Survey and Experiment: Lane Merging

HSIANG-YU CHANG

June 7, 2023

1 Introduction

An on-ramp or work-zone event may severely slow the traffic, so finding a strategy for lane-merging is important and is trending in the intelligent vehicle industry. In this report, a survey and an experiment are conducted. The survey aims for papers about different aspects of lane merging such as passing order, merging point, and trajectory. The experiment shows the performance with different strategies.

2 Survey

2.1 Passing order

In [1], an algorithm for passing order is developed. The authors assume that there is a two-lane merging scenario. To ensure safety, two vehicles must have a minimum time gap when entering the merging point. However, with cooperative adaptive cruise control (CACC), two vehicles on the same lane have a smaller minimum time gap. The authors use a dynamic programming approach to determine the optimal passing order.

In [2], a set of formulations is developed for scenarios with on-ramps and off-ramps. It helps find each vehicle's optimal speeds, accelerations at each time slice, and the optimal time starting to merge considering road density, safety, and energy consumption, which can determine the passing order. If the road density is high, it means that the traffic on the lane is too heavy and the vehicles on the lane should have the priority to pass the merging area so that they can exit faster.

In [3], a model determining whether a vehicle merges or not using the classification and regression tree (CART) method is proposed. First, the authors collect the data at a work zone merging area in reality. Then, they use the CART method to build a model based on speed, distance, vehicle type, merging decision, and time to collision (TTC), a variable that they use to measure safety. The CART is basically a serial of 'if something then A, else B' that leads to merging or not in the end.

In contrast with [3], both methods in [1] and [2] determine the passing order in a cooperative manner, which means that they consider all vehicles' actions. In [4], since the authors think that the major drawback of the ideas from papers in recent years is the extensive computational time, a non-cooperative method using a two-player game with simple payoff functions is proposed for on-ramp events. The two players are the merging vehicle (P1) and the lag vehicle (P2), and their strategies are 'merge or not merge' and 'decelerate or accelerate' respectively. Two algorithms are developed and used as the payoff functions for P1 and P2 considering the relation of the position and velocity between P1, P2, and the lead vehicle. An experiment is conducted with CACC and the speed harmonization and the approach above. It shows that both CACC and speed harmonization are essential to the approach.

In [5], a maneuver planner for an on-ramp scenario with two main lanes and one on-ramp lane is proposed. For the maneuver planner, three motion maneuvers are used, which are the car-following, cruising, and cooperative lane-changing maneuvers. The car-following maneuver is applied to vehicles whose directly preceding vehicle is already the target vehicle. The cruising maneuver is for leading vehicles in each lane. The cooperative lane-changing maneuver is for vehicles whose directly preceding vehicle is not yet the target vehicle. With these maneuvers, parameters for relative speed, gap, acceleration, status, etc., are generated in order to make the maneuver planner. Note that the parameters are all about the relations between each vehicle and its directly preceding vehicle. The maneuver planner is a model that optimizes dynamic vehicle sequences, aka the passing order, by minimizing disturbances to upstream traffic and it can be restricted to lane-changing times or lane-changing directions.

2.2 Control

After determining the passing order, the next step is to control the vehicles. Good control can reduce traffic shockwaves and merge efficiently. In [6], a vehicle platooning control with a hybrid ACC-DMPC controller is proposed to extend the merging gap. Adaptive cruise control (ACC) ensures safe interactions between vehicles. Distributed Model Predictive Control (DMPC) consists of many model predictive controls (MPC), and an MPC can predict how a vehicle will move. The authors assume that there are n vehicles denoted as $V_1 \sim V_n$, and V_1 is the closest vehicle to the lead vehicle. First, the ACC controller generates a reference time headway for V_1 while $V_2 \sim V_n$ also follows the same controller. Then, based on the information by the ACC controller and the target time headway of V_2, the MPC_1 controller can predict V_1's movement and generate a time headway for V_2 while $V_3 \sim V_n$ also follow the same controller (MPC_1). Similarly, MPC_2 generates a time headway for V_3 based on MPC_1 and so on. Note that each vehicle has its own target time headway initially, based on its functional performance. By this method, the fact that the controller and the time headway keep changing results in a smaller speed variance and a naturalistic forward-moving shockwave, which is efficient and safe for the traffic flow.

