Advances in the Hierarchical Emergent Behaviors (HEB) Approach to Autonomous Vehicles

Damian Roca, Rodolfo Milito, Mario Nemirovsky, and Mateo Valero

Abstract—Widespread deployment of autonomous vehicles (AVs) presents formidable challenges in terms on handling scalability and complexity, particularly regarding vehicular reaction in the face of unforeseen corner cases. Hierarchical Emergent Behaviors (HEB) is a scalable architecture based on the concepts of emergent behaviors and hierarchical decomposition. It relies on a few simple but powerful rules to govern local vehicular interactions. Rather than requiring prescriptive programming of every possible scenario, HEB's approach relies on global behaviors induced by the application of these local, well-understood rules. Our first two papers on HEB focused on a primal set of rules applied at the first hierarchical level. On the path to systematize a solid design methodology, this paper proposes additional rules for the second level, studies through simulations the resultant richer set of emergent behaviors, and discusses the communication mechanisms between the different levels.

Index Terms—Hierarchical Emergent Behaviors (HEB), Internet of Things (IoT), Autonomous Vehicles, Ultra Large Scale Systems (ULSS)

1 Introduction

The Internet of Things (IoT) adds a new dimension to Internet through the physical interaction with the real world. Sensors and actuators are the enablers of those interactions. Smart Cities and Smart Transportation are prime examples of large-scale systems that interact actively with the environment.

Among the latter, Autonomous Vehicles (AVs) are strongly positioning as a rapidly developing area, achieving remarkable milestones day after day. Aerial (i.e. drones) and terrestrial vehicles (i.e. cars and trucks) are already perceiving their environment using a plethora of technologies (i.e. Lidar, cameras, infrared) to reach their destination safely while avoiding collisions [1].

Nowadays most approaches in the open literature focus on developing technologies from the single vehicle perspective. Corroborations can be found in the construction of high-definition maps [2] to navigate each car and the separation of onboard hardware and software platforms [3]. Techniques to coordinate groups of vehicles focus on areas of interest such as intersections [4], [5], [6] and highway merging ramps [7], [8]. However, there is no unified theory or consensus on how to design and orchestrate such large systems with millions of vehicles and an uncountable number of external variables (i.e. pedestrians, driving rules, etc.).

Hierarchical Emergent Behaviors (HEB) [9], a recently proposed architecture, combines emergent behaviors with hierarchical decomposition to tackle this problem. These areas have been previously studied separately in different fields such as biology and engineering, but to the best of our knowledge they were not fused and applied to Ultra-Large Scale Systems (ULSS) such as AVs. HEB induces useful behaviors through local rules implemented at each AV rather than explicitly programming each action a vehicle must take in every circumstance [10]. Unlike swarms, AVs operate at different spatial and temporal levels, reason why the level (N+1) abstracts the emergent behavior of the previous level (N), widening its spatial-temporal scope and enabling HEB's natural scalability.

Relying on HEB has major benefits. The first is the absence of highly complex algorithms. The second is HEB's intrinsic adaptivity to deal with unanticipated corner cases. These objectives are achieved by moving the decision-making capabilities to the vehicles and thus allowing them to take actions based on well understood rules. These rules exploit the vehicle's contextual information (i.e. other vehicles, environment, etc.) to take real time decisions at the vehicle level, where the information is generated.

The next logical step requires the development of a design methodology to build, evaluate, and run HEB-based solutions for AVS. Towards this goal, this paper advances previous work on:

- Architectural foundations of the second level and its implications, with a focus on inter-level communication & locality and hierarchical relation between the rules, including the necessity of a leader and possible mechanisms to implement its selection.
- Demonstration of the robustness, flexibility, and smoothness of a HEB-based AV system.
- Case study to validate the previous points, incorporating new rules and experimental observations

This paper is organized as follows. Section 2 takes the concepts from the first publications and starts the devel-

- *Damian Roca and Mateo Valero are with the Barcelona Supercomputing Center (BSC-CNS) and the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. Email: {name}.{surname}@bsc.es*
- *Mario Nemirovsky is an ICREA Senior Research Professor at BSC-CNS.*
- *Rodolfo Milito is a Chief Data Scientist at Starflow Networks Inc, California, USA*

opment of a theory that fundaments HEB. Later, Section 3 presents the case study where new rules are evaluated together based on the hierarchical level they belong to. Section 4 closes the article with the conclusions and future work.

2 First Steps from High Level Concepts to a Solid Theory

Previous papers introduced the HEB concept [9], discussed the role played by Fog Computing [11], and explored through simulations a fairly rich set of emergent behaviors displayed under a variety of scenarios. The gained experience has convinced us of the potential of HEB to become an important piece in advancing the introduction of autonomous vehicles at scale. A comprehensive theory of the phenomenology of collective behavior induced by local rules, and the interaction of the different elements within the system is required to consolidate HEB's ideas.

