UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Faculty of Industrial Engineering

Academic year 2017-2018

Traffic

MP2: Coding the CTM logic

Prof. Francesc Soriguera

Students: Casteleyn Louis, 't Jampens Cedric & Janis Bormann

Abstract

This report is made in function of the class Traffic, lectured by prof. Francesc Soriguera. The purpose of this work is to code the CTM model in R to simulate traffic behavior on a freeway segment. The output of the program will be a numerical solution to the simulation of the traffic states of this segment. This information will be plotted in a contour plot to analyze the bottlenecks location and its active time. Also N-curves will be plotted to calculate total delays induced by queues. Furthermore, a comparison will be made with a previous project were multiple locations on the stretch were measured. Finally the queue lengths and durations, the possible causes of the bottleneck activation and possible solutions will be discussed.

1 Introduction

The freeway segment that will be coded is the B-23 freeway stretch. A simplified scheme depicting the on- and off ramps is shown in Figure 1.

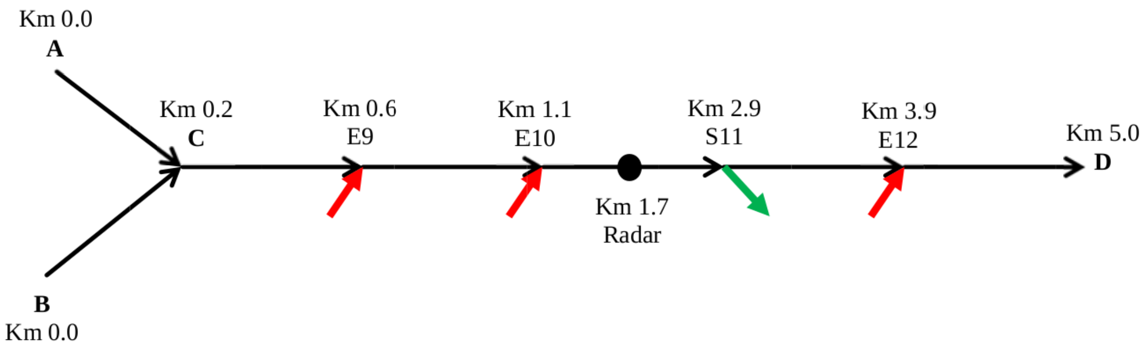


Figure 1: B-23 freeway stretch

The demand entering segments AC and BC as well as the demand willing to enter at the on-ramps is given in an Excel file. It was captured on Tuesday, June 4th, 2013 between 7:00 and 10:00 am. The data is measured in time intervals of 60 seconds. The output will be a numerical solution that is then plotted in a contour plot to visualize the location of the bottleneck(s) and its active time. Also N-curves will be plotted in order to calculate total delays. These results are than compared with a previous report where the activation of bottleneck(s) during the given time period was investigated using oblique plots.

2 Description of the freeway segment

The segment starts with the vehicles coming from two different freeways, A and B, merging into C. The section AC consists out of two lanes and BC out of one lane. The merging priorities of AC and BC at C are corresponding to the ratio of the amount of lanes:

$$mp_{AC} = \frac{lanes_{AC}}{lanes_{AC} + lanes_{BC}} = \frac{2}{3}$$

$$mp_{BC} = \frac{lanes_{BC}}{lanes_{AC} + lanes_{BC}} = \frac{1}{3}$$

Priorities follow the ratio of the lanes if congestion arises:

$$ratio = \frac{mp_{AC}}{mp_{BC}} = \frac{2/3}{1/3} = 2$$

From C onwards to D, the freeway consists out of 3 lanes and has three merges with incoming traffic flow and one diverge with outgoing flow. The lane characteristics are as follows: capacity $Q = 2000$ [veh/h/lane], jam density $k_j = 140$ [veh/km/lane] and freeflow speed $v_f = 100$ [km/h]. If the assumption of a triangular fundamental diagram is made, then the following relations can be constructed for each of the segments. This is done in Figure 12 (Appendix).

Furthermore, it is assumed that on-ramps have enough capacity to handle the receiving demand and their merging priorities when the main freeway trunk is queued are 0.2 for all on-ramps. The off-ramp (S11) has a maximum capacity of 1400 veh/h and supports a turning ratio of 0.2 for most of the morning rush. However during some intervals, the turning ratio increases to 0.25. This activates a bottleneck at the downstream end of the off-ramp because of a roundabout with traffic lights and thus a temporary decrease in capacity as given in Table 1.

Table 1: Capacity drop at off-ramp S11

Period	**Split ratio**	**Capacity**
07:53 - 08:05	0.25	1400 veh/h
08:22 - 09:10	0.25	1085 veh/h
09:45 - 09:55	0.25	900 veh/h

3 Description of the model

The diagrams of Figure 12 can be transformed to fit the CTM model as is depicted in Figure 13 (Appendix) where *S(n)* stands for the sending function and *R(n)* for the receiving function of the flow. The capacity of a block per time interval Q is calculated by multiplying the capacity of the block by the chosen time interval Δt. This has to be conform the following equation:

$$\Delta x \geq v_f * \Delta t$$

Δx is chosen 100 [m] because it is convenient to work with. With v_f equal to 100 [km/h]. This results in Δt being 3.6 [s].

$$\frac{\frac{\Delta x[m]}{1000[m/km]}}{\frac{v_f[km/h]}{3600[s/h]}} = \frac{\frac{100[m]}{1000[m/km]}}{\frac{100[km/h]}{3600[s/h]}} = 3.6\ [s]$$

The 3 diagrams of Figure 13 are important for the implementation. Depending on the segment a particular block is in, it will take on the characteristics of the corresponding diagram which is important for the calculations. This has been summarized in Table 2.

Table 2: Segment characteristics of traffic parameters

Segment	**Lanes**	**Q [veh]**	N_j **[veh]**
AC	2	4	28
BC	1	2	14
CD	3	6	42

Basically the implementation of the model can be described with the illustration in Figure 2 (the dotted thick lines shows that there are blocks in between). Since the total length of the freeway is 5 km as indicated in Figure 1 and Δx was chosen 100 m, the total stretch ends up having 50 cells. The source nodes make it possible to have a flow entering the blocks Gate A, Gate B, E9, E10 and E12 at every time step of 3.6 [s]. These gate cells are programmed to have an infinite capacity of cars ($N_j = +\infty$) so they can accommodate for the vehicles that didn't get through to the first cells on either section AC and BC. The incoming flow capacity at the gates is equal to the given demand per time step, $Q_{gate}[t] = demand_{gate}[t]$. The sink nodes also have an infinite amount of cars they can absorb as stated in the task description. We chose for an implementation that focuses on flows. This implies the creation of 3 vectors containing the flows of the sections AC, BC and CD for every time stamp. These flows are calculated through the formulas of the CTM model for homogeneous flows, merging flows and diverging flows. The green lines are the flows that are stored in the AC vector, the red lines are the flows stored in the BC vector and finally the black lines are the flows stored in the CD vector. The on-ramps also have individual vectors storing the inflow from the source and the outflow to the freeway or vice versa for the off-ramp.