In [7], a method for optimizing the approaching speed of the merging vehicle in an on-ramp event is proposed. First, the authors present a model that estimates the merging decision of the merging vehicle and the acceleration of the vehicles on the main lane. Then, they use the estimation and a set of formulations to find the optimal acceleration of the merging vehicle while minimizing the decision entropy. The model consists of two models, the acceptance model, and the motion model. The acceptance model is used for estimating whether the vehicle is going to merge or not, and it is based on the logistic regression model and uses the data collected from the real world. The data is about "whether the driver merges or not under different circumstances like distance, velocity, etc". The motion model is used for estimating the acceleration of the vehicles on the main lane corresponding to different merging decisions, and it is based on the switched Proportional-Derivative (PD) acceleration control law. The set of formulations is based on the control strategy, MPC. However, an MPC problem is hard to be solved in real-time so it uses a randomized model predictive control (RMPC) approach instead.

2.3 Experimental

There are also experimental papers about lane merging. In [8], the authors conduct an experiment to have the statistics to analyze each factor's (the desired merging positions, the size of merging gaps, the speed of the leading and following vehicles, etc.) significance in work zone scenarios, which is very helpful to methods using machine learning. The first step is to get the statistics of the desired merging positions (the drivers perform when no vehicles are around). The second is to get the statistics of the merging performances when vehicles are around, and then compare it with the statistics in the first step. Then, the authors make a model composed of the coefficients indicating the significance of each factor. The third step is to find the relations between factors and the successful merging positions (the positions after merging).

In [9], the authors analyze the effect of the late-merging strategy in a non-cooperative manner, while it has been claimed by recent papers that it is beneficial in a cooperative manner. The Intelligent Driver Model and a kinematics-based gap model are employed for the simulations in this paper. The Intelligent Driver Model is for human-like vehicle following so it's simple and only requires little input and follows the '2-second rule'. The kinematics-based gap model is for lane changing and the concept is that the preceding vehicle will accelerate and the following vehicle will decelerate to create a bigger gap, and then the merging vehicle will decelerate and merge. The vehicles in the simulations are either conservative vehicles or aggressive vehicles, and an aggressive vehicle means that it only merges late. The result shows that the late-merging strategy is beneficial to fuel consumption and vehicle flow only when the aggressive-vehicle rate is lower than 25% and benefits the most at 20% rate. The result also shows that the collision will happen frequently as the aggressive-vehicle rate rises.

In [10], a 3-class linear regression model that predicts the desired merging position is proposed. The authors also show the whole process including their thoughts, experiments, and revisions. They think that due to the driver heterogeneity, it is not sufficient to make a model with only one class, just like most of the other papers do. They also think that some factors actually affect merging decisions less and some actually affect more. Through experiments, they find that three classes are the optimal number according to the Bayesian information criterion (BIC). They also find that vehicle types, traffic density, and surrounding lane-changing events affect merging decisions a lot, while some of the related positions and speeds are not as important as expected. In [11], similarly, the authors first propose the state of decision (SOD) to measure the possibility of a vehicle (an ego vehicle) accepting a merging vehicle. Then, based on SOD, they find drivers' characteristics through experiments and propose a model that controls the merging vehicle behavior so that the ego vehicle will most likely accept it merging.

3 Experiment

The experiment is conducted with SUMO and used for simulating the two-to-one scenario with different lane-merging and lane-changing strategies. The scenario has two lanes, $lane_A$ and $lane_B$. Vehicles starting from $lane_B$ are able to change their lanes to $lane_A$ at Changing-point. A lane-changing strategy will decide whether and when a vehicle changes its lane. Finally, vehicles arrive at Merging-point, and a lane-merging strategy will determine the passing order.