The ultimate goal is the development of a comprehensive theory that: a) captures the phenomenology of the collective behaviors induced by local rules; b) relates behaviors at different hierarchical levels; c) determines with high degree of confidence the range of validity of the approach. Such a theory would provide the foundational basis for the indispensable design methodology.

Toward this goal we focus in the following Sections on the communication (intra- and inter-level), the shaping of desired global behaviors through simple modifications of Reynolds's local rules, extensions of those rules to higher levels in the hierarchy, and key architectural attributes to quantify behaviors.

2.1 The vital role of communications

HEB relies on sensorial activity and communications to induce useful behaviors. A car not capable of knowing its environment (i.e. neighboring cars, obstacles) and its own condition (i.e. speed, position) will hardly produce either safe or interesting behaviors. Current technologies such as Lidar already support these needs, and its software integration is advancing quickly.

The behavior at the first hierarchical level is largely induced by local interactions between neighboring vehicles and their environment. Platoons emerge naturally from the application of first-level local rules. Second-level rules can induce behaviors that extend the scope of first-level rules (regulating, for instance the interactions between platoons). This requires mechanisms for both intra- and inter-level communication. Taking the platoon as the elemental emergent behavior, cars need to sense each other (intra-) and simultaneously the platoon they form need to communicate with other platoons (inter-), as reflected in Figure 1. This horizontal communication between platoons at the same hierarchical level is a major HEB's differentiation with traditional hierarchical architectures that solely rely on vertical communications (i.e. tree topology).

This broader locality translates into different detection ranges at the on-board "things". Despite this fact, the elemental unit that applies the different hierarchical rules and senses the environment remains unaltered, the vehicle.

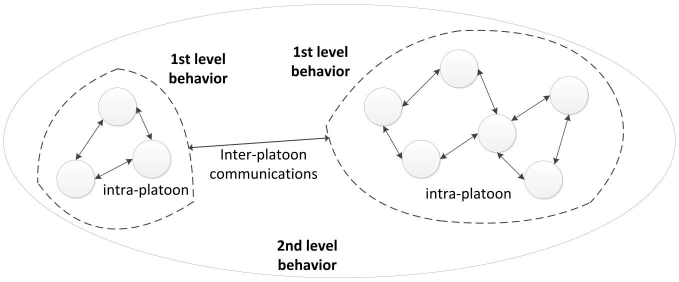


Fig. 1: HEB's representation including intra- and inter-behavior sensing and two different level behaviors (1st level with dotted lines and 2nd level with continuous line)

HEB elemental units can use a passive mechanism where each vehicle bases its behavior solely on the information sensed from its environment (also including other vehicles). Using this mechanism "things" become critical to achieve useful and secure behaviors. Another possibility could use active and direct mechanisms to code information between different entities and thus influencing the behavior directly [12]. This direct strategy reduces the pressure placed upon "things", but moves the complexity to communication, synchronization, and coordination protocols. This technique precludes some of HEB's major advantages by introducing the need that certain elements in the system such as vehicles need to be aware of the size of the platoon.

A complementary technique relies on infrastructures such as Fog Computing [13] to provide contextual information, enlarging the information scope of each vehicle beyond the on-board sensors. Vehicles can transmit information to Fog nodes located near the road to send/receive information while gaining visibility over the emergent behaviors. Hence, vehicle-to-infrastructure (V2I) communications complements the aforementioned vehicle-to-vehicle (V2V) capabilities. On one side this technique reduces the pressure placed upon vehicles. On the other side relying on a side infrastructure for certain tasks moves away from the vehicles certain degree of independence to take their own decisions. There is a possibility to limit this fact that only uses Fog nodes to receive information from the vehicles without intervening in their decision process.

2.1.1 Emergent behavior organization

Communication is essential to induce collective behaviors at every level of HEB's hierarchy. There are important differences between the first and the higher levels in the requirements and organization of the communication. At the first level this is strictly a local issue, that requires only V2V and V2X communication. In contrast, the second level requires also communication between platoons. There are several ways to approach this problem, and even variants within them. Let us outline the main ones.

The first solution calls for assigning a leader in each platoon. This leader, who could be either virtual [14] or physical [10], would drive the behavior of the platoon using the three Reynold's rules. While applying those three rules the platoon ends up following the trajectory of the leader who has a clear objective. Platoons communicate through their leaders that now represent all their constituting vehicles. It could be argued that depending on a leader goes somewhat against the grain of HEB's distributed principles by introducing a centralization element. This suggests the consideration of distributed leadership schemes, which at the cost of higher level of local intelligence, could ensure scalability and autonomy in extreme scenarios.

