

Appendix: Supplementary Material for A Multi-Modality Evaluation of the Reality Gap in Autonomous Driving Systems

1 Framework details

1.1 Sensor and LiDAR Details

Depth sensing is performed using an Apple iPhone 15 Pro’s Time-of-Flight (ToF) sensor. The Record3D app, along with its Python API [3], is used to retrieve depth images and the sensor’s intrinsic matrix. The device is connected via USB to the Jetson Nano, enabling real-time transfer of both RGB and depth streams. Point clouds are generated using a pinhole camera model, and the output is published as `sensor_msgs/PointCloud2` messages in ROS. The LiDAR center is aligned with the ToF sensor, and transformation frames are published accordingly to maintain consistent spatial alignment.

1.1.1 Modules interactions

Table 5 presents all messages passed between the different nodes of our framework. While Figure 15 shows a conceptual overview of how different ADs can be integrated within the framework.

2 ADSs Integration

2.1 Ground-Truth NPC Driver

To collect ground-truth driving behavior used in **RQ3** (lane assignment accuracy) and for training our end-to-end ADS, we implemented a simplified, non-learning control system. This system mimics a modular pipeline but replaces perception and planning with ground-truth data.

Specifically, it uses the globally referenced ground-truth positions of obstacles and a set of predefined waypoint sequences that span different portions of the lane. The controller continuously monitors obstacle positions: if an obstacle is detected within a threshold distance of 0.8,m, the system switches to an alternate waypoint sequence corresponding to the opposite side of the lane (e.g., from left to right). This behavior ensures idealized, deterministic obstacle avoidance.

The selected global waypoint list is passed directly to the waypoint follower module, which executes it using the standard pure-pursuit and PID speed control stack. This approach yields clean, repeatable driving trajectories that represent ideal behavior, making it suitable for both evaluation baselines and training supervision.

2.1.1 E2E Integration

End-to-end (E2E) models use sensor data—typically camera images—as input and directly output control commands (e.g., steering and throttle) without explicit modular decomposition. In our framework, E2E models are integrated by subscribing to the appropriate camera data stream—real,

simulated, or mixed—and publishing control commands to the car actuation interface. These commands include throttle, steering, and brake signals.

No modifications to the internal architecture of the E2E model are required. Integration is performed via a lightweight ROS wrapper that subscribes to image topics and publishes the predicted control outputs.

2.1.2 Our E2E integration

Our E2E ADS is trained using 50 laps recorded in the Room Nominal track at 60 FPS, ensuring high temporal resolution. The input consists of RGB camera images captured from the vehicle’s front-facing sensor, synchronized with expert steering labels generated by our Ground-Truth NPC Driver. To enhance generalization, we applied horizontal flipping to all samples, effectively doubling the dataset to approximately 180,000 labeled frames. Several laps included controlled off-track episodes and recovery maneuvers to expose the model to failure scenarios and reinforce robust corrective behavior. We followed the training methodology of Bojarski et al.[1]. The model was trained using the Adam optimizer[2] to minimize mean squared error (MSE) between predicted and expert steering angles. Training used a batch size of 64, a learning rate of 0.0001, and was capped at 500 epochs. Early stopping was applied using a patience of 30 epochs and a minimum validation loss improvement threshold of $\Delta\text{MSE} < 0.0005$. The final model achieved a validation MSE of 0.02 and 90% accuracy in predicting steering angles within a predefined tolerance.

2.1.3 Modular Pipelines Integration

A modular pipeline SUT separates functionality into distinct components for perception, planning, and control, allowing each to focus on a specific subtask within the autonomous driving stack. The framework supports this modularity by enabling each component to subscribe to and publish on relevant ROS topics. Perception modules can subscribe to camera or LiDAR data—whether from real, simulated, or mixed-reality sources—to detect obstacle poses in the vehicle’s frame of reference. These poses can be transformed into the world frame using either the tracking system or the simulator’s world-to-vehicle transform. Planning modules can receive the current vehicle pose from the tracking system or simulator, and obstacle poses either from the perception module or directly from ground truth data provided by the framework. Using this information, they can compute collision-free trajectories represented as sequences of waypoints. A planning module can either send one waypoint at a time to the Waypoint Following Node, along with a target speed to the PID speed controller or transmit the complete waypoint list to the next modular component: control. The control module can either directly publish throttle, brake, and steering commands to the vehicle via the car actuation interface or simulator interface, or utilize the Waypoint Following Node in combination with the PID speed controller to generate these commands. The choice depends on the desired level of abstraction and control architecture.

2.1.4 Our Modular Pipeline integration

For our modular ADS, we implement a 3-modules architecture with a LiDAR perception module, a Lattice planner, and the pure-pursuit and PID speed modules of the framework for control. *I) Perception:* The LiDAR perception module processes raw 3D point cloud data to detect and localize obstacles in the environment. In the first step, ground removal is performed using iterative RANSAC-based plane segmentation. Planes are accepted and removed if they have more than 5,000 inliers and a tilt angle less than 15° from the horizontal plane. In the second step, region of interest filtering is applied to isolate the area immediately surrounding the vehicle. A 3D spatial

crop box retains only those points within the bounds of X (vertical) = [-0.5, 1]m, Y (horizontal) = [-10, 10]m, and Z (depth) = [0, 2]m. The restriction along the X-axis removes tall vertical structures such as room pillars and walls, which are irrelevant for obstacle detection in our small-scale vehicle scenario. Similarly, Z-axis cropping limits the field of view to nearby objects, filtering out distant infrastructure like room boundaries. In the third step, the filtered point cloud is clustered using the DBSCAN algorithm, which groups points based on spatial density. Clusters are formed using a neighborhood radius ε of 0.1m and a minimum of 100 points per cluster. This method identifies distinct obstacle candidates while rejecting sparse noise. In the fourth step, each cluster is reduced to a centroid by averaging the positions of its constituent points. These centroids, initially in the LiDAR coordinate frame, are transformed into the world (map) frame using TF lookups based on the vehicle's real-time pose obtained from the tracking system. This final transformation provides globally consistent obstacle positions, which are used as input by the planning and control downstream modules.

II) Planning: The planning module uses a lattice-based approach to generate collision-free trajectories based on the current vehicle state. It receives the vehicle's pose and velocity—sourced from the tracking system or simulator—as well as globally referenced obstacle positions from the perception module. The planning process begins by converting the vehicle's Cartesian pose into Frenet coordinates with respect to the reference line, which corresponds to the map's lane center. From this Frenet state, the planner samples candidate trajectories by varying terminal lateral offsets and speeds, using quintic polynomials for lateral motion and quartic polynomials for longitudinal motion. Each candidate path is evaluated for feasibility and assigned a cost. Trajectories that violate constraints—such as exceeding road boundaries (beyond a 30% threshold), colliding with known obstacles, or surpassing maximum speed, acceleration, or curvature limits—are discarded. Remaining feasible paths are scored based on jerk, deviation from the centerline, and potential collisions. The trajectory with the lowest total cost is selected as the optimal path. This optimal trajectory is discretized into a sequence of waypoints, which are then passed to the control module for execution.

III) Control: The control module receives the list of waypoints and, using the current vehicle pose, identifies the next closest waypoint along the trajectory. It then publishes this waypoint to the waypoint following module, along with a predefined target speed. This target speed is either set via the user interface or defined through fixed configuration parameters. The waypoint following module, in conjunction with the PID speed controller, then generates and executes the low-level steering, throttle, and braking commands required to follow the planned trajectory.

3 Experimental Methodology Details

3.1 Tracking Accuracy Evaluation

To ensure precise alignment between the physical and virtual environments, we evaluated the accuracy of the Vicon motion capture system using a calibrated reference object. This object consisted of two rigidly mounted retro-reflective markers spaced exactly 240 mm apart. In both Room Nominal and Room Generalization, five measurements were taken at 30-second intervals. The mean measured distance was 240 mm, with a mean absolute error of 0.40 mm and a standard deviation of 0.0253 mm. These results confirm sub-millimeter accuracy and minimal short-term drift, providing a robust foundation for sensor fusion, vehicle localization, and simulation synchronization.

3.2 RQ1: Testing Tracks

Figure 16 shows the three environments used for **RQ1**: the nominal setup (left), the generalization setup (center), and the ablation scenario (right). Each environment features a distinct track layout, physical background, and lane structure. Figure 17 provides a top-down schematic of the corresponding track geometries used in each environment.

3.3 RQ2: Waypoint Follower Commands

For the final **RQ2** experiment, we measure the Actuation Gap—and its mitigation through Vehicle-in-the-Loop (ViL) testing—under conditions where the vehicle receives direct waypoint commands. Figure 18 illustrates the six predefined waypoint configurations used in this experiment. These trajectories were selected to test a range of lane-following behaviors, including curves, lane shifts, and straight segments.

