Merging Truck Platooning on Highways Senario Team-B

Mohamed Abdo[1]

Contents

[1] mohamed-sayedmohamed.abdo@stud.hshl.de

Abstract: Truck platooning allows freight vehicles to drive closely together in coordinated groups, improving fuel efficiency and road capacity. One of the most challenging maneuvers is *merging*, especially when a new truck or platoon joins an existing one on a highway. This paper presents the main requirements, communication signals, control strategies, and timing considerations needed for safe and efficient merging. A scenario-based model is proposed, designed for formal verificationwhich we perform using UPPAAL and Timed Automata to model timing behavior and safety constraints.

1 Motivation

Modern transportation is increasingly using cooperative driving technologies to improve fuel efficiency, traffic flow, and safety. One such technology is truck platooning, where trucks travel closely together with coordinated control. While this offers clear benefits, real-world use must handle complex scenarios like merging, splitting, and lane changes. Merging, in particular, is challenging and requires precise coordination between vehicles. Fully automated merging pushes the limits of current algorithms and demands strict safety measures [As23]. This highlights the need for clear merging protocols and system-level models to test and validate merging under different conditions [Zh21]..

2 Scenario Description

In the highway merging context, a truck (or another platoon) approaching via an on-ramp must join an already-formed platoon on the main lane. This operation necessitates the creation of a safe gap within the platoon and the coordination of both lateral and longitudinal dynamics between multiple vehicles [As23].

Maintaining safe and efficient gaps is central to platooning. The inter-vehicular distance should scale with speed and system dynamics. A suggested gap time of 0.8–1.2 seconds allows for aerodynamic benefits while preserving safety [KCS23]. During merging, this gap must be temporarily widened and then re-stabilized post-merge.

A structured **protocol** is required to manage the interaction. A typical approach involves three stages:

1. **Negotiation Phase** – The merging vehicle requests a slot; the platoon assesses feasibility.
2. **Coordination Phase** – The platoon adjusts to open the gap; merge point and time are agreed.
3. **Execution Phase** – The merging vehicle aligns and enters the gap with coordinated acceleration.

3 Requirements for Merging Platoons

3.1 Communication

Effective merging depends on real-time sharing of key data among all involved vehicles. This includes current speed, acceleration, planned trajectory, and merge intent [As23, Zh21]. The communication system must have low latency and high reliability to prevent collisions due to outdated or lost messages.

3.2 Control Cooperation

Vehicles in the platoon must be capable of decentralized decision-making based on shared data, allowing for dynamic adjustment to support the merging operation [ZZ19]. Specifically, the leading or following truck may need to accelerate or decelerate to open a sufficient gap for the merging vehicle.

3.3 Safety Constraints

Safety is paramount. Inter-vehicular distances must always satisfy predefined constraints to prevent rear-end collisions. Emergency braking and fallback strategies must be in place in the event of communication failures or sensor malfunctions [Zh21].

4 Timing Behavior

The merging maneuver must occur within a tight temporal window to avoid disrupting traffic flow or endangering vehicles. Timing constraints include the window during which the merging vehicle is aligned with the platoon, the delay in data transmission, and the response time of the control system [ERES22]. The entire interaction must be bounded within real-time deadlines to preserve the safety margin of all vehicles.

5 Modeling of Merging Scenarios using SysML/UML

In this scenario, we use SysML and UML diagrams to model a truck platoon merging, focusing on system structure, behavior, and interactions. Key diagrams include the Activity Diagram (merging workflow), Block Definition Diagram (system components), and Requirements Diagram (safety and functional needs). Sequence and State Machine Diagrams model vehicle interactions during the merge. The paper also incorporates Timed Automata and uses UPPAAL for formal verification of safety, timing, and synchronization conditions.