2.2 Behavior inducement

Nature's goals (i.e., survival and reproduction) guide animal behavior. In the case of HEB we need to cast human and societal needs (for instance, reduction of driving time, and fuel consumption) into goals.

Once the goals are determined, the task is to find rules that induce behaviors to meet them. These goal-based rules can be combined with those rules ensuring that vehicles circulate accordingly to the traffic rules (i.e. speed limitation on a per road basis). At this stage of development of HEB concepts, simulation appears as the right tool for the task. It enables exploration of the effect of some rules and the environment on others, and determination of how well the overall objectives are met. The quality of the results naturally depends on how well the simulator captures real life scenarios and their constraints.

Our previous papers explored the first level behavior of AVs as the result of applying Reynolds rules. The next step is to extend that exploration to second level behavior. The challenge is to find rules that induce desired behaviors without destroying the first level platoon formation. In Section 3 we introduce rules that accomplish different AVs maneuvers.

We envision that as the theory underlying HEB consolidates, clever application of Machine Learning (ML) techniques will allow further extension of the rules, with a richer portfolio of emerging behaviors. For related example of how ML can automize the process, see Mataric [15], [16].

2.2.1 Single-level vs multi-level approach

The multiplicity of hierarchical levels differentiates HEB with respect to preceding work in robotics focused on single level solutions [15]. The (N-1)th level enhances the scalability of the system and expands scope of achievable emergent behaviors.

We note that independently of the number of hierarchical levels, each vehicle is responsible for following a set of rules locally. Some rules are more critical than others (collision avoidance being a clear example). Hence, rules are organized in a hierarchy of dominance determined by their criticality. In practical terms, that dominance is expressed by the weights associated with each rule. These weights ponder the rules' results to determine the trajectory update. Weights are part of the rules' hyper-parameters alongside with other numerical parameters with physical meaning.

In consequence, hyper-parameters define the rules and their relationships (including their hierarchy and criticality). For instance, collision avoidance is a first level rule with weight larger than any other rule at any level, because safety is the dominant consideration. Among the parameters englobed there are the neighboring distance that vehicles use to determine their vision range (equivalent to the sensors' range) and the separation distance required between vehicles and obstacles to prevent collisions. Other parameters such as maximum speed could be regulated as hyper-parameters.

At the single level the organization of the rules depend entirely on their relative weights. A multi-level design allows building richer behavior by combining first order ones or introducing new rules. First level rules within a platoon are usually always active, although the final decision depends on what the vehicle sense of its environment. In contrast, higher level rules are activated only when certain conditions are met, given the system the ability to incorporate "intelligent awareness".

For instance, a second level condition may trigger when two platoons become in close proximity, preventing V2V interactions between vehicles in different platoons. An illustrative condition requires that a vehicle has one or more other vehicles in its platoon and that detects a platoon ahead of them with two or more vehicles. When this condition is fulfilled, then a certain second level rule may apply.

2.3 Behavior shaping

We study two approaches to the problem of shaping a given behavior induced by a set of rules: (i) slight modification of the rule/s, and (ii) tuning of the hyper-parameters. Through shaping behaviors a rule can go from being functionally correct but rough to smooth (e.g. making abrupt maneuvers more comfortable to the passengers), and from reaching a destination broadly defined to meeting a precisely defined one.

2.3.1 Rule/s modification

Designing rules that express elaborate behaviors is an organic process. It starts with a core of elementary, local rules (Reynold's platoon formation) and compose them in complex chains that achieve specific objectives (compact the platoon, move aside, increase speed until a moving obstacle gets behind, etc.).

Let us consider some concrete instantiations of rules modification. We have already presented a destination rule [9] that directs vehicles to targeted points without specifying the trajectory to follow. Each vehicle just knows its current position and its target destination. While this rule works fine at the vehicular level, the preservation of the platoon formation depends on how close the destination points specified for each individual car are, and the shape of the road. The rule can be retouched by applying it consecutively to a chain of segments, each one with its own target destination. The final coordinates remain unaltered, but the finer granularity ensures that vehicles remain close enough to induce platoon formation, hence maximizing traffic throughput [17].

An alternative implementation relies on V2V communication to exchange information regarding current positions, final destinations, and contextual information to determine intermediate target points. This latter alternative places stronger processing requirements on the vehicles on-board units (OBUs). Another option defines a vector of targets instead of a single target point at the beginning of the trajectory. When vehicles reach the proximity of a vector element, their immediate target changes to the next vector element.

