node then backs off for a certain time to retransmit the message. There is a standard algorithm (BEB) which makes the node back off for a longer time as it experiences more collisions, which is the signal of busy data traffic in the network. After a certain times of collision, the message will not be retransmitted any more and will be discarded. CSMA/CD does not support priority of message. The size of data packet frame is between 46 and 1500 bytes. The total overhead is 26 bytes.

2. *Token Ring (IEEE 802.5)* consists of a physical ring. Actually it is a collection of point-to-point connection which forms a ring. A 3-byte frame called token is passed around in the ring. When a node wants to send message, it waits for the token to arrive. When the token is

passed to this node, it seizes the token and starts transmitting. The token is released either when the node finishes transmitting, or when the maximum time for the node to hold the token is reached. The message is sent point-to-point along the ring. Each node copies the message and forwards it to the successor in the ring. If the message is destined to the node, it is transferred to the upper level application, otherwise it is just passed by the node without any modification. The token ring needs a monitor node. Priority can be supported.

3. *Token Bus (IEEE 802.4)* uses the same mechanism as token ring. The difference is that the nodes form a virtual ring. The actual location of the node does not matter, but each node knows its logical predecessor and successor. The data is passed in the virtual ring in the same manner as in the token ring. Priority is also supported by token bus.

4. *CAN (Controller Area network) Bus* uses CSMA/AMP (Carrier Sense Multiple Access/Arbitration on Message Priority) medium access mechanism. It is optimized for short messages and is message oriented. As in CSMA/CD, the nodes wait for the network to be idle to transmit. The initial part of the message frame is the arbitration field containing the priority of the message. When more than one nodes send simultaneously the arbitration is performed and the message with the highest priority wins the access to the network. The example introduced in [27] is DeviceNet whose frame has 47 bits overhead and 0-8 bytes data. The mode to transmit message is multicast, i.e. the message is sent to all the nodes in the network but only the destined ones accept it. Multicast is a transmission mode between unicast, where the message is sent to only one destined node, and broadcast, where the message is sent to all the nodes connected by the network.

In [47], three LAN protocols, token bus, CSMA/CD, and token ring, are qualitatively compared. Their conclusion can be summarized as:

1. *When the communication load is low, CSMA/CD works best among the three.* This is because that in CSMA/CD, the nodes use no bandwidth in getting access to the network. Whereas in both of the token passing protocols, token passing itself uses a great portion of bandwidth when the communication load is low. *When the communication load is high, the performance of CSMA/CD decreases quickly due to the higher*

frequency of collision. The loss rate also goes up as the load is increasing because packets are more likely to be collided for many times and be discarded eventually. The general traffic of control network is high as mentioned in 2.1.1, thus CSMA/CD is unsuitable to the control network.

2. As to the two token passing schemes, token ring suits fiber optics better. However it makes no difference for the wireless network of the vehicle control system we are interested in. *The token bus has the advantage of being easily reconfigurable.* Nodes can be dynamically added or deleted from the network. *A disadvantage of token bus is that when there are too many nodes in the network, a substantial amount of bandwidth will be wasted in passing the token.* This effect in vehicle context has been studied by Sonia Mahal in [28].

Three examples of the protocols: Ethernet (CSMA/CD), ControlNet (token bus) and DeviceNet (CAN bus) are introduced and compared in [27]. First shown is the comparison of the three protocols with different data packet size. The result indicates that CAN, as designed to transmit small data, works best when the packet size is small and worst for large size packets. The reason is that CAN has smallest overhead in data frame among the three. At the same time, the maximum data size in a packet is 8 bytes for DeviceNet. When the data to be transmitted is large, DeviceNet must fragment it before sending, thus more overhead is added.

Then the timing performances of the three protocols are compared in [27]. The timing parameters considered in the paper are:

Average Time Delay Control network requires the time delay be bounded and small.

Efficiency of the Network The ratio of the total data size to the time used to send messages containing the data.

Utilization of Network The ratio of total time used to transmit data and the total running time.

Number of Unsent Messages Control network requires the information to be sent reliably, so the number of unsent messages should be as small as possible.

Stability of Network When the number of messages in the queue of each node grows infinitely, the network is unstable.

These parameters are compared for various frequency of the messages to be sent. The results show that before the saturation of the network traffic load, the average time delays of the three networks are about the same. The efficiency of the three networks can reach 100% in this case. The utilization of Ethernet approaches 100% slowly as the frequency of the messages increases. ControlNet and DeviceNet both have 100% utilization when the network is saturated. However in these 2 protocols higher priority messages are transmitted while lower priority messages are always left in the buffer. Both Ethernet and DeviceNet have a great amount of unsent messages as the network is saturated. The ControlNet has lower number of unsent messages due to the available bandwidth of each node.

New Protocols

Some new protocols are proposed to overcome the shortcoming of the existing ones. In [45], a new Try-Once-Discard protocol is discussed to meet the requirement 3 in 2.1.1. It is meant to dynamically allocate network resources to those information sources with time-critical information. The detail will be covered in 2.1.2 below when we discuss the paper.

Another token passing protocol called priority token passing is described in [26]. The new protocol has a switching function, i.e. it can change network topology from ring to broadcast medium. The token holding node first broadcasts a pass frame which soliciting all the nodes requesting to receive the token. Then these nodes compare the priorities of their frames, and the one with the highest priority frame wins the competition. Next, the token is passed to this node and the acknowledgement is transmitted to the former token holding node. The author describe a hardware realization of this protocol. Experimentally they show that this protocol preserves the advantage of the commonly used token bus and token ring protocols but adds a smaller overhead. Therefore it is benefit to meet the requirement 4 in 2.1.1

2.1.2 Networked Control System

There are also many researchers in the field of control working on the control system with networked-induced data delay/loss. Many conventional methods in control theory are extended and new methods are developed. We can find theoretical analysis, simulation as well as experiment results in published papers. Since most of the proposed methods are problem specific and at this stage it is hard to tell which one is superior to the others, we introduce the proposed methods one by one in this section.

Stability study of networked control system

Walsh et al studied the stability of a nonlinear control system whose output sensors are distributed in many nodes and the nodes compete to transmit the new output measurement to the controller via a communication network [45] [46]. They assumed no propagation delay, error and a noise free medium, continuous time controller, and continuous availability of the new sensor data. They designed the controller without considering the network, i.e. in the control system is stable in the absence of the network. They are primarily interested in the effect of the time delay, which is dependent on the packet scheduling protocol, on closed-loop stability.

The plant dynamics is

$$\dot{x}_p(t) = A_p x_p(t) + B_p u(t) \tag{2.1}$$

$$y_p(t) = C_p x_p(t) \tag{2.2}$$

The dynamics of the controller is

$$\dot{x}_c(t) = A_c x_c(t) - B_c \hat{y}_p(t) \tag{2.3}$$

$$u(t) = C_c x_c(t) - D_c \hat{y}_p(t) \tag{2.4}$$

where $\hat{y}_p(t)$ is the most recent output of the plant received by the controller.

The error signal defined by

$$e(t) = y_p(t) - \hat{y}_p(t) \tag{2.5}$$

is the model for the network effect of networked control system. When $t \in [t_i, t_{i+1}]$, we have $\hat{y}_p(t) = y_p(t_{i+1})$. The stability condition is derived to determine the maximum allowable interval $\tau = t_{i+1} - t_i$.

In Theorem 1 of the paper, the stability condition for one-packet transmission is given using the Lyapunov stability of the original control system (2.1) (2.2). One-packet transmission means that all output signals are lumped and transmitted in one packet. Thus the scheduling of the network is not considered here. When the maximum allowable interval satisfies the requirement the networked control system is stable in the sense of Lyapunov.

Then the scheduling protocols are discussed. They argue that a static scheduling method, such as the token passing protocol, though guarantees fairness among the nodes, is brittle towards the unexpected events. This is due to the static scheduling protocol's nature of not considering packet's priority. Thus they propose a new dynamic scheduling protocol called Try-Once-Discard. In this protocol the sensor with the maximum (normalized) weighted error between the data to be transmitted and the most recently transmitted data 2.5 has the highest priority and wins the right to send data. If a data packet fails to win the network access, it is discarded and new data is used next time, thus the protocol is named TOD.

Finally the stability condition in term of maximum allowable transmission interval for both static and dynamic scheduling (TOD) is given in Theorem 2 of [45].

The comparison of the packets' priority is done by the arbitration of the priority bits in the packets' header. In [45], they assume that the priority levels of the data packets are continuous, i.e. there are infinite bits available in the header to indicate the priorities. However this is not the case in the real network, where only a few bits can be used to label the priorities. Thus in [46] they study the TOD protocol with finite discrete priority levels. In this case they cannot give the stability condition but only a condition to guarantee the ultimate uniform boundedness of the system.

Their approach has its limitation in that the network is neglected in the design of the controller. Hence the conditions are only sufficient for stability or boundedness, and tend to be too conservative. For example, one simulated system loses stability when the average delay is 0.015 seconds while the theoretical delay condition presented is 0.001.

Control using state estimator/predictor

Chan and Ozguner study a automotive control systems in [9]. In their system, the output of the system is feedback to the controller via a queued network. The sensor and controller sample the data at the same constant rate. The sensors are not distributed, therefore the output vector is sent in one packet. They assume the existence of a FIFO queue in at the sensor side. When the network is busy, the delayed data is stored in the queue. They propose to transmit the length of the queue together with the data. In this way the

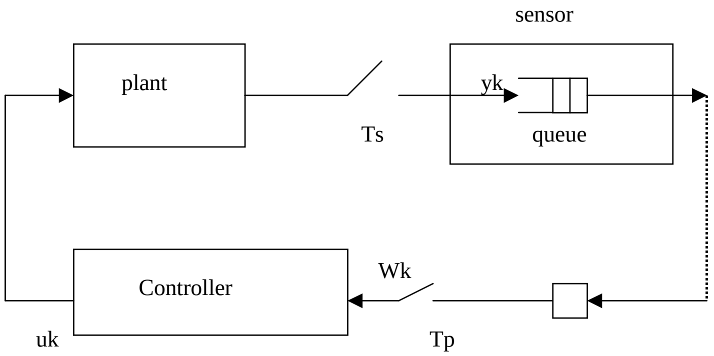

Figure 2.1: networked control system with queue on the sensor

"age" of the delayed output data can be determined in a probabilistic way and the state of the controller can be derived from the evolution equation and the input. The details are shown below.

The system is shown in figure 2.1. On the sensor there is a first-in-first-out (FIFO) queue. In every T_s interval the sensor reads the plant output and puts it in the queue. The data in the queue is sent to controller through a communication network where a random delay is introduced. Zero data loss rate is assumed. There is a single register on the control processor side. If new data arrives before the data in the register is processed, the old one is overwritten. In every T_p interval the controller reads the data ω_k in the register and uses the data to compute the input to the plant. When no new data arrives the latest data is used. T_p and T_s are assumed to be the same in this paper. The time frames of sensor and controller have a skew Δ .

The authors propose to transmit not only the data in the queue, but also the length of the queue. In this way we can know both the value of the data and the "age" of the data. The control system is the following:

$$x_{k+1} = Ax_k + Bu_k \tag{2.6}$$

At the k^{th} sampling instant, if the queue length is known to be j, then in lemma 1 of [9] it is said that the data received is either y_{k-j} or y_{k-j+1}. Basically a output (sensor) signal is received by the controller in the time interval $[(k-1)T + \Delta, kT + \Delta]$, which includes the time kT when new sensor signal is generated and stored in the queue. If the sensor signal arrives at the controller in the interval $[(k-1)T+\Delta, kT]$ then the newest data in the queue is y_{k-1}, then with the length j we know the data received by the controller is y_{k-j}. On the other hand, if the sensor signal arrives in the interval $[(kT, kT+\Delta]$, then the newest data in the queue is y_k, and the data received thus is y_{k-j+1}. There is a probability to determine which of these two cases the data is in. On basis of this probability they derive the best estimation of the state of the system. The following is the mathematical description of their approach.

If $\omega_k = x_{k-j+1}$, then we can predict that

$$\hat{x}_k = A^{j-1}\omega_k + W_j$$

where

$$W_i = \begin{bmatrix} B & AB & \dots & A^{i-2}B \end{bmatrix} \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-j+1} \end{bmatrix}$$

If $\omega_k = x_{k-j}$, then $\hat{x}_k = A^j\omega_k + W_{j+1}$. Thus the best estimation of the state is

$$\hat{x}_k = P_0(A^{j-1}\omega_k + W_j) + P_1(A^j\omega_k + W_{j+1})$$

where the optimal weight matrices P_0 and P_1 are determined by the system and the probabilities

$$p_0 = Pr\{\omega_k = x_{k-j+1}|n = j\}$$

and

$$p_1 = Pr\{\omega_k = x_{k-j}|n = j\}$$

Two simulation examples are given. The first is the vehicle speed control, and the second one is active suspension control.

From above description we can see that the approach is not suitable for the case where the data loss rate is not negligible. However in the wireless communication network as we use in the vehicle communication in AHS, data loss is unavoidable. The suitability of FIFO queue model is also a issue to discuss. As we mentioned in 3 of 2.1.1, a queue is not preferable in a communication control system. The simulation result of this paper also shows that the proposed approach does not work well in nonlinear system where simulation shows that the estimator causes unnegligible error in the performance. Also notice that like in [45] the controller here is also designed without considering the network effect. The

Reducing Communication Traffic

In [48], Tilbury et.al. try to solve the problem in another way round. In most of the proposed approaches in the field, either controllers are designed to account for the network-induced delay, or communication algorithm is developed to decrease the data delay and data loss. Whereas in this paper the authors propose a method to reduce the communication needed for the control system to work. Since the main reason of the data delay/loss in network is the communication traffic, reducing the necessary communication could greatly reduce the delay/loss. The task of reducing communication is mainly achieved by a local state estimator.

The system to be controlled is a LTI, causal discrete time system as follows:

$$\begin{aligned} X(n+1) &= AX(n) + BU(n) + BD(n) \qquad (2.7) \\ Y(n) &= CX(n) \qquad (2.8) \end{aligned}$$

with

$$\begin{aligned} U(z) &= K(z) \cdot E(z) \qquad (2.9) \\ &= K(z) - [R(z) - (C \cdot X(z) + N(z))] \qquad (2.10) \end{aligned}$$

where R, D, N, K are reference, disturbance, sensor noise, and closed-loop controller gain, respectively.

This MIMO system is implemented in distributed manner such that the i^{th} node contains the sensor which produces the i^{th} output and actuator which receives the i^{th} input from the controller. Thus each error E_i or output Y_i must be communicated to other nodes via network and feed to the controller. The authors propose a local state estimator for each of the nodes. Under normal condition the i^{th} node uses the estimated value of the errors of the other nodes $\hat{E}_j$ $(j \neq i)$ in the controller, and the estimated i_th output $\hat{Y}_i$ is compared with the actual output Y_i, which are both locally available to the i^{th} node. When the difference exceeds a threshold,a local communication logic command the actual output Y_i or error E_i to be broadcast to all the system, and the estimated value of the state is updated. In this way the communication in the network is greatly reduced. The BIBO stability of the proposed estimator is proved in the paper.

The simulation of a two-axis milling machine is shown as the example. The x-axis and y-axis components act as the distributed nodes. The results indicate that a great amount of bandwidth can be saved. For example, when the communication logic threshold error is 20% of process gain, less than 25% communication is needed to achieve 99% of the optimal performance (100% communication means communicating at every sampling time).

The highly reduced communication load could also decrease the network-induced delay, as the main source for the delay is the communication traffic. The authors do not state what specific communication protocol is used in the simulation. The effect of the proposed estimator on the existing communication protocols should be an interesting topic of study. The proposed method is not applicable to static scheduling network protocols such as the token ring and token bus.

Robust control of the delayed system

The same authors describe in [18] and [19] the μ-synthesis approach to deal with the network induced delay. The delay is assumed to be bounded and modelled as perturbation to the delay-free control system. A modified Pade approximation is used to model the random delay as following:

$$e^{-\tau\Delta s} \approx 1 - \frac{\tau s}{1+\frac{\tau}{2}s}\Delta = 1 + w(s)\Delta \tag{2.11}$$

where $|\Delta| < 1$ and

$$w(s) = \frac{\tau s}{1+\frac{\tau s}{3.465}} \tag{2.12}$$

μ-synthesis tool is then used to design the controller.

In [19], a robotics experiment is described. The robot is teleoperated to follow a preset path. The control algorithm is run on a remote computer. An on-board camera grabs the image of the environment and the image frames are sent to the remote computer as the robot moves. Thus multiple communication channels are used for this multi-media application. They compares the performance of TCP/IP, ATM and TCP/IP over ATM connections with the proposed control algorithm being implemented. The results of mean square error and maximum error are shown. TCP/IP over ATM is better than raw ATM, whereas ATM is better than pure TCP/IP. The authors claim TCP/IP over ATM has disadvantage where suddenly jumps are seen in the loop delay due to the flow control mechanism of TCP/IP.

μ-synthesis and H_∞ control are very interesting alternative approaches to the random delayed control problem. Some current research of PATH is related to them ([42] and its continuing work). We could look more into the application of relevant theories in the future work.

Predictive controller

Predictive controller is proposed in the networked control system in [5] to deal with unbounded delay. The system consists of the plant's side, which is supposed to be remote such as in outer space, and operator's side, which is to give command via communication channel to control the remote plant. On plant's side there is a local controller, which stabilizes the plant in absence of constraints. The output of the plant is feedback to the operator's side via feedback communication channel. Reference of performance, or equivalently, the desired trajectory, is given on the operator's side. There is a predictive controller implemented on the operator's side which receives desired trajectory and provides command inputs. The inputs are then sent to the plant's side over feedforward communication channel. Both feedforward and feedback channels may introduce unbounded delay to the signals they transmit.

The main idea of predictive controller is to use a model of the plant to predict the future evolution of the system and accordingly select the commanded input. A virtual sequence of future control action is then generated. Typically only the first sample(s) of this sequence are used and the rest are discarded. However, the authors propose to recover the last available virtual input sequence from the "trash can" when the real input is delayed, and apply the virtual sequence to the plant.

Let the primal system be of the form

$$\begin{aligned} x'(t) &= \phi(x(t), w(t)) && (2.13) \\ y(t) &= h(x(t), w(t)) && (2.14) \\ c(t) &= l(x(t), w(t)) && (2.15) \end{aligned}$$

In the equations above, $x(t)$ is the state of the system. When the system is discrete-time, $x'(t) = x(t+1)$, while when it is continuous $x'(t) = \dot{x}(t)$. $w(t)$ is the input. $y(t)$ is the output which is required to track the reference

$r(t)$. $c(t)$ is the vector to be constrained in a given set. The stability of the primal system is assumed because, as mentioned above, there is a controller local to the plant which stabilizes the system in absence of constraints. Let the sequence of the future control input be denoted by $[\ v(0),\ \ v(1),\ \ \dots\ ]$, then we can write them in the form of a vector θ. For instance

$$\begin{aligned} \theta &= [\ v'(0),\ \ v'(1),\ \ \dots,\ \ v'(N_u - 1)\ ]' &(2.16)\\ v(j) &= v(N_u - 1),\ \forall j = N_u, N_u + 1, \dots &(2.17)\end{aligned}$$

At each time t, an optimal sequence of future input is evaluated by solving the optimization problem

$$\theta^*(t) = \begin{cases} arg\ min\ J(t, x(t), r(t), \theta) \\ subject\ to\ c(t) \in C \end{cases} \tag{2.18}$$

where J is a performance index which depends on the predicted evolution of the primal system due to initial state $x(t)$ and input $w(\tau) = v(j)$, $\tau \in [t + j\Delta T, t + (j + 1)\Delta T]$, where ΔT is the sampling period and τ is an integer for discrete time system. Typically J is obtained by summing/integrating the squares of the tracking errors $y - r$ and inputs v on the interval $(t, t + N_y\Delta T)$. Eventually, additional constraints are taken into account, for instance terminal state constraints $x(t + N_y\Delta T) = 0$. Then in ordinary operation, we have the command input of the primal system as $w(t) = v^*(0)$, where for continuous time system the input $w(t)$ is held constant for the time interval $[t, t + \Delta T)$.

The approach proposed in the paper is to apply the last available virtual future control sequence $\{v(0), v(1), \dots\}$ to the plant whenever the feedforward delay makes the actual input $w(t)$ unavailable at time t. The virtual control sequence can be applied to the system "safely" in the sense that although the performance of the controlled plant may not be as desired, the constraint of the system will not be violated. The system then is said to be in the recovery mode. Because the virtual input is actually a semi-infinite sequence starting at the time it was derived, t, this approach can keep the system's state stay within the constraint set even when infinite delay is present. When the synchronous input is available again the system will switch back to the normal mode.

The predictive controller is used here only to satisfy the constraint requirement in the presence of delay. Thus the objective of the problem is different from what we are interested. Although the system can endure unbounded delay without violating the constraints, it is not likely for our application to use a communication architecture where such large delay can occur. As we discussed in 2.1.1 the commonly used data networks such as Internet, where infinite delay is possible (a message may never be sent in a busy network), are not suitable for the control application and will not be considered in the first place.

Delay Compensated LQG

With a series of papers including [35] [36] [37] [38] and [39], the authors derive a control law by expanding the conventional LQG control theory. The state of the original system is augmented with the pervious control inputs to account for the bounded random delays. An optimal LQG performance index of the augmented state is achieved with the proposed controller. The controller is a state feedback one, so they also derive a state estimator to deal with the process noise and output noise.

There are a series of assumption made on the system in [35] and [36]. The plant is continuous time and discrete time control input is applied via ZOH. The plant noise and output noise are both white and mutually independent. Negligible data loss rate is assumed. Other important assumptions are listed below:

1. The sensor and controller have same sampling period T with constant skew Δ_s.

2. The delay Δ_p in the processing of the control signal is constant. The skew between the instant of sensor and control signal generation is therefore $\Delta = \Delta_s + \Delta_p$, which is assumed to be smaller than T with probability 1.

3. The network-induced data latencies between the sensor and controller δ_k^{sc} and between the controller and actuator δ_k^{ca} are both bounded by T.

4. δ_k^{sc} and δ_k^{ca} are mutually independent and have the following probability

$$\begin{aligned} Pr[\zeta_k = 0] &= \alpha_k, \Leftrightarrow Pr[\delta_k^{sc} < \Delta_s] = \alpha_k \\ Pr[\zeta_k = 1] &= 1 - \alpha_k, \Leftrightarrow Pr[\delta_k^{sc} \geq \Delta_s] = 1 - \alpha_k \end{aligned}$$

where ζ_k is the measurement delay in units of sampling period T.

The relations of above values are illustrated in figure 1 of [35]. From the above assumptions we know that there are communication networks between sensor and controller as well as between controller and actuator, which is different from the configuration of the system studied in [42], although the necessity of connecting the controller and actuator directly with network, i.e. the lack of any local controller, is still a issue to discuss.

The plant dynamics without considering noise is

$$\tfrac{d\xi}{dt} = a(t)\xi(t) + b(t)u(t); \quad \xi(0) = \xi_0 \tag{2.19}$$

Discretizing it, we have

$$\xi_{k+1} = \Phi((k+1)T, kT)\xi_k + \sum_{i=0}^{1} b_k^i u_{k-1} \tag{2.20}$$

where

$$b_k^0 = \int_{kT+t^k}^{(k+1)T} \Phi((k+1)T, \tau)b(\tau)d\tau \tag{2.21}$$

$$b_k^1 = \int_{kT}^{(k+1)T} \Phi((k+1)T, \tau)b(\tau)d\tau - b_k^0 \tag{2.22}$$

and $\Phi[(k+1)T, kT]$ is the sate transition matrix from the k^{th} to the $(k+1)^{th}$ sampling instant.

In (2.21) $t^k$is the instant in $[kT, (k+1)T]$ when the new control input u_k arrives.

Augmenting the evolution equation with control inputs we have

$$x_{k+1} = A_{k+1,k}x_k + B_k u_k \tag{2.23}$$

where

$$A_{k+1,k} = \begin{bmatrix} \Phi((k+1)T, kT) & b_k^1 \\ 0 & 0 \end{bmatrix} \tag{2.24}$$

$$B_k = \begin{bmatrix} b_k^0 \\ I_m \end{bmatrix} \tag{2.25}$$

and

$$x_k = \begin{bmatrix} \xi_k \\ u_{k-1} \end{bmatrix} \tag{2.26}$$

Notice that since t^k in the integration limits of (2.21) and (2.22) is random, b_k^0, b_k^1 and thus $A_{k+1,k}$ and B_k are random processes.

The feedback controller is derived to minimize the performance index of the augmented states [35] [36]:

$$J_k(x_k, u_k) = \frac{1}{2}[x_k^T Q_k x_k + u_k^T R_k u_k] + E[J_{k+1}^*(x_{k+1}|x_k)] \tag{2.27}$$

When $k = N$, the initial condition is

$$J_k^*(x_N) = J(x_N, u_N^*) = \frac{1}{2} x_N^T P_N x_N \tag{2.28}$$

where $P_N = S$ is given.

In (2.27) and (2.28) respectively, $Q_k = \begin{bmatrix} \tilde{Q}_k & 0 \\ 0 & 0 \end{bmatrix}$ and $S = \begin{bmatrix} \tilde{S} & 0 \\ 0 & 0 \end{bmatrix}$ where $\tilde{Q}_k \in R^{n \times n}$ and $S \in R^{n \times n}$. (n is the order of the original system 2.19).

The state feedback optimal controller is then derived by using dynamic programming in common LQG problems. The resulting control law is

$$u_k^* = -F_k x_k = -F_k[\begin{array}{cc} \xi_k^T & u_{k-1}^T \end{array}]^T \tag{2.29}$$

The controller (2.29) requires full knowledge of the state of the original system (2.19). With the presence of process noise and measurement noise it is necessary to design a state estimator. It is derived in [38] and restated in [35]. We will not describe the state estimator in detail here. The state estimator is derived to minimize the mean square error. The overall solution of the problem is given by integrating the state feedback controller and the state estimator. The feedback controller (2.29) now is modified to be

$$u_k^* = -F_k[\begin{array}{cc} \hat{\xi}_k^T & u_{k-1}^T \end{array}]^T \tag{2.30}$$

Control (2.30) implies that the formulation of state estimation and state feedback can be separated, however in the state estimator the filter gain is dependent on the state of the system. Therefore the overall estimate/control system does not comply with the principle of certainty equivalence. This means the proposed state feedback controller integrated with state estimator is only sub-optimal as stated by the author in [35].

Simulation examples of air plane control are shown in [37] and [38]. The results are moderately satisfactory because of the sub-optimal nature of the approach.

The Ray group has studied the so-called integrated communication and control system (ICCS) for many years. Paper [35] is a summery of the work done in [36] [37] and [38], which also gives clearer presentation of the problem statement as well as the solution. Besides the theoretical and simulation result, the methodology used and assumptions made in the papers is typical in the stochastic approach for such systems as ICCS.

Linearization Method

In [43], the authors study a design methodology for distributed control systems in the presence of time delays. The system they study is distributed system formed by nodes which are physically separated and communicate with each other via communication networks.

The performance criteria is in general a measurement of the deviation of the output of the system from the given reference. And the performance of the system is derived to be bounded by the sum of the degradation of performance with the optimal performance. The degradation here is caused by the time delay on the communication network between nodes. The authors assume that the performance degradation is a function of these delays and Taylor's expansion is suitable for this function. Thus a linear relation of the time delay between the nodes and the performance degradation is obtained. Namely, the degradation can be expressed as the product of a performance differential function matrix determined by the design of the distributed system, and a time delay matrix with elements as the time delay between the i^{th} and j^{th} nodes. On basis of this result they propose a method to design the pattern to distribute the nodes, i.e. the best way to group the nodes such that the performance degradation is minimized. The simulation and experiment results are both given.

The approach of the paper assumes the suitability of Taylor expansion of the performance degradation as a function of time delay, which is not always true in the communication control environment. The main concern of the problem is to find the best way to distribute nodes which is not of great interest to our application.

2.1.3 Summary

From the literature review of the previous work in the field of networked control system we find that in spite of so much research in the area, there is still not any systematic method to address the typical problems. Most of the solutions are problem specific. The simulations as well as experimental results in many published papers are not satisfactory. Very little work is presented about how to design a controllers with the consideration of the network effect. We also lack universal network protocol design principles to meet certain classes of control system needs.

To us the literature to date on the control of networked systems looks like a creative but ad-hoc collection of ideas. The communication channels, estimation structures, or control distributions in the different papers are too varied and incomparable to offer a scientific method to the student or designer of networked control. There is no equivalent in this field of the information theoretic characterization of communication channels, the separation principles of estimation and control, or the optimality of specific estimation and control structures. The literature reveals that the control of networked systems is an inter-play of communication, distributed estimation, and distributed actuation. The literature also reveals the need to develop a science that provides a systematic design process. The purpose of our research is to tame the heterogeneity of communication channels by structured abstraction, identify estimation and control configurations for broad classes of networked control problems, and provide tractable computational methods that process experimentally collectible data to fill out the estimation and control structures for specific problems.

2.2 Communication Architecture Design

The objective of MOU388 is to enhance the safety of highways by integrating communication and control. The project has three goals:

1. Develop a ”cooperative friction estimation” algorithm that uses driving data communicated from many vehicles to estimate how slippery the road is.

2. Design a ”cooperative adaptive cruise control” system that uses communicated data to reduce the risk of collision between vehicles.

3. Build communication architectures that are robust enough for inter-vehicle use.

To achieve the first two objectives, 1 and 2, we need to transmit data using the communication architecture, 3, under development. Conversely, to develop the communication architecture we need 1 and 2—the applications—to provide realistic requirements and design constraints. All three objectives take realistic steps towards full highway automation but at the same time contribute to more near-term highway safety.

In the remainder of this report, we focus mostly on goal 3 above, the development of communication architectures that are robust enough for inter-vehicle use. We do so with an eye towards designing protocols that will be useful both for a fully developed, infrastructure-supported AHS, as well as for the more distributed car-to-car networks that may serve as a transition towards AHS.

An inter-vehicle communication network poses several problems that make it different from an ordinary computer network. Unlike an ordinary network, an inter-vehicle network is wireless. More importantly, it is highly dynamic since every node in the network (i.e. vehicle) is moving at high speed, changing the topology of the network with time, i.e., the absolute as well as relative position of the nodes are not fixed. Another difficulty comes from the anonymity, i.e. the vehicle has no way to find the communication ID of the vehicles in its neighborhood.

A good analogy that reveals some of the problems that an inter-vehicle communication networks presents is to imagine using a CB radio to tell a car near you that you will be changing lanes, or to warn the drivers behind you that the road ahead is slippery. Besides not knowing the correct radio channels, you would have to decide what to do if nobody is in range or if too many cars are speaking at once.

