A CONTROLLER FOR MERGING TRAFFIC ONTO A HIGHWAY

Omar Ahmad
Research and Development Project Leader
&
Huidi Tang
Senior Systems Analyst
National Advanced Driving Simulator
The University of Iowa

Abstract

This paper presents a model of a behavior to merge onto an interstate for an autonomous vehicle. This autonomous vehicle exists in a virtual environment of a microscopic traffic simulation system. The merge behavior model constitutes a part of a complete set of behaviors that represent the overall activities exhibited by a realistic and reactive vehicle in a driving simulation virtual environment. The merge behavior is an interesting behavior since the autonomous vehicle has to satisfy multiple (conflicting) conditions in order to perform a successful merge into a congested interstate. This paper presents a brief description of the overall vehicle behavior model and then describes the merge behavior model in detail.

Introduction

In the National Advanced Driving Simulator (NADS) virtual environment, autonomous vehicles are used to create realistic traffic in scenarios. Microscopic control at the level of individual autonomous vehicles leads to traffic that is used to present a realistic experience for the simulator driver. Autonomous vehicles interact with each other and react to simulator driver in real-time. Each autonomous vehicle utilizes a sophisticated behavior model that has the ability to process several distinct and competing requirements. Autonomous vehicles exhibit realistic behaviors such as tracking their lanes, following other vehicles, navigating intersections, changing lanes and merging onto highways. The paper describes our implementation for the merging onto highway behavior.

A highway merge presents an extremely dynamic, complex and highly risky traffic situation to a vehicle attempting to enter the highway. As the vehicle traverses the on-ramp, it needs to constantly monitor the traffic on the highway and make various speed adjustments, while trying to keep a safe distance to vehicles in front of and behind it on the entrance ramp at the same time. Performing a realistic merge maneuver requires accurate information from the virtual environment about the layout of the road network, the location of other vehicles in the vicinity and the physical capabilities of the vehicle being modeled.

Some work on highway merge protocols has been done in the context of the automated highway system (Gibson, 1997). However, these approaches are specifically designed to work with the automated highway system and make some assumptions that are not relevant in the NADS virtual environment.

This paper begins with a description of the autonomous vehicle model used in the NADS. It then describes in brief detail the formalism used to model the driver behaviors. It then describes in details the merge maneuver and its implementation within the context of the overall system.

Autonomous Vehicle Model

The NADS autonomous vehicle model separates the driver's behavior model from the vehicle's physical model. This approach allows the use of the same driver behavior model with a variety of physical models such as passenger cars, sports utility vehicles, trucks and emergency vehicles. Even though these vehicles are different in terms of their appearance and performance, they use essentially the same behavioral logic to move around the road network and interact with other objects.

The vehicle's physical model has been designed to receive accelerator and steering inputs from the behavior model. Our intent has been to remove the low-level responsibilities of how a vehicle moves from the behavior model and to place them with controllers that are integrated with the physical model. At the same time, the behavior model has access to the physical capabilities of the particular vehicle and thus it can adjust its decision-making and consequent demands from the physical model based on the type of the vehicle being controlled.

Presented at the IMAGE 2003 Conference
Scottsdale, Arizona 14-18 July 2003.

