Proceedings of 17th International Symposium on Advanced Vehicle Control (AVEC'26)
7-11 September 2026, Tsukuba, Japan

Modular Multi-model Motion Planner for Autonomous Vehicles

Sabrina CIUFFOLETTI [1)] Mattia PIAZZA [1)] Alice PLEBE [1)]
Gastone Pietro ROSATI PAPINI[1)]

1) Department of Industrial Engineering, University of Trento

Abstract: Modern autonomous systems rely on motion planning as a core component for safe and efficient navigation. While physical hardware varies across platforms, the primary challenge is translating high-level goals into smooth, feasible trajectories. In complex, dynamic environments, a planner must also be computationally efficient enough to react to sudden changes. Developing algorithms that maintain these properties across different operating conditions is essential for the reliability of any autonomous agent. This work presents an autonomous vehicle planning framework designed to handle both structured road-following and unstructured scenarios, such as target-gate navigation. Through a modular architecture, the system enables a smooth transition between these environments without requiring discrete shifts in the underlying logic. Furthermore, the framework is extended to support multiple vehicle models, allowing the planner to adjust its kinematic constraints based on the specific maneuver or platform. Experimental results across diverse environments demonstrate that the planner maintains boundary compliance and feasibility for the planned motion, establishing a robust foundation for a unified, multi-model planning architecture.

Keywords: emergent behavior, optimal control, vehicle planning, multi-model, multi-scenario

1. Introduction

Autonomous vehicle navigation has seen a rapid growth in recent years, leading to a wide variety of planning algorithms in the literature. However, traditional methods still face significant trade-offs.

While being established methods, search-based algorithms [8,13] and constrained optimization [14,15] struggle with real-time computation, or require "warm-start" to handle complex environments. Conversely, End-to-End Learning [5,6] offers high flexibility but lacks the formal safety guarantees required for mechatronics systems.

The approach in the works of Da Lio et al. [2,3] bridges this gap, allowing the system to achieve the flexibility of neural networks with the mathematical rigor of analytical optimization. In this approach, the conventional two-stage motion planning problem is reformulated using a biologically inspired selection mechanism. First, the planning architecture derives analytical minimum-jerk solutions to define a continuous control manifold. Multiple candidate actions from this manifold are then evaluated in parallel, implementing the principles of the *affordance competition hypothesis* [1]. These options are discretized into a 2D *salience matrix* structured across longitudinal and lateral jerk profiles. Each candidate profile is assigned a salience value that accounts for both tracking optimality and safety constraints, allowing a competitive selection process to identify the optimal trajectory. By executing the selection mechanism over a pre-validated mathematical space, the framework ensures computationally efficient and kinematically safe trajectory generation.

Despite these advancements, most planning procedures still divide navigation into structured (lane-following) or unstructured (obstacle/gate-based) categories. In these contexts, the term "multi-scenario" can be interpreted in different ways. In [12], this term is used to identify different driving conditions, while in [7], it is used to hint at the capability of considering static and dynamic obstacles. However, real-world applications often demand a single framework capable of handling structured and unstructured environments simultaneously.

Moreover, to the best of the author's knowledge, versatile options that enable consideration of different vehicle models simultaneously within a parallel optimization pool remain unaddressed in the literature. The necessity of a framework capable of considering diverse robot models stems from the requirement for operational flexibility and algorithmic generalization. In complex environments, a vehicle's motion cannot always be captured by a single kinematic configuration. For instance, high-speed path tracking and tight, zero-radius maneuvering, such as the specialized control methods investigated in [11], demand fundamentally different kinematic constraints. Rather than relying on discrete, hard-coded state-machine switching, simultaneously embedding multiple vehicle models within the planning manifold allows the system to evaluate distinct behaviors in parallel. The framework can then smoothly choose the most suitable model based on situational feasibility. Furthermore, such a unified optimization design can be deployed across a fleet of heterogeneous robots without modifying the underlying planning logic.

This paper extends the concepts in the works of Da Lio et al. [2,3], proposing a unified architecture that enables navigation of roads and gates within a consistent, salience-based framework. Therefore, this work proposes a different interpretation of "multi-scenario", referring to the ability of a single planner to operate across both structured and unstructured domains without distinction. Moreover, the flexible and modular nature of this framework allows different vehicle models to be embedded simultaneously. Therefore, the main contributions are: (i) increased flexibility through a decoupled planning pipeline, enabling the integration of additional autonomous agents or alternative decision-making logic; (ii) extension to unstructured scenarios, alongside structured environments; (iii) scenario modularity, allowing for the fusion of different environmental and model constraints into a single salience map; (iv) an lightweight C++ implementation for better accessibility and performance.

