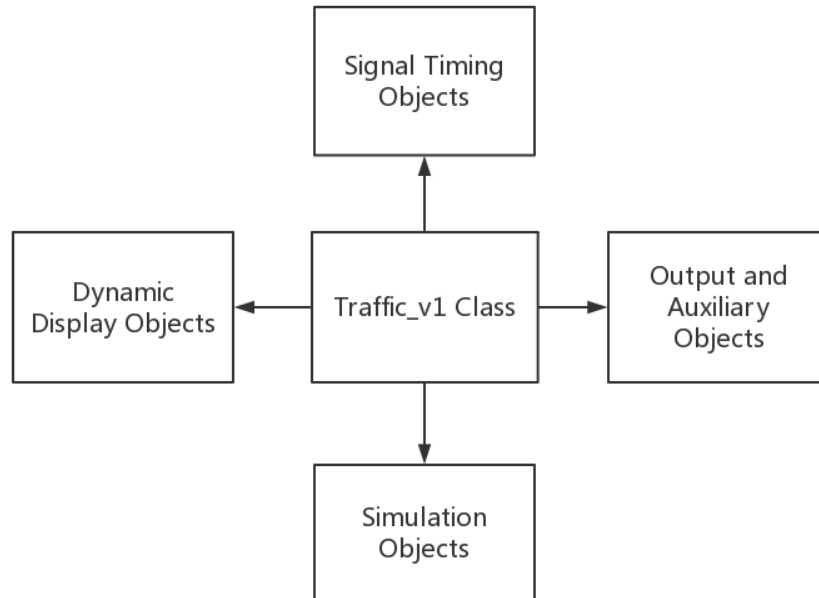# 目录

Fig. 1: Traffic_v1 类的架构

1　程序架构

我们采用C++/Qt来开发现行平台，平台的核心是一个Traffic_v1对象，如图1所示。

1.1　显示模块

这些模块用来动态显示仿真过程和控制仿真执行的流程，如设置仿真速度等。主要包括按键，标签，滑动条等对象。

当仿真每向前执行一步的时候，平台将刷新仿真模块来显示仿真过程中的具体情况，整个函数将完成如下几个步骤。

1. 绘制交通路口的基本要素

   根据信号配时模块给出的信号相位，显示模块将绘制红绿灯。同时，显示模块将绘制交通路口前的车道，目前设定绘制30m长，22.5m宽的车道

2. 绘制车辆

   绘制模块将绘制即将进入路口的车辆和驶离路口的车辆，在所有的车
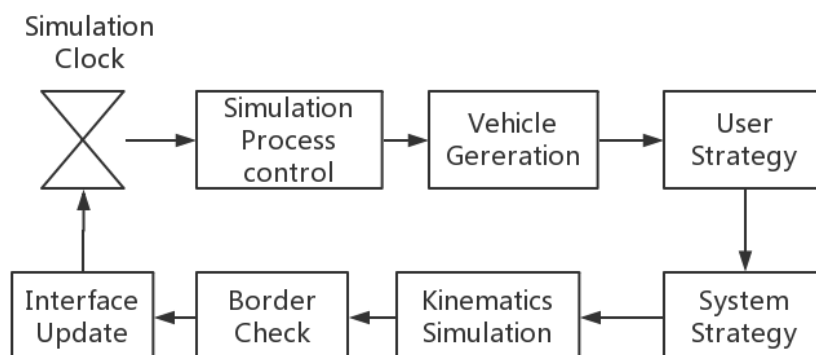
Fig. 2: 平台执行流程

辆中，即将驶入路口的车辆是接受控制的，而驶离路口的车辆是不接受控制的，平台将为他们补全整个的控制策略。

3. 绘制交通路口中的车辆轨迹线

平台不处理车辆在交通路口内的行为，因此采用绘制车辆在路口中的轨迹线来表示车辆可能存在的位置。当车辆驶入交通路口时，我们假设左转车辆在交通路口的滞留时间是3s，而执行车辆在交通路口内的滞留时间是2s，右转车辆在交通路口内的滞留时间是1s，这些参数可以通过平台代码中的参数简单的修改。在滞留时间内的车辆会在交通路口内显示一道轨迹线，表示车辆可能在的位置。

可以看出，轨迹线的交点多少可以表示交通路口的混乱程度，如果交通路口内存在着大量的轨迹线交点，交通效率将受到影响，同时交通危险性将显著提升。而这一部分应当是信号配时的主要工作。

