Towards effective V2X maneuver coordinations: state machine, challenges and countermeasures

Rafael Molina-Masegosa[a], Sergei S. Avedisov[b], Miguel Sepulcre[a],
Javier Gozalvez[a], Yashar Z. Farid[b], Onur Altintas[b]
[a] Universidad Miguel Hernandez de Elche, Elche, Spain, {rafael.molinam, msepulcre, j.gozalvez}@umh.es
[b] Toyota Motor North America R&D – InfoTech Labs, Mountain View, CA, USA, {sergei.avedisov, yashar.zeiynali.farid, onur.altintas}@toyota.com

Abstract— **Connected and automated vehicles can leverage V2X communications to coordinate their maneuvers. Maneuver coordination is expected to improve traffic efficiency and safety, but the design of maneuver coordination is a challenging task in complex traffic scenarios, as maneuvers affect not only the involved vehicles but also nearby traffic. This study introduces a reference state machine for the design of maneuver coordination. Furthermore, we identify and analyze the challenges that maneuver coordination may encounter. We quantify the relevance of each challenge and propose a set of countermeasures to enhance the robustness and effectiveness of maneuver coordination.**

Keywords— maneuver coordination, intent sharing, cooperative driving, connected and automated vehicles, CAV, V2X, MCM, MSCM, vehicular networks.

I. INTRODUCTION

Connected and Automated Vehicles (CAVs) will utilize V2X (vehicle-to-everything) communications to share information, enhancing both traffic safety and efficiency. Through V2X, CAVs can anticipate and coordinate driving maneuvers. This coordination typically involves multiple vehicles, where one vehicle initiates the maneuver request (e.g. a lane change). Upon receiving this request, the target vehicle(s) may adjust their speed or trajectory to accommodate the maneuver. Effective V2X-based maneuver coordination can enhance traffic flow and safety by enabling smoother interactions between vehicles. However, this process is complex and critical, as it not only affects the immediate planning and control of the vehicles involved in the maneuver but also has implications for nearby traffic [1]. A robust and reliable design for maneuver coordination is essential to minimize risks and ensure seamless cooperation among vehicles.

The survey reported in [2] categorizes the approaches to cooperative maneuvers into three types: implicit coordination, explicit coordination, and coordination without communication. The latter case involves predefined control logic installed in each vehicle. In implicit coordination, vehicles communicate their driving intents (such as planned and desired trajectories). The coordination of maneuvers occurs implicitly when other vehicles adjust their driving behavior in response to the shared intents from an ego vehicle, effectively accepting the maneuver request without direct negotiation. Explicit coordination, on the other hand, involves vehicles explicitly announcing and negotiating their maneuvers. This method requires communication between vehicles to reach an agreement on how to execute specific maneuvers. Various explicit coordination strategies have been proposed. For example, the study reported in [3] proposes seven distinct message types for announcing, negotiating, and coordinating maneuvers between vehicles. The study in [4] addresses scenarios involving multiple cooperating vehicles and cascading maneuvers, and [5] introduces a mechanism for prioritizing safety-critical maneuvers. These studies highlight the potential of maneuver coordination and cooperative driving to enhance traffic safety and efficiency, and efforts to standardize maneuver coordination are underway in both Europe [6] and the USA [7]. However, designing robust and effective maneuver coordinations is still in its early stages. The complexity of traffic interactions presents challenges that can affect the execution and overall effectiveness of coordinations. In this context, this paper advances the state-of-the-art by identifying, analyzing and quantifying the challenges that maneuver coordination may encounter. To this aim, we introduce a reference state machine for maneuver coordination, aligned with the Maneuver Sharing and Coordination standard SAE J3186 [7]. This state machine serves as a blueprint for evaluating the effectiveness and robustness of maneuver coordination strategies and identifying potential challenges and outcomes during coordination. We quantify the relevance and magnitude of each of these challenges and propose a set of possible countermeasures with the objective to enhance the robustness and efficiency of maneuver coordination.

