6.4 Planning and Obstacle Avoidance

Chapter Outline

Trajectory Planning Algorithms and Obstacle Avoidance

Chapter 6
External Control and Trajectory Planning

6.1 External Control Fundamentals
6.2 Control Principles and Modes
6.3 Trajectory Models and Generation
6.4 Planning and Obstacle Avoidance
6.5 Tracking and Aggressive Flight Control
6.6 Frontier Control Technologies
6.7 Platform and Communication Interfaces
6.8 Examples and Experimental Validation
6.9 Exercises and Classroom Discussion

4 Units in This Section

6.4.1 Global Path Planning
Textbook: Sec. 6.1.3.4

6.4.2 Local Planning and Real-Time Avoidance
Textbook: Secs. 6.1.3.4, 6.3.1

6.4.3 Multi-Vehicle Trajectory Planning
Textbook: Sec. 6.1.3.4

6.4.4 Planning Process and Postprocessing
Textbook: Sec. 6.1.3.5 (process Steps 2–3)

6.4.1 Global Path Planning: Planning Tasks

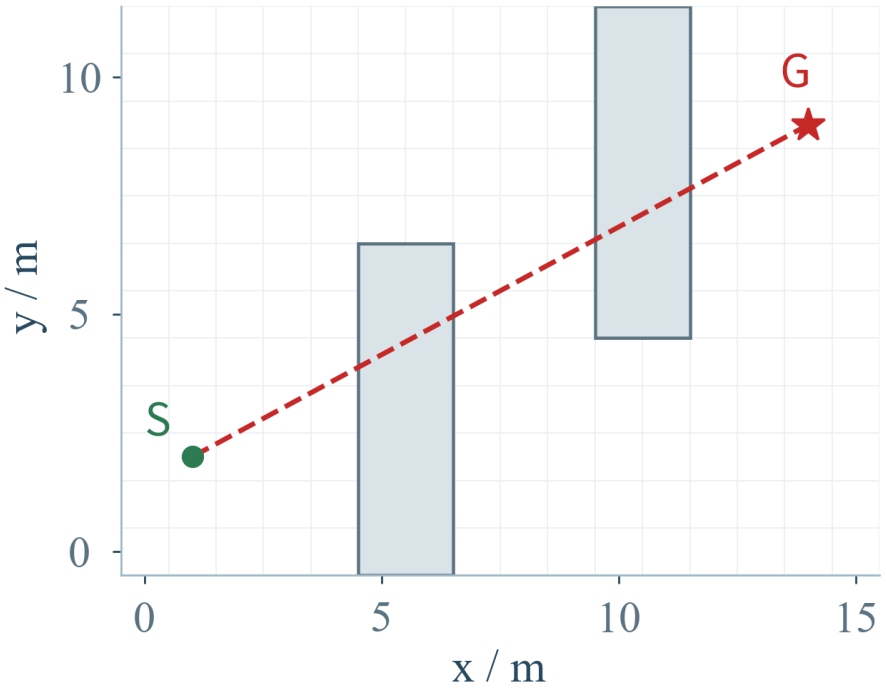
Global routes, local motion, and timed trajectories

Task	Inputs and Outputs	Checks in This Section
Global path planning	Map, start, goal → geometric route	Connected route; edges clear of obstacles
Local motion planning	State, nearby obstacles → short-horizon motion	Reachable action; can stop or turn safely
Trajectory generation	Route, timing, motion limits → reference	Continuous collision, speed, and acceleration checks

The closer an output is to execution, the more it must reflect timing and motion limits.

6.4.1 Global Path Planning: Common Map

Running example: navigate two staggered passages



Gray: inflated obstacles; point: vehicle center; red dashed line: direct connection.

Map: 16×12 cells, 1 m each. Planar translation; height ignored.

Start: (1,2) m; goal: (14,9) m. Rectangle boundaries count as occupied.

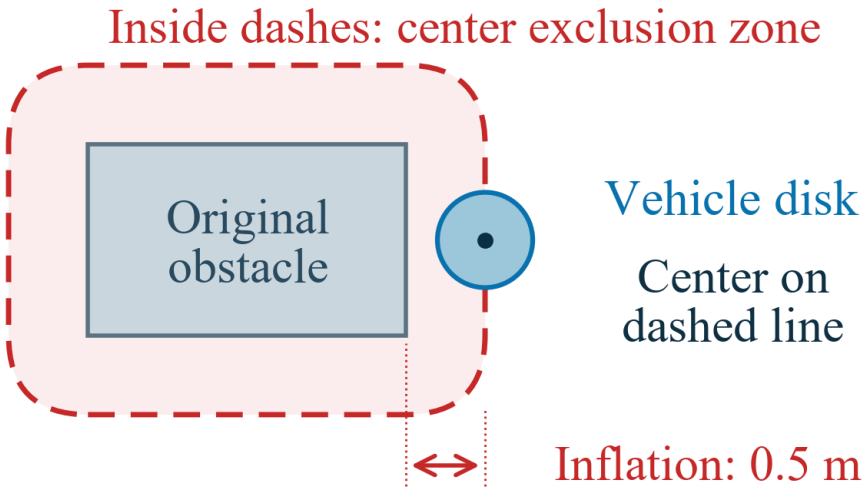
Search: first use four-connected grid moves of 1 m, then compare continuous-space sampling.

Map axes: east horizontally, north vertically.

$$(N, E) = (y, x)$$

6.4.1 Global Path Planning: Configuration Space

Inflating obstacles reduces the vehicle to a planning point



r_{body}: radius of the vehicle's bounding disk; m: safety margin; r_{inflate}: outward inflation distance, all in meters.

Configuration space describes all vehicle degrees of freedom. Here it contains only planar position, and the planning point is the center.

$$r_{\text{inflate}} = r_{\text{body}} + m = 0.3 + 0.2 = 0.5\ \text{m}$$

Check connections: collision-free endpoints do not imply a collision-free segment. Diagonal contact with a corner also counts as intersection.

6.4.1 Global Path Planning: Heuristic Search

A* adds cost so far to a lower bound on remaining cost

$$f(n) = g(n) + h(n)$$

$$h(n) = |x_n - 14| + |y_n - 9|$$

n: node; (x_n, y_n): its coordinates (m). Dijkstra uses $h = 0$.

Quantity	Meaning	Units Here
g	Lowest known cost from start to this node	m
h	Obstacle-free Manhattan distance: a lower bound	m
f	Priority for choosing the next node to expand	m

6.4.1 Global Path Planning: One Calculation

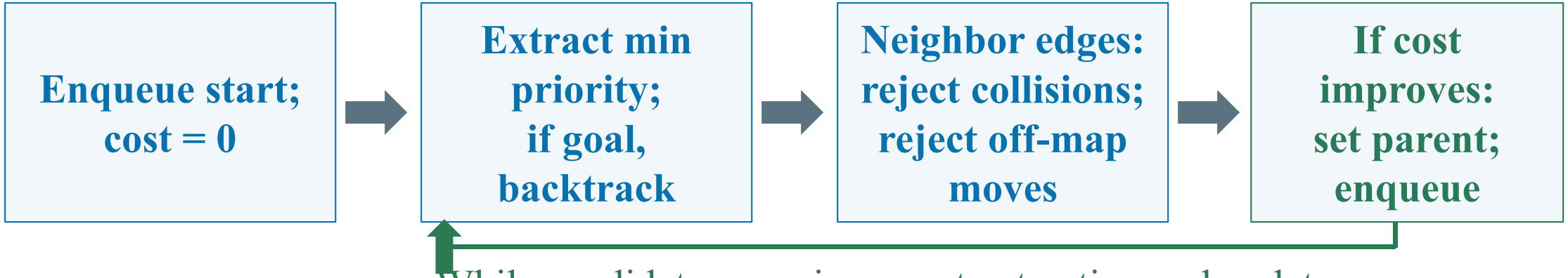
First expansion: prioritize moves toward the goal

Neighbor / m	Cost g / m	Lower Bound h / m	Priority f / m
East (2,2)	1	19	20
North (1,3)	1	19	20
West (0,2)	1	21	22
South (1,1)	1	21	22

Tie-breaking: compare h first, then insertion order east, north, west, south. Stop when the goal is dequeued.