A roadside infrastructure like Fog can facilitate the implementation of this sequence of targets. Fog nodes' expanded scope allows the smart processing of target destinations and congestion information to dynamically building the chain of segments that compose the trajectories.

2.3.2 *Hyper-parameters tuning*

The term hyper-parameter includes both the internal parameters of each rule (i.e. minimum separation distance) and the weights assigned to the rules. The core of hyper-parameters tuning takes place during the extensive experimentation phase in the controlled environment of a simulator. The simulator allows quantifying the behavior of experimental rules under a wide variety of scenarios.

This simulation phase leads to the acceptance of the rules and associated hyper-parameters, the tuning of the hyper-parameters, or, in extreme cases, downright rejection of the experimental rules. Note that the experimental rules examined not in isolation, but interacting with the whole set of rules. The tuning of the hyper-parameters provides their default values to induce interesting behaviors. Default configurations are not limited to single values rather than ranges, giving HEB a great flexibility. For example, the separation distance between vehicles can be a value between a few meters and the sensors' range, any of which can result in a platoon.

Besides the aforementioned technique, vehicles can also tune their rules' hyper-parameters in real life environments based on their observations and data they receive from Fog Computing. For instance, the separation distance can be modified to make maneuvers such as overcoming obstacles smoother. In the future, ML techniques can make vehicles learn new values for their hyper-parameters.

2.4 HEB's decision flow

HEB-enabled vehicles base their decisions on what they sense, reason why they are always observing their surroundings to detect changes to act upon. This data fundaments HEB's system and is used to check the condition for each rule to determine what rules apply in a precise moment. Each time something changes in the vehicle's environment, these conditions have to be verified immediately.

Once HEB knows what rules are applying, it is time to actually execute the code behind that set of rules obtaining as a result direction vectors. After applying their corresponding weights, these vectors are summed to obtain a single trajectory modification. Lastly, HEB acts upon the vehicle's direction to modify its trajectory based on that resultant vector. Then, the process starts again as reflected in Figure 2.

2.5 Architectural attributes

Behaviors can be visually assessed, but their rigorous characterization requires the consideration of basic attributes that reflect the intent of the designer, and that can be translated into appropriate metrics.

- Sensitivity expresses the ability of the system to react to external stimuli. A hyper-sensitive vehicle may react too soon, or too violently. For instance, a system requiring infinite precision in the tuning of its parameters is impractical. In our case, the admissible loci of the hyper-parameters extend over a (not necessarily connected) hyper-volume.
- Expressiveness refers to HEB's ability to induce new desired behaviors extending the core framework through slight modifications of the existing rules or the addition of new ones, without affecting the existing ones. For instance, the three original Reynold's rules surpassed our expectations, because slight modifications enabled novel behaviors.
- Smoothness relates to the user experience. Desired behaviors, including braking, acceleration, and change of direction shall not be brusque and respect the traffic guidelines.

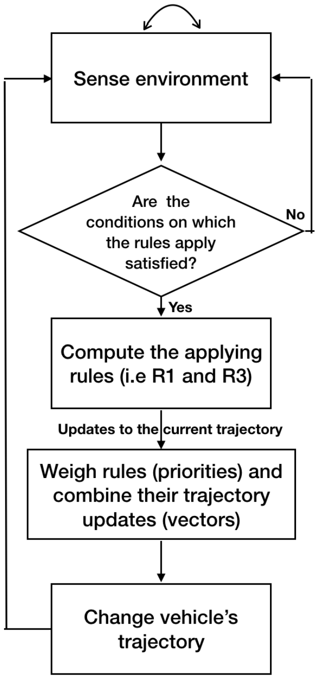

Sense environment
Are the conditions on which the rules apply satisfied?
No
Yes
Compute the applying rules (i.e R1 and R3)
Updates to the current trajectory
Weigh rules (priorities) and combine their trajectory updates (vectors)
Change vehicle's trajectory

Fig. 2: Each vehicle operating under HEB's principles determines its trajectory through this flow diagram, executed as an infinite loop till the vehicle stops.

3 Case study

This section studies the cross-level interactions and their impact on AVs behaviors by looking at a few use cases, adding/modifying rules, and tweaking hyper-parameters. The setting assumes indirect communication established by vehicles through the sensing of their surroundings but without exchanging explicit messages. There is no leader, but each car has the ability to determine the level it belongs to, and to make decisions.

3.1 Fundamentals

There are two types of terrestrial AVs represented by triangles of different colors (black and grey respectively) highlighting their direction. Obstacles define the shape of different scenarios such as highways or intersections, represented by black dots in the canvas.