1.2 仿真模块

这是整个平台的核心模块，通过设置的仿真速度，平台给出一个间隔为1ms（高速模式），10ms（中速模式），100ms（低速模式），1000ms（调试模式）的信号。

当触发信号被接受时，平台将按照如图2所示的行为进行相关的操作。

在执行过程中，进程控制模块根据用户设定的停止条件决定是否终止仿真，之后，每个道路的入口将根据车辆生成算法来生成新的车辆。之后，用户策略将根据车辆类型的不同分别和用户选择的模式应用在不同的车辆

上。同时系统策略负责离开交通路口的车辆的行为和诸如进入交通路口等边界情况的处理。

所有的策略均调整车辆的加速度，在运动学仿真模块，平台将通过车辆现有的位置，速度和加速度计算车辆下一时刻的位置和速度。

最后，平台将处理进入交通路口的车辆和即将离开研究路段的车辆的边界情况。

程序之后完成界面的重绘工作，并等待下一个触发信号的到来。

根据现行平台实现，在一般配置的笔记本电脑上，在10ms的仿真时钟间隔的情况（中速模式）下系统压力还是比较合适的，同时平台产生的仿真数据也是比较合理的。当仿真时钟间隔被调整到1ms（高速模式），在交通压力过大的情况下会出现时序问题。

1.3 输出和辅助模块

这部分模块的主要作用是在系统重绘时或一些特殊情况发生时输出仿真结果。

随着仿真的进行，系统在可执行文件所在目录根目录中建立文件夹"result"，在此文件夹内建立包含4张CSV表，一个子文件夹，这个子文件夹的命名规则是

'日期＋时间＋仿真模式选择'

四张CSV表中保存记录了不同类型的数据。

1. car.csv

   This table keeps a record of the initial speed of the vehicle (init_velocity), the thoritical time of arrival (thoritical_time) and the actual time of arriving (act_time), the thoritical time of arrival is defined as the arrival time when ths vehicle accelerates to maximum speed with maximum acceleration, and maintain a uniform linear motion till enters the intersection. The difference between the thoritical_time and the act_time can be described as the time waste driving through this area.

2. stop.csv

   This table records the total number of stops per lane at each time and the total number of stops in the system, and the proportion of the total number of vehicles.

3. road.csv

This table records the total number of vehicles leaving a certain lane at each moment and the total number of vehicles leaving the system, and their average

4. stop_time.csv

This table records the total time of each lane stop at each moment and the total time of parking within the system, as well as the parking time allocated to each vehicle.

Generally speaking, road.csv is used to observe whether the entire simulation process behaves well and the other 3 tables are used to evaluate the merits of the entire strategy.

In addition, the number of stops is determined by whether the speed of the vehicle is below a certain threshold and the stoping time is continuously accumulated as the vehicls stops, besides, the number of stops does not increase during the vehicle's long-term parking.

1.4 Signal timing module

The signal timing module meets the user requirements for setting the traffic lights freely. 3 push buttons 'set red behind'; 'set green behind'; 'set yellow behind' are used to set the phase of signal selected behind the black timeline.

Meanwhile, this interface can be used to set the cycle of the signals. Therefore, by moving the timeline or specify the timeline position, repeatedly selecting the lane to be changed, setting the signal phase. etc, users can set the traffic signals freely.

At the same time, the left interface will prompt the trajectory may exists in the intersection currently (i.e the timeline shows), to prompt the user to avoid the overlap of trajectory, and help users to more reasonable signal timing work.

1.5 The interface of the plantform

The interface of the simulation module and the signal timing module are shown as fig.3 and the fig.4.