3.4 RQ3: Obstacle and Lane Annotations

To investigate both input validity and perception-domain gaps in **RQ3**, we designed two controlled experimental configurations:

- **Obstacle Configuration:** The vehicle is positioned statically at the center of the room, while one or two obstacles are placed within its sensor field of view. This allows precise evaluation of obstacle detection consistency across sensor modalities and domains.
- **Lane Configuration:** The vehicle drives on the Room Nominal track with no obstacles present. This setup isolates lane perception performance.

Figure 19 (left) depicts the eight obstacle placements used in the static setup: four single-obstacle and four dual-obstacle configurations, with placements at 40, 80, 120, and 160 cm in front of the vehicle. On the right, the figure shows the lane annotation strategy. When the vehicle travels near one side of the lane, the corresponding lane boundary is annotated as dominant. If the vehicle is centered, the middle lane line is selected as the reference. This method ensures consistent and interpretable annotations for lane-following evaluation.

4 Extended Results

4.1 RQ1: ADS Behavior Across Modalities and Tracks

Due to space constraints in the main paper, we omitted detailed trajectory plots depicting the behavior of the ADSs under different test conditions. In this section, we present comprehensive visualizations of the top-down trajectories for all experiments across both test environments and ADS architectures.

Nominal Conditions:

- Figure 1 shows the trajectory of the end-to-end ADS on track N1 (5 runs).
- Figure 2 shows the trajectory of the end-to-end ADS on track N2 (5 runs).
- Figure 3 shows the trajectory of the modular ADS on track N1 (5 runs).
- Figure 4 shows the trajectory of the modular ADS on track N2 (8 runs).

Generalization Conditions:

- Figure 5 shows the behavior of the end-to-end ADS on track G under domain shift (1 representative run).
- Figure 6 shows the behavior of the modular ADS on track G under domain shift (1 representative run).

Ablation:

- Figure 7 illustrates the behavior of the modular ADS on track N2 using ground-truth obstacle positions (1 representative run).

4.2 RQ2: Actuation Performance Across Control Modes

In the main paper, we summarized actuation behavior using averaged values for each actuation modality—throttle, steering, braking, and waypoint following—limited to at most two readings per metric, per control type, with the exception of PID, which is reported in full.

Here, we report the full raw averages, computed across five runs per configuration:

- Table 1 reports throttle values.
- Table 2 reports steering values.
- Table 3 reports braking values.
- Table 4 reports waypoint-following results.

To complement these numerical results, we include representative plots of actuation behavior:

- Figure 8 shows throttle behavior across the three configured levels (one run per level).
- Figure 9 shows steering behavior across six discrete values (one run per value).
- Figure 10 shows braking behavior for each throttle level (one run per throttle value).
- Figure 11 shows the performance of the PID controller using 5 target speeds for 10 seconds (one representative run).
- Figure 12 shows the performance of the waypoint follower across six predefined trajectories (all runs merged).

4.3 RQ3: Cross-Domain Perception Alignment

To address **RQ3**, we evaluate whether lane and obstacle poses are correctly aligned across simulation and real-world modalities. To avoid confounding effects from learned perception modules (which may be influenced by the reality gap), we perform manual annotations of visual features.

- Figure 13 presents annotated camera-based obstacle positions in scenarios with a single obstacle.
- Figure 14 shows corresponding annotations mapped onto 3D point clouds generated from the LiDAR sensor.

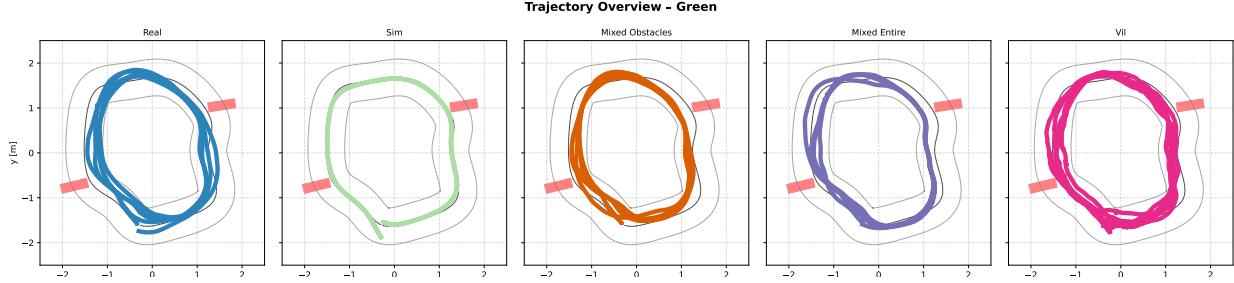


Figure 1: E2E runs on N1

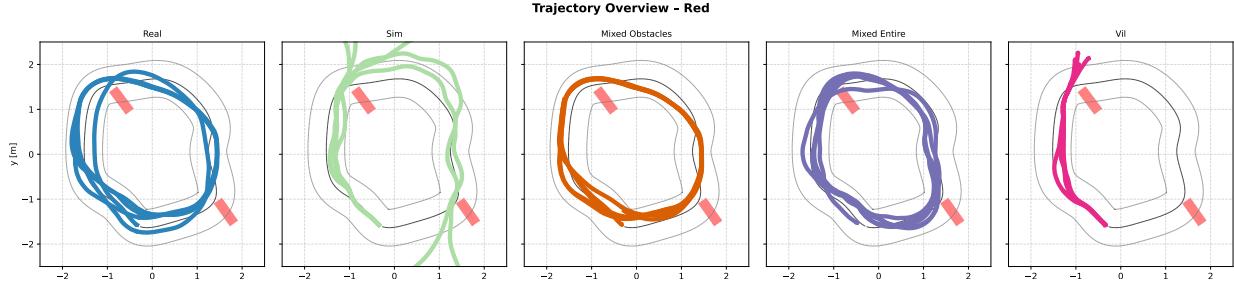


Figure 2: E2E runs on N2

These annotations allow us to isolate cross-domain alignment performance independently from ADS inference behavior.

Finally, in the paper, we aggregated perception gap metrics for the sake of space. Hereafter, the raw data is presented in Table 7

References

- [1] BOJARSKI, M., DEL TESTA, D., DWORAKOWSKI, D., FIRNER, B., FLEPP, B., GOYAL, P., JACKEL, L. D., MONFORT, M., MULLER, U., ZHANG, J., ZHANG, X., ZHAO, J., AND ZIEBA, K. End to end learning for self-driving cars. *CoRR abs/1604.07316* (2016).
- [2] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [3] SIMONIK, M. Marek-simonik/record3d: Accompanying library for the record3d ios app (<https://record3d.app/>). allows you to receive rgbd stream from ios devices with truedepth camera(s).