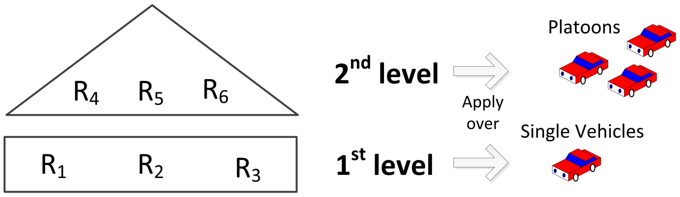


Fig. 3: Depiction of a two level rule hierarchy and the entities at each level. R_1, R_2, and R_3 are the original Reynolds rules while R_5 is derivates from them. In contrast, R_4 and R_6 are not part of Reynolds work. Single vehicles constitute the entities of the first levels, while platoons constitute the entities of the second level (in the use case of this paper).

Each vehicle implements the three original rules from Reynolds (R_1 Alignment, R_2 Separation, and R_3 Cohesion) [19]. Saber presented their theoretical description in his technical report [18].

In addition, there are two more rules conforming the basic set: R_4 and R_5. R_4 establishes a target destination point that each vehicle has to reach. Vehicles compute vectors based on their current location and their target destination to determine the direction that leads each of them to their destinations. This vector is then used to update the vehicles' trajectory along with the results of the other rules. R_5 performs the same operations as R_1, R_2, and R_3 but all aggregated in a single rule. The resulting rule (R_5) is more complex than the three Reynolds rules but induces the same behavior. Since R_5 applies over platoons instead of single vehicles, it results in the creation of a platoon of platoons. Both rules are implemented as second level rules but apply under different environmental conditions. While R_4 acts upon individual platoons (defined by the presence of two or more vehicles within a certain distance), R_5 requires at least one platoon and one vehicle of another type to be applied.

On top of those, a new rule R_6 is added. R_6 is a complex 2nd level rule that guides interactions between different types of platoons circulating along a highway. It applies over vehicles when two types of platoons detect each other (what constitutes a 2nd level membership condition). Then R_6 acts upon both platoons. The approaching platoon redirects its trajectory to the left part of the road while the approached moves toward the right side. These turns are induced through a set of target destinations with an offset rather than being specifically programmed. The basic implementation of R_6 does not alter the velocity of each platoon. The rules' weights are set to prioritize the collision avoidance above the rest to ensure safety. Figure 3 represent the rules and the level they belong to.

3.2 Methodology

We chose the Processing simulator [20] to perform our analysis of AVs. Each vehicle has the aforementioned set of rules that guide their behavior to reach a target destination without clearly specifying a trajectory. Processing offers a rich framework to simulate and visually observe a set of vehicles within an environment and how they interact. We also implement a mechanism to detect incorrect behaviors due to the violation of the rules (i.e. a vehicles colliding), complemented by a visual validation using the canvas. The overall performance of the system is evaluated using both the visual observation and the actual alarms. The code is available at our repository [21].

3.3 2nd level: Platoon of platoons

Previous work explored the idea of platoons and how to enable them either by programming [22] or by inducement [9]. The same concept can be applied at a 2nd level, resulting in a platoon of platoons. To analyze this scenario we use R_5 that applies the original rules over platoons.

Two 1st level platoons constitute a 2nd level platoon, that moves along the highway in a similar structure of that presented in Figure 1. Even though vehicles are responsible of determining its 2nd level membership to apply the proper rules through each rule condition (i.e. detect another platoon in its surroundings), the scalability is not compromised. The reason lies beneath the fact that not all vehicles are aware of the platoon behind them. Only those which sensor range allows them to detect those vehicles know about its existence. Even when the number of vehicles aware of the 2nd level is small, the imitation capabilities of the original rules makes the entire platoon to follow them, acting as a single 2nd level entity (similar effect to a wave propagation).

The separation distance between platoons constitutes a primary example of a behavior sensitivity analysis in conjunction with the sensors range (we suppose a fixed value for these ranges). When we analyze small increments of the separation distance, a 2nd level platoon still emerges. The limit in this case is fixed by the sensors capabilities. When the separation distance exceeds the sensors range the two platoons cannot "see" each other, preventing the emergent 2nd level platoon.

On the contrary, when the distance is reduced the 2nd level platoon emerges. The only major appreciation results when the separation distance is equal or smaller than the inter-platoon collision avoidance value. In this case the behavior is a larger and unique platoon. No collision occurs because the collision avoidance rule (R_2) prevails over the other rules. Facing these results, we can conclude that the sensitivity of this set of rules is low and that the resultant behaviors are quite good, emerging for a wide range of hyper-parameter values.