We explain these problems in detail in the remainder of this document: In subsection 2.2.1, we introduce two communication architectures-a Distributed Architecture and an Infrastructure Supported Architecture. Then in 2.2.2 we describe the design, analysis and test of the Medium Access Control protocol. 2.2.3 summarize the simulation of wireless mobilwe network of highway vehicles using Network Simulator (NS-2) software.

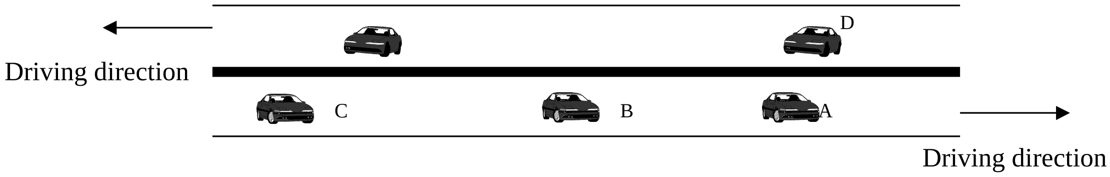

Figure 2.2: Distributed architecture on a divided highway

2.2.1 Distributed Architecture vs. Infrastructure Supported Architecture

We organize our research about two communication architectures, i.e., a Distributed Architecture (Figure 2.2) and an Infrastructure Supported Architecture (Figure 2.3). The Infrastructure Supported Architecture requires some roadside infrastructure while the Distributed Architecture realizes the networking service through the peer-to-peer interaction of vehicles alone. Our interest in the Distributed Architecture arises from its potential ease of deployment. If the expanding wireless market results in wireless modems appearing in many cars, the Distributed Architecture would enable large parts of America to enjoy support benefits without requiring the ubiquitous deployment of the required infrastructure across the nation.

Our interest in the Infrastructure Supported Architecture is due to the unpredictable evolution of the in-vehicle wireless market, the potentially greater operational benefits of partial centralization, and the possibility of public agencies mobilizing enough investment to drive at least a limited deployment of roadside infrastructure. In the future, the fusion of the two architectures may be a direction of research. We discuss the Distributed Architecture first on account of its relative simplicity and then discuss the additional features of the Infrastructure Supported Architecture.

In the Distributed Architecture, the transmitting and receiving of message is done completely by the vehicles on the road. If the receiver(s) is in the transmission range of the sender's communication equipment, there is no need for forwarding of messages. Otherwise, a vehicle will send the message to its neighbor, who, in turn, forwards the message to its neighbor until, finally, the message reaches the destination. For example, in Figure 2.2, if car A wants to send a message to car C which is out of the transmission range of car A's communication equipment, A first sends the message to B, and B forwards the message to C. In a divided highway, the oncoming

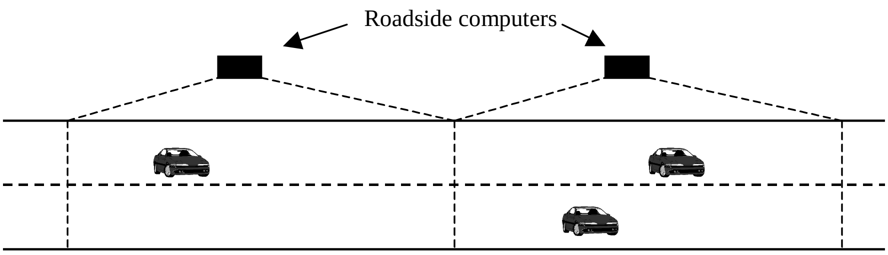

Figure 2.3: Infrastructure Supported Architecture

vehicles in the opposite direction can also help forward the messages. In Figure 2.2, car D can forward a message from A to C. From this illustration, we can see if the traffic is sparse, the store and forwarding of message only in the peer-to-peer manner may encounter some difficulties. Equivalently, when the percentage of the vehicles on the highway implemented with our communication architecture is low, the fragmentation of the network is a problem to overcome. For example, in Figure 2.2, if neither car B nor car D exists, or if they are not installed with wireless communication equipment, how could car A send the message to car C? This is one of the reasons to study the Infrastructure Supported Architecture.

In the Infrastructure Supported Architecture, we assume the existence of roadside computers on road segments. Each computer is assigned a segment of the road as in Figure 2.3. The sender vehicle transmits the message to the computer via modem. The computer can store and forward the message to the oncoming vehicles when the vehicle is in the computer's assigned segment. The computer can also forward the message via the LAN of the roadside computers until the receiving computer has the target vehicle in its segment. The advantage of infrastructure support is that the transmission of message is more reliable. The well-understood character of TCP/IP protocols used by fixed roadside computers saves effort for design and analysis. One weakness of this architecture besides the cost of building is the possible failure of the roadside computers. We will explore the backup methods to overcome this difficulty.

To implement these architectures, we assume each vehicle is equipped with:

1. GPS receivers,

2. Wireless communication system, and
3. On-board computer.

At this stage we focus mostly on the distributed architecture to understand the basic feature and difficulty we would encounter in building such communication architecture. Once we have had the a good understanding of the relatively simple case of distributed architecture we will go on to work on the design of the infrastructure-supported architecture. In subsection 2.2.2 we test and analyze a proposed MAC protocol for the network, in subsection 2.2.3 we describe some of our simulation results using network simulator software.

2.2.2 Ad hoc Medium Access Control Protocols

The function of the MAC protocol is to mitigate collisions and make up for the loss of messages when a collision does happen. For example, in Figure 2.2, if both car A and car D are sending a message to car C at the same time, there will be a collision of the data packets, and both messages will be lost. We will design the MAC protocol, which can reduce the probability of car A and car D sending message simultaneously. If however the messages collide and are both lost, the MAC protocol should determine the way for A and D to retransmit the messages such that the two messages are not likely to collide again.

We assume no carrier sensing in the system. Therefore when a vehicle wants to send a message it does not check whether the channel is idle, instead, the time to transmit is determined completely by the sending vehicle itself. Carrier sensing is applied in many network protocols such as Ethernet(see in subsection 2.1.1). Here by not doing carrier sensing we are trading the lower probability of message collision for the simplicity of the network component on each vehicle. No carrier sensing is the reason for us to design and implement a smart MAC protocol.

One possible solution is to have time slotted and randomly retransmit the message for multiple times. We illustrate this idea in Figure 2.4. We assume that the “age” of a new message is 50ms, i.e. the message has to be transmitted within 50ms after its generation. Afterward it will be regarded as out of date and be discarded. The proposed MAC would evenly slot the 50ms after the generation of the message into N slots. The transmission of a message always takes place at the beginning of a slot and the transmission of the message takes one slot of time. The proposed protocol intentionally transmits this message k times in k slots (k≤N), which are uniformly randomly chosen from the N slots. As shown in Figure 2.4 (a), if all the messages are sent at the instant they are generated and in the first slot only, quite likely there is a collision. However if the messages are sent as suggested by the new protocol, as in Figure 2.4 (b), the probability of at least one transmission of the message to get through is much higher. Intuitively, as we increase the number of transmitting of a packet, the chance of it to get through in a busy channel is also increased. However the excessive retransmission also adds overhead to the channel and makes the channel busier. At the extreme, if we transmit the packet in all of the 50 time slots from 0 to 50 ms, the performance is the same as transmitting in the first slot only. Therefore we have a tradeoff between the number of transmission and the traffic overhead, and the optimal relation between k and N should be found to achieve the least loss rate as well as not to add too much traffic load to the network.

We simulate the MAC protocol to solve the problem. Some of the results are sown in Figure 2.5, Figure 2.6 and Figure 2.7. In these simulations, we assume the length of time slots is 1ms and each transmission takes one slot of time, therefore N is constantly 50 now. We simulate the most severe situation when multiple vehicles have message to transmit at exactly the same time, i.e. the 50ms "lives" of the messages from different vehicles overlap each other completely. This situation is worse than when the 50ms periods only partially overlap, since now the probability of collision is higher.

In this case if we use the protocol illustrated in 2.4(a), i.e. the message is sent only once at the instant of the generation, all of the messages will be collided and the probability of success will be 0. However when we apply the proposed MAC protocol, the probability of success is greatly enhanced. Figure 2.5 is the probability of success for different number of transmission when 5 vehicles are transmitting simultaneously, and Figure 2.6 shows the detail of the same relation. The x-axis is the number of transmissions in the 50ms period for each vehicle or the number k mentioned above, and the y-axis is the probability of success. We can see that with the proposed MAC protocol, the message that could not be sent in the old protocol can now be transmitted successfully with the probability of up to 97%. The simulation also suggests the optimal number of transmission for an individual vehicle, which could be the guideline for the implementation of the protocol on vehicles. For 5 vehicles, we can see that transmitting 3 and/or 4 times achieve the highest probability of success. In the figures we can also see that excessive time of transmission adds overhead to the network traffic, thus makes more collision. After the maximizing value 4, the increasing number of transmissions decreases the probability of success quickly. At 15 times, the probability of success drops to only 10%. The following is an examples of the time slots each vehicle chooses to transmit the messages. There are five vehicles and each vehicle transmits the message four times. The slots are uniformly randomly chosen within from 1 to 50. We can see that for this example all the vehicles are successful in transmitting their messages.

$vehicle(1)$	11	14	27	34
$vehicle(2)$	2	6	12	41
$vehicle(3)$	3	11	47	49
$vehicle(4)$	17	22	25	45
$vehicle(5)$	1	20	23	45

Figure 2.7 shows the same relation for the case when 10 vehicles are transmitting simultaneously. It is not quite likely in the real highway network that so many vehicles are talking at the same time, however it is studied to see the performance of the MAC protocol under critical circumstance. Figure 2.7 indicates the same trend as Figure 2.5 and Figure 2.6 do. The proposed protocol can almost surely transmit the message which will be collided without implementing the MAC protocol. The difference from the 5-vehicle case is that the probability of success is lower for the same number of transmission because that more senders are competing. The highest probability of success decreases to 90%, though still quite good. And the probability of success goes below 10% when the message is transmitted for 10 times, rather than the 15 times in Figure 2.5.

The shortcoming of this protocol is the increases in communication traffic caused by the intentional retransmissions. Our further analysis aims to study the suitability of this protocol to our application and its impact to the overall communication network. The general characters for protocols should be analyzed such as stability, maximum throughput and delay features. We will also work out the way for receiver to deal with the duplicate messages from one sender.

Another possible solution is ALOHA protocol. In this protocol, time is also slotted. The packets to be transmitted are assumed to arrive according to independent Poisson processes. Only 0 or 1 packet can go

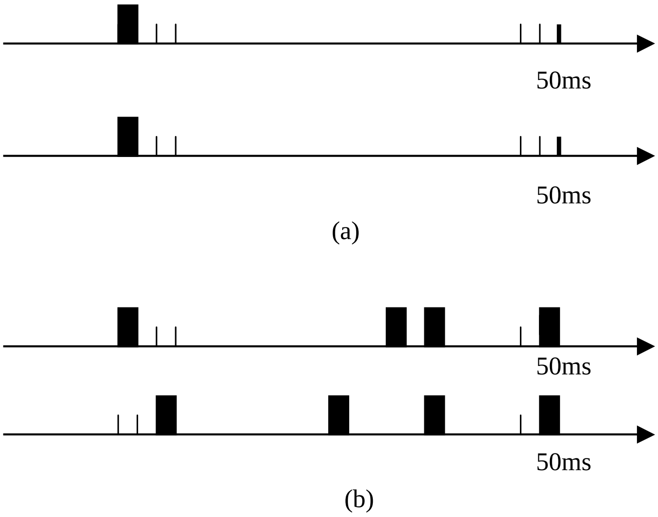

Figure 2.4: One proposed Medium Access Control Protocol: (a)Transmit message on the first slot only; (b)Transmit message multiple times on randomly selected slots

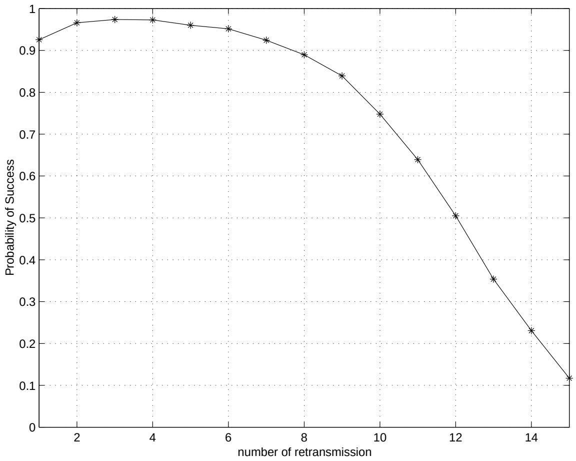

Figure 2.5: Probability of success vs. times of random transmission: 5 vehicles

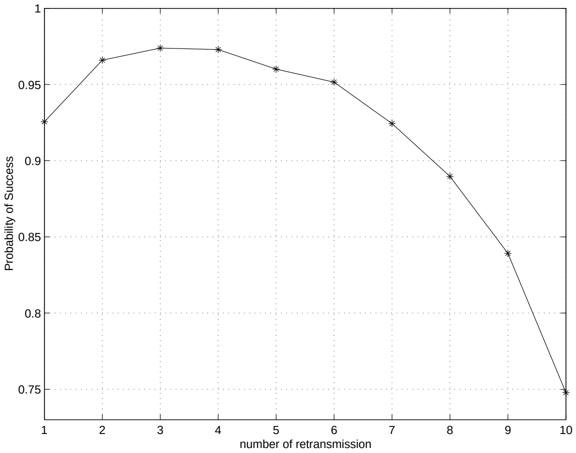

Figure 2.6: Probability of success vs. times of random transmission: 5 vehicles in detail

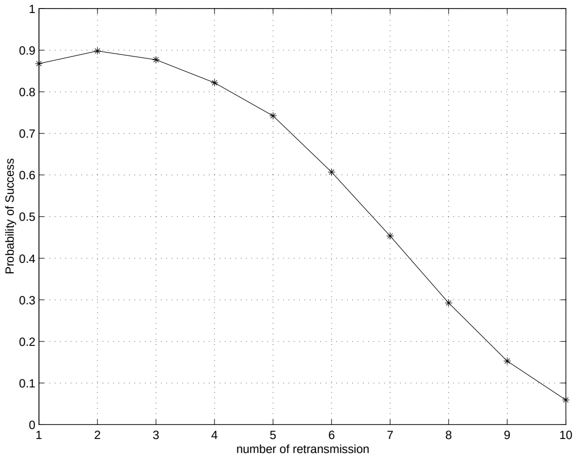

Figure 2.7: Probability of success vs. times of random transmission: 10 vehicles

through the network in each time slot. At the end of each time slot, each node obtains feedback from the receiver informing the receipt status of the message: no packet is transmitted or received (idle), one packet is transmitted and received, or multiple packets are transmitted and no one is received (collision). The nodes then decide whether to transmit or retransmit in a random manner. This protocol is a standard one and its analysis can be found in many common data network textbooks. If it is proved to be suitable for our application we can save a lot of analysis efforts. However the method requires the nodes to have knowledge of the traffic of the whole network, and the receiving nodes have to send acknowledgements to the sending node(s). These characteristics may not be good for AHS environment. On the other hand, because of the similarity of the communication mechanism to ours (multi-access channel, packet collision, randomly retransmission, etc), we should at least borrow some analysis ideas from the previous study of ALOHA.

We also notice that the message from different nodes may not have the same priority. For example, a message telling you that the road segment you will be on in two minutes is icy is not so urgent as a message telling you that a vehicle in the right lane is cutting in front of you. In this sense the latter message should have higher priority, and when the two messages reaches the destination at the same time, the latter should be preserved. We will look further into the priority requirement for the messages.

2.2.3 Vehicle-to-vehicle network simulation with NS-2

We simulate the mobile wireless network of vehicles on highway using a widely used network simulation package NS-2. The implementation and analysis of the simulation is described in this subsection.

The research on the integration of communication and control has attracted much attention in academia as well as industry. However, because of the rarity of researchers with expertise in both fields and the lack of communication between the two research communities, there are not a lot of directly useful results or solidly built theories so far.

In [7], the authors simulate and statistically analyze a vehicle communication network protocol in which a vehicle broadcasts the warning of an accident to all the following vehicles. The simulation tool they used was SHIFT, a tool designed specially to simulate the behavior of hybrid systems. The trace of packets is not supported in SHIFT, and the network components have to be written by the user. Therefore the behavior of the network cannot be fully studied, and more complicated protocols are very hard to be implemented, if not impossible, due to the limitation of the simulation tool.

The work done in these simulation is the first step toward building and analyzing of mobile wireless network of the vehicles on the highway. As far as our knowledge, there are no comparable published simulation results for the behavior of the network for highway traffic model. Nor has any work been done using the specially designed simulator of the network such as NS-2. This is the motivation of us to work in this direction. The simulation system built should enable us to study the performance of the wireless network of vehicles for different protocols on all the network layers. This simulation tool should greatly enhance our ability to analyze and test such design of network.

In this subsection we discuss the implementation of the simulation system in sub-subsection 2.2.3. 2.2.3 is the results and discussion and 2.2.3 is the summery.

Implementation

In this sub-subsection we describe the implementation of the simulation system. We present it in the network components, the platoon passing by scenario, and the vehicle cutting in scenario respectively. The network components including link layer, interface queue, network interface, radio propagation model, and antenna model are the same for both of the scenarios, therefore we describe them in one subsection. Then we describe the two scenarios, the first one is the platoon-passing-by and the second is vehicles-cutting-in.

The network components The network components are the same for both of the scenario. We describe them one by one below.

*Link Layer:*The link layer used by mobile node is same as the one used by the wired node. The only difference is that the link layer for mobile node has an ARP module connected to it which resolves all IP to hardware (Mac) address conversions. Normally all outgoing (into the channel) packets are handed down to the link layer by the Routing Agent. The link layer hands down packets to the interface queue. For all incoming packets (out of the channel), the mac layer hands up packets to the link layer which is then handed off at the node_entry_ point.

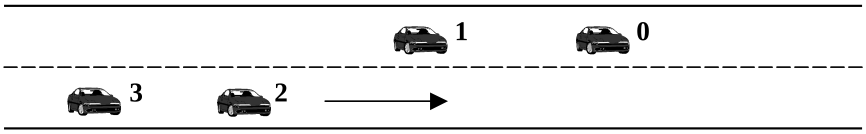

Figure 2.8: Platoon passing by scenario

Interface Queue: The PriQueue is implemented as a priority queue which gives priority to routing protocol packets, inserting them at the head of the queue. It supports running a filter over all packets in the queue and removes those with a specified destination address. The maximum number of the packets in the interface queue is set to 50.

Network Interface: The Network Interface layer serves as a hardware interface which is used by mobile node to access the channel. This interface subject to collisions and the radio propagation model receives packets transmitted by other node interfaces to the channel. The interface stamps each transmitted packet with the meta-data related to the transmitting interface like the transmission power, wavelength etc. This meta-data in packet header is used by the propagation model in receiving network interface to determine if the packet has minimum power to be received and/or captured and/or detected (carrier sense) by the receiving node (however the carrier sense is disabled in the MAC in our simulation). The model approximates the DSSS radio interface (Lucent WaveLan direct-sequence spread-spectrum).

Radio Propagation Model: It uses Friss-space attenuation at near distances and an approximation to Two Ray Ground at far distances. The approximation assumes specular reflection off a flat ground plane.

Antenna Model: An omni-directional antenna having unity gain is used by mobilenodes.

Platoon-passing-by The first scenario we simulate is called "platoon-passing-by". It is shown in Figure 2.8, where two two-vehicle platoons are driving in neighboring lanes. At the beginning of the simulation, the platoon formed by vehicles 2 and 3 is behind the one formed by vehicles 0 and 1, and the platoons are out of the communication range of each other. After some time, vehicles 2 and 3 catch up with the other two vehicles. Then the four vehicles drive side by side with almost the same velocity for some time., In the final stage of the simulation, vehicles 2 and 3 speed up again and pass the other two vehicles, until finally the two platoons are out of communication range of each other again. When doing this simulation we have in mind the braking scenario in CACC application 3, therefore we test a protocol for the transmission of braking signal and observe the effect of distance and message collision on the performance. The details of the implementation are described below.

From the point of view of a user, a special part of the mobile network extension of NS-2 is that a node movement file and a network traffic pattern file have to be written. The former file is to tell the node how to move during the simulation. Variables defined in this file include, for each node, initial position, destination, (average) velocity, and so on. The latter file determines what the network traffic pattern is for the agents attached to each node. Variables defined include the connection of agents, the type of agents, transmission rate, maximum packets number, packet size, etc. Depending on the type of the agent, not all these variables are defined. In this subsection we describe the node movement file and network traffic pattern file of the platoon passing by scenario, and the files for vehicle cutting in scenario will be introduced in the next subsection.

In the simulation, we implement a protocol like the following. In the protocol:

1. Each application of the brake by the leading vehicles(0 and 2 in Figure 2.8) generates a message.

2. 200 bytes per point to point exchange

3. Each vehicle tries to broadcasts the message as soon as the message is generated. Transmitting both with carrier sensing and without carrier sensing are tested. (see 2.1.1 and 2.2.2 for the definition and explanation of carrier sensing)

4. The message is sent omni-directionally to all vehicles in the range.

5. All vehicles in the communication range of other vehicles' can be reached in one hop. Thus the difficulty in routing is avoided.

When working on this project we still did not have in hand a good set of highway traffic data including the position, velocity, and braking distribution of the vehicles for a certain group of vehicles. Thus we have to manually write a movement file for the vehicles. We make the following assumption for the vehicle movement on the highway.

1. The simulation lasts 400 seconds
2. in average 5 brakes/min
3. 1000 feet of the range of the wireless communication system
4. 2 lanes
5. inter-vehicle distance vehicles in the same platoon = 4m; initial distance between the two platoons = 550 m.
6. velocity of vehicles 0 and 1 is 25 m/s (≈56 mph) for all the time of simulation; velocity of vehicles 2 and 3 is 30 m/s (≈67 mph) at the beginning, 25 m/s when the two platoons are side by side which happens at 105 second, and 30 m/s again at 300 second.

The above assumptions are made on the basis of highway traffic with not too high velocity (6), small inter-vehicles space (5) and frequent brakings (2). There are 4 vehicles in the scenario, hence the number of nodes in the communication network is 4. The assumed range of the communication hardware is according to the built-in physical layer codes in NS-2, which is based on the Lucent WaveLan DSSS radio interface. The reason of making the above assumption is to simulate a somehow severe network traffic situation without considering the routing difficulties.

Since each of the messages corresponds to a braking, the network traffic pattern is equivalent to the distribution pattern of the braking of vehicles. In the simulation we assume the followings on the brakings.

1. Suppose a vehicle brakes at an average rate of λ brakes per second, then the time between two successive brakes by the same vehicle is exponentially distributed

   $$Prob(\delta > t) = e^{-\lambda t}$$

   where δ is the interval of two successive brakings of the same vehicle and t stands for the time.
2. The braking of each vehicle is identically independently distributed.

The first assumption is typical for the accumulation of large number of random occurrence. The second assumption, however, is oversimplified. In real highway traffic, one braking of a vehicle is quite likely to cause several following vehicles to brake also. Also a busy highway may cause all the vehicles in the segment to brake more frequently than in the other. Thus the distributions of the braking for different vehicles are actually highly correlated. This over-simplified assumption makes the braking evenly distributed for all vehicles. A braking of a vehicle does not cause a consequent brakings as it should, thus the number of the braking in the simulation is less than in the real traffic. And the network traffic is less severe than in the real case.

Vehicles-cutting-in The second scenario is named “vehicles-cutting-in” as shown in Figure 2.9. In this scenario, vehicle 1 is following vehicle 0 at the beginning. After travelling for some time, they catch up with vehicle 2, and the latter cut in between 0 and 1. Now vehicle 2 is directly ahead of 1 and 0 is in turn directly ahead of 2. Then after the three vehicles travelling for some time, vehicle 3 cuts in between 1 and 2 in the same manner. And finally vehicle 4 cuts in between vehicles 0 and 3. The final order of vehicles in the platoon from the leader to the tail is 0, 4, 3, 2, and 1. As in the platoon-passing-by scenario, we have in mind the CACC application in designing the vehicles-cutting-in scenario. However this time the CACC scenario we consider is the cutting-in scenario in chapter 3. Although the information needs to be communicated between the vehicles in such kind of application is still a topic to study, we believe that for the CACC application, the position, velocity, and maybe acceleration of the vehicles participating in the cutting in maneuver are essential to enhance the safety, efficiency and comfortability. Thus the network traffic in this scenario is different from that of the platoon-passing-by scenario. The message here must be sent at a high and constant rate, in contrast to the low rate and bursty braking signals. Therefore we install Constant Bit Rate(CBR) UDP agent in the sender port of the vehicles in the simulation.

The protocol is like the following:

1. The packet size is 200 byte. The time interval between the successive packets of the same vehicles is 0.2 second. Therefore the sending bandwidth is 8000 bps.

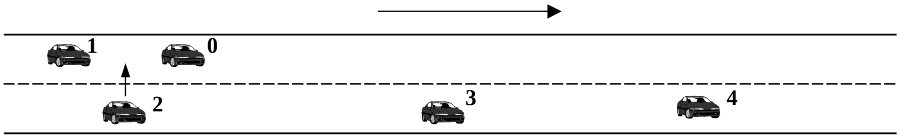

Figure 2.9: Vehicles cutting in scenario

2. Each vehicle tries to broadcast the message as soon as the message is generated. Transmitting with and without carrier sensing are both tested.

3. The message is sent omni-directionally to all vehicles in the range.

4. All vehicles in the range of other vehicles can be reached in one hop. Thus the difficulty in routing is avoided.

5. A cutting in vehicles start transmitting when the cutting in is started.

Because of the variety of the application in automated highway system, the point-to-point communication, which highly depends on the addressing and routing protocols, should be studied in future work.

The movement file and hardware assumptions are the following:

1. The simulation lasts 400 seconds

2. 1000 feet of the range of the wireless communication system

3. 2 lanes with width of 5m.

4. Velocity of vehicles 0 is always 25m/s.

5. Velocity of vehicle 1 is 25m/s when no other vehicles are cutting in; it slows down a little in the procedure of other vehicle's cutting in to make more space.

6. The longitude velocity of vehicle 2 ,3 and 4 is 20 m/s before cutting-in and is 25m/s during and after cutting in.

7. The lateral velocity for all cutting-in vehicles is 1m/s. The acceleration and deceleration period are neglected.

8. The cutting-in for vehicles 2, 3, and 4 happens at 25 second, 100 second, and 150 second respectively.

9. Inter-vehicle distance is 4m for vehicles in the same lane.

Results and discussion

The results of the simulations are shown in this section. We still present them in two subsections according to the two scenarios.

Results and discussion of platoon-passing-by In Figure 2.10, Figure 2.11 and Figure 2.12, we plot the relation between the probability of success of transmission with the average time interval of the transmissions of the same vehicle. In all of the figures, the x-axis is the average time interval between successive transmissions, and the y-axis is the probability of success. Figure 2.10 shows the relation when carrier sensing is disabled, while Figure 2.11 is the relation when carrier sensing is done. For these two figures, the dotted line stands for the measured values from the simulations and the solid line is the fitted curve of the measurements. In Figure 2.12 the above two cases are compared. Here the line dotted with squares is the measured value for the case of no carrier sensing and the solid line is its fitted curve; on the other hand, the line dotted with stars is the measured value for the case of with carrier sensing and the dashed line stands for its fitted curve.

we can see that as the braking interval increases, the network traffic is less busy, thus the probability of success is increased. We can predict that as the braking interval goes to infinity, the probability of success will converge to 1. When the braking interval is large, the protocol with and without carrier sensing tend to be the same. This is reasonable since when the result of carrier sensing is always idle, the transmission of the message is equivalent to the case when no carrier sensing is done. When the braking interval decreases, the carrier sensing can greatly mitigate the collision, therefore the probability of success does not change much even when the braking interval is quite small. At the same time, when no carrier sensing is done to prevent collision from happening, smaller message interval, or equivalently busier network traffic reduce the probability of success dramatically. Therefore we can see that the carrier sensing can protect the performance of the protocol from being affected too much by the load of the network. However the carrier

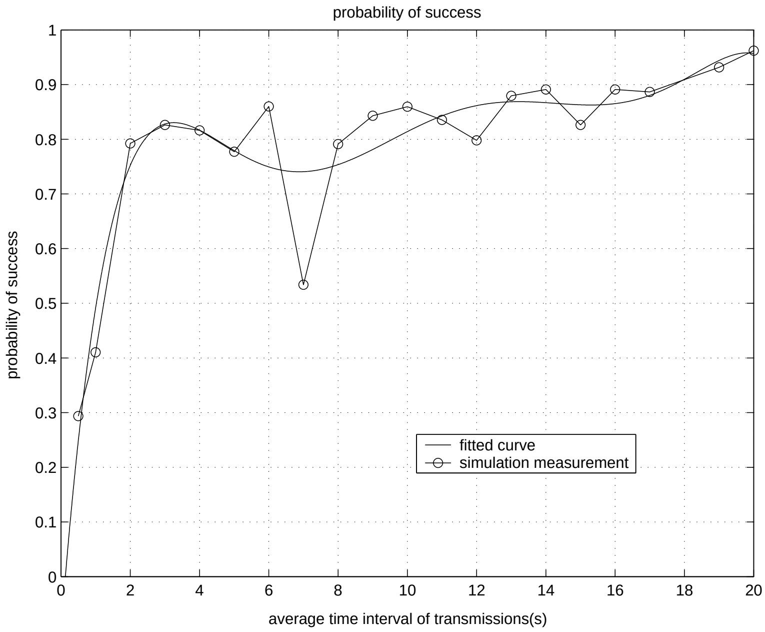

Figure 2.10: Probability of success vs. average transmission interval:Without carrier sensing

sensing makes the implementation of the communication system complicated and may not work well for some of the wireless environment. And the values of the message intervals which ruin the performance of the protocol do not commonly appear in the usual highway vehicle applications. Hence whether to include carrier sensing in the protocol is still an open question. We still do not have explanation for the overshoot at 3 second in Figure 2.10.