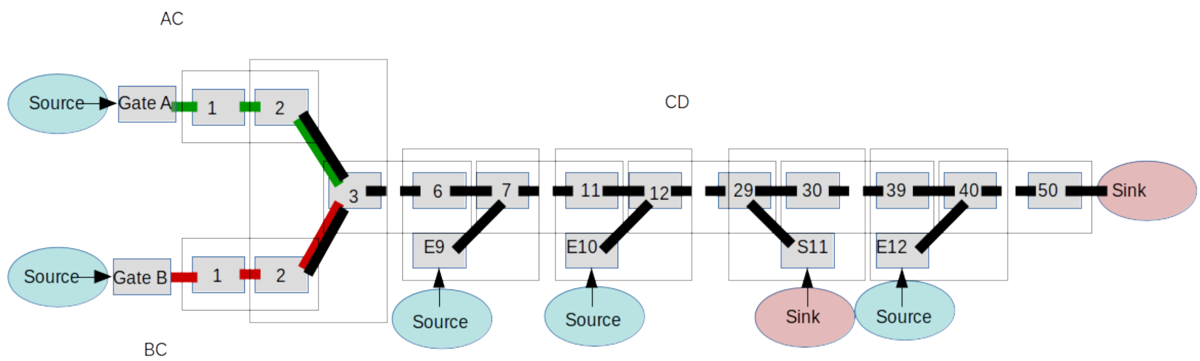


Figure 2: Illustration of implementation

These vectors are stored sequential over time (3000 time steps). It has been chosen to start at the last cell, 50, since a traffic flow works similar. First cars leave the cell, then new cars can fill up the empty space. A for-loop than iterates over all 50 cells and does this for every time step, thus 3000 times. Three matrices were created with the number of rows equal to the number of time steps and the number of columns equal to the amount of cells in a segment, 2 or 48 cells. These are filled up by adding the inflow and subtracting the outflow. Homogeneous, merging and diverging sections have different formulas to

compute the flow. This is solved using if-statements to check what cell is currently being assessed. The boxes on Figure 2,containing the cells, indicate the different if-statements used.
As mentioned before, following the CTM model, calculations depend on the section where the block is located: homogeneous, merging and diverging. These three situations are illustrated in Figure 3.

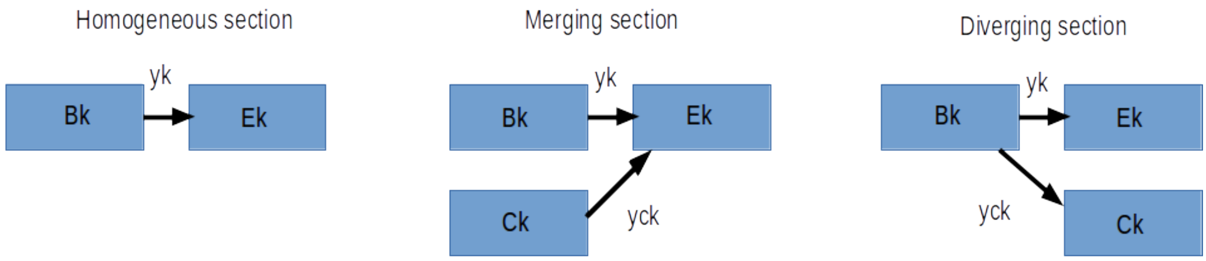


Figure 3: Flows in homogeneous, merging and diverging sections

Homogeneous sections The final flow is the minimum of the following elements that can be derived from Figure 13 in the Appendix. *n(t)* is the amount of vehicles in the cell in function of the time t.

$$y_k(t) = min \begin{cases} S_{bk}(n) \\ R_{ek}(n) \end{cases}$$

with

$$S_{bk}(n) = min \begin{cases} n_{bk}(t) \\ Q_{bk} \end{cases}$$

and

$$R_{ek}(n) = min \begin{cases} Q_{ek} \\ \left(\frac{w}{v_f}\right)_{ek} * (N_{ek} - n_{ek}(t)) \end{cases}$$

If the minimum is found as $S_{bk}(n)$, the flow is determined by the upstream cell and the blocks are in a free flow state (left part of the diagram). On the other hand, if the minimum is found as $R_{ek}(n)$, than the blocks are in a state of congestion.

Merging sections The final flow is calculated as follows:

if $R_{ek} > S_{bk} + S_{ck} \implies$

$$y_k = S_{bk}; y_{ck} = S_{ck}$$

else if $R_{ek} \leq S_{bk} + S_{ck} \implies$

$$y_k = med \begin{cases} S_{bk} \\ R_{ek} - S_{ck} \\ P_k * R_{ek} \end{cases}$$

$$y_{ck} = med \begin{cases} S_{ck} \\ R_{ek} - S_{bk} \\ P_{ck} * R_{ek} \end{cases}$$

With S_{bk} and S_{ck} calculated with the formulas of homogeneous sections and P_k and P_{ck} as the merging priorities.

Diverging sections Finally, the divergent flows are computed as:

$$y_{bk} = min \begin{cases} S_{bk} \\ \frac{R_{ek}}{\beta_{ek}} \\ \frac{R_ck}{\beta ck} \end{cases}$$

$$y_k = \beta_{ek} * y_{bk}$$

$$y_ck = \beta_{ck} * y_{bk}$$

These calculations are programmed as functions that are then used on the appropriate situations. Then finally, the blocks are updated sequentially with the flows:

$$n_i(t+1) = n_i(t) + y_i(t) - y_{i+1}(t)$$

with $y_i(t)$ the flow between i-1 and i during time step t.

4 Code description

First of all, beginning conditions of the CTM and the initialization of variables were implemented. Then, an algorithm has been written that fills up the matrices containing the vehicle flow and cell density per time step for the 3 different segments. The pseudo code of the core program is given in this section. The simple CTM algorithm makes use of 3 functions that also were defined in the program and make the calculations that were illustrated in Section 3 in order to compute the different flows. They have been attached in the appendix.
The actual code can be found in the following link: https://pastebin.com/graX3kSz

Algorithm 1: Main algorithm that fills up the cell matrix

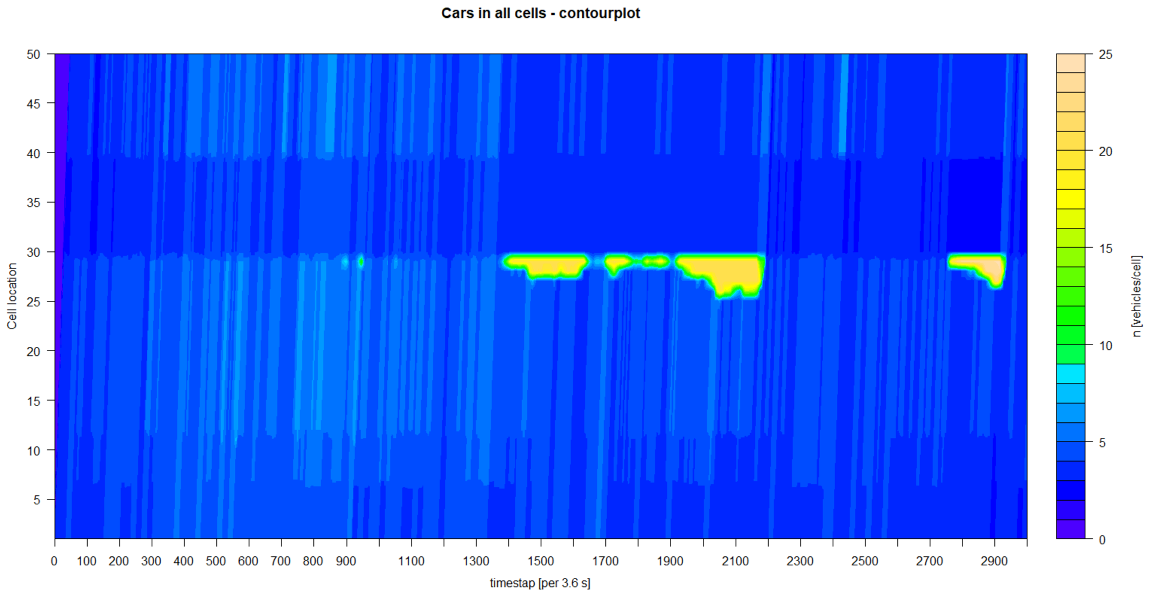
Input : vehicle count entering the freeway at every time step
Output: matrix containing the amount of vehicles per cell per time step

```
for every time step t do
    for for every cell i starting at i = 50 do
        if i at homogeneous section then
            set variables according to characteristics of homogeneous blocks;
            calculate S(n), R(n) and flows using hom_flow function;
            update cell [i,t] in matrix;
        else if i at merging section then
            set variables according to characteristics of merging blocks;
            calculate S(n), R(n) and flows using merg_flow function;
            update cell [i,t] in matrix;
        else if i at diverging section then
            if i at rush hour then
                adapt capacity;
            else
                capacity stays fixed;
            end
        set variables according to characteristics of diverging blocks;
        calculate S(n), R(n) and flows using diverg_flow function;
        update cell [i,t] in matrix;
    end
    initialize cells [t+1] ← cells[t]
end
```

