2.Vehicle Generations in Excel

2020112921 刘欣豪

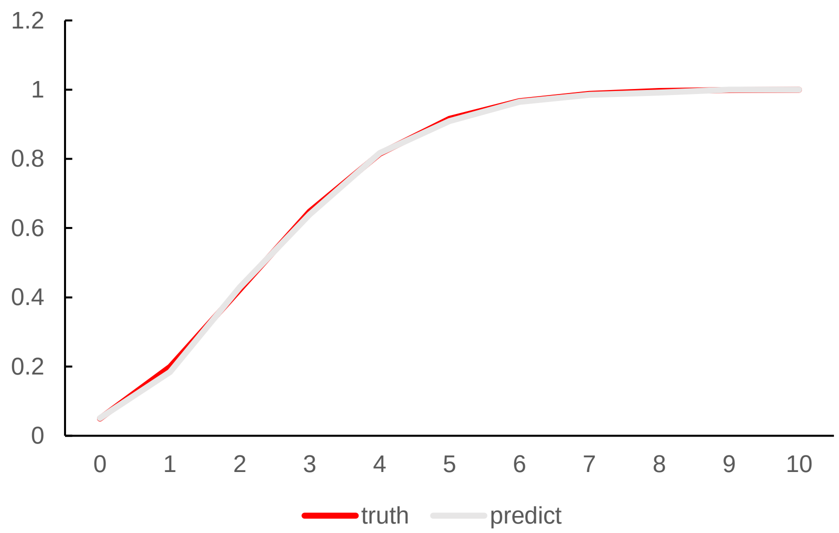
- POISSON

Define the mean = 3,
then the probability-distribution (Keep three decimals) is

0	1	2	3	4	5	6	7	8	9	10
0.050	0.199	0.423	0.647	0.815	0.916	0.966	0.988	0.996	0.999	1.000

I generate 1000 random numbers between 0 and 1, and choose the number what they belong to. Now results are as follows:

num	2	3	4	1	5	6	0	7	9	8
frequency	0.245	0.210	0.178	0.133	0.092	0.056	0.051	0.020	0.008	0.007



High fit

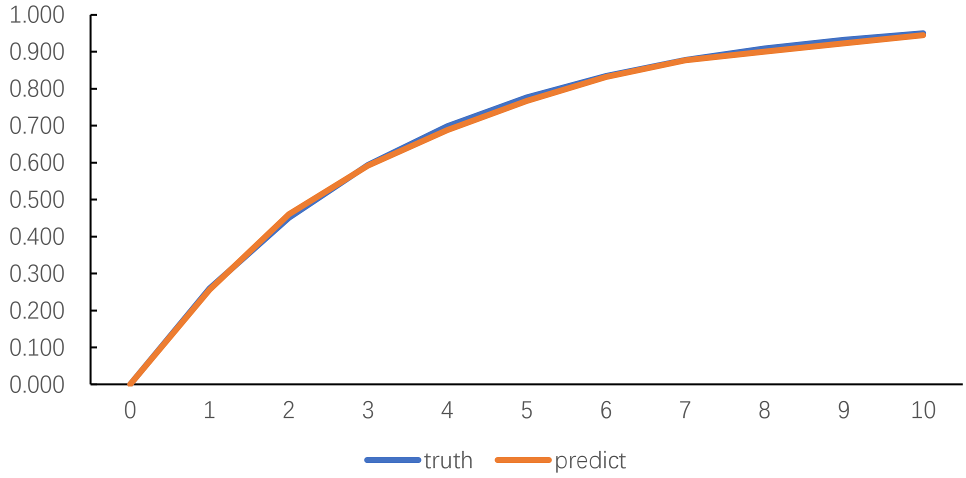
- NED

Define the μ = 0.3
then the probability-distribution (Keep three decimals) is

0	1	2	3	4	5	6	7	8	9	10
0.000	0.259	0.451	0.593	0.699	0.777	0.835	0.878	0.909	0.933	0.950

I generate 1000 random numbers between 0 and 1, and choose the number what they belong to. Now results are as follows:

num	0	1	2	3	4	5	6	7	8	9
frequency	0	256	204	132	96	79	65	45	23	23



3.number of overtakes between bicycles and mopends

2020112921 刘欣豪

Try to derive and compute the expected number of overtakes between bicyles and mopends, i.e., $E(OT_{bic-mop})$.

