Car-following Models, Gipps Model

CIVE.5490, UMass Lowell

Zhengbing He, Ph.D.
https://www.GoTrafficGo.com

March 28, 2024

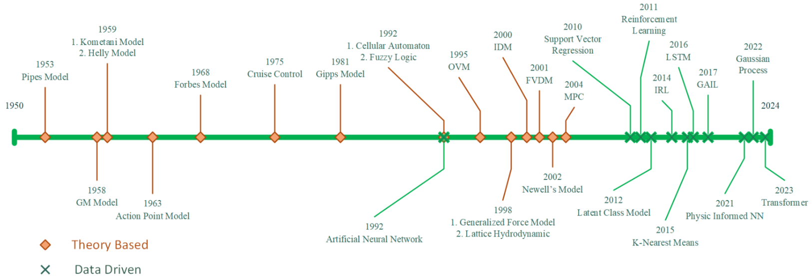
Outline

Outline

Introduction

- Gipps, P. (1981). *A behavioural car-following model for computer simulation*. Transportation Research Part B
- The Gipps model is a reaction-time-based car-following model, i.e., The speed and location of a vehicle is updated for every reaction time τ, i.e. $v_i(t+\tau) = V(t)$
- To implement the Gipps model, the Runge-Kutta scheme is needed, because it is given in a discrete-time formulation of speed

Introduction



Tianya Zhang, et al., Car-Following Models: A Multidisciplinary Review, arXiv:2304.07143v, 2024

Outline

Model structure

- The Gipps model was proposed in the discrete-time generic formulation. The speed function reads

$$v_i(t+\tau) = V[x_i(t), x_{i-1}(t), v_i(t), v_{i-1}(t)]$$

- Thus, we solve it by following

$$\boxed{\dot{x}_i(t) = v_i(t+\Delta t) = V(\cdot)} \xrightarrow{\textbf{RK}} \boxed{x_i(t+\Delta t)}$$

Outline

Model detail

Combining the speed in free-flow and congested conditions, the speed function reads.

$$v_i(t+\tau) = \min\left\{v_i^{\text{free}}(t+\tau), v_i^{\text{cong}}(t+\tau)\right\}$$

It turns out the smaller one between the derived free-flow speed and congested speed.

Speed function in congested conditions

$v_i^{\text{cong}}(t)$ is proposed as follows

$$v_i^{cong}(t+\tau) =$$

$$B_i\left(\frac{\tau}{2}+\theta\right)+\sqrt{B_i^2\left(\frac{\tau}{2}+\theta\right)^2+B_i\left\{2[x_{i-1}(t)-x_i(t)-L_{i-1}]-\tau v_i(t)-\frac{v_{i-1}(t)^2}{\hat{B}_{i-1}}\right\}}$$

- $B_i < 0$, deceleration;
- $\hat{B}_{i-1}$, braking rate of vehicle $(i-1)$ estimated by vehicle i;
- θ, safety margin time (talk later);
- L_i, vehicle length

Speed function in congested conditions

If we take $\theta = \frac{\tau}{2}$ as shown in Wikipedia, the function (more recommended) is simplified as

$$v_i^{cong}(t+\tau) =$$

$$B_i\tau + \sqrt{B_i^2\tau^2 + B_i\left\{2[x_{i-1}(t) - x_i(t) - L_{i-1}] - \tau v_i(t) - \frac{v_{i-1}(t)^2}{\hat{B}_{i-1}}\right\}}$$

Speed function in free-flow conditions

The speed in free-flow conditions is fitted from empirical data as

$$v_i^{\text{free}}(t+\tau) = v_i(t) + 2.5A_i\tau\left(1 - \frac{v_i(t)}{V_i^{\max}}\right)\left(0.025 + \frac{v_i(t)}{V_i^{\max}}\right)^{\frac{1}{2}}.$$

- $V_i^{\max}$, maximum speed;
- A_i is the acceleration.