3.1 Setup

There are four connected roads in the scenario, which are $Road_{pre}$, $Road_c$, $Road_m$, and $Road_{end}$ from left to right. $Road_{pre}$ and $Road_{end}$ are only for a better look. $Road_c$ and $Road_m$ are where vehicles change and merge their lanes, and the lengths are both 250 meters. For readability, we say $Road_c$ is from position 0 to position 250, and $Road_m$ is from position 250 to position 500. Changing-point and Merging-point are set at position 247 and position 497, which are 3 meters before the end of $Road_c$ and $Road_m$ respectively. The scenario is shown in Figure 1.

The safe distance, d_s, is used for changing lanes. A vehicle on $lane_B$ is able to change its lane only when no vehicles on $lane_A$ behind it are d_s meters closer to it. $W^=$ and W^+ are the minimal waiting times (time gaps) for two vehicles from the same lane and two vehicles from different lanes to merge.

There is only one type of vehicles. To be best close to reality, the max speed of a vehicle is 14 m/s (= 50.4 km/hr), and a vehicle will always drive at the max speed if possible. The car-following model of the vehicle is set to default, which is carFollowing-Krauss in SUMO. A parameter in the car-following model, tau, is set to default, 1 (sec), which means that a vehicle always wants to maintain 1 second time headway between the vehicle right in front of it.

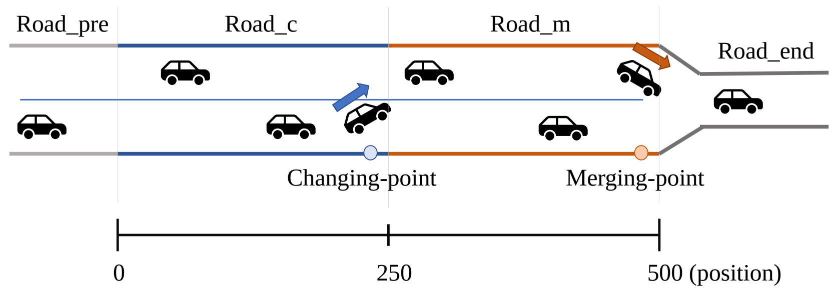

Figure 1: Road

However, some parameters cannot be close to reality. The acceleration and deceleration are set to 5 and 7 m/s^2, which are way too high compared to reality. One of the reasons is that if they are close to reality, all vehicles play extremely safe, which increases the costs of changing and merging lanes, and leads to bad results. However, in reality, if some vehicles play safe, others will take advantage and perform riskily, which will speed up the traffic. Another reason is that instead of slowing down, there should be more changing lanes at more positions because the gaps between vehicles are bigger. The last reason is that low acceleration and deceleration increase the traffic wave and make the simulation complicated. The error will increase when implementing different strategies because it is hard to track and predict all vehicles' behaviors. Due to the same reasons, sigma, a parameter that determines the imperfection of a vehicle following the car-following model, is set to 0. Since the acceleration and deceleration are high, d_s are set to 28 m (twice the value of the max speed), which is lower than reality.

3.2 Strategy

There are three lane-changing strategies implemented in the experiment, No-change, All-change, and Safe-change.

No-change simply means there is no changing lane in the simulation. All-change means that all vehicles starting from $lane_B$ will change their lanes. Safe-change means that there is no waiting on $lane_A$. As soon as a vehicle on $lane_B$ arrives at Changing-point, it changes its lane if vehicles on $lane_A$ don't have to wait for it, otherwise, it won't.

There is only one lane-merging strategy implemented in the experiment, FIFO. The earlier the vehicle arrives in $road_m$, the earlier the vehicle merges.

Note that the strategies process BEFORE the vehicles drive instead of DURING the driving. For example, when each vehicle arrives in $road_m$ is predicted based on their arrival times (the time when a vehicle is at position 0), and a passing order will then be generated based on FIFO before they actually arrive in $road_m$. Therefore, a bad prediction model will lead to a wrong and unexpected passing order and result. In the experiment, the prediction model is simply assuming that vehicles always drive at the max speed. Thus, a good setup of parameters is important to keep the error lower.

3.3 Simulation and Result

Python and Traci, a module for controlling every time slice of a simulation, are used to run the simulations. A time slice means 0.1 seconds. A given number of vehicles whose arrival times are randomized will be generated initially on both lanes. Two adjacent vehicles on the same lane have at least 1 second gap and at most 7 seconds gap between their arrival times.

