Safety issues in collaborative vehicle control

N. F. Maxemchuk
Columbia University
nick@ee.columbia.edu

Abstract

Collaborative driving systems are complex, distributed systems that control physical vehicles. They can reduce accidents, decrease fuel consumption, reduce commute times and increase the capacity of highways. However, errors in their implementation can cause unsafe conditions that result in the loss of human life. Our objective is to find ways to design, verify, and quantify safety in these systems.

In this paper we describe a collaborative driving system that assists in safely and quickly merging vehicles when highways merge, following tolls, and at construction or accident sites. We describe 1) an architecture that partitions the application into modules that can be tested and modified independently, 2) a communication protocol that provides an unconventional set of services that simplifies the implementation of the system, and 3) a strategy for cooperation between the human operator and the system.

1. Introduction

The objectives of cooperative driving systems are to improve safety, decrease travel times, reduce fuel consumption, and increase the capacity of roadways. Computer controlled systems improve safety by responding faster and more reliably than human operators. For instance, an antilock braking system can apply and release the brakes independently on each wheel more quickly, and with greater precision, than a human operator. Systems with sensors improve on computer assisted systems by detecting conditions that require action faster than the human operator. And, systems with communications improve on sensors by 1) notifying nearby vehicles that actions, such as applying the brakes, have been executed before the result of the action can be detected, and by 2) negotiating actions with nearby vehicles before the actions are executed. Cooperative driving reduces congestion by decreasing response times to changing traffic conditions and by decreasing safety margins, such as the distance between vehicles. Decreasing congestion decreases delays and fuel consumption and increases the capacity of a highway.

The rule of thumb for safe vehicle operation is a 2 second gap between vehicles. Therefore, the maximum safe arrival rate in a lane on a highway is .5 vehicles per second, independent of the speed that the vehicles travel. The only safe way to increase the capacity of the highway is to use computer response times to reduce to safe gap to less than 2 seconds.

Multiple lane merges on a congested roadway force drivers to contend with one another. Assisted lane merging can reduce accidents and delays at these dangerous locations by adjusting the spacing between vehicles and allowing the merges to occur safely at higher speeds. These systems are the most technically challenging of the collaborative driving applications. They require cooperation and planning between vehicles in addition to controlling the speed, braking and maneuvers of the individual vehicles.

Systems that control or coordinate vehicles operate in a difficult environment. Not all vehicles on a roadway can collaborate because the systems are deployed incrementally, are evolving, and are not standardized. The cars and trucks that participate may have very different braking and acceleration characteristics. Some malicious or malfunctioning vehicles may insert misleading information into the system. The wireless communications environment has a high error rate and low bandwidth. The radar and sensors being installed in automobiles are inaccurate. And, the environment is uncontrolled. In addition to vehicles, roadways have pedestrians, bicycles, pets, wild life, stationary obstacles and an occasional collapsed bridge.

It is unreasonable to expect automated vehicles to properly respond to every conceivable condition. Instead, automated systems will assist, rather than replace the human operator. They will improve on the drivers responses in common situations that have been carefully analyzed and tested, and place the vehicle in the safest possible condition, and return control to the human operator whenever an unexpected, or untested condition occurs. They will defer to the human operator and minimize the effect of an inevitable collision. An architecture and communications protocol are described to simplify the control programs and increase the number of assisted operations that can be tested.

The architecture, described in section 3, simplifies collaborative merges by dividing the

complex operation into several control functions and identifying operations, such as constructing an accurate map of the locations of nearby vehicles, that are required by several control functions. The common functions are implemented in separate modules that provides a well defined services.

At a minimum, communications protocols in vehicle applications must provide guaranteed delivery and bounded delays. The communications protocol, described in section 4, also guarantees that all common messages in two vehicles are received in the same order, and maintains a list of neighboring vehicles that are cooperating. The voting procedure that is part of this protocol is also used to abort collaborative operations dependent upon changing conditions and to remove anomalous information. When these services are not provided by the protocol layer, the control functions must perform similar operations, and are more complex. We have found efficient methods of providing these services in the protocol layer.