II. MANEUVER COORDINATION DESIGN

A. State Machine

Fig. 1 presents the state machine designed in this study to implement maneuver coordinations. The model outlines all possible states, the transitions between them, and the messages exchanged at each state. Our implementation is compliant with the general framework outlined by SAE in [7], and follows the approach presented in [8] where vehicles exchange their planned and desired trajectories to implicitly coordinate maneuvers. The planned trajectory represents the driving intentions of a vehicle in the short term. The desired trajectory is the trajectory that a vehicle would like to follow, but cannot follow because it overlaps with the planned trajectory of another vehicle that has the right of way. The state machine represents the interactions between two vehicles: a host vehicle (HV), which initiates the maneuver coordination request, and a remote vehicle (RV), which, upon agreeing to collaborate, adjusts its trajectory to enable the requested maneuver. The initial state is referred to as *Intent Sharing*. In this state, a vehicle is not engaged in any coordination and solely transmits *Intent* messages. These messages are broadcasted to neighboring vehicles, conveying the intended planned trajectory for the upcoming seconds and the transmitting vehicle's identification (ID). We follow the message generation rules proposed in [9] to decide when and

.

how these messages should be generated. In particular, we follow the *Tracking Trajectories* proposal, where vehicles generate new *Intent* messages when their planned trajectory has significantly changed or when their last *Intent* message was transmitted over a second ago. This approach efficiently supports coordinations while controlling the channel load.

If a vehicle that is in the *Intent Sharing* state decides to initiate a maneuver coordination, thereby becoming the HV, it transitions to the *HV Negotiation* state. In this state, the vehicle requests a coordination to the intended RV. To do so, it sends *Request* messages with minimum period (100 ms) following [9] until it either receives a *Response* message from the intended RV accepting the requested maneuver coordination or the *Negotiation Timeout* expires. This timeout is a time limit to the negotiation process, and it is currently configured as a fixed time limit from the *Coordination Triggering* (*CT*) time, which is the moment when HV decides to request a maneuver coordination. If the *Negotiation Timeout* expires, HV abandons the current maneuver negotiation process and returns to the *Intent Sharing* state, where it may attempt to initiate another coordination or not. If the HV receives a *Response* message from the RV indicating that it accepts the maneuver coordination request, the HV transitions to the *HV Execution* state as the negotiation is considered complete, and the HV focuses on executing the maneuver. Throughout the execution phase, the HV remains in the *HV Execution* state continuously broadcasting *Confirmation*[1] messages with minimum period. This message is sent by the HV to confirm to the RV the reception of its *Response* message and to inform that it is executing the maneuver. The HV concludes the execution of the maneuver coordination process when the maneuver has been successfully completed or an *Execution Timeout* expires. This timeout sets a time limit for executing the maneuver. To configure the *Execution Timeout*, the HV first estimates a *Coordination intended finish* (*CIF*) time, which is when the maneuver coordination is expected to be completed according to the maneuver initially planned by the HV through its internal logic. However, such planning may not be accurate in complex scenarios [10]. Therefore, instead of setting the *Execution Timeout* precisely at the *CIF* time, a predefined margin is added to this *CIF* time to calculate the *Execution Timeout*. This additional margin provides resilience against inaccuracies. After the maneuver execution phase is completed, the HV returns to the *Intent Sharing* state. We should highlight that the *Request* message always includes the *CT* time and the *CIF* time so that the RV can synchronize its timeouts (*Negotiation Timeout* and *Execution Timeout*) with those of the HV. In addition, the *Request*, *Response* and *Confirmation* messages also include the IDs of the HV and the RV as well as the ID of the maneuver that is being negotiated.

If a vehicle that is in the *Intent Sharing* state receives a *Request* message from an HV, it must decide whether to become the RV for the requested coordination. If the intended RV decides to decline the request, it remains in the *Intent Sharing* state but sends its next *Intent* message with minimum interval (100 ms) so that the HV is aware as soon as possible that it is not accepting HV's maneuver coordination request. The intended RV will send its following *Intent* messages following the message generation rules. On the other hand, if the vehicle accepts the request for coordination, it transitions to the *RV Negotiation* state. In this state, the RV sends *Response* messages with minimum interval to HV until it receives a *Confirmation* message from the HV or the *Negotiation Timeout* is reached. If the *Negotiation Timeout* is reached, the vehicle returns to the *Intent Sharing* state. If a *Confirmation* message is received, the RV transitions to the *RV Execution* state as it considers the negotiation complete and begins executing the requested maneuver while sending *Intent* messages following the message generation rules. The RV deems the execution complete when it either receives an *Intent* message from the HV (which implies that HV left the *HV Execution* state) or the *Execution Timeout* is reached. In either case, the RV returns to the *Intent Sharing* state.