Outline

1. Introduction
2. Model structure
3. Model detail
   - Speed function in congested conditions
   - Speed function in free-flow conditions
4. How is the model derived?
5. Parameter settings

Part-1: Leading Vehicle

- If vehicle $(i-1)$ brakes as hard as desirable at time t, the vehicle will move to position x^*_{i-1} and the speed drops to zero as well, i.e.,

$$\boxed{x^*_{i-1} = x_{i-1}(t) + \frac{v_{i-1}(t)^2}{2B_{i-1}}} \tag{1}$$

- Refer to the well-known uniform acceleration, the Equations of Motion are written as follows,

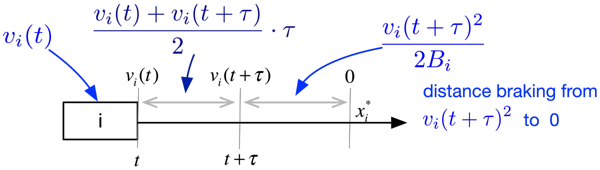
$$\begin{cases} v_t = v_0 + at \\ s = v_0 t + \frac{1}{2}at^2 \end{cases}$$

- Given v_0 (i.e., $v_{i-1}(t)$), a (i.e., B_{i-1}), calculate s

Part-2: Following Vehicle

- Vehicle i will not react until time $(t+\tau)$, and will stop before reaching x_i^*:

$$\boxed{x_i^* = x_i(t) + \frac{v_i(t)+v_i(t+\tau)}{2}\cdot\tau + \frac{v_i(t+\tau)^2}{2B_i}} \qquad (2)$$



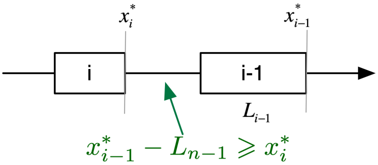
- The term, $\frac{v_i(t)+v_i(t+\tau)}{2}\cdot\tau$, is the distance that the vehicle moves within the reaction time τ.
- It assumes uniform acceleration, also second-order Runge-Kutta scheme.

Part-3: Safety

- For safety reasons, vehicle i must ensure that

$$\boxed{x^*_{i-1} - L_{n-1} \geqslant x^*_i} \tag{3}$$

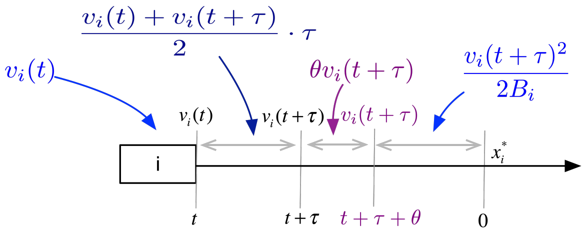
i.e., the rear bumper of the leader is in front of the front bumper of the follower.



Part-4: Human Error

- To allow the driver make mistakes, an additional delay θ is added. Thus, we have

$$\boxed{x_{i-1}^{*} - L_{n-1} \geqslant x_i^{*} + \theta v_i(t+\tau)} \tag{4}$$



Substituting Equation 1 and 2 into Inequality 4.

$$x^*_{i-1} = x_{i-1}(t) + \frac{v_{i-1}(t)^2}{2B_{i-1}} \qquad (1)$$

$$x^*_{i-1} - L_{i-1} \geqslant x^*_i + \theta v_i(t+\tau) \qquad (4)$$

$$x^*_i = v_i(t) + \frac{v_i(t) + v_i(t+\tau)}{2} \cdot \tau + \frac{v_i(t+\tau)^2}{2B_i} \qquad (2)$$

$$x_{i-1}(t) + \frac{v_{i-1}(t)^2}{2B_{i-1}} - L_{i-1} \geqslant v_i(t) + \frac{v_i(t) + v_i(t+\tau)}{2} \cdot \tau + \frac{v_i(t+\tau)^2}{2B_i} + \theta v_i(t+\tau)$$

In real traffic, it is possible for the driver of vehicle i to estimate all the values in Equation 1 and 2 except B_{i-1} (i.e., deceleration of the leading vehicle) by direct observation.

Thus B_{i-1} should be replaced by an estimate $\hat{B}_{i-1}$, and we finally obtain:

$$v_i(t+\tau) \le B_i\left(\frac{\tau}{2}+\theta\right) + \sqrt{B_i^2\left(\frac{\tau}{2}+\theta\right)^2 + B_i\left\{2[x_{i-1}(t) - x_i(t) - L_{i-1}] - \tau v_i(t) - \frac{v_{i-1}(t)^2}{\hat{B}_{i-1}}\right\}}$$

Outline

1. Introduction
2. Model structure
3. Model detail
   - Speed function in congested conditions
   - Speed function in free-flow conditions
4. How is the model derived?
5. Parameter settings