The interface of the signal timing module is activated if the user click the 'Edit Traffic Light' Push button. And if the user close the the interface of the signal timing module, the changes made by user will be saved and the interface of the simulation module will be activated.
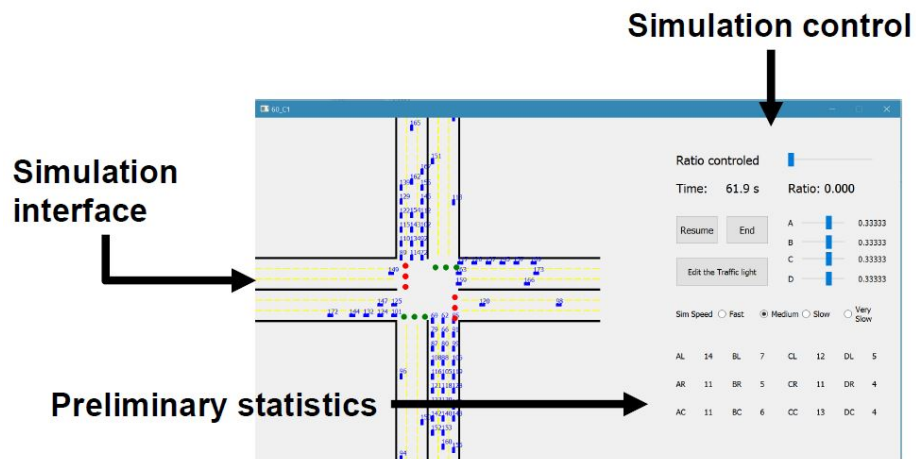
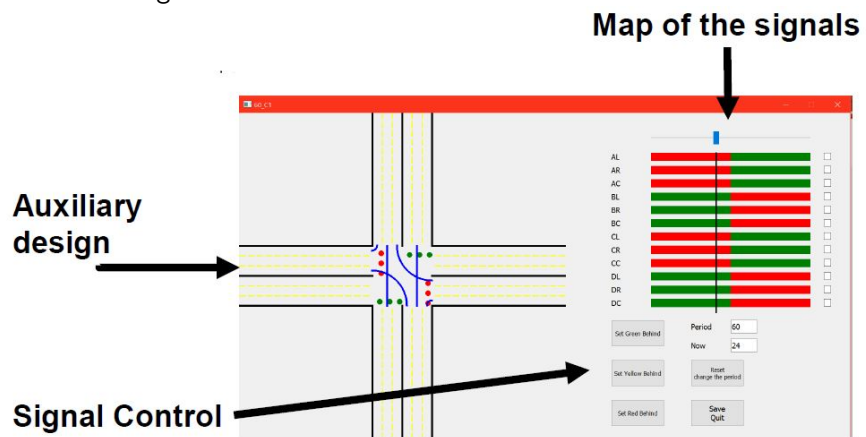Fig. 3: The interface of the simulation module

Fig. 4: The interface of the signal timing module

2   A brief description of the algorithm

2.1   basic assumption and parameters

Firstly, it is assumed that the physical quantity which is under control directly is the acceleration of the vehicle, while other physical quantities are set because of the theorem given by kinematics, such as

$$v(t + \Delta_t) = v(t) + a\Delta_t$$

$$x(t + \Delta_t) = x(t) + v(t)\Delta_t + 0.5a\Delta_t^2$$

However, the acceleration of the vehicle can not be controled precisely, a random noise distributed by normal dirtribution is added into the result given by all of the strategies, especially the ones may operated by human.

As for the speed limit,the platform considers the maximum absolute value of the acceleration of the vehicle is a_{max}, which is generally used in the acceleration process and tbe braking process. Also, the maximum speed limit is set to v_{max}, while the minimum speed limit is set to v_{min}.

No matter which strategy is chosen, if the acceleration given by the strategy is greater than the maximum acceleration a_{max} or less than $-a_{max}$, or the calculated speed is greater than the maximum speed or less than the minimum speed, the program will automatically set these values to the boundary value. When the vehicle has to stop because of the red light, the minimum speed of the vehicle is set to 0, describing the parking behavior which may exist, however, the behavior of reversing is never allowed.

Also, we assume the expected speed of the vehicle, v_{exp}, is set to 80% of the maximum speed limit.

In order to simpliy the simulation process, we assume that during the process of driving, there is no lane changing behavior. This hypothesis is also acceptable in the real life, because of the restrictions imposed by driving habits and traffic laws & regulations, the vehicle should finish changing lane before enter the traffic junction. Therefore this assumption is reasonable. What's more, just from the perspective of traffic efficiency, lane-changing can only reduce the average efficiency of the traffic, therefore, as the primary inspection of the platform is efficiency, lane-changing situation is not taken into consideration.