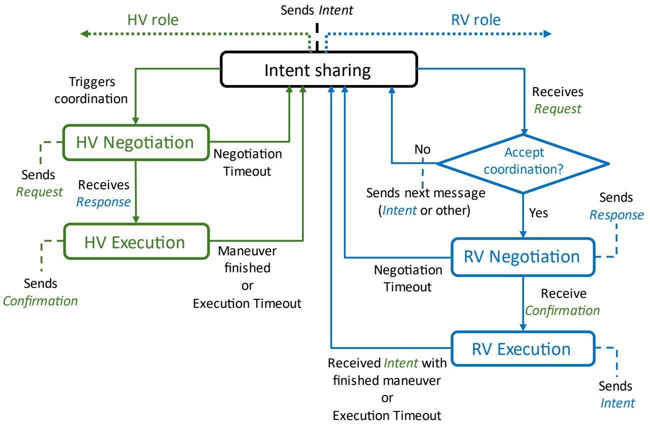

Fig. 1. State machine model for maneuver coordination.

B. *Possible Outcomes of a Maneuver Coordination*

A maneuver coordination can conclude in several ways. The coordination may proceed seamlessly, and the maneuver be executed successfully. However, both the negotiation and execution phases can fail due to several causes and the maneuver will not be completed. Following our state machine model, Fig. 2 illustrates a taxonomy of all possible outcomes of a maneuver coordination. The figure depicts a flow diagram showing how each outcome is reached. The possible outcomes can be classified into three categories: successful coordination, unsuccessful negotiation or unsuccessful execution. In the case of a successful coordination (SC), the maneuver is completed as intended by the HV when triggering the coordination. An unsuccessful negotiation (UN) may be due to several reasons:

- UN1: none of the *Request* messages sent by HV are received by RV before the *Negotiation* timeout, leading HV to abort the negotiation.
- UN2: none of the *Response* messages sent by RV to accept the coordination are received by HV before the *Negotiation Timeout*. Both vehicles abort the negotiation.
- UN3: none of the *Confirmation* messages sent by HV are received by RV before the *Negotiation Timeout*. RV aborts the negotiation, and HV aborts the execution because RV does not execute the maneuver necessary for HV to execute its desired trajectory.
- UN4: the RV does not accept a *Request* from HV because it is already involved in another coordination as HV.

---

[1] The confirmation message in our implementation has a similar purpose than the *Maneuver Reservation* message defined in SAE [7].

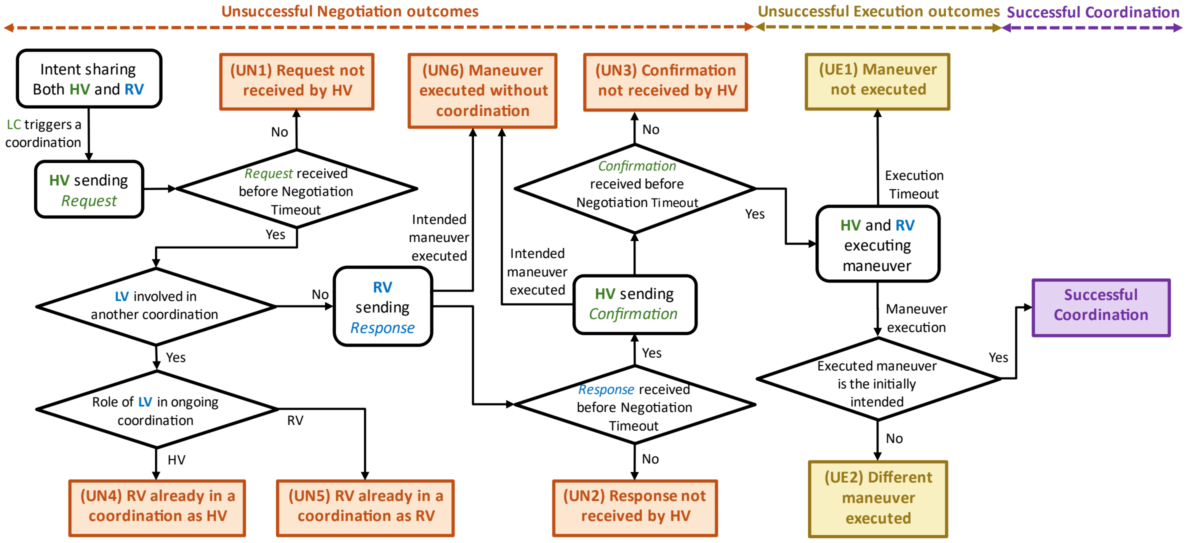

Fig. 2. Possible outcomes of a maneuver coordination.

