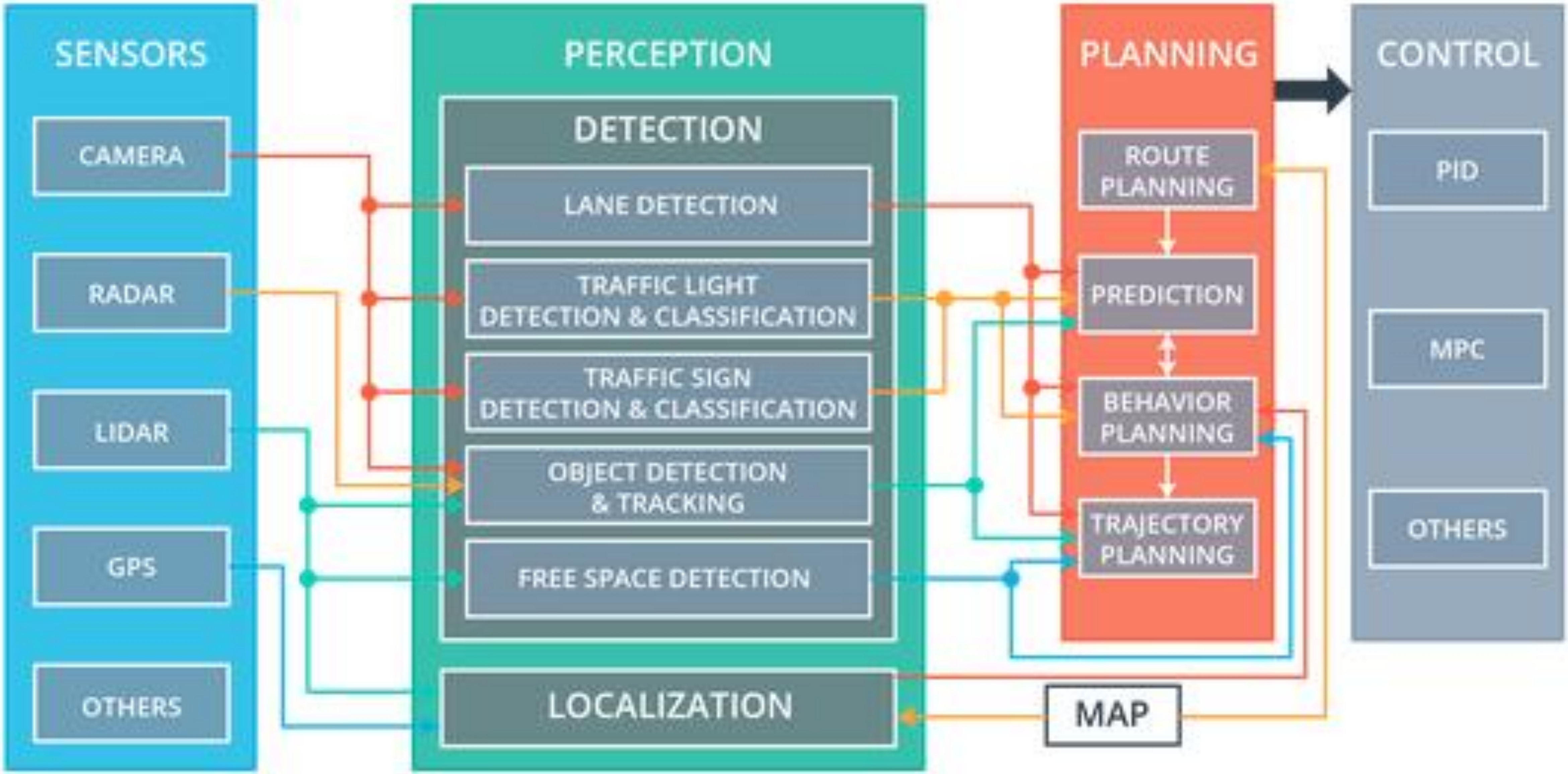
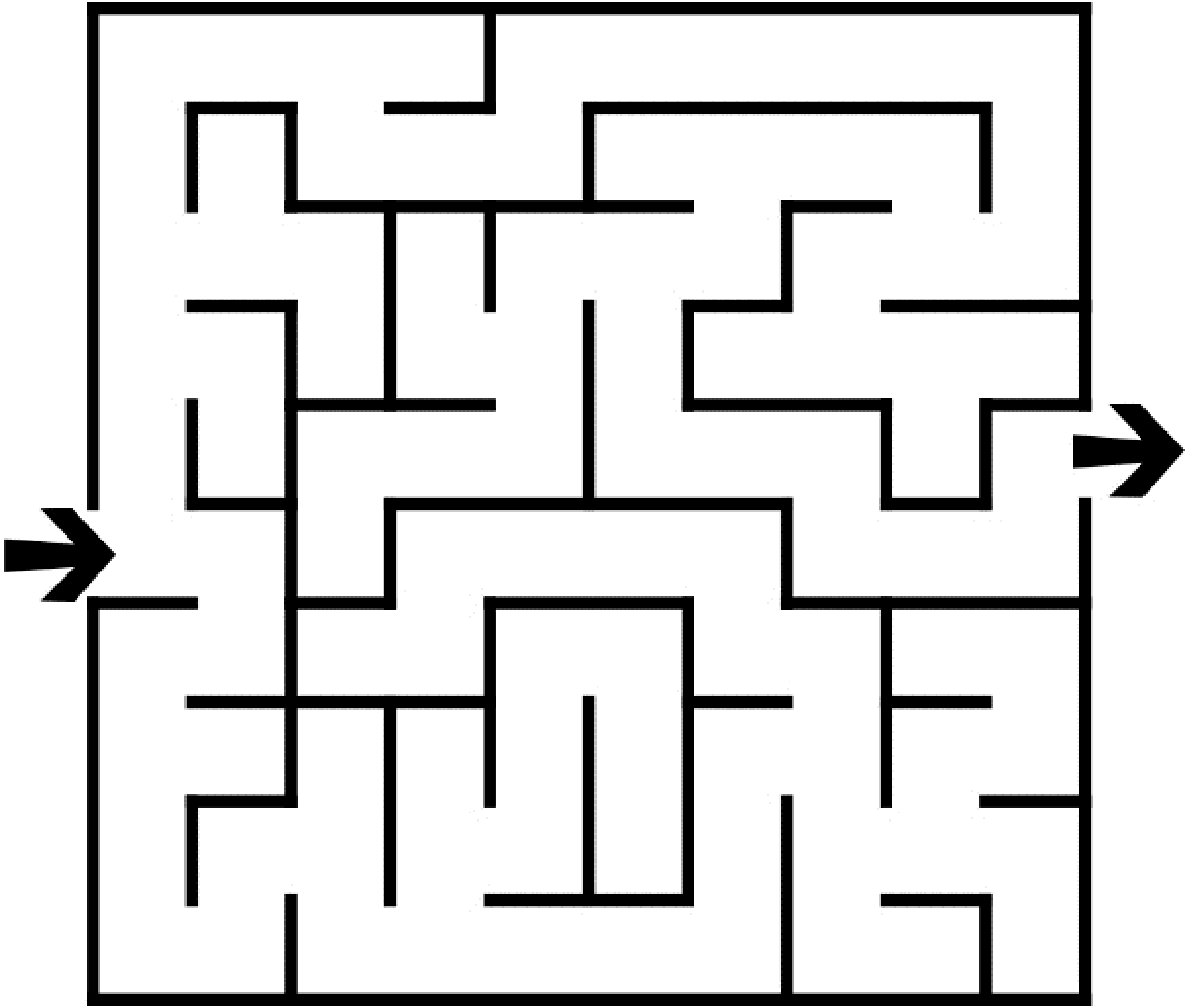

SENSORS
CAMERA
RADAR
LIDAR
GPS
OTHERS
PERCEPTION
DETECTION
LANE DETECTION
TRAFFIC LIGHT DETECTION & CLASSIFICATION
TRAFFIC SIGN DETECTION & CLASSIFICATION
OBJECT DETECTION & TRACKING
FREE SPACE DETECTION
LOCALIZATION
PLANNING
ROUTE PLANNING
PREDICTION
BEHAVIOR PLANNING
TRAJECTORY PLANNING
MAP
CONTROL
PID
MPC
OTHERS

Trajectory planning

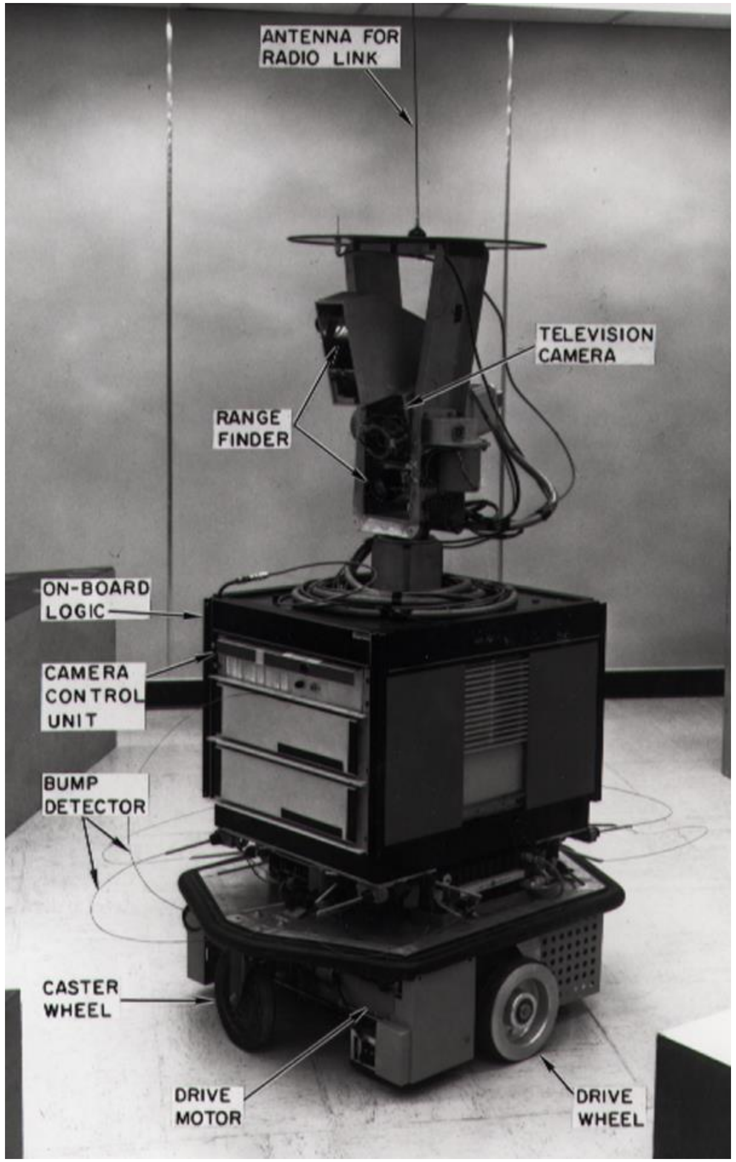


A*

F = G + H

F is the total cost of the node.
G is the distance between the current node and the start node.
H is the heuristic — estimated distance from the current node to the end node.