2. Collaborative Merges

Toll plazas frequently have 3 or 4 times as many lanes leaving the tolls as there are on the highway and cause major delays on congested roadways. Similar delays occurs when the the number of lanes on a highway is reduced. Typically, the traffic flows at a higher speed following the merge, so reducing the merge time can reduce the delay. High speed merges on highways also cause serious accidents.

Intelligent merge systems use inter-vehicle communications to share sensor readings between vehicles and plan faster, safer merges. Cooperative merging is a complex operation. In order to test the system, we partition it into a coordination and planning function and three vehicle control functions, speed control, braking, and steering, that are currently being implemented separately.

Speed control: Cruise control currently sets the vehicle speed. If the preceding vehicle is traveling more slowly, a driver must manually exit the system. Intelligent cruise control [1, 2] uses sensors to detect the preceding vehicle and automatically adjust the target speed.

Inter-vehicle communications improves on the operation of stand alone systems by sharing sensor readings with nearby vehicles. Speed adjustments are made based on vehicles that cannot be detected by sensors, or even seen by the human operator. Combining estimates on the position of vehicles from several sensors improves the accuracy of the information. In addition, communications reports changes in a vehicles speed or braking before the effects can be detected.

Using computer control to reduce the speed improves on the drivers response time and decreases the probability of an accident. Automatically accelerating increases the throughput of a congested highway by eliminating rolling traffic jams [3].

Distributed braking: Sensors are currently used to determine the distance to the vehicle in front of a car, or an obstacle behind a car, and provide a warning of an impending collision. These systems will be integrated with anti-lock brakes to prevent a vehicle from hitting the car in front of it. They can also reduce braking to avoid being rear ended, however, the automated system must defer to the driver. The driver may be stopping to avoid hitting a pedestrian. When the driver slams on the brakes and a collision will occur, the automated system can match the speed at impact, then apply maximum braking to stop both vehicles, reducing both the stopping distance and the damage caused by the collision.

Drivers are aware of several vehicles ahead or behind themselves, which is further than sensors can detect. Inter-vehicle communications transfers sensor readings between vehicles, to imitate the driver, and can extend the driver's vision. Information on braking can be transmitted between vehicles before the result of applying the brakes is detected, the equivalent of watching the brake lights. In addition, braking rates can be negotiated so that a car doesn't try to stop faster than the truck behind it.

Coordinated lane changes and merges: Lane control systems use sensors to detect vehicles in a blind spot [4, 5]. These sensors determine when it is safe to change lanes following a negotiated maneuver, or when merging with vehicles that are not part of the cooperative maneuvers. Lane changes will be executed using steering control mechanisms similar to those in self-parking systems.

Inter-vehicle communications is a necessary in cooperative lane merges to negotiate a vehicle sequence and control vehicle spacing and speed. Several strategies for merging a single lane entering a highway are described and analyzed in [5]. In addition, vehicle to infrastructure communications can make merges more fair. The locations of the merges are often fixed. A detector at the merge location can measure the arrival rates and set the merge ratio for equal delays.

3. Architecture

The architecture partitions a complex problem into manageable pieces. The components provide services that are tested separately, and have well

defined interfaces that determine the information that must be passed in each direction.

In the previous section we partitioned cooperative merging into three functionally smaller pieces. These pieces contain common functions that we can separate out, as shown in figure 1. Identifying common components not only simplifies the three functions, but allows us to test smaller independent pieces, and to independently modify and improve each component.

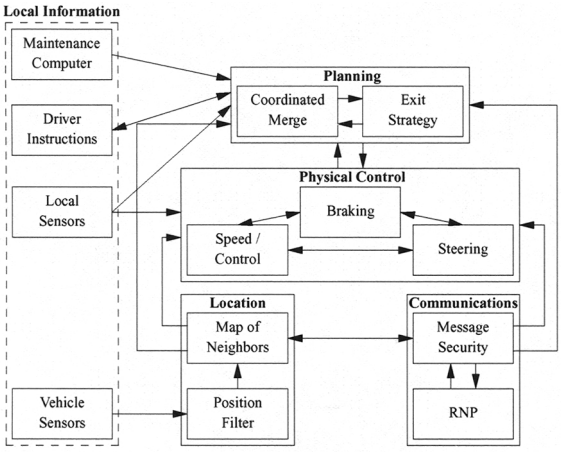

Figure 1. Architecture of Coordinated Merging

There are five modules in this architecture. The modules are similar to the layers in a communications architecture, but intermodule communications is not as constrained.