- UN5: the RV does not accept a *Request* from HV because it is already involved in another coordination as RV.
- UN6: HV executes the intended maneuver without coordination before the negotiation ends. This can occur if traffic conditions differ from the initial plan, and the HV can perform the maneuver without coordination.

An unsuccessful execution (UE) may be due to:

- UE1: the negotiation is successful, but the execution of the intended maneuver cannot be completed before the *Execution* timeout due to traffic conditions differing from those initially expected when planning the maneuver.
- UE2: the negotiation is successful, but HV decides to perform a different maneuver due to traffic conditions differing from those initially expected.

III. NUMERICAL EVALUATION

In this section we simulate maneuver coordinations according to the proposed state machine (Fig. 1) to quantify and understand the possible maneuver coordination outcomes. To this aim, we apply the design of maneuver coordinations represented in Fig. 1 to coordinated lane changes. In a coordinated lane change, the RV performs a controlled deceleration to provide HV with sufficient space to change lanes safely [1]. Vehicles exchange *Intent* messages following the *Tracking Trajectories* message generation rules presented in [9]. These rules adapt the transmission interval between *Intent* messages (between 100 ms and 1 s) depending on how much the vehicle's trajectory deviates from its previously shared trajectory. *Request*, *Response* and *Confirmation* messages are all transmitted with the minimum period of 100 ms. The time interval between the *CT* time and the *Negotiation Timeout* is set to 1 second so that the negotiation process is robust against communication errors. The time interval between the *CIF* time and the *Execution Timeout* is set to 3 seconds. In this study, the length of the planned and desired trajectories is 5 seconds. The length of the planned and desired trajectories determines the maximum time between the coordination triggering time and the *CIF* time, as the *CIF* time cannot extend beyond the planned and desired trajectory length. We should note that, due to the margin added to the *CIF* time to calculate the *Execution Timeout*, the time between the *CT* time and the completion of the maneuver can extend up to 8 seconds.

The evaluation is done using the simulation platform presented in [1], which integrates traffic and V2X simulations to design and test maneuver coordinations. The platform utilizes the network simulator ns-3 for simulating V2X communications and the VANET Highway Mobility module [11] for modelling the vehicles' mobility. We consider a 5km long highway scenario with six lanes (three in each direction) and periodic boundary conditions. Simulations have been conducted with average traffic densities ranging from 10 to 25 vehicles per kilometer per lane. In these simulations 80% of the vehicles in the scenario are cars and 20% are trucks. Each vehicle has a desired speed that represents the speed it would travel at in an empty scenario. It is randomly determined following a uniform distribution with a possible deviation of ±20% around the average desired speed. The average desired speed is set to 120 km/h for cars and 80 km/h for trucks.

Table I reports the percentage distribution of possible outcomes (Fig. 2) from all triggered maneuver coordinations during the simulations. The table shows that successful coordinations are predominant at lower densities but decrease as density increases. Unsuccessful negotiations primarily occur when RV is engaged in another coordination, while unsuccessful negotiations due to communication errors are minimal under the simulated conditions. We should note that unsuccessful negotiations due to RVs involved in other coordinations increase with the density as the density increases the probability that vehicles in close vicinity want to execute a maneuver coordination at the same time. Furthermore, there is no mechanism to prevent the HV from repeatedly attempting to initiate coordination after an unsuccessful negotiation due to the intended RV being involved in another coordination. Consequently, the HV typically triggers a similar coordination attempt shortly after the aborted negotiation, repeating this process multiple times. This behavior further increases the number of unsuccessful negotiations. The percentage of unsuccessful executions also rises with the density due to the complexity of interactions among vehicles that reduce the likelihood of a vehicle executing a maneuver as initially intended. This is because of a higher number of lane changes and decelerations that are challenging to anticipate accurately when designing and planning a maneuver coordination. This is highlighted by the predominance of the outcome UE1 when quantifying unsuccessful executions. In this case, the HV fails to change lanes before the execution timeout.