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9	10	11	12	13	14
6	5	4	5	6	7	8	9	10	11	12	13	14	15



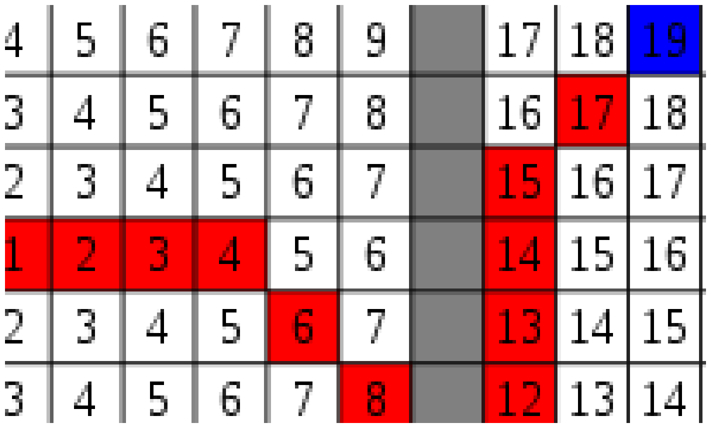
A* steps

1. Add the starting square (or node) to the **open list**.

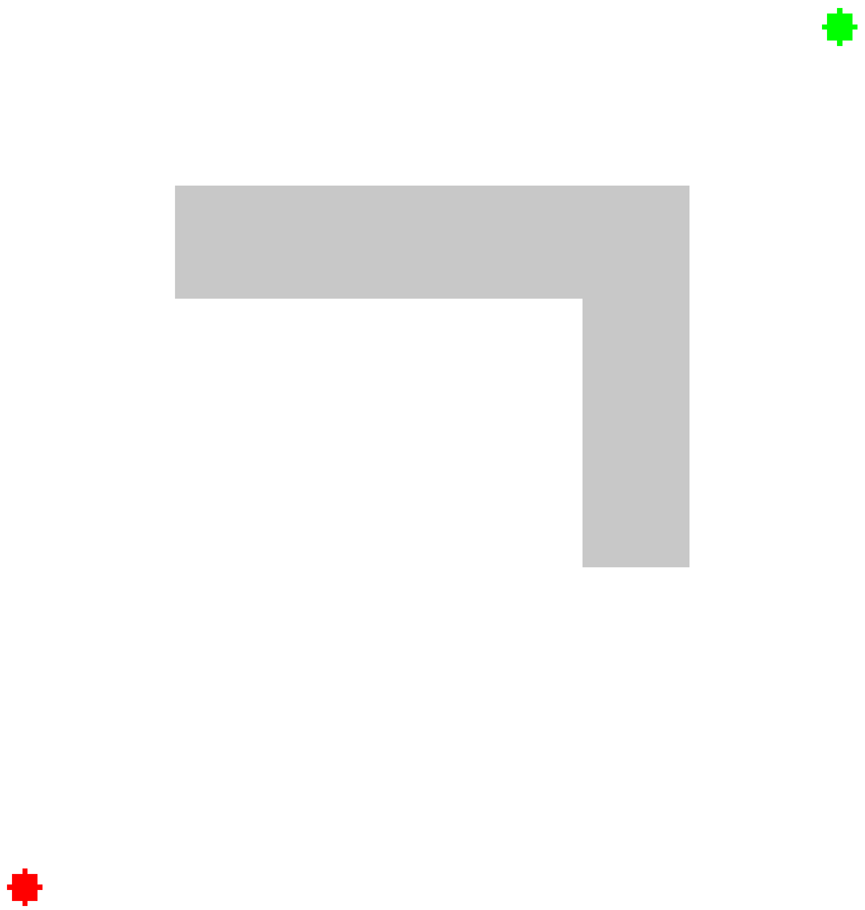
2. Repeat the following:
2.1. Look for the lowest **F** cost square on the **open list**. We refer to this as the **current square**. Put in on the **closed list**.
2.2. For each of the adjacent squares to the **current square:**
•If it is not reachable or if it is on the **closed list**, ignore it. Otherwise do the following.
•If it isn't on the open list, add it to the **open list**. Make the **current square** the parent of this square. Record the **F**, **G**, and **H** costs of the square.
•If it is on the **open list** already, check to see if this path to that square is better, using **G** cost as the measure. A lower **G** cost means that this is a better path. If so, change the parent of the square to the **current square**, and recalculate the **G** and **F** scores of the square.

3. Stop when you:
•Add the target square to the **closed list**, in which case the path has been found, or
•Fail to find the target square, and the **open list** is empty. In this case, there is no path.

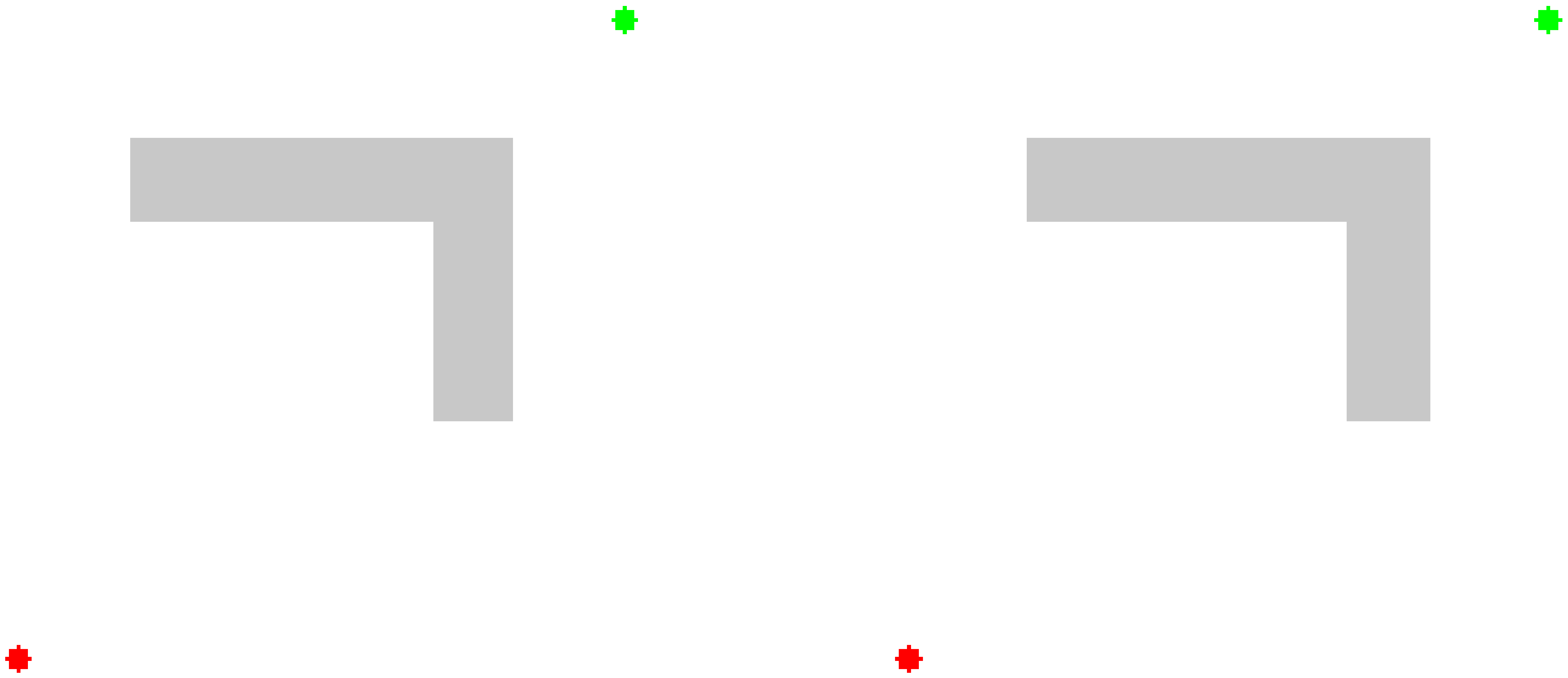
4. Restore the path.