对于 bicycle 与 mopeds，mopeds 的速度为 v_i，bicycle 的速度为 v_j

$$E(OT_{bic-mop}) = XT\int_{v_i=a}^{b}\int_{v_j=v_i}^{c} k_1 k_2 p_M(v_i) p_B(v_j)(v_j - v_i)dv_j dv_i$$

$$E(OT_{bic-mop}) = XTk_1 k_2\int_{v_i=a}^{b}\int_{v_j=v_i}^{c} p_M(v_i) p_B(v_j)(v_j - v_i)dv_j dv_i$$

5.Pips-forbs model simulation

2020112921 刘欣豪

Example – **Assignment**: Replicate this (by making up data of the lead vehicle).

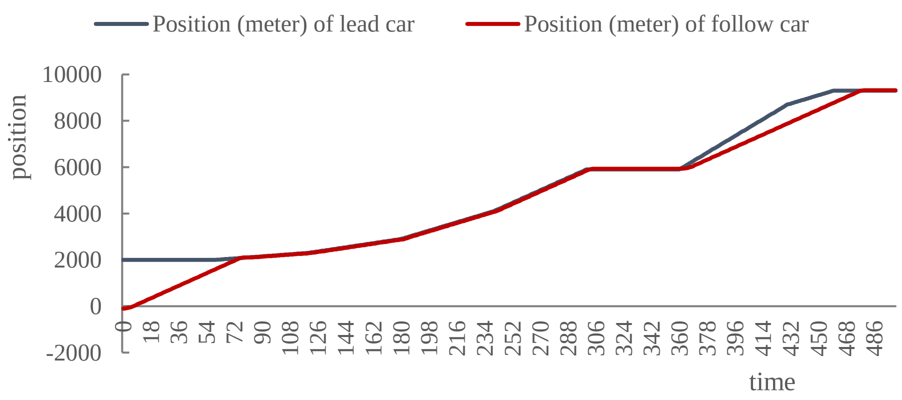
The spreadsheet shared with you has positions of the lead vehicle over time. Use the pseudocode to simulate the positions of the follower vehicle.

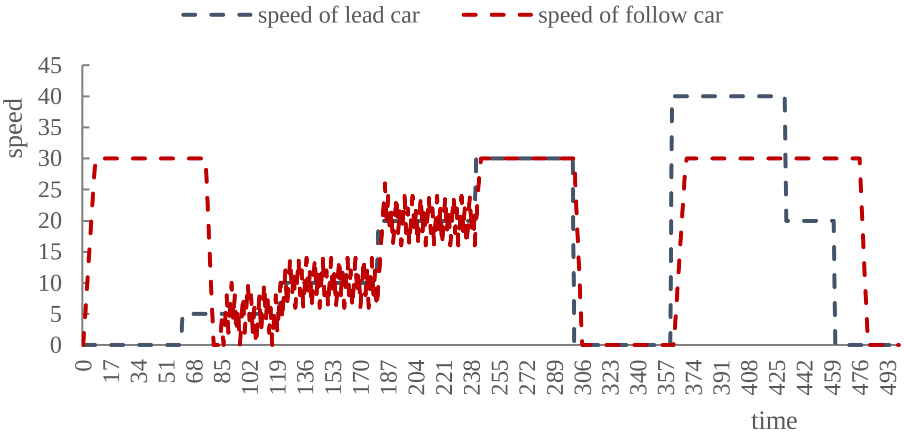
Assume that the follower vehicle starts at $x = -102$, has a maximum acceleration and deceleration of 4 m/s^2 and -6 m/s^2. Suppose that the desired speed is 30 m/s.