Fig. 3 depicts the number of successful and unsuccessful coordinations per vehicle per hour as a function of the density. The figure shows an overall increase in all three outcomes with density, reflecting the growing need for lane change coordinations in denser traffic conditions. Despite the increase in the absolute number of successful coordinations, Table I shows that the percentage of successful coordinations decreases with the density as the traffic complexity increases the percentage of unsuccessful coordinations due to failed negotiations or executions of the maneuvers. The number of unsuccessful negotiations increases with density as more vehicles are engaged in coordinations, and the probability that a vehicle requests a coordination to a vehicle that is already executing one increases. It is also important to highlight that as the density increases, the number of unsuccessful executions surpasses that of successful coordinations due to the complexity traffic interactions that prevent a vehicle executing the maneuver it had initially planned. Improving maneuver coordination and execution not only benefits the vehicles directly involved in the maneuver, but also neighboring vehicles that may be affected by the maneuver (e.g. due to a deceleration from a RV to open a gap for an HV).

TABLE I. PERCENTAGE OF MANEUVER COORDINATION OUTCOMES

Outcome	**Density (vehicles/km/lane)**			
	10	**15**	**20**	**25**
(SC) Successful Coordinations	62.28	54.81	43.05	27.03
(UN) Unsuccessful Negotiation	9.29	8.36	14.21	24.93
(UN1) Failed Request transmission	0.05	0.02	0.07	0.06
(UN2) Failed Response transmission	0.00	0.04	0.01	0.02
(UN3) Failed Confirmation transmission	0.00	0.00	0.00	0.00
(UN4) RV in another coordination as HV	6.62	5.73	7.41	11.75
(UN5) RV in another coordination as RV	2.62	2.57	6.69	13.10
(UN6) HV changes lanes during negotiation	0.00	0.00	0.03	0.00
(UE) Unsuccessful Execution	28.43	36.83	42.73	48.04
(UE1) HV not able to change lanes before timeout	26.36	33.29	38.29	43.11
(UE2) HV changes to another lane than initially intended	2.07	3.54	4.44	4.93

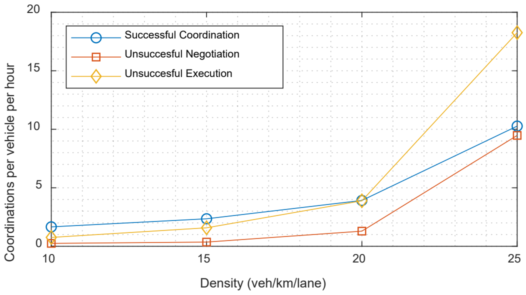

Fig. 3. Number of successful or unsuccessful coordinations per vehicle and per hour.

IV. CHALLENGES AND COUNTERMEASURES

This section analyzes the challenges identified in maneuver coordination that explain the outcomes discussed in Section III and the results depicted in Fig. 3 and Table I. We then propose potential countermeasures, and discuss how they can help address and mitigate the challenges identified.

A. *Challenges*

We have identified the following main five challenges for the design of robust and effective maneuver coordinations.

Challenge 1: HV/RV desynchronization of the execution phase. We have observed the risk that the execution phase completion may not be synchronized between HV and RV. This observation applies to outcome (UE2) but it can even occur in the case of a successful coordination. As shown in Fig. 4, HV finishes its execution when the intended maneuver is completed and then returns to the *Intent Sharing* state, where it begins sending *Intent* messages according to generation rules. These rules might result in that *Intent* messages are not transmitted with the minimum interval, so RV continues to consider the execution ongoing until it receives an *Intent* message from HV (or the *Execution Timeout* is reached). This unnecessarily augments the time RV is in the *RV Execution* state, and hence cannot participate in other maneuvers. This time further augments if the first or initial *Intent* messages sent by HV after execution completion experience any transmission errors. To avoid this challenge, RV must be informed of the end of the execution by HV as promptly and reliably as possible.

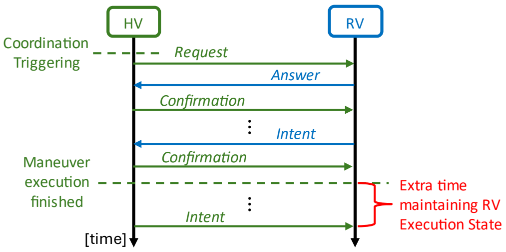

Fig. 4. Desynchronization of the execution phase at the HV and RV.