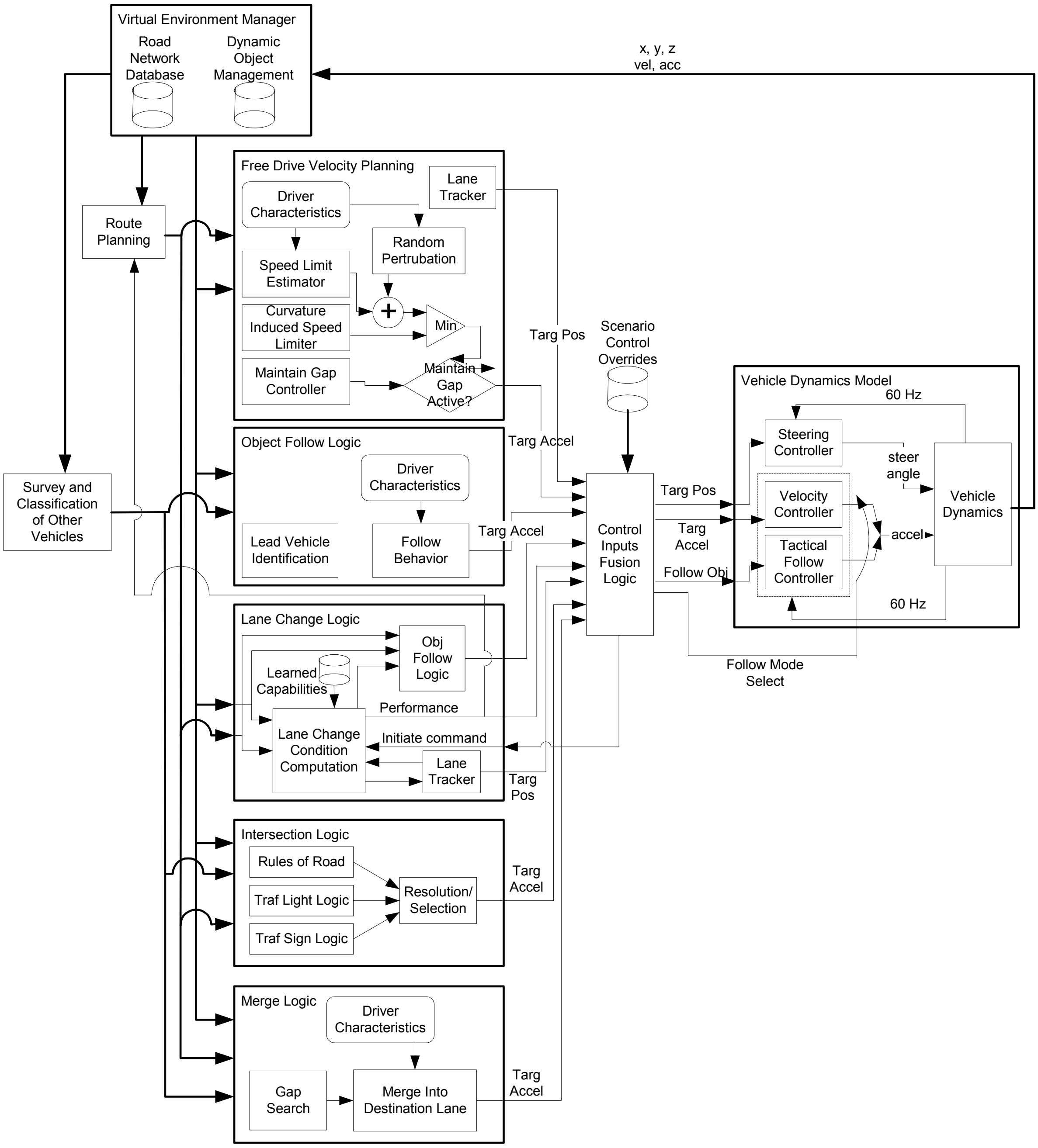

Fig. 1 Autonomous Driver Model Block Diagram.

Fig. 1 displays the autonomous driver model. The block diagram has been somewhat simplified to provide a better understanding of the framework within which the merge maneuver operates. The figure demonstrates the aforementioned separation between the dynamic model and associated controllers and the behavioral model. The figure also shows how the model concurrently evaluates multiple goals and satisfies multiple constraints. Specifically, the Control Inputs Fusion Logic block, illustrated in Fig. 1, receives control inputs from multiple sub-models and determines how to fuse such inputs. The output of the fusion block is then directed to the dynamics model (Ahmad, Papelis, Bulusu & Gade, 2001).

The vehicle's behavior model has been implemented using the Hierarchical Concurrent State Machine (HCSM) formalism (Cremer, Kearney & Papelis, 1995). This formalism builds upon standard state machines by making them hierarchical and concurrent. Thus, we are able to model vehicle behavior as a tree of state machines where each part of the tree implements a separate behavior. The ability to execute sibling HCSMs concurrently enables multiple goals to be evaluated. A detailed description of the HCSM formalism is beyond the scope of this article but more details can be found in Papelis & Ahmad (2001).

The autonomous vehicle has the ability to move around the road network and perform normal traffic operations on its own. Autonomous vehicles display the following major behaviors: lane tracking, following other vehicles, lane changes, navigating intersections and performing highway merges. Each one of these major behavior categories encapsulates several other behaviors and functionality. For instance, the lane tracking behavior encapsulates the following functionality: keeping the vehicle positioned in the center of its lane, obeying speed limits, slowing down on curves in the road and obeying external commands to control the vehicle's velocity and maintain specific positioning with respect to another object such as the simulator driver.