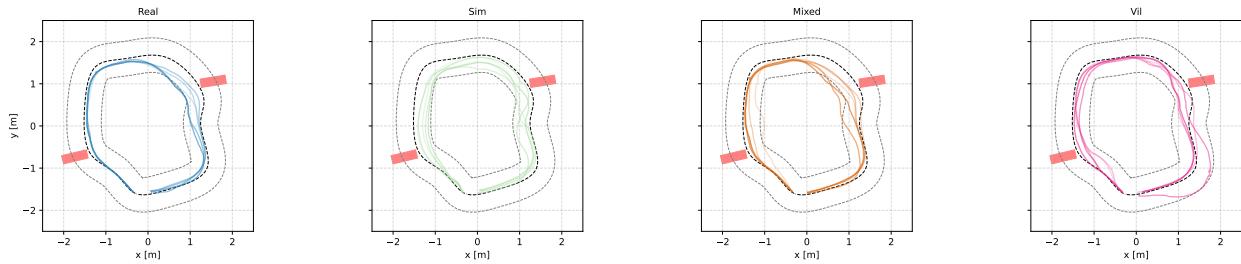


Figure 3: Modular runs on N1

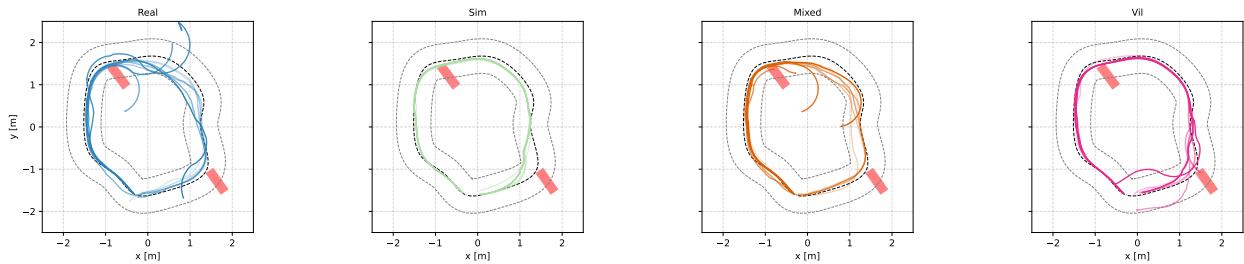


Figure 4: Modular runs on N2

t=0.34	Sim	Mapped
Distance (Δ from Real)	1.580 ± 0.075	-0.005 ± 0.203
Average Speed (Δ from Real)	-0.210 ± 0.005	0.000 ± 0.046
Trajectory Diff (Pointwise)	0.428 ± 0.059	0.006 ± 0.001
Trajectory Diff (Frechet)	0.404 ± 0.035	0.001 ± 0.000
p/d (Trajectory vs Real)	p= $1.13e-04$, d= -0.43	p= $9.97e-01$, d= -0.00
p/d (Speed vs Real)	p= $7.30e-31$, d= 0.83	p= $7.30e-01$, d= 0.02
t=0.365	Sim	Mapped
Distance (Δ from Real)	0.979 ± 0.029	-0.006 ± 0.260
Average Speed (Δ from Real)	-0.569 ± 0.003	-0.208 ± 0.064
Trajectory Diff (Pointwise)	0.210 ± 0.069	0.009 ± 0.001
Trajectory Diff (Frechet)	0.263 ± 0.067	0.001 ± 0.000
p/d (Trajectory vs Real)	p= $7.21e-02$, d= -0.21	p= $9.99e-01$, d= -0.00
p/d (Speed vs Real)	p= $3.70e-09$, d= 1.05	p= $6.80e-01$, d= 0.06
t=0.39	Sim	Mapped
Distance (Δ from Real)	0.061 ± 0.029	-0.007 ± 0.290
Average Speed (Δ from Real)	-0.502 ± 0.003	-0.001 ± 0.038
Trajectory Diff (Pointwise)	0.098 ± 0.028	0.011 ± 0.001
Trajectory Diff (Frechet)	0.089 ± 0.023	0.001 ± 0.000
p/d (Trajectory vs Real)	p= $6.49e-01$, d= -0.03	p= $9.98e-01$, d= 0.00
p/d (Speed vs Real)	p= $1.03e-93$, d= 1.35	p= $6.78e-01$, d= 0.02

Table 1: RQ1 Throttle results

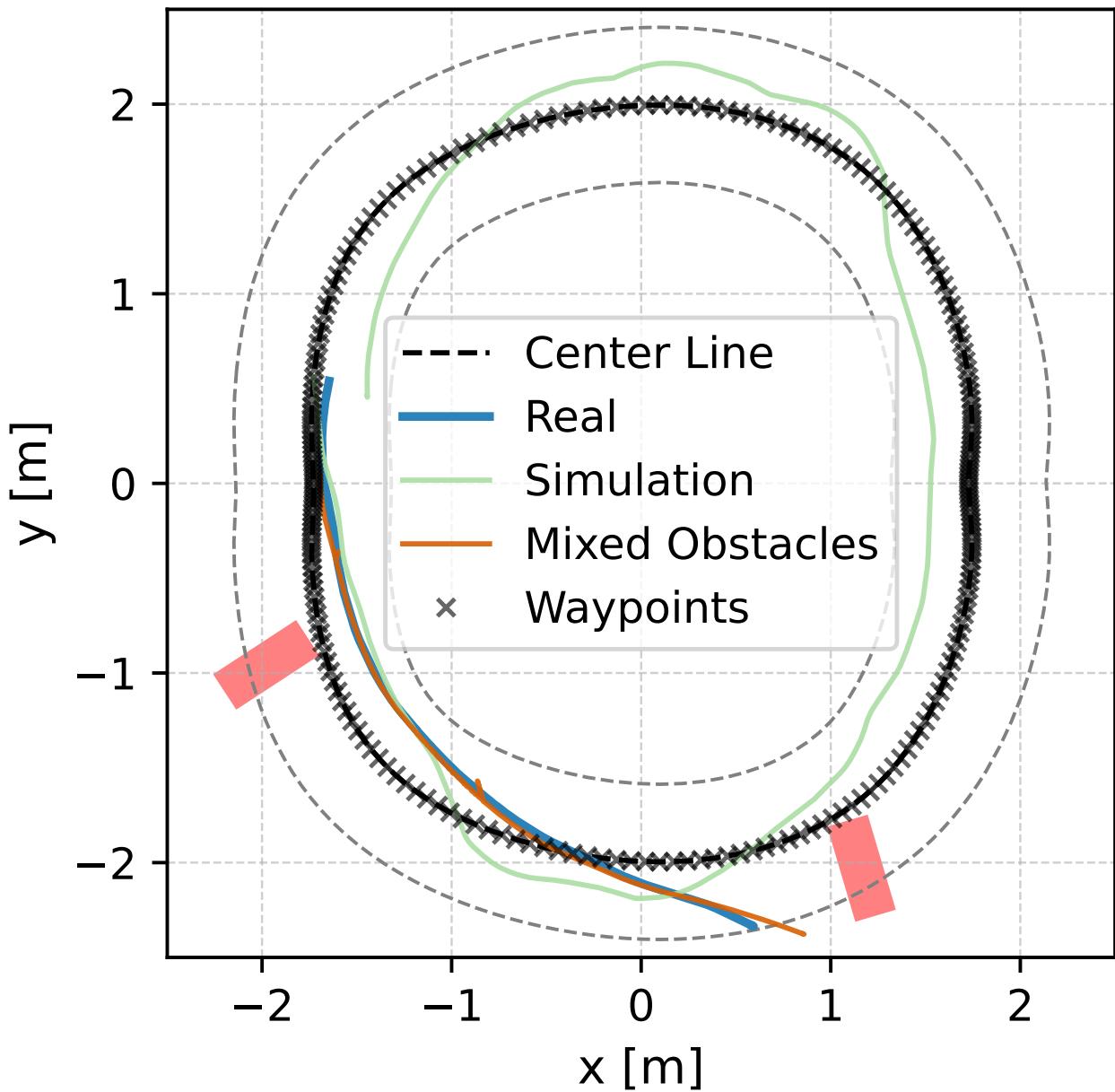


Figure 5: E2E run on G (1 representative run)

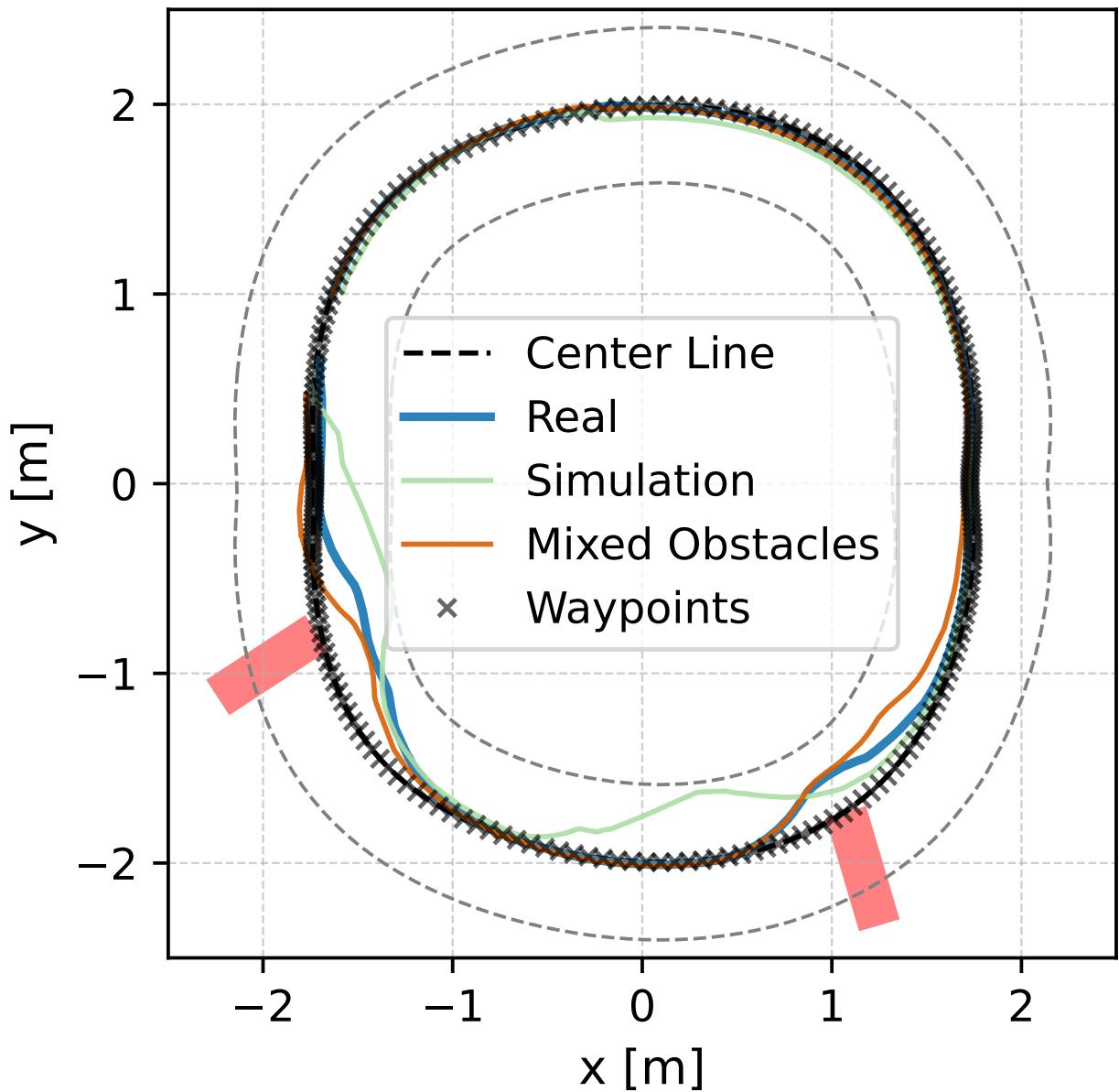


Figure 6: Modular run on G (1 representative run)