输出 output 表（部分）：

Time (second)	Position (meter) of lead car	speed of lead car	Position (meter) of follow car	speed of follow car	acceleration
0	2000	0	-102	0	4
1	2000	0	-98	4	4
2	2000	0	-90	8	4
3	2000	0	-78	12	4
4	2000	0	-62	16	4
5	2000	0	-42	20	4
6	2000	0	-18	24	4
7	2000	0	10	28	4
8	2000	0	40	30	4
9	2000	0	70	30	4
10	2000	0	100	30	4

仿真结果：





出现撞车现象且在渐变区域不稳定。

代码：

```
import pandas as pd

datas = pd.read_excel('Lead+car+data.xlsx')

for i in range(0,500):
    s = datas.iloc[i,1]-datas.iloc[i,3]
    v = datas.iloc[i,4]
    smin = 6 * (v/4.47 + 1)
#      print(smin)
    if s<smin:
        datas.iloc[i+1,4] = max(0,v-6)
        datas.iloc[i,5] = -6
    else:
        datas.iloc[i+1,4] = min(30,v+4)
        datas.iloc[i,5] = 4
    datas.iloc[i+1,3] = datas.iloc[i,3]+datas.iloc[i+1,4]

datas.to_excel('output.xlsx',index  =False)
```

6.Pips model 均衡分析（基本图关系推导）

2020112921 刘欣豪

$$s=\frac{v}{4.47}l+l$$

已知：$k_j=\frac{L}{l},k=\frac{L}{s}$

即：$\frac{k_j}{k}=\frac{v}{4.47}+1$

整理得：

$$v=4.47(\frac{k_j}{k}-1)$$

7.Pips-forbs model simulation

2020112921 刘欣豪

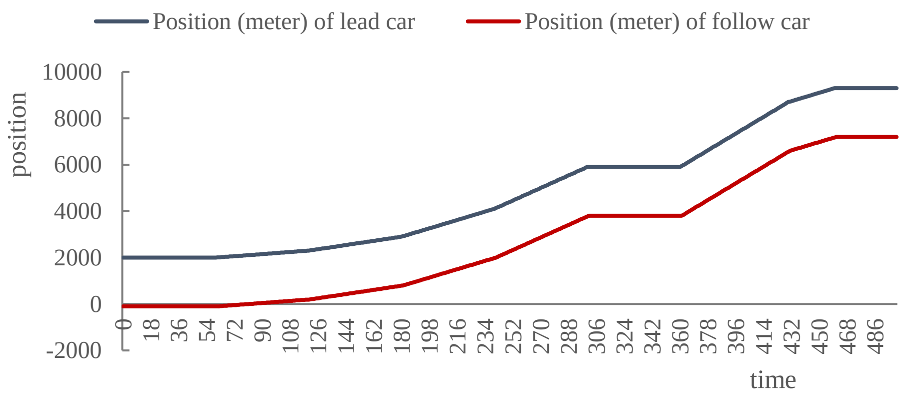
Exercise

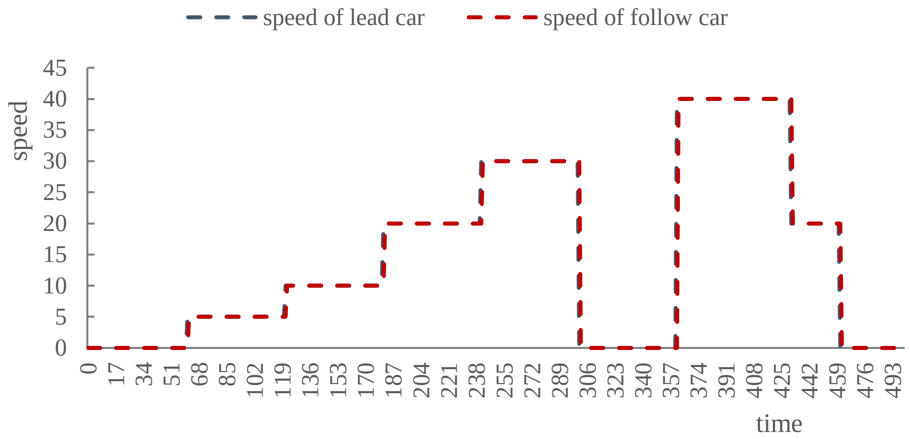
Execute simulation of the model by Chandler, Hermann, and Montroll

