Check Your Gap or Get Scrapped

An Investigation of a Car Following Model

Kaeshav Danesh   Kevin Phan

April 23, 2022

Table of Contents

Table of Contents

Introduction

- Traffic flow theory deals with modeling vehicular flow.
- Focus on microscopic model which model cars as a single unit.
- Goal: Model and examine
  1. Homogeneous traffic
  2. Obstacles on a road
  3. Multi-lane bottleneck

Table of Contents

Introducing variables

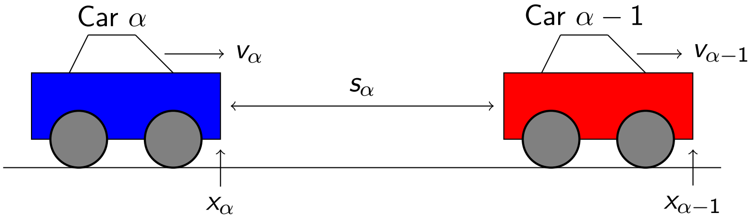

Figure 1: Defining index, position, velocity, and gap of a car.

- x_α, position of αth car.
- v_α, velocity of αth car.
- a_α, acceleration of αth car.
- s_α, gap of αth car.
- Will denote the car $\alpha - 1$ by car l.

Coupled Differential Equations

$$\frac{\mathrm{d}x_\alpha(t)}{\mathrm{d}t} = v_\alpha(t),$$
$$\frac{\mathrm{d}v_\alpha(t)}{\mathrm{d}t} = a_{\mathrm{mic}}(s_\alpha, v_\alpha, v_l).$$

Each car following model has a specific acceleration function: $a_{\mathrm{mic}}(s_\alpha, v_\alpha, v_l)$.

Full Velocity Difference Model

$$a_{\mathrm{mic}}(s_\alpha, v_\alpha, v_l) = \frac{v_{\mathrm{opt}}(s) - v_\alpha}{\tau} - \gamma \Delta v,$$

$$v_{\mathrm{opt}}(s) = \max\left(0, \min\left(v_0, \frac{s - s_0}{T}\right)\right).$$

---

v_0	desired speed
s_0	minimum distance gap
T	time gap
τ	speed adaptation time
γ	speed difference sensitivity

Optimal Velocity Function

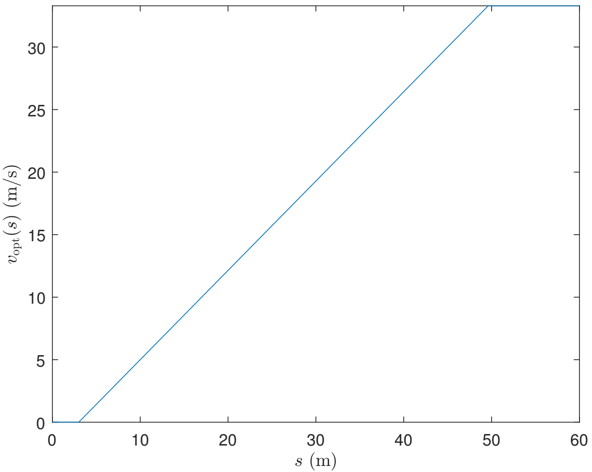

Figure 2: Graph of the optimal velocity function over a range of gaps.

Table of Contents

Forwards Euler Method Scheme

During lane changes, the acceleration function is not continuous, so Euler's Method would be the most stable:

$$v_\alpha(t+\Delta t) = v_\alpha(t) + a_{\mathrm{mic}}(s_\alpha(t), v_\alpha(t), v_l(t))\Delta t,$$
$$x_\alpha(t+\Delta t) = x_\alpha(t) + \frac{v_\alpha(t) + v_\alpha(t+\Delta t)}{2}\Delta t.$$