Figure 2.13 to Figure 2.18 show the bandwidth of the nodes (vehicles) in the network when the average message interval is 15 seconds, for which value there is not great difference between the protocols with and without carrier sensing. In all of them, the x-axis is the time in seconds and the y-axis is the bandwidth in Mega-bit per second (Mbps). Each impulse stands for a message packet being sent or received at the corresponding time. Since only the leader vehicles (Figure 2.8) send messages, sending bandwidth of only vehicles 0 and 2 are plotted in Figure 2.13 and Figure 2.14. The figures confirms that the time interval between the messages is exponentially distributed. Figure 2.15 and Figure 2.17 are the receiving bandwidth of vehicle 0 and 2 respectively. Since 0 and 2 are the only vehicles transmitting

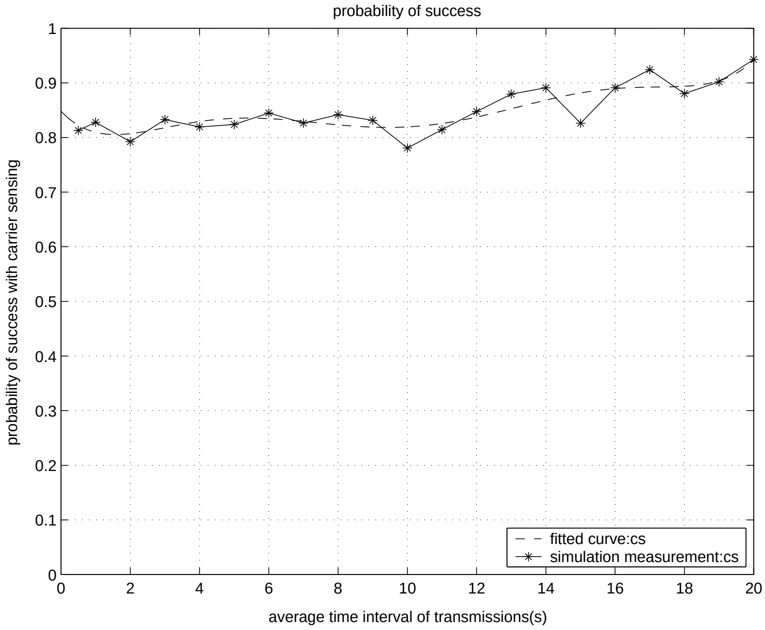

Figure 2.11: Probability of success vs. average transmission interval:With carrier sensing

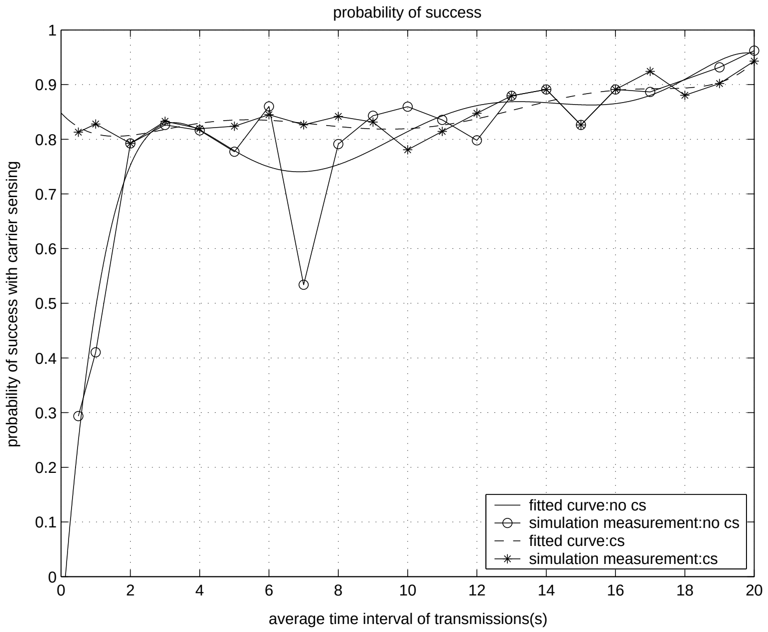

Figure 2.12: Probability of success vs. average transmission interval

in the simulation, all the packets received by vehicle 0 are sent from vehicle 2 and all the packets received by vehicle 2 are from vehicle 0. Comparing Figure 2.15 with Figure 2.14, and Figure 2.17 with Figure 2.13, we can see that from time 0 to about 70 second, they cannot receive packets sent by each other because they are out of communication range of each other. The same happens from about 350 second to the end of simulation. In the time between vehicle 0 receives most of the packets vehicle 2 sent while vehicle 2 receives most of the packet vehicle 0 sent. Figure 2.16 and Figure 2.18 show the receiving bandwidth of vehicles 1 and 3. They are always within the communication range of their leaders, namely vehicles 0 and 2 respectively. They can receive packets from the other sending vehicle only when they drive into the senders' range, which is what happens between 70 second and 350 second in this example. In this period of time the packets received by vehicles 1 and 3 are almost the same and are approximately the combination of the packets sent by both vehicles 0 and 2. Notice that Figure 2.12 shows probability of success of transmission of about 90% for braking interval of 15 seconds, therefore some of the packets sent by the two senders collide and are discarded.

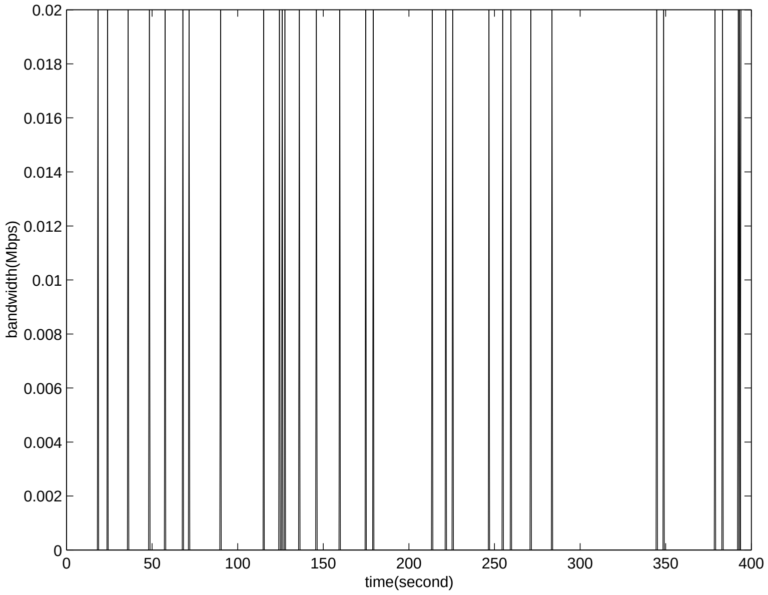

Figure 2.13: Platoons Passing By: Sending bandwidth of vehicle 0

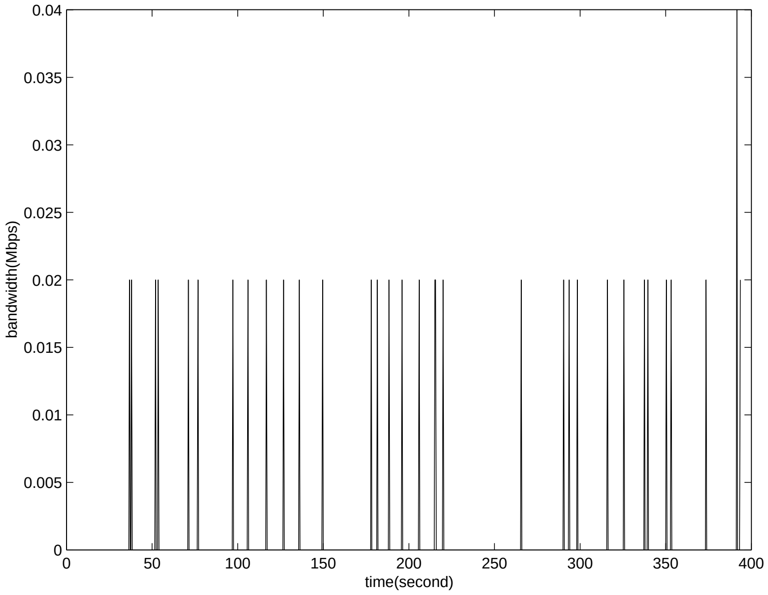

Figure 2.14: Platoons Passing By: Sending bandwidth of vehicle 2

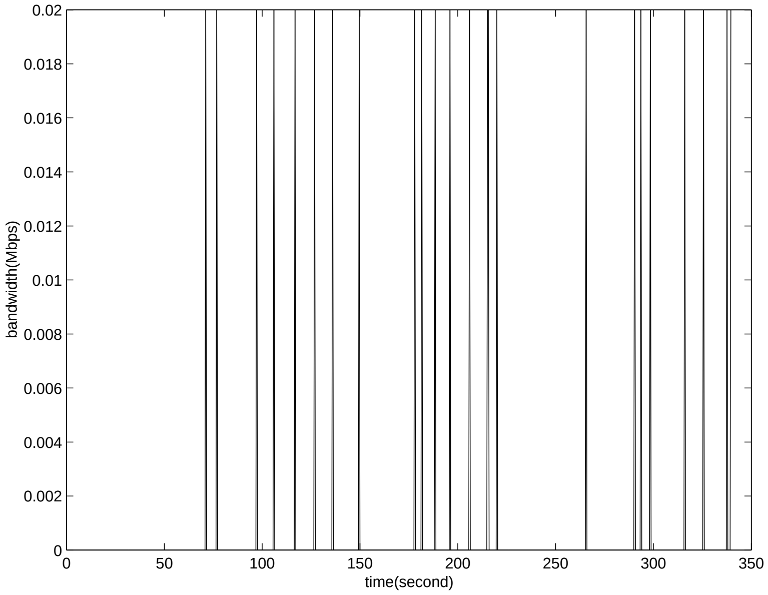

Figure 2.15: Platoons Passing By: Receiving bandwidth of vehicle 0

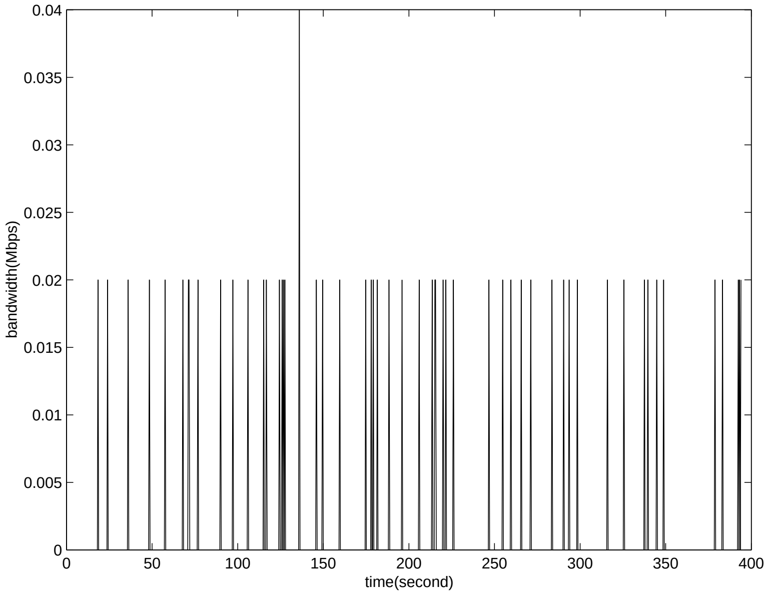

Figure 2.16: Platoons Passing By: Receiving bandwidth of vehicle 1

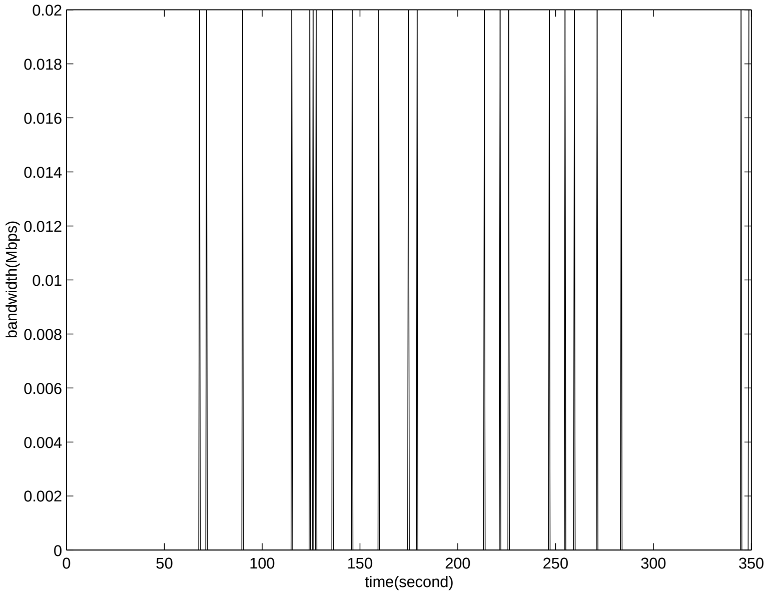

Figure 2.17: Platoons Passing By: Receiving bandwidth of vehicle 2

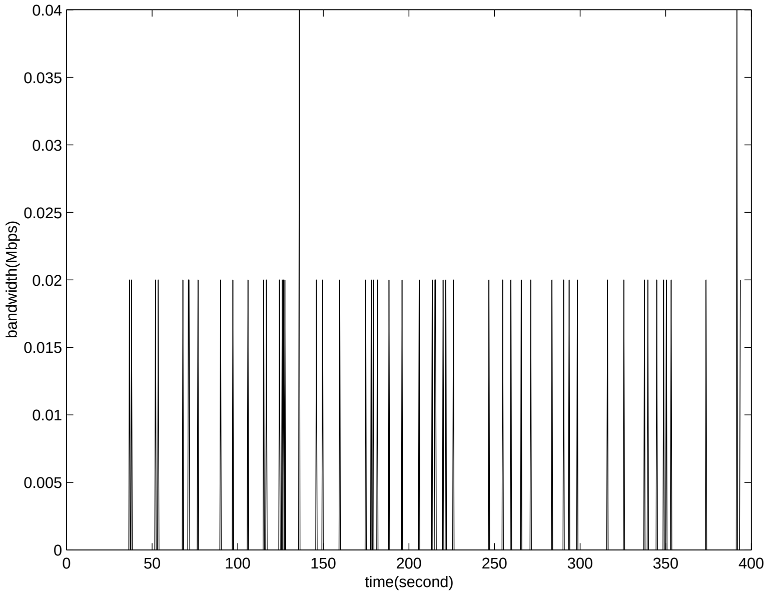

Figure 2.18: Platoons Passing By: Receiving bandwidth of vehicle 3

Results and discussion of vehicles-cutting-in Figure 2.19, Figure 2.20, Figure 2.21 and Figure 2.22 are the sending bandwidth of vehicles 0, 2, 3, and 4(see Figure 2.9). Figure 2.23 and Figure 2.24 are the receiving bandwidth of vehicle 1 when the packets are sent with no carrier sensing and with carrier sensing respectively. For all of the figures, the x-axis is time in second and the y-axis is the bandwidth in Mpbs. Since vehicle 1 is the only vehicle which can receive all the packets sent from other vehicles and which never sends packets, we plot its receiving bandwidth to observe the effect of multiple senders on it. From the sending bandwidth we can see that except vehicle 0 who sends packets all the time, the cutting-in vehicles 2,3, and 4 start transmitting only when their cutting-in maneuver starts, which is what we expect. The sending bandwidth is constantly 0.008Mbps for all the senders. In Figure 2.24 we can see that with the help of carrier sensing the packets can arrive without collision and the receiving bandwidth of vehicle 1 is the sum of all sending bandwidth at the time, therefore the receiving bandwidth changes to 0.016 Mbps at 25 second, 0.024 at 100 second, and 0.032 at 150 second. In Figure 2.23, since there is no carrier sensing, most of the packets are collided and only those from one sender of them can get through, depending on the order of sending in each cycle. Thus the change in sending bandwidth does not affect the receiving bandwidth.

Summery and Future Work

As the first step toward a somehow ambitious plan, the NS-2 simulation produces the results which serve as the guideline for the future study. This project proved NS-2 a powerful tool for the simulation of the kind of network we aim to build for vehicles. The finished work provides the Otcl codes and C++ codes as the basis of future implementation, such as the MAC protocol we discussed in 2.2.2. The results are trustful and give us insight into the nature of the mobile wireless network.

In the future, there is a long way to go to achieve the ultimate goal of the research in this direction. First we have to apply the data of real highway traffic which should include the position, velocity, braking distribution, lane changing, etc. We may use a highway traffic simulator such as Paramics to write the file of such data into the movement scenario file and network traffic pattern file. Or even better we may use the data measured from the real highway traffic. We are already working on this now. If the highway traffic simulator is used as the origin of the data, the consistence of the software

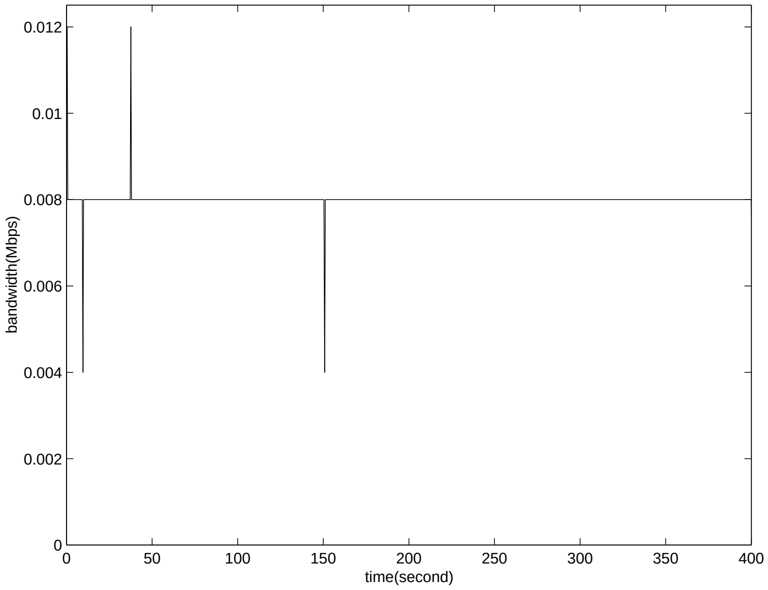

Figure 2.19: Vehicles Cutting in: Sending bandwidth of vehicle 0

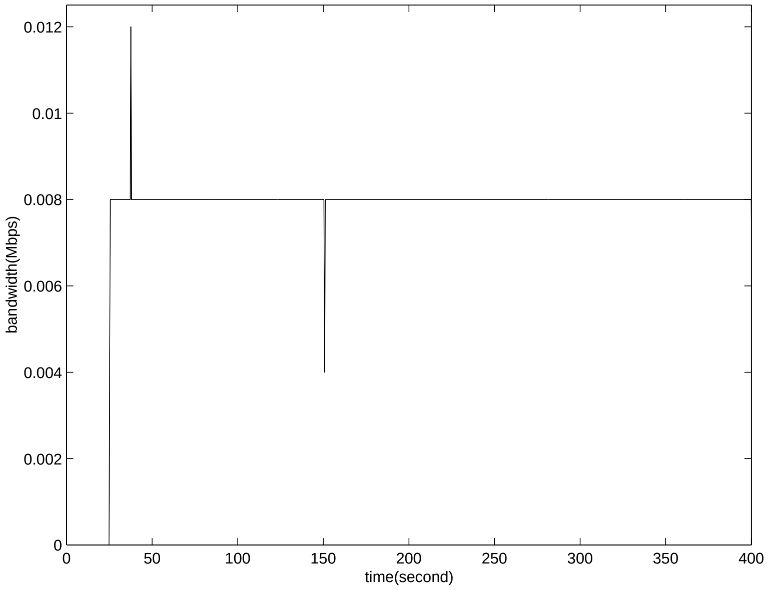

Figure 2.20: Vehicles Cutting in: Sending bandwidth of vehicle 2

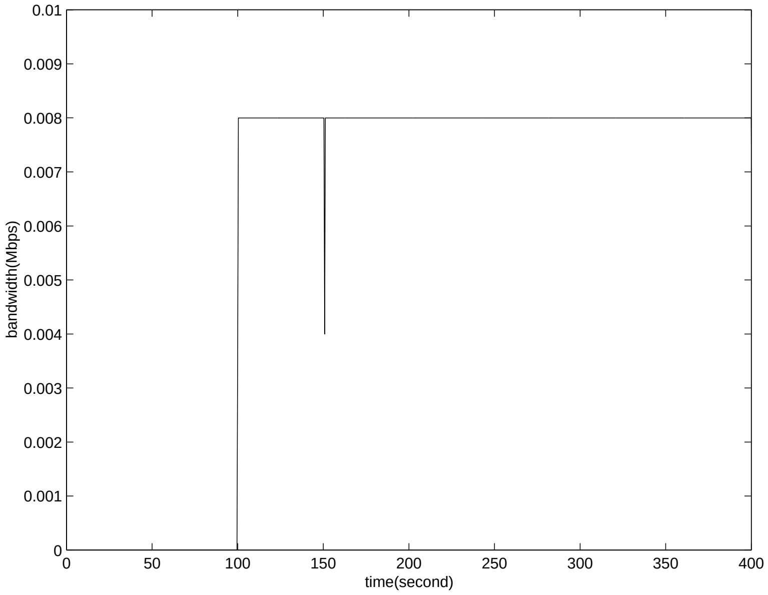

Figure 2.21: Vehicles Cutting in: Sending bandwidth of vehicle 3

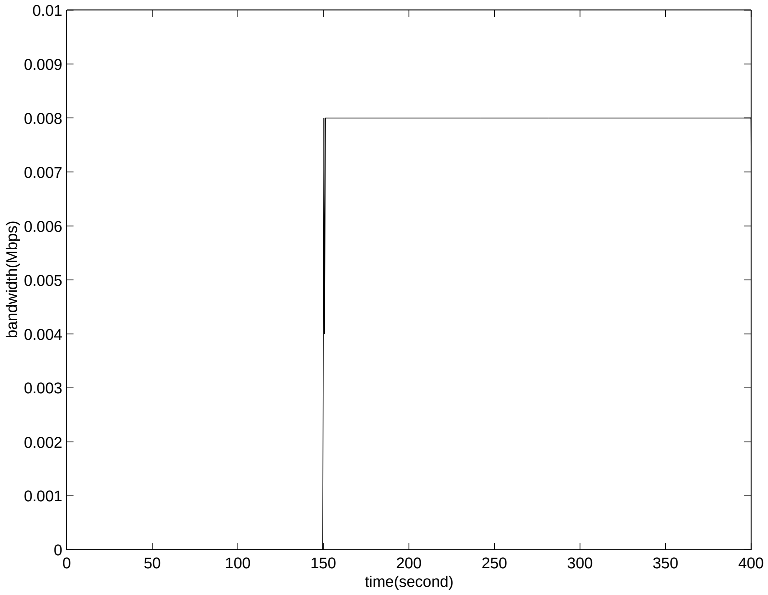

Figure 2.22: Vehicles Cutting in: Sending bandwidth of vehicle 4

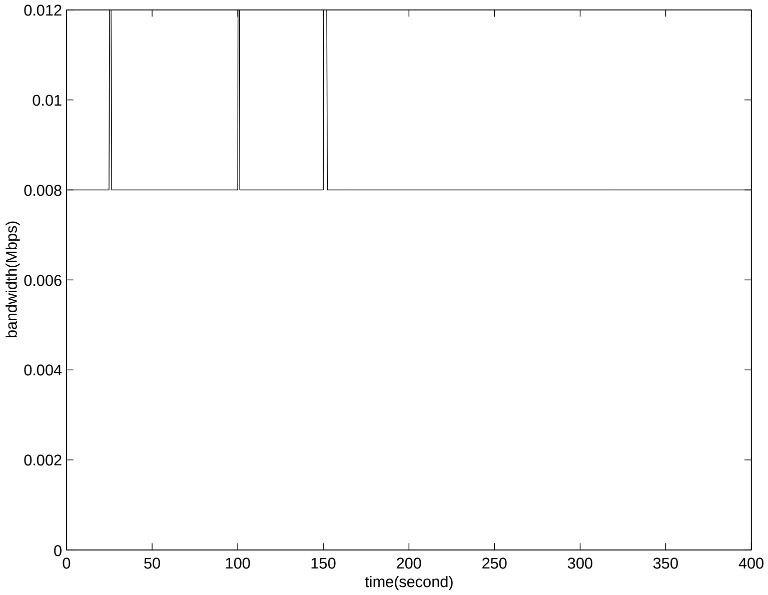

Figure 2.23: Vehicles Cutting in: Receiving bandwidth of vehicle 1 (without carrier sensing)

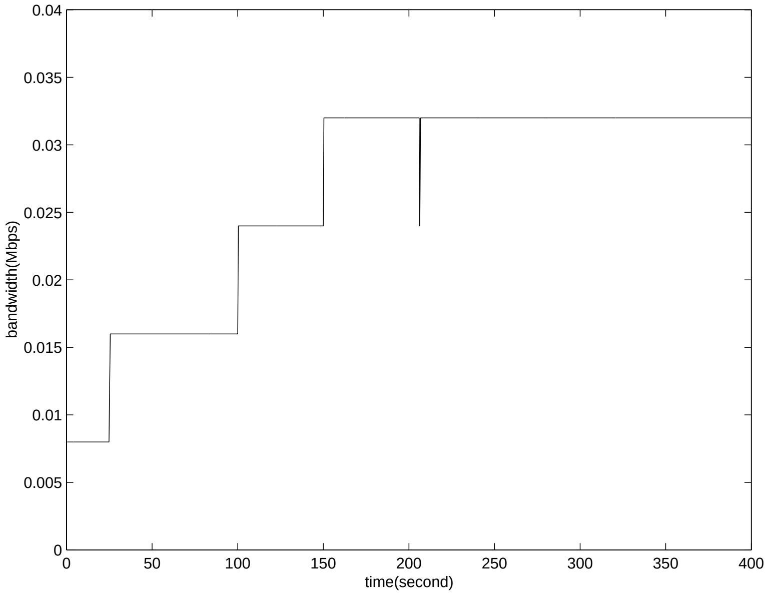

Figure 2.24: Vehicles Cutting in: Receiving bandwidth of vehicle 1 (with carrier sensing)

could be a problem to study.

Another modification is on the radio propagation model. The propagation model in this project uses Friss-space attenuation at near distances and an approximation to Two Ray Ground at far distances. We should modify such a model to account for the Doppler effect caused by the highway speed of the transmitters and receivers.

Lastly, once we have got satisfactory result for the ad hoc network we will go on studying the infrastructure-supported architecture, which is readily supported by the mobile IP extension of NS-2 and could be built easily on basis of the simulation system described here.

Chapter 3

Cooperative Adaptive Cruise Control

Current production cruise control systems maintain a set speed of a vehicle with no information about other vehicles. On the other hand Adaptive Cruise Control (ACC) Systems use sensor information to monitor vehicles ahead, and change cruise speed accordingly. Researchers in PATH are developing a more robust and reliable ACC by adjusting its performance to the driving conditions. This entails an appropriate combination of sensory data, physically based dynamic models, and logical situation models in an intelligent data fusion framework.

The implementation of the communication architecture studied in MOU388 should enhance the performance of the ACC controller significantly. With the aid of inter-vehicle communication, the controller relies not only on the measurements of the on-board sensors and radars of the controlled vehicle, but also the messages transmitted from other relevant vehicles. In this sense the ACC is not achieved by the ACC vehicle itself but cooperatively by all the vehicles involved in the specific maneuver. We call this new ACC method Cooperative Adaptive Cruise Control (CACC) to distinguish it from the conventional ACC.

The relation of ACC,CACC and paltooning/AHS is like the following.

1. *Cruise Control(CC): Vehicle is controlled to maintain a preset velocity without considering other vehicles*

2. Adaptive Cruise Control (ACC): Uses onboard sensor information to monitor vehicles ahead, and changes cruise velocity accordingly. Cruise control is done by the vehicle itself.

3. *Cooperative Adaptive Cruise Control (CACC) Besides the onboard sensor information, uses the communicated messages from vehicles ahead to realize ACC. Cruise control is done cooperatively by all the vehicles involved*

4. Platooning/AHS: More vehicles involved, more cooperation for more complex maneuvers, more order

From the top of the list to the bottom, the method is getting more and more complicated. In this structure, CACC is an important step toward the fully automatic highway system, at the same time it could be a marketable product itself in the near term.

A simulation model for BMW experimental vehicle was constructed using the SIMULINK simulation package by researchers in the Vehicle Dynamics and Control Laboratory of U. C. Berkeley. They also developed an ACC control law with nonlinear sliding surface control technique. The control law was integrated with the vehicle model and the over-all model was simulated successfully [22]. On basis of their model and controller we study the effect of vehicle-to-vehicle communication on ACC.

In this chapter we describe the cutting-in scenario and its results in section 3.1, the braking scenario and its results in section 3.2, and summarize the the CACC research in section 3.3.

3.1 Cutting-in scenario

The first scenario we test is the lane changing (Figure3.1). When a vehicle on neighboring lane cuts in, the ACC vehicle first slows down and then accelerates to make the distance between itself and the "new" preceding vehicle (the vehicle that just cut in) converge to the desired value. The ACC system uses on-board radar of ACC vehicle to detect a cutting-in vehicle when the latter passes the border of the two lanes. For CACC, we command the cutting in vehicle to send a "turning light" signal at the beginning of lane change, when the cutting-in vehicle is at the center of the lane it is in originally. We expected receiving this signal to help the ACC vehicle respond earlier to the cutting-in, since now the ACC vehicle knows about the cutting-in at the beginning of the lane change instead of after the cutting-in vehicle has passed the lane border. We added this signal in the design of the controller and ran the simulation to see the effect.

In the simulation, for simplicity we assume that the cutting-in vehicle has the same longitude velocity as the old preceding vehicle. We assume that the lane width is 10 feet and the lateral acceleration is $0.1g$. Then it take 5 seconds for the cutting-in vehicle to go laterally from the center of the neighboring lane to the center of the lane the ACC vehicle is on. In both ACC and CACC cutting-in scenarios, the cutting-in vehicle starts its lane changing at 7.5 second, passes the lane border at 10 second and arrives at the center of the target lane at 12.5 second.

Figure 3.3 and Figure 3.4 are the range and velocity of the ACC and CACC vehicles respectively. In Figure 3.3, the upper figure is the velocity of the ACC vehicle and the preceding vehicle, where the solid line is the velocity of the ACC vehicle and the dashed line is the velocity of the preceding vehicle. Notice that the cutting-in vehicle becomes the new preceding vehicle at 10 second. The bottom figure of Figure 3.3 is the range between the ACC vehicle and the preceding vehicle, including both the "original" preceding vehicle, as well as the "new" preceding vehicle or the cutting-in vehicle. The solid line is the real range while the dashed line is the desired range which is derived from a empirical formula to guarantee both safety and comfort.

In the simulation. The preceding vehicle is driving at a constant velocity of 12.5 m/s for the whole procedure of simulation. At the beginning, the ACC vehicle drives at 25 m/s and the range is 150 m. If the cutting-in does not happen, then as the ACC vehicle approaches the preceding vehicle and under the command of the ACC controller, the ACC vehicle will slow down to the same velocity of the preceding vehicle. At the same time the range will converge to the desired value, which is set to be 25 m here. However at 10 second, a cutting-in vehicle is detected, which becomes the new preceding vehicle. This detection causes the range to drop instantaneously at 10 second. The ACC controller changes its command and makes the vehicle brake very hard after the detection, which we can see clearly in the change of velocity after 10 second. After about 5 seconds, it is safe for the ACC vehicle to accelerate back to the velocity of the preceding vehicle, and eventually the range converges to the desired value.