输出 output 表（部分）：

Time (second)	Position (meter) of lead car	speed of lead car	Position (meter) of follow car	speed of follow car	acceleration
0	2000	0	-102	0	0
1	2000	0	-102	0	0
2	2000	0	-102	0	0
3	2000	0	-102	0	0
4	2000	0	-102	0	0
5	2000	0	-102	0	0
6	2000	0	-102	0	0
7	2000	0	-102	0	0
8	2000	0	-102	0	0
9	2000	0	-102	0	0
10	2000	0	-102	0	0

仿真结果：





轨迹高度重叠，不符合现实

代码：

```python
import pandas as pd
datas = pd.read_excel('Lead+car+data.xlsx')

for i in range(0,500):

    datas.iloc[i,5] = datas.iloc[i,2] - datas.iloc[i,4]
    datas.iloc[i+1,4] = datas.iloc[i,4]+datas.iloc[i,5]
    datas.iloc[i+1,3] = datas.iloc[i,3]+datas.iloc[i+1,4]

datas.to_excel('output.xlsx',index  =False)
```

8.GM model simulation

2020112921 刘欣豪

Use the spreadsheet to simulate the GM model

$\tau = 1s, \alpha = 0.8, m = 1, l = 1$

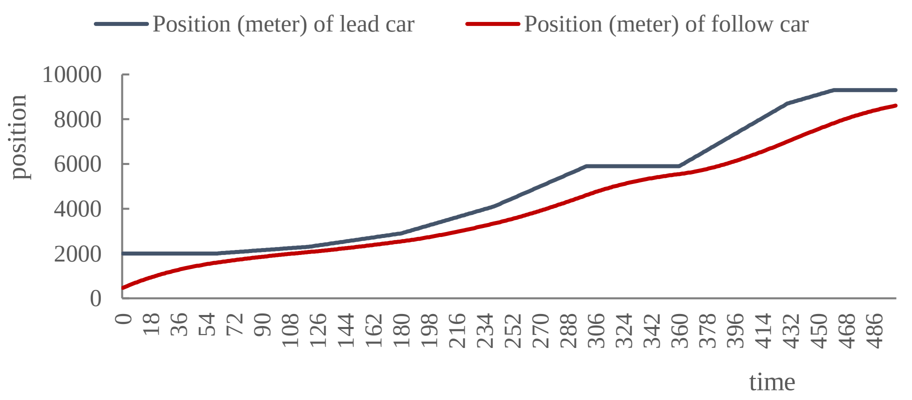
starting x = $467m$

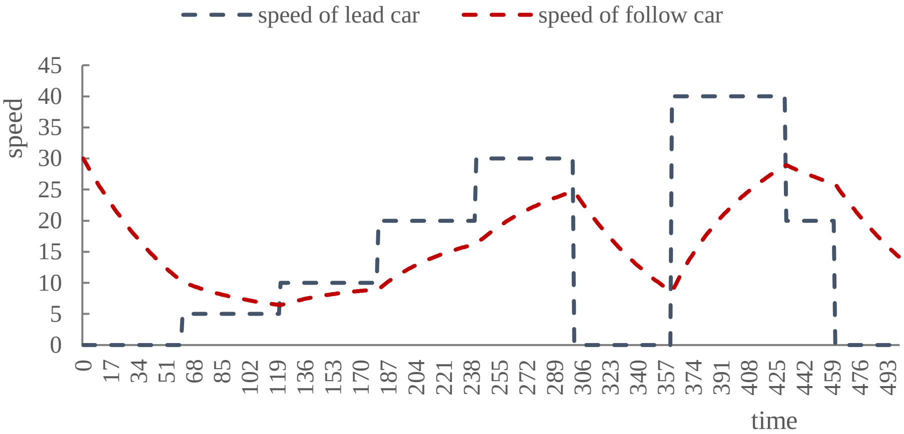
the speed of follow car is $30m/s^2$

输出 output 表（部分）：

Time (second)	Position (meter) of lead car	speed of lead car	Position (meter) of follow car	speed of follow car	acceleration
0	2000	0	467	30	0
1	2000	0	496.5303	29.53033	-0.46967
2	2000	0	525.5967	29.06632	-0.46401
3	2000	0	554.2046	28.60791	-0.45841
4	2000	0	582.3596	28.15506	-0.45285
5	2000	0	610.0673	27.70772	-0.44734
6	2000	0	637.3332	27.26584	-0.44187
7	2000	0	664.1626	26.82939	-0.43645
8	2000	0	690.5609	26.39831	-0.43108
9	2000	0	716.5334	25.97256	-0.42575
10	2000	0	742.0855	25.55209	-0.42047