6.4.1 Global Path Planning: Search Procedure

The search loop keeps costs and parent nodes



$$g_{\text{new}}(q) = g(n) + c(n, q)$$

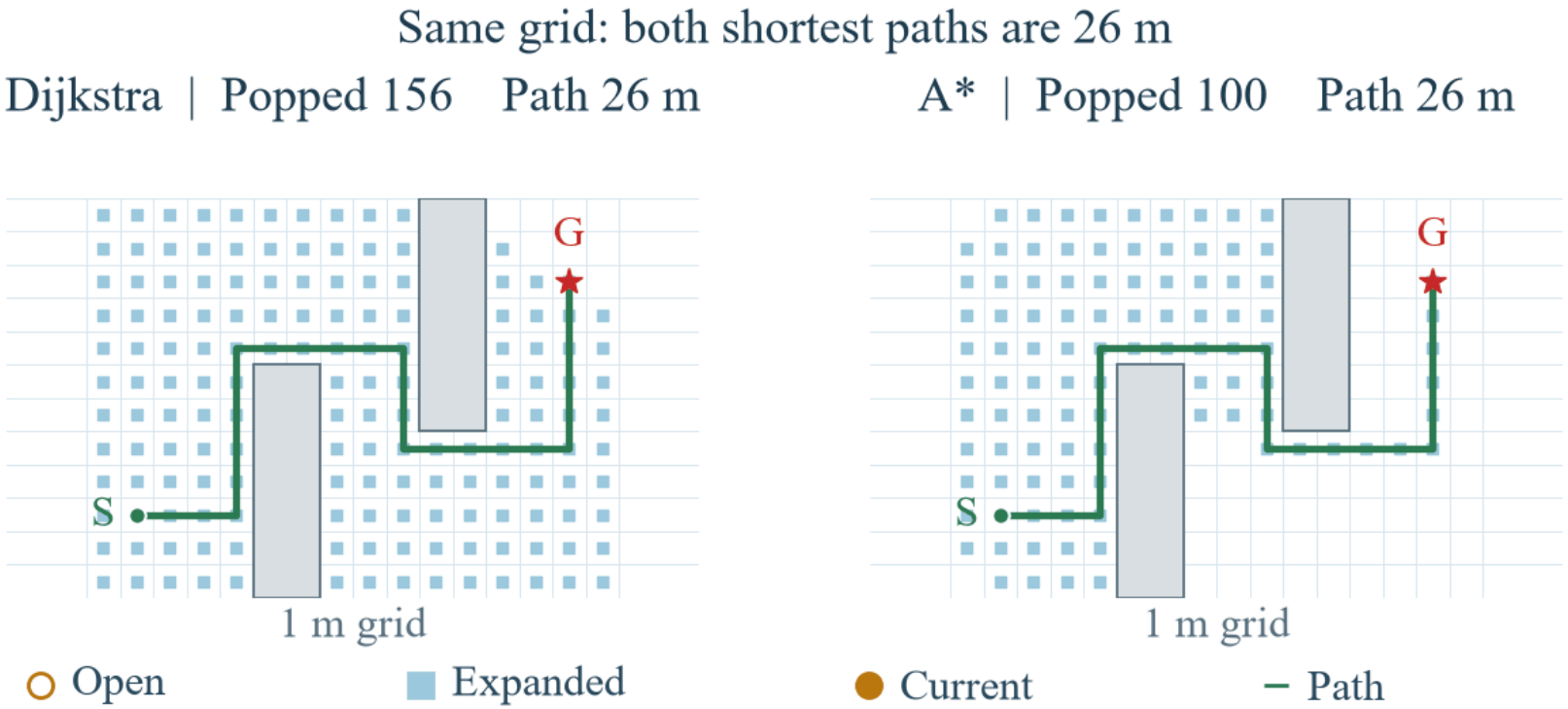
n: current node; q: neighbor; $c(n, q)$: edge cost (1 m here).

Output: parent links record the route. Backtrack from the goal, then reverse the sequence.

Failure: exhausting the candidate set means no route under the current graph and rules. A different resolution may change the result.

6.4.1 Global Path Planning: Complete Results

Both searches return 26 m on the same grid



Dijkstra: 156 expanded nodes (light blue squares)

A*: 100 expanded nodes; green shows the returned path

6.4.1 Global Path Planning: Optimality Conditions

Shortest-path claims have explicit conditions

Dijkstra: a finite graph with nonnegative edge costs; extract nodes by their lowest known cost.

A*: an admissible heuristic never overestimates remaining cost. Here Manhattan distance is also consistent.

Consistency: $h(n) \leq c(n, q) + h(q)$, where q is a neighbor of n.

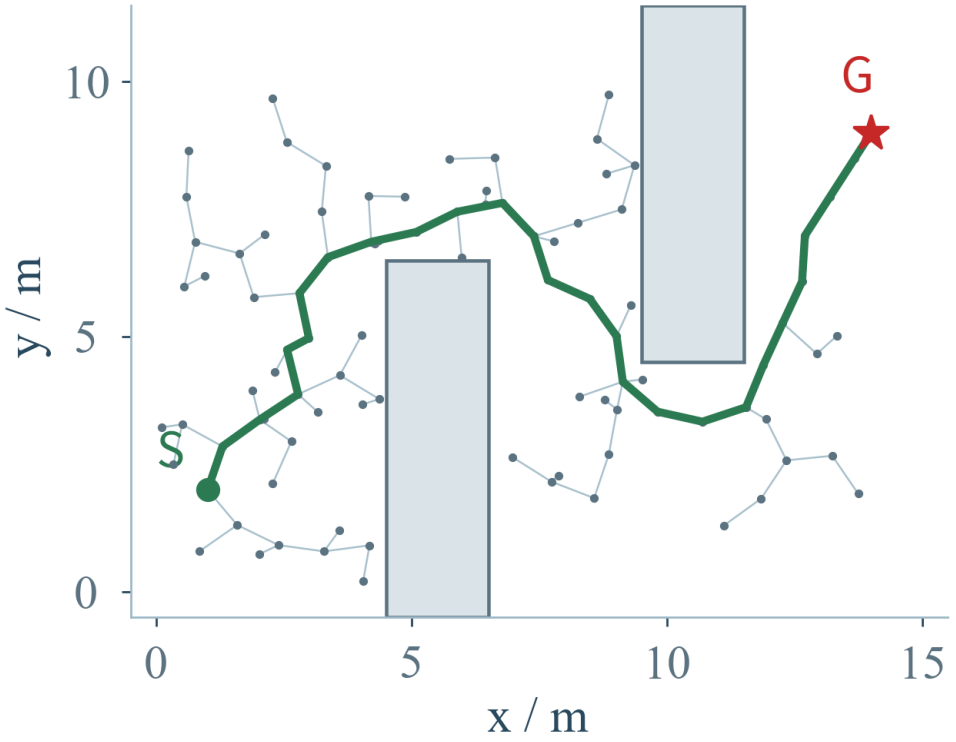
Eight-connected counterexample: a diagonal move from (0,0) to (1,1). Manhattan distance estimates 2 m, an overestimate.

Counterexample: $\sqrt{2}$ m $<$ 2 m. The estimate exceeds the actual cost; use Euclidean or diagonal distance instead.

"Shortest" must specify the map, connection rules, and cost.

6.4.1 Global Path Planning: Rapidly-Exploring Random Tree (RRT)

RRT extends an existing tree toward random samples



Random tree and one returned route

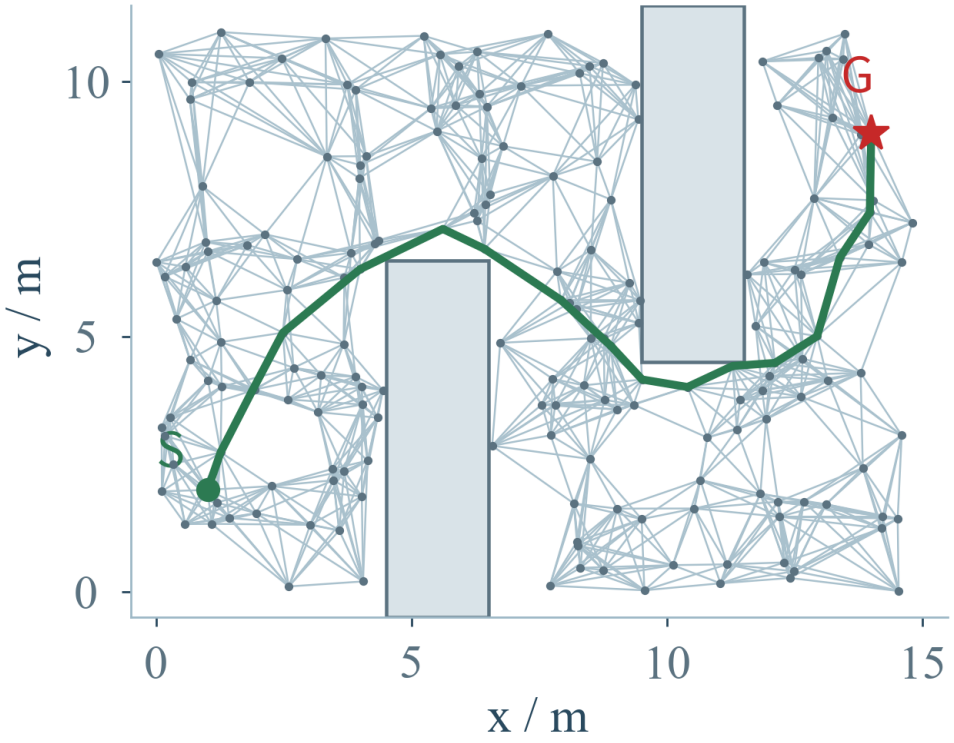
Loop: sample → nearest node → bounded-step extension → check the whole segment for collision.

Parameters/result: step 0.9 m, goal bias 0.1, seed 17; 94 tree nodes, route length 22.682 m.

Use: a single continuous-space query. Narrow passages are hard to sample; changing seed or implementation changes the tree and length.

6.4.1 Global Path Planning: Probabilistic Roadmap (PRM)

PRM builds a reusable roadmap, then connects each query



Reusable roadmap and one query

Build: 180 collision-free samples; try 10 nearest neighbors per point and check each entire connecting edge.