In Figure 3.4 both the layout of the figures as well as the meaning of the lines are the same as in Figure 3.3. The only difference is that the ACC vehicle is changed to CACC vehicle. We can see in that the CACC vehicle

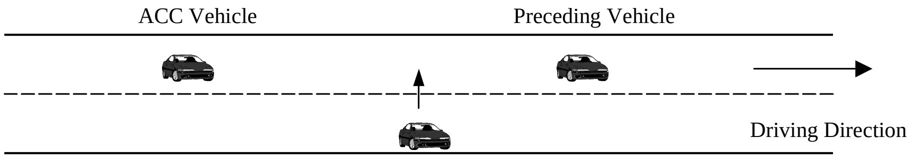

Figure 3.1: Cooperative Adaptive Cruise Control in Cutting-in Scenario

brakes at 7.5 second instead of at 10 second when the cutting-in is detected by the on-board radar. This is because of the vehicle-to-vehicle communication takes place at the instant the cutting-in vehicle starts its lane changing maneuver. The communication helps the ACC vehicle respond earlier and therefore easier, which we can see more clearly in the next four figures. The point we need to pay attention to here is that from the range-time response as well as the velocity-time response we can see that the performances of ACC controller and the CACC controller are almost the same. Therefore we should look at the control effort to see the benefit of the vehicle-to-vehicle communication in adaptive cruise control.

Figure 3.5 and Figure 3.6 are respectively the acceleration of the ACC and CACC for the simulation described above. In both the figures, the solid line is the commanded acceleration and the dashed line is the actual acceleration achieved by the vehicle model. Consistent to the changes in the velocity and range, we can see the the vehicle brakes very hard after it is notified about the cutting-in vehicle. As we discussed above, for the ACC vehicle, the cutting-in is detected by the on-board radar at 10 second, thus the vehicle has to brake pretty hard and the minimum value of the acceleration is -3m/s^2. However for the CACC case, the controlled vehicle knows about the cutting-in at 7.5 second from the communicated signal, therefore it can brake earlier and less hard. The lowest value of acceleration is -2m/s^2. By combining vehicle-to-vehicle communication with ACC and forming CACC, up to 33% of control effort is saved. The control task is made easier, safer and more comfortable for the passenger.

Figure 3.7 and Figure 3.8 confirm above results. These two figures are the phase plots of the ACC and CACC range errors respectively. The x and y axes are the range error and range rate of the controlled vehicle. The purpose of controller is to drive the vehicle such that the both the range error and range velocity converge to zero, and the phase plot goes to the origin. The

$x - y$ plane is further divided into four regions. We describe them one by one [22].

1. High Speed Cut-In: The upper region of the phase plane. Vehicles detected in this region are moving faster than the ACC vehicle. Given enough time, they will move safely away from the ACC vehicle. Hence the ACC vehicle should ignore the detected vehicle and track its driver-set velocity.

2. Normal: The region close to the origin. The Vehicles in this region are travelling at about the same relative velocity of the ACC vehicle. This is the standard vehicle-following scenario and the controller should try to force the state to the origin of the range-range rate plot (i.e. range error and range rate converging to zero).

3. Low Speed Detection: The right-bottom region. Vehicles detected in this region are moving slower than the ACC vehicle, so the brakes must be applied to prevent a collision. Since the vehicle spacing is quite large, an aggressive controller is not needed. Range tracking is not important at this point, so the range error gain should be zero. The range rate should be reduced using a low $\dot{r}$ gain which results in low decelerations.

4. Low Speed Cut-In: The left-bottom region. This region represents vehicles which have cut-in front of the ACC vehicle and have a lower velocity. Obviously this is a critical situation and an aggressive controller is needed to prevent a collision.

The Normal, Low Speed Detection, and Low Speed Cut-In regions all use the same sliding mode controller with gains tuned appropriately for the given region. The High Speed Cut-In region uses two PID controllers for desired velocity tracking: one for large velocity errors and one for small velocity errors. As described above the Low Speed Cut-In region is the critical region and the control must be aggressive in it. A good controller should make the phase stay less time in the critical region. Comparing Figure 3.7 and Figure 3.8 we can see the the phase of the CACC error does not enter this critical region while the phase of the ACC error does. This confirms our conclusion that the communication makes the adaptive cruise control easier and safer.

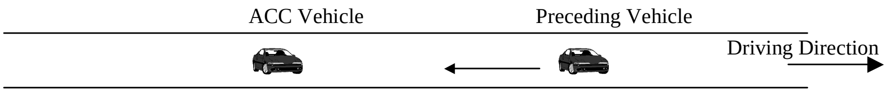

Figure 3.2: Cooperative adaptive cruise control in Braking Scenario

3.2 Braking scenario

We then study the effect of communication on the response of ACC vehicle to the braking of preceding vehicle. The scenario is shown in Figure 3.2. With the old ACC method, on-board radar is used to measure the distance between the ACC vehicle and the preceding vehicle, and the distance is numerically differentiated to get the relative velocity. These signals are feed to the controller to maintain the desired distance between the two vehicles. To improve the performance of the controller, we exploited CACC idea by making the preceding vehicle send a "braking light" signal, i.e. a message to warn the ACC vehicle about the braking. Whenever the ACC vehicle receives the signal it knows immediately about the braking, instead of after the computing of velocity. We included this signal in the adaptive cruise controller. We expect faster response caused by this additional signal. The results we get so far are shown in Figure 3.9, 3.10, 3.11 and 3.12.

Figure 3.9 and Figure 3.10 are the range and velocity for ACC and CACC respectively. In both of the figures, the upper figure is the velocity. The dashed line stands for the preceding velocity, with braking of -3m/s^2 at 20 second. The solid line is the velocity of the ACC or CACC vehicle, which tries to track the preceding velocity but with a delay caused by the radar measurement and the vehicle dynamics. It is this time response that we hope to improve, namely we hope the velocity of the controlled vehicle to track the preceding velocity faster and better.

Figure 3.11 and Figure 3.12 are the acceleration of the same simulation. consistent with the velocity results we can see sharp drop of the acceleration at 20 second.

However the simulation show no appreciable improvement. The reason is that the original controller already has very good time response character, and the delay of the radar measurement is quite small (in millisecond), which makes further improvement extremely hard. We can see clearly from the abrupt drop of acceleration in Figure 3.11 how fast the well-designed original

ACC controller can respond.

3.3 summery

Simulation shows significant saving in the control effort (up to 33%) with the same level of performance in the cutting-in scenario. At the same the position and velocity of the ACC vehicle stayed less time in the critical control condition because of the addition of the turning light signal. These results confirm our belief that inter-vehicle communication is of great benefit for the vehicle control applications. We are encouraged to continue the study both on the CACC controller design as well as the network protocol research in the future. In above simulation the controller is designed to process the light signal in a quite crude way (whenever the ACC vehicle receives a brake light signal the controller applies a preset constant deceleration to the vehicle). It is quite possible that a refined controller can achieve better performance. We will also include other information than the simple ”light” signals in the communicated messages such as the acceleration, relative position, etc. Hopefully it can improve the performance further. Although the initial simulation results are not satisfactory for the braking scenario, we still plan to do further study of the problem in hope that the improvement above in the system can enhance its performance.

In both the analysis and simulations for the above scenarios we assumed that the signals or messages were already received, without considering the performance of the communication system. We show some of the initial work in section 2.2.3. We would simulate the combined system of the vehicle model, the modified CACC controller, and the communication architecture in the future. The ACC controller should be further modified with the inclusion of the additional information.

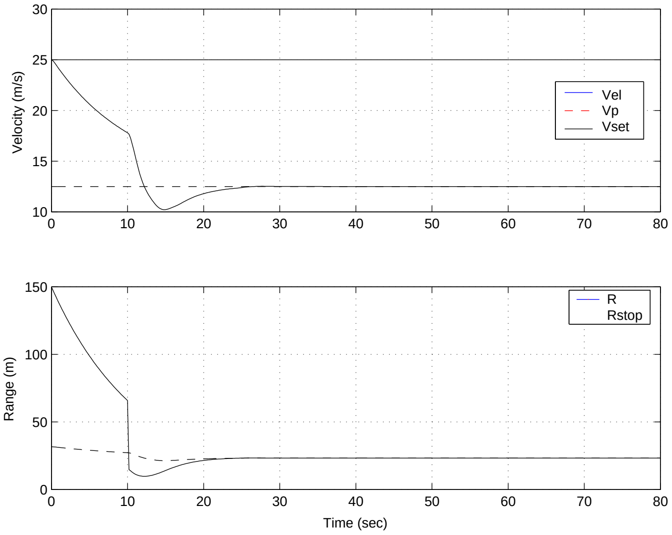

Figure 3.3: Velocity and Range of the vehicles in *ACC* Cutting-in; *Upper:*Velocity;*Bottom:*Range

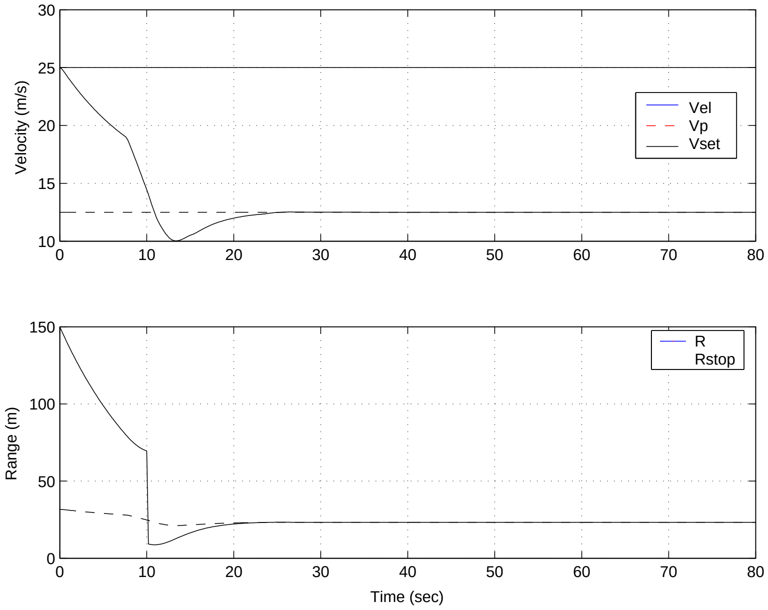

Figure 3.4: Velocity and Range of the vehicles in *CACC* Cutting-in; *Upper:*Velocity;*Bottom:*Range

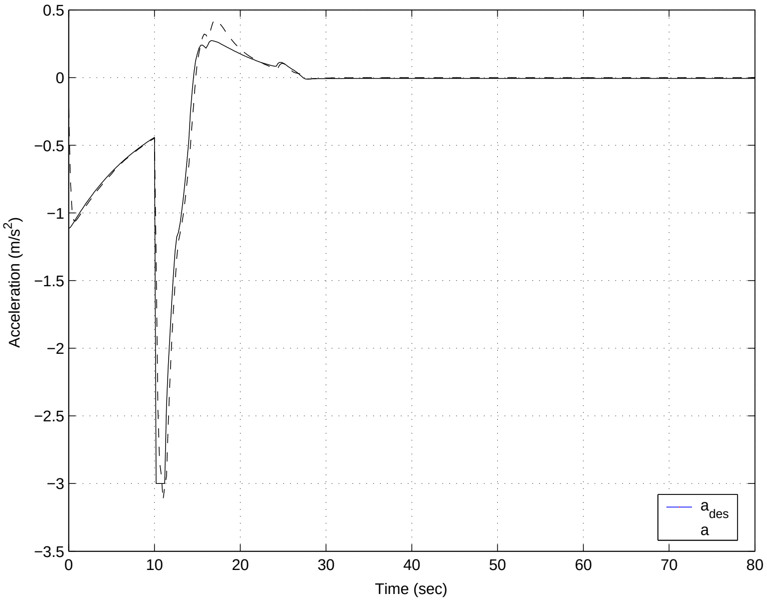

Figure 3.5: Acceleration of *ACC* vehicle (Cutting-in)

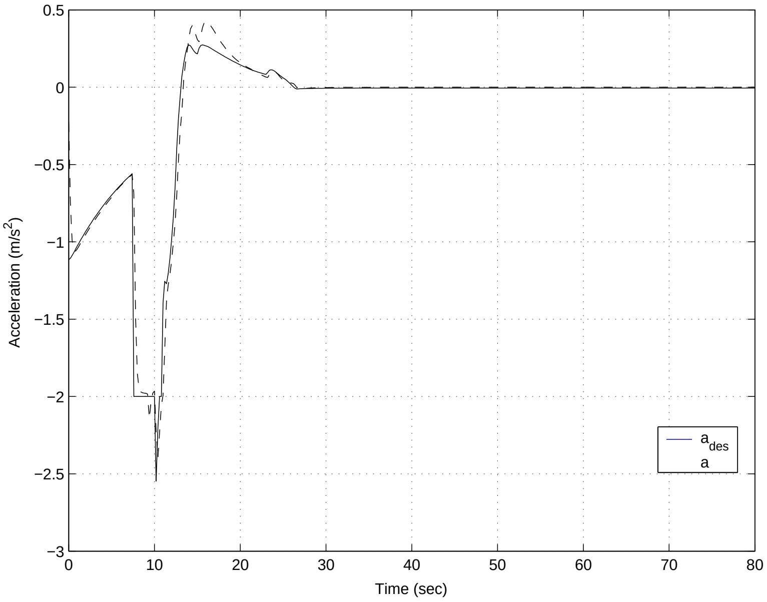

Figure 3.6: Acceleration of *CACC* vehicle (Cutting-in)

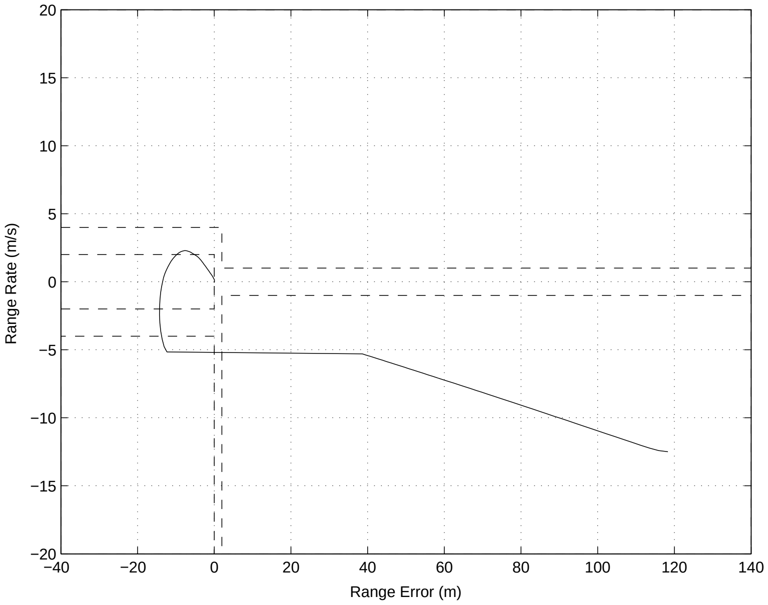

Figure 3.7: Phase plot of the *ACC* vehicle (Cutting-in)

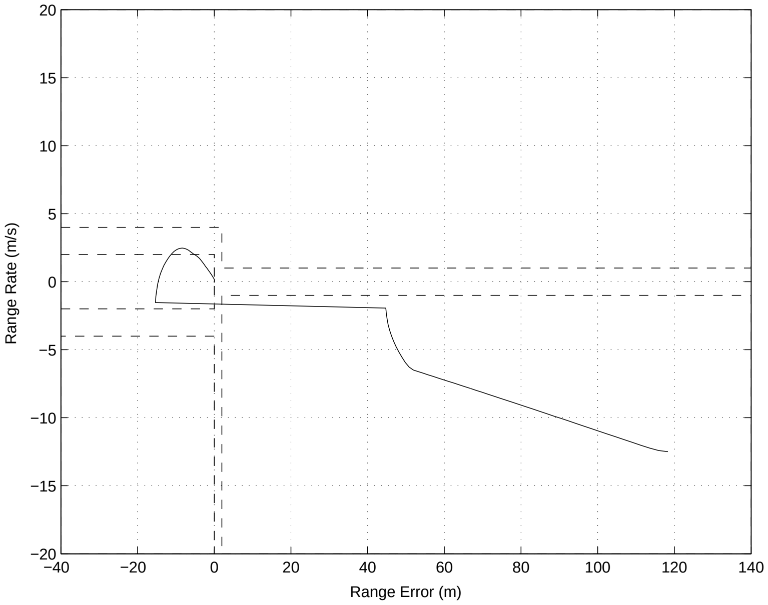

Figure 3.8: Phase plot of the *CACC* vehicle (Cutting-in)

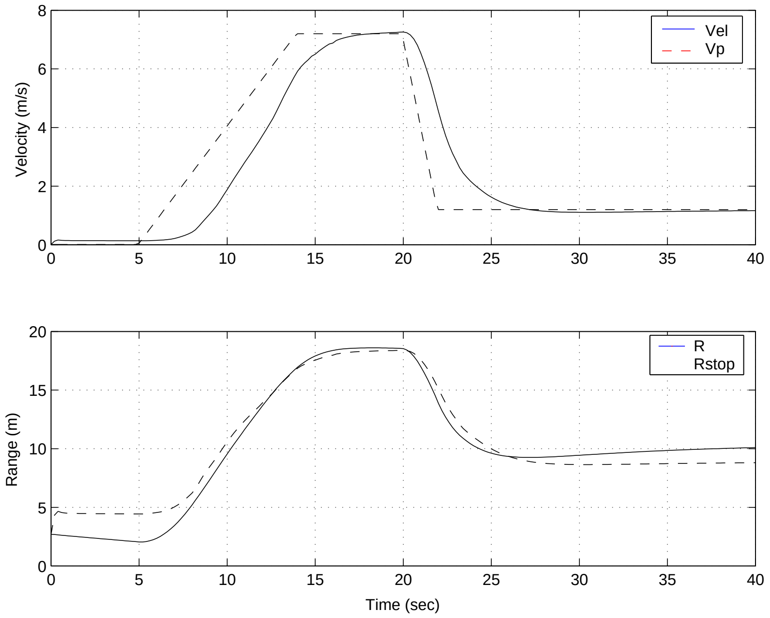

Figure 3.9: Velocity and Range of the vehicles in *ACC* Braking; *Upper:*Velocity;*Bottom:*Range

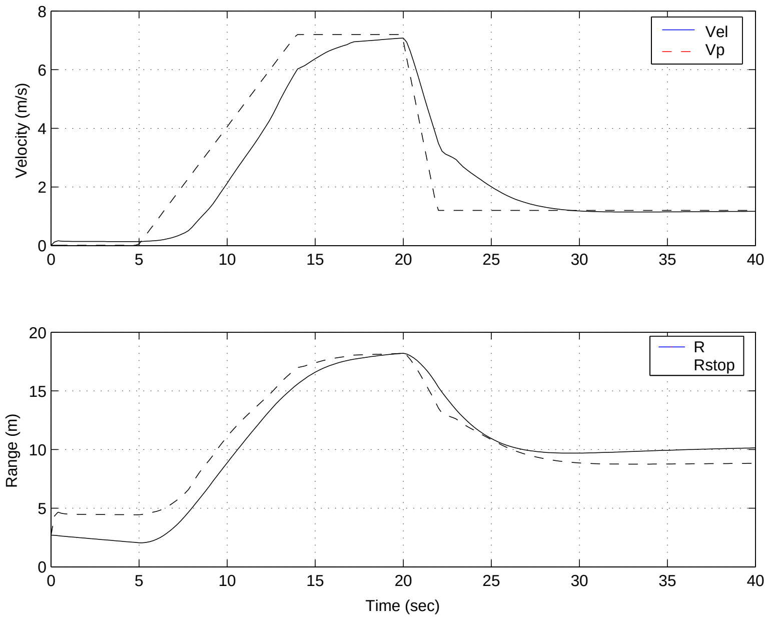

Figure 3.10: Velocity and Range of the vehicles in *CACC* Braking; *Upper:*Velocity;*Bottom:*Range

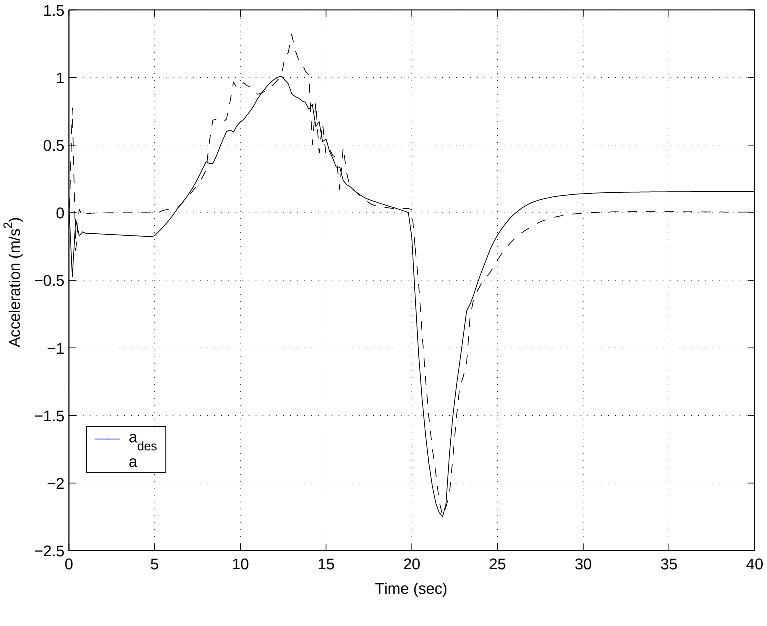

Figure 3.11: Acceleration of *ACC* vehicle (Braking)

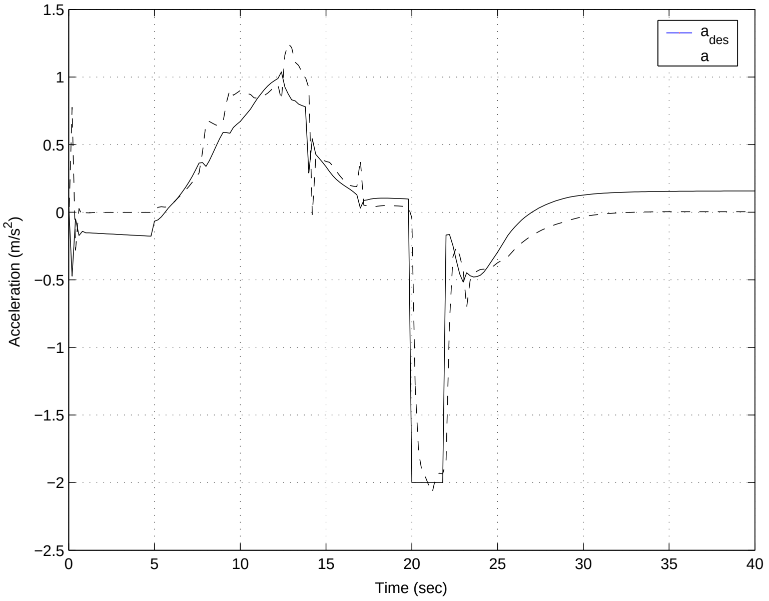

Figure 3.12: Acceleration of *CACC* vehicle (Braking)

Chapter 4

Cooperative Estimation

4.1 Introduction

The idea of what we call “cooperative estimation” is that every vehicle on a highway is a driving condition “sensor.” As it passes through a section of highway, it speeds up and slows down, perhaps indicating something about the traffic conditions; its wheels may slip, indicating something about the road condition; if it is equipped with a radar for an automated cruise control (ACC), the vehicle probably has collected some information about traffic density. This information could be valuable to the drivers that follow—whether they be human drivers or machine drivers—as they plan a strategy to cope with the road ahead.

As inexpensive, effective telecommunications technology increasingly becomes available, it may soon be possible to take advantage of such sharing. For example a human driver working with an intelligent navigation system could predict traffic jams, compare travel times, and choose an alternative route. A machine driver like an ACC could receive information about the road condition ahead and adjust how aggressively it drives.

Reaping such benefits does not necessarily imply a massive infrastructure or new equipment for every vehicle. Recently, researchers in the intelligent vehicle community have begun exploring the idea of ad hoc vehicle-to-vehicle networks using short range radios. Such networks require no infrastructure and are ideal for communicating data which is not essential to vehicle operation, but which is nevertheless nice to have. Messages can be broadcast to whatever vehicles are in range, or they can hop from vehicle to vehicle, or they can even sit latent in a travelling vehicle and hop to the target vehicles when they come within range. Initial research results using ad hoc networks show that they may be sufficiently reliable to transmit non-critical information even when market penetration is low and participating nodes are sparse [8].

We do not focus on the network design here, however. Instead, we develop ways to use communications to cooperatively estimate two quantities that might be of interest to the driver:

1. *Traffic speed and travel time*

2. *Coefficient of friction and safe acceleration levels*

The first quantity is useful for route planning, and the second is useful for choosing speeds and following distances.

By developing applications before feasible networks exist, we hope to: first, better understand the costs and benefits of vehicular networks; and second, understand how end-user requirements might drive network level specifications. We develop the algorithms with vehicle-to-vehicle ad hoc networks in mind, but they would work equally well with a fully developed infrastructure as well.

The remainder of this chapter is structured as follows: In section 4.2, we develop a way to estimate best-case and worst-case travel times using communicated velocity information. Using standard parameter estimation techniques from statistics , we are able to paint a very clear picture of the traffic flow on the road ahead. Section 4.3 then addresses the harder problem of road condition estimation. We introduce the maximum friction coefficient μ_{max} to quantify road friction, but the problem with μ_{max} is that it varies from vehicle to vehicle and that no vehicle can estimate it very well. Using communicated information, though, we can use noisy estimates of μ_{max} collected from many vehicles on a section of roadway to create a road friction index that is useful for driving. Next, in section 4.4, we modify the algorithms of sections 4.2 and 4.3 so that they are more adaptable, require less data transmission, and offer drivers more privacy. Finally, we conclude by summarizing the advantages and disadvantages of the algorithms presented here and by offering some directions for future work.

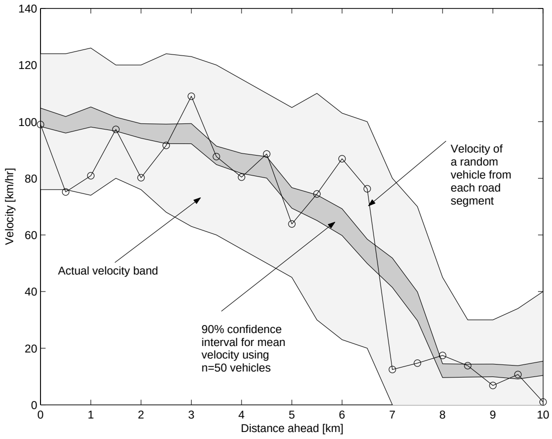

Figure 4.1: Traffic jam: Wide band is the range containing 95% of actual velocities; circles are single vehicle velocities taken from each road segment; narrow grey band is 90% confidence interval for mean velocity gotten from cooperative estimation.

4.2 Traffic Flow Estimation

Most drivers would like to know if there will be a traffic jam ahead and how much it will slow them down so that they can decide whether or not to take an alternate route. The simplest way to get this information is to check visually or to listen to a traffic report. However, traffic reports are only on intermittently and even if the traffic jam is visible, it is difficult to judge how severe it will be.

A simple cooperative approach using inter-vehicle communications, is to query vehicles ahead to find out their speeds. For example, we could query one vehicle every 500m on the road ahead for the next several kilometers and ask them their speeds. Figure 4.1 shows the results of such a query. The vehicle speeds are assumed to be normally distributed around a mean which decreases slightly until 5km, at which point it decreases rapidly to a near stand-still at 8km. The standard deviation of the speeds fluctuates slightly as a function of distance. The light grey region encompasses the mean speed each 500m plus or minus two standard deviations. Thus, the light grey region includes the speeds of approximately 95% of the vehicles on the highway. The solid line with circles shows the result of the velocity query for one vehicle each 500m. (The darker grey band will be discussed shortly.)

To arrive at a rough travel time for this stretch of road, we can divide each segment length by the vehicle speed associated with that segment and add the travel times. Using the data points marked by circles, we get a predicted time of 50 minutes to travel the next 11km. However, the travel time calculated using the mean velocity at each segment is 21 minutes. The 29 minute discrepancy is due to the noisy data points at kilometers 7 and 10. These vehicles were stopping in stop-and-go traffic and substantially biased our estimate.

This example highlights several problems with a direct querying system:

1. We must locate a vehicle in each segment that is equipped with radios and request information from it. It might be possible avoid soliciting a vehicle if each vehicle broadcasted its velocity information to vehicles behind it, but such a strategy could quickly generate many messages, particularly if vehicles many kilometers back needed the information.

2. We must query this vehicle at a time when its velocity is "representative" of that road segment. In particular, this is important for the low-speed, stop-and-go situations we are most interested in.

The cooperative alternative to direct querying that we propose is to keep a pool of velocity data points for each road segment localized in a message that is passed around that segment. Vehicles passing through the segment add their data to the pool and from time to time statistics that summarize the data are processed and passed back to vehicles that might find them useful.

Figure 4.2 summarizes such a strategy. Each of the segments has a database associated with it. For an ad hoc network, the database can be passed from vehicle to vehicle, or one "master vehicle" for that segment can keep it. If no properly equipped vehicles are present in the segment, or if the vehicle with the data is leaving the area, the database can be passed back to an equipped vehicle that will soon be in the area. With a roadside infrastructure, the database would stay in the computer for this road segment. Independent of the implementation, the data pool stays near its associated road segment as much as possible and its statistics are broadcast back to users. (In section 4.4, we discuss how to avoid keeping a large pool of data at the road segment. In that section, it will be an iteration that remains attached to each road segment and the iteration will "contain" the data.)

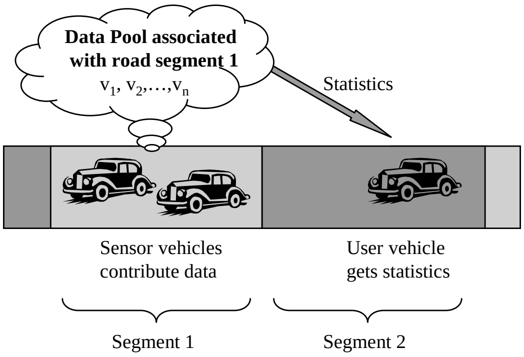
Data Pool associated with road segment 1
v1, v2,…,vn
Statistics
Sensor vehicles contribute data
User vehicle gets statistics
Segment 1
Segment 2

Figure 4.2: Data pools associated with road segments. Vehicles in the road segment are “sensors” and contribute to the pools. “User” vehicles further down the road receive statistics in a streamlined message that hops far down the road.