仿真结果：





效果较好

代码：

```
import pandas as pd
datas = pd.read_excel('Lead+car+data.xlsx')
for i in range(0,500):
    dv = datas.iloc[i,2]-datas.iloc[i,4]
    dx = datas.iloc[i,1]-datas.iloc[i,3]
    datas.iloc[i+1,5] = 0.8*datas.iloc[i,4]*dv/dx
#     datas.iloc[i+1,5] = 0.8*dv/dx
    datas.iloc[i+1,4] = max(0,datas.iloc[i,4]+datas.iloc[i+1,5])
    datas.iloc[i+1,3] = datas.iloc[i,3]+datas.iloc[i+1,4]
datas.to_excel('output.xlsx',index  =False)
```

9.读书笔记

2020112921 刘欣豪

交通流反应刺激模型是一种用于描述人类驾驶员在面对不同交通情形时产生的行为反应的模型，能够为交通规划和设计领域提供了重要参考。该模型将驾驶员对交通刺激的反应划分为三个主要阶段，包括感知、决策和执行。以下是我对该模型的阅读笔记（主要为不同模型的参数标定）。

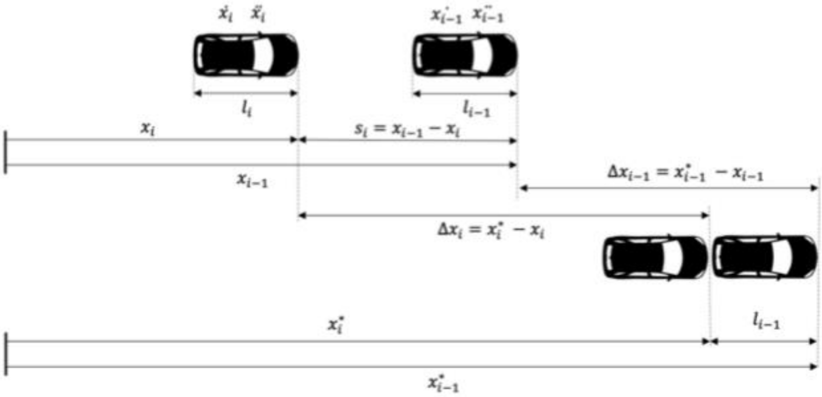
参数标定是一个重要的步骤，它的目的在于根据实际数据对模型参数进行确定，以使模型能够更好地描述实际场景中的驾驶员行为。不同的参数标定可能会对模型的输出结果产生显著影响。因为调整参数标定可以影响到模型的行为反应特征和预测准确性。一方面，如果模型参数被过度调整，可能会导致模型在描述驾驶员行为时失去真实性和准确性，从而影响到模型的预测结果。另一方面，如果模型参数被过度简化，那么模型也可能会无法准确描述驾驶员的行为。因此需要通过不断地调整和优化参数标定，最终找到最优模型参数以匹配不同的交通情况。

表 14-2 早期的单域模型

作者	模型	参数
Greenshields [1]	$v = v_f(1 - \frac{k}{k_j})$	v_f, k_j
Greenberg [2]	$v = v_m \ln\left(\frac{k}{k_j}\right)$	v_m, k_j
Underwood [3]	$v = v_f e^{-\frac{k}{k_m}}$	v_f, k_m
Drake [4]	$v = v_f e^{-\frac{1}{2}\left(\frac{k}{k_m}\right)^2}$	v_f, k_m
Drew [5]	$v = v_f\left[1 - \left(\frac{k}{k_j}\right)^{n+\frac{1}{2}}\right]$	v_f, k_j, n
Pipes-Munjal [6] [7]	$v = v_f\left[1 - \left(\frac{k}{k_j}\right)^{n}\right]$	v_f, k_j, n