Lane Changes

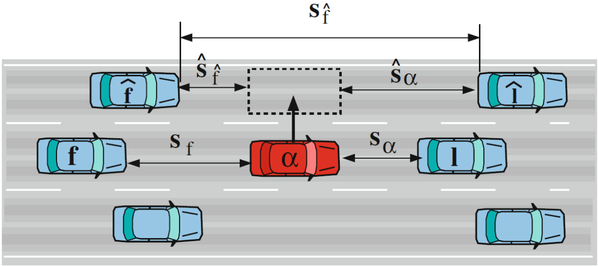

Figure 3: Multi-lane notation. From Treiber and Kesting *Traffic Flow Dynamics*

Define minimum $\hat{s}_{\hat{f}}$ and $\hat{s}_\alpha$ needed to change lanes:

$$s_{\text{safe}}(v_{\hat{f}}, v_\alpha) = v_{\text{opt}}^{-1}\left[v_{\hat{f}} - \tau b_{\text{safe}} + \tau\gamma(v_{\hat{f}} - v_\alpha)\right],$$

$$s_{\text{adv}} = s_\alpha + v_{\text{opt}}^{-1}\left[\tau(\Delta a + a_{\text{bias}} + \gamma(v_l - v_{\hat{l}}))\right].$$

Pseudocode for FVDM

Algorithm 1 Simplified algorithm for FDVM with lane changes

Require: Initial state variables for each car at $t = 0$.
Require: carArr, an array of cars.

```
for i = 1 :numsteps do
    for j = length(carArr):−1:1 do
        State variables of jth car ← Update jth car by a timestep.
        New lane of jth car ← carArr(j).changeLane()
    end for
    sort(carArr)
end for
```

Table of Contents

Homogeneous Traffic

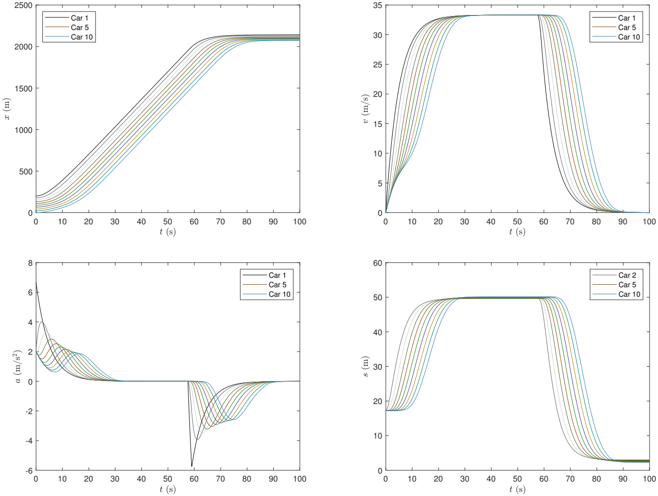

Figure 4: Position, velocity, acceleration, and gap versus time

Obstacle

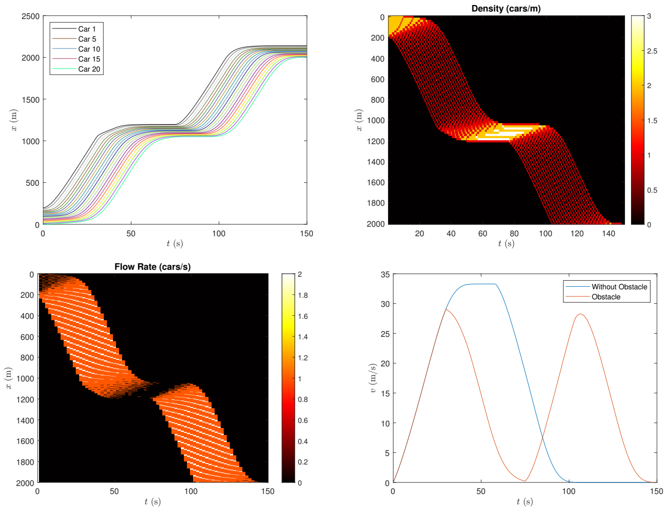

Figure 5: Position versus time graph.

Multi-lane Bottleneck

Conclusion