2. Control Vehicle Model

A general agent can be described by a nonlinear kinematic model in a Frenet-Serret coordinate frame [4]. This curvilinear coordinate system is introduced because, compared to Cartesian coordinates, it allows a straightforward derivation of the control-oriented model equations.

The system model can therefore be written as:

$$\begin{cases} \dot{s}(t) = \dfrac{u(t)\cos(b(t)) - v(t)\sin(b(t))}{1 - n(t)\kappa(s(t))} \\ \dot{n}(t) = v(t)\cos(b(t)) + u(t)\sin(b(t)) \\ \dot{b}(t) = \Omega(t) - \kappa(s(t))\dot{s}(t), \end{cases} \tag{1}$$

where the physical interpretation of each variable is outlined in Table 1.

This general formulation can be simplified to describe various robotic platforms; in this work, the focus is on its reduction to the bicycle and unicycle models. It is important to note that the models derived in this section are specifically used as constraint models inside an optimal control problem (OCP) framework to plan trajectories. They are simplified to capture how the robot behaves over a short planning horizon, while the actual simulator calculates the full, complex physics step-by-step. To make the trajectory planning easier to solve, the control design assumes that the longitudinal and lateral motions are decoupled.

2.1. Bicycle

To get a practical control model for the bicycle configuration, specific simplifications are applied, separately for the longitudinal and lateral dynamics.

Longitudinal In the longitudinal case, the established simplifications assume a sufficiently small heading error $b(t)$ such that $\cos(b) \approx 1$, a negligible lateral velocity in the vehicle frame ($v(t) \approx 0$), and a small lateral displacement $n(t)$ relative to the curvature radius $1/\kappa(s(t))$ such that $1 - n(t)\kappa(s(t)) \approx 1$. Under these assumptions, the system in eq. 1 for the longitudinal motion of a bicycle model reduces to:

$$\begin{cases} \dot{s}(t) = u(t) \\ \dot{u}(t) = a(t) \\ \dot{a}(t) = j(t) \end{cases} \tag{2}$$

where $u(t)$ is the forward speed, $a(t)$ is the acceleration, and the control input $j(t)$ is the longitudinal jerk.

Lateral For the lateral trajectory optimization, the model is built under the hypotheses of a constant longitudinal velocity $u(t) = V_0$ and a zero side-slip angle ($\beta(t) = 0$). Additionally, the vehicle is assumed to maintain a steady-state curvature $\kappa_G(t)$ satisfying $\Omega = V_0\kappa_G(t)$, while small-angle and small-displacement approximations are applied such that $\cos(b(t)) \approx 1$, $\sin(b(t)) \approx b(t)$, and $1 - n(t)\kappa(s(t)) \approx 1$. To simplify the equations, a relative angular velocity state is defined as $b_d(t) = \Omega(t) - \kappa(s(t))V_0$. By choosing the global yaw acceleration as the control input, $r(t) = \dot{\Omega}(t)$, the lateral dynamics can be written as a linear system:

$$\begin{cases} \dot{n}(t) = V_0\, b(t) \\ \dot{b}(t) = b_d(t) \\ \dot{b}_d(t) = r(t) - \dot{\kappa}(s(t))V_0 \end{cases} \tag{3}$$

The complete set of states for the bicycle model can be thus written as: $\mathbf{x} = [s(t), u(t), a(t), n(t), b(t), b_d(t)]^T$.

2.2. Unicycle

While the longitudinal kinematics of the unicycle is the same as the bicycle model, the lateral behavior is handled differently. The main goal of this configuration is to manage a "rotate-on-itself" behavior. During this maneuver, the robot can experience large and rapid changes in its relative heading angle $b(t)$. Because of this, the small-angle approximations ($\cos(b(t)) \approx 1$, $\sin(b(t)) \approx b(t)$) are no longer valid. This leaves the system nonlinear and difficult to use directly inside a linear prediction horizon.

