Cooperative Highway Merging for Heterogeneous Autonomous Traffic

Nilesh Suriyarachchi
nileshs@umd.edu

Department of Electrical and Computer Engineering
and the Institute for Systems Research
University of Maryland - College Park, MD, USA
Introduction - Motivation

- The highway on-ramp merging bottleneck
  - Reduced Throughput (Mean Velocity)
  - Increased Delay
  - Increased Fuel Consumption

- Heterogeneous Traffic
  - Different vehicles → Diverse needs

- Connected Autonomous Vehicles (CAVs)
  - Enhanced sensing
  - Enhanced communication
    - Vehicle to Infrastructure (V2I) Communication
    - Vehicle to Vehicle (V2V) Communication
1. Optimal closed-form solution to centralized problem

2. Rule-based sequence and velocity assignment

3. Virtual slot-based allocation method

- Other methods exist to solve the optimal control problem but don’t have real-time solutions
- None of these methods handle heterogeneous traffic

Problem Definition

Vehicle Model

- Assume existence of a low level controller for lane keeping and longitudinal actuation \( (w_i) \)
  \[
  \dot{s}_i = f(t, s_i, w_i), \quad s_i(t_i^0) = s_i^0
  \]

- High level velocity control scheme
  \[
  \dot{s}_i = v_i \quad v_i(t) = u_i(t) \quad \text{for } i \in \{1, \ldots, n\}
  \]

- For each vehicle \( i \),
  \[
  p^i(t) \in \mathbb{R}_+^2, \quad b^i \in \{0, 1\}
  \]

- Full CAV State:
  \[
  x_i(t) = [s_i(t), v_i(t), p^i(t), b^i, l^i, a_{i\text{max}}^i, a_{i\text{min}}^i]^T
  \]

Output: Compute optimal velocities \( u_i \) and merging sequence \( q \)
Handling heterogeneous traffic

- Priority assignment based on:
  - Vehicle type, size and mass
  - Emergency vehicle
  - On-ramp or Mainline vehicle
  - Current speed
  - Vehicle’s future intent
  - Waiting time

- Types of priority:
  - Speed Prioritization $p_s^i$
  - Speed Variation Prioritization $p_v^i$

- Note: Consider flow over effect for high priority vehicles in the back of the queue
Optimal merging control formulation

- Maximize throughput (Mean Velocity) while minimizing control \((u_i)\) applied

\[
\min_{\{u_i,q\}} \sum_{i=1}^{n} p_s^i \lambda (u_i - \bar{v})^2 + p_v^i (1 - \lambda) (u_i - v_i)^2
\]

\[s.t. \quad C(\{x_i\}, \bar{v}, L_m, L_r, M_s)\]

- However, this optimization problem is practically intractable
  - Constraint set \(C\) contains constraints with respect to the order of the vehicles
  - The solution to this problem, also yields the optimal sequence \(q\)
  - Problem becomes too complex to be solved in real time

- What do we do?
  - Problem can be reformulated as a mixed-integer quadratic optimization problem
    - Solved by iterating over all possible merging sequences \(q\) with exhaustive search
  - This essentially decouples the problem into:
    - Generating the merging sequence
    - Computing the optimal velocity commands \(u_i\) to achieve merging sequence \(q\)
Sequence Generation

- The problem:
  - Total number of possible sequences grows exponentially with the number of vehicles in the control zone
  - For $n=30$ ($m=15$, $r=15$),
    \[
    \frac{(m + r)!}{m!r!} = 155117520
    \]
  - Number of sequences limited by:
    - Length of the control zone
    - Requirement for FIFO order on each lane

- Our approach:
  - Rule based heuristic to select sequences $q$ that adequately represent the search space
    - The optimal sequence lies in between the extremes of,
      - Distance to merge ($d_M$)
      - Time to merge ($t_M$)
    - Inject ramp vehicles into the mainline queue while maintaining precedence order
      - Insertion based on following logic:
        \[
        (1 - \alpha) \cdot t_M^i + \alpha \cdot d_M^i, \quad \forall i \in \{1, \ldots, n\}
        \]
Proposed method* Pipeline

Optimal Velocity Computation - Objective function

- Objective function to obtain control $u_i$ for a given merging sequence $q$:

$$J(u_i|q) = \sum_{i=1}^{n} p_s^i \lambda (u_i - \bar{v})^2 + p_v^i (1 - \lambda)(u_i - v_i)^2$$

- Minimize control effort and maximize throughput
  - For a fixed density ($m_D$),
    Throughput ($m_T$) is maximized when Mean velocity ($m_V$) is maximized
    $$m_T = m_V * m_D$$

- Incorporating the variable $b^i$ into priority $p^i$ the problem reduced from MIQP to QP
Optimal Velocity Computation - Constraints - I

- Constraints ensure reachability and safety

- Control $u_i$ is bounded based on:
  - Speed limits on the highway
    \[ 0 \leq u_i(t) \leq \bar{v} \]
  - Acceleration capabilities of vehicles
    \[ a_{min}^i \Delta t \leq u_i(t) - v_i(t) \leq a_{max}^i \Delta t \]
    - $\Delta t$ is based on the time resolution of computations
Optimal Velocity Computation - Constraints - II

- Compute expected position of vehicle $i$ after time $\Delta t$

\[ s_i(t + 1) = s_i(t) - \Delta t \frac{v_i(t) + u_i(t)}{2} \]

- Rear-end collisions are then avoided by:

\[
|s_j(t + 1) - s_{j'}(t + 1)| \geq l^j + M_{sr} \quad \forall j, j' \in \{1, \ldots, m\} \\
|s_k(t + 1) - s_{k'}(t + 1)| \geq l^k + M_{sr} \quad \forall k, k' \in \{1, \ldots, r\}
\]