1. The local information module monitors the vehicle and its environment. Its components include; a) the maintenance computer that detects mechanical failures in an automobile, b) a driver interface to monitor the controls, such as the brake pedal, c) local sensors that monitor the instrumentation, such as the speedometer, the tire rotation relative to the road, used in anti-lock brakes, and road conditions, and, d) radar sensors that detect nearby vehicles or obstacles.
2. The planning module implements the merging strategies and safely return control to the driver when unexpected conditions occur.
3. The vehicle control module coordinates the speed, braking and steering of this vehicle with nearby vehicles.
4. The communication module addresses security issues, and accesses the services provided by RNP. All communications passes through the security program which a) attaches the vehicle's signature to transmitted messages, b) checks and records the credentials of each received message, so that vehicles that insert malicious messages can be prosecuted, and c) uses the RNP voting procedure to detect and remove inconsistent messages.
5. The mapping system combines the radar sensors at different vehicles to improve the estimates of vehicle locations, and constructs a map of nearby vehicles. Communications guarantees that neighbors have the same map.

4. The Reliable Neighborcast Protocol

In this section we describe the reliable neighborcast protocol, RNP [6]. Inter-vehicle communications protocols are surveyed in [7, 8].

RNP uses a neighborcast model, figure 2. Each vehicle communicates with a set of nearby vehicles, its neighborhood. The neighborhoods for nearby vehicles overlap, but also contain vehicles that are different.

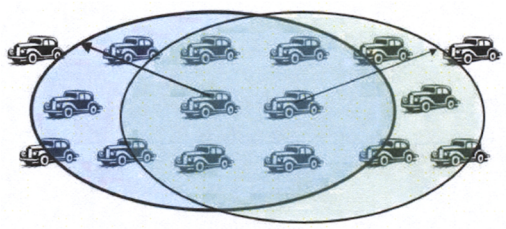

Figure 2. Neighborcast

RNP is implemented as an overlay on a set of overlapping groups, section 4. 2. The groups use the mobile reliable broadcast protocol, MRBP, section 4. 1. The services provided by RNP are described in section 4. 3.

4.1 MRBP

MRBP [9] reliably delivers messages that are broadcast in a wireless region to all of the receivers, in the same order. It provides the service with very little overhead by passing an acknowledgement token, first used in [10]. The protocol uses an aggressive token passing strategy and a distributed voting strategy to continue to operate as the receiver group changes.

A source uses the dedicated short range communications protocol, DSRC [11] to transmit its message to the current token site. The source message is also received by the other receivers. Receivers periodically pass a token that also lists the source messages it acknowledged. Recovery of a missing token begins shortly after its scheduled time, then any missing acknowledged messages are recovered. All receivers place the messages in the order that they are acknowledged. When there are no losses, the token message guarantees that a source message is delivered to all of the receivers.

A receiver passes the token at its scheduled time, whether or not it has recovered the token from the

previous receiver. When all of the transmitted tokens are recovered, if multiple tokens acknowledge the same broadcast message, the message is placed in the sequence using the earliest token that acknowledged it.

An acknowledgement may be missing at a receiver because the receiver failed to recover it, because the receiver that was scheduled to transmit it left the group, or because of collisions in the DSRC protocol. The token passing interval can be smaller than the retry interval in DSRC. A distributed voting strategy is used to construct a common list of transmitted tokens at all of the receivers. When a receiver's token is not included in the list, it is removed from the group. The same voting mechanism constructs a common list of sequenced messages at all of the receivers. A source uses this list to determine that its message is received by all receivers. A receiver that wants to enter a group transmits a message, and becomes part of the group when its message is voted into the message sequence.

Receivers vote on a token by listing which earlier tokens they are missing when the pass their own token. The vote begins after all receivers have had an opportunity to recover a missing token, and ends one token cycle time later.

When the vote on the token transmitted by receiver r_i is tallied; A receivers have the message, B receivers do not have the message, and, C receivers have left the group and did not vote. r_i is voted out of the group when $A < m/2$, where m is the number of receivers in the group when the vote starts.