To maintain linearity and simplicity for this alignment behavior, the unicycle lateral OCP model restricts operations to a stationary or near-stationary condition. This operational envelope dictates a null forward longitudinal velocity ($u(t) = 0$) alongside a zero lateral velocity state ($v(t) = 0$). Furthermore, a small lateral displacement approximation relative to the track curvature radius is enforced, yielding $1 - n(t)\kappa(s(t)) \approx 1$. Applying these constraints to eq. 1 yields the simplified system:

$$\begin{cases} \dot{n}(t) = 0 \\ \dot{b}(t) = \Omega(t) \\ \dot{\Omega}(t) = r(t) \end{cases} \tag{4}$$

This simplified representation enables the OCP planner to compute smooth, minimum-jerk angular paths to correct the heading before the robot transitions back to the bicycle model for nominal operations.

The complete set of states for the unicycle model can be thus written as: $\mathbf{x} = [s(t), u(t), a(t), n(t), b(t), \Omega(t)]^T$.

Table 1: Variables for the curvilinear model

Symbol	Description	Unit
$u(t)$	Longitudinal velocity in vehicle reference frame	$[\mathrm{m\,s^{-1}}]$
$v(t)$	Lateral velocity in vehicle reference frame	$[\mathrm{m\,s^{-1}}]$
$\Omega(t)$	Yaw Rate	$[\mathrm{rad\,s^{-1}}]$
$s(t)$	Longitudinal coordinate of the path	[m]
$n(t)$	Lateral deviation from the reference path	[m]
$b(t)$	Heading error relative to the path tangent	[rad]
$\kappa(s)$	Path curvature at arc-length s	$[\mathrm{m^{-1}}]$
$b_d(t)$	Derivative of the heading angle error $b(t)$	$[\mathrm{rad\,s^{-1}}]$
$j(t)$	Longitudinal jerk	$[\mathrm{m\,s^{-3}}]$
$r(t)$	Angular acceleration $\dot{\Omega}(t)$	$[\mathrm{rad\,s^{-2}}]$

3. Planning Framework

The presented framework follows the structure reported in Figure 1. The pipeline is divided into an offline phase, where the analytical closed-form motion primitives are computed, and an online phase. Passing these pre-computed primitives to the online component reduces the computational burden on the real-time control loop, thereby enabling faster trajectory generation. Furthermore, the modular, decoupled nature of this design allows individual blocks to be modified independently. This scalability is critical when integrating new vehicle models or operational behaviors, as specific blocks can be adapted without altering the overarching structure of the pipeline.

3.1. Optimal Control

The proposed architecture employs a jerk-optimized planner, in which optimality conditions are derived analytically via Pontryagin's maximum principle (PMP) [9]. The selection of a jerk-minimization objective is biologically motivated, since empirical studies by Da Lio et al. [3] demonstrate that urban drivers naturally tend to exhibit minimum-jerk profiles. The motion primitives resulting from the OCP provide a basis for the subsequent stages of the pipeline. Rather than relying on purely numerical optimization, the production of an analytical solution allows for direct integration into the decision-making layer.

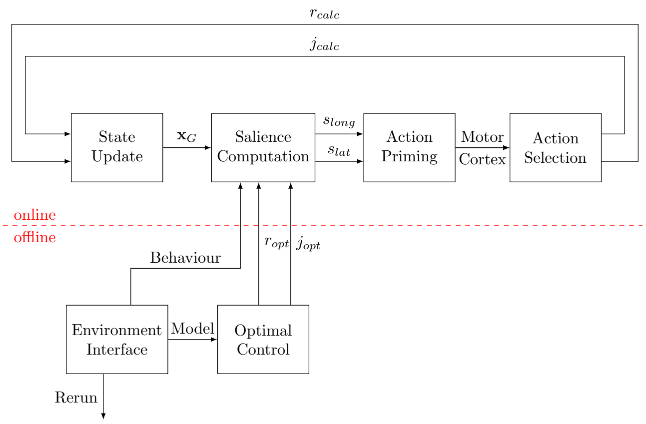


Fig. 1: As illustrated in the diagram, the framework is structured into distinct thematic blocks, each serving a specific purpose within the pipeline. Due to the decoupled nature of this architecture, individual blocks can be modified independently without affecting the rest of the system. A key feature of this design is the clear division between online and offline operations. By solving the optimal control problem offline to derive analytical, closed-form motion primitives, the computational overhead of the online planning components is significantly reduced.

The trajectory generation is structured as a constrained optimization problem aimed at minimizing the overall control effort. Incorporating the system dynamics derived in Section 2 along with the required boundary conditions, the formulation is defined as:

$$\begin{aligned} \underset{\mathbf{x}(t),\mathbf{u}(t)}{\text{minimize}} \quad & \int_0^{t_f} \mathbf{u}(t)^2 \, dt \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \\ & \mathbf{b}[\mathbf{x}(t_0), \mathbf{x}(t_f)] = 0 \end{aligned} \tag{5}$$

Here, the control input $\mathbf{u}(t)$ represents either the longitudinal jerk $j(t)$ or the angular acceleration $r(t)$, depending on the considered motion.