Therefore, the car driving in the lane can be described as a queue, i.e. Q_{in}

2.2   The Generation of the vehicles

It is assumed that the arrival process of the vehicle is approximately a Poisson process, which is limited and adjusted by the following algorithm

Assuming that the intensity of the Poisson process is λ, the arrival time interval of the vehicle S_n, S_{n-1} can be expressed as

$$P((S_n - S_{n-1} \leq t) = 1 - e^{-\lambda t}$$

Because the simulation time scale is set to 0.1s, the average hourly traffic flow is

$$\bar{F} = 36000\lambda$$

For the traffic flow in one direction, after the average hourly traffic flow is set by the traffic flow adjustment slider described above, The vehicles in this direction will be generated according to the Poisson process with a specified intensity of $\lambda = \dfrac{\bar{F}}{3600}$, and will be pushed into the pending queue $Q_{pending}$ in each direction.

After generating the vehicle, the next step is to consider which lane the vehicle belongs to, and it is assumed that the possibility of the vehicle appearing in each safe lane is same. Firstly, the condition to determine a safe lane is that the distance of last vehicle in the lane and line which the vehicles are generated (S_{start}) is greater than the safe distance S_{safe}, that is,

$$X_{-1} - S_{start} \leq S_{safe}$$

Screening all lanes that meet the situation above in this direction, and place the vehicle in $Q_{pending}$ into one of these lanes randomly, while the position of the vehicle newly generated is set to S_{start}.

If there is no lanes suitable for placing the vehicle currently, the vehicles will remain in $Q_{pending}$ and wait for the next simulation session.

The generation of the vehicle is done in the second step of the simulation cycle, just following the process control model.

2.3   The formation and evacuation of the queue in the intersection

To describe the phenomenon of queuing aat the intersection, a queue Q_{block} is introduced to each lane towards the intersection. And according to the

current signal phase and the length of the queue, all of the control strategies mentioned take these simple algorithm below.

1. Red Light

   If the distance between the first vehicle in Q_{in} and the last vehicle in Q_{block} is less than the control range $S_{control}$, the first vehicle in Q_{in} brake and is expected to stop just behind the last vehicle in Q_{block}. The expected distance between these two vehicles is defined as S_{stop}, describing the distance between the vehicles in the Q_{block}.

   If the Q_{block} is empty and the distance between the first vehicle in Q_{in} and the stopping line is less than $S_{control}$, the first vehicle in Q_{in} brake and is expected to stop just on the stopping lineS_{end}.

   If the first vehicle in the $Q_{in}$is too far that neither of the two above are satisfied, the first vehicle in the Q_{in} is supposed to drive freely.

   Also, the vehicles in the Q_{block} should stop and wait until the light truns green.

2. Green Light distance between the first vehicle in Q_{in} and the last vehicle in Q_{block} (if it exists) should not be less than S_{safe}. And a braking is taken if the distance mentioned above is too short.

Therefore, the brief idea of the algorithm above can be described as the fake code below

```
01 if (Q_block.empty()){
02   if (Light == Green)
03     Q_in.first().drive_freely();
04   else
05     Q_in.first().brake_to(S_end);
06 }
07 else{
08   if(Q_block.last().pos-Q_in.first().pos< S_control)
09     Q_in.first().brake_to
10     (Q_block.last().pos-S_stop);
11   else
12     Q_in.first().drive_freely();
13 }
```

Where the brake_to(desired_pos) method means braking to a desired position.

Meanwhile, the growth and the dissipation of the queue Q_{block} follow the method below.

1. Increase

   If the first vehicle in Q_{in} is closer than S_{stop} from the last vehicle in Q_{block} (it is set to S_{end} if the Q_{block} is empty), it is removed from the Q_{in} and pushed into the Q_{block}

2. Dissipation

   When the traffic light is green, all of the vehicles in the Q_{block} is moving forward at the speed of v_{dis}, since the speed is relatively low, the acceleration process and any other phenomenon can be ignored. When the vehicle moves over the stopping lane and enters the intersection, it is removed from the Q_{block}

And the method above can be described as the fake code below

```
01 if (light == green){
02  Q_block.all_move_forward(v_dis)
03 }
04 else{
05  if(!Q_block.empty()){
06   if(Q_block.last().pos-Q_in.first().pos
07     < S_stop){
08     Vehicle v;
09     v=Q_in.getfirst();
10     Q_block.push(v);
11   }
12  }
13  else{
14   if(S_end-Q_in.first().pos< S_stop){
15     Vehicle v;
16     v=Q_in.getfirst();
17     Q_block.push(v);
18   }
19  }
```

20
21 **}**

All above describe the formation and evacuation of the queue in the intersection and its interaction with other models. The differences caused by the assumption can be ignored. Thanks to this model, it is easy for us to decouple the driving model and the queuing model, which makes it much easier to implement the other algorithm

2.4   Basic models

In the description of manual driving and automatic driving vehicle models, the car-following model and free driving model are frequently used models. Therefore, we will describe these models before discussing the control strategies.