Challenge 2: RV/HV Desynchronization of a coordination abortion. If an RV does not receive a *Confirmation* message from HV before the *Negotiation Timeout* (Fig. 5), it aborts the coordination process (outcome UN3) and moves to the *Intent Sharing* state. HV is not aware that RV has terminated the coordination process until it receives an *Intent* from RV and remains in *HV Execution* state waiting for the maneuver to be executed by RV. HV keeps transmitting *Confirmation* messages with minimum interval until it receives the first *Intent* message from RV or the *Execution timeout* is reached. To address this challenge, HV must be informed that RV has aborted a coordination as promptly and reliably as possible to avoid unnecessary extra time in the *HV Execution* state. Outcome UN6 occurs when the negotiation is aborted instantly by the HV because unexpected traffic changes have allowed the HV's driving intentions to be fulfilled without needing coordination before the negotiation ends. In this case, the HV aborts the negotiation and returns to the *Intent Sharing* state. However, the RV remains in the *Negotiation* state until it either receives an *Intent* message from the HV or the *Negotiation Timeout* is reached. This situation also represents a desynchronization of a coordination abortion between HV and RV, and in this case, RV must be informed as promptly as possible that HV has aborted the coordination.

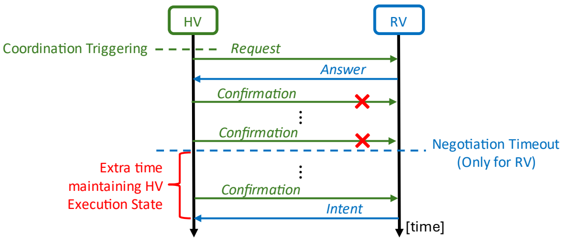

Fig. 5. Desynchronization of a coordination abortion.

Challenge 3: Simultaneous maneuvers. In the case of outcomes (UN4) and (UN5), the intended RV does not accept the coordination because it is already involved in another coordination, as depicted in Fig. 6. The intended RV disregards a *Request* from a different vehicle and continues transmitting messages according to its current state related to the ongoing coordination. It is important that vehicles involved in maneuver coordinations include the ID of the HV, RV and Maneuver in all coordination-related messages to prevent ambiguities between multiple possible maneuvers. The vehicle that requested the coordination will deduce that the RV is not accepting its coordination upon receiving subsequent messages from the RV that do not match the expected *Response* with the specified ID fields. Currently, our implementation does not enforce a waiting interval for vehicles before attempting to coordinate again with the same intended RV. Consequently, vehicles might repeatedly trigger the same maneuver coordination request until traffic conditions change or the intended RV completes its ongoing coordination. This can result in numerous unnecessary coordination requests that could be reduced if the intended RV informs of its ongoing maneuver's *Execution Timeout*. Vehicles that would like to request a maneuver coordination to RV can then refrain from initiating further coordinations until after the *Execution Timeout*.

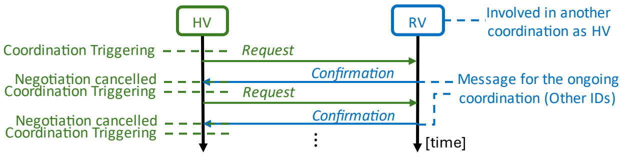

Fig. 6. Simultaneous maneuvers.

Challenge 4: Cancelled maneuver coordinations. In outcome UE1, the maneuver is not finally executed, leading to both HV and RV exiting the *Execution* state due to the *Execution Timeout* (Fig. 7). This occurs when traffic conditions differ from the planned scenario, preventing the maneuver from being executed as initially intended. To reduce the risk of having to cancel maneuvers, it is necessary to accurately estimate traffic conditions (and their evolution) and design robust maneuver coordination protocols. However, it is very challenging to eliminate this risk due to the complexity of traffic interactions, and it would be interesting for vehicles to have the option to cancel the execution of a maneuver coordination before reaching the *Execution Timeout*. This would prevent unnecessary controlled maneuvers, e.g. a prolonged deceleration to create a gap for RV during a lane change maneuver coordination that ultimately gets cancelled, and enable both HV and RV to engage in other coordinations sooner.

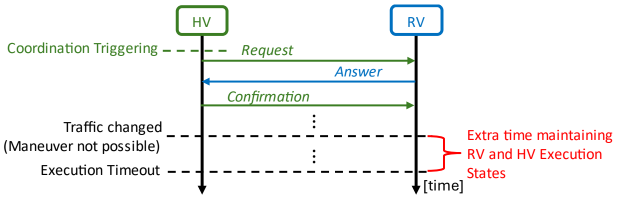

Fig. 7. Cancelled maneuver coordinations.

