RBE 550 – Motion Planning
Homework 4: Valet Parking
Section 6.5 – Path Animation
Mohamed Eljahmi

1. Overview
Each simulation run produced an animation of the Hybrid A* trajectory from start to finish.
The visualization tool sim/animate.py interpolated discrete planner poses and rendered oriented bounding boxes for the vehicle, overlaid on the obstacle map. Three animations were generated:

File	Description
robot_valet.gif	Differential-drive delivery robot
car_valet.gif	Ackermann-steered car
truck_valet.gif	truck with trailer articulation

Static frames (planned_path_*.png) accompany the animations for documentation.

2. How was the path generated?
All trajectories were generated by the **Hybrid A*** algorithm (Section 6.2).
The planner searches for the continuous configuration space SE(2) by expanding kinematically feasible rollouts rather than discrete grid steps.
Each edge is collision-checked using the **Separating Axis Theorem (SAT)**.
Motion primitives differ per vehicle:

- **Robot:** Differential-drive arcs enabling zero-radius turns.
- **Car:** Forward/reverse Ackermann arcs with wheelbase = 2.8 m.
- **Truck + Trailer:** Coupled tractor–trailer rollout, hitch distance = 5 m.

The resulting trajectories are continuous, dynamically plausible, and collision-free within the sampled resolution.

3. What is being delivered by the Delivery Robot?
The differential-drive robot represents a compact **delivery platform** transporting light parcels or food items between designated pickup zones and parking bays. Its animation demonstrates how small indoor or campus couriers can autonomously navigate narrow aisles or parking spaces using the same planning framework.

4. Most Egregious Parking Failure
During testing, the **truck + trailer** occasionally clipped an obstacle corner while reversing into the bay—an error typical of human drivers handling long articulated vehicles in tight spaces.
Comparable real-world examples include videos of tractor-trailers jack-knifing during parallel parking maneuvers, illustrating the same geometric coupling challenges that the planner faces.

4.1 Parking-Lane Feature and Docking Difficulties
An initial feature that divided the parking bay into fixed parking lanes was abandoned due to excessive complexity and limited time. The concept aimed to provide lane-level guidance for docking, but the implementation proved incompatible with the varied geometries of the differential-drive robot, Ackermann car, and truck–trailer system. Uniform lane dimensions failed to accommodate differences in vehicle width, overhang, and trailer articulation, which frequently resulted in collisions or infeasible approach angles near the bay entrance. Attempts to generalize the lane width or expand the bay dynamically increased code complexity without producing consistent gains in planner reliability. Consequently, the lane mechanism was removed, and final parking maneuvers were executed directly through the Hybrid A* planner using continuous free-space search. The remaining system plans directly to the goal pose defined within the parking bay. This approach simplifies geometry and avoids brittle failures associated with predefined lanes,

though it still requires fine-resolution rollouts and relaxed goal tolerances for accurate alignment.

5. **Major Challenges in Creating Accurate Paths**
   - **Kinematic Constraints:** Ackermann and articulated geometries restrict feasible curvature, limiting direct access to the goal pose.
   - **Collision Rejection Cost:** At high obstacle density, most motion primitives fail SAT checks, increasing search time.
   - **Discretization Trade-off:** Finer angular resolution improves accuracy but multiplies runtime.
   - **Visualization Synchronization:** Aligning simulation timestep with animation frame rate to achieve smooth playback without skipping intermediate states.
   - **Environment Complexity:** Narrow goal bays produce unavoidable local minima requiring reverse motions and multiple steering adjustments.

6. Challenges in Creating Accurate Paths

Accurate path generation required careful management of environment boundaries and obstacle placement. During world creation, the perimeter of the grid had to maintain adequate clearance between the boundary walls and the nearest obstacles. This margin needed to be sufficient for the active vehicle to maneuver safely; otherwise, vehicles—particularly the car and truck–trailer—occasionally moved beyond the map boundary before re-entering to complete their trajectories. Although changing the random seed sometimes alleviated the issue, it persisted under higher obstacle densities, where reduced clearance lowered the effectiveness of the collision checker and increased the likelihood of boundary incursions. The current implementation mitigates these behaviors to a degree, yet residual cases remain in dense environments. A comprehensive solution would involve redesigning the world generator to apply adaptive obstacles spacing that scales with vehicle dimensions and turning radius. Such modifications were not implemented due to time constraints but remain a priority for future versions of the planner and environment generator.