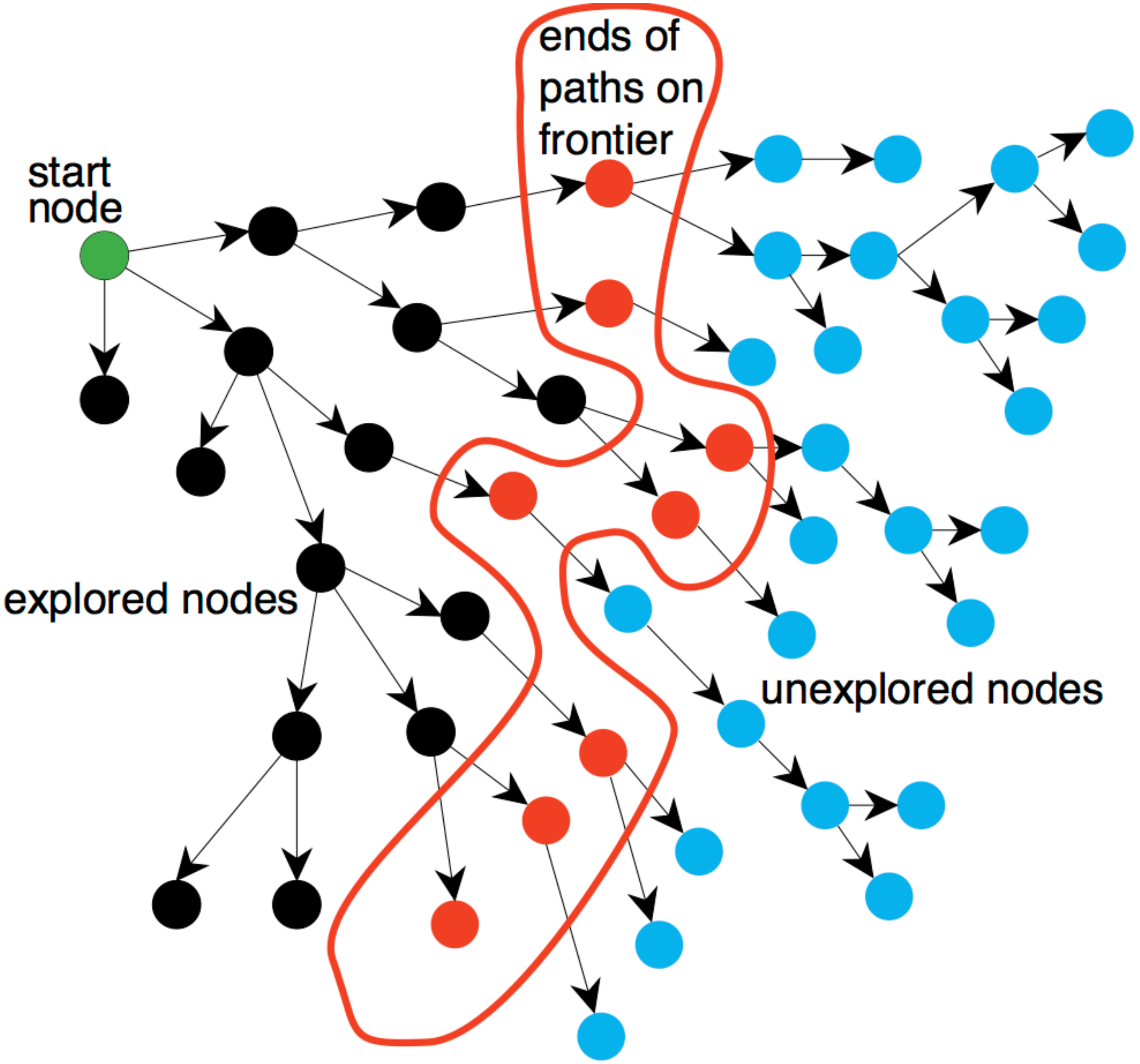
A* vs. Dijkstra's Algorithm



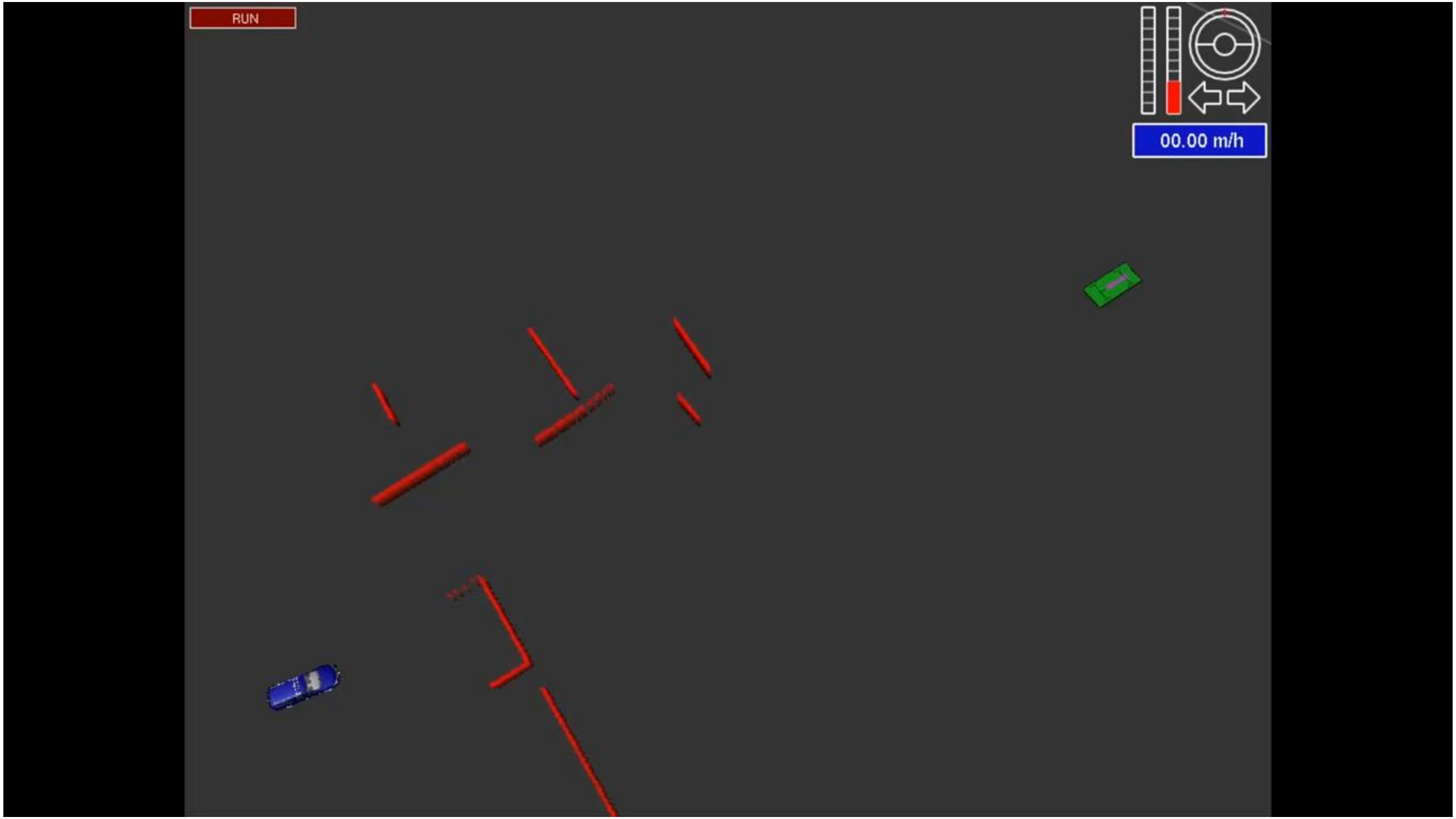
A* vs. Dijkstra's Algorithm



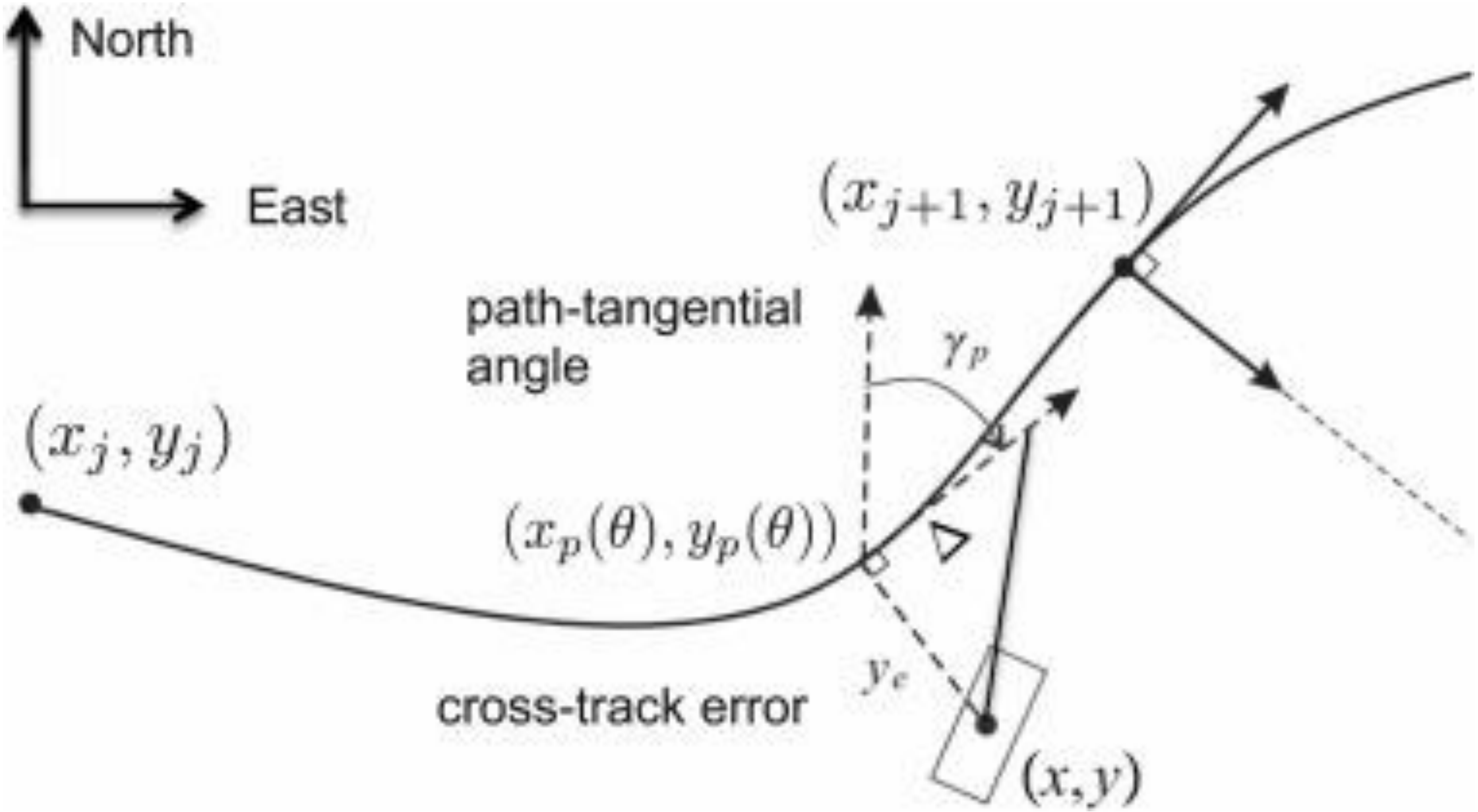
A*


ends of
paths on
frontier
start
node
explored nodes
unexplored nodes

A*

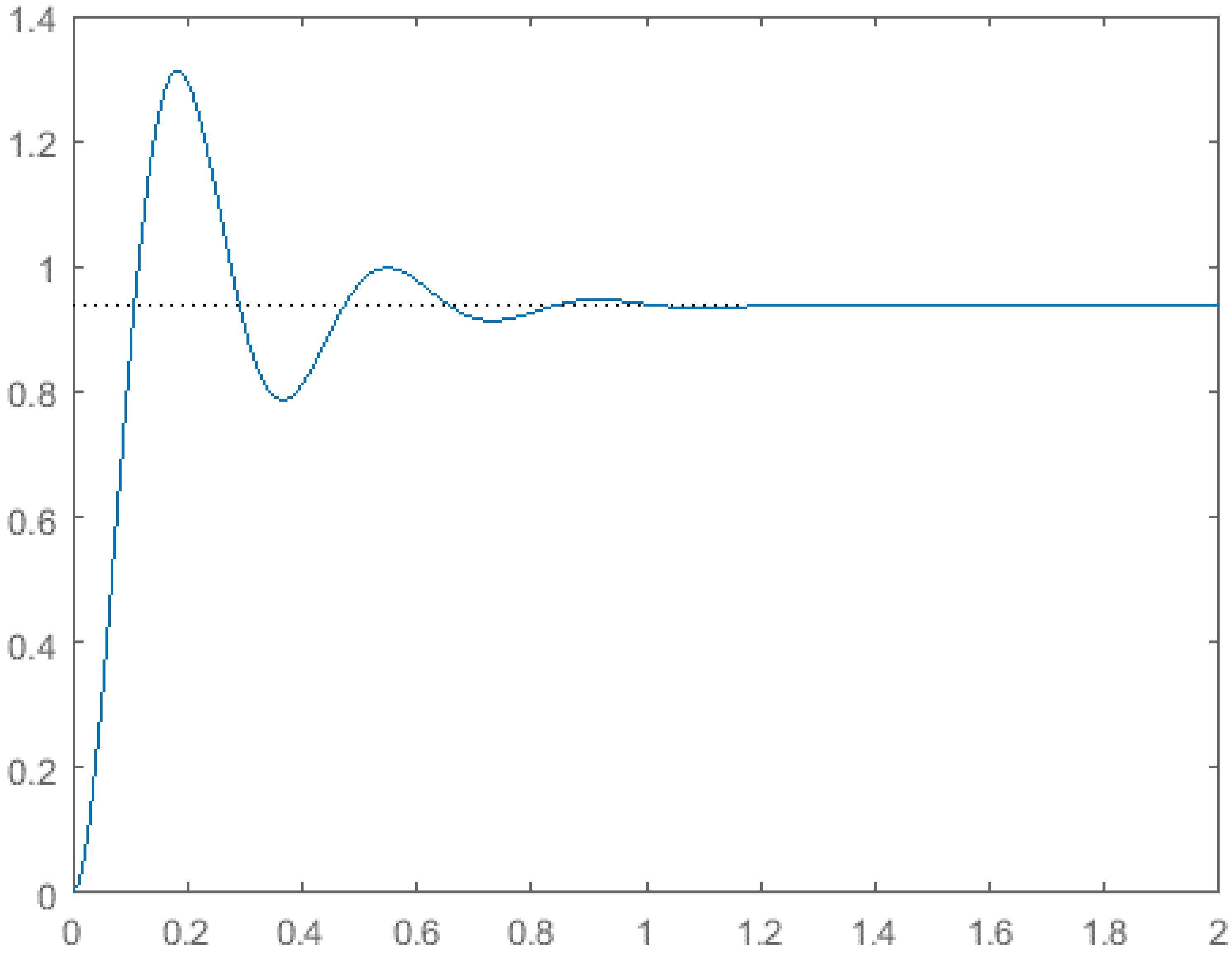


Cross Track Error