2.4.1   The implementation of the free-driving method

When there is no speed guidance, a vehicle's driving behavior can be roughly classified into free-driving and car-following, which will be described in this section and the next section. $S_{control}$ is defined as the distance of interaction between vehicles. Thus, the two driving models are chosen according to the rules below.

1. When the distance to the front vehicle from the vehicle which we are concerned about is less than $S_{control}$, the driving choose the car-following strategy.

2. When the distance is greater than $S_{control}$, the driver choose the free-driving strategy, which means trying to reach the expected speed, v_{exp}, during this period, a smaller acceleration may be chosen if the current speed of vehicle is almost the same as v_{exp}, while a greater acceleration should be chosen if the current speed is too low, such as the vehicle is stopped.

   Also, because some random noise may be introduced to the system, if the speed of the vehicle is higher than v_{max}, a minor deceleration should be taken.

2.4.2   The implementation of the car-following method

As mentioned above, when the vehicles is close enough to each other, the car-following method is chosen. In the strategy we used, the commonly used driving psycho-physical model—Wiedemann model is adopted, that is,

$$a_n(t + \Delta_t) = \frac{[\Delta v_{n,n-1}(t)]^2}{2[\Delta x_{n,n-1}(t) - S_{exp}]} + a_{n-1}(t)$$

In the above equation, S_{exp} represents the expected minimum safe-following distance. Since the speed of the vehicle in the intersection may have a big range, S_{exp} should not be a constant. Therefore, according to the regulations on safe distance in Regulation on the Implementation of the Road Traffic Safety Law of the People's Republic of China, we use the linear correlation model of the minimum safe following distance and the speed of the front vehicle, therefore, the S_{exp} can be described like this,

$$S_n(t) = \alpha v_{n-1}(t) + S_{safe}$$

While the $S_{safe}$is the same as the distance between the vehicles in 2.3, and the constant α may be set by according to the experience.

Once the distance between the vehicles is less than S_{exp}, which means the distance between the vehicles is too short, an emergency brake is taken owing to the specific situation. In most cases, the acceleration of the vehicle is set to $0.5a_{max}$

2.5   Manual driving model

The plantform provides three models to verify the effectiveness of the platform, and the first model is manual driving model.

As the name of the model indicates, manual driving model is used to describe the behavior of a vehicle driving by human. This model is often used to provide a 'basic' traffic efficiency, while other control strategies are expected to behave better than this simple model.

Firstly, vehicles which are far (greater than S_{inter}) from the intersection are taken into consideration. Because they are so far from the intersection, they do not consider the influence of the traffic light.

Therefore, these vehicles mat take the car-following method or the free-driving method. If a vehicle is the first one in Q_{in}, it has to take the the free-driving method.

As for the ones which are closer to the intersection, different method will be chosen to meet the different situation below.

1. Green traffic light with an empty waiting queue Q_{block}

   In this situation, the behavior of the vehicles is similar to the behavior described above.

   In addition to the behavior mentioned, the remain time of the green light is calculated, if the remaining time is less than T_{safe}, the possibility of entering the intersection during the green light may be taken into consider, if it is difficult for the vehicle to pass the stopping line before the light turns red, the vehicle may behave like the traffic light is red.

2. Green traffic light while the waiting queue Q_{block} is not empty

   During this situation, the vehicles discussed may brake or accelerate to set its speed to v_{dis}, and try to enter the queue Q_{block}, once it enters the Q_{block}, it will move with the other vehicles in the queue.

   Also, if the remaining time is less than T_{safe}, the program may consider slow down the vehicle and prepare to stop.

3. Red traffic light

   In this case, the vehicles discussed will behave just like what we have discussed in 2.3

What's more, we focus on portraying the behavior of the first vehicle in the driving queue Q_{in}, since the vehicles behind it can be controled easily by car-following method.