The algorithm then follows the steps of the PMP, and the resulting system of canonical differential equations is integrated to obtain a closed-form analytical solution for the motor primitive.

3.2. Salience Calculation

Subsequently, the jerk-minimized trajectories are integrated into the decision-making layer to determine their control *salience*. By evaluating this salience metric, the system assesses the quality of the candidate trajectories relative to environmental constraints, ensuring that the selected control output is both mathematically optimal and context-aware.

To evaluate these trajectories in real time, the continuous control space is discretized. Specifically, a finite pool of candidate trajectories is generated by sampling the initial control inputs $[j_0, r_0]$ across a bounded operational interval.

The salience calculation for these sampled candidates is divided into distinct motion categories. These consist of the longitudinal and lateral (Section 2.1) dynamics of the bicycle model, alongside a specialized, rotation-prevalent alignment behavior belonging to the unicycle configuration (Section 2.2). Finally, the total salience is computed as a weighted sum of the longitudinal contribution and either the lateral or the alignment contribution, which dynamically compete within the same control variable manifold.

3.2.1. Longitudinal

The main goal of the longitudinal layer is to guide the vehicle from its initial state to the target state without exceeding the speed limit. Thus, its salience computation consists of a weighted sum of multiple items: some enforce the road constraints, others induce low jerk solutions, and others consider the physical limits of the chosen vehicle.

More specifically, the salience components for the longitudinal motion are: (i) the maneuver duration from an initial condition to a final one; (ii) the OCP cost $J = \int_0^{t_f} j(t)^2 \, dt$; and (iii) a velocity constraint penalty.

The first component (i) includes the optimal offline trajectory in the salience computation $j_{opt}(T, \mathbf{x}_0, \mathbf{x}_f)$. In fact, the temporal efficiency of the maneuver is determined by finding T that satisfies:

$$j_{opt}(T, \mathbf{x}_0, \mathbf{x}_f) = j_0 \tag{6}$$

where j_0 represents the discretized jerk value evaluated. This process is seen as a root-finding problem; thus, it is solved through the bisection method.

While these constraints could theoretically be integrated directly into the OCP formulation, doing so would eliminate the analytical closed-form solutions and require computationally intensive online numerical optimization.

3.2.2. Lateral

The lateral motion aims at avoiding deviating from the reference path. As for the longitudinal case, the lateral salience is a weighted sum of different components. In particular, it accounts for: (i) the time to reach the road boundary; (ii) the time required to re-enter the bounds if the agent is located outside; and (iii) the initial lateral jerk r_0. Similarly to the previous case, the time required to go from the initial to the final conditions, T is found through a variant eq. (6), but for the lateral control:

$$r_{opt}(T, \mathbf{x}_0, \mathbf{x}_f) = r_0 \tag{7}$$

The framework also allows for the integration of multiple lanes. Rather than implementing specific lane-changing logic, the salience is computed identically for every available lane, and the resulting values are combined using a min operator. By introducing the variable $s_{lat}(r_0)$ to represent the salience for the lateral control r_0, the global lateral salience for the whole multilane road can be defined as:

$$s_{lat}(r_0) = \min_{i \in \text{lanes}} w_i \cdot s_{lat_i}(r_0) \tag{8}$$

This approach allows for lane changes to occur naturally, as the system always selects the lane with the lowest salience value. In this context, the weights (w_i) act as a biasing factor: for instance, assigning a lower weight to the right-hand lane causes the vehicle to stay or move in that direction, even if the costs of the lanes are similar.

3.2.3. Alignment

The alignment behavior is applied when the vehicle is strongly skewed from the centerline direction. The main goal of this rotation-heavy motion is to straighten the vehicle and reduce the heading error to near zero. To evaluate this maneuver, the alignment salience accounts for: (i) the time needed to align with the road direction while bringing the angular velocity to zero; (ii) the analytical OCP cost $J = \int_0^{t_f} r(t)^2 \, dt$; (iii) the current heading error; (iv) an angular velocity constraint to prevent excessive spinning; and (v) a heading overshoot penalty to ensure a smooth stabilization. Similar to the longitudinal and lateral cases, the time required to complete the alignment in component (i) is calculated online by solving the root-finding problem in eq. (7).