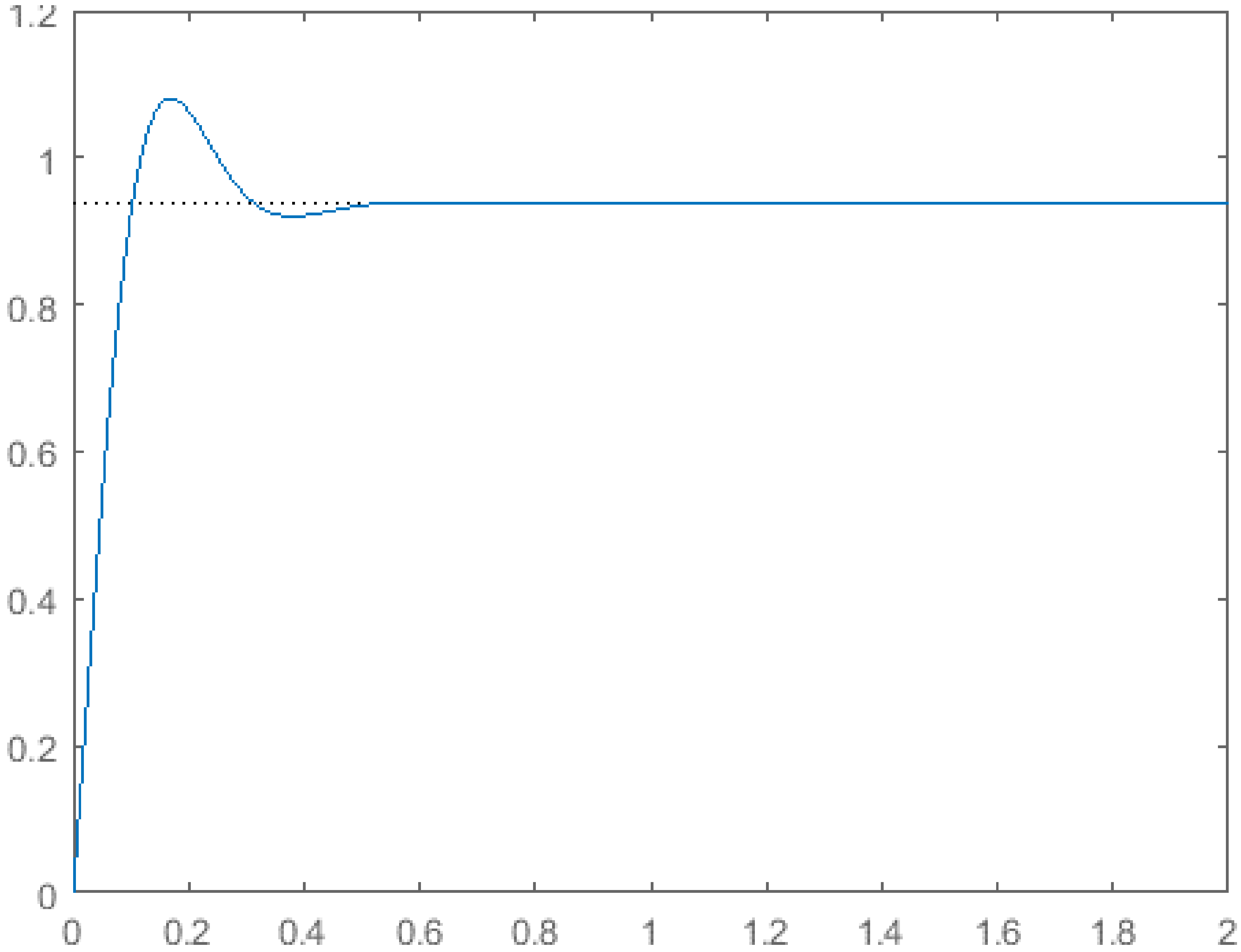
Proportional control

$$u(t) = K_{\mathrm{p}} e(t)$$



PD control

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{d} \frac{de(t)}{dt}$$



PID control

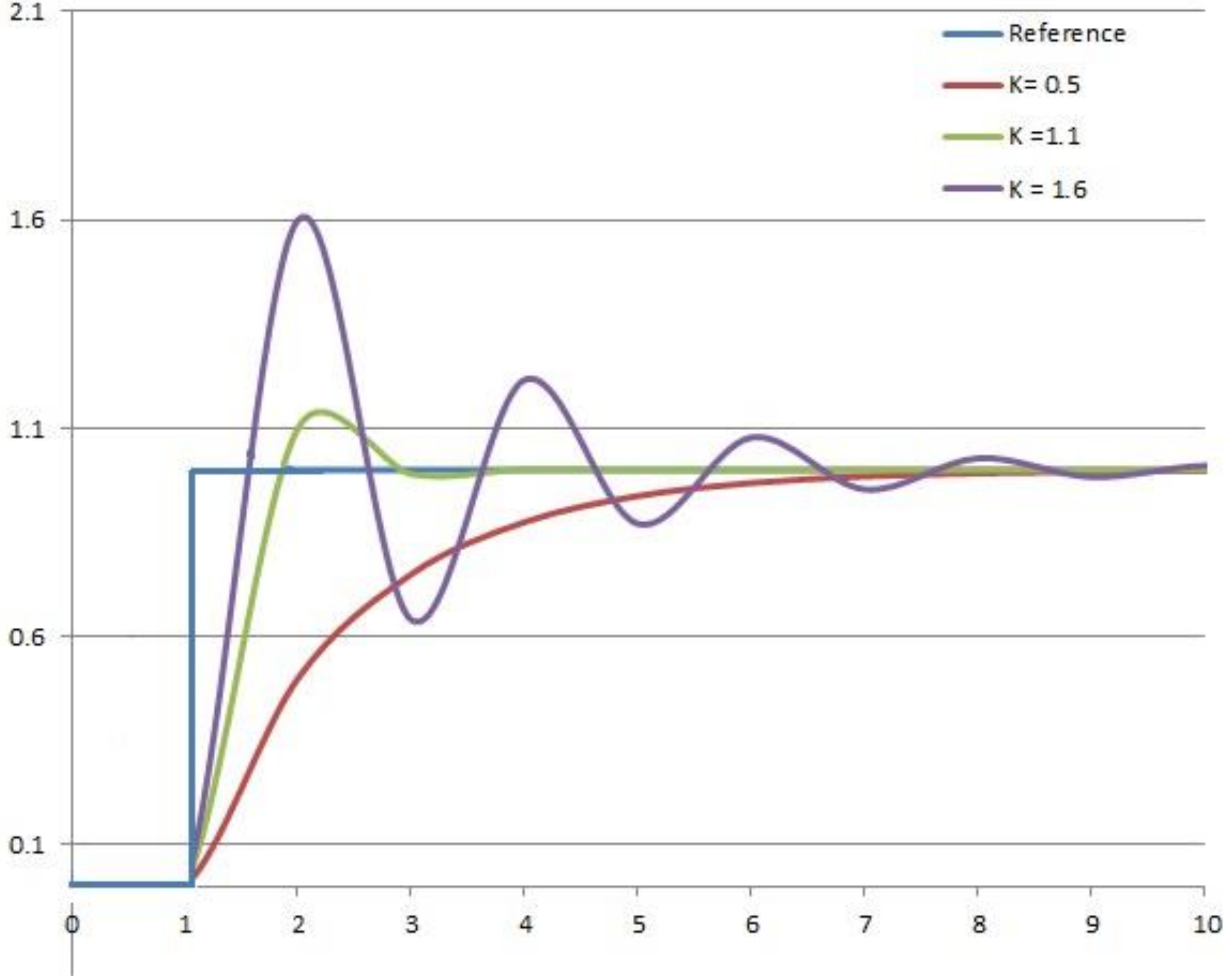
$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t')\,dt' + K_\mathrm{d} \frac{de(t)}{dt}$$