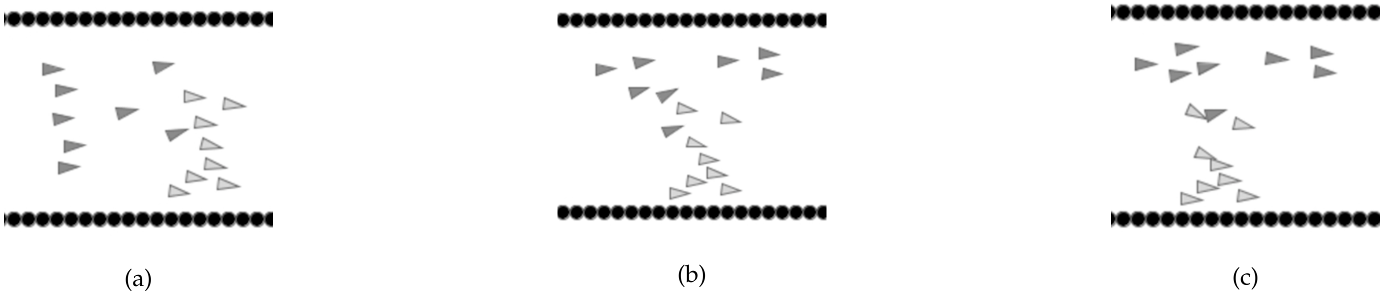
Having a second level platoon results in an optimized traffic flow. Vehicles and infrastructure can adjust the separation distance and the velocities to adapt to road conditions without affecting 2nd level platoon entities.

3.4 Highway overcoming maneuver

This scenario highlights the overcoming maneuver optimization between platoons. For this purpose, we disable R_5 and activate R_6, the new rule designed to induce behaviors when two platoons face each other. Remember that by default R_6 does not modify the velocity.

3.4.1 1st level implementation

The natural comparison to the aforementioned maneuver arises from a single level implementation. Each vehicle faces the situation alone rather than in conjunction with the rest of the platoon. This difference is reflected in the conditions required to apply R_6 as a 2nd level rule. It is only necessary



(a) (b) (c)

Fig. 4: Temporal sequential representation of the overtaking maneuver with R_6 implemented as a first level rule

to detect another type of vehicle ahead to start the maneuver but not to be a part of a platoon or detect another one. R_6 weight is set to the same value of the other rules except R_2.

Figure 4 shows a highway where a faster platoon (black) is about to reach a slower platoon (grey). When the rule is implemented as a first level rule, the behavior obtained is not smooth as an architect would like. However, the resultant behavior is modified and the interaction suffers a small optimization. The slower vehicles move towards the right side (as individual elements, any action is taken as a platoon). This fact is exploited by the faster vehicles that move to the left side for an overtaking maneuver.

As expected, in most of the simulations single vehicles cross the slower platoon instead of continuing attached to their original behavior. This action could put in danger the safety of both platoons, plus implies a strong modification of the first level behaviors (rupture of the platoon behavior). The faster platoon faints to overtake the other.

3.4.2 2nd level implementation

Trying to avoid those situations, R_6 is implemented as a second level rule. As mentioned earlier, the key differentiation between a 1st and a 2nd level implementation lies beneath the applying conditions for the rule. In R_6, these conditions are to be part of a platoon and to detect the presence of a different platoon (or part of it) ahead or behind. In consequence, each vehicle acquires membership awareness to apply 2nd rules as part of a platoon. Despite this fact, vehicles may not know the total size of the platoon to not compromise its scalability. R_6 now influences the vehicles' trajectories only when those conditions are satisfied.

Although it may seem a simple modification, the fact that vehicles try to stay within its platoon rather than taking individual actions results in the absence of crossing cars between platoons. Figure 5 depicts a temporal sequence of images to capture the overtaking maneuver.

Different observations arise from these figures. First, the shape of the platoon changes, but this is not a problem since it was not specifically programmed. In fact, this behavior is desirable because thinner and longer shapes result, facilitating the maneuvers. Second, the destination target modification from a single point to a set of points applies only when R_6 conditions are satisfied (be part of a platoon and detect another one). Within the context of the highway analysis, this set of targets is determined by a vertical offset to induce that movement to the sides (left or right) over a spatial sequence in the horizontal dimension. If only the vertical offset is applied together with a single destination point the behavior reaction was not fast enough to ensure a smooth maneuver.

Although the objective was accomplished with this rule, an architect can explore how to enhance the resultant behavior through the modification of rule. This process is referred to as "Behavior Shaping", as explained in Section 2.3.

3.5 Behavior shaping: a practical example

Since formations revealed as a way to optimize interactions, this the first technique analyzed. Using a rectangular formation induced through the initial positions of the AVs resulted in smoother maneuvers in both the intersection and the highway scenarios without the need for traffic lights [23], [24].