Query: connect start and goal, then search the existing graph. This route is 20.034 m.

Limits: narrow passages may be missed;
recheck connections after map changes.

6.4.1 Global Path Planning: Algorithm Tradeoffs

Distinguish feasibility from optimality when comparing results

Method	Run Length	Mechanism and Limits
Dijkstra / A*	26.000 m	Shortest four-connected path (horizontal/vertical only)
Basic RRT	22.682 m	Feasible random-tree route; no optimality guarantee
Basic PRM	20.034 m	Shortest in the fixed roadmap, not all continuous space

Lengths: sampling permits arbitrary edge directions; the grid only horizontal/vertical moves. These lengths do not directly rank methods.

RRT* selects parents and rewires; PRM* scales neighborhoods with sample count.

Asymptotic optimality: under the assumptions, cost approaches the optimum as samples grow without bound. Finite budgets give no optimality guarantee.

Bi-RRT grows two trees; Informed RRT* focuses on cost-improving regions; Lazy PRM defers collision checks.

Chapter Outline

Trajectory Planning Algorithms and Obstacle Avoidance

Chapter 6
External Control and Trajectory Planning

6.1 External Control Fundamentals
6.2 Control Principles and Modes
6.3 Trajectory Models and Generation
6.4 Planning and Obstacle Avoidance
6.5 Tracking and Aggressive Flight Control
6.6 Frontier Control Technologies
6.7 Platform and Communication Interfaces
6.8 Examples and Experimental Validation
6.9 Exercises and Classroom Discussion

4 Units in This Section

6.4.1 Global Path Planning
Textbook: Sec. 6.1.3.4

6.4.2 Local Planning and Real-Time Avoidance
Textbook: Secs. 6.1.3.4, 6.3.1

6.4.3 Multi-Vehicle Trajectory Planning
Textbook: Sec. 6.1.3.4

6.4.4 Planning Process and Postprocessing
Textbook: Sec. 6.1.3.5 (process Steps 2–3)

6.4.2 Local Planning and Real-Time Avoidance: Artificial Potential Field (APF)

Potential fields combine goal and obstacle guidance

Local scene: a new circular obstacle above the left obstacle, centered at (6,9) m with safety radius 1 m; local goal (8,9) m.

p: position; **q**: goal; d: boundary distance; **n**: outward unit vector.

$$\mathbf{F}_a = k_a(\mathbf{q} - \mathbf{p})$$

$$\mathbf{F}_r = k_r\left(\frac{1}{d} - \frac{1}{d_0}\right)\frac{\mathbf{n}}{d^2}, \quad 0 < d < d_0$$

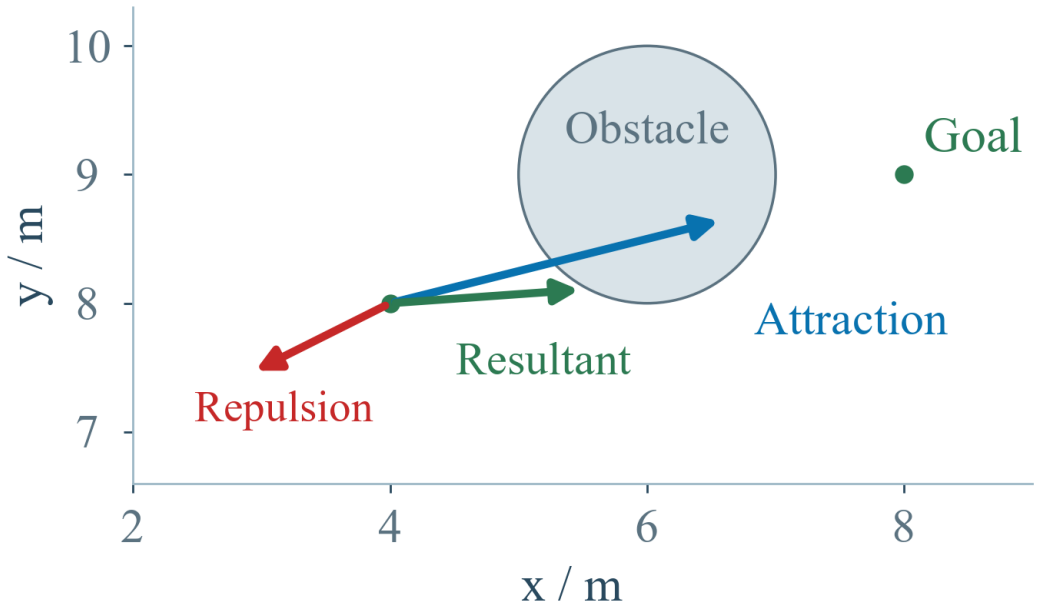
$$\mathbf{v} = \mathrm{sat}_{v_{\max}}(\mathbf{F}_a + \mathbf{F}_r), \quad \mathbf{p}^+ = \mathbf{p} + \Delta t\,\mathbf{v}$$

Use $k_a = 1\ \mathrm{s}^{-1}$, $k_r = 6\ \mathrm{m}^4/\mathrm{s}$, $d_0 = 3$ m; repulsion is zero outside the influence range.

Attraction and repulsion have units m/s. sat keeps direction, limits speed; time step in s.

6.4.2 Local Planning and Real-Time Avoidance: One Calculation

One potential-field step: add, limit speed, then update



One update from the same local state

Attraction: $\mathbf{F}_a = (4,1)$

Boundary distance: $d = \sqrt{5} - 1 \approx 1.236$ m

$$\mathbf{F}_r \approx (-1.671, -0.835)$$

$$\mathbf{F} \approx (2.329, 0.165)$$

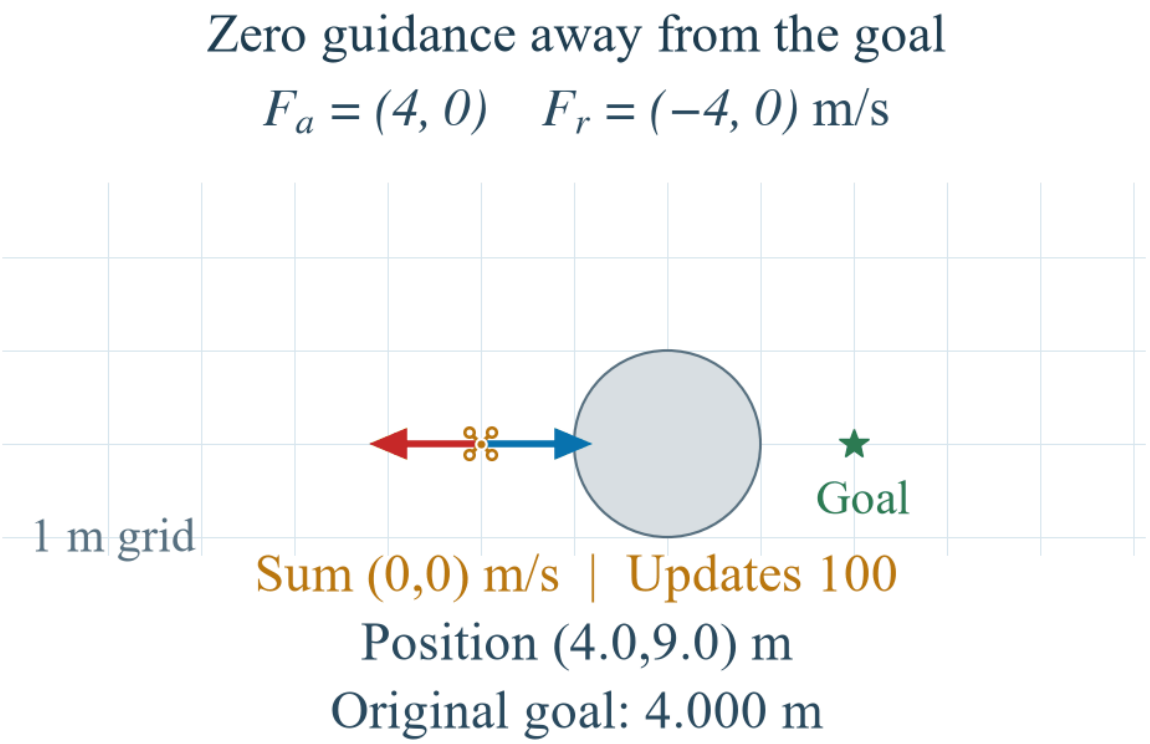
$$\mathbf{v} \approx (0.998, 0.071)\ \text{m/s}$$

Input position: (4,8) m;
speed limit: 1 m/s.

$$\Delta t = 0.2\ \text{s} \quad \Rightarrow \quad \mathbf{p}^{+} \approx (4.200, 8.014)\ \text{m}$$

6.4.2 Local Planning and Real-Time Avoidance: Equilibria and Stagnation

Zero guidance does not necessarily mean arrival



Attraction and repulsion cancel, causing stagnation

Reset the initial state to (4,9) m: position is unchanged after 1, 10, and 100 steps, still 4 m from the goal.

Cause: guidance based only on current position does not search for a route around the other side of the obstacle.

Response: switch the intermediate goal or trigger global replanning. Do not infer arrival directly from low speed.