Proportional

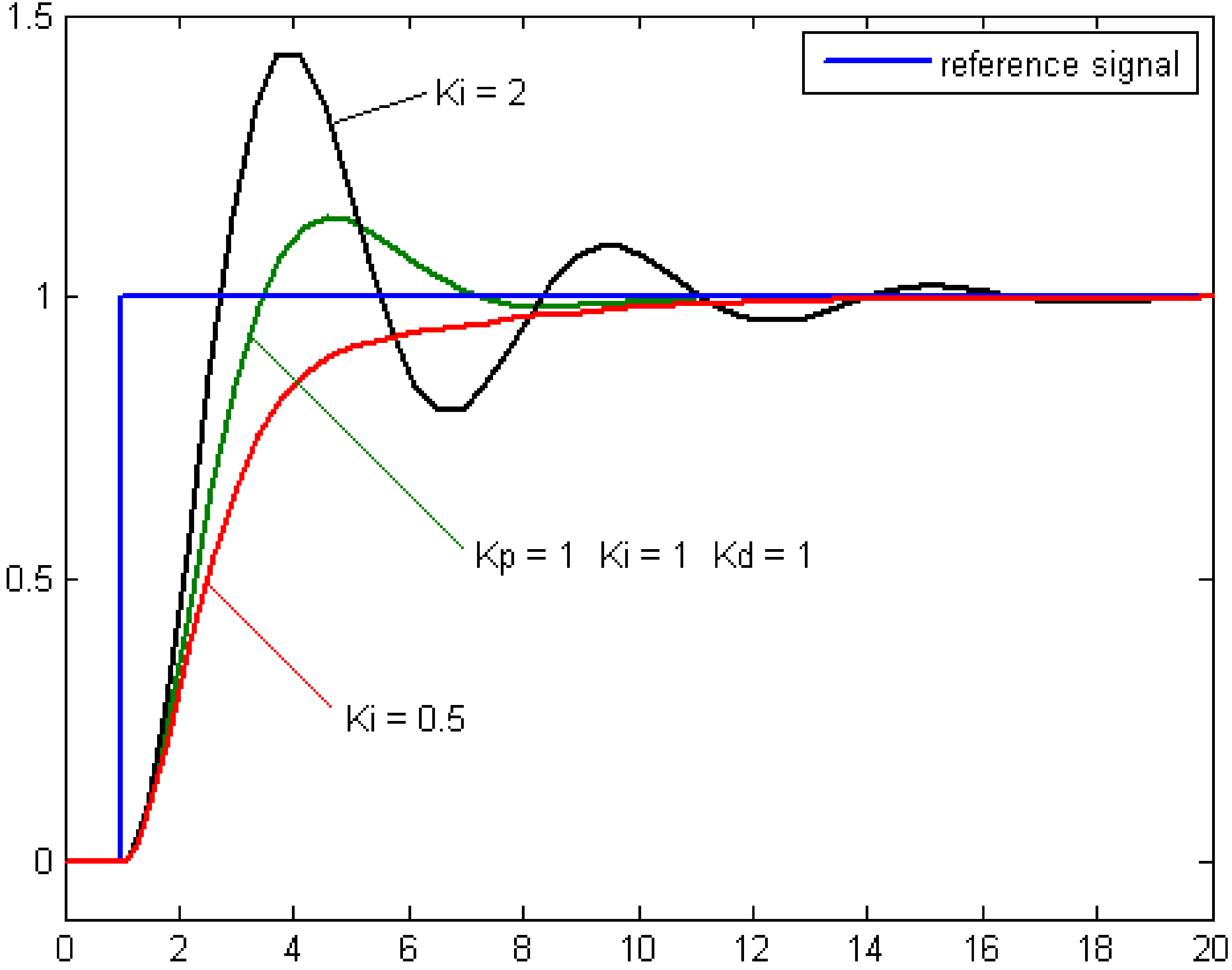
Integral

Differential

Varying Kp

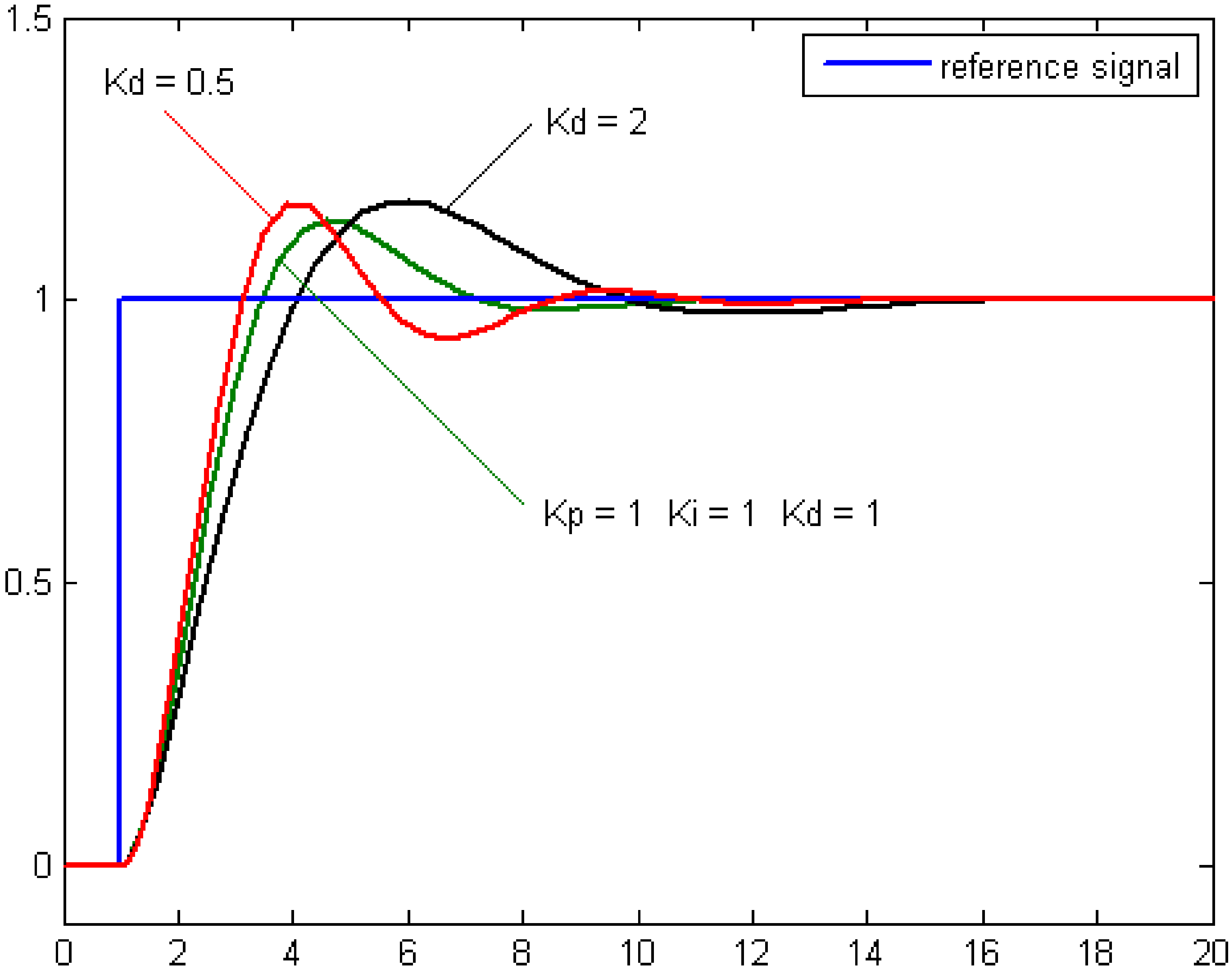


Varying Ki



reference signal
Ki = 2
Kp = 1 Ki = 1 Kd = 1
Ki = 0.5
1.5
1
0.5
0
0
2
4
6
8
10
12
14
16
18
20

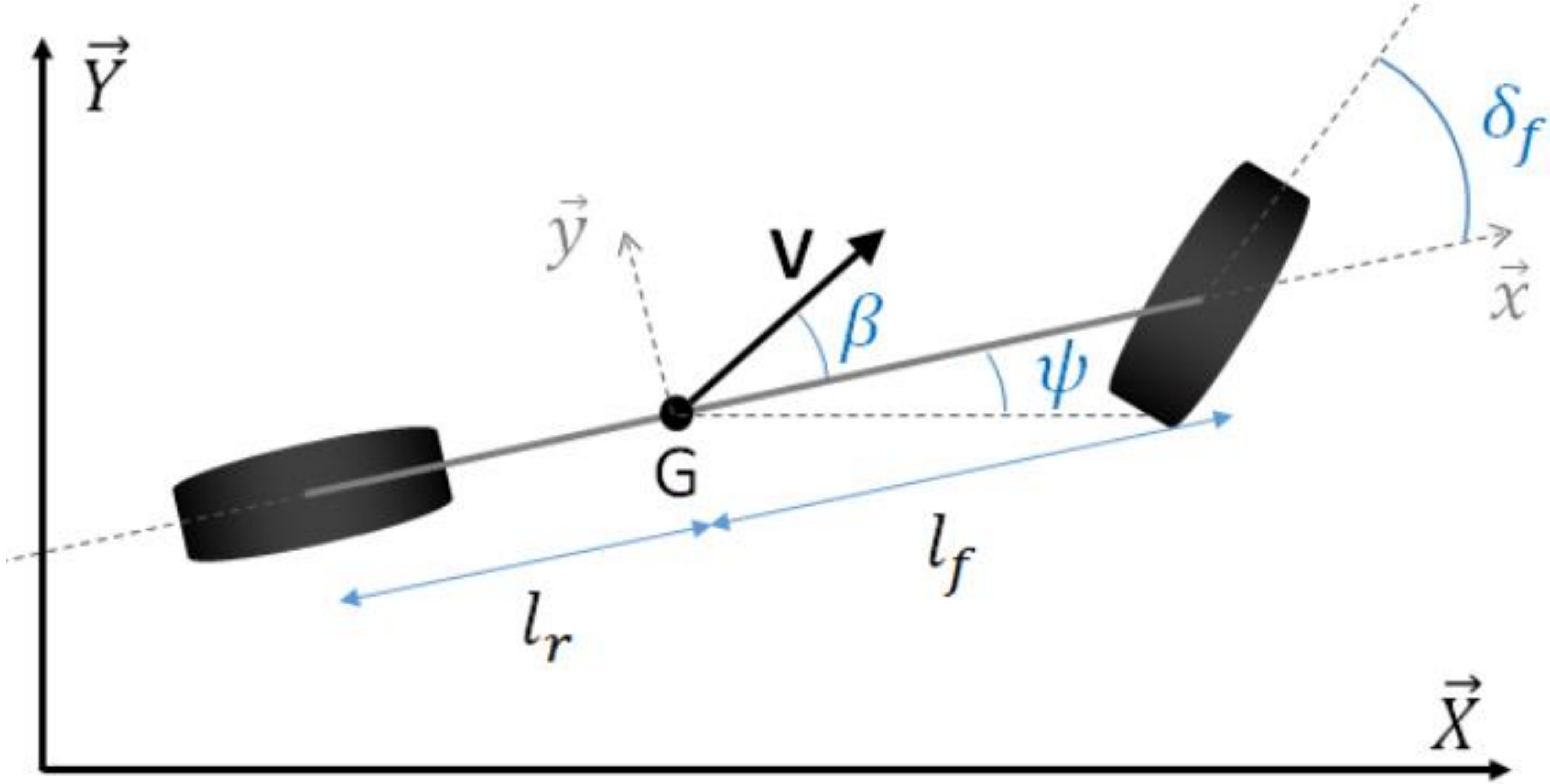
Varying Kd



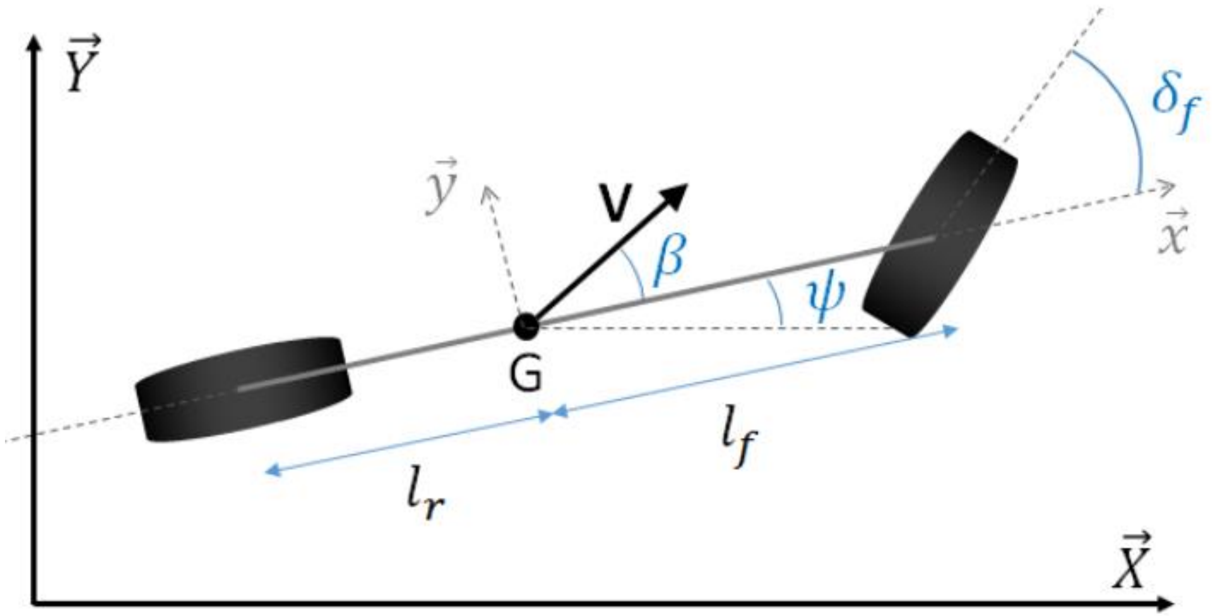
reference signal
Kd = 0.5
Kd = 2
Kp = 1 Ki = 1 Kd = 1
1.5
1
0.5
0
0
2
4
6
8
10
12
14
16
18
20

PID control in simulated environment

Kinematic bicycle model



Kinematic bicycle model



$$
\begin{aligned}
\dot{X} &= V\cos\left(\psi+\beta\left(u_2\right)\right)\\
\dot{Y} &= V\sin\left(\psi+\beta\left(u_2\right)\right)\\
\dot{V} &= u_1\\
\dot{\psi} &= \frac{V}{l_r}\sin\left(\beta\left(u_2\right)\right)
\end{aligned}
$$

u_1 - acceleration

u_2 - steering angle

$$\beta(u_2) = \arctan\left(\tan(u_2)\frac{l_r}{l_f+l_r}\right)$$

Predictive model

$$x_{t+1} = x_t + v_t \cos(\psi_t) dt$$

$$y_{t+1} = y_t + v_t \sin(\psi_t) dt$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{l_f} \delta_t dt$$

$$v_{t+1} = v_t + a_t dt$$

$$cte_{t+1} = f(x_t) - y_t + v_t \sin(e\psi_t) dt$$

$$e\psi_{t+1} = \psi_t + \psi des_t \frac{v_t}{l_f} \delta_t dt$$

MPC cost function and constraints

- Cross-track error.
- Heading error.
- Speed cost.
- Steering cost.
- Acceleration cost.
- Steering rate change.
- Acceleration rate change (jerk).