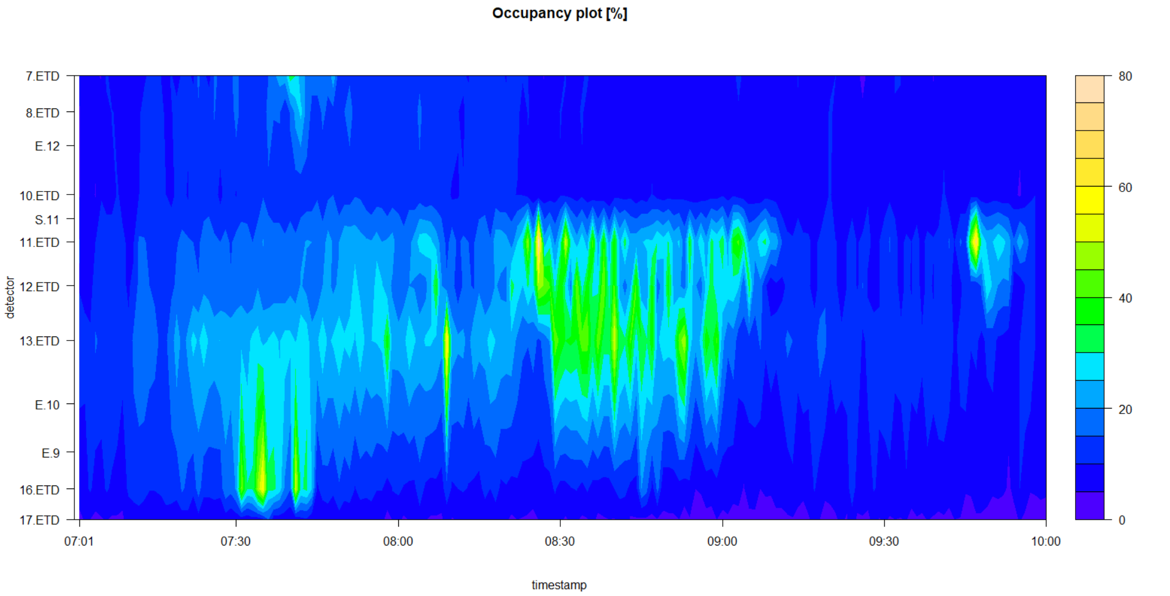
5 Solution using the Cell Transmission Model

5.1 Traffic conditions

The traffic conditions can be visualized using a contour plot as shown in Figure 4(a). The color scale indicates the amount of cars, i.e. density, in the cells. As can seen on the plot, congestion occurs most of the time starting at cell 29. Figure 2 indicates that this is at the off-ramp S11.