3.3. Motor Cortex

Once the longitudinal and lateral saliences are determined, they are combined to define the motor cortex. Inspired by the affordance-based architecture proposed in [2], this map provides a topographic organization of the motion space, where alternative driving actions are represented as competing salience distributions over a two-dimensional grid. Specifically, each cell in the grid corresponds to a unique combination of longitudinal and lateral jerk, with its value representing the total salience of that maneuver. Grouping similar actions in close proximity ensures the planner's explainability, as the relationship between specific control inputs and their environmental costs, such as those arising from road boundaries, can be clearly visualized. While the original formulation established this topographic concept, this work extends the principle to integrate heterogeneous planning behaviors and vehicle models into a single, unified decision-making manifold.

The size of the motor cortex is 81×81, where the null element is in position $[41, 41]$, and the range for the control inputs is ± 1.0 m s^{-3} in the longitudinal case, while ± 5.0 rad s^{-2} for the lateral one. These boundaries ensure smooth acceleration changes, reflecting a non-aggressive driving style. Furthermore, this specific matrix size provides a comprehensive discretization of the control interval while remaining computationally lightweight.

3.4. Behaviours

A key strength of this approach lies in its modularity. This feature allows diverse behaviors and scenarios to be implemented interchangeably using the same underlying motor cortex concept. While the validation in this work focuses on structured and unstructured driving, which are further discussed in the following sections, this specific selection does not limit the broader scope of potential applications. These base behaviors can be merged in various combinations, enabling multiple affordances to be evaluated simultaneously, in alignment with Cisek's affordance competition hypothesis [1].

3.4.1. Structured

The structured scenario is defined by a global reference path (road-like scenario), which can accommodate either single- or multi-lane configurations. Within this environment, the vehicle is required to strictly respect the road boundaries while minimizing jerk to ensure smooth tracking behavior.

3.4.2. Unstructured

In contrast to the structured scenario, the unstructured case does not assume a global reference path. Instead, the environment is defined by a series of discrete gates, each characterized by three primary attributes: a central position, a target orientation, and a gate width. This configuration enables the vehicle to navigate based on local objectives rather than a fixed road geometry. Each gate is associated with a clothoid curve that connects the current vehicle position to the target. These curves serve as local reference paths; however, unlike the structured environment, a separate path is generated for each gate. Consequently, all gates are evaluated simultaneously. The individual motor cortices associated with each candidate path are merged to construct a single, unified motor cortex for final action selection.

3.4.3. Mixed

The integration of the two previous behaviors, structured and unstructured, is performed in the mixed environment. In this scenario, the road is accounted for as a physical constraint from which the vehicle should not stray, and the gates serve as local goals. Thus, if a path leading to a gate is found to be outside the road boundaries, that maneuver is temporarily discarded. However, since the route is dynamically recomputed at each step, a gate that is currently unreachable may become feasible as the vehicle progresses, allowing it to be reconsidered in future iterations. Also, in this case, the behavior returns a single motor cortex, in which all the possible actions are simultaneously considered.

3.4.4. Alignment

Finally, the remaining behavior considered in this framework is the alignment behavior. This mode generates a purely rotational motion that enables the vehicle to rapidly correct its heading when it is severely deviated from the road direction. Due to the kinematic assumptions inherent to the unicycle model (Section 2.2), this behavior is strictly restricted to low velocities, as executing sharp rotations at higher speeds could result in unstable or unpredictable vehicle dynamics. Moreover, the alignment behavior cannot function independently; it must always operate concurrently with one of the root behaviors because it requires an underlying reference path. To avoid introducing excessive uncertainty, this behavior is merged pairwise with the active base scenario to generate a single, unified motor cortex for final action selection.

3.5. Action Selection

The final stage of the proposed architecture is the action selection process, which reduces aggregated motor cortex information to a single control command. For this purpose, the global minimum of the salience map is retrieved through a *winner takes all* procedure.

However, because the discretization of the motor cortex can result in multiple maneuvers sharing the same optimal cost, a secondary selection is implemented. In these cases, the system shifts its focus from external navigation goals to internal vehicle dynamics, selecting the candidate that minimizes jerk. By employing this sequential logic, the agent guarantees a unique solution for the control input while optimizing for both environmental constraints and passenger comfort.