$$J = \sum_{t=1}^{N} w_{cte}||cte_t||^2 + w_{e\psi}||e\psi_t||^2 + w_v||v_t - v_{target}||^2$$

$$+ \sum_{t=1}^{N-1} w_\delta||\delta_t||^2 + w_a||a_t||^2$$

$$+ \sum_{t=2}^{N} w_{rate_\delta}||\delta_t - \delta_{t-1}||^2 + w_{rate_a}||a_t - a_{t-1}||^2$$

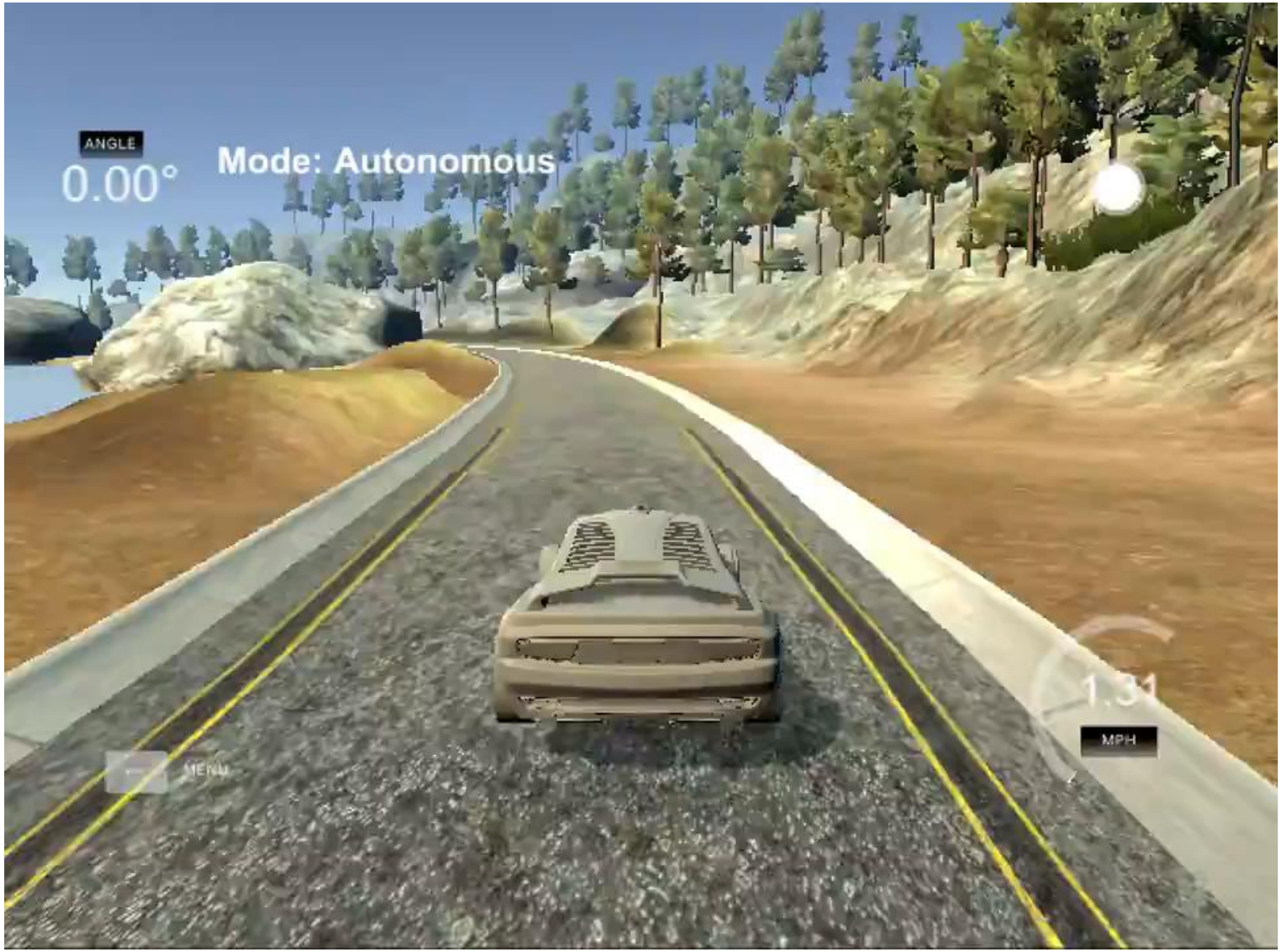
Constraints:

$$\delta \in [-25°, 25°]$$

$$a \in [-1, 1]$$

Solve for next N points with QP solver.

MPC in simulated environment

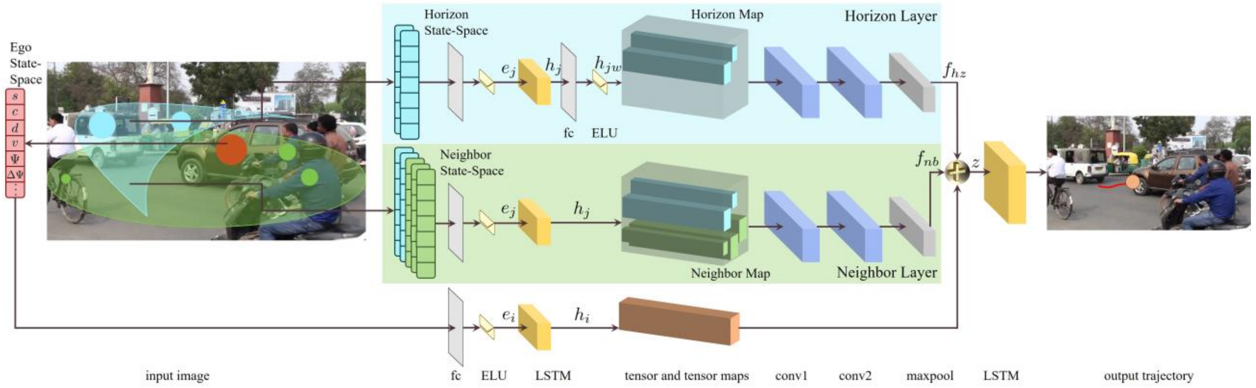


Prediction

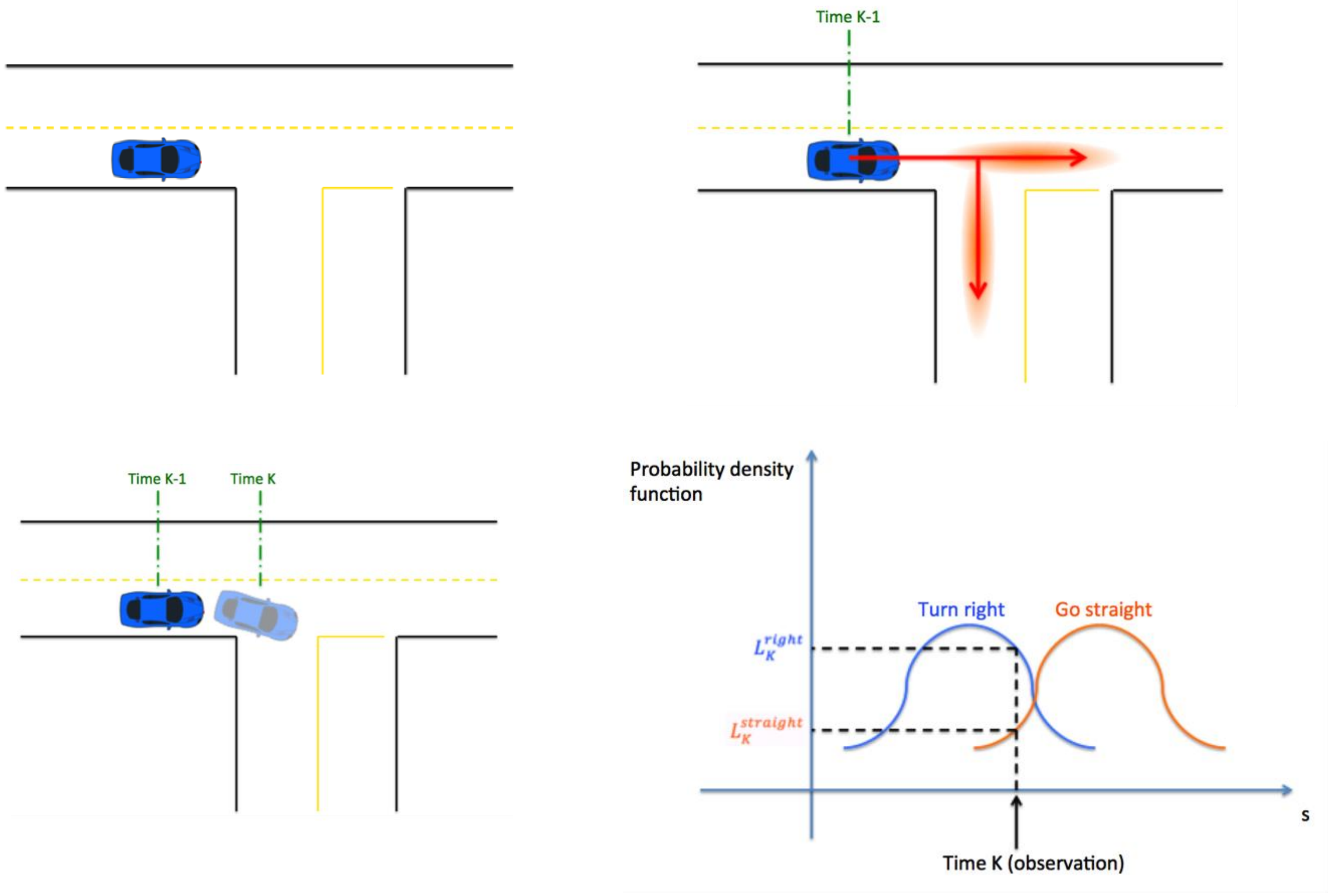
Model-based

$$\mu_k^{(i)} = \frac{\mu_{k-1}^{(i)} L_k^{(i)}}{\sum_{j=1}^{M} \mu_{k-1}^{(j)} L_k^{(j)}}$$