An important remark is how adaptive the rules are to enable more behaviors, in this case through a formation. Instead of designing a new rule or modifying the existing ones, setting studied initial positions to each vehicle at the beginning of the simulation created the formation. Later, applying the three original rules resulted in the maintenance of the formation for their entire trajectory even when facing other platoons.

Another technique alters the hyper-parameters of the platoons. For instance, if the intra-platoon separation distance is modified upon the detection of the other platoon the total surface occupied by each platoon is smaller and in consequence the space for the overtaking maneuver becomes larger, reducing the risk of potential collisions or strange maneuvers (i.e. 90 degrees turns). Similarly to the target destination modification, hyper-parameters are modified upon the fulfillment of the 2nd level membership conditions. When the faster platoon has completed the overtaking, both platoons can return to their original separation distances because 2nd level conditions are no longer satisfied.

The platoon velocity is a different parameter to consider. In this case, upon the detection of a slower platoon ahead the faster platoon can moderate its speed to have more control over its maneuvers. On the contrary, the slower platoon can momentarily accelerate to move aside faster, leaving more space for the overtaking maneuver. These accelerations or speed modifications are relative to the current vehicles' speed and always adapt to the traffic safety regulations (i.e. road limitations). One possibility is to use Fog nodes to provide contextual information to the AVs, contributing to determine the N-level membership condi-

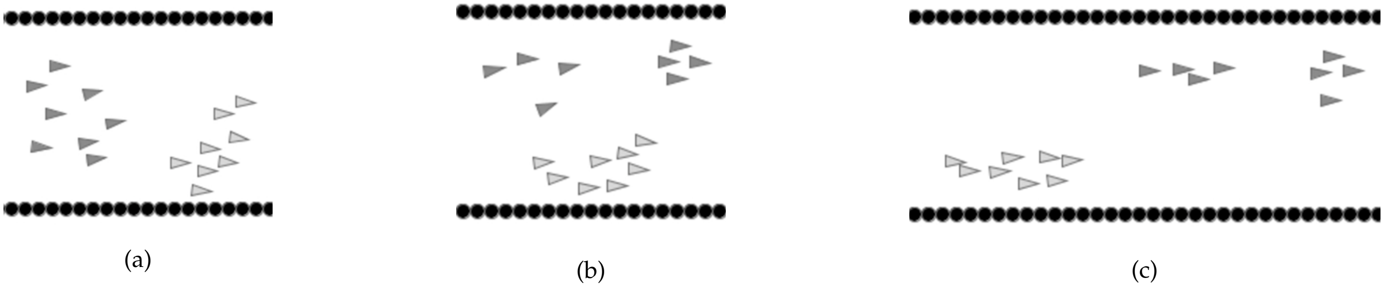


Fig. 5: Temporal sequential representation of the overtaking maneuver with R_6 implemented as a second level rule. Each vehicle gained consciousness of platoon membership (at least partially) to apply second level rules.

tions. These external nodes could also act directly upon the vehicles exploiting its larger visibility.

3.5.1 Phase-based rules

Till now vehicles had a simple second level condition (i.e. be part of platoon and detect another one), but more detailed conditions can be studied based on their impact over the resultant behaviors. For instance, within a condition we can have different phases based on contextual information, inducing more robust behaviors.

For example, we can use a three phase illustrative example. The first phase is activated upon the detection of the slower platoon ahead. In that moment this phase triggers the aside movements. The second phase is activated when the distance between platoons is smaller. In this situation, the slower platoon accelerates for a moment and the faster platoon maintain its speed. This momentary acceleration does not compromise the overtaking maneuver thanks to the target modification and its short burst nature. The third and final phase is activated when the distance between platoons is negligible and both have a clear path ahead, accelerating the faster platoon to complete a faster overtaking maneuver. At the end, when this maneuver is completed and the distance between platoons is larger, the 2nd level conditions are no longer satisfied and default conditions are restored. In this example only the inter-platoon distance has been considered as contextual information, but many more parameters can be used.

The aforementioned techniques revealed how small modifications with low complexity have a great effect on induced behaviors. In future implementations, machine learning techniques can guide the behavior shaping process in its different approaches (hyper-parameters and phases), as proven by the first level approach.

4 Conclusion

This paper has made some solid steps along the goal of systematizing HEB's concept into a solid design methodology. In particular, we have advanced the understanding of the communication mechanisms required between the different hierarchical levels, discussed the desirable attributes of vehicular behaviors, and demonstrated through simulations how simple second level rules can enrich the space of emergent behaviors.

Future work includes furthering the understanding of the steps to build new behaviors without affecting the quality of the existing ones, and leveraging machine learning techniques to tune the rules hyper-parameters.