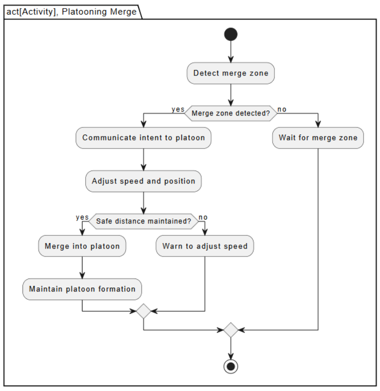

Fig. 1: Activity Diagram

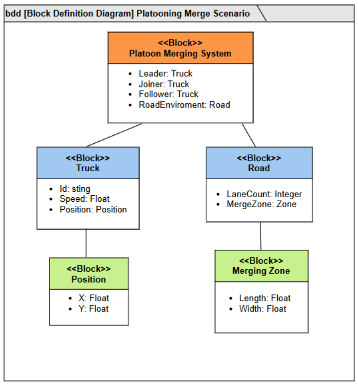

Fig. 2: Block Definition Diagram

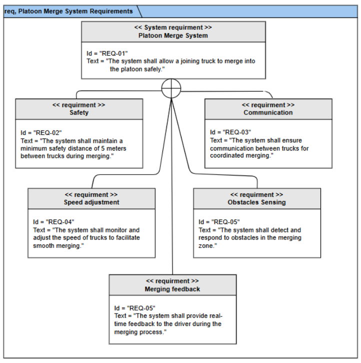

Fig. 3: Requirements Diagram

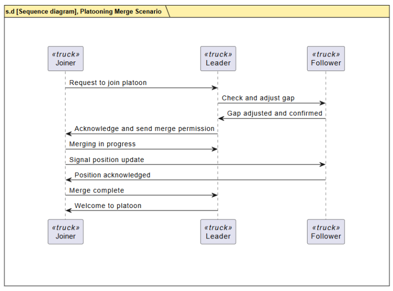

Fig. 4: Sequence Diagram

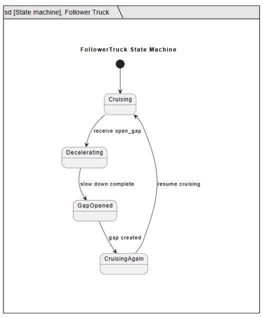

Fig. 5: State Machine Diagram 1

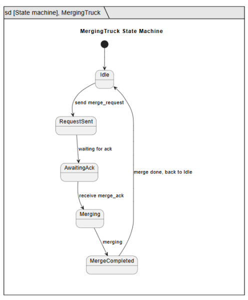

Fig. 6: State Machine Diagram 2

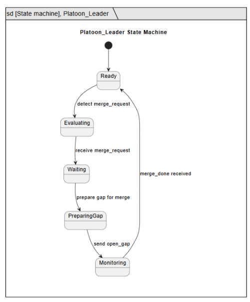

Fig. 7: State Machine Diagram 3

6 Formal Modeling with UPPAAL

We use **UPPAAL** and **Timed Automata** to model the merging protocol. Each vehicle (merging truck, platoon leader, follower) is modeled as an automaton. Synchronizations and timing constraints ensure correct coordination.

We verify: **Safety** (no collisions), **Liveness** (merge completes), and **Timing** (within deadlines). **Start Merge Signal** To synchronize execution, the `MergingTruck` emits `start_merge!`. The `PlatoonLeader` listens for `start_merge?` to switch to monitoring.

Signals / Events Modeled

`merge_request` – `merge_ack` – `open_gap` – `start_merge` – `merge_done`

UPPAAL simulation:

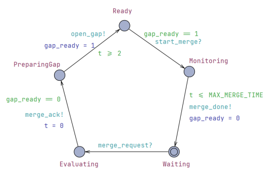

Fig. 8: PlatoonLeader: Coordinates the merge.

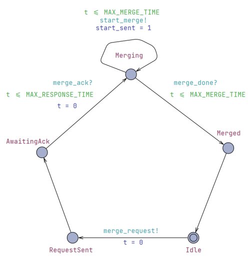

Fig. 9: MergingTruck: Sends request, merges.