Taking all of the situations into consideration. the manual driving model can be abstract as the fake code below.

```
01  //the code below just consider the vehicle v
02  if(S_end-v.pos>S_inter)//far from intersection
03  {
04    //code block 1
05    if(car-following_requirement_met())
06      //the car-following requirement is   met
07      car_following();
```

```
08   else
09     //the head car of the queue or
10     //the car-following requirement isn't met
11     free_driving();
12   }
13   else //close to the intersection
14   {
15   if(light==red)
16   method1.3();//the method mentioned in 1.3
17   else
18   if(Q_block.empty()){
19     if(time_remain>T_safe)
20     //same as the code block 1
21     else
22       if(!pass_check())
23       //same as the red light case
24     }
25   else
26     if(time_remain>T_safe)
27     acc_to_speed(v_dis);
28     else
29       if(!pass_check())
30       //same as the red light case
31     }
32
33   }
```

The plantform will traverse the queue Q_{in}, and during each iteration, the acceleration of the vehicle in processing will be set according to the model given above.

2.6   The single-vehicle cooperative speed guidance model

Applied some changes on the car-following strategy, we designed a cooperative speed guidance model which only takes the influence of the traffic light into consideration.
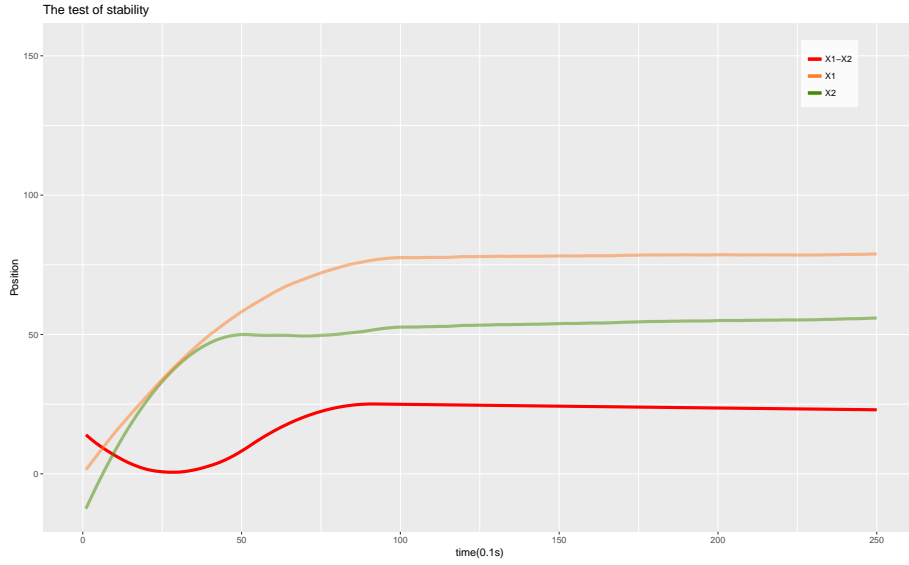
The test of stability

Fig. 5: The position of the vehicles and the distance between them

2.6.1 Modification about the car-following method

Firstly, when the distance to the front vehicle from the vehicle which we are concerned about is less than $S_{control}$, the car-following model in chosen. In addition to the basic car-following method, it is changed to the following form when $\Delta x_{n,n-1}(t) \leq S_{exp}$

$$a_n(t + \Delta_t) = -sgn(\Delta v_{n,n-1}(t))\frac{[\Delta v_{n,n-1}(t)]^2}{2[\Delta x_{n,n-1}(t) - S_{exp}]} + a_{n-1}(t)$$

which means, if the speed of the following vehicle is lower, the strategy will suggest that this vehicle should accelerate.

With this minor modification, some experimental researches show that the changed car-following method works well and can keep a stable distance between the vehicles.

As the fig.5 shows, the vehicle following takes the modified car-following method and the front vehicle decelerate with a large random noise. From the figure it can be found that the distance between these two vehicles will be stable near S_{exp}, no matter what initial situation is chosen.

2.6.2 The cooperative vehicle strategies

Besides the chaning on car-following method, the free-driving model is also replaced by the following cooperative vehicle strategy.