4. Experimental Results

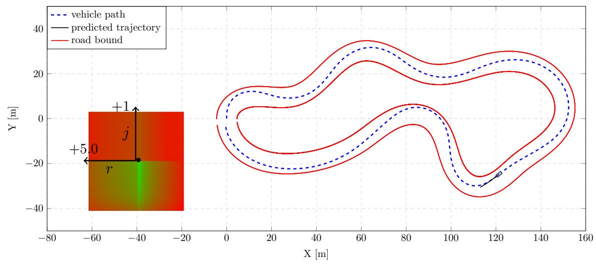
To evaluate the performance of this framework, the different scenarios are tested in a custom simulated environment developed in Rerun [10]. To avoid introducing extra uncertainty or low-level tracking errors, the low-level controllers are bypassed, and the planned control inputs are directly integrated into the vehicle's state update model.

The evaluation includes two base scenarios, structured and unstructured, alongside three merged variations: structured-unstructured (mixed), alignment-structured, and alignment-unstructured. Testing these combinations validates the flexibility of the framework and demonstrates how effectively the motor cortex layer merges competing behaviors.

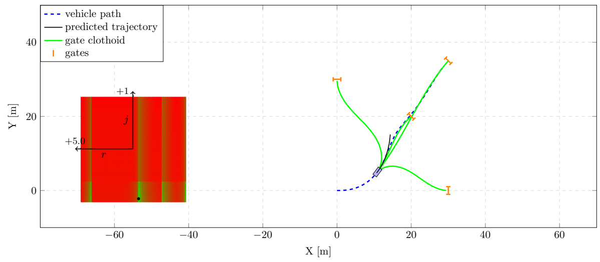
Depending on the present behaviors within these scenarios, the vehicle updates its state using different kinematic constraints. When the alignment behavior is present, the system uses the unicycle model to allow for in-place rotation. For all other scenarios, the vehicle state is updated using the standard bicycle model.

4.1. Base Scenarios

Figure 2 illustrates the performance of the framework within the base scenarios.

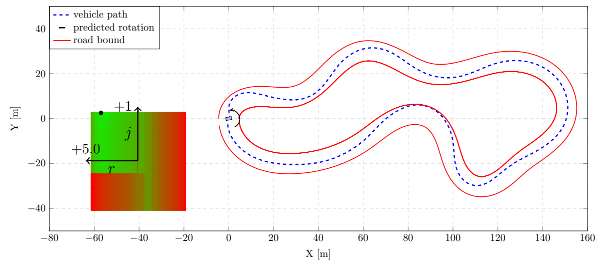


(a). Performance in the structured environment, where the vehicle path (blue dotted line) remains within the road bounds. A snapshot highlights the current vehicle position, the predicted trajectory, and the corresponding motor cortex visualization.

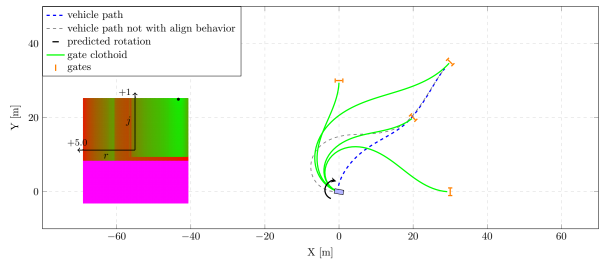


(b). Performance in the unstructured environment, where the possible gates and their associated trajectories are displayed. As in the previous case, a snapshot highlights the current vehicle position, the predicted trajectory, and the corresponding motor cortex visualization, where the peaks of salience produced by the different gates are clearly visible.

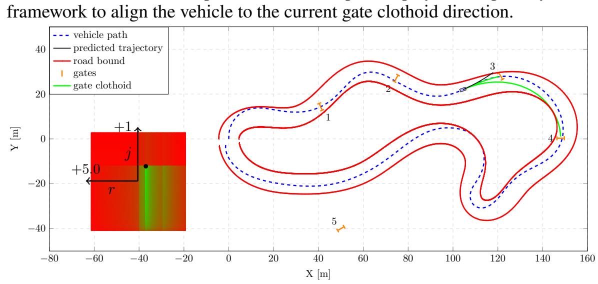
Fig. 2: Performances in the base scenarios



(a). Performance of the integration of the alignment behavior inside the structured scenario. The plot displays the capability of the framework to align the vehicle to the current road direction.



(b). Performance of the integration of the alignment behavior inside the unstructured scenario. As in the previous case, the plot displays the capability of the framework to align the vehicle to the current gate clothoid direction.



(c). Performance of the merging of the structured and the unstructured scenarios. The plot displays a correct navigation of the vehicle, as it passes through the gates that allow it to stay on the road, while it bypasses the ones that would lead it to stray from it.