(a) Contour plot of MP2

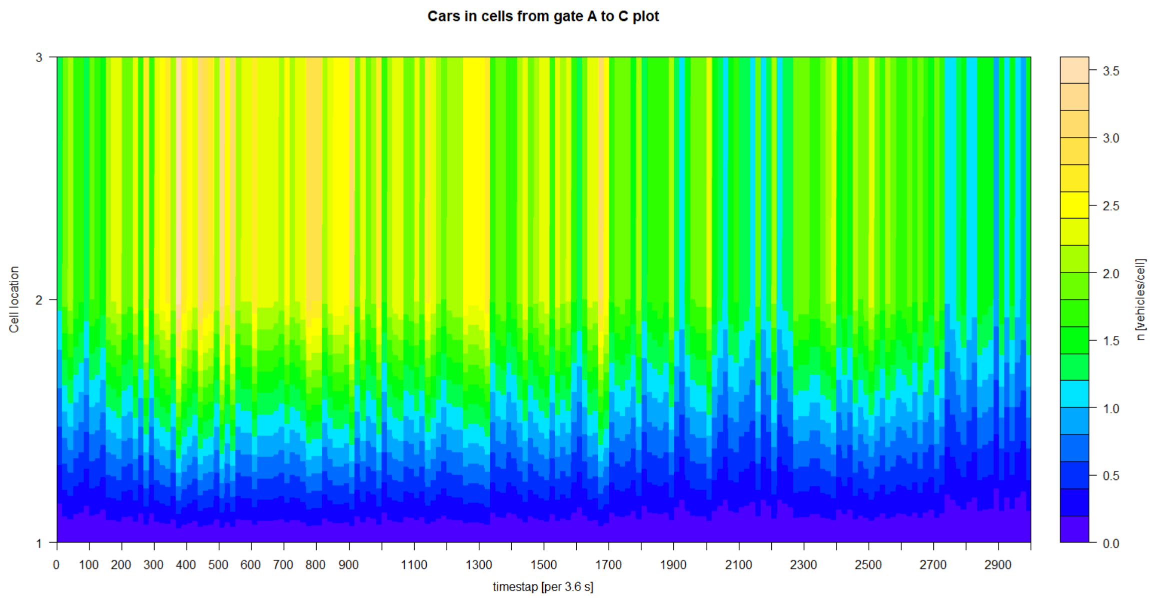


(b) Contour plot of MP1

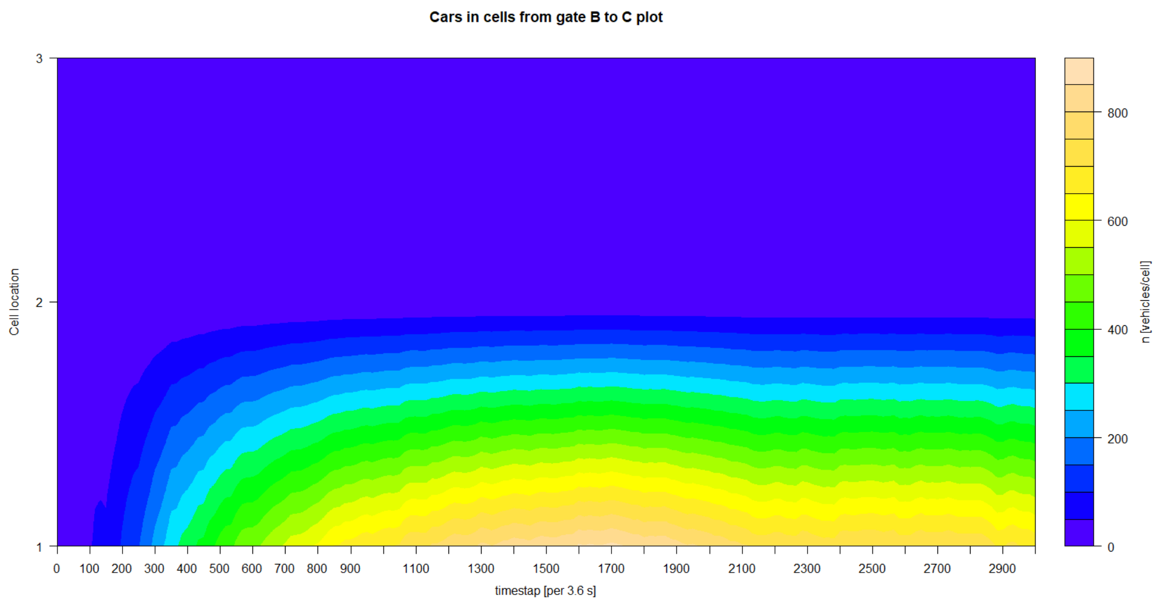
Figure 4: Comparison of contour plots

Congestion, and thus queues, start to appear when more vehicles than the critical density N_0 $[veh]$, in this case equal to the capacity flow of three lanes, $Q_0 = 6$ $[veh]$, are stored in one cell during one time step. Hereby, traffic can not travel at free flow speed and not all vehicles that came in the cell, can leave the cell during the next time step.
A first period in which the bottleneck at off-ramp S11 becomes active has been estimated between time step 1350 and 2200, which corresponds to 08:21:00 and 09:12:00. A second period of congestion appears between time step 2750 and 2950, which corresponds to 09:45:00 and 09:57:00. This is logical since the capacity drop at the off-ramp happens in this period. If compared with the contour plot made in MP1 (Figure 4(b)), then the location of the bottleneck and the time of its activation seem to be consistent. However, the contour plot made in MP1 shows more detail (since it has more locations where there is a vehicle count). This gives a better image of the traffic situation and the length of the queue. This is something that is hard to read from the first contour plot. However, as mentioned before, the location and activation of the bottleneck seems to be accurate enough.
Notice that congested traffic reaches further upstream in the contour plot from the previous mini project as traffic is more realistic due to real life measurements at each detector, and in this mini project, using the simulation tool, traffic behaves optimal and thus reaches less far upstream. Finally, the congested state at detector 17.ETD between 07:03:00 and 07:45:00 can barely seen on Figure 4(a). An explanation for this could be that the data at 17.ETD is aggregated over all lanes, while data now is segregated between two segments. Furthermore, it could be that the congestion is due to a factor which lays before point A or B, which will be discussed later on in the next paragraph.

In order to visualize the situation before the merge at C, in sections AC and BC, Figure 5(a) and 5(b) were constructed. Gate A (cell 1 in the contour plot) is able to let all the demanding flow per time step into the segment since this amount of vehicles never reaches the cells maximum amount of flow, $Q_{cap} = 4$ $[veh]$. On the other hand, Gate B, is not able to push all incoming traffic towards the first cell of the segment BC per time step. This is due to the fact that at point B two lanes merge into one lane and reduces the maximum capacity, and thus penalizing these drivers. Hereby, it 'stores' the excess demand since it has infinite jam density ($N_j = +\infty$). As can be seen on the contour plot, the gate cell in B stays congested until the final measurement at the last time step. Conclusively, this merging from two lanes into one at B makes that point B can be classified as a bottleneck. This bottleneck becomes active from the start and remains active until the end. As the maximum capacity flow in AC is never reached, and the amount of vehicles sent through BC is at maximum capacity at all time, the merge in C will never cause any congestion as long as no queue arises at this point from downstream bottlenecks.



(a) Contour plot of Gate A



(b) Contour plot of Gate B

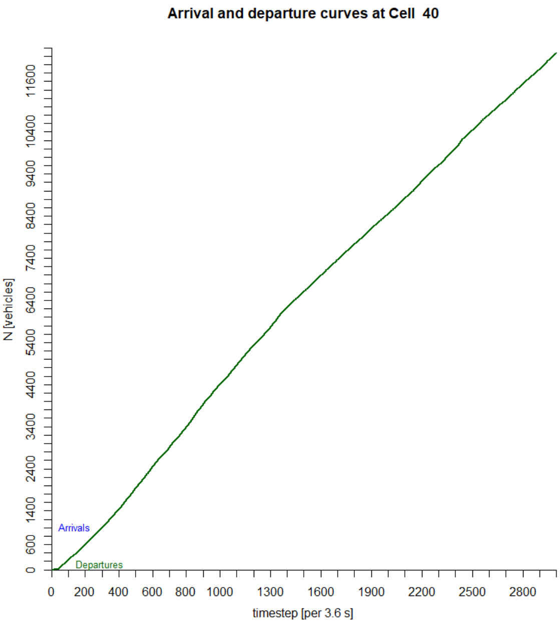
Figure 5: Contour plots at gate cells A and B

Another way to visualize states of congestion is given by plotting the sending, receiving and actual flows in function of time per link. This can be done as S(n) and R(n) are also being stored in separate vectors and has been shown in Figure 14 and Figure 15 (Appendix). The dotted blue line represents R(n) and the gray line S(n). Finally the red dotted line represents the actual flow. In Figure 14, the actual flow in the AC segment is equal to S(n) since it is at all times below R(n) and homogeneous. The BC segment is not able to conduct all the flow since R(n) restricts the flow: they coincide. This is receiving restriction has been discussed above and was visible on the contour plot of this segment as well. However Figure 15 is different. It visualizes the S11 off-ramp and the cells just before it. The S(n) curve between cell 29 and 30 is lower then the R(n) curve, but because of the flow restriction at the off-ramp, the bottleneck is activated, making it impossible to send the wanted flow through the link on the freeway following S11, as this model is a FIFO queuing model. A summary of the links for all on- and off-ramps is given in the Figure 16. Important to notice at E12, there are two small time intervals in which S(n) exceeds R(n), this happens around time step 900 and 2450. Note that during the periods around these time steps, the actual flow does not follow S(n) even though it is below R(n). This is because of the merging priorities that also restrict the flow at this section. The S(n) on the on-ramps of E9 and E10 is never restricted by R(n) and thus there is never queue here.

5.2 Delay and discharge flows

Considering the curves of S(n) and R(n) of the previous section. The arrival and departure curves of the congested cells were constructed. Moreover, the excess in each cell where congestion appears, has been calculated as the difference between the arrived and departed vehicles per time step. The total delay within a cell is equal to the total amount of excess during all time steps multiplied by the length of one time step, 3.6 [s].

On-ramp E12 There are two periods in which the amount of vehicles within cell 40 and the on-ramp can not travel in one time step through the cell, these are located around time step 830 and 2410 and are approximately 35 time steps, or 2 minutes long. Due to our assumption of the limit of 3 minutes, we can not conclude that this merge is a bottleneck which becomes active. However, the arrival and departures curve has been shown in Figure 6. In this cell, the total delay is equal to $0.3 [veh * h]$.



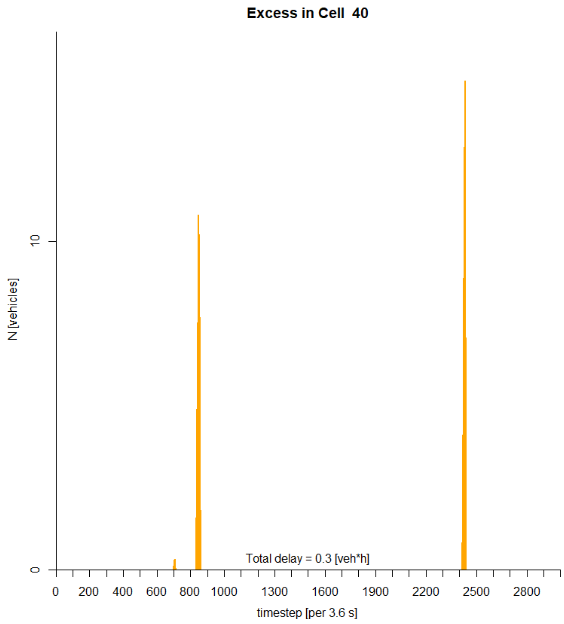


Figure 6: Excess in cell 40 at merge E12

Hereby, the arrival and departures curve at the on-ramp E12, in which vehicles can travel through in the same time step as it is only a gate cell, has also been constructed in Figure 7. Hereby, the total delay that vehicles experience is equal to 0.3 $[veh * h]$, so that both delays are correlated due to the peak in demand at E12 or peak in traffic coming from cell 39 on the freeway.