In general, an autonomous vehicle moves around the road network along a specified path or a path generated at random and continues to do so until specifically instructed to do otherwise. A variety of run-time instructions can be sent to an autonomous vehicle during the simulation such as maintaining a specific velocity, maintain positioning relative to another vehicle, performing a lane change maneuver and turning a specific direction at an upcoming intersection among several others. Due to limited space, we omit further details of the overall vehicle model. The readers are urged to refer to Papelis, Ahmad & Schikore (2001) for more details.

Merge Maneuver

Fig. 2 illustrates a typical highway merge maneuver. The point P represents the transition point where the on-ramp lane first intersects the highway's rightmost lane. The point P1 marks the location that the merging vehicle needs to stop at the latest when a gap could not be found. If the merging vehicle stops beyond the point P1 then it creates a collision hazard for the vehicles traveling on the highway.

As a vehicle enters the on-ramp, it has to start monitoring the traffic on highway around and before the spot where the ramp merges with the highway. In particular, the merging vehicle makes an assessment of the density and velocity of the traffic on the highway so that it can accelerate to bring its own speed up to the speed of the highway traffic. The vehicle then finds a gap that gives it a big enough space-cushion for a safe merge.

The timing for the gap search is critical. The merging vehicle cannot pick a gap too soon since, by the time it tries to perform the actual merge, the gap may no longer be appropriate due to various reasons. For example, vehicles in front of the merging vehicle on the on-ramp may have slowed down; or the acceleration or deceleration of the vehicles between the target gap have rendered it inappropriate. On the other hand, if the merging vehicle waits too long, it may not find an appropriate gap at all; thus forcing the merging vehicle to come to a stop. In this case, it may be unable to stop at the desired location, which could cause collisions. It is also susceptible to being hit from the rear, as vehicles behind the merging vehicle may not realize it is slowing down or expect it to stop. When an appropriate gap is found, the merging vehicle signals in advance and follows its lane to smoothly blend into highway traffic.

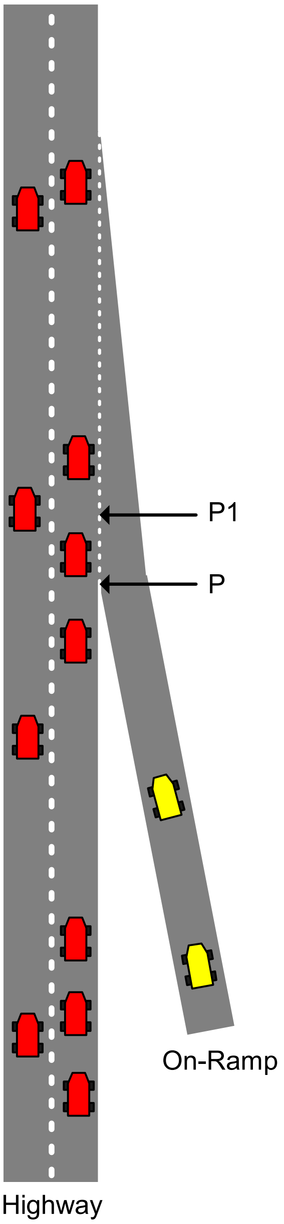

Fig. 2 Highway merge maneuver.

The Merge HCSM

Fig. 3 displays the HCSM that is used to implement the merge behavior. This HCSM encapsulates the behavior explained in the previous section and is also displayed as a part of the overall autonomous vehicle model in the Merge Logic box from Fig. 1. For the remainder of this paper, the term HCSM or state may be used interchangeably to refer to the same state machine.

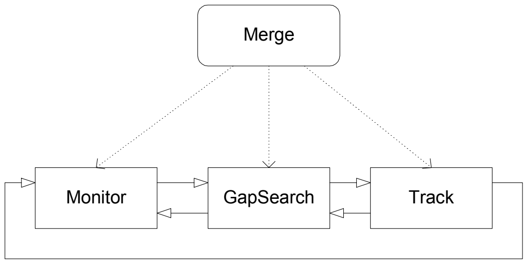

Fig. 3 The Merge HCSM.

In Fig. 3, the Merge HCSM has 3 child HCSMs or states: Monitor, GapSearch and Track. The parent-child relationship is presented by the dotted lines from Merge to each of its child states. These 3 states are represented by rectangles and are connected to each other by 5 transitions that show how control transfers from one state to another. Transitions are represented by the solid lines between the 3 states. In the given configuration, only one child state may be active at any given moment and these 3 child states are referred to as being sequential HCSMs. If the child states lacked transitions between them then they would be referred to as being concurrent and all 3 would execute simultaneously and the parent HCSM would have to monitor the outputs of all 3 states.