We assume that the data pool for a particular section of road contains n recent velocity samples, $v_1, v_2, \ldots, v_n$, from the population of vehicles that have passed through the segment. Obtaining an interval estimate for the mean velocity in this road segment is a standard statistical problem. If the velocities are normally distributed, then the $(1-\gamma)\cdot 100\%$ confidence interval for the mean velocity is given by

$$\bar{v}_n \pm t_{n-1,1-\gamma/2}\sqrt{\frac{S_n^2}{n}} \tag{4.1}$$

where $\bar{v}_n = (1/n)\sum_{i=1}^{n} v_i$ is the sample mean using n data points, $S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(v_i - \bar{v}_n)^2$ is the sample variance using n data points, and $t_{n-1,x}$ is the xth quantile of the Student T distribution with $n-1$ degrees of freedom.

Figure 4.2 shows the performance of such an estimator. The probability density function (not to scale) for the velocities in this segment of road is shown along the velocity axis, and the shaded region is the 90% confidence interval for the mean velocity in this road segment plotted against n, the number of vehicles contributing to the data pool associated with this road segment. After 30 vehicles contribute, the interval becomes quite thin and by dividing the segment width by the lower and upper limits of the confidence interval, we have a 90% certain estimate of the mean time it will take to

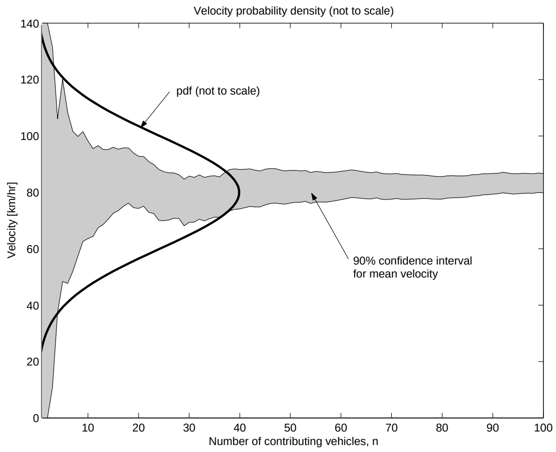

Figure 4.3: Cooperative estimation for a single road segment: Probability density function of traffic speeds (not to scale) and 90% confidence interval for mean velocity as a function of number of vehicles contributing to data pool.

traverse this road segment. This travel time estimate or the raw statistics $\bar{v}_n$ and S_n^2 could then be passed back to the "user" vehicles.

Using this statistically-based cooperative estimation scheme, we return to the example traffic jam example of figure 4.1. The thin grey stripe is the 90% confidence interval for the mean velocity at each of the road segments, generated using data from $n = 50$ vehicles.

We arrive at a worst case travel time of 22 minutes for this 11km segment by dividing the segment widths by the lower limit of the mean velocity interval. Similarly, the best case travel time is 16 minutes. Recalling from above, the mean travel time was 21 minutes, and the travel time gotten from single samples from each road segment was 50 minutes.

Thus, cooperative estimation is one way to obtain accurate travel times and to detect traffic jams without having to accurately pinpoint the locations of specific vehicles or track them. (The vehicles need to know what road segment they are in, but this requires only very rough positioning.) Coupled with a GPS-equipped mapping system, such a system could detect a traffic jam, predict how long it would take, possibly compare with data from other routes, and then suggest an alternative.

To work, though, a cooperative velocity estimation system would need enough market penetration so that at least 30 data points in each road segment could be collected fast enough so that changes in traffic patterns could be ignored. In addition, the network would need to be able to maintain the data pools for each road segment as well as transmit the longer range messages containing statistics about the traffic ahead. Finally, drivers would have to be comfortable with the fact that their vehicle was sharing data about its whereabouts and speed. We address the data pool size and privacy concerns in section 4.4, "Practical Issues," below.

4.3 Coefficient of Friction Estimation

Another important quantity to estimate—especially for machine driving systems—is the road condition. Here, we first establish an index μ_{max} to quantify road condition, and we describe sources of uncertainty in measuring it. Next, we introduce a quantity μ_{safe} which gives us a conservative estimate of a vehicle's maximum possible acceleration on a road segment. Finally, we develop and demonstrate an algorithm similar to that of section 4.2 to estimate μ_{safe}.

A way to quantify how easily a vehicle will skid on a road is with the maximum coefficient of friction, μ_{max}. For a given wheel, the normalized traction force, μ, is

$$\mu := \frac{\sqrt{F_\xi^2 + F_\eta^2}}{N_\zeta},$$

where F_ξ, F_η, and N_ζ are the tire forces along the axis of the coordinate system (ξ, η, ζ) attached to the tire. For this wheel μ_{max} is then the maximum of the magnitude of μ:

$$\mu_{max} := \max|\mu|$$

Here, we consider only longitudinal motion, so the side force F_η can be neglected, giving

$$\mu = \frac{F_\xi}{N_\zeta} \tag{4.2}$$

and $\mu_{max} = \max|\mu| = \max|F_\xi/N_\zeta|$. For the remainder of this section when we discuss μ_{max}, we mean this longitudinal-only quantity.

When a vehicle of mass m has the same μ_{max} at all four of its tires, the largest longitudinal acceleration it can achieve (neglecting grade and wind) is the maximum of the sum of the longitudinal forces at its tires divided by the vehicle mass:

$$\begin{aligned}
|a|_{max} &= \max |\frac{F_{\xi_{11}} + F_{\xi_{12}} + F_{\xi_{21}} + F_{\xi_{22}}}{m}| \\
&\leq \frac{\mu_{max} N_{\zeta_{11}} + \mu_{max} N_{\zeta_{12}} + \mu_{max} N_{\zeta_{21}} + \mu_{max} N_{\zeta_{22}}}{m} \\
&\approx \frac{\mu_{max} mg}{m} = \mu_{max} g
\end{aligned}$$

So estimating μ_{max} gives us an upper bound on the acceleration, in g's, that a vehicle can achieve. ABS, TCS and VDC systems start working when the driver has demanded an acceleration larger than $\mu_{max} g$, but neither they nor a μ_{max} estimator can increase this acceleration limit.

This means that a maximum coefficient of friction estimator can tell us when we risk crashing, but it can do very little to help us once a crash is kinematically inevitable. Thus, a μ_{max} estimator is useful primarily because it can inform drivers of dangerous conditions so that they can change their driving style to prevent emergency situations.

Unfortunately, predicting what μ_{max} will be if a vehicle finds itself in an emergency situation is difficult for two reasons:

1. μ_{max} varies depending on physical parameters, including vehicle type, tire type, road type, and location on a particular road segment. For example, [14] shows that the standard deviation of μ_{max} for the same test surface can be as large as 0.05 (compared to μ_{max} values that typically range from 0.1 to 1.1). In [30] and [29], variations on snowy and icy surfaces are studied.

2. Without actually making the vehicle skid, μ_{max} is notoriously difficult to measure. Over the past decade, researchers have taken significant steps towards developing systems to predict what μ_{max} will be for vehicles without actually making the vehicle skid. For example, in [16] a system using a combination of optical sensors was able to give an unbiased prediction of μ_{max}. In [20], [24], [40], [11] and others, researchers explored so-called "slip-based" techniques and showed that they may be able to give acceptable μ_{max} estimates. Despite these advances, a typical measurement of μ_{max} taken without skidding can still be expected to have significant noise, or worse, significant bias.

For this analysis, we assume that the physical μ_{max} is normally distributed with variance σ_p^2 around some mean value $\mu_{max_{mean}}$ We further assume that the instantaneous measurement of μ_{max} for a particular vehicle, μ_i, (where i indexes the vehicle) is normally distributed with variance σ_m^2 around the true value of the maximum friction coefficient, μ_{max_i}, for that vehicle. Thus, we must work with unbiased noisy measurements of the physically variable quantity μ_{max}.

Assuming that we knew the underlying distribution for the physical μ_{max}, a reasonable quantity to use to set driving policy would be a "safe" coefficient of friction, μ_{safe}, chosen so that if we needed to use all of the road's friction, we could be reasonably certain that our vehicle's μ_{max_i} would be larger than μ_{safe}. Since the physical distribution of μ_{max} is assumed to be normal with mean $\mu_{max_{mean}}$ and variance σ_p^2, this theoretically desirable μ_{safe} is given by

$$\mu_{safe} = \mu_{max_{mean}} - z_{1-\gamma}\sigma_p \tag{4.3}$$

where γ is the confidence level and z_p is the xth quantile for the unit normal distribution.

Figure 4.3 shows an underlying physical μ_{max} distribution (not to scale), as well as μ_{safe} for $\gamma = 0.05$. It also shows 100 noisy measurements of the maximum friction coefficient,$\mu_1, \mu_2, \ldots, \mu_{100}$, given by the equation

$$\mu_i = \mu_{max_i} + w_i \tag{4.4}$$

where μ_{max_i} is taken from the distribution $N(\mu_{max_{mean}}, \sigma_p)$ and where the measurement noise w_i comes from the distribution $N(0, \sigma_m)$.

To estimate μ_{safe} from n of the noisy measurements, $\hat{\mu}_{safe_n}$ at a confidence level γ, we use techniques similar to those used to derive equation 4.1 (see, for example [41]) to get

$$\hat{\mu}_{safe_n} = \bar{\mu}_n - t_{n-1,1-\gamma}\sqrt{\frac{S_n^2}{n} + \frac{S_n^2}{1 + \frac{\sigma_m^2}{\sigma_p^2}}} \tag{4.5}$$

where $\bar{\mu}_n$ is the sample mean of the n measurements $\mu_1, \mu_2, \ldots, \mu_n$, S_n^2 is the sample variance of the n maximum friction coefficient measurements. Since the physical and measurement variances σ_p^2 and σ_m^2 are typically unknown, we lower bound the ratio of variances and substitute this lower bound in place of the variance ratio in order to have a conservative value of $\hat{\mu}_{safe_n}$.

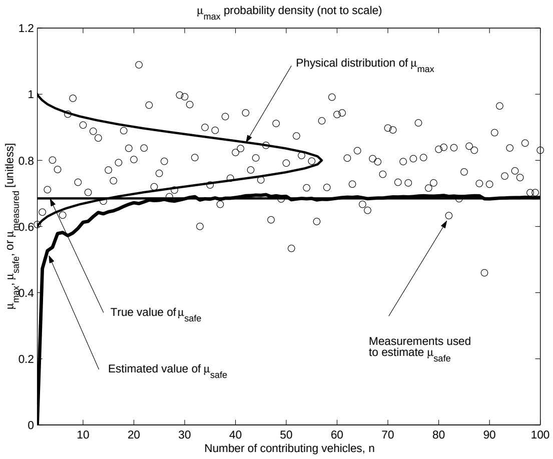

Figure 4.4: Cooperative road condition estimation for a single road segment: Physical distribution of μ_{max}, theoretical μ_{safe}, noisy μ_{max_i} measurements, and $\hat{\mu}_{safe}$.

The dark solid line in figure 4.3 shows the convergence of this μ_{safe} estimator as a function of the number of vehicles contributing to the data pool for a particular road segment. The correct value for the variance ratio mentioned above is used in this figure. When a smaller value of the variance ratio is used, $\hat{\mu}_{safe_n}$ converges to a lower value. After approximately 30 vehicles, the estimator converges.

Figure 4.3 shows how this estimator works in a cooperative estimation scenario. The road is mostly dry and therefore has a maximum friction coefficient of about 1 with standard deviation of approximately 0.06. In the center of the road, though, there is a slippery segment with a large standard deviation. The dotted line shows the $\mu_{max_{mean}}$ for the road, and the solid line below it which bounds the light grey region is the physical μ_{safe}. The line below that is $\hat{\mu}_{safe_n}$ obtained using $n = 50$ data points per road segment and equation 4.5. (The variance ratio is fixed to a conservative value for this simulation.) Finally, the circles represent the responses we would have gotten if we had forgone the more complex estimation scheme and simply queried one vehicle per road segment for its current measurement of the maximum friction coefficient. We see that the cooperative friction estimation scheme presented here provides a conservative estimate of μ_{safe} that is much better than any of the vehicles could have obtained alone.

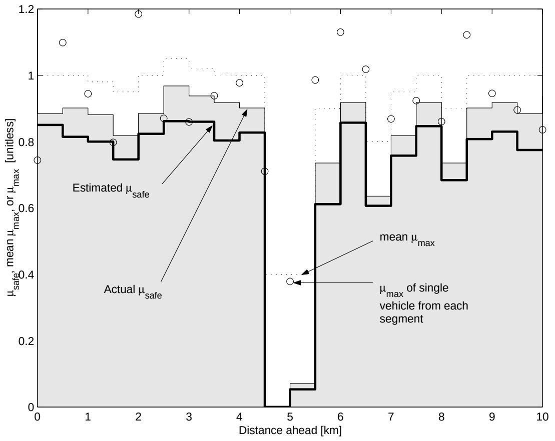

Figure 4.5: Cooperative road condition estimation to detect a slippery section of road.

4.4 Practical Issues

The cooperative estimation schemes proposed above bring up three problems: First, maintaining a data pool for a road segment becomes more difficult as the number of participants, n grows. Even if the data pool resides in a "master vehicle," it still needs to be re-transmitted when the "master vehicles" leaves the road segment. Second, the estimates assume that there are steady state conditions. And third, motorists might be concerned about transmitting their speed and position

4.4.1 Recursive formulation

Making the sample mean and sample variance calculations recursive remedies two of the problems mentioned above. It remedies the bandwidth problem by allowing the amount of data used to calculate the sample statistics to grow to hundreds or even thousands of vehicles while only requiring that the communication system pass a few key quantities. As an interesting side-effect, it lessens privacy problems because it requires that only conglomerated data passes over the network.

We obtain the recursive formula for the sample mean by rearranging its definition. (It can also be obtained from an appropriately chosen Kalman Filter—see [21], for example.) Let $\bar{\mu}_n$ denote the sample mean of the maximum friction coefficient using n samples, and let μ_i denote the i^{th} sample of the friction coefficient. Then we obtain $\bar{\mu}_n$ in terms of the previous sample mean, $\bar{\mu}_{n-1}$, and the latest sample, μ_n, as follows:

$$\bar{\mu}_n = \frac{1}{n}\sum_{i=1}^{n}\mu_i = \frac{1}{n}\mu_n + \frac{1}{n}\sum_{i=1}^{n-1}\mu_i = (1-\frac{1}{n})\bar{\mu}_{n-1} + \frac{1}{n}\mu_n \tag{4.6}$$

Similar but more involved rearrangement also yields a formula for S_n^2, the sample variance of the friction coefficient using n samples, in terms of the previous sample variance S_{n-1}^2, the previous sample mean $\bar{\mu}_{n-1}$, and the latest coefficient of friction sample μ_n:

$$S_n^2 = \frac{n-2}{n-1}S_{n-1}^2 + \frac{1}{n}(\mu_n - \bar{\mu}_{n-1})^2 \tag{4.7}$$

So only the three numbers n, $\bar{\mu}_{n-1}$, and S_{n-1}^2 need to be passed from vehicle to vehicle in order to calculate the statistics we used in the previous two sections. Participating vehicles receive this information, store the quantities they are interested in, add their own data, and then pass the message on. Without tracing the route that the message took from vehicle to vehicle, it becomes difficult to assign a data point to any particular vehicle.

4.4.2 Time varying data

Both the velocity distribution or the maximum coefficient of friction distribution on a section of road can change on time scales of tens of minutes. For example, an accident can quickly produce a traffic jam and the arrival of dusk can freeze sections of wet road. It is therefore desirable that cooperative estimators "forget" old data and weigh recent data more heavily.

For example, one way to achieve this for the sample mean of the maximum coefficient of friction is to use the formula

$$\bar{\mu}_n^w = (1-\alpha)\bar{\mu}_{n-1}^w + \alpha\mu_n \tag{4.8}$$

where $\bar{\mu}_n^w$ is the weighted sample mean using n data points, $\bar{\mu}_{n-1}^w$ is the weighted sample mean using $n-1$ data points, μ_n is the new measurement of the coefficient of friction, and α is a number between 0 and 1. Large α gives more weight to new data and "forgets" old data more easily. (See [41]

Chapter 13 for more detail.) Similarly, a recursion for the sample variance that forgets old data is

$$S_n^{2w} = (1 - \frac{n}{n-1}\beta)S_{n-1}^{2w} + \beta(\mu_n - \bar{\mu}_{n-1}^{w})^2 \tag{4.9}$$

where S_n^{2w} is the weighted sample variance of the maximum coefficient of friction using n data points, S_{n-1}^{2w} is the weighted sample variance using $n-1$ data points, β is a number between 0 and 1 that determines "forgetting" speed, and μ_n and $\bar{\mu}_{n-1}^{w}$ are as above. In the special case when the probability density of the coefficient of friction does not change, 4.9 gives an unbiased estimate of the variance of μ.

To calculate a time varying μ_{safe}, these weighted quantities can then be used in equation 4.5 instead of their un-weighted counterparts. Figure 4.4.2 shows the performance of such a μ_{safe} estimator for a single road segment. The first 100 vehicles that pass through the road segment experience a mean μ_{max} of 0.8 with a standard deviation of 0.07. The mean μ_{max} value is plotted as a dotted line. After the 100th vehicle, the maximum friction coefficient suddenly drops to a mean value of 0.3 with the same standard deviation. The shaded region shows the corresponding drop in the ideal μ_{safe} value that we would calculate if we knew the probability densities of μ_{max}. Of course, we do not know the probability densities of μ_{max}, so to calculate an estimate of μ_{safe}, we use noisy estimates of μ_{max} from each of the contributing vehicles as in Section 4.3. The solid line shows this estimate of μ_{safe} using obtained from equation 4.5, but using equations 4.8 and 4.9 (with $\alpha = \beta = 0.03$) for the mean and variance.

Approximately 40 vehicles after the drop, the estimate of μ_{safe} is converged. Assuming a traffic flow of 1000 vehicles/hr, 10 percent of which participate in the estimation, this adjustment period is about 25 minutes.

4.5 Conclusions and future work in cooperative estimation

We have shown two applications where cooperative estimation was able to generate estimates that are more useful than those which a vehicle could generate on its own:

1. Velocity and travel time estimation

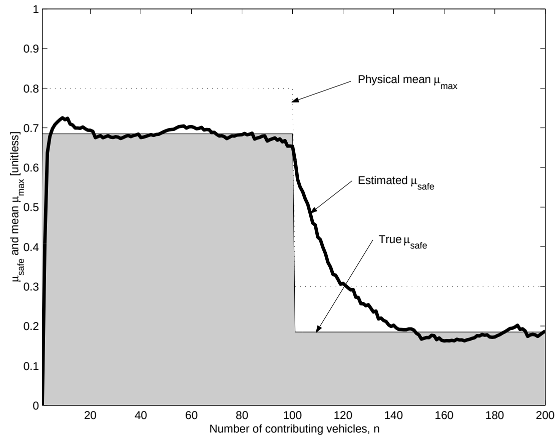

Figure 4.6: At iteration 100, mean coefficient of friction reduces with a corresponding decrease in μ_{safe}. Estimate of μ_{safe} that is calculated using the recursions of section 4.4.2 adjusts to the change.

2. Coefficient of friction estimation

The algorithms capture the experience of many vehicles in a few key statistics and require a relatively small amount of communication. They can work either with an ad hoc network or a centralized network.

However, several issues need to be resolved before such a cooperative estimation scheme can be practical:

1. Realistic models: We used greatly simplified highway "models" for this study. A traffic simulator would be a superior test-bed for evaluating the benefits of cooperative estimation.

2. Integration with other technologies: For a human driver, just driving a vehicle is taxing. Making sense of travel times and estimates of μ_{safe} is more than most drivers can handle. Thus, the improved information that cooperative estimation can provide will be wasted unless it can be presented in a way that is meaningful and non-distracting to the driver.

3. Low market penetration: When market penetration is low, it is unlikely that a cooperative estimation system will work very well, if at all. How a system can be made to work when participants are sparse is not yet resolved.

4. Network architecture: The field of automotive networking is still evolving. It must be determined if cooperative estimation fits in with other uses of automotive networks and if reliable yet inexpensive communication architectures for automotive networking can be developed.

5. Implications of better informed vehicles: If all vehicles equipped with cooperative travel time estimation exit the highway when they detect a traffic jam ahead will the system still be helpful?

Chapter 5

Slip-based Road Condition Estimation

5.1 Introduction

As we mentioned in Chapter 4, by "tire/road friction estimation" we mean that we would like to predict how easily a vehicle will skid on a road without actually making the vehicle skid.

As we mentioned before, one way to quantify how easily a vehicle will skid is with the maximum coefficient of friction, μ_{max}. In this chapter, we investigate the μ_{max} estimation problem at its crux at the vehicle level.

The so-called "slip-based" μ_{max} estimation strategy we pursue here is an exciting research area with potential near-term applications, especially for AHS and adaptive cruise controls. Before discussing the details of this strategy, however, let us motivate the problem more deeply than we did in the previous chapter.

5.1.1 Applications and requirements for μ_{max} estimator

A μ_{max} estimator is useful primarily because it can inform drivers –human drivers and machine drivers–of dangerous conditions so that they can change their driving style to prevent emergency situations. Below, we summarize how they might use μ_{max} information.

To demonstrate how a μ_{max} estimator might be useful for a machine driver, consider designing an intelligent cruise control system according to the following specification: " The maximum operating range of the used radar is 100m. For any detected object in the roadway–for example, a wall of cars creeping along in a traffic jam–the cruise control must be able to stop the vehicle safely without hitting the object."

To meet the specification, the driver must adjust the driving speed according to the operating range of the radar and the road condition. From kinematics, the stopping distance as a function of the velocity v is $d = v^2/\left(2\mu_{max}g\right)$. With $v = 150$km/h and $\mu_{max} = 1$ (dry road), the necessary stopping distance is 89m. However, if the road is wet so that μ_{max} is 0.7, the distance needs to be 127m. And if the road is icy so that μ_{max} is 0.2 the vehicle needs 445m to stop safely. If we have no μ_{max} estimator, we are forced to choose a conservative driving speed to accommodate the worst case when μ_{max} is 0.2.

If we have a μ_{max} estimator, on the other hand, we can adapt the cruise control's driving policies according to road conditions. For example, the system could disable itself if there is evidence that μ_{max} is less than, say, 0.5. Drivers would not be able to operate the system in adverse conditions, and it would be guaranteed to never "out-drive its radar."

For automated highway systems—another type of machine driver—a μ_{max} estimator could do more than just help to set driving policies. In an environment with vehicle/roadside communication, each vehicle could be a μ_{max} sensor that reports back to a roadside database. The roadside could then construct a friction vs. position map of the roadway that would be useful for pinpointing slippery spots and adjusting driving and maintenance to compensate.

A μ_{max} estimator could also help human drivers because there are surprisingly few ways for them to estimate μ_{max}. Checking if the road is snowy or icy or wet is often effective, but a visual inspection cannot pick up black ice or ice under snow. Wet pavement after the first rain in weeks looks similar to wet pavement that has been washed clean by several days of rain, despite their different coefficients of friction [33]. It is precisely in these deceptive situations where a μ_{max} estimator could have the greatest safety benefit for human drivers. In order to realize this benefit, though, the estimator would need to be fairly precise, and it would have to present its results in a way that is relevant and non-distracting to the driver.

5.1.2 State of the art

This chapter develops a "slip-based" μ_{max} estimator that works during braking. Figure 5.1 shows where this contribution fits into the much larger research area of tire-road friction estimation, and it provides a framework within which we examine recent μ_{max} estimation results in the literature.

As the top branch of Figure 5.1 shows, tire-road friction estimation research can roughly be divided into "cause-based" approaches and "effect-based" approaches. "Cause-based" strategies try to measure factors that lead to changes in friction and then attempt to predict what μ_{max} will be based on past experience or friction models. "Effect-based" approaches, on the other hand, measure the effects that friction (and especially reduced friction) has on the vehicle or tires during driving; they then attempt to extrapolate what the limit friction will be based on this data. For example, if a human driver sees ice on the road and uses past experience to conclude that the road will be slippery, he is using a cause-based μ_{max} estimation strategy. If he does not see the ice, spins his tires while accelerating, and then concludes that the road must be slippery, then he is using an effect-based estimation strategy. Below, we first review some results of cause-based μ_{max} estimation research, and then we examine effect-based research, focusing special attention on the category of "slip-based" methods.

Cause-based μ_{max} prediction

Numerous parameters "cause" μ_{max} to be a certain value. In [3], Bachmann classifies them as *vehicle parameters* like speed, camber angle, and wheel load; *tire parameters* like material, tire type, tread depth, and inflation pressure; *road lubricant parameters* like type (water, snow, ice, oil), depth, and temperature; and *road parameters* like road type, micro-geometry, macro-geometry, and drainage capacity. A cause-based μ_{max} predictor must be able to measure the most significant friction parameters and then produce an estimate of μ_{max} from a database with information about the effects of these parameters.

Many of the parameters affecting μ_{max} are easily determined—for example, speed, tire type, approximate wheel load, and camber. However, measuring two of the parameters that significantly affect friction—lubricant and road type—requires special sensors. This need for extra sensors is one of the main disadvantages of cause-based friction estimation approaches.

As the "Lubricants" branch of Figure 5.1 shows, several researchers have built special lubricant sensors for friction estimation. The optical sensors described in [15] and [1] can detect water films and other lubricants by examining how the road scatters and absorbs light directed at it. Optical sensors have also been constructed to detect the road surface roughness characteristics [1].

Once the parameters affecting friction are known, they must be passed into a friction model of some sort to obtain a μ_{max} prediction. This friction model could be theoretical or physically based, but many researchers have suggested using neural networks and other learning algorithms instead. In [15], for example, the μ_{max} prediction software uses data interpolation, associative storage, and system identification techniques. The disadvantage of this type of nonphysical model is that it loses accuracy when conditions deviate from the conditions under which it was "trained."

Nevertheless, experimental results have shown that cause-based μ_{max} estimators can often deliver high accuracy. For example, in [15], a cause-based method using data from a wetness sensor and a surface roughness sensor gives a μ_{max} estimate that is within 0.1 of the real value of μ_{max} in 92% of experiments. Since the key sensors were optical, these results were obtained with zero friction demand. That is, the driver did not need to achieve high levels of μ to get a useful estimate of μ_{max}.

As we mentioned above, though, these advantages of good accuracy and zero friction demand come with three main disadvantages: First, cause-based systems often require extra sensors. Second, they may need extensive "training" to work properly. And third, they may have difficulties accurately predicting friction under exceptional conditions for which they have neither sensors nor training.

Effect-based μ_{max} prediction

As Figure 5.1 shows, researchers have pursued at least three types of "effect-based" μ_{max} estimators. They are acoustic approaches, tire-tread deformation approaches, and slip-based approaches. We briefly review the acoustic and tire-tread approaches first, and then we provide a more detailed review of slip-based approaches, since the new work of this chapter is in this area.

In an acoustic approach, a microphone is mounted to "listen" to the tire, and the sound that the tire makes is used to infer something about μ_{max}.

According to [15] and [1], the tire noise correlates with the friction demand and deformation of the tire tread, so it is an effect of tire-road friction. At the same time, though, these authors show that the noise is also correlated with parameters that affect friction such as road type, presence of water, and speed. Thus, tire noise indicates something about both the causes and the effects of tire-road friction, so it could have just as easily been classified as a cause-based approach. Regardless of how one classifies this approach, the complex nature of tire noise makes it difficult to use for predicting μ_{max}.

The tire-tread deformation approach uses sensors embedded into the tire tread to measure the x, y, and z deformations of the tread as a function of its position in the road-tire contact patch. These deformations are the direct result of x, y, and z force transmission in the contact patch and therefore contain information about the total longitudinal, lateral, and normal forces as well as their geometric distributions in the contact patch. This is useful for estimating μ_{max} because individual tire tread elements often exceed the holding power of the road long before the tire as a whole exceeds μ_{max} and starts sliding. Thus, we see the effects of the μ_{max} limit on the tire before we see its effect on vehicle performance.

For example, even in a free-rolling tire, the tread deforms in the longitudinal direction as it flattens to enter the contact patch and then retakes its natural shape on exiting. The shear stresses associated with this free-rolling deformation can be quite large—as much as 100 kPa, compared to normal pressures on the order of 200 kPa [13]. If the road-tire interface is unable to provide enough shear stress because μ_{max} is small, certain parts of the contact patch may slide slightly, leading to changes in the tread deformation geometry that are correlated with μ_{max}. When friction demand is non-zero, one might expect even more local sliding in the contact patch, potentially providing more information about μ_{max}.