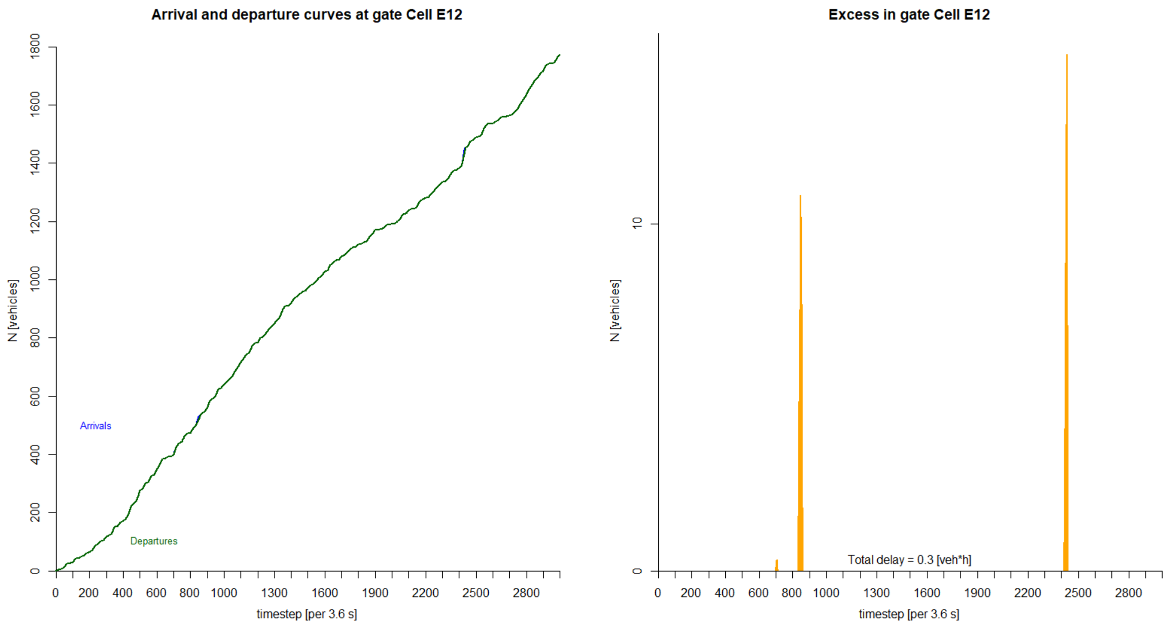


Figure 7: Excess in gate cell at E12

Off-ramp S11 There are several periods in which traffic goes into congestion in cell 29, this is the cell after which vehicles can turn off to the off-ramp S11. Only two main periods are being labeled as active bottleneck times, due to the assumption of 3 minutes. The arrivals and departures curve of this cell has been constructed in Figure 8. Note that, in this cell, the total delay is equal to 14.2 $[veh * h]$. In the excess diagram, at some points, the queue reaches in the first active period cell 25, so then the queue length is equal to 500 [m]. This can be seen in Figure 9. The total delay in cells 28 to 25 are respectively 7.54 $[veh * h]$; 3.2 $[veh * h]$; 1.33 $[veh * h]$ and 0.02 $[veh * h]$. In cell 29, the bottleneck becomes active at 08:22:26 until 09:11:38, which lay close to our approximation discussed earlier at the contour plots. These values are then being compared to the MP1, and lay very close to each other.

Table 3: Bottleneck's first activation time period

Activation times	**MP1 [time]**	**MP2 [time]**
Activation	08:22:00	08:22:26
Deactivation	09:10:00	09:11:38

Next, an appropriate scaled N-curve has been implemented dividing different sections of the diagram related to congestion which was detected earlier on. The free flow sections are labeled with black lines and the congested sections are labeled with orange lines. The first active period can be seen graphically at interval 11 in Figure 10. For each interval, the discharge flow at this cell has been calculated and summarized in Table 7 (Appendix). At interval 11, the discharge flow is equal to 3307.8 $[veh/h]$. A comparison with the previous mini project has been made in Table 4. There, it can be seen the before activation the flow is slightly lower in MP2, but slightly higher during congestion.

Table 4: Bottleneck's first activation flows

Discharge flows	**MP1 [veh/h]**	**MP2 [veh/h]**
Before activation	4151.4	3932.7
During activation	3053.7	3307.8
After deactivation	3258.0	3273.3

During the second active period, the queue only reaches cell 27, so that the queue length is then equal to 300 [m]. The exact activation time is at 09:45:11 and remains until 09:56:13, which was again very close to our estimated values earlier while using the contour plot. These values lay very close to the ones obtained in MP1, so the CTM can be seen as accurate to estimate activation times.

Table 5: Bottleneck's second activation time period

Activation times	**MP1 [time]**	**MP2 [time]**
Activation	09:44:00	09:45:11
Deactivation	09:54:00	09:56:13

This period can be seen graphically at interval 13 in Figure 10. Here, the discharge flow is equal to 2939.7 $[veh/h]$. A comparison with the previous mini project has been made in Table 6. There, it can be seen that flows are close to each other, however, in this mini project, slightly higher.

Table 6: Bottleneck's second activation flows

Discharge flows	**MP1 [veh/h]**	**MP2 [veh/h]**
Before activation	3258.0	3273.3
During activation	2868.0	2939.7
After deactivation	2972.2	3022.0

The average queue discharge rate is then equal to 3240.3 $[veh/h]$. If we compare this value to the first mini project, in which it was 3021.7 $[veh/h]$, than the value is slightly higher using the CTM. Again assuming that the simulation generates idealistic situations, not incorporating e.g. lane changing behavior. Measurements on multiple locations as MP1 would give a more realistic image of what is actually happening.