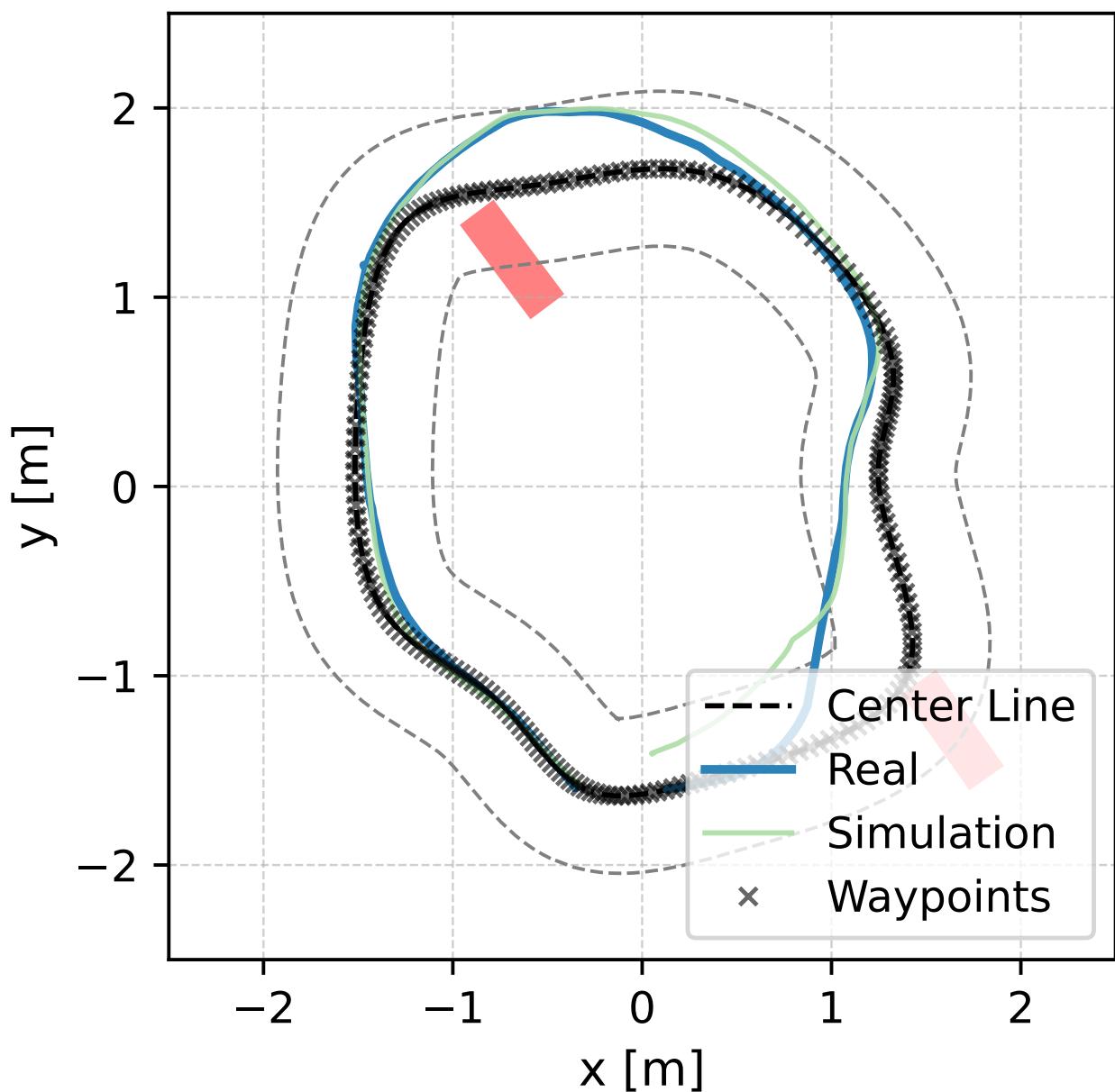


Figure 7: Modular Ablation run on N2 (1 representative run)

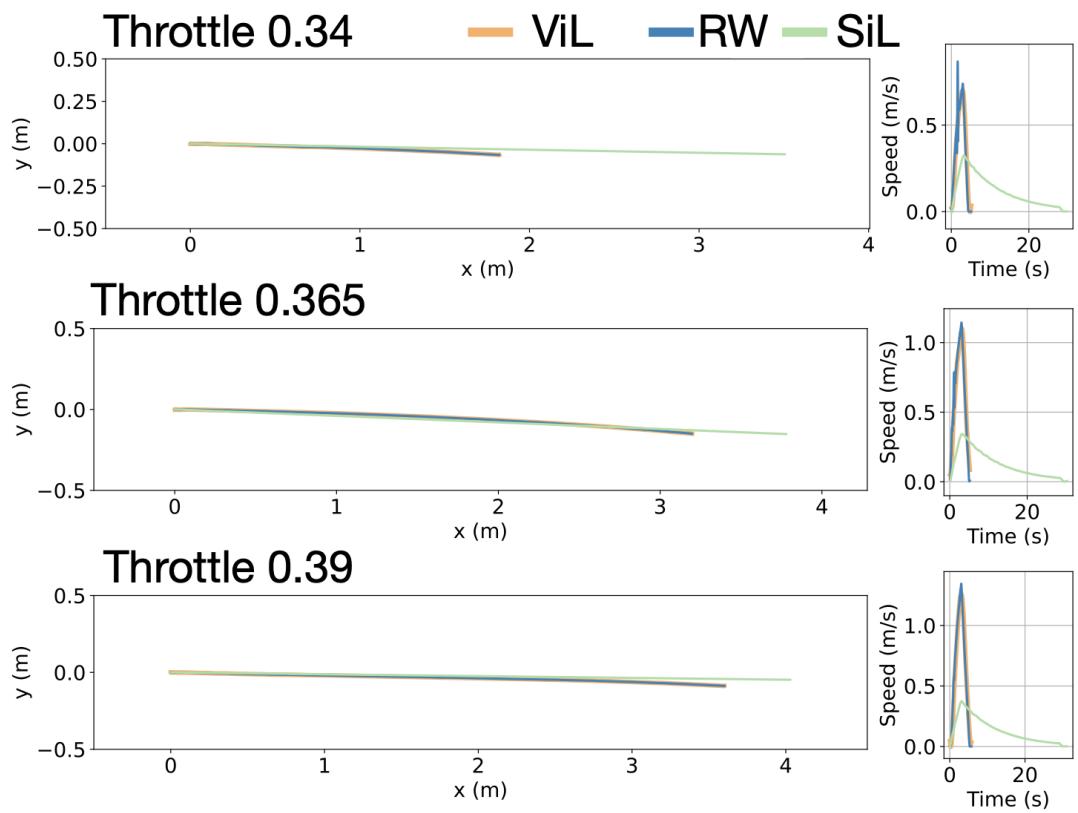


Figure 8: Throttle experiment (1 representative run)