6.4.2 Local Planning and Real-Time Avoidance: Avoidance Example

The installed avoidance model uses filtered coordinates

[installation directory]/RflySimAPIs/6.RflySimExtCtrl/1.BasicExps/e11_Avoidance_Segment/e5.1/sim1.0/

Guide: e5.1/Readme.pdf; model: e5_1_Avoidance_Segment.slx; data: SIM1.mat

$$\boldsymbol{\xi} = \mathbf{p} + k_0\mathbf{v}, \qquad k_0 = 1\ \mathrm{s}$$

Position/velocity feedback → filtered coordinates → avoidance → new position reference. ξ: current position extrapolated by k_0 s at current velocity. This distance starts avoidance earlier at higher speed; the teaching APF uses current position directly.

Model parameters: goal (25,6) m; obstacle center (12,0) m; geometric radius 2 m, designed safety radius 3 m.

Known issue: initialization calls missing legacy model e5_1_Avoidance_Segment_2017b. Until corrected, use SIM1.mat on "Reading the Data".

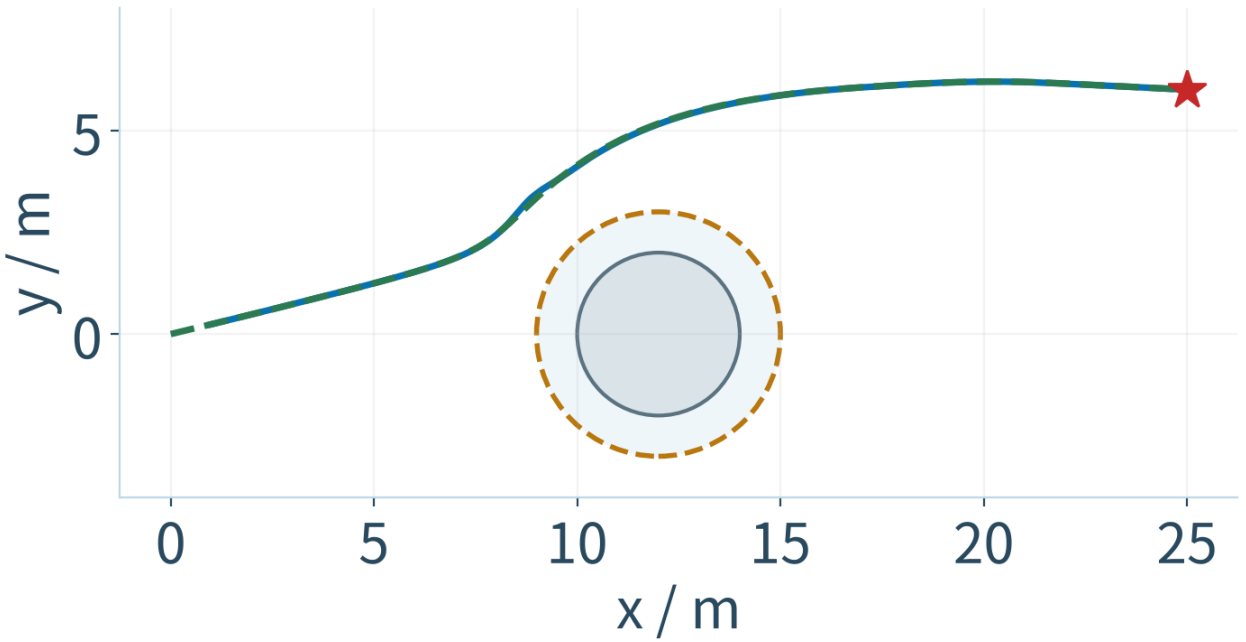
6.4.2 Local Planning and Real-Time Avoidance: Example Run Steps

e11 example: procedure and expected behavior

Initialize: in MATLAB, enter e5.1/sim1.0 and run startSimulation.m.
Set the goal: open e5_1_Avoidance_Segment.slx. In block X, set P_wp in turn to [25;6], [25;0], and [25;-6].
Simulate and plot: run once per goal, then run e5_plot.m to plot desired and actual position paths.
Expected: from (0,0) m, desired and actual paths nearly coincide, passing around the obstacle to the goal. With goal (25,0) m, start, obstacle center, and goal are collinear and deadlock occurs.
Common failures: if initialization cannot find e5_1_Avoidance_Segment_2017b, use "Reading the Data" until the platform is corrected. If avoidance fails, first check P_wp, then whether the three points are collinear (guide FAQ Q1).

6.4.2 Local Planning and Real-Time Avoidance: Reading the Data

Saved data: reference and feedback pass above the obstacle



SIM1.mat samples: solid blue reference; dashed green feedback.

Read: load SIM1.mat. Reference: x_desired, y_desired; feedback: x_feedback, y_feedback.

Check: minimum sampled feedback distance to center (12,0) m ≈4.493 m, exceeding the 3 m safety radius by 1.493 m.

Final value: (24.990,6.001) m, near the goal. Gray disk: radius 2 m; dashed circle: safety radius 3 m.

6.4.2 Local Planning and Real-Time Avoidance: Dynamic Window Approach (DWA)

The dynamic window searches only reachable velocities

Dynamic window approach (DWA): here use a planar ground-vehicle model with forward speed limited to 1 m/s.

v_0: current speed; $a_{\max}$: acceleration bound; $v_{\max}$: speed limit; Δt: update period.

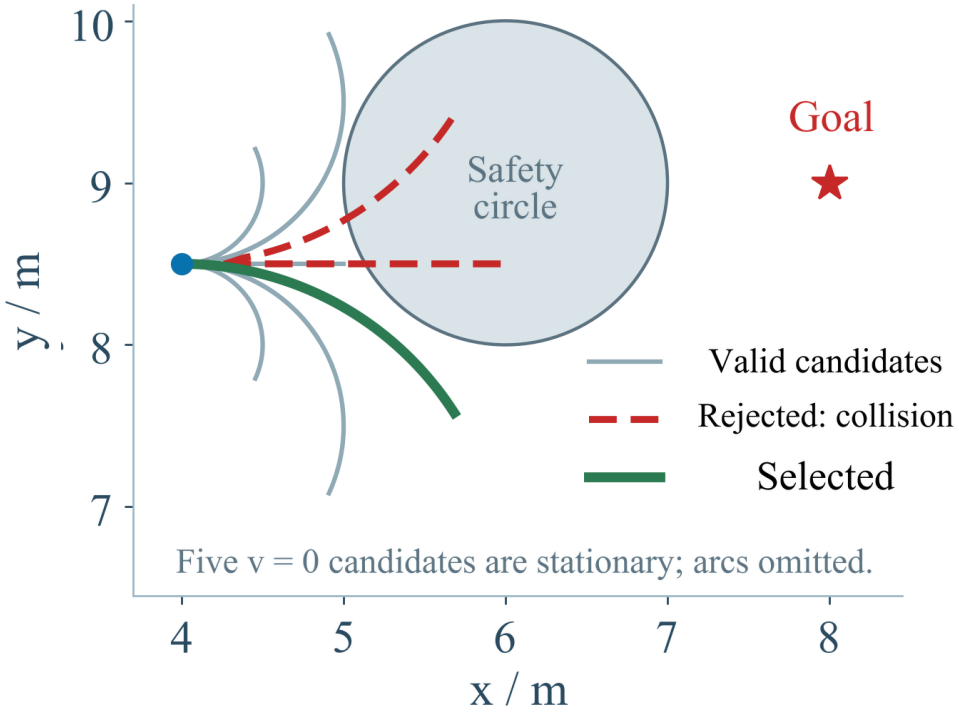
$$v \in [v_0 - a_{\max}\Delta t,\ v_0 + a_{\max}\Delta t] \cap [0, v_{\max}]$$

$$v_0 = 0.5\ \mathrm{m/s},\ a_{\max} = 1\ \mathrm{m/s^2},\ \Delta t = 0.5\,\mathrm{s} \Rightarrow v \in [0,1]\ \mathrm{m/s}$$

Speed candidates: forward 0, 0.5, 1 m/s; angular −1, −0.5, 0, 0.5, 1 rad/s.
Angular reachability: from 0 rad/s with angular acceleration ≤2 rad/s², ±1 rad/s is reachable after 0.5 s.
Braking check: deceleration 1 m/s²; check each 2 s prediction arc and its braking extension.

6.4.2 Local Planning and Real-Time Avoidance: Candidate Results

Dynamic window results: filter feasibility, then score



2 s candidate motions checked against the circular obstacle

Input: (4,8.5) m, heading along +x; goal (8,9) m, using the same safety circle.

Selected: speed 1 m/s, angular velocity −0.5 rad/s; minimum net clearance 0.202 m.