References [15], [6], [2], and [1] describe a tire-tread deformation sensor and give experimental results for a μ_{max} estimator that uses tread deformation. The sensor consists of a magnet vulcanized into the tread of a kevlar-belted tire (to avoid signal distortion from a steel belt) and a detector fixed to the inner surface of the tire. Experiments using this apparatus show that even with zero friction demand it is possible to detect very low μ_{max} surfaces from tire-tread deformation data. Furthermore, the system does not need to know why the road is slippery to work since it only measures the effects of low μ_{max}. Thus, it is immune to many of the problems of cause-based μ_{max} identifiers. While very promising, this approach has the

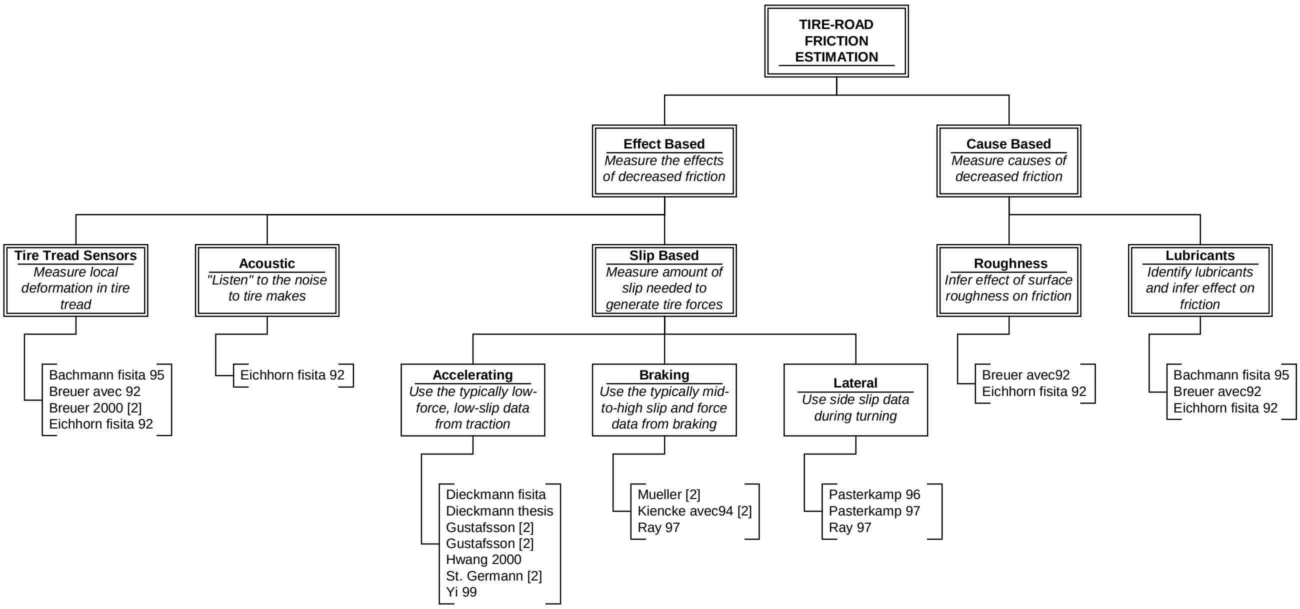
TIRE-ROAD FRICTION ESTIMATION
Effect Based
Measure the effects of decreased friction
Cause Based
Measure causes of decreased friction
Tire Tread Sensors
Measure local deformation in tire tread
Acoustic
"Listen" to the noise to tire makes
Slip Based
Measure amount of slip needed to generate tire forces
Roughness
Infer effect of surface roughness on friction
Lubricants
Identify lubricants and infer effect on friction
Bachmann fisita 95
Breuer avec 92
Breuer 2000 [2]
Eichhorn fisita 92
Eichhorn fisita 92
Accelerating
Use the typically low-force, low-slip data from traction
Braking
Use the typically mid-to-high slip and force data from braking
Lateral
Use side slip data during turning
Breuer avec92
Eichhorn fisita 92
Bachmann fisita 95
Breuer avec92
Eichhorn fisita 92
Dieckmann fisita
Dieckmann thesis
Gustafsson [2]
Gustafsson [2]
Hwang 2000
St. Germann [2]
Yi 99
Mueller [2]
Kiencke avec94 [2]
Ray 97
Pasterkamp 96
Pasterkamp 97
Ray 97

Figure 5.1: A sampling of tire-road friction estimation research.

disadvantage that it requires a sophisticated instrumented tire with a wireless data link to the vehicle—something that appears to be several years in the future on production vehicles.

It is primarily the desire to minimize sensors that makes the third effect-based approach—the slip approach—so attractive. As the "slip-based" branch of Figure 5.1 shows, researchers have worked to develop slip-based μ_{max} estimators using data from traction, braking, and steering maneuvers. From current research results, it appears that slip-based estimators using traction data may be capable of classifying roads into three or four friction levels using only standard ABS wheel speed sensors. Estimators using braking data may provide additional data with the addition of a longitudinal accelerometer, and slip-based estimators using steering data may work with a yaw rate gyroscope and a lateral accelerometer. Although accelerometers and yaw rate gyroscopes are not standard equipment on most cars, the increasing popularity of vehicle dynamics control systems (which require these sensors) may soon change this. Here, we focus mostly on the traction and braking cases since they relate most closely to the new work presented later in the chapter.

For traction or braking, the slip, s, of a wheel is the scaled difference between the longitudinal translational speed of that wheel, v, and the rotational speed of the wheel. We use the definition:

$$s = \frac{\omega r - v}{\max(\omega r, v)} \tag{5.1}$$

where ω is the angular speed of the wheel and r is the effective tire radius. For a discussion of the slip definition in (5.1) see Appendix A and A.

A braking wheel has a smaller rotational speed than its translational speed, so for braking this equation has negative numerator. The max in the denominator forces this negative velocity difference to be normalized by v, resulting in $s = -1$ if the braking wheel locks. On the other hand, an accelerating wheel has a positive numerator, and the denominator becomes ωr so that $s = +1$ if the vehicle stands still while the wheels spin.

The friction coefficient, μ, at a tire is related to the amount of slip at that tire. The most well-known model for this relationship is the so-called "Magic Formula" [4] which we plot in Figure 5.2 for traction and braking on a variety of road surfaces. Figure 5.2 shows that μ is an increasing function of s until a critical slip value in the neighborhood of 5%-20%, where μ reaches μ_{max} and then decreases.

The idea of longitudinal slip-based μ_{max} estimation is to use data collected from low-s, low-μ maneuvers—the part of the slip curve near the origin where slip is only a percent or two—to predict the maximum μ of the slip curve. As we will discuss in detail in Section 5.3, the hypothesis that the part of the slip curve near the origin contains information about the maximum coefficient of friction is a contradiction to commonly accepted tire theory.

However, there is a small body of experimental evidence in the literature that indicates that it may be possible to tell something about μ_{max} using information from maneuvers with μ that is away from zero, but still small enough to be classified as a normal driving maneuver. For example, Gustafsson (1997) uses a Kalman filter to estimate the offset and slope of μ vs. slip data obtained for traction with $\mu \leq 0.3$. The slip values were obtained from standard wheel speed sensors, and the μ values were obtained from a static engine map (for the longitudinal forces in the numerator of μ) and a normal force shift model (for the normal forces in the denominator of μ). Extensive testing on various road surfaces shows that this estimated slope, along with other indicators, allows for classification of roads as either "gravel," "high friction," "slippery," or "very slippery." Yi, Hedrick, and

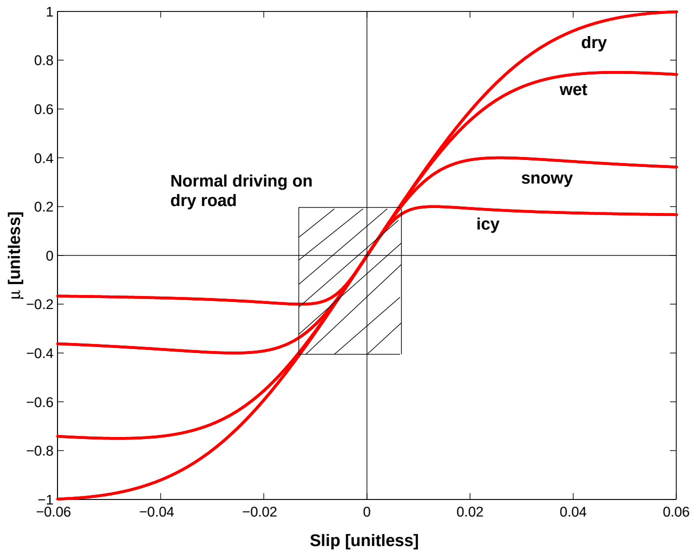

Figure 5.2: Normalized longitudinal force, μ, vs. longitudinal slip, computed using "Magic Formula"

Lee (1999) provide still more experimental evidence that a slip-based μ_{max} estimator could work for normal traction. Linear fits to experimental μ vs. slip data obtained for $\mu \leq 0.2$ show a distinct difference between wet and dry concrete.

Less work has been done on slip-based μ_{max} estimators that work during braking. There are two main reasons for this: The first reason is that the velocity term, v is harder to obtain during braking since all four wheel are slipping. The second reason is that most driving is done with the engine and only occasional use of the brakes; hard braking is particularly rare. But considering braking information for tire-road friction estimation can be advantageous for two reasons. First, a friction estimation system that uses information during acceleration and braking will have greater availability than one that uses only acceleration information. Second, the amount of traction force during a braking maneuver is usually higher than during acceleration, providing more friction demand for a μ_{max} estimator. For example, Ray (1997) takes advantage of moderate excitation during braking and steering to obtain good μ_{max} estimation. Those rare braking maneuvers that use a large percentage of the maximum available friction are extremely useful to a friction estimation system. For example, we use them here to "calibrate" data taken during low friction demand maneuvers.

5.1.3 Structure of this Chapter

The remainder of this chapter is structured as follows: In Section 5.2, we attack the μ_{max} estimation problem by fitting experimentally obtained slip curves with approximations to the nonlinear Magic Formula Tire Model with a fixed longitudinal stiffness. The difficulties we encounter with this nonlinear method lead us to try a linear curve fitting method.

However, this leads us to a fundamental contradiction, which we discuss in detail in Section 5.3. The contradiction is that commonly accepted tire theory like the so-called "brush model," which we present briefly in Section 5.3.1, predicts that the behavior of a tire in the low friction demand part of its slip curve should not be correlated with μ_{max}. However, our own experiments, as well as the results of several other experiments in the literature indicate that there may be a correlation between μ_{max} and the behavior of a tire's slip curve under low friction demand. In Sections 5.3.2 and 5.3.3, we offer two explanations for this correlation and present evidence to support the existence of each. We conclude that a correlation can exist and in the following sections we exploit it to estimate μ_{max} during braking.

In Section 5.4, we design and experimentally verify a slip-based linear μ_{max} estimator that works during braking. This fills in a gap in the literature since most researchers to date have focused on slip-based estimation during driving. A problem, though, is that our estimator—and probably most other slip-based estimators in the literature—is not robust. For example, changes in tire pressure can skew its predictions.

This leads us to Section 5.5, which discusses a self calibration algorithm for slip-based estimators. We show how high friction demand data can be used to calibrate an estimator so that it is robust to tire wear, inflation changes, aging, and other changes.

The last section discusses the advantages and shortcomings of our work—and of the slip-based μ_{max} estimation idea in general. Finally, three appendices discuss issues that are important to our algorithm but were too lengthy to fit into the main text.

5.2 Nonlinear vs. Linear Approaches

Before attempting to identify μ_{max} with a realistic sensor-set, we first explore the μ_{max} identification problem using a non-production sensor that directly measures traction force during braking [44]. In the next section we will eliminate this sensor using an estimator, but using the road force sensor has two benefits. First, we decouple the estimator/observer design from the basic physics of the problem. Second, we generate a set of "truth" results that we will later use to verify our traction force estimator and μ_{max} identification algorithm.

5.2.1 Approximating the real slip curves through nonlinear estimation curves

A very intuitive approach for estimating the maximum friction coefficient is to determine the longitudinal wheel slip and the friction coefficient and use this data to estimate the whole nonlinear characteristic of the slip curve. The maximum of this curve is then the maximum available tire-road friction. We will now demonstrate the basic problem with such an approach.

Consider a tire-road friction model as e.g. proposed in [25]

$$\mu\left(s\right) = \mu'\left(s=0\right)\frac{s}{c_1 s^2 + c_2 s + 1} \tag{5.2}$$

where s is the longitudinal wheel slip, $\mu'\left(s=0\right)$ is the slope of the slip curve for $s = 0$, which is assumed to be constant and to be known, and c_1 and c_2 are free parameters which are estimated such that the slip curve given by (5.2) fits the slip and friction coefficient measurements.

The circles in Figure 5.3 show measured longitudinal wheel slip values and friction coefficients at different times during a braking incident. Slip was derived from measurements using ABS standard wheel speed sensors and traction force was measured with the above mentioned traction force sensor. The normal force was calculated using a normal force observer, which is described later. The solid line is the estimated slip curve calculated with (5.2), where $\mu'\left(s=0\right)$ has been set to 230. The parameters c_1 and c_2 have been estimated using the recursive least squares method.

Figure 5.3 demonstrates very good the basic limitation of an approach where the whole nonlinear slip curve characteristic is estimated. Note that during braking slip and friction coefficient are negative and therefore μ_{max} is

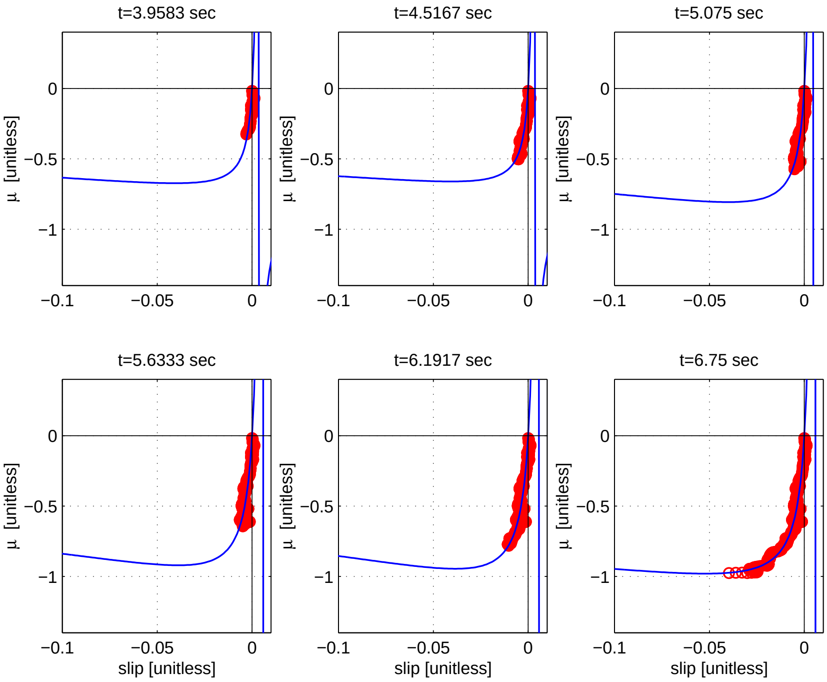

Figure 5.3: Measured (circles) and estimated slip curves (solid line) during braking using the friction model in (5.2). During braking, slip and friction coefficient are negative. It is demonstrated that the minimum of the estimation curves depends on the smallest measured friction coefficient.

the norm of the minimum of the slip curve. The minimum of the estimated curve tends to depend on the smallest measured friction coefficient value. That is approximations in a friction coefficient range near zero would result in an underestimation of the real minimum and estimates of μ_{max} would not be very reliable. Only with relatively high negative friction coefficients the minimum of the estimated curve represents μ_{max} (in this example μ_{max} is about 1). Thus, this method gives only satisfactory estimates if the measured friction coefficients approach the minimum of the slip curve, that is in the braking case if the wheels almost lock. An estimation method for μ_{max}, however, should give reliable estimates far before we reach the minimum (braking) or maximum (driving). Several different tire-road friction models similar to the one given in (5.2) have been investigated and none of them arrived at satisfactory estimations if real measured data is used. We therefore conclude that approximating slip curves through nonlinear estimation curves is not a promising approach.

5.2.2 Linear μ_{max} Identification

The intuitively attractive nonlinear μ_{max} identification approach of the previous section is troublesome because we try to approximate the slip curve by a nonlinear curve with a relative maximum (driving) or minimum (braking), even for a friction coefficient range where the shape of the slip curve is almost linear.

We avoid this over-fitting by using a linear approximation curve instead of a nonlinear curve. However, this introduces the new problem of correlating the slope of the linear fit to the maximum friction coefficient μ_{max}. The majority of the next section is devoted to finding if, and why, such a relationship might exist and then to determining how to exploit it for μ_{max} estimation. Here, we first derive the linear approximation curve and discuss some issues that affect its usefulness for μ_{max} prediction.

Following Gustafsson [20] we write the linear regression equation so that slip s is a linear function of μ,

$$s = \frac{1}{k}\mu + \delta \,. \tag{5.3}$$

(5.3) follows from

$$\mu = k\,(s - \delta)\ , \tag{5.4}$$

where k is the slope and δ is the horizontal offset of the regression line. We determine k and δ using a least squares method. μ and s are measured. For the proposed method we use (5.3) instead of (5.4) for two reasons. First, the former equation allows us to separately estimate the physically different parameters $1/k$ and δ while the latter equation forces us to estimate the physically meaningless product $k\delta$. Thus, the former equation is more useful when the regression is accomplished with a Kalman filter or a recursive least squares method because covariance matrices can be chosen using physical intuition. Second, since the noise in the slip is much larger than the noise in μ, the former equation tends to give less biased slopes of the regression lines.

Intuitively, a set of μ and slip measurements, $(\mu(i), s(i)), i = 1, ..., N$, should tell us more about μ_{max} if they include some points with high values of μ. To express this when we report a least squares estimate of k and δ we use $k_{\mu_{cut}}$ and $\delta_{\mu_{cut}}$ which are the least squares estimates of k and δ gotten from a set of measurements with $\max_i |\mu(i)| = \mu_{cut}$. We refer to μ_{cut}, which can only be between $-\mu_{max}$ and $+\mu_{max}$ as the *friction demand.*

A high friction demand does not guarantee stable estimates of the regression parameters $1/k$ and δ. For example, a cluster of points very near $(\mu = 0.5, s = 0.01)$ could mean that $k_{0.5}$ is infinity and that $\delta_{0.5} = 0.01$ or that $k_{0.5}$ is 50 and that $\delta_{0.5} = 0$, depending on exactly how they are situated. More precisely, the variance of $k_{\mu_{cut}}$ is related to the variance of the μ measurements, so a large variance in μ is needed for low variance estimates of the regression parameters. Borrowing from Gustafsson [20], where this issue is discussed in detail, we refer to the variation in μ needed to get stable parameter estimates as *excitation.*

Using the linear approach in (5.3) we can determine the slope k and the horizontal offset δ for a given road condition. In Figure 5.4 k and μ_{max} of experimental slip curves for different road conditions are plotted. The experiments were conducted on dry and soapy road surfaces. During the experiments the test vehicle was braked until the wheels locked and the slip curves were obtained using standard ABS wheel speed sensors and the above mentioned traction force sensor.

In Figure 5.4 (A) k was determined using only measurements with $\mid \mu \mid \in [0, 0.2]$. We call this k according to the discussion above $k_{0.2}$ In Figure 5.4 (B) k and μ_{max} are plotted for the same slip curves but for a larger μ range, $\mid \mu \mid \in [0, 0.4]$. The $k_{0.2}$-μ_{max} data points seem to be randomly distributed. This changes for a larger μ range. For $\mid \mu \mid \in [0, 0.4]$ smaller slopes of the regression line seem to indicate smaller μ_{max} values. A closer analysis shows that the

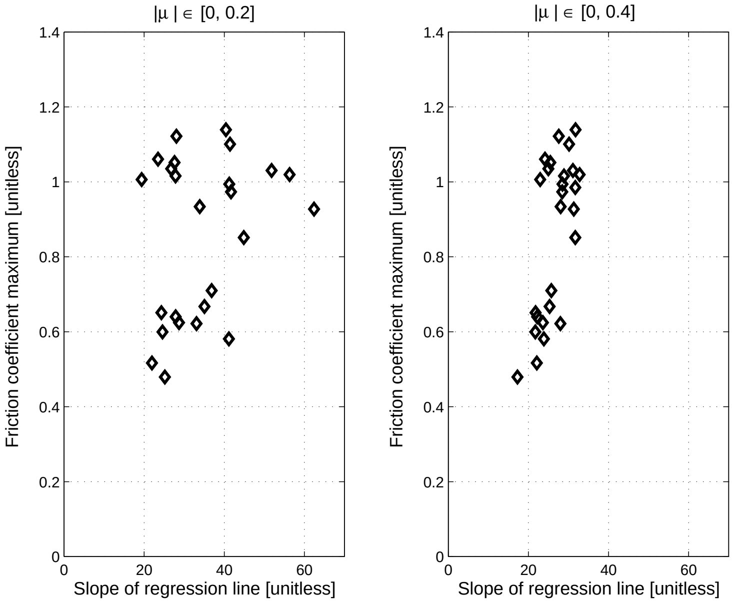

Figure 5.4: μ_{max} and slope k of regression lines for experimentally determined slip curves using ABS wheel speed sensors and a traction force sensor. Experiments were conducted on dry and soapy road surfaces. Two different μ ranges were considered, (A) $\mid \mu \mid \in [0, 0.2]$ and (B) $\mid \mu \mid \in [0, 0.4]$, showing a k-μ_{max} correlation for $\mid \mu \mid \in [0, 0.4]$

random distribution in Figure 5.4 (A) is mainly due to noise in the slip and traction force measurements. If a larger friction coefficient range is considered noise becomes less significant and we observe a correlation between k and μ_{max}. This correlation can be utilized to deduce μ_{max} from the slope of the regression line k without actually reaching μ_{max} by locking the wheel. Although similar results can be found in the literature (cf. Table 5.1 of the next section) the correlation between k and μ_{max} observed in Figure 5.4 (B) seems to be a contradiction to common tire theory. In the next section, we discuss this contradiction in detail.

5.3 A Contradiction

For slip-based methods to work, it is necessary that we can infer the maximum of the slip curve from the behavior of the slip curve at relatively small slips. This, however, contradicts the intuition that the traction force at the maximum of the slip curve –where most of the material particles of the wheel in the contact area are sliding—is dominated by the friction characteristic between wheel and road, while for small slip values—where most of the wheel material particles stick to the road surface—the traction force is determined by the structural stiffness characteristic of the tire carcass. Since these are basically independent properties it should not be expected that the maximum friction can be deduced from the small slip behavior.

In Section 5.3.1, we derive a commonly accepted tire model called the brush model that captures the intuition above. Simulation and theoretical results using this model predict very clearly that there should be no relationship between the behavior in the low friction demand part of a slip curve and μ_{max}. In Section 5.3.2, we make the case that, despite what brush model theory says, there does appear to be correlation. We then propose two explanations for this phenomenon. The first possibility is that the brush model does not accurately predict the behavior of real tires—especially on slippery surfaces. The second possibility is that the brush model is still correct, and that the correlation is an artifact of the nonlinearity of slip curves away from the origin. We conclude that the truth is probably some combination of these explanations. But whatever the reason for the correlation, it appears that we can exploit it to estimate μ_{max}.

5.3.1 Slip curve analysis using a longitudinal brush model

If we neglect the influence of interlayer effects, like e.g. hydroplaning, the so-called "brush-model" helps in analyzing the basic contact behaviour between wheel and road and the resulting slip curves of a given tire/road combination.

The idea behind the brush model is that we represent the tire rubber which is in contact with the road surface as small brush elements with shear stiffness k_b (cf. Figure 5.5). When a brush element enters the contact region it is undeformed and the shear stress in this element is zero. As the brush element travels through the contact area it deflects and the shear stress increases. The tangential deflection of the element, e_x, depends on the relative velocity between wheel and road in the point of contact. Let ω be the wheel angular velocity, r is the wheel radius and v is the wheel hub velocity and define slip as

$$\sigma_x = \frac{v - \omega r}{\omega r} \, . \tag{5.5}$$

Note, that this slip definition is different to the one given in (5.1). We chose this definition here for the derivation of the brush model to be consistent with other papers in the field of tire modelling (see e.g. [32]).

We find for the deflection e_x

$$e_x = (\omega r - v)\, \Delta t = -\xi_B \sigma_x \, , \tag{5.6}$$

where ξ_B is the longitudinal distance of the base point of the brush element on the wheel measured from the leading edge

$$\xi_B = \omega r \Delta t \, .$$

Δt is the time interval where the brush base points move from the leading edge to new positions within the contact area. The deflection e_x increases until the shear force in the brush element reaches the maximum available Coulomb friction force. Let μ_c be the Coulomb friction coefficient and $p\,(\xi_B)$ the normal pressure distribution within the contact area. We then determine the maximum deflection $e_{x_{max}}$ from

$$e_{x_{max}} = \frac{\mu_c p\,(\xi_B = \xi_s)}{k_b} \, , \tag{5.7}$$

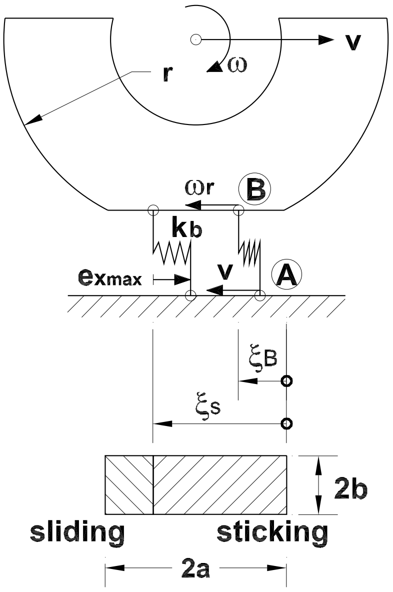

Figure 5.5: Brush model subject to drive slip. Brush elements are modeled as shear springs.

where ξ_s denotes the break-away point, that is the point where the brush starts sliding. We assume a normal pressure distribution which is parabolic in rolling direction and constant in lateral direction

$$p\left(\xi_B\right) = p_0 \left(1 - \frac{a^2 - 2a\xi_B + \xi_B^2}{a^2}\right) , \tag{5.8}$$

where $2a$ is the contact length. p_0 is the maximum of the parabolic distribution

$$p_0 = \frac{3N_\zeta}{8ab} ,$$

where N_ζ is the vertical force acting on the wheel and $2b$ is the width of the contact area. Substituting (5.8) into (5.7) results in

$$e_{x_{max}} = \frac{\xi_s\left(2a - \xi_s\right)}{2a\theta} \tag{5.9}$$

with the parameter

$$\theta = \frac{4}{3}\frac{a^2bk_b}{\mu_c N_\zeta} .$$

With (5.6) $e_{x_{max}}$ becomes

$$e_{x_{max}}\left(\xi_B = \xi_s\right) = \xi_s|\sigma_x| \tag{5.10}$$

and from (5.9) and (5.10) it follows

$$\xi_s = 2a\left(1 - \theta|\sigma_x|\right) . \tag{5.11}$$

Now, we calculate the traction force by integrating the shear stress of each brush element over the length $2a$ and the width $2b$ of the rectangular contact area on the wheel, in which the base points of the brush elements which contact the road surface are located:

$$F_\xi = 2b\left(\int_0^{\xi_s} ke_x d\xi_B - \int_{\xi_s}^{2a} ke_{x_{max}}\frac{\sigma_x}{|\sigma_x|}d\xi_B\right)$$

and we obtain

$$F_\xi = -\mu_c N_\zeta\left(1 - \left(\frac{\xi_s}{2a}\right)^3\right)\frac{\sigma_x}{|\sigma_x|} . \tag{5.12}$$

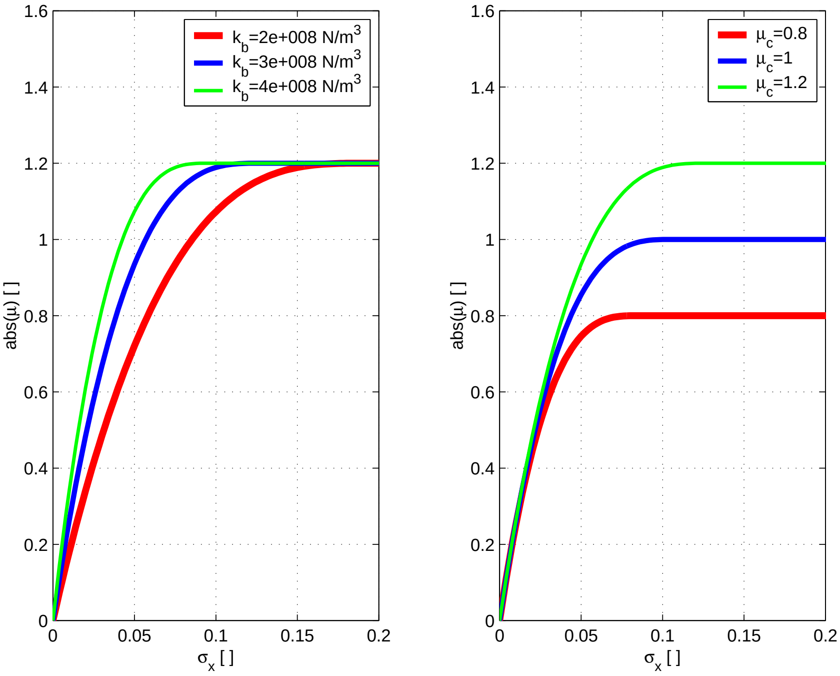

Figure 5.6: Simulated slip curves using the brush model (A) Different longitudinal carcass stiffnesses and (B) Different road surfaces.

To calculate the shear stiffness k_b for a given slip curve we determine the slope of this curve at zero slip

$$k|_{\sigma_x=0} = -\frac{1}{N_\zeta}\frac{\partial F_\xi}{\partial \sigma_x}|_{\sigma_x=0} = -\frac{1}{N_\zeta}\frac{\partial F_\xi}{\partial \xi_s}\frac{\partial \xi_s}{\partial \sigma_x}|_{\sigma_x=0}$$

and we obtain with (5.12) and (5.11)

$$k_b = \frac{N_\zeta}{4a^2b}k|_{\sigma_x=0}\,. \tag{5.13}$$

In Figure 5.6 slip curves are plotted using the tire brush model in (5.12). Figure 5.6 (A) shows slip curves for different longitudinal stiffnesses of the tire carcass and in Figure 5.6 (B) slip curves for different road surfaces are plotted. To simulate different tire carcass stiffnesses and road surfaces we altered the brush shear stiffness, k_b, and the Coulomb friction coefficient, μ_c, respectively.

The plots show that different carcass stiffnesses result in different slopes for the linear part of the slip curve but the maximum remains the same. Different road conditions, on the other hand, change the maximum friction coefficient while the slope of the slip curve at small slip values remains unaffected.

The simulation results obtained by the brush model support the above mentioned intuition that it should be difficult to deduce the maximum available friction coefficient μ_{max} from small slip behavior. The correlation between the slope of the regression line k and μ_{max}, which we observed in Figure 5.4 (B), seems therefore to be a contradiction to our findings using the brush model. In the next two sections, we establish that the results of Figure 5.4 are not isolated, and then we propose two explanations for this correlation.

5.3.2 Friction-dependent Longitudinal Stiffness?

Table 5.1 summarizes evidence in the literature that the low friction demand part of the slip curve may be correlated with μ_{max}. Although the research community will have to produce more evidence of such a correlation before being able to make any final conclusions, we can use the limited data available to us to make some generalizations.

On icy or snowy roads, the slope of the slip curve near zero typically appears to be smaller. It is conceivable that parts of the tire rubber in the

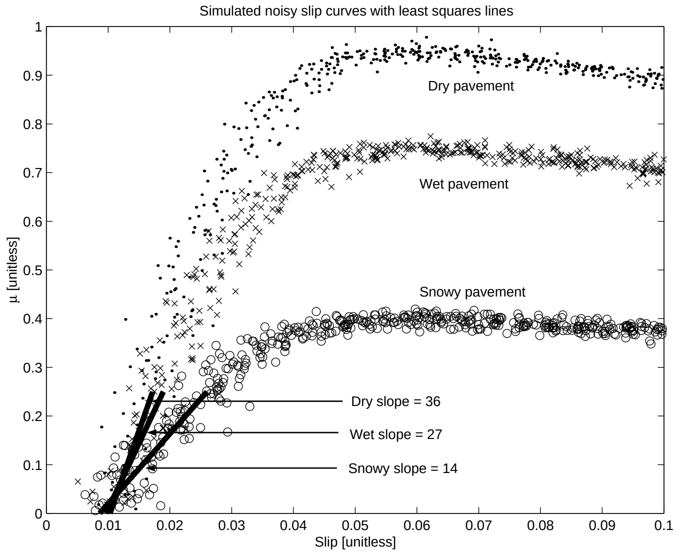

Figure 5.7: *Illustration of the "Secant Effect:"* The three slip curves are calculated from the "Magic Formula" and all have the same longitudinal stiffness ($BCD = 40$). Yet the least squares lines using data for $\mu < 0.25$ have different slopes.

contact area penetrate the ice or snow layer and stick to the road surface while other parts are only supported by the ice or snow layer without touching the road surface. It is also possible that hydrodynamic effects in the micro range, similar to hydroplaning in the macro range, affect the traction between wheel and road surface. In both cases the contact behavior could be no longer only dominated by "Brush Model" effects. Besides of the longitudinal carcass stiffness and Coulomb friction the interaction with the layer between wheel and road could influence the traction characteristic. We call this theory the "friction-dependent longitudinal stiffness." It is a fascinating possibility, but we do not investigate it further in this chapter and instead refer the interested reader to [11] or [12].