If you had more time, how would you improve your simulation?

If additional time were available, several areas of the simulation could be strengthened. The planning framework could benefit from a two-stage approach: a coarse Hybrid A* search to reach a pre-dock pose, followed by a finer, reverse-biased phase for precise parking. This would help the car and truck–trailer align more reliably inside the bay and reduce unnecessary steering corrections near the goal. The world generator would also need revision. Obstacle spacing and boundary margins should scale automatically with each vehicle's size and turning radius.

At present, the same grid spacing is used across all models, which can cause larger vehicles to brush against the perimeter or exceed map limits in dense scenes. Collision checking could be improved by testing the full swept volume of each motion segment rather than a few discrete samples. This would catch brief contacts that the current sampling step might miss and reduce false-positive clearances near corners. Finally, the visualization and control layers could be made more expressive. Adding a simple trajectory follower would allow continuous motion playback instead of discrete pose jumps, and rendering the trailer hitch geometry would make articulation more straightforward. Automating the collection of runtime metrics and seed sweeps would also make it easier to compare planner performance under different conditions. Overall, the next version should focus on smoother docking behavior, adaptive world generation, and more precise feedback from both the planner and the visualization tools.

a- Robot Animation

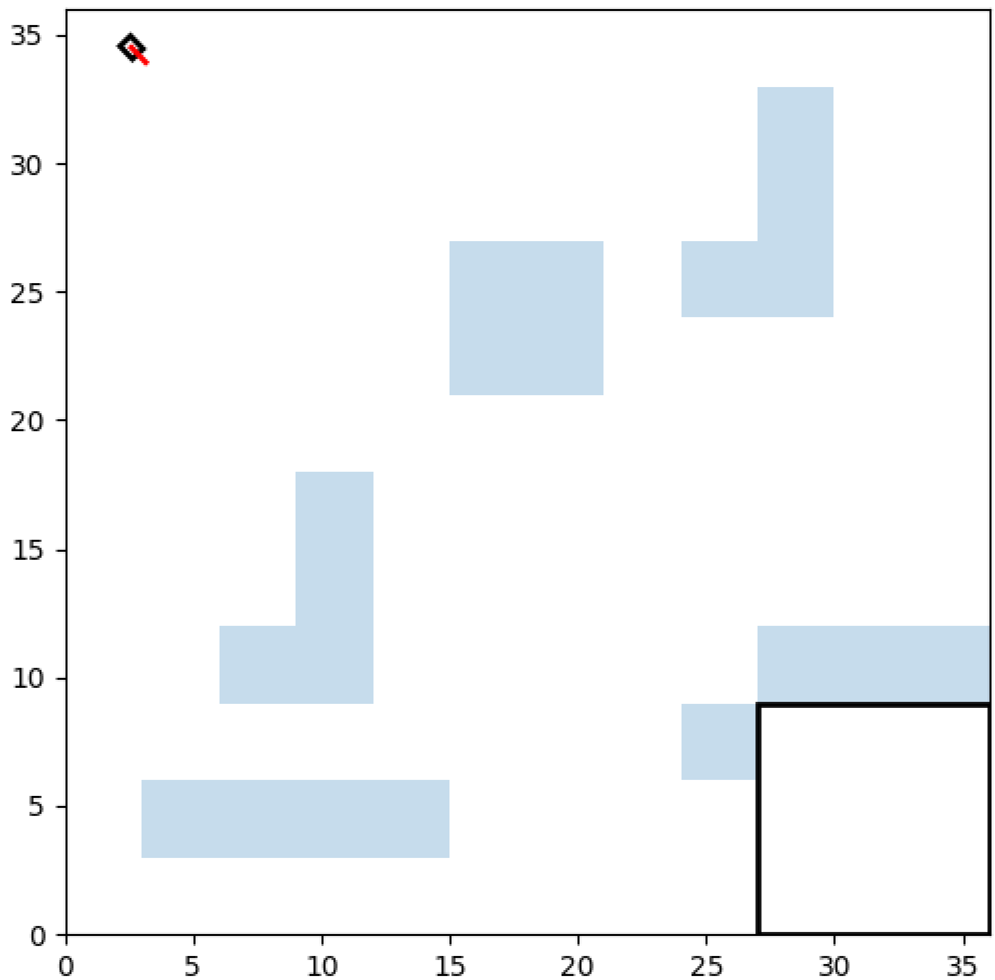
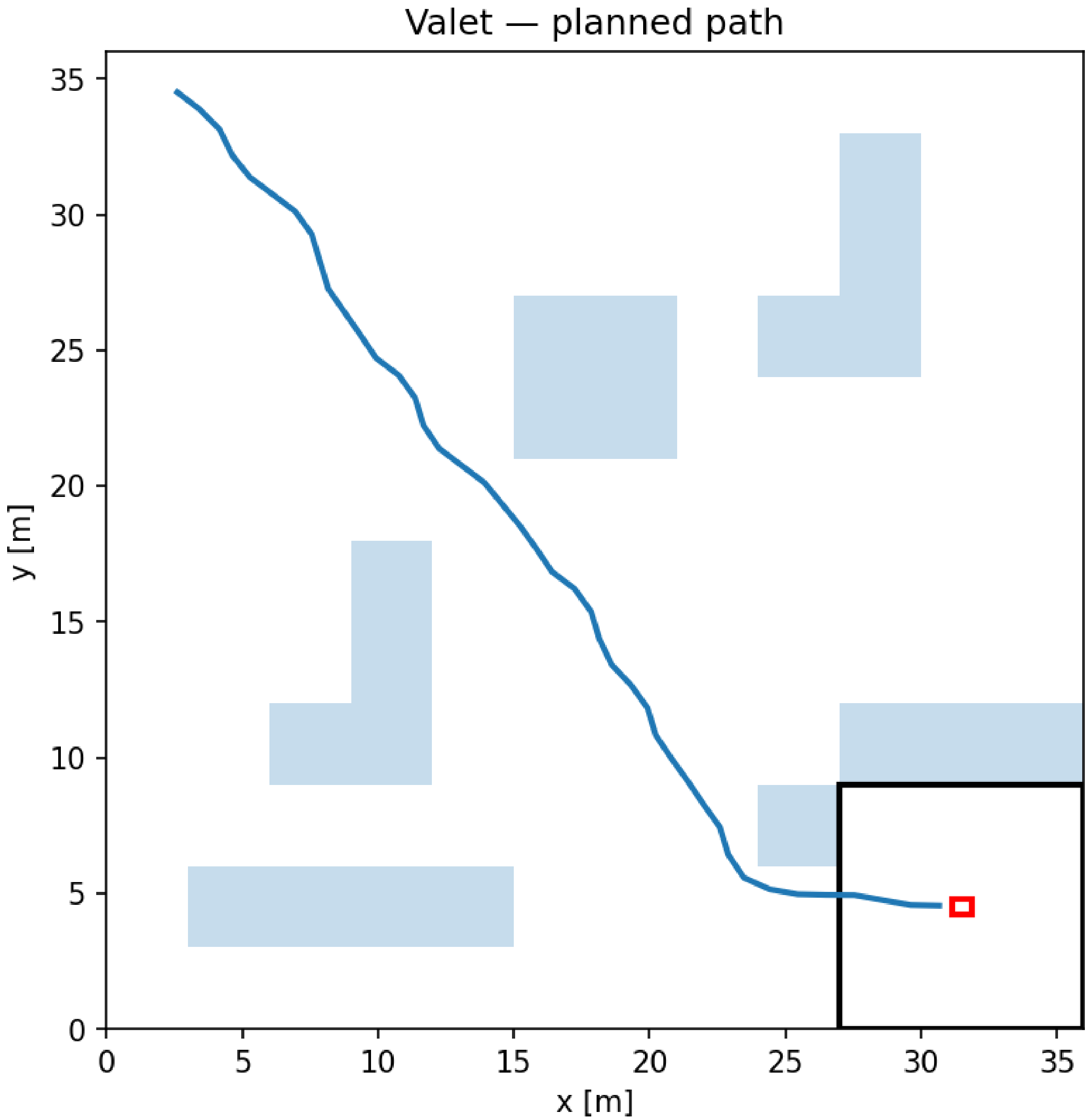