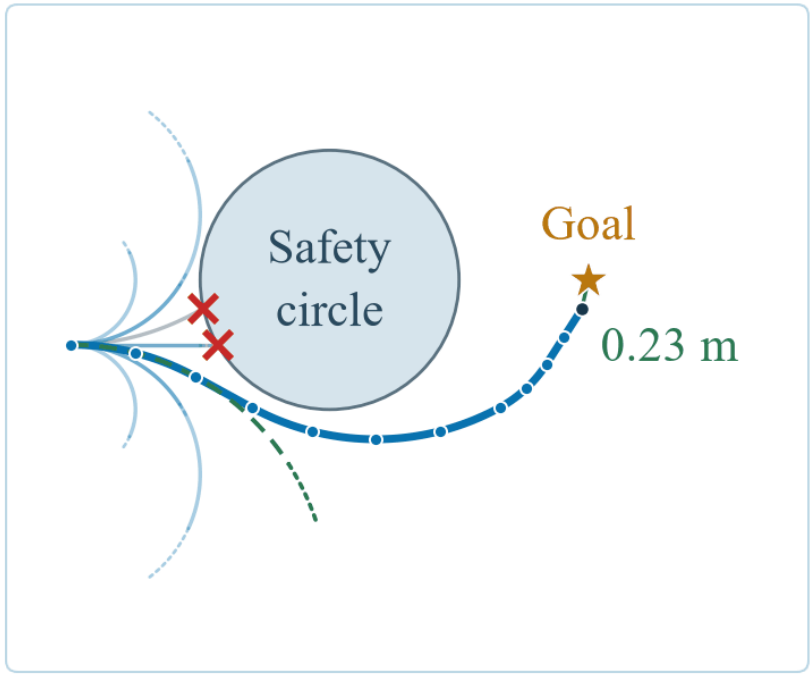
Filter: 13 of 15 candidates pass.
Score: minimize J among those that pass.

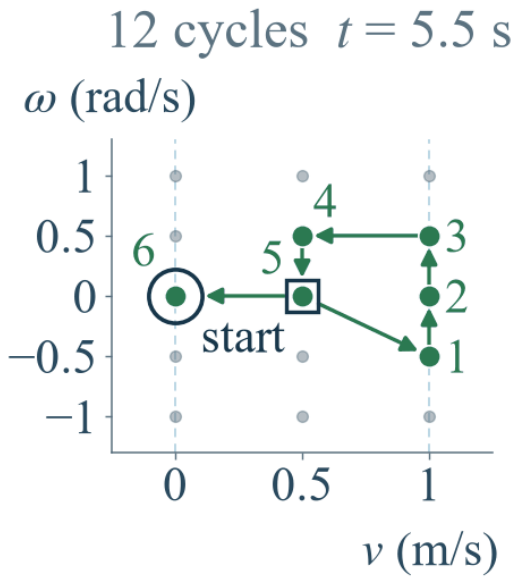
$$J = d_{\text{goal}} + 0.15|\omega| - 0.15v - 0.10\min(c, 2)$$

d_{goal}: predicted endpoint distance to goal; c: minimum net clearance;
v, ω: speed and angular velocity. Normalize each by a unit reference value.

6.4.2 Local Planning and Real-Time Avoidance: Receding-Horizon Execution

Replanning every 0.5 s changes the route from its first prediction





Min clearance 0.15 m

Stop: 0.23 m to goal

Receding-horizon DWA: simulation, not platform measurements.

Dynamic window: green points 1–6 are successive selected velocities. Dashed box: reachable in 0.5 s; it moves with current velocity.

Filter/score: red crosses reject colliding candidates; minimize J over feasible ones. First positive angular velocity: iteration 4.

Execution: 11 solid-blue route segments differ from the first green dashed prediction. The vehicle stops 0.23 m from the goal.

6.4.2 Local Planning and Real-Time Avoidance: Model Predictive Control (MPC)

MPC optimizes a future horizon and executes only the first step

Current state Local goal → **Predict N steps and optimize inputs** → **Execute one step, then get the new state**

$$p_{k+1} = p_k + \Delta t\, v_k + \frac{1}{2}\Delta t^2 a_k \qquad v_{k+1} = v_k + \Delta t\, a_k$$

Minimize $J = \sum_{k=0}^{N-1}\left[(p_{k+1} - p_{\text{ref}})^2 + \lambda a_k^2\right]$, subject to $|v_k| \le v_{\max}$, $|a_k| \le a_{\max}$ and obstacle-avoidance constraints.

k: prediction index; N: horizon (10 in Complete Results); λ: acceleration weight; p, v, a: position (m), velocity (m/s), acceleration (m/s²); constant acceleration within each step.

Receding-horizon optimization needs timely feedback and an acceptable solve time.

6.4.2 Local Planning and Real-Time Avoidance: One Optimization

One-axis prediction: the acceleration bound changes the optimum

Initial state: position 0 m, velocity 1 m/s. Predict 1 s toward position 2 m, with acceleration limited to ± 0.6 m/s².

$$p_1 = 1 + \frac{1}{2}a, \quad J = (2 - p_1)^2 + 0.25a^2 = 1 - a + 0.5a^2$$

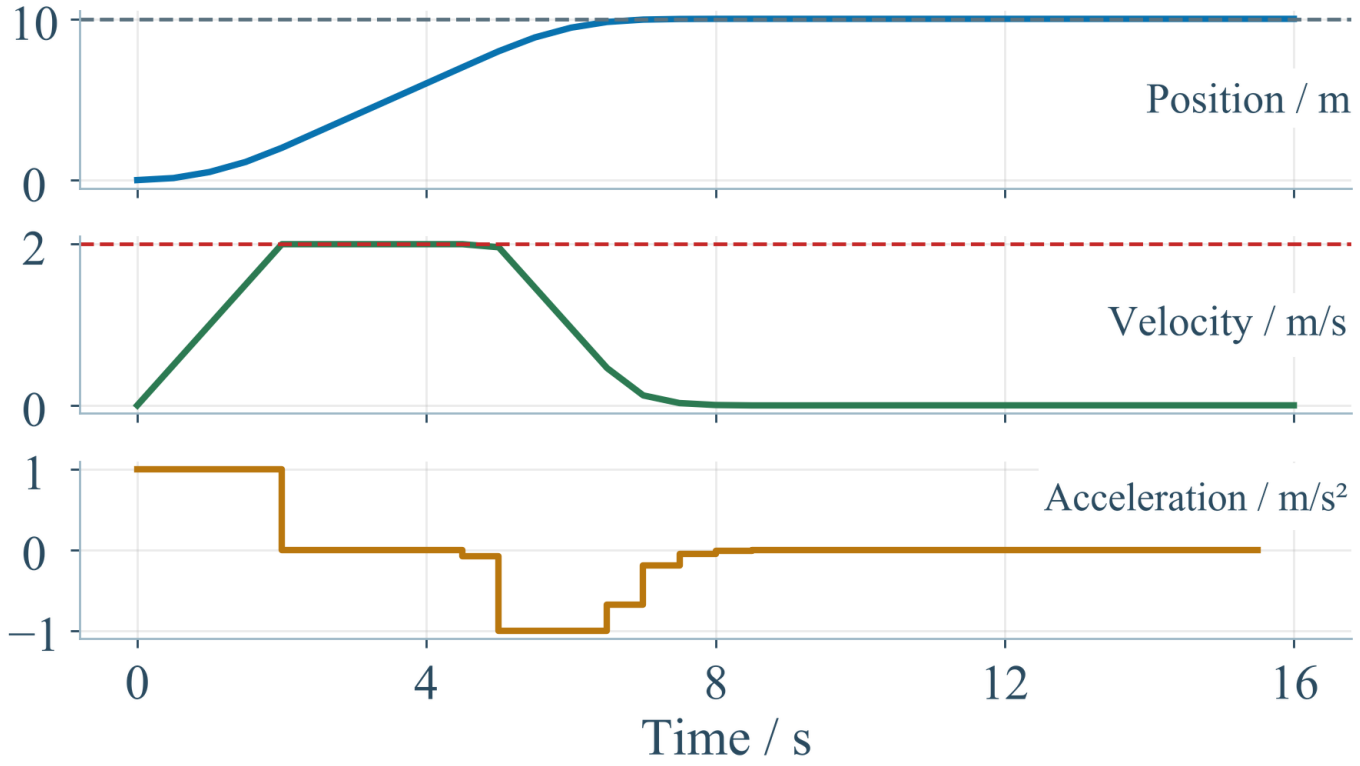
$$\frac{\mathrm{d}J}{\mathrm{d}a} = -1 + a = 0 \quad \Rightarrow \quad a = 1$$

$$a^* = 0.6\ \mathrm{m/s^2}, \quad p_1 = 1.3\ \mathrm{m}, \quad v_1 = 1.6\ \mathrm{m/s}$$

J trades terminal error against acceleration cost; normalize position by 1 m and acceleration by 1 m/s².

6.4.2 Local Planning and Real-Time Avoidance: Complete Results

Full receding-horizon result: accelerate, cruise, decelerate



Closed-loop calculation for ideal one-axis MPC

Initial state and goal: start at rest at 0 m and stop at 10 m.

Parameters: update every 0.5 s; horizon 10 steps; $|\text{velocity}| \leq 2$ m/s, $|\text{acceleration}| \leq 1$ m/s².

Cost: simplified MPC + zero-velocity target. Position/velocity weights: 1/0.5 (steps 1–9), 5/2 (step 10); input weight 0.05.

Scope: one axis, no obstacles. Avoidance usually needs nonconvex constraints or safe corridors.