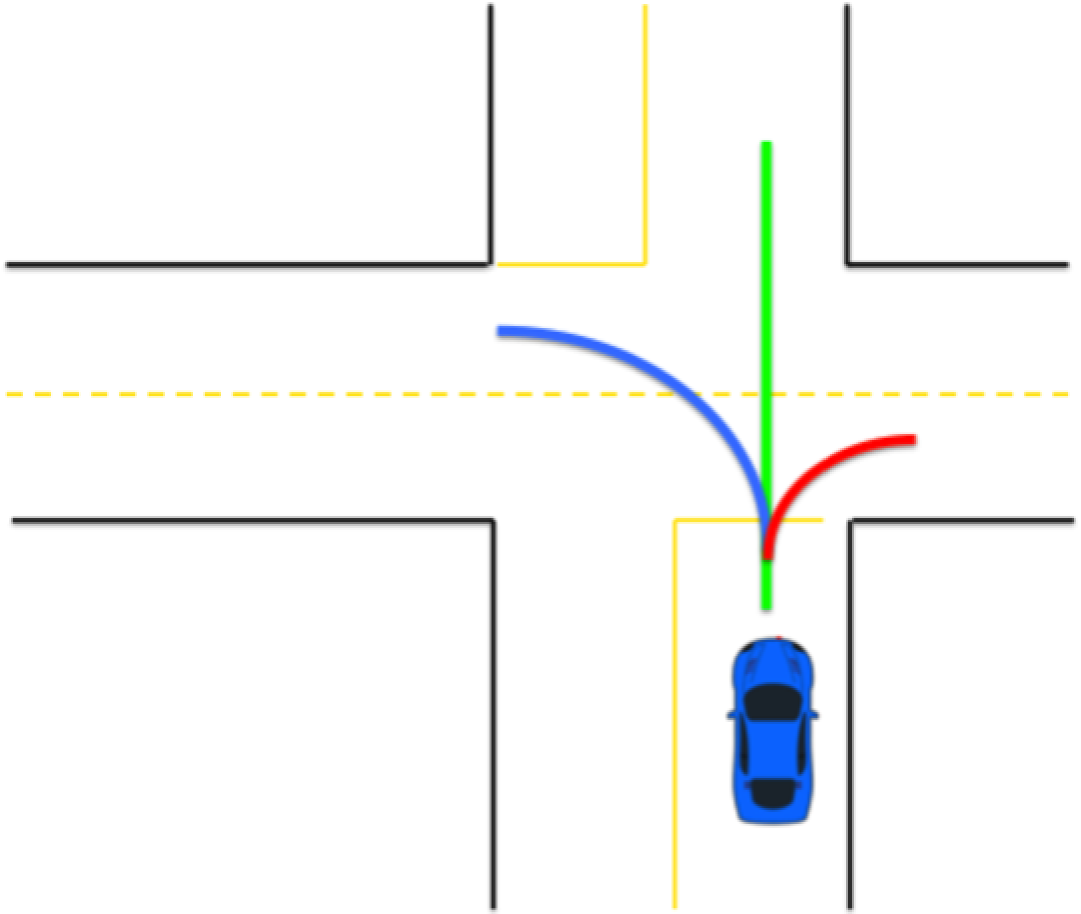
Data-driven

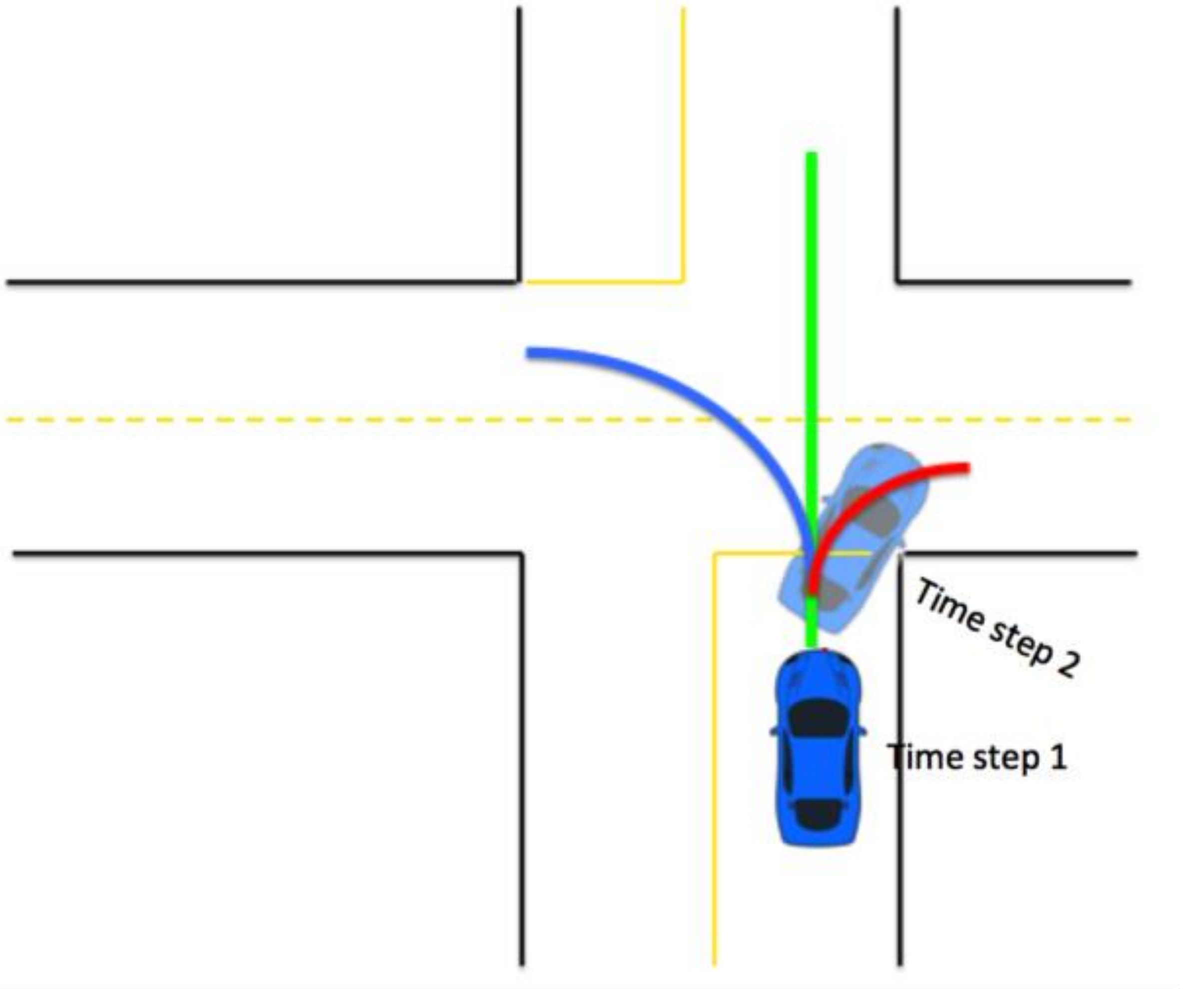


Model based prediction

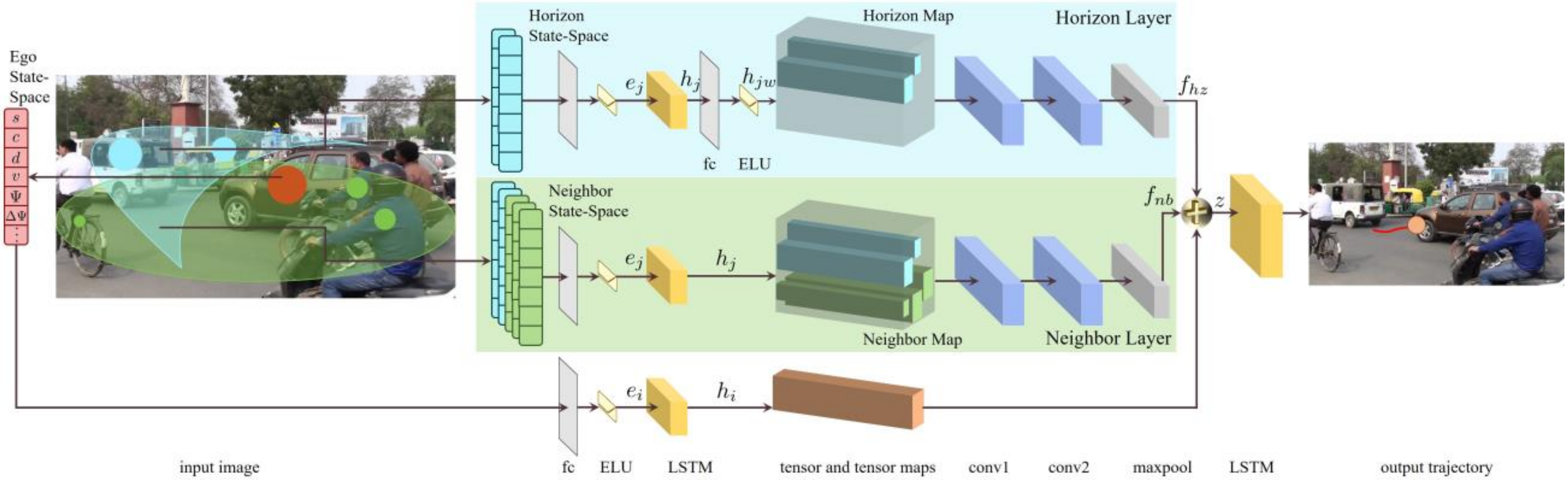


Data-driven prediction, cluster trajectories





TraPHic



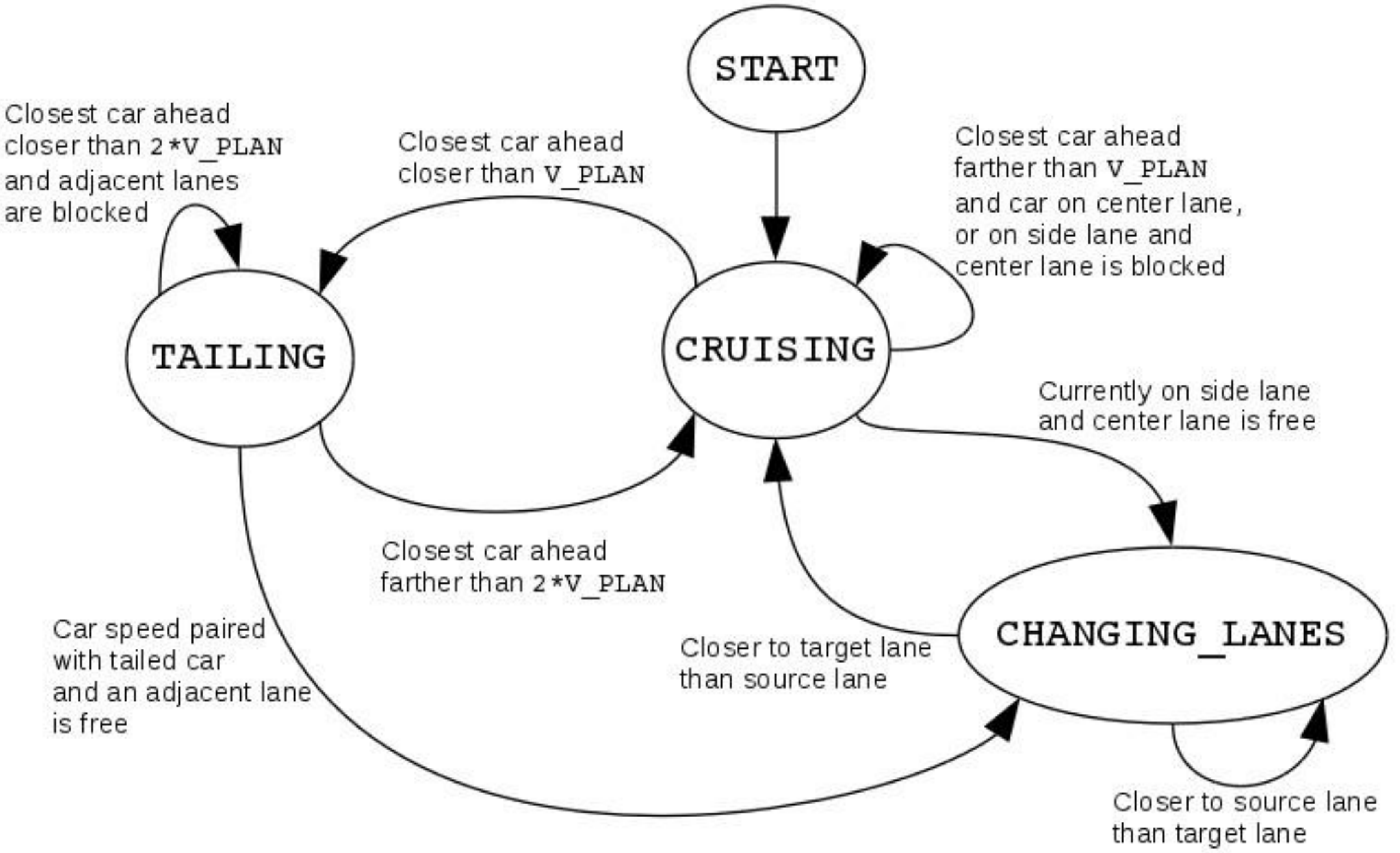
https://arxiv.org/abs/1812.04767

TraPHic

https://arxiv.org/abs/1812.04767

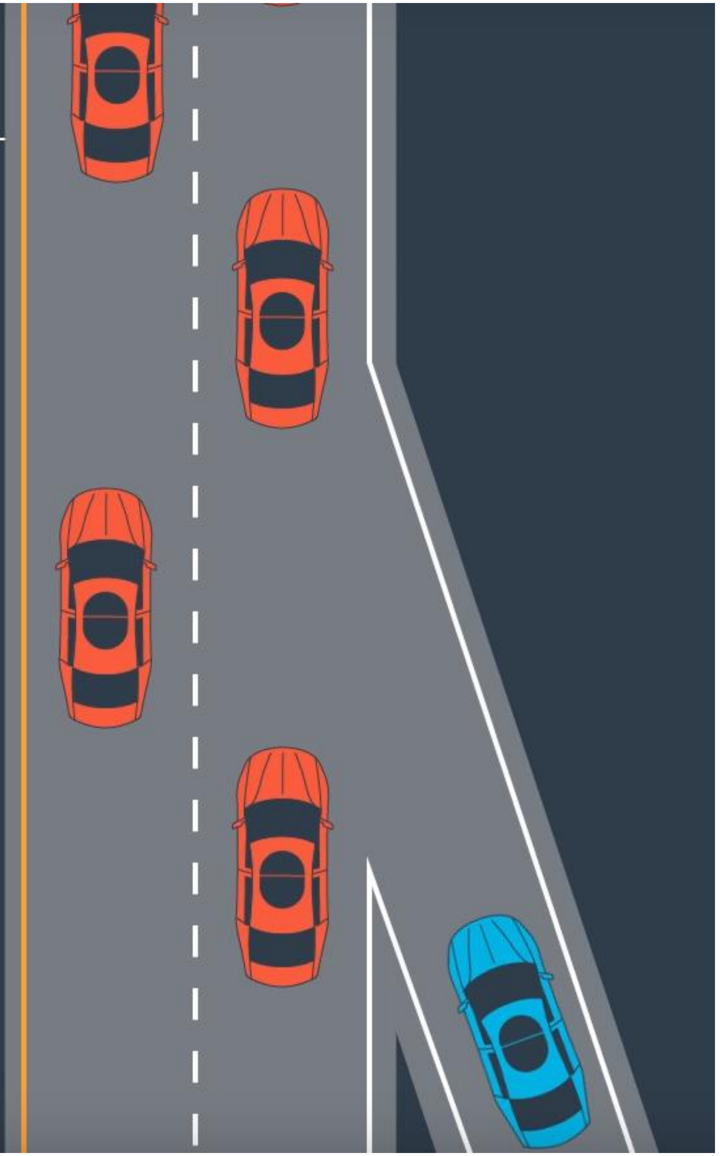
Behavior

Behavior with Finite State Machines

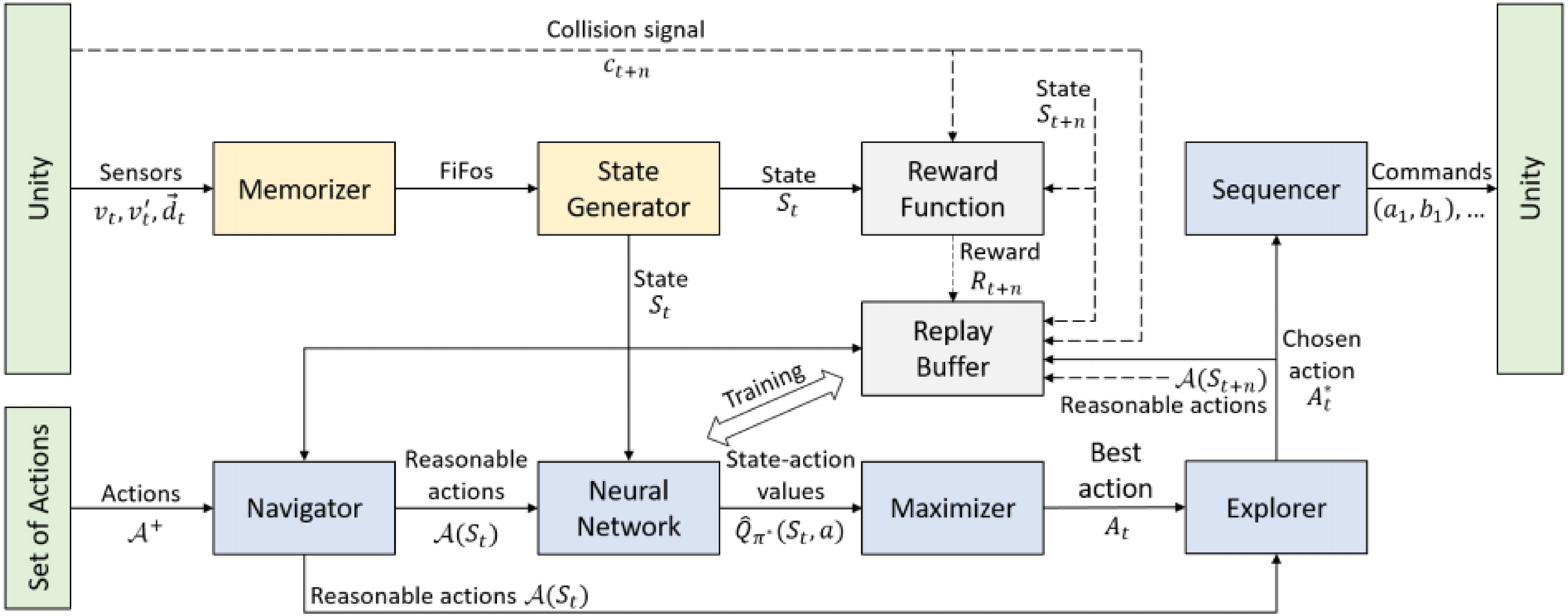


Cost functions for FSM

Class	Position	Velocity	Acceleration
Feasibility	Avoids Collision?		Acceleration is feasible for car?
Safety	Buffer Distance	Speed ~= traffic speed	
Legality	Stays on Road?	Speed < speed limit?	