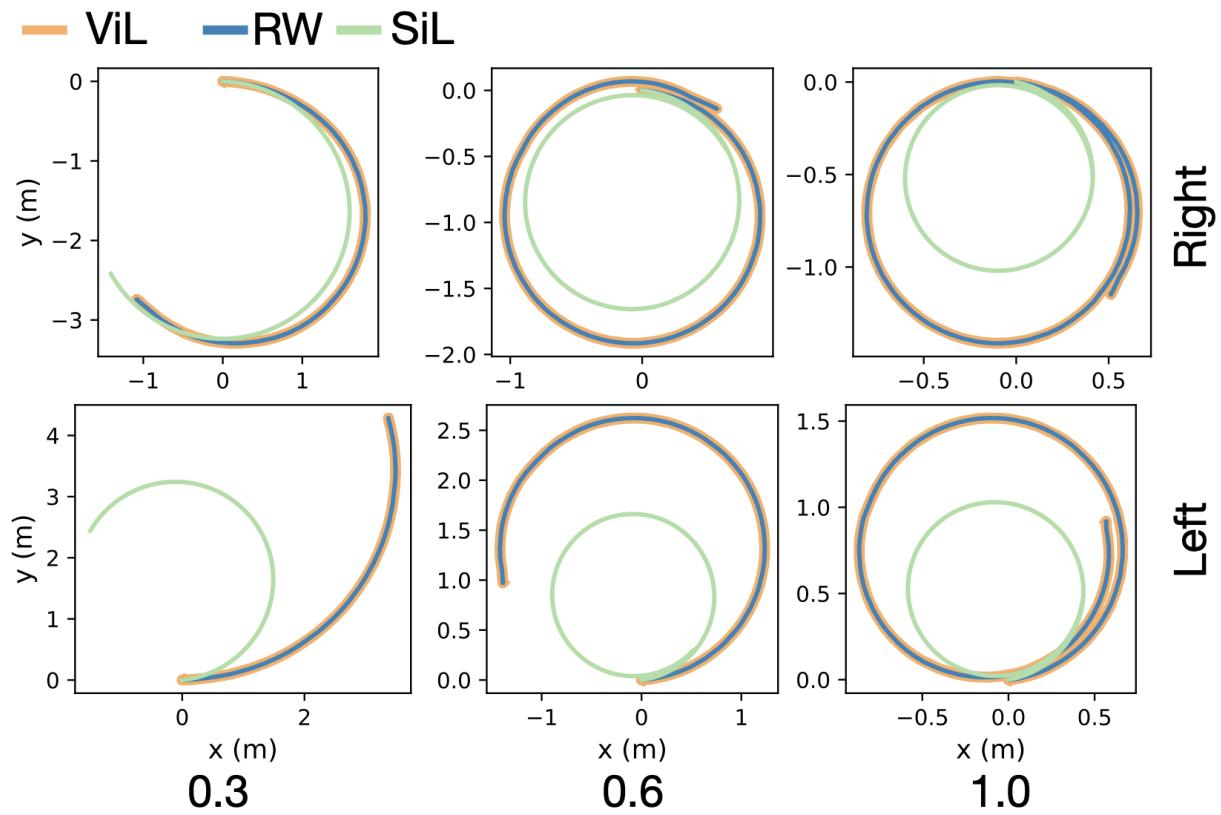


Figure 9: Steering experiment (1 representative run)

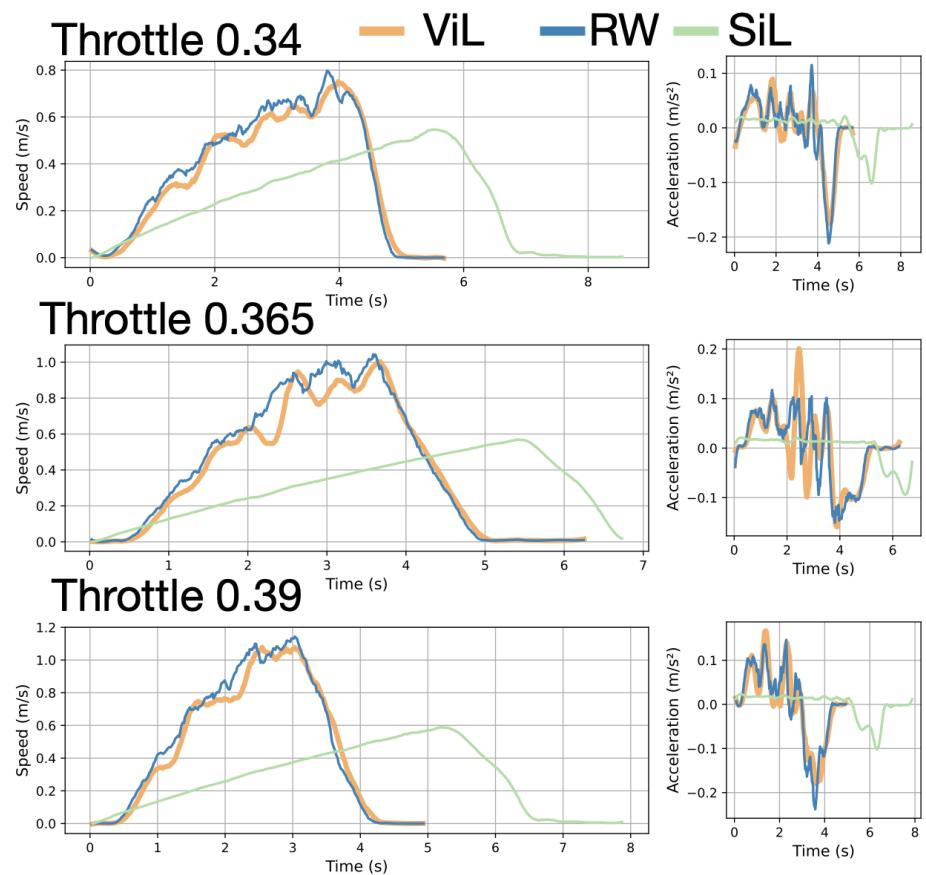


Figure 10: Braking experiment (1 representative run)

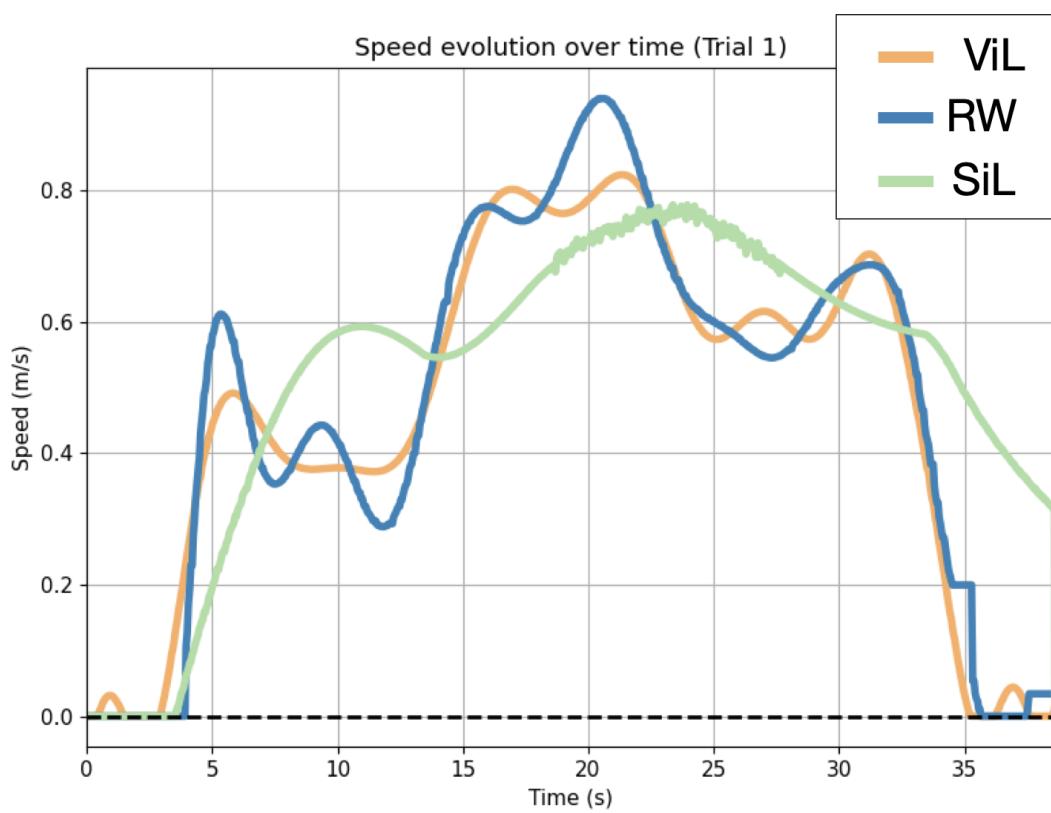


Figure 11: PID experiment (1 representative run)