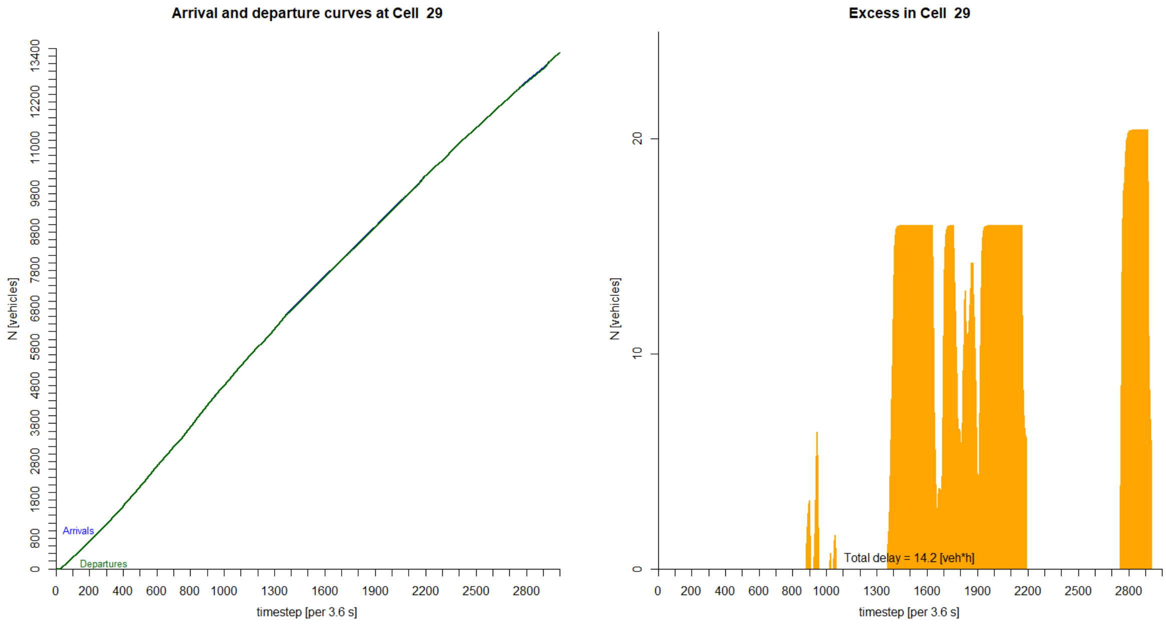


Figure 8: Excess in cell 29 at bottleneck S11

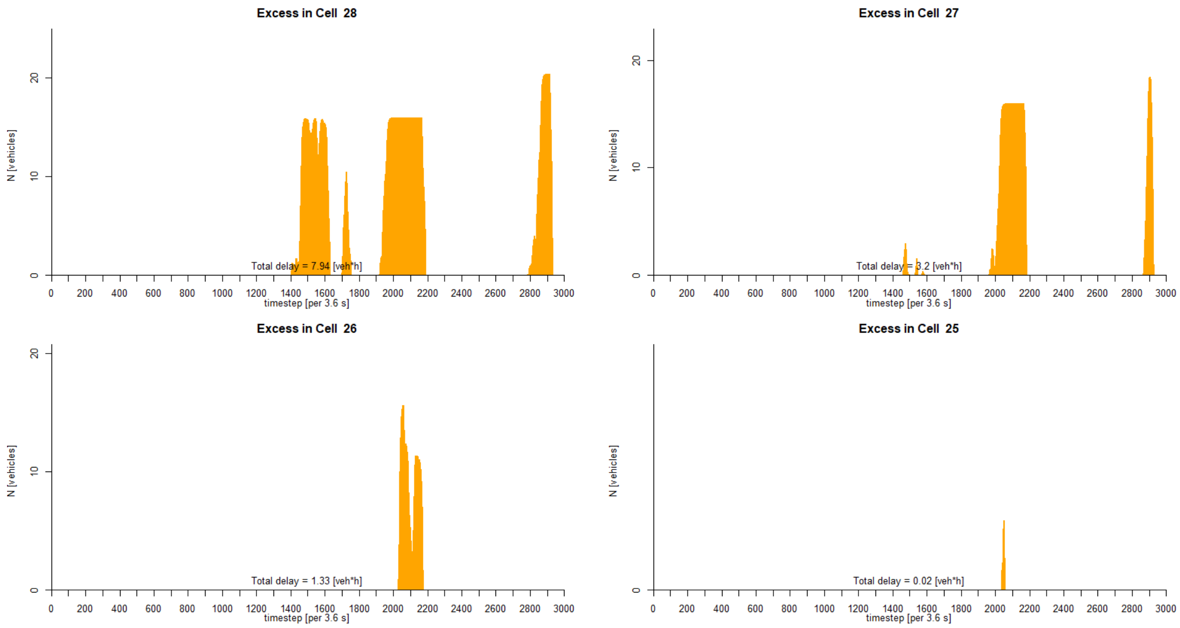


Figure 9: Excess in cells before bottleneck S11

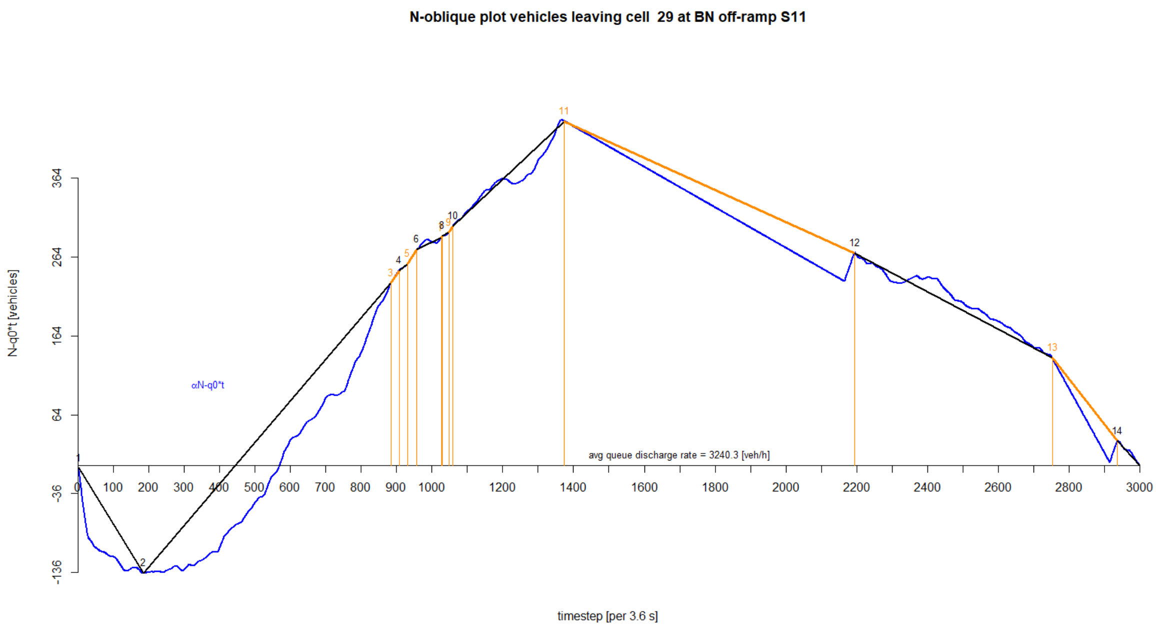


Figure 10: Bottleneck at off-ramp S11

Lane merge B Here, the bottleneck becomes active from time step one and never deactivates within this time period of 3 hours, which is very bad. This congestion can be declared due to the lane drop when the freeway goes into point B. As stated before, this penalty can be introduced to obtain the optimal solution for the freeway in point C, so that no queues arise on section AC. Here, the arrivals and departures curve have been constructed in Figure 11. It can be seen that after time step 1, they never coincide again. In the gate cell, thus the section in front of the lane merge, the total delay is equal to 1925.1 $[veh * h]$. Thus over the three hour period, the average amount of vehicles in the queue is equal to 641.7 [veh]. This is a huge amount, if we follow the reasoning of a two-lane section which has a jam density of 28 [veh/cell], and a cell has a length of 100 [m/cell], the average queue length could be at least $d_{queue} = \dfrac{641.7\ [veh] * 100\ [m/cell]}{28\ [veh/cell]} = 2291.8\ [m]$, which is very long. The discharge flow at each point during this period is equal to the maximum flow, i.e. capacity, of $Q_{BC} = 2\ [veh/timestep]$, which also can be seen in Figure 14 (Appendix). Note that this congestion is not specifically seen in MP1, as there the measured data at 17.ETD is been aggregated between the two lanes of section AC and the two lanes merging in B.