Challenge 5: Communication errors and channel load. It is possible that several *Request* or *Response* messages may not be correctly received before the *Negotiation* timeout, which impacts outcomes UN1 and UN2. If this is the case, it is likely that the HV will attempt to trigger the coordination again, as its situation may not have significantly changed, which will augment the channel load. The loss of consecutive messages is more likely to happen due to channel congestion than propagation errors since vehicles involved in a maneuver are in close vicinity. While we have not observed this challenge in our simulations, this challenge cannot be overlooked as the channel load can significantly increase with the traffic density and multi-V2X service scenarios where vehicles broadcast different type of messages simultaneously (e.g. cooperative awareness messages, CAMs; basic safety messages, BSMs; collective perception messages, CPMs, etc.).

B. Countermeasures

We propose the following countermeasures to address the challenges previously identified.

Countermeasure 1: Implementation of a *Maneuver Execution Status* message which is sent by RV with minimum interval when it is in the *RV Execution* state instead of sending *Intent* messages. The *Maneuver Execution Status* message would include all the IDs corresponding to this maneuver (HV, RV, Maneuver) and the *Execution Timeout*. This timeout is shared by RV to inform when the execution is going to be finished. With this message, the RV can inform other vehicles interested in a maneuver coordination that it is already involved in a coordination, and it can only get involved in other coordination after the *Execution Timeout*.

Countermeasure 2: RV leaves its *Execution* state if it receives a message from HV other than a *Confirmation* message, which will mean that HV also left its *Execution* state. This reduces the amount of time that RV is unnecessarily in *Execution* state for a maneuver that has already finished. Similarly, HV leaves its *Execution* state if it receives a message from RV other than a *Maneuver Execution Status* message, meaning that RV is not in its *Execution* state anymore.

Countermeasure 3: *Confirmation* messages should also include the *Execution Timeout* to inform other vehicles about the time until which an RV or HV is involved in an ongoing maneuver.

Countermeasure 4: After receiving a *Request*, *Response*, *Confirmation* or *Maneuver Execution Status* message with either the ID of the HV or the RV, the next message is transmitted with minimum interval, even if it is an *Intent*. This is can promptly inform the other vehicle that the ego vehicle considers the execution finished if it has moved to the *Intent Sharing* state.

Countermeasures 5: Implement a *Cancellation* state and a *Cancellation* message for the HV so that it can cancel the execution of a maneuver before the *Execution Timeout*.

Countermeasure 6: Congestion control protocols should reduce the channel load and the probability of communication errors. This should be done without disrupting the execution of a maneuver. In this case, protocols that control the number of messages that each application can generate (e.g. DCC at Facilities layer in the case of ETSI standards) would be more effective than protocols that control the number of messages that a vehicle can transmit per second (e.g. DCC at Access layer in ETSI) [12]. This is the case because the protocols at the access layer can drop messages and alter the execution of a maneuver following the state machine model. On the other hand, the protocols that adapt the messages at the application level do not drop messages but adjust their timing.

We discuss now how each challenge can benefit from the identified countermeasures to increase the robustness and efficiency of maneuver coordinations.

Challenge 1: Desynchronization of the execution phase. The time a RV is unnecessarily in *RV Execution* state after HV has finished a coordination can be reduced by introducing the *Maneuver Execution Status* message (countermeasure 1), requiring HV to send its next message promptly after receiving a *Maneuver Execution Status* message (countermeasure 4), and making RV leave the *RV Execution* state if it receives a message from HV other than a *Confirmation* message (countermeasure 2).

Challenge 2: Desynchronization of a coordination abortion. Requiring HV or RV to promptly send its next *Intent* message after receiving a *Request*, *Response*, *Confirmation* or *Maneuver Execution Status* message (countermeasure 4) minimizes the time in which the vehicle that sent any of those messages is still unnecessarily involved in the coordination. The early transmission of *Intent* must be recognized as an abort signal (countermeasure 2).

Challenge 3: Simultaneous maneuvers. An RV that is engaged in a maneuver does not engage in additional coordinations requested by other vehicles. Including the *Execution Timeout* in *Status* and *Confirmation* messages (countermeasures 1 and 3) informs these other vehicles until when RV is engaged in its ongoing maneuver so that they avoid unnecessarily sending *Request* messages that will be rejected by the target RV until it finishes its ongoing coordination.