10.Gipps 推导

2020112921 刘欣豪



有如下关系：

$$x_{i-1}^* - x_i \ge x_i^* - x_i + l_{i-1}$$

整理后可得

$$(x_{i-1}^* - x_{i-1}) + (x_{i-1} - x_i) \ge x_i^* - x_i + l_{i-1}$$

即：

$$\frac{{v_{i-1}}^2(t)}{2a_{i-1}} + s_i \ge \frac{v_i(t) + v_i(t+\tau)}{2}\tau_i + \frac{v_i^2(t+\tau)}{2a_i} + l_{i-1}$$

$$\frac{v_i^2(t+\tau)}{2a_i} + \frac{v_i(t+\tau)}{2}\tau_i - \left(\frac{{v_{i-1}}^2(t)}{2a_{i-1}} + s_i - l_{i-1} - \frac{v_i(t)\tau_i}{2}\right) \le 0$$

解得

$$v_i(t+\tau_i) \le -a_i\tau_i + \sqrt{{a_i}^2{\tau_i}^2 + a_i\left(-v_i(t)\tau_i + \frac{v_{i-1}(t)^2}{2a_{i-1}} - 2l_{i-1} + 2s_i\right)}$$

11.Find the capacity of LCM

2020112921 刘欣豪

已知：

$$k=\frac{1}{s}=\frac{1}{(\gamma v^2+\tau v+l)[1-\ln(1-\frac{v}{v_f})]}$$

则：

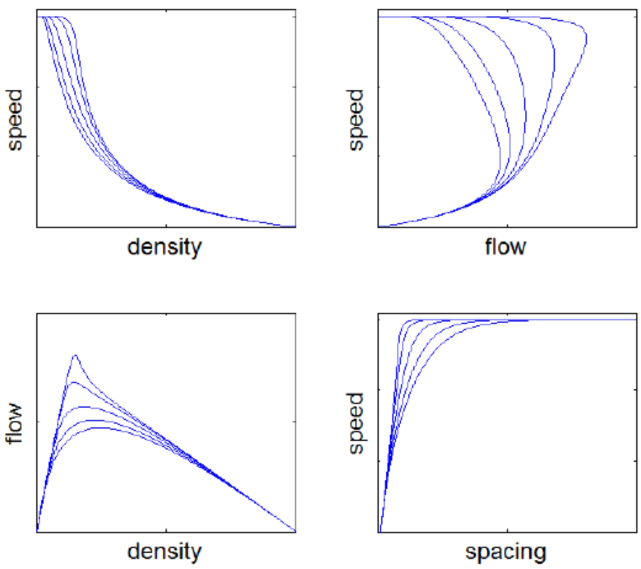
$$q=kv=\frac{v}{(\gamma v^2+\tau v+l)\left[1-\ln\left(1-\frac{v}{v_f}\right)\right]}$$

$$\frac{dq}{dk}=v+k\frac{dv}{dk}=v+k\frac{1}{\frac{dk}{dv}}$$

$$\frac{dq}{dk}=v-\frac{(\gamma v^2+\tau v+l)\left[1-\ln\left(1-\frac{v}{v_f}\right)\right]}{(2\gamma v+\tau)\left[1-\ln\left(1-\frac{v}{v_f}\right)\right]+\frac{(\gamma v^2+\tau v+l)}{v_f-v}}$$

确定 capacity 时，可先令流量对密度的一阶导为 0，求得最佳速度或最佳密度，最后回代球的流量的最大值。

根据课件内容：



12.CF=KW

2020112921 刘欣豪

CF：

当 $s_i^n \le \delta + \tau v_f$ 时：$x_{i+1}^n = x_i^{n-1} - \delta$

已知 $s_i = x_i^{n-1} - x_i^n$

$s_i = x_{i+1}^n - x_i^n + \delta = v_i^n \tau + \delta$

则 $\frac{1}{k} = \frac{q}{k}\tau + \delta$；即：$q = \frac{1}{\tau} - \frac{\delta k}{\tau}$