Fig. 1 — Robot Valet Animation: Smooth differential-drive trajectory; no collisions.

The following is the still frame of the Robot planned path:



b- Car Animation

Fig. 2 — Car Valet Animation: Feasible Ackermann path with reverse arcs into the goal bay.

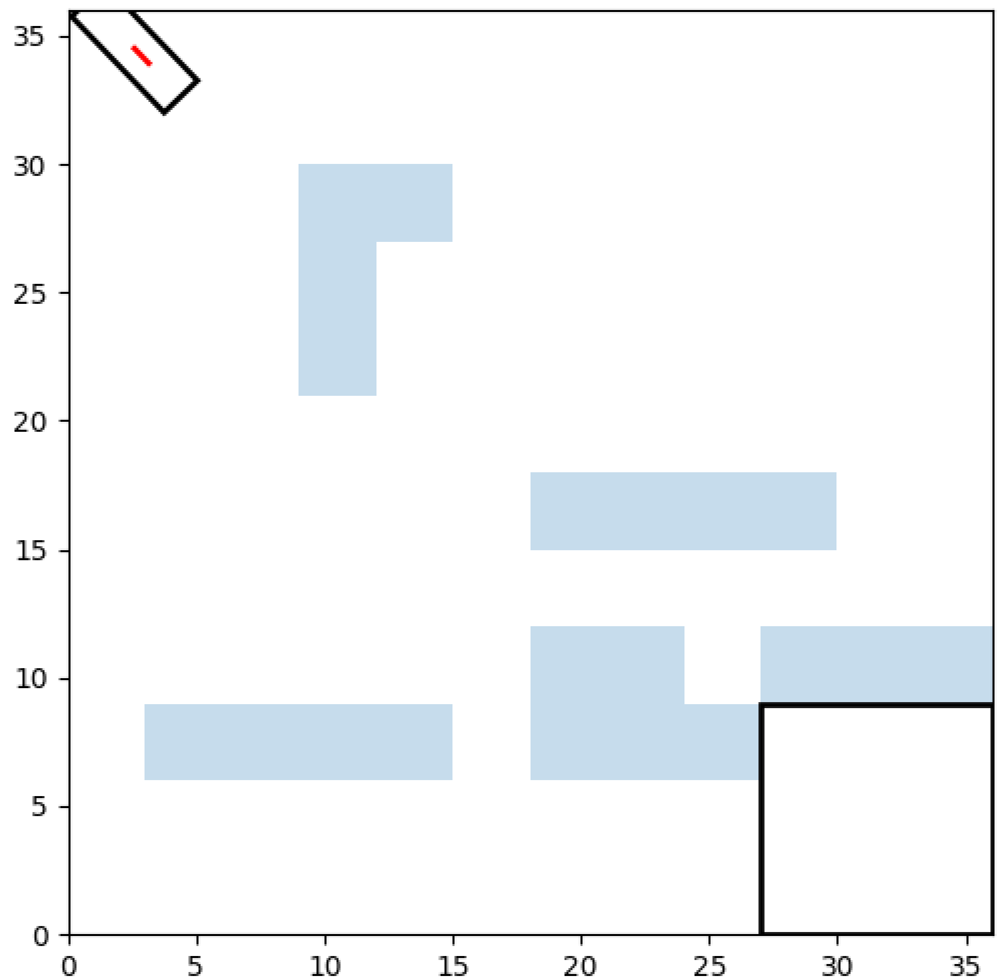
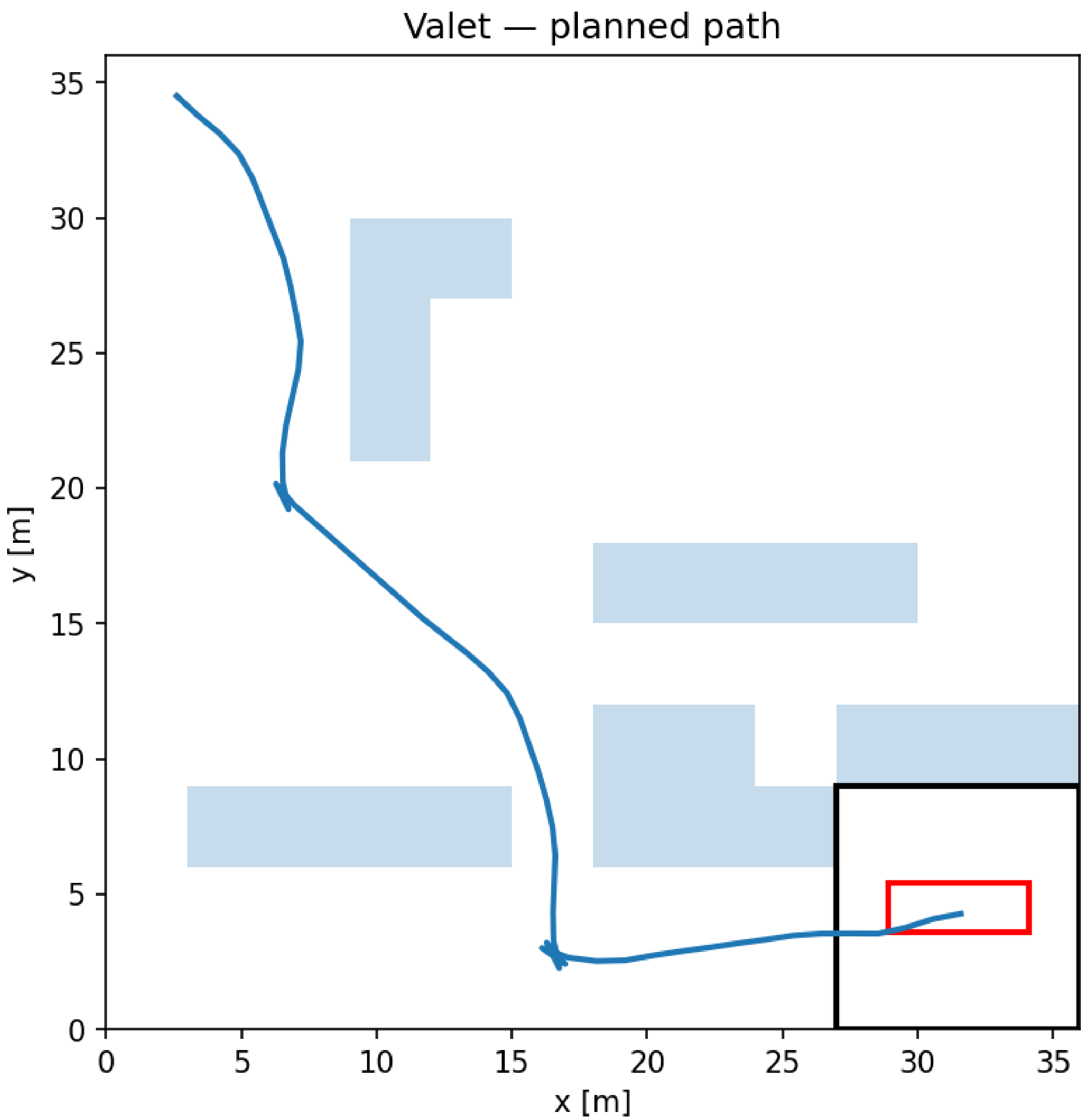


Fig. 2 — Car Valet Animation: Feasible Ackermann path with reverse arcs into the goal bay.