Each receiver may not recover all of the votes. Receiver r_j makes the correct decision or leaves the group. If $A_j \geq m/2$, then $A \geq m/2$, and r_i remains in the group. If $B_j > m/2$, then $B > m/2$, $A < m/2$, and r_i is removed from the group. If $B_j \leq m/2$, and $A_j < m/2$, r_j is uncertain whether or not $A < m/2$, and leaves the group itself. If $A_j \geq m/2$, but r_j has not recovered the token then r_j leaves the group.

4.2 RNP Implementation

RNP is implemented on top of overlapping MRBP groups, each of which covers a length of a one-dimensional space on a highway, as shown in figure 3. A vehicle transmits in all of the MRBP groups that cover its location. A vehicle's neighborhood extends a distance f in front and behind the vehicle, and it can communicate with each of its neighbors in at least one of its groups as long as the overlap between broadcast groups is $\geq f$. The number of groups covering an area should be small to avoid transmitting and acknowledging messages too many times. Vehicles communicate in at most 2 groups when the group size is $> 2f$, and the edge of a group does not overlap the center of an adjacent group. The size of a group should also be small, to circulate the token and obtain the guarantees as quickly as possible.

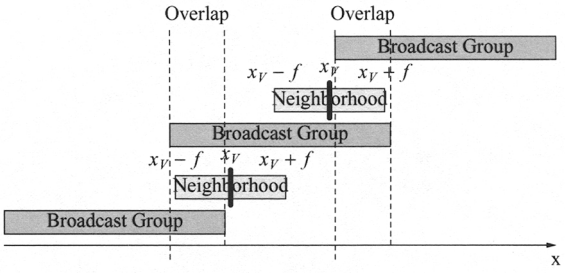

Figure 3. A one-dimensional RNP network

Increasing the size of the overlap between broadcast groups, as shown in figure 4, allows a vehicle to enter a broadcast group before it must use that group to communicate with a neighbor. This maintains communications between neighbors as they change broadcast groups.

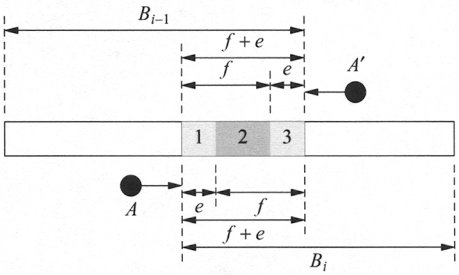

Figure 4. Vehicles changing broadcast groups when the minimum overlap is increased by e

Vehicle A starts to join broadcast group B_i when it enters region 1. Its neighborhood extends at most f. Therefore, all of its neighbors are in B_{i-1} until it reaches region 2. Vehicle A has the time it takes to cover distance e to join B_i, which is approximately a token rotation time. For instance, if it takes at most 5 seconds for a vehicle to join a broadcast group, the average speed of vehicles is 50 mph, all vehicles travel within 5 mph of the average speed, and the broadcast group moves forward at the average vehicle speed, then $e \geq 7.5$ feet maintains communications.

In reference [6], we describe a distributed protocol that moves groups with vehicles, expands or contracts groups as vehicles move with respect to one another, joins adjacent groups to prevent more than two groups from covering an area, and splits groups to maintain small groups. A vehicle reports its current location and suggested change to the group when it transmits its token, but the change doesn't take place until the token is voted into the list. Voting guarantees that all of the vehicles change the group in the same way at the same time.

RNP filters the messages from the MRBP groups,

and performs partial sequencing. It removes messages from vehicles outside the neighborhood, and duplicate messages received in more than one group. A receiver only recovers missing acknowledged messages from vehicles in its neighborhood, or a vehicle joining the group, and only votes on those messages. Partial sequencing guarantees that all vehicles that receive two messages places them in the same order. The message sequence in two groups may be different when the earlier of two messages is retransmitted in one of the groups. RNP constructs a unique sequence by using a time stamp on the tokens to determine which message was acknowledged first in any group, and sending a message to the other groups to give that message the earlier time stamp. The correction message is acknowledged, and must be recovered by all of the receivers in the neighborhood.