A parent HCSM and its children execute in a specific order. Each HCSM has associated with it code that runs before its children are executed, referred to as PreActivity, and code that runs after it's children have finished execution, referred to as PostActivity. Usually, an HCSM uses the PreActivity phase to send relevant information down to its children, before they execute their own PreActivity, and the PostActivity phase to collect information from its children, after they execute their own PostActivity. Since only one child may be active at any given moment in the Merge HCSM, it simply collects information from its one active child. Currently, each child outputs a target acceleration that is returned by the Merge HCSM to the Control Inputs Fusion Logic (shown in Fig. 1) during the PostActivity phase. Furthermore, each transition also has code associated with that determines when control should transfer from one state to another.

The rest of this section discusses each of the merge states in detail along with their transitions and what has to happen in order for control to transfer from one function to another.

Monitor

By default, the Monitor state is active. In this state, the merging vehicle looks at the other vehicles on the highway and calculates the average velocity of vehicles on the highway behind point P. It sets the target acceleration based on the higher value of the speed limit and the average velocity previously calculated. This helps the merging vehicle achieve a velocity that is consistent with the expected velocity that it will need to merge onto the highway. This state remains active even when the autonomous vehicle is not in an on-ramp and returns no target acceleration in that case.

Control switches from the Monitor to the GapSearch state when the merging vehicle's time-to-arrival to point P1 is less than 5 seconds.

GapSearch

In the GapSearch state, the merging vehicle looks for the gap between the vehicles on the highway that it wants to merge into. It compares its own time-to-arrival to point P1 against the time-to-arrival of the vehicles on the highway to point P1 to find a gap. It also checks to make sure that it picks a gap that won't result in a collision with the vehicle before or after the gap.

In the event that the merging vehicle is unable to find a suitable gap, it calculates the minimum distance that it needs to travel in order to come to a stop at point P1. If the merging vehicle is within that distance, it applies its brakes and attempts to come to a stop at point P1.

Once a gap has been successfully located, control switches from the GapSearch to the Track state.

Alternatively, control switches back to Monitor if the merging vehicle sees that vehicles on the highway have moved out of the way and there is no longer a need to find a gap.

Track

In the Track state, the merging vehicle adjusts its speed to remain on target to merge into the selected gap. It calculates its target acceleration so that it meets the gap at point P1.

Once the merging vehicle has successfully merged onto the highway, control switches from the Track State back to the Monitor state.

Alternatively, if the gap closing down is no longer available, control switches back to the GapSearch state to search for another gap.

Conclusion

This paper has described a vehicle behavior model for merging onto a highway and how this model fits into the framework of vehicle's overall behavior model. The HCSM formalism allows us to encapsulate the behavior of the merge model in a tree of hierarchical state machines. Even though the merge maneuver is extremely dynamic and complex behavior, this structured design along with information from the real-time virtual environment provides the accuracy need to accomplish this maneuver.

References

Ahmad, O., Papelis, Y., Bulusu, S., & Gade, V. (2001). Automatic Learning by Autonomous Driver Agents as Applied to Performing Realistic Lane Change Maneuvers. Paper presented at the International Symposium on Measurement, Analysis and Modeling of Human Functions 2001. Hokkaido University, Sapporo, Japan.

Cremer, J., Kearney, J., & Papelis, Y. (1995). HCSM: A Framework for Behavior and Scenario Control in Virtual Environments. ACM Transactions of Modeling and Computer Simulation, 5(3): 252-267.

Gibson, Carl H. (1997). Automated Highway Merging Prootocols and their Effectiveness on Highway Operations and Vehicle Performance. M.S.E. Thesis, University of California, Berkeley, USA.

Papelis, Y. & Ahmad, O. (2001). A Comprehensive Microscopic Autonomous Driver Model for Use in High-Fidelity Driving Simulation Environment. Proceedings of the 80th Annual Transportation Research Board Meeting. Washington D.C., USA.

Papelis, Y., Ahmad, O. & Schikore, M (2001). Scenario Definition and Control for the National Advanced Driving Simulator. ESV: 17th International Technical Conference on the Enhanced Safety of Vehicles, Report No. 346. Amsterdam, Netherlands.