5.3.3 The secant effect

Another possible explanation for this apparent correlation is what we call the "secant effect". Figure 5.7 demonstrates the "secant effect" graphically. The three slip curves with $\mu_{max} = 0.4$, 0.75, and 1.0 were calculated using

Table 5.1: Work in the literature indicating that there may be a correlation between slopes of slip curves and μ_{max}

Source	Evidence of a correlation
Breuer, Eichhorn, and Roth, [1]	Graph of slip curves measured on various surfaces shows that those measured on snow and ice have lower initial slopes than those measured on high μ_{max} surfaces like dry and wet pavement. Lack of linear zone noted in text.
Dieckmann, [12]	Very accurate "integral" method of measuring slip produces small difference in slope of slip curve when car driven from wet surface to snowy surface. Constructs a simulation that gives similar results.
Dieckmann, [11]	Same measurement method as above. Amount of slip required to overcome wind and rolling resistance found to be different on icy, wet, and dry roadways.
Gustafsson, [20]	Four different tires tested with many test results for each tire. For all tires, slip curves measured on snowy and icy surfaces tended to have less steep initial slopes than those measured on dry asphalt.
Hwang and Song [23]	Slope of slip curve on dry asphalt road found to be significantly steeper than slope of slip curve measured on an ABS test road with $\mu_{max} \approx 0.3$.
Yi, Hedrick, and Lee [24]	Slope of slip curves near zero slip for wet concrete roads found to be steeper than for dry concrete roads.

the “Magic Formula” and then artificial noise (with statistics based on our experimental noise) was added. For each of the curves, the Magic Formula parameters were chosen so that $BCD = 40$. Thus, each of the slip curves has the same slope at $\mu = 0$. Next, we restricted ourselves to slip data corresponding to values of μ less than 0.25 and calculated the parameters $1/k$ and δ for the regression equation in (5.1) that minimized the cumulative squared estimation error

$$\epsilon_s = \frac{1}{N}\sum_{i=1}^{N}(s(i) - \frac{1}{k}\mu(i) - \delta)^2 \tag{5.14}$$

Here, i indexes the N data pairs $(\mu(i), s(i))$ with $\mid \mu(i) \mid \in [0, 0.25]$. Despite the fact that the longitudinal stiffness BCD is 40 for all three curves, their regression lines have slopes of 14 (for $\mu_{max} = 0.4$), 27 (for $\mu_{max} = 0.75$), and 36 (for $\mu_{max} = 1$). Thus, the slope of the regression line k represents not only the longitudinal tire carcass stiffness, BCD, but also the shape of the whole slip curve. For an increasing friction coefficient range k reflects also the curvature behaviour of the slip curve. The simulated slip curves in Figure 5.7 demonstrate that this effect leads to a correlation between k and μ_{max}, even if we restrict ourselves to a relatively small slip and friction coefficient range.

In the next section we will derive a μ_{max} identifier which utilizes the $k - \mu_{max}$ correlation schown in Figure 5.6 (B). This identifier observes and analyzes slip curves during braking and uses a $k - \mu_{max}$ correlation rule to deduce μ_{max} from k.

5.4 Slip-based Linear μ_{max} Identifier for Braking

Next, we introduce a μ_{max}-estimation method that exploits the correlation we studied in the previous sections. It works during braking and needs only information from standard sensors on a production vehicle. The above mentioned traction force sensor will be used here only for verification purposes. The estimation algorithm is based on a method proposed in [20] which estimates μ_{max} during driving. The distinguishing characteristic of the work presented is that we estimate the maximum friction coefficient μ_{max} during braking and not during accelerating. Furthermore, we will present a calibration technique, which can account for slow changes in vehicle parameters, like for example inflation pressure or tire wear.

Figure 5.8 summarizes the method we use to estimate μ_{max}. The first part of this section concentrates on the block "Slip curve observer" and the blocks below – the measured or estimated values which we need for the slip curve observer. Referring to the diagram, they are: "Slip," which we describe first and where we show how we determine slip; "Normal force," which we describe second and where we present a normal force observer; and "Traction force," which we describe last and which illustrates the algorithm we use to observe the traction force. The second part of this section focuses on the block "Analysis of the slip curve". We will demonstrate that the observed slip curves show a good correspondence with the "real curves," which we measured using the traction force sensor, and we determine the slope of the regression line k for the observed slip curves. The third part, "Inference of μ_{max}", investigates the correlation between the slope of the regression line and the maximum friction coefficient μ_{max}.

5.4.1 Slip curve observer

To obtain a slip curve during driving we need to know the friction coefficient, μ, and the longitudinal wheel slip, s. In the following it is described, how these values are determined using standard sensors on a production vehicle.

Determination of the longitudinal wheel slip

We define the longitudinal wheel slip s according to (5.1).

During the experiments we braked with the front left wheel only. This may cause a moment about the vertical vehicle axis which could result in a slip angle at the braked wheel. We compared slip curves during braking where we braked with the front left wheel and with both front wheels. Under our test conditions the curves were so similar, that we decided to neglect the slip angle.

In this work both wheel angular velocity and wheel hub velocity are measured using standard ABS wheel speed sensors at the front and rear wheels. The measured angular velocity of the left front wheel is used for ω and the measured angular velocity of one of the rear wheels multiplied by the compressed wheel radius is used for v. Note that a production car brakes with all four wheels, and in this case the wheel angular velocity multiplied by the tire radius becomes a poor approximation of the wheel hub velocity. This difficulty can be overcome by using a GPS, a vehicle speed observer (see e.g. [10]) or using the estimated vehicle speed from a production vehicle control system, and use the vehicle speed as the wheel hub velocity.

Determination of the normal force

To determine the friction coefficient μ we need to know the normal force N_ζ. To avoid additional sensors to directly measure this force we use a simplified model of the vertical vehicle dynamics (cf. Figure B.1). The input of this model is the traction force acting on the wheel during braking, which is determined by the traction force observer that we describe in the next section.

The states are the vertical displacement and velocity of the center of gravity of the car body, u_z and $\dot{u}_z$, and the rotation angle and rotational speed of the car body about the lateral axis, φ_y and $\dot{\varphi}_y$, respectively. The normal force, for example at the front, left wheel, is calculated from the states and suspension constants according to (cf. (B.1))

$$N_{\zeta_{11}} = m_{wh} g + c\left(l_f \varphi_y - u_z\right) + d\left(l_f \dot{\varphi}_y - \dot{u}_z\right) . \tag{5.15}$$

The calculation of the states is described in Appendix B.

Determination of the traction force

Using standard automotive sensors, we cannot measure the traction force F_ξ directly. In the following it is shown how F_ξ can be obtained by using an observer algorithm which utilizes only wheel speed measurements. For the observer we assume that the deceleration of the vehicle during braking can be described by the following equation of motion (cf. (B.7) in Appendix B)

$$m\ddot{u}_x = -F_r - F_d + F_\xi \tag{5.16}$$

with $m = (m_c + 4m_{wh})$. m_{wh} is the mass of one wheel, m_c is the car body mass, F_r is the sum of all rolling resistances acting on the wheels and $F_d = c_d v^2$ is the aerodynamic drag force with drag coefficient, c_d, and vehicle speed, v. During the experiments we used differential braking and only the front, left wheel was used for braking. Then, (5.16) becomes

$$\left(m + \frac{J_{12}}{r_{12}^2} + \frac{J_{21}}{r_{21}^2} + \frac{J_{22}}{r_{22}^2}\right)\ddot{u}_x = -F_r - F_d + F_{\xi_{11}} , \tag{5.17}$$

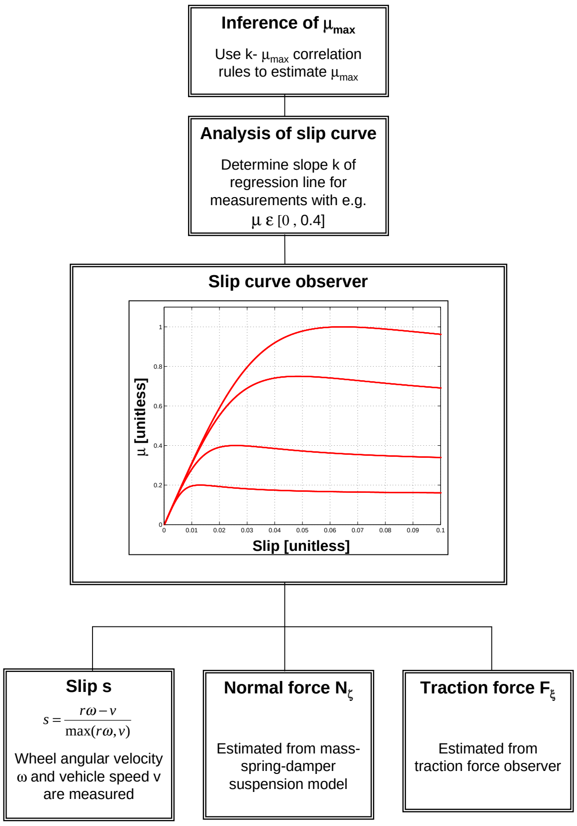
Inference of μ_{max}
Use k- μ_{max} correlation rules to estimate μ_{max}
Analysis of slip curve
Determine slope k of regression line for measurements with e.g. $\mu \varepsilon [0, 0.4]$
Slip curve observer
μ [unitless]
Slip [unitless]
0
0.2
0.4
0.6
0.8
1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Slip s
$s = \frac{r\omega - v}{\max(r\omega, v)}$
Wheel angular velocity ω and vehicle speed v are measured
Normal force N_ζ
Estimated from mass-spring-damper suspension model
Traction force F_ξ
Estimated from traction force observer

Figure 5.8: μ Estimation strategy used in this chapter.

where J_{ij} and r_{ij} are the moments of inertia and radii of the front, right and both rear wheels and $F_{\xi_{11}}$ is the traction force acting on the front, left wheel.

We experimentally determined the rolling resistance, we assumed a drag coefficient and we determined vehicle speed by measuring the wheel speed of one of the free rolling wheels). For the calculation of the traction force using (5.16) it then remains to determine the acceleration of the vehicle $\ddot{u}_x$. One way to obtain the acceleration is to numerically differentiate the angular velocity of one of the free rolling wheels. However, for a production vehicle all four wheels are braked. Another option is to measure the acceleration with an accelerometer. To avoid this sensor and to provide a method which works on production vehicles we propose to utilize the differentiated and filtered wheel speed of the braked wheel. Although $\dot{\omega} r$ of a wheel during braking is not necessarily equal to the deceleration of the vehicle, in particular not when the wheel is locking, the resulting estimate of the traction force using the angular velocity of the braked wheel was satisfactory for the analysis of our experiments (see discussion in Appendix C).

There are many different ways to differentiate and filter the measured wheel angular velocity. A method which worked well for our purposes is described in the following. Consider the low-pass filter equation for the wheel angular velocity in the time domain

$$\dot{\omega}_{filt} = \frac{\omega - \omega_{filt}}{\tau_\omega} \, , \tag{5.18}$$

where $\dot{\omega} \approx \dot{\omega}_{filt}$ if the filter is fast. We determine ω_{filt} by calculating

$$\omega_{filt}\left(k+1\right) = \omega_{filt}\left(k\right) + \frac{\Delta t}{\tau_\omega}\left(\omega\left(k\right) - \omega_{filt}\left(k\right)\right) \, , \tag{5.19}$$

where Δt is the sample time and k is the sample number. The traction force is calculated by rearranging (5.16) (or (5.17) in the case that only the front, left wheel is used for braking). Hence,

$$F_\xi\left(k\right) = m\ddot{u}_x\left(k\right) + F_r + F_d\left(k\right) \, , \tag{5.20}$$

with

$$\ddot{u}_x\left(k\right) = r\left(k\right)\frac{\omega\left(k\right) - \omega_{filt}\left(k\right)}{\tau_\omega} \, , \tag{5.21}$$

where τ_ω is sufficiently small.

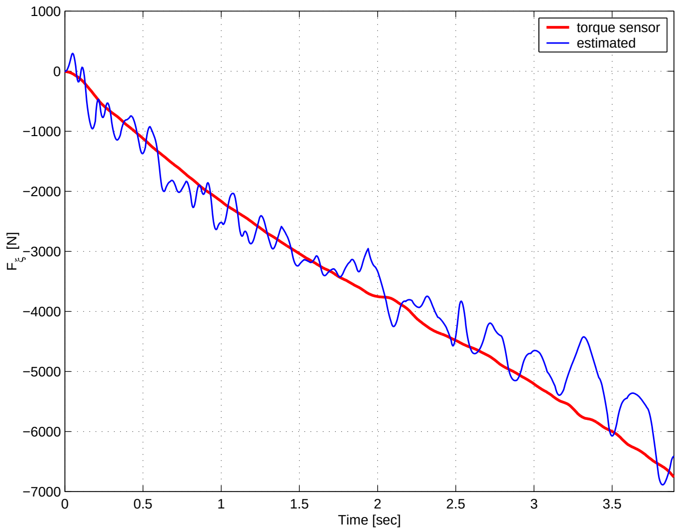

Figure 5.9: Comparison between measured and estimated traction force.

To investigate the quality of the observed traction force in (5.20) we compare it with measurements. We measure the traction force by using the above mentioned traction force observer. In Figure 5.9 we see the comparison between the measured and estimated traction force at the front, left wheel during a braking maneuver on dry road where only the front, left wheel was used for braking. We see a satisfactory match of the measured and the observed curve.

5.4.2 Analysis of the observed slip curves

In the last section it has been shown how slip curves can be observed during braking using only production vehicle sensors. These curves are now analyzed. We will show that they match well with the "real" slip curves and we determine the slope of the regression line, k, which we need to deduce μ_{max}.

The following slip curves have been determined during single braking incidents. We calculate slip according to (5.1) and filter this signal with a low-pass filter. The friction coefficient is defined as traction force over normal force. We obtain the traction force by using the traction force observer described above. The normal force is calculated using (5.15). To create different road conditions we constructed an apparatus which sprays a water-

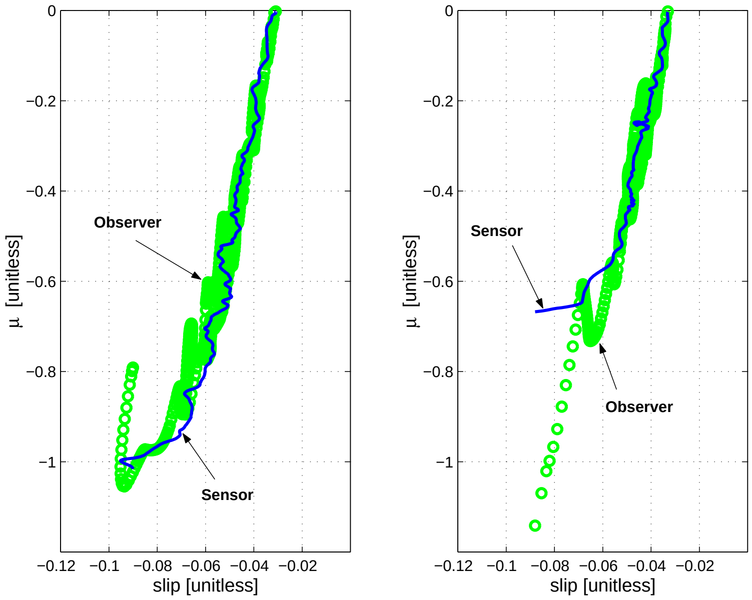

Figure 5.10: Slip curves during braking. (A) dry road surface (measured=solid, observed=circles). (B) Soapy road surface (measured=solid, observed=circles)

soap mixture in front of the braked wheels. Figure 5.10 shows slip curves for a dry and a soapy road surface. For the dotted curve the traction force has been estimated using the traction force observer, the solid curves have been determined by using traction force sensor measurements.

Figure 5.10 shows that the slip curves obtained by using the proposed traction force observer are very similar to the "real" slip curves. The quality of the observed slip curves decrease for higher slip and higher friction coefficient values. This, however, does not have much effect on the slope of the regression line, since we determine the slope at slip values where the friction coefficient is relatively small.

We use the linear regression model in (5.3) and determine the reciprocal of the slope of the regression line $1/k$ and the offset on the slip axis δ using recursive least squares.

In Figure 5.11 k and δ of the regression line for the observed slip curves from Figure 5.10 are plotted against the observed friction coefficient. The estimation algorithm started at the beginning of the braking maneuver and stopped when the braked wheels locked.

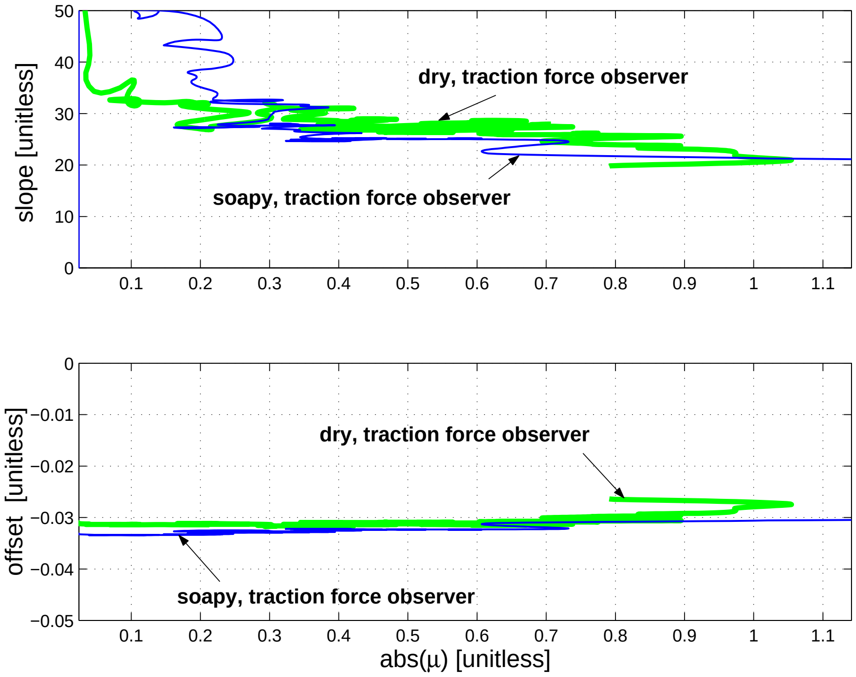

Figure 5.11: Estimated slope and offset vs. friction coefficient for the observed slip curves from figure 5.10.

In Figure 5.11 we see fluctuation in k at small friction coefficient values μ due to the fact that the noise in μ is large compared to the mean μ value. After this transient, k smoothes out and then begins decreasing as μ approaches the peak of the slip curve.

The graphs show that for friction coefficients between 0.4 and 0.8 the slope of the regression line is smaller for the soapy and higher for the dry road surface. The offset δ in Figure 5.11 is also different for the two different road conditions. However, δ mainly depends on our slip definition (cf. Appendix A) and cannot be used for the identification.

In the following, we use the difference in k to estimate μ_{max}.

5.4.3 Inferring μ_{max}

In order to infer the maximum available friction from the slope of the regression line of the observed slip curve we now investigate the k-μ_{max} correlation on dry and soapy road surfaces.

For the investigation we measured and observed slip curves for different road conditions. We use the measured curves to determine the "real" μ_{max} value and calculate the slope of the regression line for the observed curves.

The outcome of this analysis is plotted in Figure 5.12 where we determined the slope of the regression line for each observed slip curve employing only data with $\mid \mu \mid \in [0, 0.4]$. According to the discussion above we call this slope $k_{0.4}$. The circles are the $k_{0.4}$-μ_{max} data points. The vertical thick line at $k_{0.4} = 29.5$ indicates the mean $k_{0.4}$ of all slip curves on dry road. With respect to the notation in the next section we call this slope k^*. We determined k^* by calculating $k_{0.4}$ for a slip curve which included slip and friction coefficent data from all of our experiments on dry road.

Based on these results we conclude the following (neglecting two data points which deviate the most):

- On dry road $k_{0.4}$ varies in the interval $[22.9, 32.7]$ and the maximum friction coefficient range is $[0.925, 1.14]$.

- On soapy road $k_{0.4}$ is in the interval $[17.3, 25.65]$ and the maximum friction range is$[0.48, 0.71]$.

- For $\mu \in [0, 0.4]$ the mean slope of all dry slip curves, k^*, is 29.5.

- The maximum $k_{0.4}$ on soapy road is 87% of k^*.

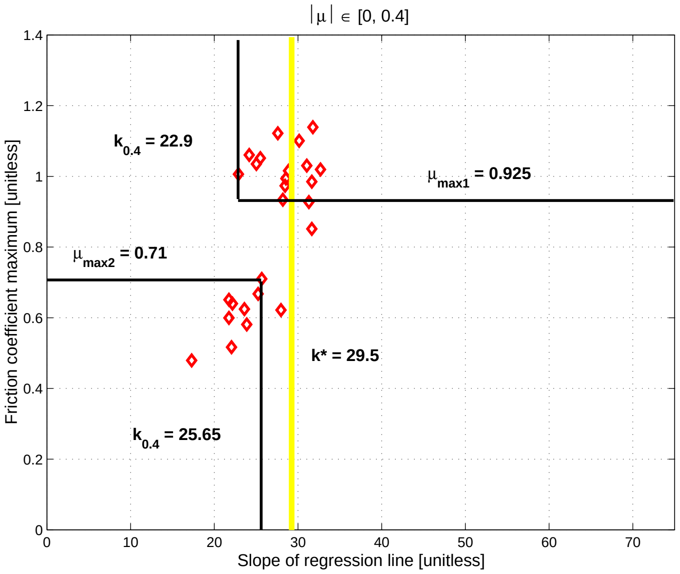

Figure 5.12: Maximum friction coefficient against slope of regression line for $\mu \in [0, 0.4]$.

- The minimum $k_{0.4}$ on dry road is 78% of k^*.

From these observations we can derive different rules for the correlation between $k_{0.4}$, which we determine from the observed slip curve, and the "real" μ_{max} value, which is usually unknown. One possible rule is the following: Let's assume k^* is known and we have determined $k_{0.4}$ during a braking event. Then, according to our experimental results the following holds:

$$\begin{aligned} k_{0.4} > 87\%\, k^* \quad &\Rightarrow \quad \mu_{max} > 0.925 \\ k_{0.4} < 78\%\, k^* \quad &\Rightarrow \quad \mu_{max} < 0.71\,. \end{aligned}$$

However, this rule or any other rule relating the slope of the regression line with the maximum friction coefficient is not universally applicable. It can be different for a different car and might even differ for the same car when certain vehicle parameters change. In the next section we will discuss this in more detail and we will present a self-calibration method which adapts the above given $k_{0.4} - \mu_{max}$ correlation rule to changes in vehicle parameters, like for example tire inflation pressure or tire wear.

5.5 Robustness and Self-calibration

The relationship between $k_{0.4}$ and μ_{max} of Figure 5.4 in the previous section is not universally applicable, and it is not robust. In this section, we first explain why this relationship is so precarious, and then we develop a "self-calibration" algorithm to correct the problem. To develop this "self-calibrator," we first express our friction estimation rule in terms of a regression line slope k^* that is known to correspond to a high friction road. This works well at first, but as parameters change, the original value of k^* becomes obsolete and the friction estimation rule that depends on it gives incorrect results. Thus, we need to develop a way to continuously update k^* corresponding to a high friction road. To do this, we use the medium-to-high friction demand braking maneuvers that normally occur during driving to "select" data for a special calibration set, and then we use the calibration set to calculate an updated value of k^*.

5.5.1 Precarious Relationship

Before developing a self-calibration algorithm, though, let us demonstrate what we mean when we say that $k - \mu_{max}$ relationships are precarious.

Figure 5.13 shows k ranges for dry pavement from experiments using seven different tire/vehicle combinations. (Data was compiled from papers [20], [24], [23], and [31] and came exclusively from experiments with passenger vehicles, as opposed to tire testing apparatus.) k corresponding to a dry road with $\mu_{max} \approx 1.0$ can be expected to range between approximately 15 and 100, depending on the vehicle/tire combination and test conditions.

Most of this variation is probably due to tire types. For example, measurements on tire testers indicate that winter tires with deep treads have significantly less steep slip curves than summer tires. Similarly, worn tires typically have steeper slip curves than unworn tires, owing to their less pronounced tread [2], [3].

The vehicle type—front wheel drive or rear wheel drive—may also be important because it determines which wheels are used to measure velocity and which wheels are used to measure slip (in the driving case). Normal force shifts during acceleration and deceleration subtly affect the radii of the front and rear tires in different ways, possibly altering the measured slip slope. (Appendix A discusses this difficulty in detail). Regardless of the reason for the slope variation, Figure 5.13 shows that it is not reasonable to expect

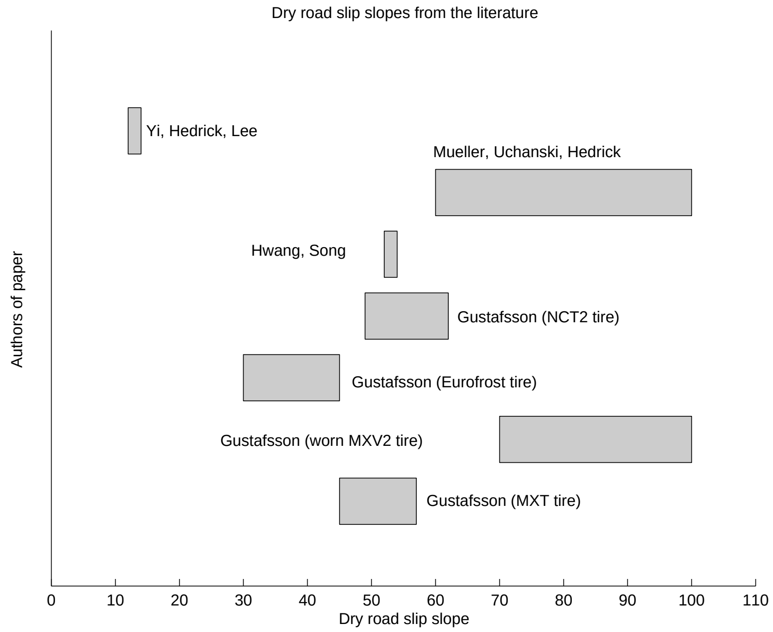

Figure 5.13: Approximate range of dry road slip slopes from the literature.

that a $k - \mu_{max}$ relationship that holds for one car/tire combination will hold equally well for a different car/tire combination.

In fact, even considering a single car, it it unlikely that a $k - \mu_{max}$ relationship that works this month will work next month or next year. Above, for example, we saw that tire wear can affect slip slope. In addition, researchers have noticed that inflation pressure can change the slope of a slip curve significantly. Gustafsson [20] noted a 20% decrease in the slope when the tire pressure of the slipping wheel was decreased by 0.5 bar (a realistic decrease between fillings), and others have noticed similar sensitivity [17]. Very large swings in the ambient temperature may also be important, both because temperature affects tire pressure and because temperature affects the flexibility of rubber.

5.5.2 Relative Thresholds

Despite all of these uncertainties, numerous studies indicate that relative changes in the slope of the linear part of the slip curve are useful to indicate relative changes in friction. For example, our results in Figure 5.12 show a drop in $k_{0.4}$ from approximately 30 to approximately 25, corresponding to a drop in μ_{max} from ≈ 1 to ≈ 0.6 when the road is wet. Hwang and Song [23] report a slope drop from ≈ 53 to ≈ 19 when μ_{max} drops from ≈ 1 to ≈ 0.3, and Gustafsson [20] and Dieckmann [11],[12] both report qualitatively similar results.

As a result, it has been proposed to express thresholds for road classification as a function of the slope of the linear part of the slip curve known to come from a high friction road, which we denote k^* here. The left graph of Figure 5.14 demonstrates this idea with our experimental data. μ vs. slip data from different braking maneuvers are conglomerated together, and $k_{0.4}$ is calculated for the ensemble, producing what we call the "factory calibration" curve for this tire (since the only setting where we would realistically know ahead of time that the road surface is high friction is a testing facility of some sort). In Figure 5.14, we then draw a line at a slip slope of, say, $0.87 \cdot k^*_{factory}$ and give the classification rule that if $k_{0.4} > 0.87 \cdot k^*_{factory}$, then it is reasonably certain that the surface has $\mu_{max} > 0.925$. The rule is misclassifies 2 points, but given that the distinction between wet and dry roads is a relatively subtle one from a contact mechanics perspective (as opposed to the difference between, say, dry pavement and snow), it performs reasonably well.

5.5.3 Adapting k^*

Expressing classification thresholds in terms of $k^*_{factory}$ works well until something happens to change the underlying high-friction slip curve. The center panel of Figure 5.14 shows this problem. Here, the μ vs. slip data from the experiments of the left panel was altered so that the underlying high friction slope of the linear part of the slip curve decreased from ≈ 33 to 15. This could be due to, say, a decrease in tire pressure. The "high friction" slope threshold at 25.65 that worked in the left panel now misclassifies all of the data points.

Thus, our estimate of the underlying high-friction k^* needs to be adaptive. The trouble with adapting it, though, is that we must first be sure that the data being used for this "self-calibration" comes from high-friction roads. For example, if the vehicle has been driving on snowy roads for the past several hours, we would not want to use this μ vs. slip data to re-calculate k^*. Doing so would gradually result in the snowy road being classified as a high friction surface.