6.4.2 Local Planning and Real-Time Avoidance: Comparing Local Methods

Three local methods solve the problem differently

Method	Main Output	Common Limits and Responses
APF: potential field	Local direction or motion guidance	Local minima and stagnation; use intermediate goals or escape logic
DWA: dynamic window	Short-horizon reachable velocity candidate	Short horizon, finite candidates; check braking and keep replanning
MPC: predictive control	Input sequence satisfying modeled constraints	Sensitive to model and solve; on timeout, use a validated policy

Global routes provide a way around obstacles; local methods keep responding to current state.

Chapter Outline

Trajectory Planning Algorithms and Obstacle Avoidance

Chapter 6
External Control and Trajectory Planning

6.1 External Control Fundamentals
6.2 Control Principles and Modes
6.3 Trajectory Models and Generation
6.4 Planning and Obstacle Avoidance
6.5 Tracking and Aggressive Flight Control
6.6 Frontier Control Technologies
6.7 Platform and Communication Interfaces
6.8 Examples and Experimental Validation
6.9 Exercises and Classroom Discussion

4 Units in This Section

6.4.1 Global Path Planning
Textbook: Sec. 6.1.3.4

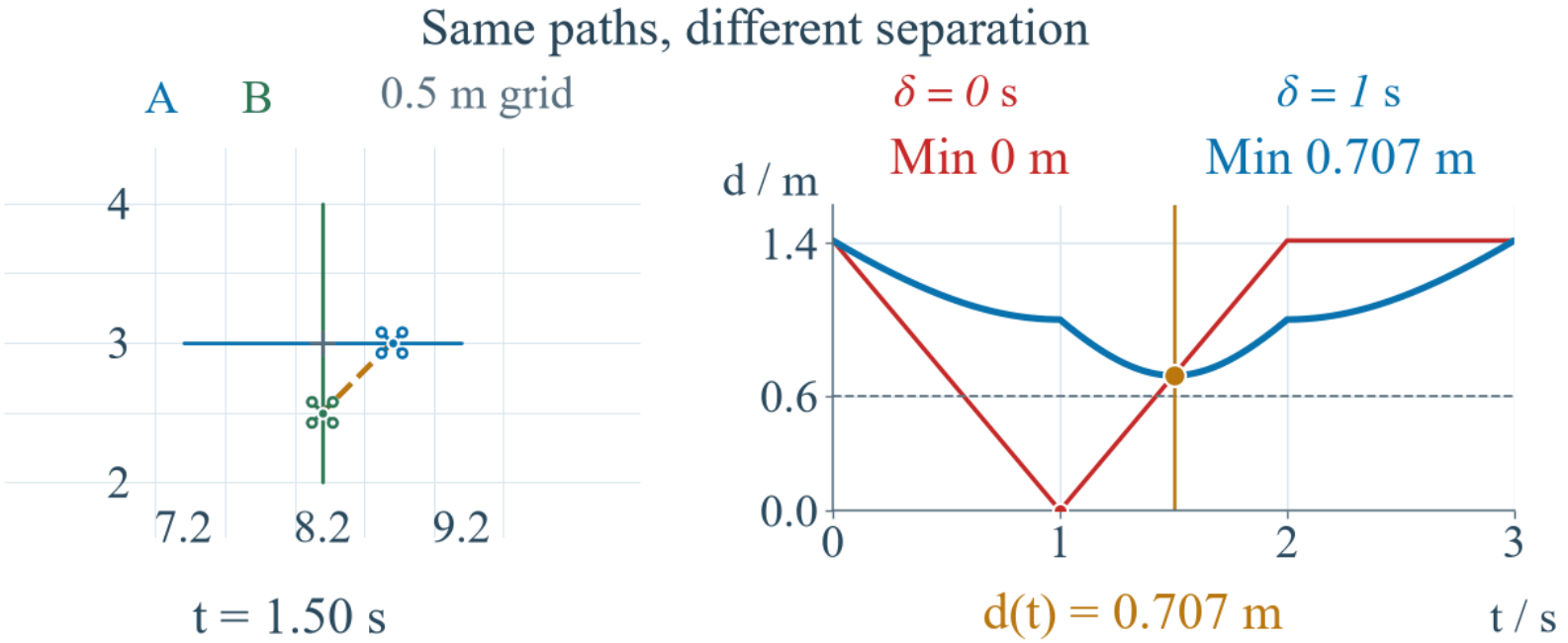
6.4.2 Local Planning and Real-Time Avoidance
Textbook: Secs. 6.1.3.4, 6.3.1

6.4.3 Multi-Vehicle Trajectory Planning
Textbook: Sec. 6.1.3.4

6.4.4 Planning Process and Postprocessing
Textbook: Sec. 6.1.3.5 (process Steps 2–3)

6.4.3 Multi-Vehicle Trajectory Planning: Space–Time Conflicts

Two individually feasible routes can conflict at an intersection



A: (7.2,3) → (9.2,3) m; B: (8.2,2) → (8.2,4) m; both at 1 m/s.

Gray dashed line: base separation 0.6 m. Simultaneous starts give zero separation; delaying B by 1 s gives a minimum of 0.707 m.

6.4.3 Multi-Vehicle Trajectory Planning: Continuous Distance

Staggered starts: closest approach falls between the two crossings

Keep constant speed 1 m/s; t, δ are in s and d in m. δ is B's departure delay.

$$d^2(t) = (t-1)^2 + (t-1-\delta)^2, \quad \delta \le t \le 2$$

$$t^* = 1 + \frac{\delta}{2}, \qquad d_{\min} = \frac{\delta}{\sqrt{2}}$$

B's Delay	Minimum Separation	Total Task Time
0 s	0 m: collision	2 s
1 s	0.707 m: above base threshold 0.6 m	3 s
1.3 s	0.919 m: meets 0.9 m with margin	3.3 s

Table: waiting/stops checked. Formula minimum applies only while both vehicles move.

6.4.3 Multi-Vehicle Trajectory Planning: Robustness and Coordination

Clock error, localization, and delay consume separation margin

$$d_{\text{req}} = d_{\text{base}} + e_A + e_B + v_{\text{rel,max}}\tau$$

$$d_{\text{req}} = 0.6 + 0.05 + 0.05 + 2 \times 0.1 = 0.9\ \text{m}$$

d_{req}: required center separation; d_{base}: base separation 0.6 m.
e_A, e_B: each vehicle's localization-error bound, 0.05 m; $v_{\text{rel,max}}$: relative-speed bound, 2 m/s.
τ: bound on shared timebase offset, here 0.1 s.

Reschedule: B delayed by 1.3 s: 0.919 m separation meets 0.9 m, with only 0.019 m margin.

Share: trajectories and timebase, priorities, validity periods, replanning triggers, and loss-of-link actions.

Check: waiting regions, beyond the prediction horizon, and stopping periods; recheck after new obstacles appear.

6.4.3 Multi-Vehicle Trajectory Planning: Coordination Methods

Multi-vehicle methods must find routes and handle scale

Method	Coordination Mechanism	Tradeoffs
Sampling	PRM/RRT in joint space; or coordinated individual paths	Growing dimension; timing coordination still needed
Optimization	MPC/optimal control: joint trajectories and separation	Explicit constraints; nonconvex problems are sensitive to initial guesses and computing resources
Swarm heuristics	Particle swarm, ant colony, genetic search	Useful for allocation/search; finite iterations give no optimality guarantee

10 vehicles with 3D positions: 30 position dimensions and $10(10-1)/2 = 45$ pairwise distance constraints.

Chapter Outline

Trajectory Planning Algorithms and Obstacle Avoidance

Chapter 6
External Control and Trajectory Planning

6.1 External Control Fundamentals
6.2 Control Principles and Modes
6.3 Trajectory Models and Generation
6.4 Planning and Obstacle Avoidance
6.5 Tracking and Aggressive Flight Control
6.6 Frontier Control Technologies
6.7 Platform and Communication Interfaces
6.8 Examples and Experimental Validation
6.9 Exercises and Classroom Discussion

4 Units in This Section

6.4.1 Global Path Planning
Textbook: Sec. 6.1.3.4

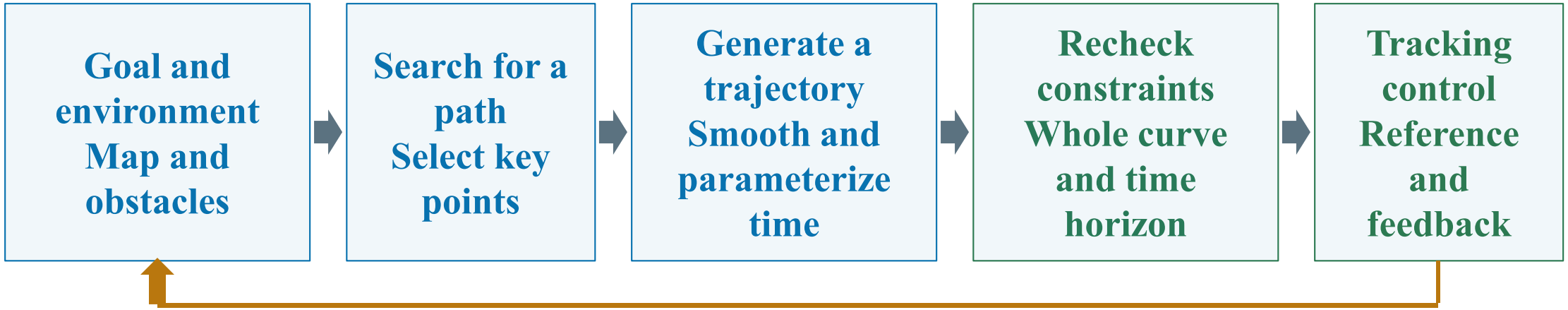
6.4.2 Local Planning and Real-Time Avoidance
Textbook: Secs. 6.1.3.4, 6.3.1