Challenge 4: Cancelled maneuver coordinations. Implementing a cancellation process (countermeasure 5) allows for swift maneuver coordination termination in response to changing traffic conditions.

Challenge 5: Communication errors and channel load. The implementation of congestion control protocols at the Facilities layer (countermeasure 6) reduces the probability of consecutive communication errors that disrupt the operation of a maneuver coordination.

V. CONCLUSIONS

This study contributes to the design of robust maneuver coordination by analyzing the challenges that maneuver coordinations may encounter in complex and dynamic traffic scenarios. We introduce a reference state machine model for maneuver coordination that outlines all possible states, transitions between states, and messages exchanged at each state. We then identify a taxonomy of all possible outcomes of a maneuver coordination and determine how each outcome is reached for a successful or unsuccessful coordination. We quantify the relevance and impact of each outcome and identify the main challenges affecting maneuver coordination. Finally, we propose a a set of countermeasures to enhance the robustness and effectiveness of maneuver coordination. Future work will involve testing these countermeasures and integrating them into the design of maneuver coordinations to improve their robustness and effectiveness in complex and dynamic traffic scenarios.

ACKNOWLEDGEMENTS

Universidad Miguel Hernandez de Elche (UMH) work was supported in part by grant TED2021-130436B-I00 funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR, and by UMH through a research grant for the dissemination of science, technology and innovation 2024.

REFERENCES

[1] A. Correa, S.S. Avedisov, M. Sepulcre, A.H. Sakr, R. Molina-Masegosa, O. Altintas and J. Gozalvez, “On the Impact of V2X-based Maneuver Coordination on the Traffic,” *Proc. IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, Online, pp. 1-5, 25-28 April 2021.

[2] B. Häfner, V. Bajpai, J. Ott and G. A. Schmitt, "A Survey on Cooperative Architectures and Maneuvers for Connected and Automated Vehicles," in *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 380-403, First quarter 2022.

[3] M. Mizutani, M. Tsukada, H. Esaki, “AutoMCM: Maneuver Coordination Service with Abstracted Functions for Autonomous Driving”, *Proc. 24th IEEE International Conference on Intelligent Transportation (ITSC)*, Indianapolis, USA, pp. 1069-1076, 19-22 Sept. 2021.

[4] D. Maksimovski and C. Facchi, "Negotiation Patterns for V2X Cooperative Driving: How complex Maneuver Coordination can be?," *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, Hong Kong, Hong Kong, 2023, pp. 1-7.

[5] D. Maksimovski and C. Facchi, "Decentralized V2X Priority Maneuver Coordination: Evaluation in Small Scale Scenarios," *2023 IEEE International Automated Vehicle Validation Conference (IAVVC)*, Austin, TX, USA, 2023, pp. 1-8.

[6] ETSI TS 103 561, “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Maneuver Coordination Service”, V0.0.8 (2024-06), (Draft).

[7] SAE J3186, “Application Protocol and Requirements for Maneuver Sharing and Coordinating Service”, March 2023.

[8] B. Lehmann, H. J. Günther and L. Wolf, “A Generic Approach towards Maneuver Coordination for Automated Vehicles,” *Proc. IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, Hawaii, USA, pp. 3333-3339, 4-7 Nov. 2018.

[9] R. Molina-Masegosa, S. S. Avedisov, M. Sepulcre, Y. Z. Farid, J. Gozalvez and O. Altintas, "V2X Communications for Maneuver Coordination in Connected Automated Driving: Message Generation Rules," in *IEEE Vehicular Technology Magazine*, vol. 18, no. 3, pp. 91-100, Sept. 2023.

[10] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang and H. Chen, "A Survey on Trajectory-Prediction Methods for Autonomous Driving," in *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652-674, Sept. 2022.

[11] H. Arbabi and Michele C. Weigle, “Highway Mobility and Vehicular Ad-Hoc Networks in ns-3,” *Proc. of the Winter Simulation Conference*. Baltimore, MD, USA, pp. 2991-3003, 5-8 Dec. 2010.

[12] G. Thandavarayan, M. Sepulcre and J. Gozalvez, "Cooperative Perception for Connected and Automated Vehicles: Evaluation and Impact of Congestion Control," in *IEEE Access*, vol. 8, pp. 197665-197683, 2020.