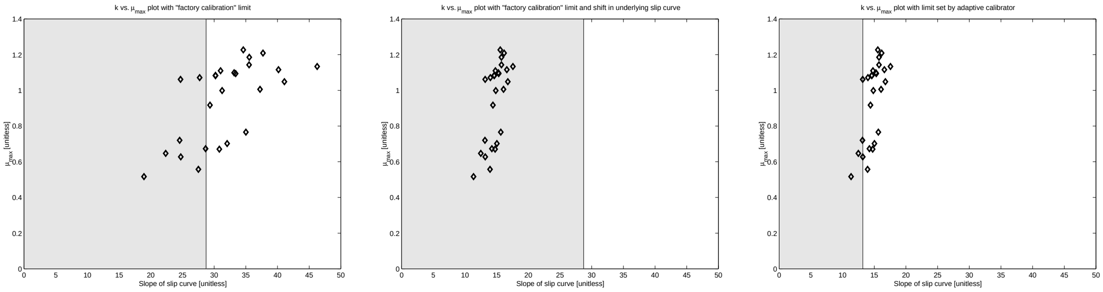

Figure 5.14: *Left:*μ_{max} vs. $k_{0.4}$ and "high friction" line given by $0.9 \cdot k^*_{factory}$. *Center:* Same as left, but the underlying physical slip slope drops by a factor of two, resulting in misclassification. *Right:*Same as center, but the "high friction" line is now given by $0.9 \cdot k^*_{adaptive}$, correcting misclassification problem.

One way to resolve this problem is to require data to somehow "prove" that it came from a high friction surface before using it to update k^*. A straightforward way of doing this is to wait for situations where the tires attain moderate-to-large μ and then to use a data window surrounding this event to update k^*.

Figure 5.15 and the right panel of Figure 5.14 show the results of this strategy when the underlying slope of the linear part of the slip curve gradually decreases. We maintained a first-in-first-out buffer μ vs. slip data for the 10 most recent maneuvers that were "proven" to come from a dry road. To find a new value of k^*, we calculated $k_{0.4}$ for the ensemble of the μ vs. slip data. To be admitted to the first-in-first-out buffer, a maneuver had to have a friction demand greater than 0.6. Like the center panel of Figure 5.14, these figures show hybrid experimental/simulation results. To simulate changes in the underlying slip curves over a long period of time, experimental data files from approximately 20 braking maneuvers on dry pavement and 10 brakingmaneuvers on lubricated pavement were put in a pool, and for 200iterations a random maneuver was selected. When a maneuver was selected, a new underlying slope of the linear part of the slip curve was superimposed on the experimental μ vs. slip data to generate a slip curve with the same noise and bias problems as the original, but a different underlying slope. For the first 100 maneuvers, the underlying slope decreased gradually from 33 to 15, and then for the last 100 maneuvers it remained fixed at 15. We see in the right panel of Figure 5.14 that the new

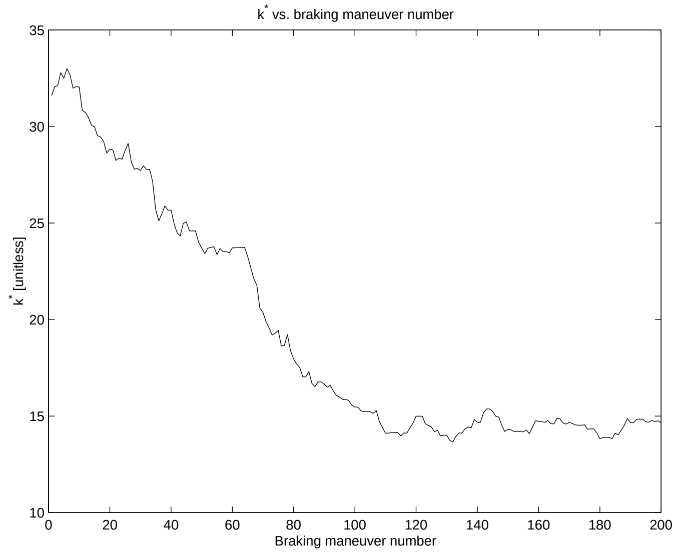

Figure 5.15: $k^*_{adaptive}$—the estimate of the underlying high friction k^* plotted vs. braking maneuver. The true underlying slope of the linear part of the slip slope falls from 32 to 15 between the first and the 100th maneuver.

classification line $0.87 \cdot k^*$ is viable, although it tends towards classifying roads as high friction. Figure 5.15 shows that the estimate of k^* gotten from the buffer is faithful to the underlying slope of the linear part of the slip curve. The one-to-two maneuver flat spots in the graph happen when maneuvers from lubricated pavement are excluded from the buffer (because they do not attain a high enough μ).

A potential problem with this type of self-calibration procedure is that high friction demand maneuvers may not happen very frequently. Although we have not conducted friction demand studies for normal drivers, we were able to find evidence in the literature that drivers demand medium to high amounts of friction quite regularly. For example, in [2] the longitudinal and lateral accelerations demanded by "normal" and "sporty" drivers during an 80 km test drive are plotted. On several occasions the normal driver demands accelerations greater than 2m/s in traction and 4m/s in braking, corresponding approximately to friction demands of $\mu = 0.4$ and $\mu = 0.5$ respectively.

5.6 Conclusions and Recommendations

This work shows that there is potential for using slip-based techniques during moderate excitation braking maneuvers to estimate μ_{max}. Using wheel speed and brake pressure data from single braking maneuvers with $\mu \leq 0.4$, we were able to distinguish between dry and soapy pavement. The braking approach to slip-based μ_{max} estimation is complementary to approaches using traction data. A vehicle spends more time driving than braking, making the traction approach valuable. However, vehicles tend to reach higher μ values during braking, making the braking approach valuable.

It appears that one of the most valuable uses of braking data for slip-based road condition estimation might be for calibrating the rather precarious relationship between slip slope and μ_{max}.

Chapter 6

Conclusion

Research conducted under MOU 388 produced four major contributions that will be of use to the highway safety and AHS communities:

1. In Chapter 2, a protocol for highly mobile networks was designed and shown to work in simulation.

2. In Chapter 3, a simulation study to investigate the potential benefits of a "Cooperative Adaptive Cruise Control" was executed. It was found that communicated warnings can reduce the accelerations demanded by adaptive cruise control systems during cut-in and severe braking maneuvers.

3. A concept that we dub "Cooperative Estimation" was introduced in Chapter 4. Using an ad-hoc communication network, information was shared between vehicles, and it was shown through simulation studies that this kind of cooperation produced estimates of travel time and road condition that were significantly more useful than the estimates that any individual vehicle could provide by itself. Cooperative estimation is a useful concept, but whether it evolves further in the future will depend on whether the same types of estimates can be gotten using simpler technology.

4. In Chapter 5, so-called "slip-based" road friction estimators were investigated through extensive experimentation, simulation, and literature review. It was shown that this type of estimator can produce road friction estimates without using any dedicated road condition

sensors. We expect that this area will be an exciting research and development direction in the next few years with potential applications in the near future, and we were happy to have contributed to this nascent technology.

Appendix A

On-Vehicle Slip Measurements

As we mentioned in the text of Chapter 5, measuring slip for friction estimation presents several difficulties. First, we must choose a definition of slip, and second, we must find a way to calculate slip according to this definition with the available measurements. We treat the definition problem first and then discuss the calculation problem. We find that the choice of slip definition has only a minor effect on results, while the method of calculation may have a significant effect.

A.1 Slip definition

For this work, we used the following definition for longitudinal slip s:

$$s := \frac{r\omega - v}{\max(r\omega, v)}$$

where v means the component of the velocity of the wheel along the longitudinal axis of the tire; ω is the spin velocity of the wheel measured at the wheel center; r is the effective rolling radius of the tire and is defined as $r = v/\omega_0$, where ω_0 is the spin velocity that the wheel would have if it were rolling freely, but under identical conditions to those under which ω is measured.

In the literature, the word "slip" takes on different definitions. The idea of all of the definitions, however, is the same: slip is the relative difference of the slipping wheel's circumferential velocity and its translational velocity. The differences arise from different choices of sign in the numerator and from different choices of the normalizing velocity in the denominator.

The different choices of sign for the numerator seem to cause only minor confusion. We chose the numerator in our slip definition to be $r\omega - v$ so that traction gives positive slip and braking gives negative slip.

On the other hand, the choice of normalizing velocity for the denominator does occasionally cause confusion. There are three obvious candidates, each with its own advantages:

1. $r\omega$, the circumferential velocity.
2. v, the translational velocity.
3. Some combination of the two, usually $r\omega$ for traction and v for braking.

The circumferential velocity $r\omega$ is a good choice because it appears directly in derivations for the brush model (See, for example, the brush model section above, or the derivation in [32]). The resulting slip quantity, which we denote as $\sigma_x = (r\omega - v)/(r\omega)$, has been called the "theoretical slip" [32],[4]. As long as no "brush" in the contact patch is "saturated," the brush theory predicts that the friction force will be a linear function of σ_x. This relation is true in both traction and braking, making σ_x appealing if we wish to relate slopes of slip curves obtained during traction to those during braking.

However, normalizing by $r\omega$ poses a problem during heavy traction. When the wheel's circumferential velocity is 50% of the wheel's translational velocity, σ_x has a value of -1. If the wheel slows further, σ_x becomes even more negative and approaches $-\infty$ as the wheel velocity approaches zero. Thus, σ_x takes values from $-\infty$ for a locked wheel during braking to $+1$ for a spinning wheel in traction.

From a practical perspective wheel locking during braking is much more common than wheel spinning during traction, so many slip definitions have v in the denominator. The resulting slip quantity, which we denote as $\kappa = \frac{r\omega - v}{v}$, takes values from -1 for a locked braking wheel to $+\infty$ for a spinning but non-translating wheel. It is this more "practical" slip that goes into the empirical "Magic Formula" of [4].

σ_x is related to κ by $\sigma_x = \kappa/(1+\kappa)$. This has the consequence that the longitudinal force is not a linear function of the practical slip κ—even in the idealized case where no brushes "saturate." Figure A.1 shows the effect of this nonlinearity. The vertical axis is μ and the horizontal axis is interpreted as the theoretical slip, σ_x or the practical slip, κ, depending on the curve. The solid curve is the linear μ vs. σ_x curve that one obtains from

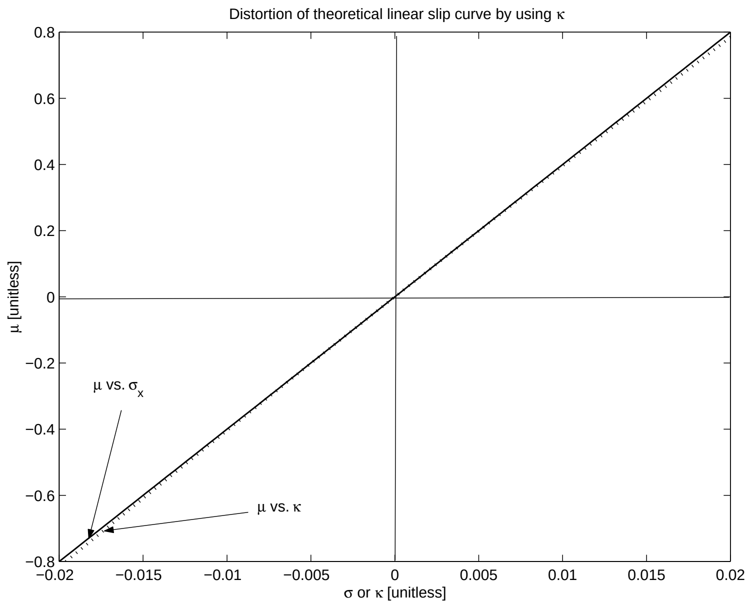

Figure A.1: Distortion of linear μ vs. σ_x relationship by instead using κ.

the brush model if no brushes saturate: $\mu = k\sigma_x$, where k is the longitudinal stiffness. The dotted curve has the equation $\mu = k\frac{\kappa}{1+\kappa}$ and is what we would obtain if we measured the force vs. practical slip relationship for a tire whose underlying physics obeyed the brush model.

We see from Figure A.1 that the choice of slip normalization factor—either $r\omega$ or v—has only a very small effect as long as slip is less than a few percent. So in the regions where slip-based friction estimators operate, the normalizing velocity has little effect on results, especially in comparison to the effects of noise and tire radius changes. Consequently, we chose our normalizing velocity based on higher slip behavior.

When slip is larger than a few percent, using v as a normalizing velocity distorts the theoretical brush model slip curves significantly. But at these higher slips, even the undistorted brush model curves tend to diverge from experimental results so much that tire modelers often use semi-empirical corrections—or they drop analytical models in favor of empirical ones. In this case, there is the temptation to choose a slip definition based on numerical convenience rather than on theoretical significance.

That in fact is what we do when we use equation A.1 for slip. The max-function in the denominator forces this negative velocity difference to be normalized by v, resulting in $s = -1$ if the braking wheel locks. On the other hand, an accelerating wheel has a positive numerator, and the denominator becomes $r\omega$ so that $s = +1$ if the vehicle stands still while the wheels spin. Thus, normalizing by $\max(r\omega, v)$ gives a numerically convenient slip value between -1 and $+1$ and has a negligible effect on the shape of the braking slip curve at the slip values of interest for our purposes.

Appendix B

Vehicle Equations of Motion

In the following we derive the equations of motion for the vehicle during braking.

Step 1: Separate masses from elastic foundation and constraints (cf. Figure B.1).

Step 2: Apply Newton's 2. law to the car body, where the point CG is the center of gravity of the car body. For sake of simplicity we assume that the drag force F_d acts on the center of gravity of the car body. The height h is the distance from CG to the ground. r is the tire radius.

$$\begin{aligned} m_c\ddot{u}_x &= -F_d - F_{cx_f} - F_{cx_r} \\ m_c\ddot{u}_z &= -m_c g - N_{cz_f} - N_{cz_r} \\ J_c\ddot{\varphi}_y &= -N_{cz_r}l_r + N_{cz_f}l_f + T_{B_f} + T_{B_r} + \left(F_{cx_f} + F_{cx_r}\right)(h-r) \end{aligned}$$

Newton's 2. law for the front and rear wheels (front: f, rear: r) gives

$$\begin{aligned} 2m_{wh}\ddot{u}_{x_{wh_{f/r}}} &= F_{cx_{f/r}} - F_{r_{f/r}} + F_{\xi_{f/r}} \\ 2m_{wh}\ddot{u}_{z_{wh_{f/r}}} &= -2m_{wh}g + N_{cz_{f/r}} + N_{\zeta_{f/r}} \\ 2J\dot{\omega}_{f/r} &= -T_{B_{f/r}} - F_{\xi_{f/r}}r \end{aligned}$$

m_{wh} and J are the mass and moment of inertia of the wheel, respectively.

Step 3: Consider kinematical constraints

1. $u_{z_{wh_{f/r}}} \equiv 0 \qquad \Rightarrow \qquad \ddot{u}_{z_{wh_{f/r}}} = 0$

2. $u_{x_{wh_{f/r}}} \equiv u_x - (h-r)\,\varphi_y \qquad \Rightarrow \qquad \ddot{u}_{x_{wh_{f/r}}} = \ddot{u}_x - (h-r)\,\ddot{\varphi}_y$

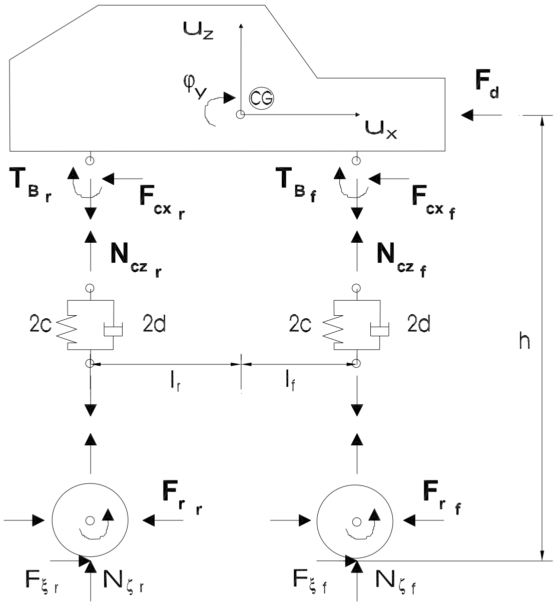

Figure B.1: Vehicle model.

From the first constraint equation follows

$$N_{\zeta_{f/r}} = 2m_{wh}g - N_{cz_{f/r}} \tag{B.1}$$

and the second results in

$$F_{cx_{f/r}} = -F_{\xi_{f/r}} + F_{r_{f/r}} + 2m_{wh}\left(\ddot{u}_x - (h-r)\,\ddot{\varphi}_y\right) \,.$$

We obtain for the remaining degrees of freedom

$$\begin{aligned}
m\ddot{u}_x &= -F_d - F_r + F_\xi + 4m_{wh}\,(h-r)\,\ddot{\varphi}_y && \text{(B.2)}\\
m_c\ddot{u}_z &= -m_c g - N_{cz_f} - N_{cz_r} &&\\
J_v\ddot{\varphi}_y &= -N_{cz_r}l_r + N_{cz_f}l_f + T_{B_f} + T_{B_r} + (h-r)\,(-F_\xi + F_r) &&\\
&\quad +4m_{wh}\,(h-r)\,\ddot{u}_x && \text{(B.3)}\\
2J\dot{\omega}_{f/r} &= -T_{B_{f/r}} - F_{\xi_{f/r}} && \text{(B.4)}
\end{aligned}$$

with $m = m_c + 4m_{wh}$, $J_v = J_c + 4m_{wh}\,(h-r)^2$, $F_r = F_{r_f} + F_{r_r}$ and $F_\xi = F_{\xi_f} + F_{\xi_r}$. Note the wheel inertia coupling term in (B.2) and (B.3) which results from the fact, that u_x and φ_y are displacements of the center of gravity of the car body and not of the whole vehicle. For sake of simplicity we neglect this coupling term. Furthermore, we assume that the brake torques $T_{B_{f/r}}$ can be approximated by the traction forces times the wheel radius. To understand the error introduced by this approximation, consider the moment balance for one braked wheel i

$$J\dot{\omega}_i = -T_{B_i} - F_{\xi_i} r \, . \tag{B.5}$$

For a longitudinal acceleration μg the traction force F_ξ for one wheel is approximately one fourth of the total longitudinal force, so $|F_\xi| \approx |\mu g m/4|$. If the wheel does not lock, then $r\dot{\omega}_i$ is a reasonable approximation of the longitudinal acceleration (cf. Appendix C), so $|\dot{\omega}_i| \approx |\mu g/r|$ and $|J\dot{\omega}_i| \approx |J\mu g/r|$. Therefore, the percent error in the traction force estimate introduced by neglecting the wheel inertia term is $|J\dot{\omega}_i/(F_{\xi_i} r)| \cdot 100\% \approx 4J/(mr^2) \cdot 100\%$, which for our test vehicle amounts to 3.7%. We therefore neglect the wheel inertia terms and determine the brake torques by

$$T_{B_{f/r}} \approx -F_{\xi_{f/r}} r \, . \tag{B.6}$$

The equations of motion become

$$\begin{aligned}
m\ddot{u}_x &= -F_d - F_r + F_\xi \qquad\qquad \text{(B.7)}\\
m_c\ddot{u}_z &= -m_c g - N_{cz_f} - N_{cz_r}\\
J_v\ddot{\varphi}_y &= -N_{cz_r} l_r + N_{cz_f} l_f + (h-r)\, F_r - F_\xi h\\
J_{11}\dot{\omega}_{11} &= -T_{B_{11}} - F_{\xi_{11}}\\
J_{12}\dot{\omega}_{12} &= -T_{B_{12}} - F_{\xi_{12}}\\
J_{21}\dot{\omega}_{21} &= -T_{B_{21}} - F_{\xi_{21}}\\
J_{22}\dot{\omega}_{22} &= -T_{B_{22}} - F_{\xi_{22}}
\end{aligned}$$

where, for example, 11 is the front, left and 12 the front, right wheel.

Step 4: Determine the vertical spring and damper forces

$$\begin{aligned}
N_{cz_f} &= 2\left(c\left(-l_f\varphi_y + u_z\right) + d\left(-l_f\dot{\varphi}_y + \dot{u}_z\right)\right)\\
N_{cz_r} &= 2\left(c\left(l_r\varphi_y + u_z\right) + d\left(l_r\dot{\varphi}_y + \dot{u}_z\right)\right) \, .
\end{aligned}$$

Appendix C

Measuring the deceleration of the vehicle during braking

To avoid additional sensors we use the differentiated and filtered wheel speed of the braked wheels to determine the deceleration of the vehicle during braking, i.e. the vehicle acceleration a equals the wheel acceleration $\dot{\omega}$ times the wheel radius r. This, of course, is not true when the wheel slips, but differentiating and rearranging the definition of the slip in (5.1) gives a bound on the percent error in $\ddot{u}_x$, ϵ_a, that this assumption introduces:

$$\epsilon_a = \left|\frac{r\dot{\omega} - \ddot{u}_x}{\ddot{u}_x}\right| 100\% \leq \left(|s| + \left|\frac{\dot{s}v}{\ddot{u}_x}\right|\right) 100\% \tag{C.1}$$

So the assumption introduces the most error when $\ddot{u}_x$ is small, or $\dot{s}$ and v are large. Under the conditions we used in our tests this did not pose a problem, as Figure C.1 shows. Here, we plot the measured acceleration and the acceleration gotten from $r\dot{\omega}$ of the braked wheel. The mean error appears small, except for the end of the braking maneuver, where the wheel starts locking. In an implementation, one should mind this assumption carefully.

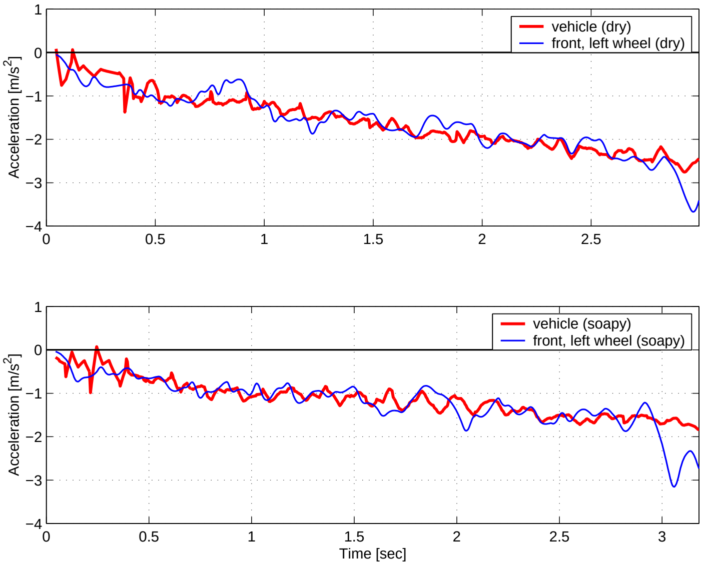

Figure C.1: Wheel acceleration times the wheel radius of the braked wheel and acceleration of the vehicle on dry and soapy road surface, showing that the error expressed by (C.1) is small for our test conditions.

Bibliography

[1] U. Eichhorn B. Breuer and J. Roth. Measurement of tyre/road friction ahead of the car and inside the tyre. *Proceedings of AVEC '92 (International Symposium on Advanced Vehicle Control)*, pages 347–353, 1992.

[2] Th. Bachmann. The importance of the integration of road, tyre, and vehicle technologies. *FISITA XXth World Congress, Montreal, Canada*, September 5, 1995 1995.

[3] Th. Bachmann. Wechselwirkungen im prozeß der reibung zwischen reifen und fahrbahn. Reihe 12 360, Fortschritt-Berichte VDI, 1998.

[4] Pacejka Bakker and Lidner. A new tire model with an application in vehicle dynamics studies. *SAE Transactions, Journal of Passenger Cars*, 98(SAE 890087):101–113, 1989.

[5] A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. *Proceedings of the 37th conference on Decision and Control*, pages 2133–2138, 1998.

[6] Bartz M. Karlheinz B. Gruber S. Semsch M. Strothjohann Th. Breuer, B. and C. Xie. The mechatronic vehicle corner of darmstadt university of technology–interaction and cooperation of a sensor tire, new low-energy disc brake and smart wheel suspension. *Proceedings of FISITA 2000, Seoul, Korea*, June 12-15 2000.

[7] L. Briesemeister and G. Hommel. Overcoming fragment in mobile ad hoc networks. *KICS*, 2000.

[8] L. Briesemeister and G. Hommel. Overcoming fragmentation in mobile ad hoc networks. *Unpublished*, 2000.

[9] H. Chan and U. Ozguner. Closed-loop control of systems over a communications network with queues. *International Journal of Control*, 62(3):493–510, 1995.

[10] A. Daiß and U. Kiencke. Estimation of vehicle speed fuzzy-estimation in comparison with kalman filtering. *IEEE Conference on Control Applications*, pages 281–284, 1995.

[11] Th. Dieckmann. Assessment of road grip by way of measured wheel variables. *Proceedings of FISITA '92 Congress (*Fédération Internationale de Sociétés d'Ingénieurs des Techniques de l'Automobile*), London, GB*, 2, Safety the Vehicle and the Road:75–81, June 7-11 1992.

[12] Th. Dieckmann. *Der Reifenschlupf als Indikator für das Kraftschlußpotential*. Ph.d. thesis, University of Hannover, 1992.

[13] John C. Dixon. *Tires, Suspension and Handling*. Society of Automotive Engineers, Warrendale, PA, 1996.

[14] N. Ebert. Sae tire braking traction survey: A comparison of public highways and test surfaces. *Transactions of the SAE*, (SAE 890638):735–742, 1989.

[15] U. Eichhorn and J. Roth. Prediction and monitoring of tyre/road friction. *XXIV FISITA Congress, London, GB*, Volume 2, "Safety, the Vehicle, and the Road":67–74, June 7-11 1992.

[16] U. Eichorn and J. Roth. Prediction and monitoring of tyre/road friction. *XXIV FISITA Congress, London*, 2:67–74, June 1992.

[17] J.C. (Stanford University Mechanical Engineering Department) Gerdes. Private communications, 2000-2001.

[18] F. Goktas, J. Smith, and R. Bajcsy. μ-synthesis for distributed control systems with network-induced delays. *Proceedings of the 35th Conference on Decision and Control*, pages 813–814, 1996.

[19] F. Goktas, J. Smith, and R. Bajcsy. Telerobotics over communication networks. *Proceedings of the 36th Conference on Decision and Control*, pages 2399–2403, 1997.

[20] F. Gustafsson. Slip-based tire-road friction estimation. *Automatica*, 33(6):1087–1099, June 1997.

[21] M. Hayes. *Statistical Digital Signal Processing and Modeling.* Wiley, New York, 1996.

[22] K. Hedrick, D. Godbole, R. Rajamani, and P. Seiler. Stop and go cruise control. Technical report, California PATH, January 1999.

[23] Wookug Hwang and Byung suk Song. Road condition monitoring system using tire-road friction estimation. *Proceedings of AVEC 2000, Ann Arbor, Michigan*, pages 437–442, August 22-24 2000.

[24] J.K. Hedrick K. Yi and S.C. Lee. Estimation of tire-road friction using observer based identifiers. *Vehicle System Dynamics*, 31:233–261, 1999.

[25] U. Kiencke and A. Daiß. Estimation of tyre friction for enhanced abs systems. *Proceedings of AVEC '94*, 1994.

[26] K. Kobayashi, M. Kameyama, and T. Higuchi. Communication network protocol for real-time distributed control and its lsi implementation. *IEEE Transaction of Industrial Electronics*, 44(3):418–425, 1997.

[27] F. L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks for manufacturing systems. *Proceedings of the ASME, Dynamics Systems and Control Division*, November 1999.

[28] S. Mahal. Effects of communication delays on string stability in an ahs environment. Master's thesis, University of California at Berkeley, May 2000.

[29] D.P. Martin and G.F. Schaefer. Tire-road friction in winter conditions for accident reconstruction. *Transactions of the SAE*, (SAE 960657):732–750, 1996.

[30] S. Sanderson B. Chafe M.J. Macnabb, R. Baerg and F. Navin. Tire/ice friction values. *SAE Technical Paper*, (SAE 960959):49–57, 1996.

[31] Uchanski M. Müller, S. and J.K. Hedrick. Slip-based tire-road friction estimation during braking. To appear in *Proceedings of IMECE'01 (2001 ASME International Engineering Congress and Exposition), USA*, November 11-16 2001.

[32] H.B. Pacejka and R.S. Sharp. Shear force development by pneumatic tyres in steady state conditions: A review of modelling aspects. *Vehicle System Dynamics*, 20:121–176, 1991.

[33] Petersson and Santesson. *Experimental Slip-based Road Condition Estimation*. Master's thesis, Lund Institute of Technology and Univeristy of California at Berkeley, 2000. Lund Report no. 5635.

[34] Reza Raji. Smart networks for control. *IEEE Spectrum*, pages 49–55, June 1994.

[35] A. Ray. Output feedback control under randomly varying distributed delays. *Journal of Guidance, Control and Dynamics*, 17(4):701–711, 1994.

[36] A. Ray and L.-W. Liou. A stochastic regulator for integrated communication and control systems: Part i- formulation of control law. *Journal of Dynamic Systems, measurement and Control*, 113:604–611, 1991.

[37] A. Ray and L.-W. Liou. A stochastic regulator for integrated communication and control systems: Part ii-numerical analysis and simulation. *Journal of Dynamic Systems, measurement and Control*, 113:612–619, 1991.

[38] A. Ray, L.-W. Liou, and J. Shen. State estimation using randomly delayed measurements. *Journal of Dynamic Sys. Measurement, and Control*, 115:19–26, 1993.

[39] A. Ray and N. Tsai. Compensatability and optimal compensation under randomly varying distributed delays. *International Journal of Control*, 72(9):826–832, 1999.

[40] L. Ray. Nonlinear tire force estimation and road friction identification: Simulation and experiments. *Automatica*, 33(10):1819–1833, 1997.

[41] S. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Harcourt Academic Press, 2 edition, 2000.

[42] P. Seiler, R. Sengupta, and K. Hedrick. Analysis of communication delays in vehicle control problems. To appear, 2000.

[43] D. Tilbury, H. Yook, and N. Soparkar. A design methodology for distributed control systems to optimize performance in the presence of time delays. *Proceedings of the American Control Conference*, 2000.

[44] Hill Uchanski, Gobbi and Hedrick. Experimental comparison of smooth and non-smooth adaptive control laws for automobile brakes. *Proceedings of MTNS 2000 (Mathematical Theory of Networks and Systems), Perpignan, France*, June 19-23 2000.

[45] G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *Proceeding of American Control Conference*, pages 2876–2880, 1999.

[46] Gregory C. Walsh, Octavian Beldiman, and Linda Bushnell. Asymptotic behavior of networked control systems. *CCA*, 1999.

[47] J. Wheelis. Process control communications: Token bus, csma/cd, or token ring? *ISA Transactions*, 32:193–198, 1993.

[48] J. K. Yook, D. M. Tilbury, H. S. Wong, and N. R. Soparkar. Trading computation for bandwidth: State estimators for reduced communication in distributed control systems. *Proceedings of the Japan-USA Symposium on Flexible Automation*, July 2000.