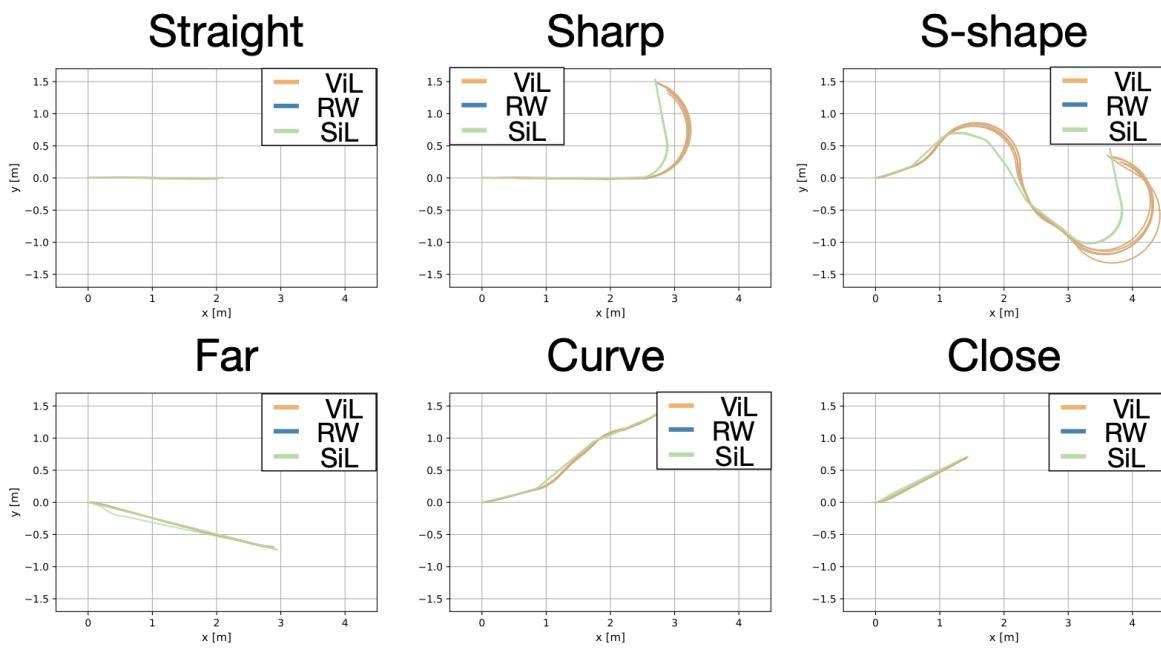


Figure 12: Waypoint Follower experiment (all 5 runs)

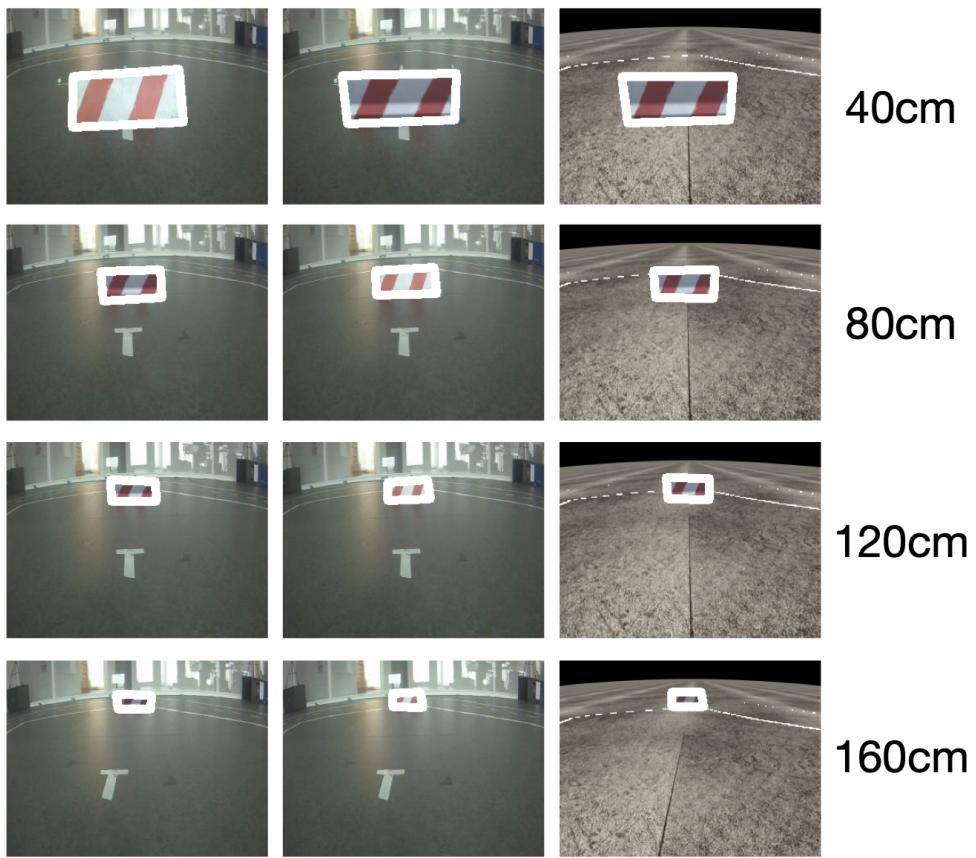


Figure 13: Obstacles annotations in the three sensing domains (camera)

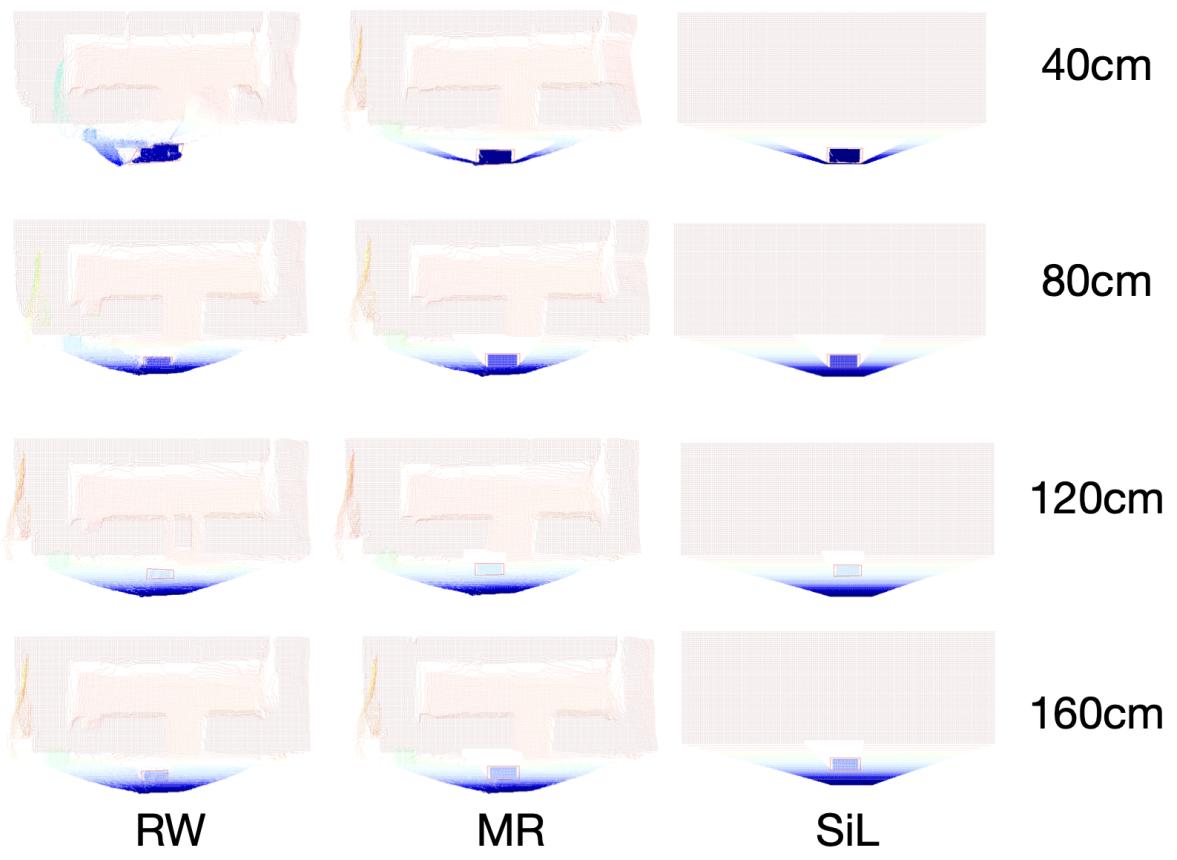


Figure 14: Obstacles annotations in the three sensing domains (LiDAR)