The following is the still frame of the Car planned path:



C- The Truck and Trailer Animation:

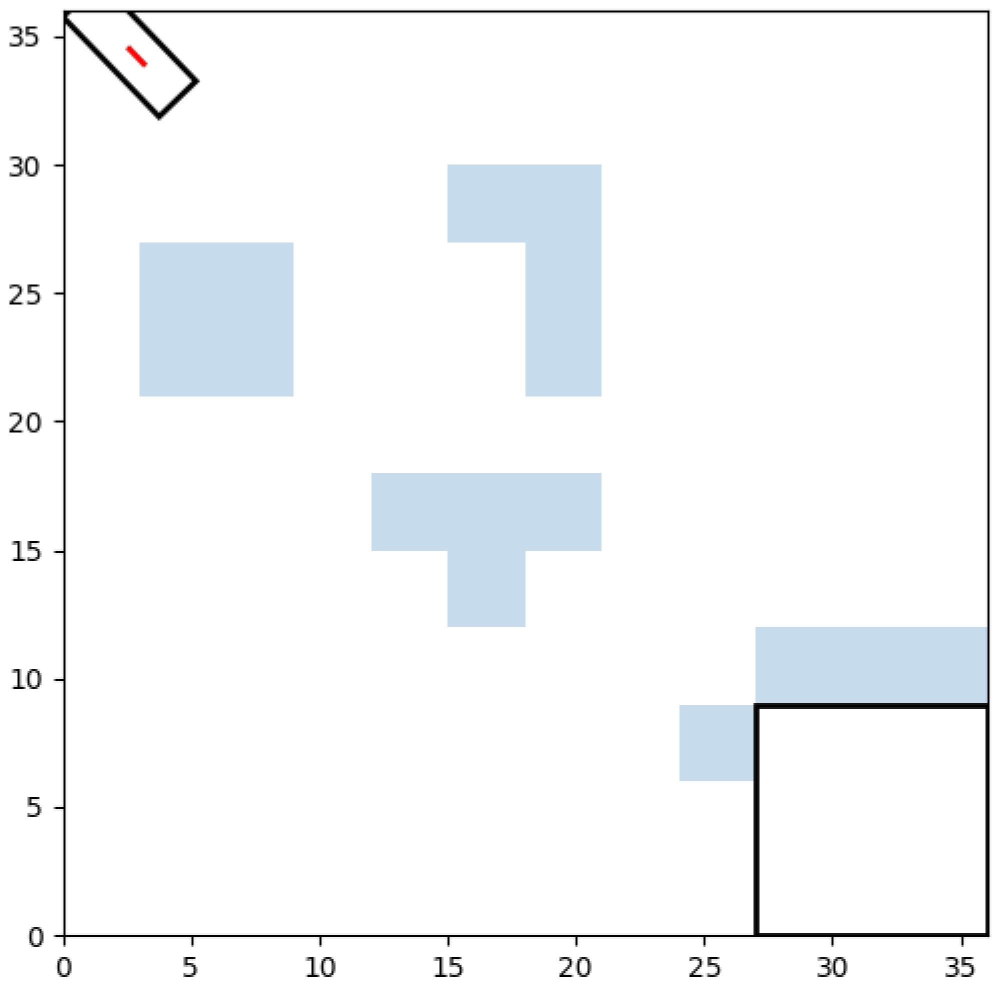
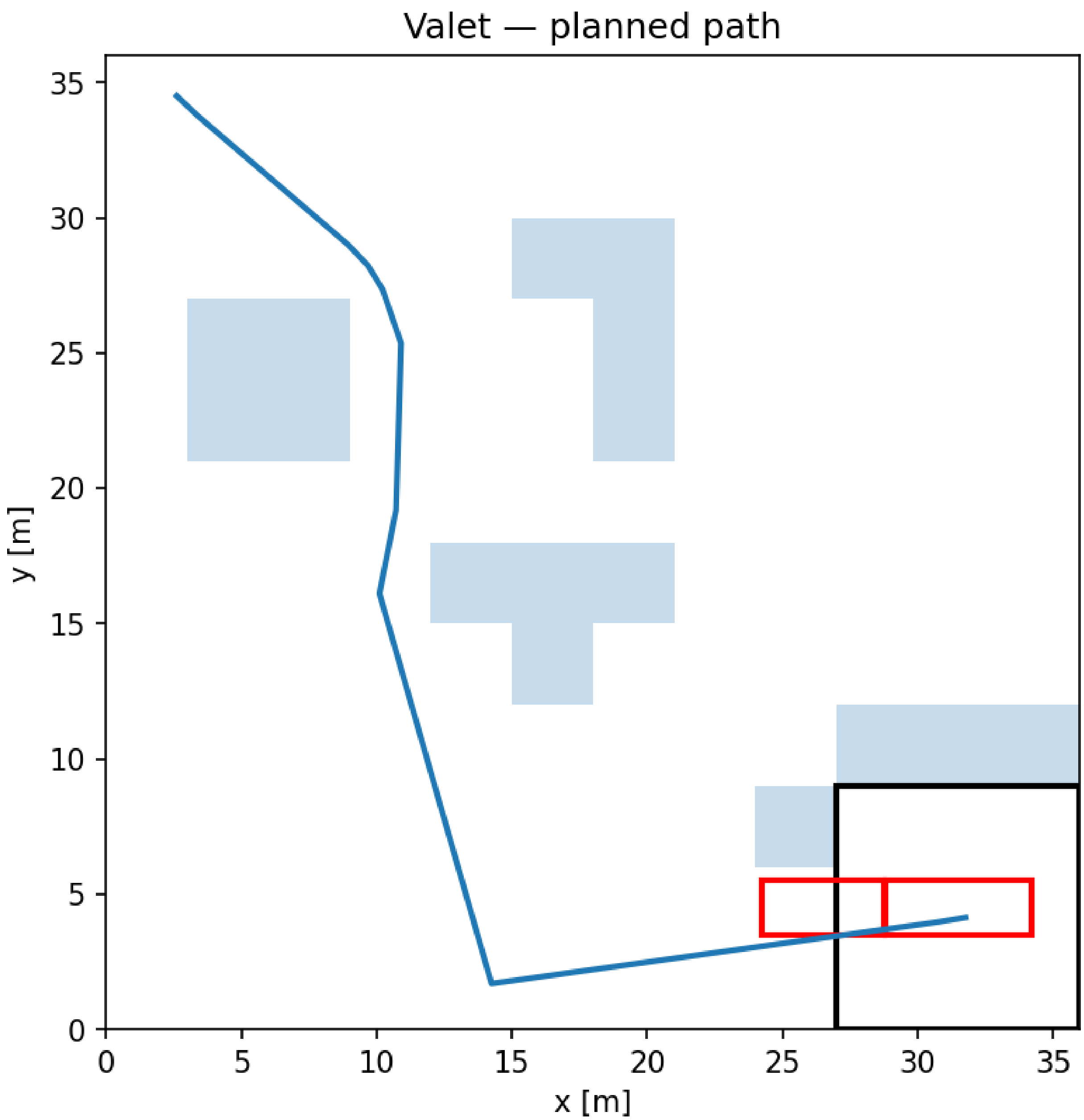


Fig. 3 — Truck + Trailer Animation: Longest trajectory with tight maneuvering and occasional near-collision at the bay entrance.

The following is the still frame of the Truck and Trailer planned path:



7. Observations

- **Robot:** Shortest and fastest motion; compact footprint simplifies avoidance.
- **Car:** Moderate runtime; wider turning radius causes larger clearance margins.
- **Truck + Trailer:** Slowest and most complex; articulation introduces additional curvature limits.
- **Density effect:** Increasing obstacle density drastically lengthens runtime, especially for articulated vehicles, because most sampled rollouts collide and are discarded.

8. Challenges

- Synchronizing animation timing with the planner's continuous poses.
- Maintaining kinematic realism without full dynamics simulation.
- Handling the trailer's swing motion and ensuring correct visualization of hitch articulation.