Parameter settings

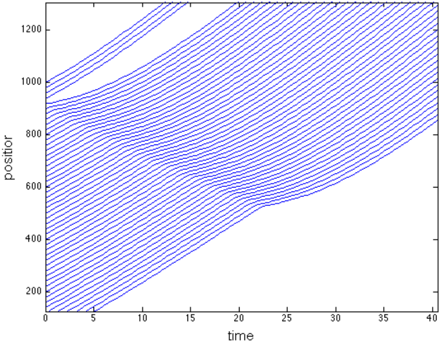
Wilson (2001) conducted simulation experiments on a ring road:

- In the experiment of "stable uniform flow"
  - uniformly distributed initial positions;
  - total vehicle number $N = 50$;
  - $A_i = 1.7\ m/s^2$;
  - $V_i^{\max} = 30\ m/s$;
  - $B_i = -3\ m/s^2$;
  - $\hat{B}_{i-1} = -3.5\ m/s^2$;
  - $\tau = 2/3$;
  - $\theta = 1/3$;
  - $L_i = 6.5\ m$.
- In the experiment of "unstable uniform flow, leading to a travelling wave",
  - $\hat{B}_{i-1}$ is reduced to be $2.8\ m/s^2$

Following works

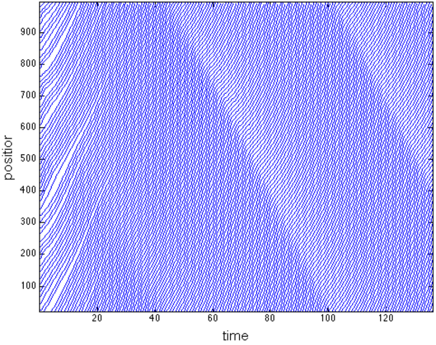
- **Prove that it is right**: the performance of the model is consistent with reality in the microscopic and macroscopic perspectives, such as
  - (Simulated or analytically deducted) fundamental diagram
  - Time-space of trajectories
  - Stability analysis
- Thus, proposing a CF model usually needs a number of work.

Following works



Gipps model + straight road

Following works



Gipps model + ring road

Matlab codes

ing/OneDrive/Simulation/matlab/CF Model/Generic/A_1_Simulation.m*

```matlab
clear;
clc;


%% Setting
global lengthOfRoad numberOfVehicle
lengthOfRoad=10000;
numberOfVehicle=400;

global lengthOfTimeStep
numberOfTimeStep=1000;
lengthOfTimeStep=0.5;

global percentOfInitialDifference
percentOfInitialDifference= [0 0.3];

global speedLimit
speedLimit= ConvertKMH2MS(100);


%% Intialization
seed = RandStream('mt19937ar','Seed',111);
RandStream.setGlobalStream(seed);

global AllPosition AllSpeed
AllPosition= NaN(numberOfVehicle, numberOfTimeStep);
AllSpeed= NaN(numberOfVehicle, numberOfTimeStep);


%%
for time_i=1:1:numberOfTimeStep

    disp(time_i);

    %% load vehicles
    if time_i==1

        AllPosition(:,1)= SetInitialPosition(AllPosition(:,1));
        AllSpeed(:,1)= randi([0,round(speedLimit)],numberOfVehicle,1);
%         AllSpeed(:,1)= speedLimit;

    %% Simulation run
    else

        for veh_i=1:1:numberOfVehicle

            newSpeed= UpdateSpeed(veh_i, time_i);
            newPosition= UpdatePosition(veh_i, time_i, newSpeed);

            AllPosition(veh_i, time_i)= newPosition;
            AllSpeed(veh_i, time_i)= newSpeed;

        end

    end
end


%%
save('SimulationData.mat');
```

ing/OneDrive/Simulation/matlab/CF Model/Generic/A_2_Plot.m

```matlab
clc;
clear;


load('SimulationData.mat');

figure();

for veh_i=1:numberOfVehicle

    indexStart=1;

    loop=0;
    for time_i=1:1:numberOfTimeStep-1

        thisPosition= AllPosition(veh_i,time_i);
        nextPosition= AllPosition(veh_i,time_i+1);

        if thisPosition > nextPosition

            %%%%%%%%%%%%%%
            loop=loop+1;
            color= GetColor(loop);

            indexEnd=time_i-1;

            range= indexStart : 1 : indexEnd;
            plot(range, AllPosition(veh_i, range), color);
            hold on;
            indexStart= time_i+1;

        end

    end


    loop=loop+1;
    color= GetColor(loop);

    range= indexStart: 1 : numberOfTimeStep;
    if size(range,2)>1
        plot(range, AllPosition(veh_i, range), color); hold on;
    end
end


set (gcf,'Position',[0,0,1500,800])
```

Thank you!