KW：同理如上

对于另半边：

CF：

$x_{i+1}^n = x_i^n + \tau v_f$

$v = v_f$

KW：

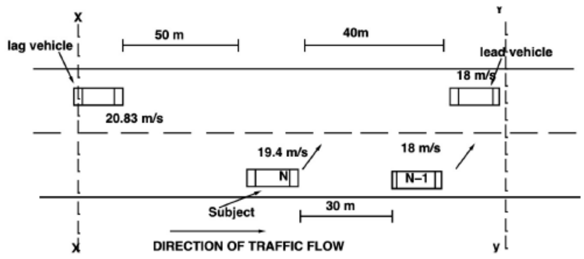
$x_{i+1}^n = x_0^n + i\tau v_f + \tau v_f$

$v = v_f$

13.Changing lanes

2020112921 刘欣豪

For the scenario shown in the following figure, find the probability of changing lanes



Assume that

- $\sigma^{lead} = \sigma^{lag} = 2$
- $\beta^{lead} = \beta^{lag} = 1$
- $X_i^{lead,TL}(t) = X_i^{lag,TL}(t) = 0.8$
- $\nu_i = 0.7$ and $\alpha^{lead} = \alpha^{lag} = 1.2$

代入：

$$\mathbb{P}\left[g_i^{x,lead,TL}(t) > g_i^{x,cr,lead,TL}(t) | TL, \nu_i\right] \mathbb{P}\left[g_i^{x,lag,TL}(t) > g_i^{x,cr,lag,TL}(t) | TL, \nu_i\right]$$
$$= \Phi\left(\frac{\ln(g_i^{x,lead,TL}(t)) - \beta^{lead} X_i^{lead,TL}(t) - \alpha^{lead}\nu_i}{\sigma^{lead}}\right)$$
$$\Phi\left(\frac{\ln(g_i^{x,lag,TL}(t)) - \beta^{lag} X_i^{lag,TL}(t) - \alpha^{lag}\nu_i}{\sigma^{lag}}\right)$$

$$\phi(\frac{\ln(50)-0.8-1.2*0.7}{2})\phi(\frac{\ln(40)-0.8-1.2*0.7}{2})$$

$$\phi(1.14)\phi(1.02)$$

经查表 0.8729*0.8461=0.7386=73.86%

15.Cell Transmission Model

2020112921 刘欣豪

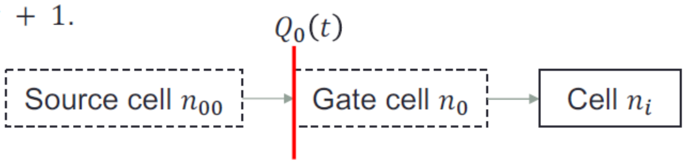
Consider a link with three cells. Suppose that time is divided into 1 second intervals.

- At most 10 vehicles can move from one cell to the next cell in one time step, i.e., $q_{max}\Delta t = 10$.
- The maximum number of vehicles that can fit a cell $N_i = 30$.
- $w/v_f = 2/3$
- The demand for vehicles trying to enter the link is known $d(t)$.

Now also assume that at the downstream end, a traffic light is red from $t = 0$ to $t = 9$ and will turn green forever at $t = 10$. Use the spreadsheet provided to calculate cell occupancies over time.

设置 demand by a cell pair，

A *source* cell numbered *00* with an infinite number of vehicles ($n_{00}(0) = \infty$) that discharges into an empty *gate* cell *0* of infinite size, $N_0(t) = \infty$. The inflow capacity $Q_0(t)$ of the gate cell is set equal to the desired link input flow for next time interval $t + 1$.



则如图 1：

time	0	1	2	3	4	5	6	7	8	9	10
n00	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
n0	0	10	10	10	10	10	10	10	10	10	14. 4
n1	0	0	10	10	10	10	10	10	14. 1	21. 6	26. 2
n2	0	0	0	10	10	10	13. 3	21. 1	26. 3	28. 5	29. 4
n3	0	0	0	0	10	20	26. 7	28. 9	29. 6	29. 9	20
nout	0	0	0	0	0	0	0	0	0	0	10

time	11	12	13	14	15	16	17	18	19	20	21	22
n00	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
n0	21. 9	26. 2	25. 3	22. 3	19. 5	17. 5	16. 3	15. 6	15. 3	15. 1	15. 1	15
n1	28. 4	24. 9	20. 8	18	16. 5	15. 7	15. 3	15. 1	15. 1	15	15	15
n2	23. 1	18. 8	16. 6	15. 7	15. 3	15. 1	15	15	15	15	15	15
n3	16. 7	15. 6	15. 2	15. 1	15	15	15	15	15	15	15	15
nout	20	30	40	50	60	70	80	90	100	110	120	130