To show the effects of different algorithms, simulations are run with different $W^=$ and W^+, and the average running times (sec) are shown in Table 1 and Table 2. The scenario in Table 1 has 4 vehicles on each lane, and the scenario in Table 2 has 10. Figure 2 and Figure 3 are the line charts of Table 1 and Table 2. For every $W^=$ and W^+, 10 inputs are generated and simulated with different algorithms. Note that $W^=$ is set to 14 time slices (= 1.4 sec) and W^+ is set to one to three times the value of $W^=$.

The result shows that when W^+ is close to $W^=$, No-change and Safe-change's average running times are almost the same, and All-change's is higher than theirs. As W^+ increases, approximately, No-change and Safe-change's average running times increase linearly, and All-change's remains the same. When W^+ is more than 1.8 times the value of $W^=$, All-change's average running time is the lowest, the next is Safe-change's, and the last is No-change. It also shows that the effects of different algorithms of the 10-vehicle scenario are more noticeable and more consistent than the ones of the 4-vehicle scenario since the chart in Figure 3 is steadier than the chart in Figure 2.

The reason for the result is that changing lanes has cost, but changing lanes may lower the running time because more vehicles will be on the same lane. When W^+ is close to $W^=$, changing lanes barely benefits, so All-change is not a good strategy. As W^+ increases, changing lanes benefits more so All-change has a lower average running time.

3.4 Conclusion

In the experiment, we can find that changing lanes has the potential to speed up a lane-merging scenario. However, it may be complicated if we want to put it into reality. First, a good prediction model if we want to process before the whole event. Second, a good algorithm to calculate the cost and the benefit of changing lanes. Third, how to decrease or take advantage of the traffic wave. Last but not least, safety concern, which is overlooked in this experiment. The concept is worth exploring and improving.

