Minimum gaps

# Definitions

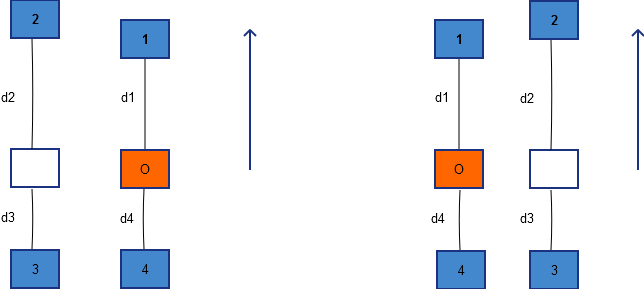
Maximum speed for all cars is

Maximum acceleration for all cars is

Maximum deceleration for all cars is

Current time step is

Next time step is



The figure above illustrates two situations. The orange car is the one in consideration and its 4 neighbors. ’s are the corresponding distance between cars. Each situation is corresponding to the lane where the orange one is. The arrows specify the direction of cars.

# Minimum gap with the car in front

Car ahead C1, speed v1t

My car CO, speed vOt

Current distance between C1 and CO: d1

Time needed for a car at speed v to stop from this moment in second:

m = (ceiling) (1)

Distance needed for a car to stop from the current speed in meter:

dneed =

Consider for the next time step t+1:

In the worst case, C1 may break, so min(v1t+1 ) = max( v1t – decM, 0)

If cars need to reduce speed to 0, after i time steps, C1 advances (v1t – i\*decM )with i is the number of seconds passed from the time t+1 when it started to brake, and CO advances (vOt+1 – i\*decM ). Hence the difference in distance at each time step is the difference in speed b/w two cars:

If : need reserved space for the worst case,

reserved gap = (2)

with m is the time CO needs to reduce speed to 0 from the speed (see equation 1)

Else, we don’t need it, reserved gap = 0

So **for the next time step t+1**, the distance needed for CO at time step t is:

D1\_need(t) = distance to advance with speed + reserved gap (3)

with:

min() =

max() =

Similarly, we could calculate the distance needed for d2 in case there is no car running parallel with CO on the other lane.

# Minimum gap with the car behind on the other lane

Car behind C3, speed v3t

My car CO, speed vOt

Current distance between C3 and CO: d3

A safety distance for C3 at time step is, based on equation (2):

D3\_need(t) = distance to advance with speed + reserved gap behind (4)

with:

min() =

max() =

If , reserved gap behind = 0

Else:

reserved gap behind = m \* (5)