Acknowledgments

Damian Roca work was supported by a Doctoral Scholarship provided by Fundación La Caixa. This work has been supported by the Spanish Government (Severo Ochoa grants SEV2015-0493) and by the Spanish Ministry of Science and Innovation (contracts TIN2015-65316-P).

References

[1] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle detection techniques for collision avoidance systems: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2318–2338, 2015.

[2] Waymo maps for self-driving cars. [Online]. Available: https://medium.com/waymo/building-maps-for-a-self-driving-car-723b4d9cd3f4

[3] Software defined vehicles. [Online]. Available: http://www.electronicdesign.com/automotive/bmw-and-audi-want-separate-vehicle-hardware-software

[4] A. de La Fortelle, "Analysis of reservation algorithms for cooperative planning at intersections," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 445–449.

[5] I. H. Zohdy and H. Rakha, "Game theory algorithm for intersection-based cooperative adaptive cruise control (cacc) systems," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 1097–1102.

[6] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Optimal control and coordination of connected and automated vehicles at urban traffic intersections," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 6227–6232.

[7] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, "Optimal vehicle trajectory planning in the context of cooperative merging on highways," *Transportation research part C: emerging technologies*, vol. 71, pp. 464–488, 2016.

[8] J. Rios-Torres and A. A. Malikopoulos, "Automated and cooperative vehicle merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2017.

[9] D. Roca, D. Nemirovsky, M. Nemirovsky, R. Milito, and M. Valero, "Emergent behaviors in the internet of things: The ultimate ultra-large-scale system," *IEEE Micro*, vol. 36, no. 6, pp. 36–44, 2016.

[10] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil, "Coordination and navigation of heterogeneous mav–ugv formations localized by a hawk-eye-like approach under a model predictive control scheme," *The International Journal of Robotics Research*, 2014.

[11] D. Roca, R. Milito, M. Nemirovsky, and M. Valero, "Tackling iot ultra large scale systems: Fog computing in support of hierarchical emergent behaviors," in *Fog Computing in the Internet of Things*. Springer International Publishing, 2017, pp. 33–48.

[12] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[14] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3. IEEE, 2001, pp. 2968–2973.

[15] M. J. Mataric, "Interaction and intelligent behavior." DTIC Document, Tech. Rep., 1994.

[16] M. J. Matarić, "Reinforcement learning in the multi-robot domain," *Autonomous Robots*, vol. 4, no. 1, pp. 73–83, 1997.

[17] R. Hall and C. Chin, "Vehicle sorting for platoon formation: Impacts on highway entry and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 405–420, 2005.

[18] R. O. Saber, "A unified analytical look at reynolds flocking rules," California Institute of Technology Pasadena, Tech. Rep., 2003.

[19] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*. ACM, 1987.

[20] Processing simulation framework. [Online]. Available: https://processing.org/

[21] Github heb repository. [Online]. Available: https://github.com/damianroca/HEB

[22] P. Varaiya, "Smart cars on smart roads: problems of control," *Automatic Control, IEEE Transactions on*, 1993.

[23] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, 2012.

[24] M. A. S. Kamal, J.-i. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1136–1147, 2015.

Damian Roca is a researcher in the Unconventional Computer Architectures and Networks Group at the Barcelona Supercomputing Center. His research interests include Emergent Behaviors, Autonomous Vehicles, the Internet of Things (IoT), and Fog Computing. Roca holds a PhD in Computer Architecture from the Universitat Politècnica de Catalunya . Contact him at: damian.roca@bsc.es

Rodolfo Milito is the Chief Data Scientist at Starflow Networks Inc.. He is currently engaged in IoT, Big Data and Machine Learning, and working on a distributed compute, storage, and network platform from the edge to the core of the network nicknamed "Fog Computing". Milito has a PhD in EE (Stochastic Control Systems) from UIUC. Contact him at rmilito@starflownetworks.com

Mario Nemirovsky is a Catalan Institution for Research and Advanced Studies (ICREA) Senior Research Professor at the Barcelona Supercomputing Center. His research interests include computer architectures, high-performance computing, the Internet of Things, and emerging on-chip interconnect technologies. Nemirovsky has a PhD from the University of California, Santa Barbara. Contact him at mario.nemirovsky@bsc.es.

Mateo Valero is a full professor at Computer Architecture Department at the Universitat Politècnica de Catalunya and the Director of the Barcelona Supercomputing Center. His research focuses on high-performance computer architectures. His main awards: Seymour Cray, Eckert-Mauchly, Harry Goode, ACM Distinguished Service. He is Honorary Doctorate by 7 Universities. He is a fellow of IEEE and ACM and is and Intel Distinguished Research Fellow. He is member of 5 academies. Contact him at mateo.valero@bsc.es