First of all, we define two parameters, the maximum time cost T_{max} is defined as the time cost if the vehicle firstly decelerates at the acceleration of $-a_{max}$ and maintain a uniform linear motion at the speed of v_{min} until it reaches the intersection. Also, the minimum time cost T_{min} is defined as the time cost if the vehicle firstly accelerates at the acceleration of a_{max} and maintain a uniform linear motion at the speed of v_{max} until it reaches the intersection. It is obvious that the actual time cost of a vehicle is less than T_{max} and greater than T_{min}, and both the T_{max} and the T_{min} depengs only on the initial speed v_{init} of the vehicle starting from the start line.

Then, the strategy determines the earliest green time T_G in the interval $(T_{min}, T_{max}]$ with the accuracy in second. If there is no green time during this interval, a free-driving strategy is applied to the vehicle.

If a proper T_G could be set, an acceleration or a deceleration may be applied to the vehicle to adjust its motion at the aim of reaching the stopping line at the time of T_G. During this period, if the vehicle is affected by the preceding vehicle, the car-following model may be applied to the vehicle again.

Meanwhile, the behavior the vehicle takes is similar to the one in the manual driving method.

Generally speaking, this strategy is the optimize version of the car-following model with the consideration of the influence of the traffic light.

2.7 The multi-vehicle cooperative speed guidance model

This is a simple connect-vehicle strategy with the communication between the vehicles.

The strategy is operated from the first vehicle in Q_{in} to the last vehicle in Q_{in}

First of all, the first vehicle in Q_{in} determine its arrival time T_G according to the method mentioned in section 2.6.2, and we set the T_G determined by the n-th vehicle in Q_{in} is $T_G^{[n]}$, e.g. the first vehicle determines $T_G^{[1]}$

After chosing the $T_G^{[1]}$, an acceleration or a deceleration may be applied to the vehicle to adjust its motion at the aim of reaching the stopping line at the time of $T_G^{[1]}$, if the $T_G[1]$ does not exist, the vehicle takes an acceleration and then stops at the stopping line.

Then, it is assumed that the [n+1]-th vehicle in Q_{in} can acquire the $T_G^{[n]}$ of the n-th vehicle in Q_{in}, including whether it exists.

The [n+1]-th vehicle determins its $T_G^{[n+1]}$according to its T_{min} and T_{max} and make sure that the $T_G^{[n+1]} - T_G^{[n]} \geq 1$second, owing to the fact that the following vehicle is supposed to arrive later than the preceding vehicle.

If the $T_G^{[n]}$ does not exist, $T_G^{[n]+1}$ should be set freely, only depends on the T_{max} and the T_{min} of the vehicle.

Then the acceleration of the [n+1]-th vehicle is set according to the expected time of arrival. The strategy then goes for the following vehicles.

Since the expected time of arrival is carefully designed to make sure that the following vehicle always arrives the stopping line later than the preceding vehicle, even the vehicles which do not have an expected arrival time will take an acceleration, the probability of the rear-ending is minor, a rear-ending will be solve by some simple emergency settings.

2.8 The processing of the strategies and the combination of the strategies

2.8.1 Single strategy simulation

The plantform treats the strategies as an iteration from the last vehicle in the Q_{in} to the first one in that, for the reason that the following vehicles may acquire the acceleration of the preceding car which should be fixed until the acceleration is set according to the strategy chosen by user.

Therefore, if only a specific strategy is chosen, the program behaves like a loop from the last to the first of Q_{in}.

2.8.2 Combination of the strategies

Only the combination of the manual driving vehicle and the speed guidance models are concerned about, therefore, the plantform only include two kinds of combination simulation, which is the combination between the manual driving vehicle(mentioned in 2.5) and the single-vehicle cooperative speed guidance model (mentioned in 2.6); and the combination between the manual driving vehicle and the multi-vehicle cooperative speed guidence model (mentioned in 2.7)

During the vehicle generating step, the control strategy of the vehicle is set by random method according to the type of combination and the ratio set by user.

After generate the type of the control strategy, the user-strategy step of the simulation set the acceleration of the vehicle using the strategy it has. It is assmued that if a vehicle using multi-vehicle cooperative speed guidance model is following a manual driving vehicle, the expected arrival time of the manual driving vehicle T_G does not exist, therefore, the speed guidance model could set its expected arrival time T_G freely.

2.9   Add on algorithms

Besides this, some add on algorithm is added to treat some accidents may happen accidentally. And the implementations of these specific algorithms may be found in the code of the plantform.