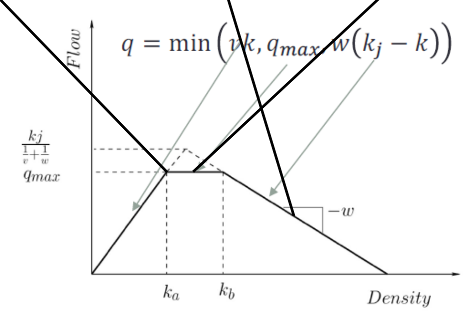


图 1：0-22s 仿真结果

附件 1：

40s 仿真完整表格

time	n00	n0	n1	n2	n3	nout
0	9999	0	0	0	0	0
1	9999	10	0	0	0	0
2	9999	10	10	0	0	0
3	9999	10	10	10	0	0
4	9999	10	10	10	10	0
5	9999	10	10	10	20	0
6	9999	10	10	13.33333	26.66667	0
7	9999	10	10	21.11111	28.88889	0
8	9999	10	14.07407	26.2963	29.62963	0
9	9999	10	21.60494	28.51852	29.87654	0
10	9999	14.40329	26.21399	29.42387	19.95885	10
11	9999	21.87929	28.35391	23.11385	16.65295	20
12	9999	26.1957	24.86054	18.80658	15.55098	30
13	9999	25.30559	20.82457	16.63618	15.18366	40
14	9999	22.31824	18.03231	15.66784	15.06122	50
15	9999	19.46096	16.45599	15.26343	15.02041	60
16	9999	17.45765	15.66095	15.10141	15.0068	70
17	9999	16.25985	15.28792	15.03834	15.00227	80
18	9999	15.6119	15.12153	15.01429	15.00076	90
19	9999	15.28499	15.05004	15.00527	15.00025	100
20	9999	15.12836	15.02019	15.00192	15.00008	110
21	9999	15.05625	15.00801	15.0007	15.00003	120
22	9999	15.02409	15.00314	15.00025	15.00001	130
23	9999	15.01012	15.00121	15.00009	15	140
24	9999	15.00418	15.00046	15.00003	15	150
25	9999	15.0017	15.00018	15.00001	15	160
26	9999	15.00069	15.00007	15	15	170
27	9999	15.00027	15.00002	15	15	180
28	9999	15.00011	15.00001	15	15	190
29	9999	15.00004	15	15	15	200
30	9999	15.00002	15	15	15	210
31	9999	15.00001	15	15	15	220
32	9999	15	15	15	15	230
33	9999	15	15	15	15	240
34	9999	15	15	15	15	250
35	9999	15	15	15	15	260
36	9999	15	15	15	15	270
37	9999	15	15	15	15	280
38	9999	15	15	15	15	290
39	9999	15	15	15	15	300
40	9999	15	15	15	15	310

附件 2：

代码

```
import pandas as pd

table = pd.read_excel('demand.xlsx')
table.head()

Q = 10
N = 30
table['n00'] = 9999
table.iloc[0,2:7]=0
table.head()

for i in range(len(table)):
    # 当0s时已初始化完毕
    if i==0:
        continue

    for j in range(2,6,1):
        q1 = min(10,table.iloc[i-1,j-1],2/3*(N-table.iloc[i-1,j])) #q1为流入，q2为流出
        if i<=9 and j==5: #此为红灯状态
            q2 = 0
        elif j==5: #红灯结束但为最后一个cell
            q2 = min(10,table.iloc[i-1,j])
        else: #红灯结束其它cell
            q2 = min(10,table.iloc[i-1,j],2/3*(N-table.iloc[i-1,j+1]) )
        table.iloc[i,j] = table.iloc[i-1,j]+q1-q2
    # 统计流出量
    table.iloc[i,6] = table.iloc[i-1,6]+q2
table.to_excel('output.xlsx',index=False)
```