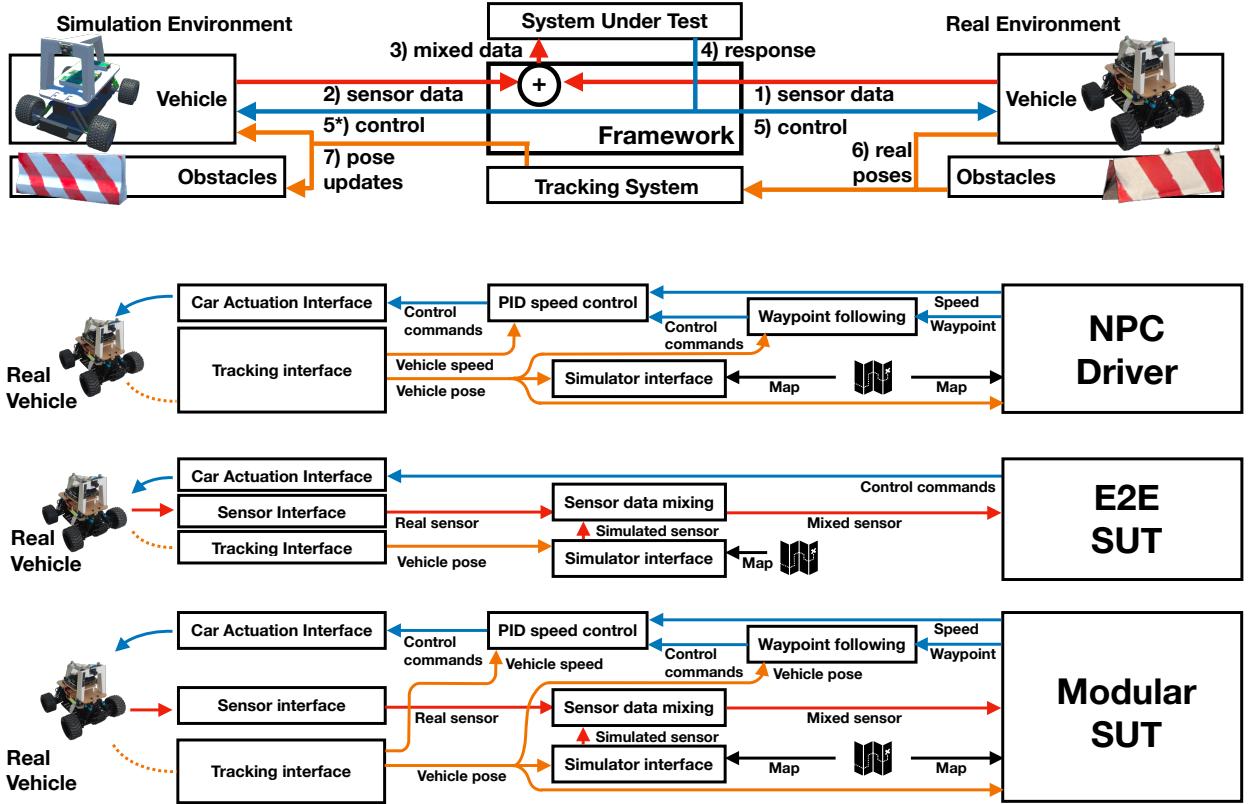


Figure 15: Examples of different ADS integrations in the framework



Figure 16: From left to right: Track N1 and Track N2 in room Nominal, and track G in room Generalization

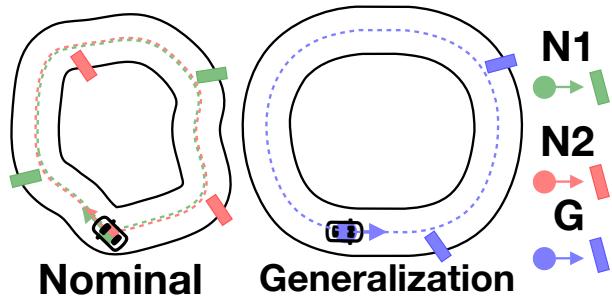


Figure 17: Top down plot of the three testing tracks

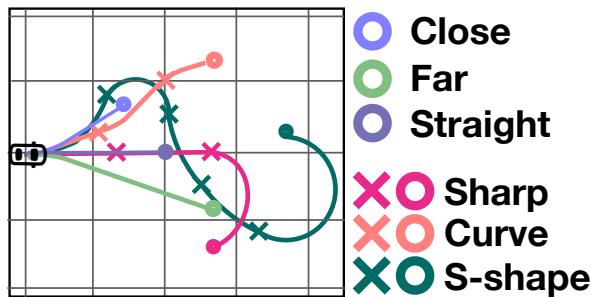


Figure 18: Top down plot of the six Waypoint Follower experiments

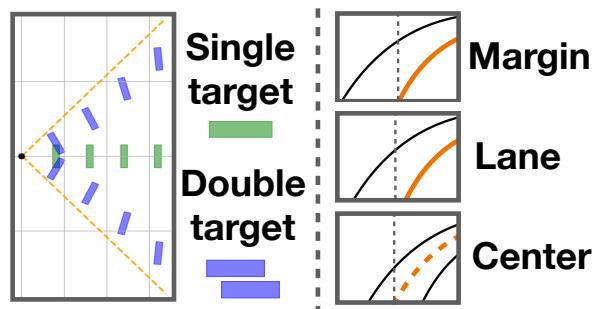


Figure 19: RQ3 configurations

s=0.3	Sim	Mapped
Radius (Δ from Real)	-0.123 \pm 0.001	0.002 \pm 0.070
Length (Δ from Real)	-0.533 \pm 0.233	-0.190 \pm 0.338
Trajectory Diff (Pointwise)	0.172 \pm 0.064	0.014 \pm 0.001
Trajectory Diff (Frechet)	0.180 \pm 0.045	0.020 \pm 0.017
p/d (Trajectory vs Real)	p=6.41e-01, d=0.03	p=9.35e-01, d=0.01
s=0.6	Sim	Mapped
Radius (Δ from Real)	-0.155 \pm 0.001	-0.001 \pm 0.007
Length (Δ from Real)	-0.838 \pm 0.183	-0.010 \pm 0.139
Trajectory Diff (Pointwise)	0.109 \pm 0.005	0.012 \pm 0.000
Trajectory Diff (Frechet)	0.114 \pm 0.014	0.001 \pm 0.000
p/d (Trajectory vs Real)	p=3.25e-01, d=-0.10	p=9.99e-01, d=0.00
s=1.0	Sim	Mapped
Radius (Δ from Real)	-0.201 \pm 0.000	-0.000 \pm 0.005
Length (Δ from Real)	-2.450 \pm 0.016	-0.010 \pm 0.215
Trajectory Diff (Pointwise)	0.185 \pm 0.005	0.011 \pm 0.001
Trajectory Diff (Frechet)	0.175 \pm 0.005	0.001 \pm 0.000
p/d (Trajectory vs Real)	p=2.53e-01, d=-0.12	p=9.98e-01, d=0.00
s=-0.3	Sim	Mapped
Radius (Δ from Real)	-1.708 \pm 0.001	0.003 \pm 0.192
Length (Δ from Real)	0.588 \pm 0.244	-0.040 \pm 0.152
Trajectory Diff (Pointwise)	1.581 \pm 0.048	0.011 \pm 0.001
Trajectory Diff (Frechet)	0.880 \pm 0.178	0.005 \pm 0.003
p/d (Trajectory vs Real)	p=2.55e-07, d=0.59	p=9.84e-01, d=0.00
s=-0.6	Sim	Mapped
Radius (Δ from Real)	-0.496 \pm 0.001	0.000 \pm 0.031
Length (Δ from Real)	-0.517 \pm 0.088	-0.009 \pm 0.183
Trajectory Diff (Pointwise)	0.552 \pm 0.177	0.012 \pm 0.001
Trajectory Diff (Frechet)	0.624 \pm 0.029	0.001 \pm 0.000
p/d (Trajectory vs Real)	p=2.71e-07, d=0.56	p=9.96e-01, d=-0.00
s=-1.0	Sim	Mapped
Radius (Δ from Real)	-0.238 \pm 0.000	0.000 \pm 0.004
Length (Δ from Real)	-1.929 \pm 0.045	-0.013 \pm 0.136
Trajectory Diff (Pointwise)	0.231 \pm 0.120	0.010 \pm 0.001
Trajectory Diff (Frechet)	0.250 \pm 0.148	0.001 \pm 0.000
p/d (Trajectory vs Real)	p=3.62e-02, d=0.37	p=9.97e-01, d=-0.00

Table 2: RQ1 Steering results

t=0.34	Sim	Mapped
Braking (Δ from Real)	0.068 ± 0.000	0.021 ± 0.000
Average Speed (Δ from Real)	-0.093 ± 0.000	-0.018 ± 0.000
Average Accel. (Δ from Real)	0.007920 ± 0.000000	-0.000331 ± 0.000000
p/d (Speed vs Real)	$p=1.49e-161, d=0.51$	$p=3.57e-03, d=0.05$
p/d (Accel vs Real)	$p=2.16e-35, d=-0.23$	$p=5.16e-01, d=-0.01$
t=0.365	Sim	Mapped
Braking (Δ from Real)	-0.050 ± 0.000	0.067 ± 0.000
Average Speed (Δ from Real)	-0.135 ± 0.000	-0.015 ± 0.000
Average Accel. (Δ from Real)	0.009088 ± 0.000000	0.000668 ± 0.000000
p/d (Speed vs Real)	$p=1.49e-161, d=0.51$	$p=3.57e-03, d=0.05$
p/d (Accel vs Real)	$p=2.16e-35, d=-0.23$	$p=5.16e-01, d=-0.01$
t=0.39	Sim	Mapped
Braking (Δ from Real)	-0.088 ± 0.025	0.064 ± 0.183
Average Speed (Δ from Real)	-0.209 ± 0.013	-0.021 ± 0.071
Average Accel. (Δ from Real)	0.007656 ± 0.001604	0.001570 ± 0.004817
p/d (Speed vs Real)	$p=1.49e-161, d=0.51$	$p=3.57e-03, d=0.05$
p/d (Accel vs Real)	$p=2.16e-35, d=-0.23$	$p=5.16e-01, d=-0.01$