6.4.3 Multi-Vehicle Trajectory Planning
Textbook: Sec. 6.1.3.4

6.4.4 Planning Process and Postprocessing
Textbook: Sec. 6.1.3.5 (process Steps 2–3)

6.4.4 Planning Process and Postprocessing: Planning Process

The complete process links route checks to execution feedback



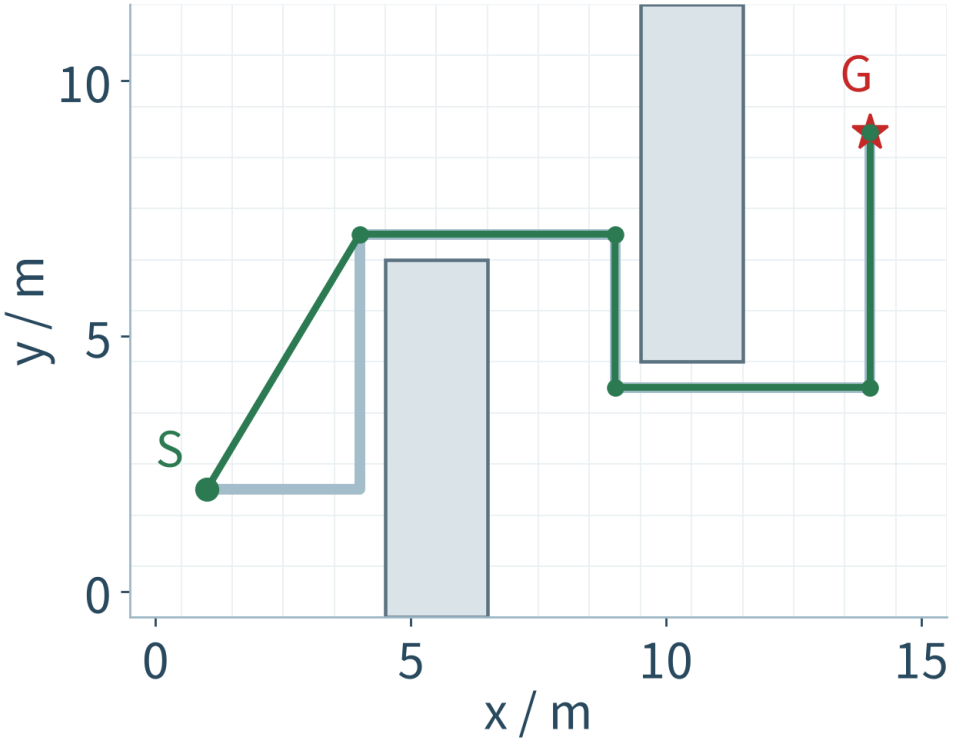
Feedback updates state; environmental changes can trigger a new search

If checks fail, return to key-point selection, smoothing, or timing; revise and check again.

If checks pass, deliver position, velocity, and acceleration references. Section 6.5 develops trajectory tracking and feedforward compensation.

6.4.4 Planning Process and Postprocessing: Key-Point Selection

Simplify a path only when a direct connection is clear



Original path and collision-checked shortcuts

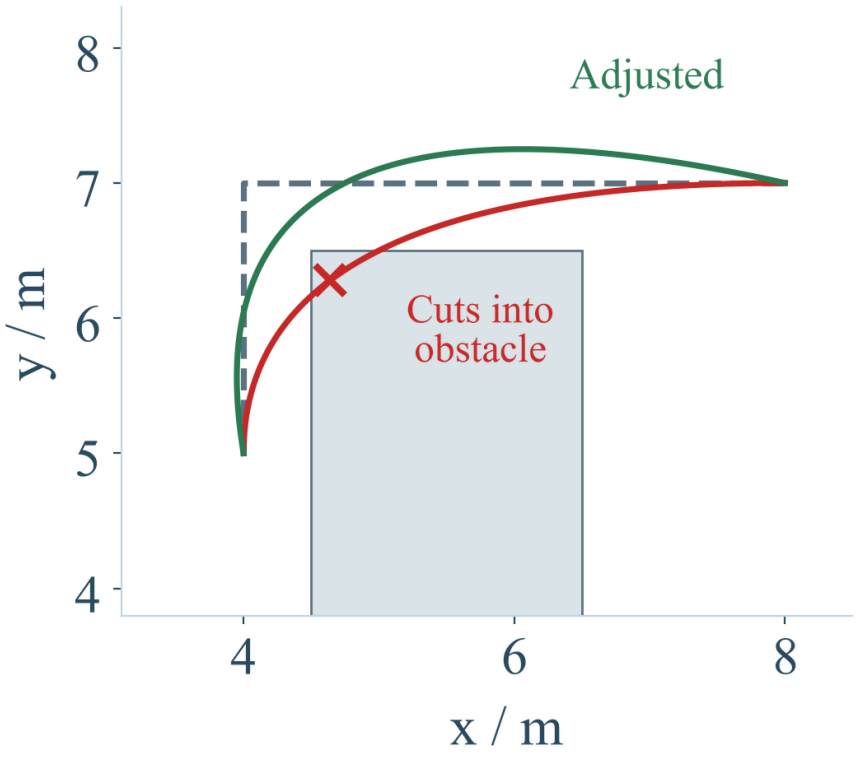
Rule: from the current point, find the farthest visible successor on the path. The whole connecting segment must be traversable.

This result: 27 points → 6; 26.000 m → 23.831 m.

Still needed: retain clearance, smooth corners, add timing, and recheck motion constraints.

6.4.4 Planning Process and Postprocessing: A Smoothing Counterexample

A smooth curve can cut into an obstacle avoided by the polyline



Quadratic Bézier corner cutting and correction

Polyline: (4,5) → (4,7) → (8,7) m.
Endpoints: $\mathbf{P}_0, \mathbf{P}_2$; control point: $\mathbf{P}_1$.
Parameter: $s \in [0,1]$.

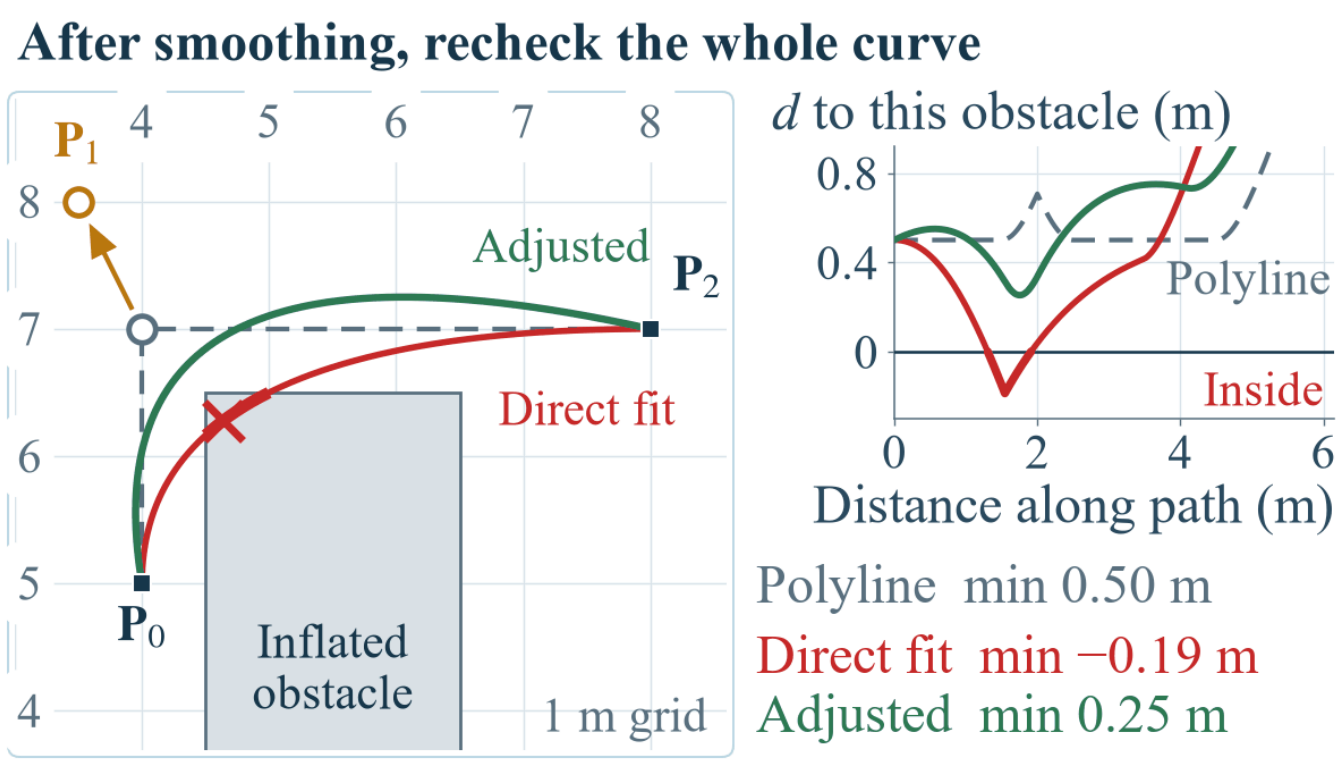
$$\mathbf{p}(s) = (1-s)^2\mathbf{P}_0 + 2(1-s)s\mathbf{P}_1 + s^2\mathbf{P}_2$$

Red: control point (4,7) m; at s=0.4 the curve enters the obstacle.