Fig. 3: Merged behaviours

Structured Environment In the structured environment (Figure 2a), the results demonstrate accurate path-following, as the vehicle remains strictly within the road boundaries. The motor cortex visualization represents the pool of candidate actions considered by the vehicle; here, a slight right-turn maneuver is correctly selected as the winning trajectory to match the right-hand curve of the road. The plotted predicted trajectory further confirms that the vehicle properly follows the road constraints. Due to the minimum jerk objective, the vehicle occasionally approaches the road boundaries closely, as observed in the two subsequent curves at the bottom of the track. This behavior occurs because the planner prioritizes motion smoothness over aggressive centerline tracking, allowing the vehicle to utilize the available road width to minimize control effort.

Unstructured Environment For the unstructured scenario, Figure 2b shows the test environment with multiple target gates and their associated clothoid curves. Here, the motor cortex visualization displays four distinct candidate actions, represented by the four green regions in the matrix, which correspond to each gate, and highlights the selection of the trajectory requiring the minimum control effort. This choice is supported by the predicted trajectory, which points directly toward the chosen gate. Due to the assumption of steady-state curvature, the predicted trajectory aligns closely with the path at the beginning but may show some deviations toward the end of the horizon. However, this does not degrade tracking performance because of the receding-horizon nature of the planner, which constantly recomputes the trajectory at a high frequency.

4.2. Merged Scenarios

To evaluate the flexibility of the motor cortex, the base scenarios are merged with the alignment behavior as well as with each other. This merging process is restricted to two behaviors at a time to avoid introducing additional uncertainty. The resulting performance across these different combinations is depicted in Figure 3.

Alignment in structured and unstructured environments Integrating the alignment behavior enables the vehicle to realign with the road when its heading is severely deviated. This condition occurs exclusively at the start of both scenarios, due to the low velocity assumption discussed in Section 3.4.4, where the vehicle is initialized with a large heading error. This initial phase is illustrated in Figures 3a and 3b, which display snapshots captured during the first instants of the simulation. While the overarching objective of these merged scenarios remains identical to the base cases, an initial alignment maneuver takes place first. The plots demonstrate that the vehicle correctly predicts the required rotation direction to achieve realignment. Specifically, although the current vehicle orientation (represented by the blue rectangle) is heavily skewed relative to the road, the resulting trajectory (the blue path) successfully aligns with the road direction. This confirms that while the vehicle is misoriented at the exact moment of the snapshot, it subsequently executes the predicted rotation to correct its heading and complete the task. To demonstrate the strength of this behavior, Figure 3b also contrasts this result with a baseline trajectory generated under identical initial conditions but without the alignment behavior enabled (the gray path).

Mixed Scenario In the mixed scenario, the structured and unstructured behaviors are merged, which ensures the vehicle remains within the road boundaries while simultaneously passing through the designated gates. The performance illustrated in Figure 3c reflects this dual objective: the vehicle path stays strictly within the road bounds and successfully clears the available gates, with the sole exception of gate 5. Since gate 5 is located outside the road boundaries, the planner correctly bypasses it, as attempting to reach it would violate the safety constraints of the structured road layout. This behavior is further confirmed by the snapshot of the motor cortex and the planned trajectory. The visualization reveals that while candidate actions for both gates 3 and 4 are evaluated simultaneously, gate 3 is ultimately selected due to its higher proximity.

5. Conclusion

This work presents a modular, salience-based framework capable of integrating multiple operational scenarios and vehicle models within a unified pipeline. The decoupled nature of this architecture allows for the seamless modification of individual pipeline blocks without requiring changes to the rest of the implementation. To validate this approach, both unicycle and bicycle models were evaluated across a series of environments. From a broader perspective, the proposed planner can be interpreted as an evolution of the affordance competition architecture presented in [2]. While the original framework focused on the emergence of complex driving behaviours through the interaction of competing affordances, the present work generalizes the same principles to support multiple operational scenarios and multiple vehicle models , as validated here using both unicycle and bicycle models.

The experimental results demonstrate the robustness and flexibility of the framework across distinct driving conditions. In structured environments, the system successfully enforces road boundaries as a hard constraint while minimizing jerk to ensure smooth trajectory tracking. In unstructured scenarios, the vehicle navigates discrete gates by generating and weighting localized clothoid paths, leveraging the motor cortex to select the lowest-cost action. Furthermore, the mixed scenarios highlight the planner's ability to safely prioritize competing objectives, executing gate-seeking behaviors while correctly bypassing targets that violate the overriding road boundary constraints. Finally, the alignment behavior proves highly effective at correcting severe initial heading errors at low velocities, stabilizing the vehicle during initialization phases.