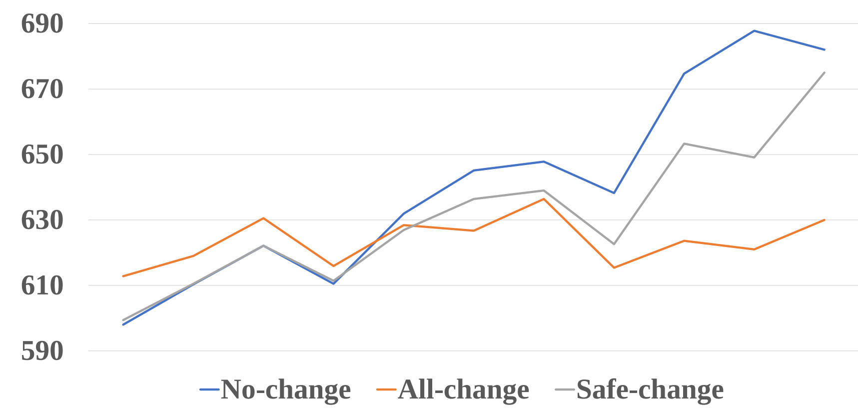

Figure 2: Running times (in seconds) when 4 vehicles are on each lane

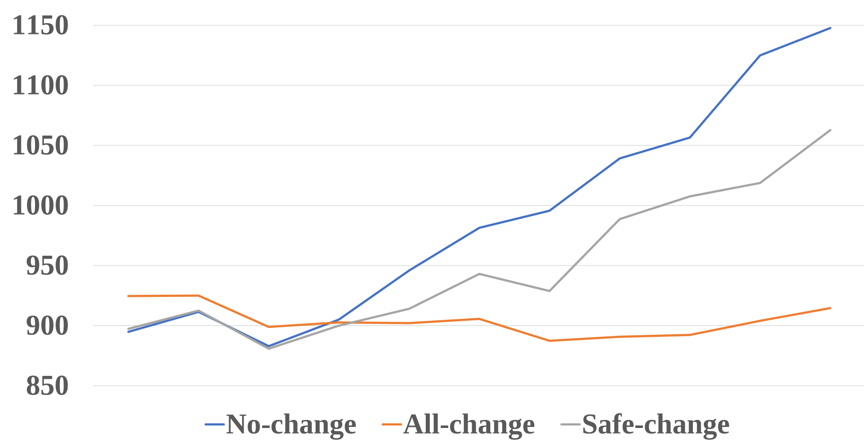

Figure 3: Running times (in seconds) when 10 vehicles are on each lane

Table 1: Running times (in seconds) when 4 vehicles are on each lane

$W^{=}, W^{+}$	No-change	All-change	Safe-change
14, 14	598	612.8	599.4
14, 16.8	610.3	619	610.5
14, 19.6	622.1	630.5	622.1
14, 22.4	610.5	615.9	611.4
14, 25.2	631.9	628.4	626.9
14, 28	645.1	626.7	636.4
14, 30.8	647.8	636.4	639
14, 33.6	638.2	615.4	622.6
14, 36.4	674.7	623.6	653.3
14, 39.2	687.8	621	649.1
14, 42	682	630	675

Table 2: Running times (in seconds) when 10 vehicles are on each lane

$W^{=}, W^{+}$	No-change	All-change	Safe-change
14, 14	894.8	924.6	897.3
14, 16.8	911.4	925	912.5
14, 19.6	882.9	898.9	880.8
14, 22.4	905	902.6	900
14, 25.2	945.9	902.1	914
14, 28	981.4	905.6	943
14, 30.8	995.6	887.4	928.8
14, 33.6	1039.2	890.7	988.7
14, 36.4	1056.5	892.2	1007.6
14, 39.2	1124.9	904	1018.7
14, 42	1147.7	914.6	1062.7

References

[1] Shang-Chien Lin, Hsiang Hsu, Yi-Ting Lin, Chung-Wei Lin, Iris Hui-Ru Jiang, and Changliu Liu. "A Dynamic Programming Approach to Optimal Lane Merging of Connected and Autonomous Vehicles". *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[2] Zhuoqun Wu, Xudong Liu, and Liguo Zhang. "Decentralized Optimal Control of Connected and Automated Vehicles at Merge Areas". *40th Chinese Control Conference (CCC)*, 2021.

[3] Jinxian Weng, Shan Xue, and Xuedong Yan. "Modeling Vehicle Merging Behavior in Work Zone Merging Areas During the Merging Implementation Period". *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[4] Sercan Akti, Ismet Goksad Erdagi, Mehmet Ali Silgu, and Hilmi Berk Celikoglu. "A Game-Theoretical Approach for Lane-Changing Maneuvers on Freeway Merging Segment". *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[5] Na Chen, Bart van Arem, and Meng Wang. "Hierarchical Optimal Maneuver Planning and Trajectory Control at On-Ramps With Multiple Mainstream Lanes". *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[6] Gihyeob An and Alireza Talebpour. "Vehicle Platooning Control for Merge Coordination: A Hybrid ACC-DMPC Approach". *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

[7] Hiroyuki Okuda, Kota Harada, Tatsuya Suzuki1, Shintaro Saigo, and Satoshi Inoue. "Design of Automated Merging Control by Minimizing Decision Entropy of Drivers on Main Lane". *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[8] Li Li and Dong Zhang. "Spatial Characteristics of Merging Decision Making and Implementation at Highway Work Zone". *4th International Conference on Transportation Information and Safety (ICTIS)*, 2017.

[9] David Gundana, R. Austin Dollar, and Ardalan Vahidi. "To Merge Early or Late: Analysis of Traffic Flow and Energy Impact in a Reduced Lane Scenario". *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[10] Gen Li, Yiyong Pan, Zhen Yang, and Jianxiao Ma. "Modeling Vehicle Merging Position Selection Behaviors Based on a Finite Mixture of Linear Regression Models". *IEEE Access*, 2019.

[11] Hiroyuki Okuda, Tatsuya Suzuki, Kota Harada, Shintaro Saigo, and Satoshi Inoue. "Quantitative Driver Acceptance Modeling for Merging Car

at Highway Junction and Its Application to the Design of Merging Behavior Control". *IEEE Transactions on Intelligent Transportation Systems*, 2019.