4.3 Protocol Services:

The services provided by a protocol can simplify the applications that use the protocol. RNP has layers of delivery guarantees with different delay bounds that are used for different functions in the applications. The DSRC protocol guarantees that a source message is delivered to the current token site.

The recovery mechanisms in MRBP guarantee delivery to all operable receivers in a group. Each receiver begins recovery of an acknowledgment at its scheduled time then recovers missing acknowledged messages. The delay for the guarantee is bounded on the order of several token passing intervals, and is used for emergency operations, such as instructing vehicles to quickly applying their brakes.

The voting and filtering mechanism in RNP guarantees sequenced delivery to all receivers in a neighborhood. This is used for collaborative maneuvers, to guarantee that all participants simultaneously execute the same operations, or to abort the maneuver when some participants have not recovered the instructions. The voting provides different levels of participation for different maneuvers. For instance, some collaborations, such as deciding the order of the vehicles at a merge, may require a majority vote while others, such as maneuvering a vehicle between vehicles in an adjacent lane, requires unanimous consent. The votes are also used to identify disagreements between the measurements or instructions from different vehicles to identify malicious or malfunctioning users.

The MRBP/RNP combination guarantees continuous communications between all nearby vehicles. MRBP provides continued communications as vehicles enter or leave the area covered by the group. And, the overlap in the RNP architecture guarantees that vehicles are in a broadcast groups before they must enter a neighborhood.

5. Conclusion

Computer assisted lane merging will reduce accident rates on highways, decrease fuel consumption, decrease commute times, and increase the capacity of highways. All national priorities. However, the implementation is complex and we must guarantee that the system "does no harm". We are moving toward this goal by only assisting drivers in well tested situations, breaking the complex architecture into simpler pieces, and providing guarantees in the communications protocol that eliminate many of the special cases and conditions which would have to be tested in the applications.

REFERENCES

[1] B. Arem, C. Driel, R. Visser, "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics," IEEE Transactions on Intelligent Transportation Systems, Vol. 7, Issue 4, Dec. 2006, pp 429-436.

[2] Y. Zhang, E. B. Kosmatopoulos, P. A. Ioannou, "Autonomous Intelligent Cruise Control Using Front and Back Information for Tight Vehicle Following Maneuvers," IEEE Transactions on vehicular technology, Vol. 48, No. 1. January 1999.

[3] N. F. Maxemchuk, P. Tientrakool, T. Willke, "The Role of Communications in Cyber-Physical Vehicle Applications," The Book of Automotive Informatics and Communicative Systems: Principles in Vehicular Networks and Data Exchange, 2009.

[4] Heng Wei, M. Pavithran, Qing-An Zeng, "A Dynamic Merge Metering-Based Unconventional Alternative to Traffic Control at Highway Bottlenecks," IEEE Intelligent Transportation Systems Conference, 2007, Sept. 30 2007-Oct. 3 2007, pp.331-336.

[5] Z. Wang, L. Kulik, K. Ramamohanarao, "Proactive traffic merging strategies for sensor-enabled cars," ACM workshop on Vehicular ad hoc networks, VANET '07, September 2007, pp 39-48.

[6] N. F. Maxemchuk, P. Tientrakool, T. Willke,, "Reliable Neighborcast," IEEE Trans. on Vehicular Technology, vol. 56, iss. 6, Part 1, Nov. 2007, pp. 3278-3288.

[7] T. Willke, P. Tientrakool, N. F. Maxemchuk, "Survey of vehicle-to-Vehicle Communications," Accepted for publication: IEEE Communications Surveys.

[8] M. L. Sichitiu, M. Kihl, "Inter-vehicle communication systems: a survey" IEEE Communications Society Surveys and Tutorials, 2008, vol. 10, iss. 2, pp. 88-105.

[9] T. L. Willke, N. F. Maxemchuk, "Coordinated Interaction Using Reliable Broadcast in Mobile Wireless Networks," Networking 2005, May 2 - 6, 2005, University of Waterloo, Waterloo Ontario Canada.

[10] J-M. Chang, N. F. Maxemchuk, "Reliable Broadcast Protocols," ACM Transactions on Computer Systems, Vol. 2, No. 3, Aug. '84, pp. 251-273.

[11] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," Proc. 1st ACM workshop on Vehicular ad hoc networks, October 2004.