Comfort	Near center of current lane		Low change in acceleration (jerk)
Efficiency	Desired Lane	Speed ~= speed limit	

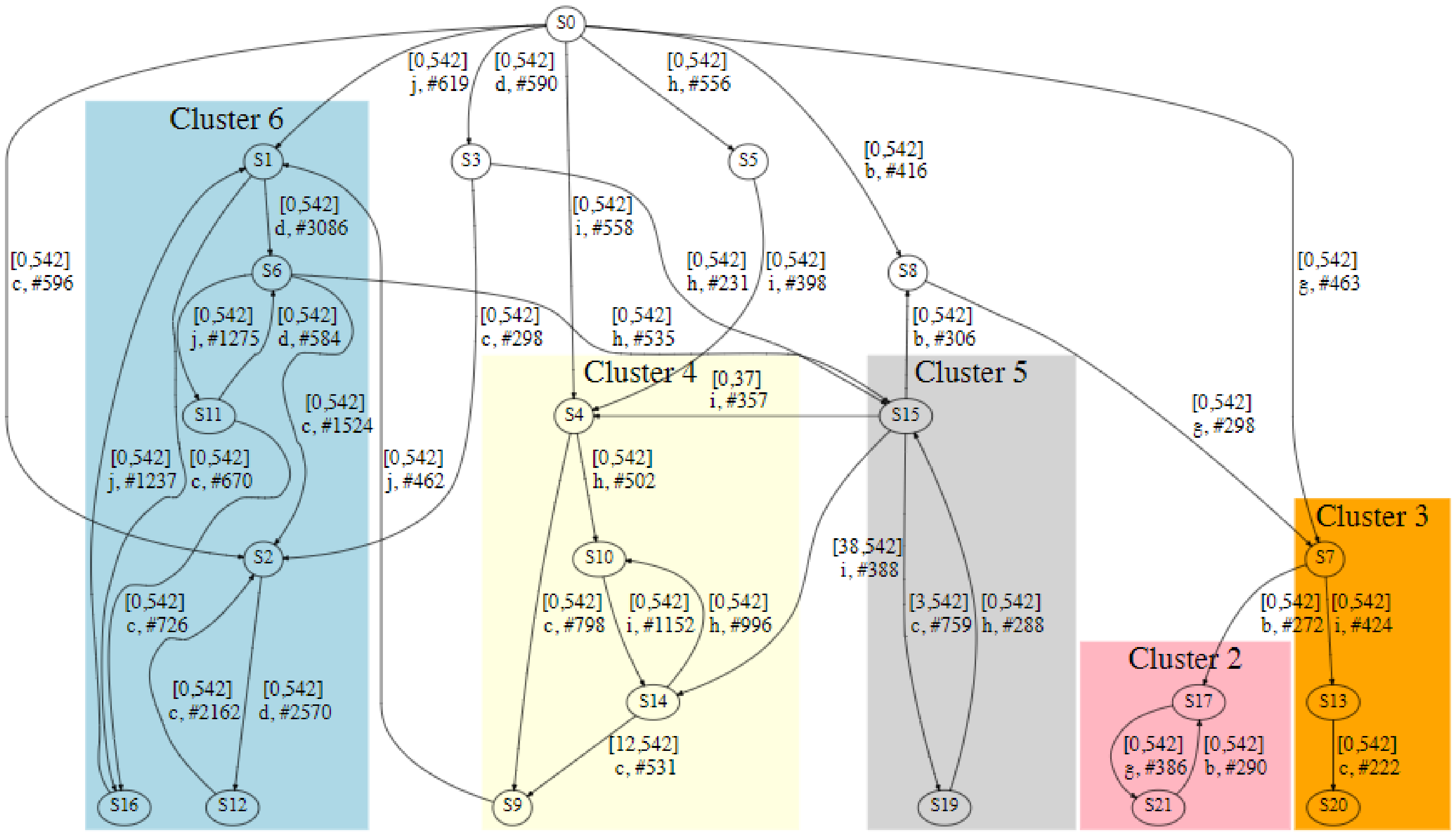


Deep Reinforcement Learning for FSM



https://arxiv.org/pdf/1811.07868.pdf

Learned states



Cluster ID	Dominating states	Description
1	0,2,3,8,13	without significant meaning
2	17, 21	steady long distance car-following
3	7,13,20	intermediate process
4	4, 9, 10, 14	steady medium distance car-following
5	12, 15, 19	intermediate process
6	1, 2, 6, 11, 12, 16	steady short distance car-following

https://www.sciencedirect.com/science/article/pii/S2405896317307735

Homework

- Create a PID controller for a simple robot.
- Write cost functions for FSM for lane changing behavior.