Automated Vehicle Merging Maneuver Implementation for AHS

Xiao-Yun Lu, Han-Shue Tan, Steven E. Shiladover and J. Karl Hendrick
Objectives and Results

- Propose a real-time algorithm to compute a smooth reference movement trajectory for the merging vehicle based on the speed of the main lane vehicle
- A real-time implementation of a general merging algorithm for automated highway system (AHS) is presented
- The experiment is conducted on the real environment and the speed trajectories of the vehicles involved are shown
1. Introduction

- Automated highway systems (AHS) offers the potential of significant improvements in traffic flow volume and stability
- Contains only vehicles having completely automated control of lateral and longitudinal vehicle motions
- Vehicle merging is generally formed as the problem of one vehicle from the entry lane merging between two vehicles travelling in a platoon in the main lane
- Roadside computer performs coordination tasks involving determining vehicle identity, selection of two vehicles in the main line between which to merge, passing information to relevant vehicles, coordinating platooning vehicles on the main road
1. Introduction (cont.)

- Automated merging maneuver is broken down into 2 phases: 
  1) splitting two vehicles on the main line 
  2) Adjusting speed and acceleration of the merging vehicle

- Introduce the concept of virtual platooning, that is, to shift the time of platoon formation before the start of lateral movement
2. Adaptive Merging Algorithm
   
   The longitudinal control problem

   Two main requirements:

   1. Two vehicles in the main lane execute “split” creating a spacing of size $2l_{des\-follow} + l_2$ before the lateral movement takes place

   2. The speed of the merging car is adjusted in real time according to the front vehicle

   Merging ahead of a platoon, merging behind a platoon, and merging in between two vehicles in a platoon can all be reduced to the situation that one vehicle merge between two chosen vehicle
2. Adaptive Merging Algorithm

- Layout

- The algorithm computed a smooth reference trajectory for the merging car
- Use a longitudinal controller that is fast enough to track the trajectory
Outline

3. Implementation issues
   - Magnets are installed in both main lane and merging lane at a regular distance. If magnets
   - The speed measurement is obtained by fusing speed readings from four wheel sensors
   - The relevant information such as speed measurements and distance is communicated using WaveLan radio while medium access is regulated by token ring passing
   - The optimal dynamic back-stepping sliding surface controller is adopted as the longitudinal controller for its strength for distance control
4. **Test results**

- The starting position on the main lane is 170~190m away from the merging point, and the merging car starts from 153m to the merging point
- The maximum speed of the front vehicle was set to 21.0 km/h, 24.3 km/h and 27.6 km/h for three different test cases
- Distance error, speed error of the controller on merging vehicle relative to the computed reference speed computed by the algorithm is presented
4. Test results

Fig. 4. RFS test track, Maximum speed 21.0 km/h.
There are in fact three phases for speed and distance error. The merging car starts by following the front car, then accelerates at a lower acceleration and finally increases its speed to catch up with the speed of the front car.
The Adaptive Merging Algorithm

- The mathematical model for merging

\[ v_{md}(t_{merg}) = v(t_{merg}) \]
\[ \int_{t_{merg}}^{T_{virt}} v_p(t) dt = |Q_{start-1} Q_{virt}| \]
\[ \int_{t_{merg}}^{T_{virt}} v_{md}(t) dt = |Q_{start-2} Q_{virt}| \]
\[ |Q_{virt-1} Q_0| + (l_1 + l_{des-follow}) = |Q_{virt-2} Q_0| \]
\[ v_{md}(T_{virt}) = v_p(T_{virt}) \]
\[ a_{md}(T_{virt}) = a_p(T_{virt}) \]
The Adaptive Merging Algorithm

- The first condition depends on the performance of longitudinal controller
- The second, the third and the fourth conditions specify the position of the front car, the merging car and the back car, respectively
- The last two conditions dictate that the speed and the acceleration of the platoon and the merging car should match when the virtual platoon is formed
The Adaptive Merging Algorithm

- The longitudinal controller

\[
v_{\text{mul}}(t) = \begin{cases} 
(1 - \alpha(t)) v(t_{\text{merg}}) + \alpha(t) v_p(t), & t_{\text{merg}} \leq t \leq T_{\text{virt}} \\
v_p(t), & T_{\text{virt}} < t \leq T_{\text{merg}} 
\end{cases}
\]

\[
\alpha(t) = \alpha_0^\beta(t), \quad \beta > 0
\]

\[
\alpha_0(t) = \frac{\int_{t_{\text{merg}}}^{t} v_p(s) \, ds}{\int_{t_{\text{merg}}}^{t} v(s) \, ds + \text{dist}_{\text{para}}}
\]

\[
\text{dist}_{\text{para}} = |Q_{\text{start} - 1} Q_0| - |Q_{\text{start} - 2} Q_0| + l_1 + l_{\text{des} \_ \text{follow}} > 0
\]

- The trajectory could be divided into two phases
  1. Adjust both speed and distance such that the virtual platoon is completed at the end of the interval.
  2. The merging vehicle follows the speed and the acceleration of the platoon in main lane, and the real merging can be executed at \( t = T_{\text{merg}} \).
The Adaptive Merging Algorithm

- $\alpha(t)$ varies with time and is closely related to the speed of the merging vehicle and the platoon
- A larger $\beta$ leads to earlier formation of the virtual platoon
- Before the lateral movement, the back car follows the front car. After the merging car moving into the gap, a new platoon is then formed and a transition has to be in place to make the back car follow the merging car.