By managing and combining these behaviors, the framework provides a practical realization of Cisek's affordance competition hypothesis, enabling the simultaneous evaluation of multiple environmental affordances while maintaining interpretability. Future work will focus on developing a more general merging procedure to establish a unified methodology and on introducing a dedicated obstacle avoidance module.

6. Acknowledgments

This work was supported under the FIS 2 Call, Grant Assignment Decree No. 1236 adopted on 01/08/2023 by the Italian Ministry of University and Research (MUR), for the project FIS-2023-03684 'Structured neural network framework for modeling and control of autonomous systems – Neu4mes', CUP E53C24003800001. The authors would like to thank Prof. Mauro Da Lio, whose previous work on affordance competition and layered control architectures laid the foundations for the present research. The authors are also grateful for his valuable feedback and insightful comments during the development of this work.

References

(1) P. Cisek. Cortical mechanisms of action selection: the affordance competition hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485):1585–1599, 04 2007.

(2) M. Da Lio, A. Cherubini, G. P. Rosati Papini, and A. Plebe. Complex self-driving behaviors emerging from affordance competition in layered control architectures. *Cognitive Systems Research*, 79:4–14, 2023.

(3) M. Da Lio, R. Dona, G. P. Rosati Papini, and K. Gurney. Agent Architecture for Adaptive Behaviors in Autonomous Driving. *IEEE Access*, 8:154906–154923, 2020.

(4) R. Donà, G. P. Rosati Papini, M. Da Lio, and L. Zaccarian. On the stability and robustness of hierarchical vehicle lateral control with inverse/forward dynamics quasi-cancellation. *IEEE Transactions on Vehicular Technology*, 68(11):10559–10570, 2019.

(5) S. M. Grigorescu, B. Trasnea, L. Marina, A. Vasilcoi, and T. Cocias. Neurotrajectory: A neuroevolutionary approach to local state trajectory learning for autonomous vehicles. *IEEE Robotics and Automation Letters*, 4(4):3441–3448, 2019.

(6) C. J. Holder and T. P. Breckon. Learning to drive: Using visual odometry to bootstrap deep learning for off-road path prediction. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 2104–2110. IEEE, 2018.

(7) C. Li, H. Chen, J. Sun, W. Fang, V. Palade, and X. Wu. M2pp: Effective path planning based on multi-phase particle swarm optimization and multi-scenario adaptative dwa. *IEEE Transactions on Automation Science and Engineering*, 23:2717–2733, 2026.

(8) M. Piazza, M. Piccinini, S. Taddei, and F. Biral. Mptree: A sampling-based vehicle motion planner for real-time obstacle avoidance. *IFAC-PapersOnLine*, 58(10):146–153, 2024.

(9) L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. The mathematical theory of optimal processes. Authorized translation from the Russian. Translator: K. N. Trirogoff. Editor: L. W. Neustadt. New York and London: Interscience Publishers, a division of John Wiley & Sons, Inc., viii, 360 p. (1962)., 1962.

(10) Rerun Development Team. Rerun: A visualization sdk for multimodal data, 2022. Available at https://github.com/rerun-io/rerun.

(11) K. Sim, J. Kim, and J. Gim. A kinematics-constrained grid-based path planning algorithm for autonomous parking. *Applied Sciences*, 15(20):11138, 2025.

(12) Z. Wang, M. Zhang, S. Liang, S. Yu, C. Zhang, and S. Du. A multi-objective path-planning approach for multi-scenario urban mobility needs. *Algorithms*, 18(1), 2025.

(13) S. Zhang, Y. Chen, S. Chen, and N. Zheng. Hybrid a-based curvature continuous path planning in complex dynamic environments. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1468–1474. IEEE, 2019.

(14) X. Zhang, Z. Zang, X. Chen, Y. Lu, J. Qi, and J. Gong. Optimization-based trajectory planning for autonomous vehicles in scenarios with multiple reference lines. *Control Engineering Practice*, 163:106407, 2025.

(15) J. Zhou, B. Olofsson, and E. Frisk. Interaction-aware motion planning for autonomous vehicles with multi-modal obstacle uncertainty predictions. *IEEE Transactions on Intelligent Vehicles*, 9(1):1305–1319, 2023.