- Substituting expected position and using FIFO precedence logic

\[
\begin{align*}
    u_j - u_{j+1} &\geq (v_{j+1} - v_j) + \frac{2}{\Delta t} (s_j - s_{j+1} + l^j + M_{sr}) & \forall j, j + 1 \in \{1, \ldots, m\} \\
    u_k - u_{k+1} &\geq (v_{k+1} - v_k) + \frac{2}{\Delta t} (s_k - s_{k+1} + l^k + M_{sr}) & \forall k, k + 1 \in \{1, \ldots, r\}
\end{align*}
\]
Optimal Velocity Computation - Constraints - III

- Compute time to merge $t^i_m$ of vehicle $i$
  \[
  t^i_m = \frac{s_i(t+1)}{u_i}
  \]

- Lateral collisions in merge zone are then avoided by,
  \[
  t^i_m \leq t^{i+1}_m - \frac{l^i}{u_i} - \frac{l^{i+1}}{u_{i+1}} - M_{sl} \quad \forall i, i + 1 \in q
  \]

- Substituting expected position and simplifying we get,
  \[
  u_{i+1}(s_i - \frac{\Delta t}{2}v_i + l^i + M_{sl}) \leq u_i(s_{i+1} - \frac{\Delta t}{2}v_{i+1} - M_{sl}) \quad \forall i, i + 1 \in q
  \]

- This constraint also ensures merging order of sequence $q$ is followed
Optimal Sequence Selection and Low-level Control

- Run the optimization for each sequence in parallel
- Obtain the sequence $q$ for which the minimum objective value was achieved
- Provide the command velocities $u_i$ corresponding to $q$, to the low-level controller
  - Zero communication delay is assumed for this V2I transmission
- Low-level controller executes the command
  - Compute acceleration commands to achieve desired command velocity
- Note: This method allows merging sequence to be changed at every update cycle
  - Essential feature for heterogeneous traffic
    - If an EV enters the control zone it should be given precedence
Proposed method* Pipeline

Experimental Setup - Simulation

- Simulation Software: SUMO
- Interface platform: TraCI
- Optimization Software: Gurobi (Version 9.1.1)
- Circular loop structure used to simulate a continuous stream of vehicles with constant ratios between vehicle classes
- The percentage of mainline vehicles to ramp vehicles is easily adjusted

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control zone length</td>
<td>300 m</td>
</tr>
<tr>
<td>Simulation duration</td>
<td>2 hrs</td>
</tr>
<tr>
<td>Time step ($\Delta t$)</td>
<td>100 ms</td>
</tr>
<tr>
<td>Maximum number of sequences</td>
<td>12</td>
</tr>
<tr>
<td>Speed limit ($\bar{v}$)</td>
<td>27 m/s</td>
</tr>
<tr>
<td>Mainline-Ramp rerouting ratio ($r_p$)</td>
<td>0.5</td>
</tr>
<tr>
<td>Objective trade-off ($\lambda$)</td>
<td>0.7</td>
</tr>
<tr>
<td>$M_{ST}$</td>
<td>2.0</td>
</tr>
<tr>
<td>$M_{sl}$</td>
<td>10.0</td>
</tr>
</tbody>
</table>
Proposed Method Simulation Video
Results: Methods Compared

- Compare against other real-time capable methods
  - Baseline Method
    - No cooperation between vehicles
    - Ramp vehicles give way to mainline vehicles
  - FIFO based Greedy Method
    - Vehicles assigned merging order based on entrance order to control zone
  - Zipper Merge Method
    - Merge sequence based on distance to merge
    - Alternating between ramp and mainline vehicles
  - Our Proposed Method
    - Real-time Priority-based Cooperative Highway Merging for Heterogeneous Autonomous Traffic
Results: Main performance indicators

- Advantages are evident at higher density levels
  - Complex methods unnecessary for low density levels
- Mean velocity improvements at high densities
  - 282% improvement over baseline
  - 81% improvement over zipper merge
Results: Delay faced

- Once again, advantage of the proposed method is evident at higher density levels
  - Cooperative methods perform drastically better
- EVs face 17% less delay than other vehicle classes
- Improvements in delay faced
  - 92% improvement over baseline
  - 58% improvement over zipper merge
Results: Fuel consumption

- Once again, advantage of the proposed method is evident at higher density levels
  - Cooperative methods perform drastically better
- Fuel savings in our method can be further improved with parameter tuning
  - Often at the cost of reduced overall throughput
- Fuel savings for trucks play an important role
Real Time Capability

- Algorithm update frequency: 10Hz (update every 100ms)

- Average time to optimize for one sequence: 25ms
  - Therefore for 12 sequences: 300ms
  - However due to parallelization: 50ms (12 core CPU)

- Algorithm is more than capable of real-time operation

- Note: The quality of the generated control can be improved even further
  - Searching over a larger number of sequences
  - Requires more computation power to maintain real-time operability
Summary

- Introduce a novel Cooperative Highway Merging method
  - Heterogeneous Autonomous Traffic (Priority-based)
  - Parallel Computation Architecture (Real-time Operation)
  - Constrained QP optimization problem

- Performance verified through SUMO simulation
  - Comparison to other real-time capable methods
  - Improvements in throughput, delay faced and fuel consumption

- What’s next?
  - Multi-lane total highway capacity utilization extension
  - Decentralized control with a V2V Communication based formulation
  - RL based approach for the control aspect of this work
Thank You!

Questions?
nileshs@umd.edu