Table 3: RQ1 Braking results

Straight	Sim	Mapped
Distance (Δ from Real)	0.007 ± 0.003	-0.000 ± 0.014
Trajectory Diff (Euclidean)	0.072 ± 0.045	0.008 ± 0.003
Trajectory Diff (Frechet)	0.045 ± 0.000	0.003 ± 0.000
Cumulative Error (Δ from Real)	-0.029 ± 0.001	0.002 ± 0.023
p/d (Trajectory vs Real)	$p=5.30e-01, d=-0.03$	$p=9.83e-01, d=0.00$
Close	Sim	Mapped
Distance (Δ from Real)	0.004 ± 0.006	0.000 ± 0.015
Trajectory Diff (Euclidean)	0.086 ± 0.031	0.008 ± 0.001
Trajectory Diff (Frechet)	0.031 ± 0.000	0.001 ± 0.000
Cumulative Error (Δ from Real)	0.027 ± 0.006	0.000 ± 0.013
p/d (Trajectory vs Real)	$p=4.80e-01, d=0.03$	$p=9.23e-01, d=-0.00$
Far	Sim	Mapped
Distance (Δ from Real)	0.101 ± 0.061	0.000 ± 0.068
Trajectory Diff (Euclidean)	0.115 ± 0.048	0.010 ± 0.004
Trajectory Diff (Frechet)	0.048 ± 0.000	0.004 ± 0.000
Cumulative Error (Δ from Real)	-0.004 ± 0.001	0.001 ± 0.003
p/d (Trajectory vs Real)	$p=7.68e-01, d=-0.01$	$p=9.93e-01, d=0.00$
Sharp	Sim	Mapped
Distance (Δ from Real)	0.032 ± 0.006	0.000 ± 0.026
Trajectory Diff (Euclidean)	0.385 ± 0.025	0.012 ± 0.005
Trajectory Diff (Frechet)	0.025 ± 0.000	0.005 ± 0.000
Cumulative Error (Δ from Real)	-0.075 ± 0.004	0.004 ± 0.030
p/d (Trajectory vs Real)	$p=1.78e-01, d=0.06$	$p=9.89e-01, d=0.00$
Curve	Sim	Mapped
Distance (Δ from Real)	0.131 ± 0.013	0.000 ± 0.046
Trajectory Diff (Euclidean)	0.137 ± 0.054	0.012 ± 0.004
Trajectory Diff (Frechet)	0.054 ± 0.000	0.004 ± 0.000
Cumulative Error (Δ from Real)	0.089 ± 0.003	0.004 ± 0.009
p/d (Trajectory vs Real)	$p=1.27e-01, d=-0.07$	$p=9.70e-01, d=0.00$
Stest	Sim	Mapped
Distance (Δ from Real)	0.073 ± 0.012	-0.000 ± 0.046
Trajectory Diff (Euclidean)	0.506 ± 0.044	0.014 ± 0.005
Trajectory Diff (Frechet)	0.044 ± 0.000	0.005 ± 0.000
Cumulative Error (Δ from Real)	0.054 ± 0.009	0.002 ± 0.066
p/d (Trajectory vs Real)	$p=1.93e-01, d=0.06$	$p=9.85e-01, d=-0.00$

Table 4: RQ1 Waypoint results

Module	Inputs		Outputs	
	Signal	Message type	Signal	Message type
Simulator	Real car pose	geometry_msgs/PoseStamped	Sim. car pose	geometry_msgs/PoseStamped
	Real obstacles poses	geometry_msgs/PoseArray	Sim. obstacles poses	geometry_msgs/PoseArray
	Control commands	[Throttle, steering, braking]	Camera output (full or masked)	sensor_msgs/Image (RGB)
	Throttle factor	std_msgs/Float32	Depth output (full or masked)	sensor_msgs/Image (Depth)
Car Actuation interface	Control commands	[Throttle, steering, braking]	Sim car speed	std_msgs/Float32
	Throttle factor	std_msgs/Float32		
Tracking interface			Real car pose Real obstacles poses Real car speed	geometry_msgs/PoseStamped geometry_msgs/PoseArray std_msgs/Float32
Sensors interface (Camera)			Camera output	sensor_msgs/Image (RGB)
Sensors interface (LiDAR)			Depth output	sensor_msgs/Image (Depth)
Sensor data mixing	Real image Sim image	sensor_msgs/Image (Depth or RGB) sensor_msgs/Image (Depth or RGB)	Mixed image	sensor_msgs/Image (Depth or RGB)
LiDAR pointcloud generation	Depth image	sensor_msgs/Image (Depth)	PointCloud	sensor_msgs/PointCloud2
	Camera matrix	[3X3 matrix]		
PID speed control	Vehicle speed Target speed	std_msgs/Float32 std_msgs/Float32	Throttle factor	std_msgs/Float32
Waypoint following	Car pose Waypoint	geometry_msgs/PoseStamped [x,y]	Control commands	[Throttle, steering, braking]

Table 5: Inputs and outputs of each framework module

Obstacles	Camera IoU	LiDAR Dist (m)
2 obstacles @ 40cm	0.75	0.08
2 obstacles @ 80cm	0.59	0.05
2 obstacles @ 120cm	0.42	0.04
2 obstacles @ 160cm	0.29	0.14
1 obstacle @ 40cm	0.81	0.03
1 obstacle @ 80cm	0.73	0.03
1 obstacle @ 120cm	0.90	0.06
1 obstacle @ 160cm	0.68	0.12
Lanes	Lane Dist (pixels)	
center lane	9.71 ± 6.62	
center lane counter	5.40 ± 3.24	
left lane	5.08 ± 5.81	
left lane counter	1.03 ± 1.32	
left margin	6.05 ± 7.09	
left margin counter	5.32 ± 4.02	
right lane	17.16 ± 27.18	
right lane counter	6.46 ± 3.71	
right margin	12.09 ± 7.78	
right margin counter	12.97 ± 4.18	

Table 6: Perception input quality

Metric (Lane Image)	SiL vs Real	MR vs Real
Correlation Coefficient (\uparrow)	0.26 ± 0.09	0.76 ± 0.07
Histogram Intersection (\uparrow)	$4.07e-03 \pm 3.06e-04$	0.69 ± 0.05
IFD (\uparrow)	1.70 ± 0.01	1.58 ± 0.03
KL Divergence (\downarrow)	2.12 ± 0.71	0.48 ± 0.19
LBP Histogram Similarity (\uparrow)	44949.23 ± 1582.73	73079.99 ± 725.74
MSE (\downarrow)	98.76 ± 0.68	21.94 ± 5.98
NMI (\downarrow)	$8.11e-01 \pm 8.41e-03$	$8.97e-01 \pm 9.18e-03$
Perceptual Distance (\downarrow)	19.57 ± 2.12	8.83 ± 1.58
PSNR (\uparrow)	28.19 ± 0.03	35.71 ± 1.26
SSIM (\uparrow)	0.23 ± 0.01	0.81 ± 0.05
Texture Similarity Dist. (\downarrow)	533.19 ± 40.00	181.80 ± 35.05
Frechet Distance (\downarrow)	$2.09e-03 \pm 2.30e-04$	$3.15e-04 \pm 7.12e-05$
Metric (Camera)	Sim vs Real	MR vs Real
Correlation Coefficient (\uparrow)	-0.38 ± 0.08	0.91 ± 0.10
Histogram Intersection (\uparrow)	$3.43e-03 \pm 1.50e-03$	0.75 ± 0.04
IFD (\downarrow)	$1.74e+00 \pm 8.01e-03$	1.69 ± 0.03
KL Divergence (\downarrow)	2.62 ± 0.09	0.01 ± 0.02
LBP Histogram Similarity (\uparrow)	42887.06 ± 4031.66	75565.28 ± 798.72
MSE (\downarrow)	106.20 ± 1.82	9.12 ± 6.64
NMI (\uparrow)	0.86 ± 0.02	$9.47e-01 \pm 4.58e-03$
Perceptual Distance (\downarrow)	17.39 ± 2.44	4.56 ± 3.34
PSNR (\uparrow)	27.87 ± 0.07	39.90 ± 3.09
SSIM (\uparrow)	0.21 ± 0.01	0.93 ± 0.05
Texture Similarity Dist. (\downarrow)	465.65 ± 33.65	9.12 ± 7.37
Frechet Distance (\downarrow)	$1.34e-03 \pm 1.47e-04$	$1.80e-04 \pm 1.46e-04$
Metric (LiDAR)	Sim vs Real	MR vs Real
Max Distance Error (m \downarrow)	1.068 ± 0.395	0.713 ± 0.435
Mean Distance Error (m \downarrow)	0.108 ± 0.037	0.039 ± 0.037
Std Dev of Error (m \downarrow)	0.179 ± 0.068	0.090 ± 0.082

Table 7: Image and LiDAR similarity metrics for SiL and MR against real data