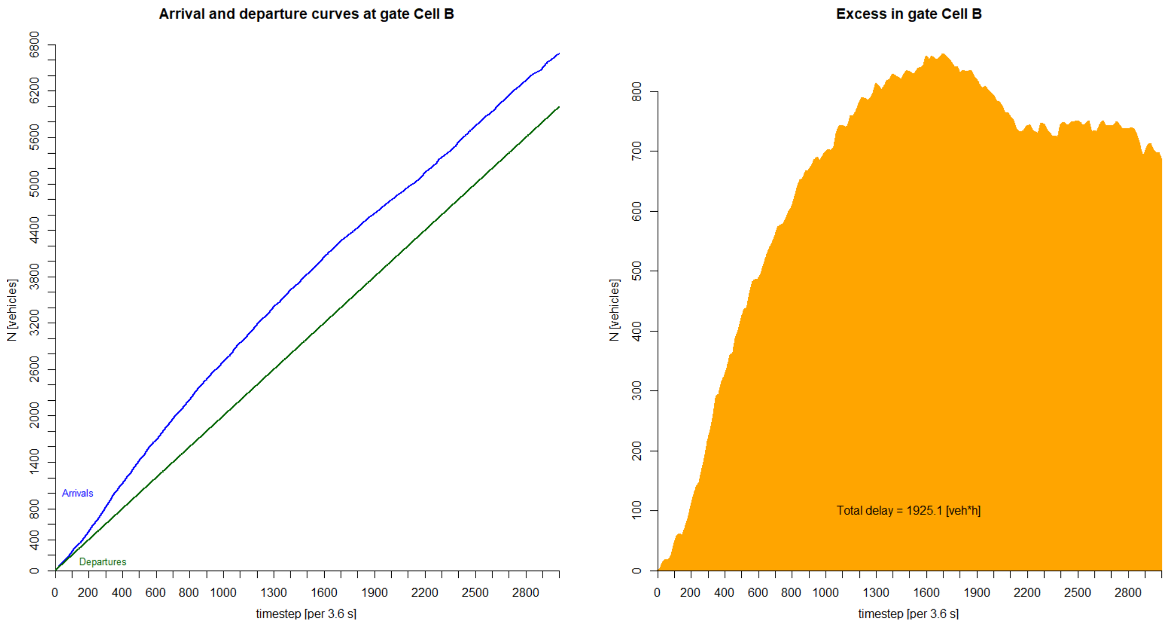


Figure 11: Excess in gate cell B at merge

6 Discussion of solution and conclusion

As mentioned before, 2 bottlenecks were identified: Gate B and off-ramp S11. As explained in Section 2, the BC segment exists out of only one lane while segment AC consists out of two lanes. Taking this into account, the critical and jam density of the AC segment is the double as the one from the BC segment as shown in Figure 14. Because the S(n) curve exceeds the value of 2 on multiple occasions, this second lane is necessary to keep traffic in free flow at all times. However the BC section is all the time restricted by the R(n) curve, this could be on purpose to keep the AC segment in a free flow. Penalizing one section to benefit the majority of the system. A solution could be to maintain 2 lanes until the merge but keep control on the traffic flowing in C by on-ramp metering.
A proposal for the second bottleneck at off-ramp S11 could be to install smart traffic lights at the roundabout further downstream the S11 off-ramp that changes priorities according to the flows in the rush hour. However, this is under the condition that the flows at different entrances of the roundabout are independent of each other. Congestion at off-ramps is also due to lane changing behavior, this is not implemented in the CTM model and thus has little meaning here. However, in a more realistic simulation, a priority lane for exiting vehicles could increase the flow.
We can conclude that the CTM is accurate to estimate the discharge flows and activation/deactivation times of the most prominent bottleneck. However, to estimate queue lengths and less prominent bottlenecks and due to the FIFO logic, the CTM can be seen as less accurate.

Appendix

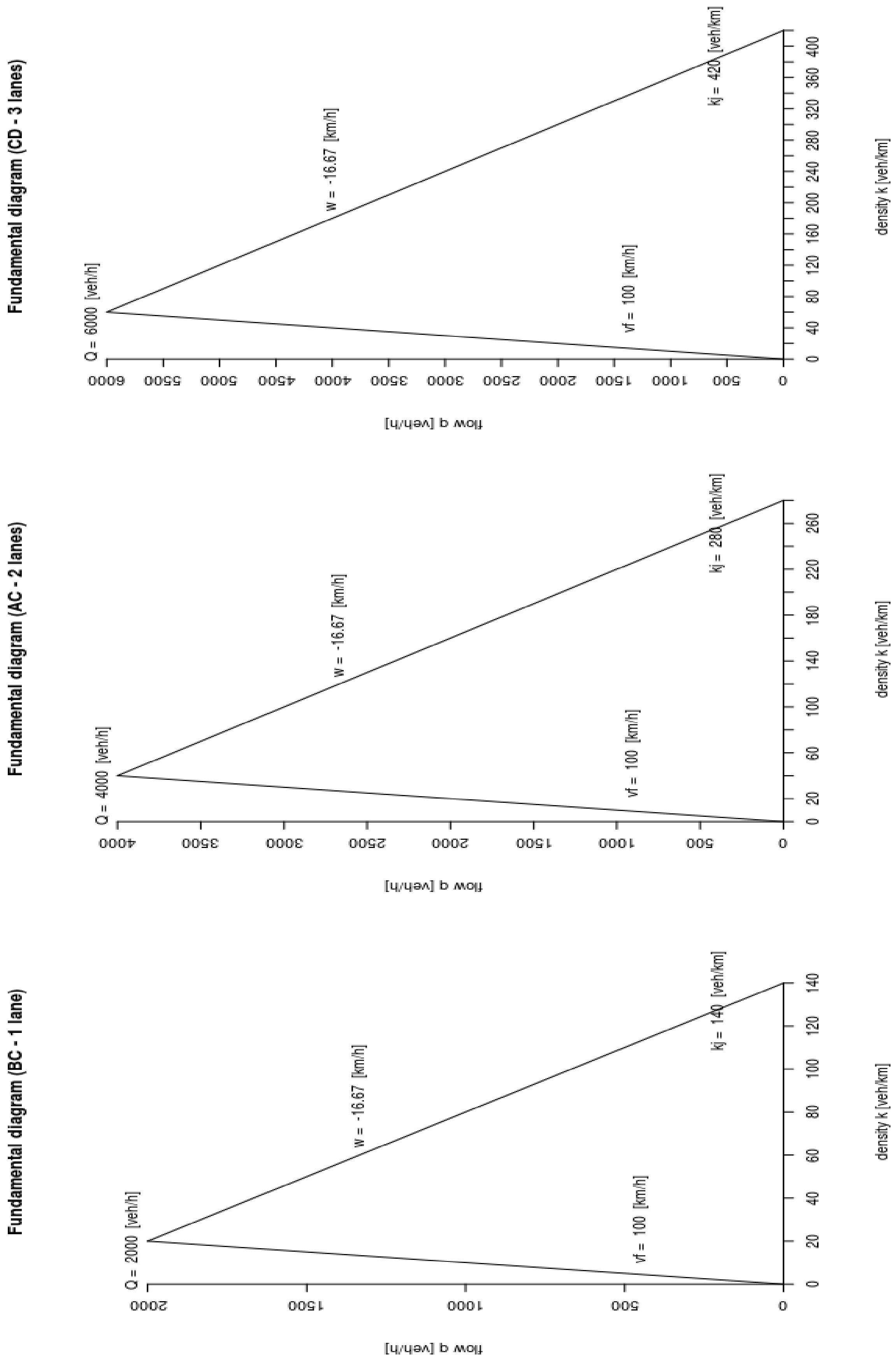


Figure 12: Fundamental diagrams for segments BC (1 lane), AC (2 lanes) and CD (3 lanes)