Green: move the control point to (3.5,8) m. Check the whole curve, not just control points.

6.4.4 Planning Process and Postprocessing: Checking the Smoothed Curve

Recheck the whole curve; control points alone can mislead



Smoothed-curve distance check: analytic-geometry illustration.

Distance plot: vertical axis is obstacle distance d; $d < 0$ means inside the obstacle. Gray dashed: original polyline.

Control points alone: all three are outside, yet the curve lies inside for $0.35 < s < 0.50$, reaching 0.19 m penetration.

After correction: control point (3.5,8) m; recompute the whole curve. Minimum distance 0.25 m; the entire curve passes.

6.4.4 Planning Process and Postprocessing: Speed and Acceleration Limits

Straight-line timing review: motion limits over 3 m

$$p(t) = L(10s^3 - 15s^4 + 6s^5), \quad s = t/T, \quad L = 3\ \text{m}$$

Endpoints: zero velocity and acceleration. p: line displacement; T: completion time (s).

$$v_{\max} = 1.875L/T \le 2\ \text{m/s}$$

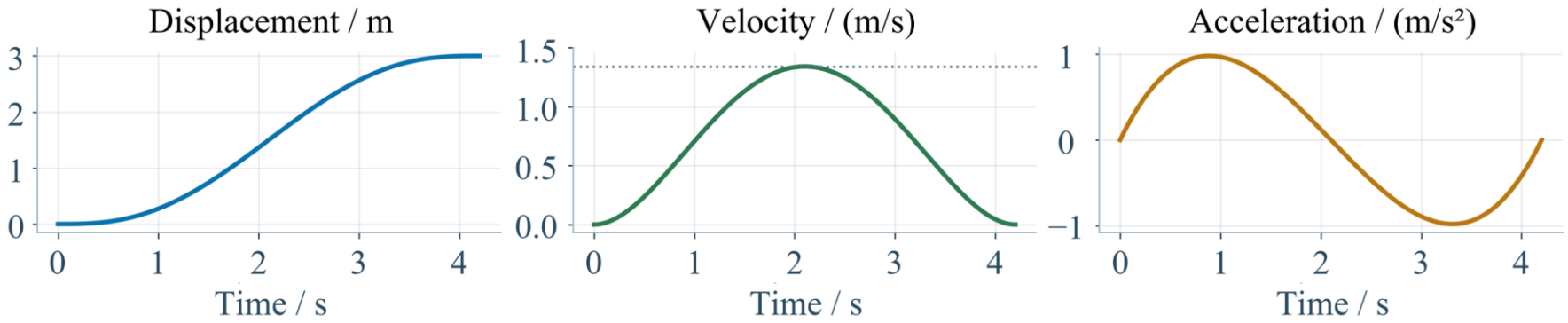
$$a_{\max} = \frac{10\sqrt{3}}{3}\frac{L}{T^2} \le 1\ \text{m/s}^2$$

Peak locations: speed at $s = 1/2$, acceleration at $s = 1/2 - \sqrt{3}/6$.
Speed gives $T \ge 2.813$ s; acceleration gives $T \ge 4.162$ s.
Choose $T = 4.2$ s and retain extra margin appropriate to actual tracking performance.

Curved path: $\mathbf{p}(t) = \mathbf{r}(\sigma(t)), \quad \mathbf{v}(t) = \mathbf{r}'(\sigma(t))\dot{\sigma}(t)$

6.4.4 Planning Process and Postprocessing: Complete Timed Trajectory

The 4.2 s trajectory meets the stated ideal constraints



Displacement, velocity, and acceleration for the same straight line

Analytic peaks: 1.339 m/s and 0.982 m/s²; endpoint velocity and acceleration are zero.

Curvature adds normal acceleration. Fixed-wing/ground vehicles: minimum turn radius and non-omnidirectional constraints.

Postprocessing: B-splines or polynomials; minimum jerk (third derivative) or snap (fourth derivative); constrain continuity.

Primes denote differentiation with respect to path parameter σ: $\mathbf{a}(t) = \mathbf{r}''(\sigma(t))\dot{\sigma}(t)^2 + \mathbf{r}'(\sigma(t))\ddot{\sigma}(t)$

6.4.4 Planning Process and Postprocessing: Final Checks

Continuous collision checking needs more than a few samples

Space: check every entire segment or curve, including inflated obstacles, map boundaries, and predicted moving obstacles.

Time: moving obstacles and vehicle trajectories need the same timebase. Check waiting and stopped periods too.

Motion: check speed, acceleration, curvature, and required higher derivatives. On exceptions, use a feasible stop or replan.

Use speed bound 2 m/s and interval 0.2 s; m_{sample}: additional sampling margin.

$$m_{\text{sample}} \geq v_{\max} \frac{\Delta t}{2} = 2 \times 0.1 = 0.2 \text{ m}$$

Static obstacles: each sample must be more than 0.2 m from the inflated obstacle.
Conditions: a known full-horizon speed bound and checks at both endpoint samples.

Learning Resources

Resources for algorithms and further reading

Discrete search, configuration space, and sampling-based planning

https://lavalle.pl/planning/

Assumptions for RRT* and PRM* asymptotic optimality

https://arxiv.org/abs/1105.1186

MPC models, prediction horizons, and feedback

https://www.do-mpc.com/en/latest/theory_mpc.html

Avoidance guide: [installation directory]/RflySimAPIs/6.RflySimExtCtrl/1.BasicExps/e11_Avoidance_Segment/e5.1/Readme.pdf

Classroom Exercises

Exercises: heuristic bounds, potential fields, and braking

Question	Key Points
Eight-connected moves, diagonal cost √2 m: from (0,0) to (3,1), is Manhattan distance admissible?	Actual cost $2+\sqrt{2}\approx 3.414$ m; Manhattan 4 m overestimates. Use diagonal distance or Euclidean $\sqrt{10}\approx 3.162$ m.
APF from (4,8.5) m: what is the combined guidance? Is speed limiting active?	Attraction (4,0.5), repulsion ≈(−3.144,−0.786); sum ≈(0.856,−0.286) m/s. Magnitude 0.90 <1, so no speed limiting.
At 1 m/s with deceleration 1 m/s², what minimum braking distance is needed?	Constant-deceleration stop: $d=\frac{v_0^2}{2a}=0.5$ m. Also allow for delay and model error, and check collisions along the braking segment.

Classroom Exercises

Exercises: change only departure time or duration

1. Keep both vehicle routes unchanged and require 0.8 m separation. What is B's minimum delay?

Answer: simultaneous motion requires $\delta/\sqrt{2} \geq 0.8$, hence $\delta \geq 1.131$ s.
A 1.2 s delay gives minimum separation 0.849 m; waiting and stopped intervals stay at least 1 m apart.

2. For the same 3 m line, reduce acceleration to 0.5 m/s² while keeping speed limited to 2 m/s. What minimum T is needed?

Answer: acceleration gives $T \geq 5.886$ s; speed needs only 2.813 s. Choose 5.886 s, giving peak speed about 0.956 m/s.

Classroom Exercises

Exercises: saved data, moving obstacles, and smoothing

Question	Key Points
Using SIM1.mat: minimum feedback distance to center (12,0) m, then net clearance from the 2 m geometric radius?	min(hypot(x_feedback-12,y_feedback)) ≈4.493 m; net clearance 4.493 − 2 = 2.493 m, or 1.493 m beyond the 3 m safety circle.
For moving obstacles, what extra information do APF, DWA, and MPC need?	All need predicted future obstacle positions or a conservative envelope. APF: repulsion; DWA: checks each time along the arc; MPC: time-varying constraints.
Smoothing variant: (4,5)→(4,7.2)→(8,7.2) m, with control point (4,7.2). Obstacle: x∈[4.5,6.5], y≤6.5 m. Does it pass?	No. $x=4+4s^2$, $y=5+4.4s-2.2s^2$. For $s\approx0.35$–0.44 the curve enters the obstacle by up to about 0.11 m. All three control points are outside; checking only them fails.

Section Summary

After finding a route, make the trajectory executable

Global: Dijkstra/A* search a graph; RRT/PRM build connections by sampling.
Results depend on graph construction and cost.

Local and multi-vehicle: APF, DWA, and MPC update motion from state.
Multiple vehicles also need timing, separation, and exception-action coordination.

Postprocessing: removing points, smoothing, and adding timing can change feasibility.
Check the whole curve, full time horizon, and motion limits.

Common error: finding a route, obtaining a smooth curve, meeting ideal constraints, and executing accurately
are four separate checks.