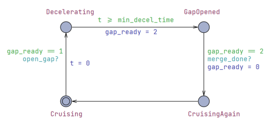

Fig. 10: FollowerTruck: Opens a safe gap.

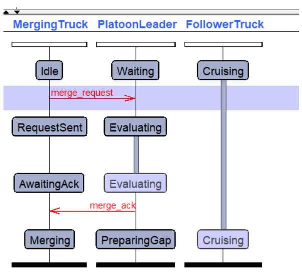

Fig. 11: Request sequence: MergingTruck initiates the merge.

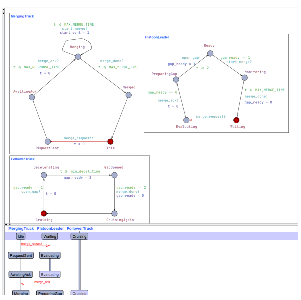

Fig. 12: Overview: All vehicles transition.

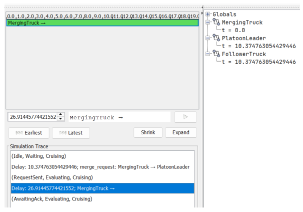

Fig. 13: Timing behavior: Clocks and guards.

Identify Relevant Control Behavior for the Merging Scenario

- Sending merge requests.
- Evaluating if merging is feasible.
- Opening a gap (follower truck slows down).
- Starting the merge (synchronizing via `start_merge` signal).
- Monitoring merge completion.

Guarantee Distance to the Preceding Truck for Safety and Spacing During Merging

- Use a clock (`t`) measuring the time since the gap opening started.
- Guards that prevent transitions until this time passes, Ex:$t \geq min\_gap\_time$.
- Variables (like `gap_ready`) to represent safe spacing state.

Handle Communication Failures (Robustness & Stability)

If no `merge_ack` is received within `MAX_RESPONSE_TIME`, the merging truck returns to idle or retries.

If the leader does not receive a `start_merge` signal within the expected time, it can abort or reset.

- Use clock guards, Ex:`t > MAX_RESPONSE_TIME`.
- Ensure the system does not deadlock.

The system eventually either completes the merge or safely aborts.

7 Conclusion

Merging in truck platooning presents significant challenges requiring coordinated control, real-time communication, and strict adherence to timing constraints. A formal model, such as a network of timed automata, can be used to verify the correctness of merging protocols under worst-case scenarios using tools like UPPAAL. Future work should focus on integrating sensor redundancy and infrastructure assistance to further enhance safety and scalability.

8 Declaration of Originality

I am Mohamed Abdo, herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker.

03/04/2025 Lippstadt Mohamed Abdo

Bibliography

[As23] Asadi, Mohammad et al.: Safe and efficient cooperative on-ramp merging using connected and automated vehicles. IEEE Transactions on Intelligent Vehicles, 8(1):122–135, 2023.

[ERES22] Elbery, Ahmed; Rakhshan, Ali; El-Sherbeeny, Yasser M: Model predictive control-based merging maneuver for connected and automated vehicles at highway on-ramp. IEEE Transactions on Intelligent Vehicles, 7(1):54–63, 2022.

[KCS23] Karamati, Mahdieh; Chen, Yifan; Shahbakhti, Mahdi: A review on inter-vehicular gap control in truck platooning systems. Sensors, 23(1):1–24, 2023.

[Zh21] Zhang, Yifan et al.: Cooperative on-ramp merging for connected and automated vehicles using decentralized control. Transportation Research Part C: Emerging Technologies, 133:103413, 2021.

[ZZ19] Zhou, Yiheng; Zhao, Ding: Distributed cooperative merging control of connected and automated vehicles. Transportation Research Part B: Methodological, 128:215–241, 2019.