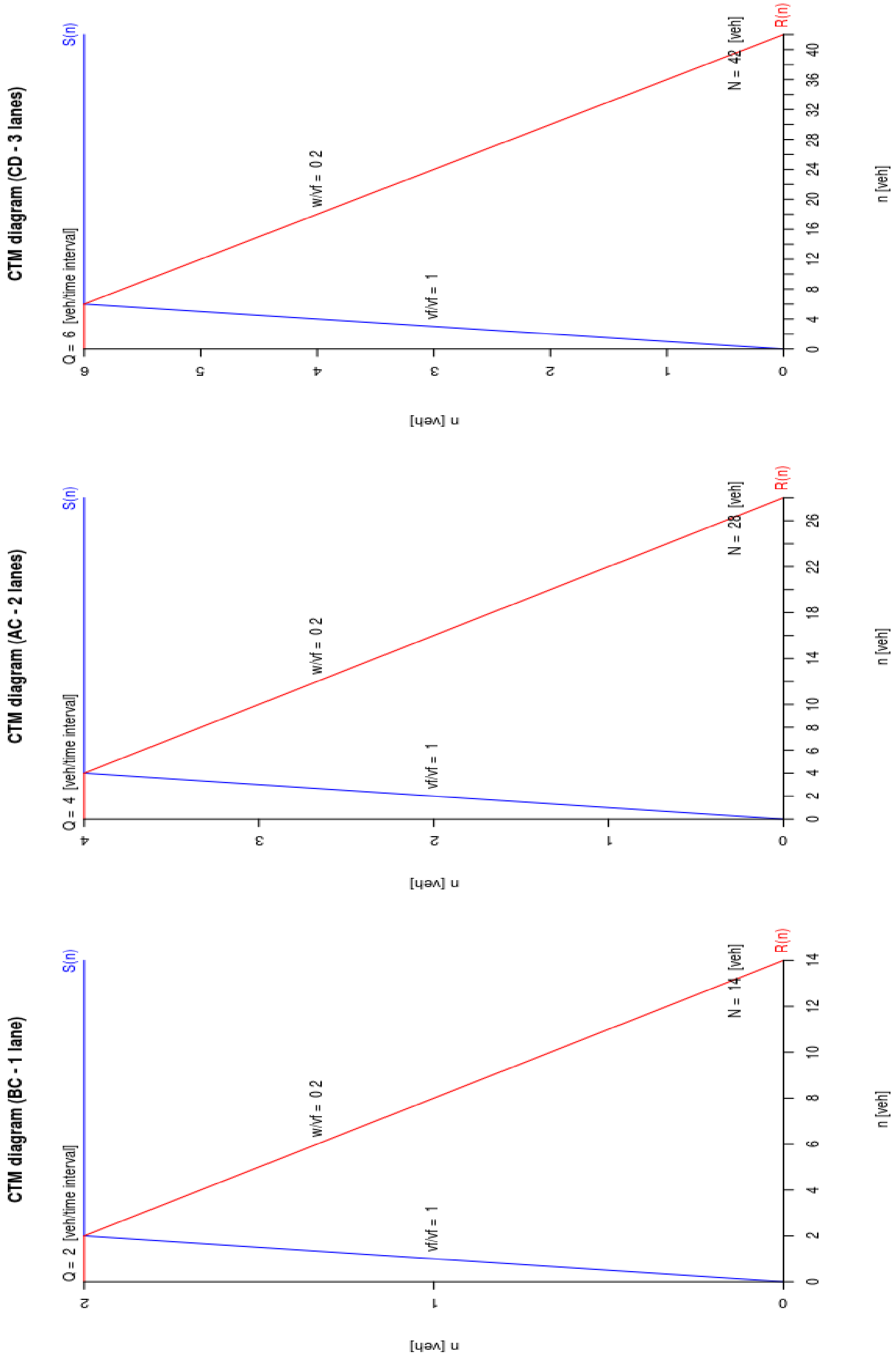


Figure 13: CTM diagrams for segments BC (1 lane), AC (2 lanes) and CD (3 lanes)

Algorithm 2: Function to calculate the flows between blocks in a homogeneous section

```
function hom_flow (n_bk, Q_bk, Q_ek, slope_ek, N_ek, n_ek);
Input  : characteristics of cell bk and cell ek
Output: S_bk, R_ek, y_k
S_ck ← min(n_bk, Q_bk);
R_ek ← min(Q_ek, slope_ek * (N_ek − n_ek));
y_k ← min(S_bk, R_ek);
return S_bk, R_ek, y_k
```

Algorithm 3: Function to calculate the flows between blocks in a merging section

```
function merge_flow (n_bk, Q_bk, Q_ek, slope_ek, N_ek, n_ek, P_k, P_ck, n_ck, Q_ck);
Input  : characteristics of cell bk,ek and ck
Output: S_bk, R_ek, y_k
S_bk ← hom_flow(n_bk, Q_bk, Q_ek, slope_ek, N_ek, n_ek);
R_ek ← hom_flow(n_bk, Q_bk, Q_ek, slope_ek, N_ek, n_ek);
S_ck ← hom_flow(n_ck, Q_ck, Q_ek, slope_ek, N_ek, n_ek);
if R_ek > S_bk + S_ck then
    y_k ← S_bk;
    y_ck ← S_ck;
else
    y_k ← median(S_bk, R_ek − S_ck, P_k * R_ek);
    y_ck ← median(S_ck, R_ek − S_bk, P_ck * R_ek);
end
return S_bk, R_ek, y_k
```

Algorithm 4: Function to calculate the flows between blocks in a diverging section

```
function diverg_flow
  (n_bk, n_ek, n_ck, Q_bk, Q_ek, Q_ck, slope_ek, slope_ck, N_ek, N_ck, beta_ek, beta_ck);
Input  : characteristics of cell bk,ek and ck
Output: S_bk, R_ek, R_ck, y_bk, y_k, y_ck
S_bk ← hom_flow(n_bk, Q_bk, Q_ek, slope_ek, N_ek, n_ek);
R_ek ← hom_flow(n_bk, Q_bk, Q_ek, slope_ek, N_ek, n_ek);
R_ck ← hom_flow(n_bk, Q_bk, Q_ck, slope_ck, N_ck, n_ck);
y_bk ← min(S_bk, R_ek/beta_ek, R_ck/beta_ck);
y_k ← beta_ek * y_bk;
y_ck ← beta_ck * y_bk;
return S_bk, R_ek, R_ck, y_bk, y_k, y_ck
```

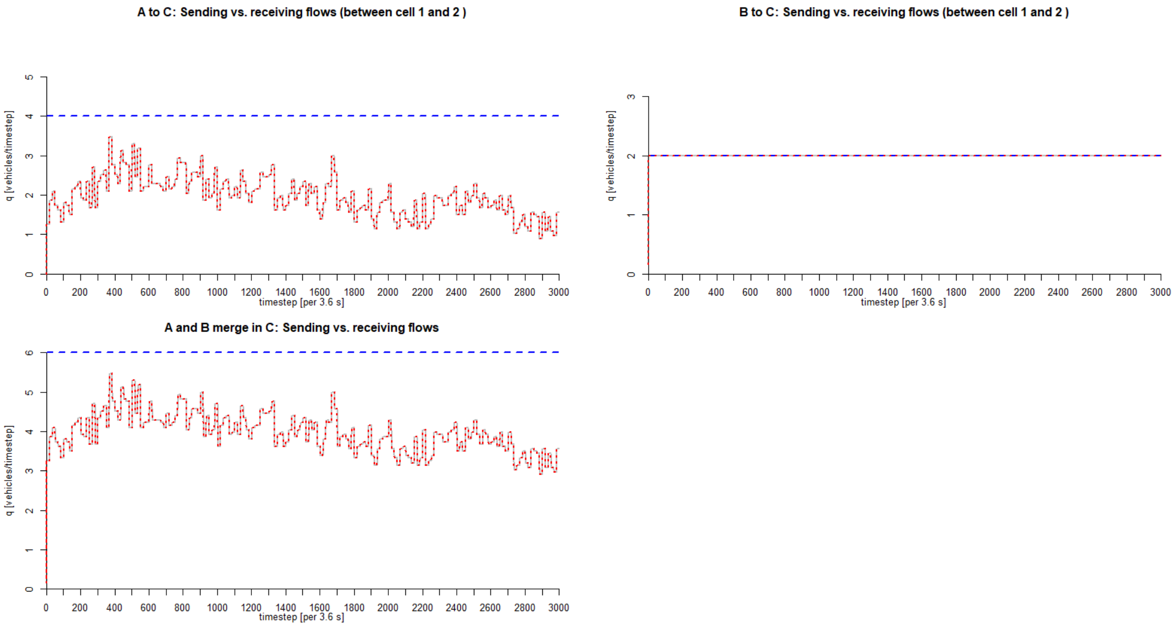


Figure 14: Merging flows at C

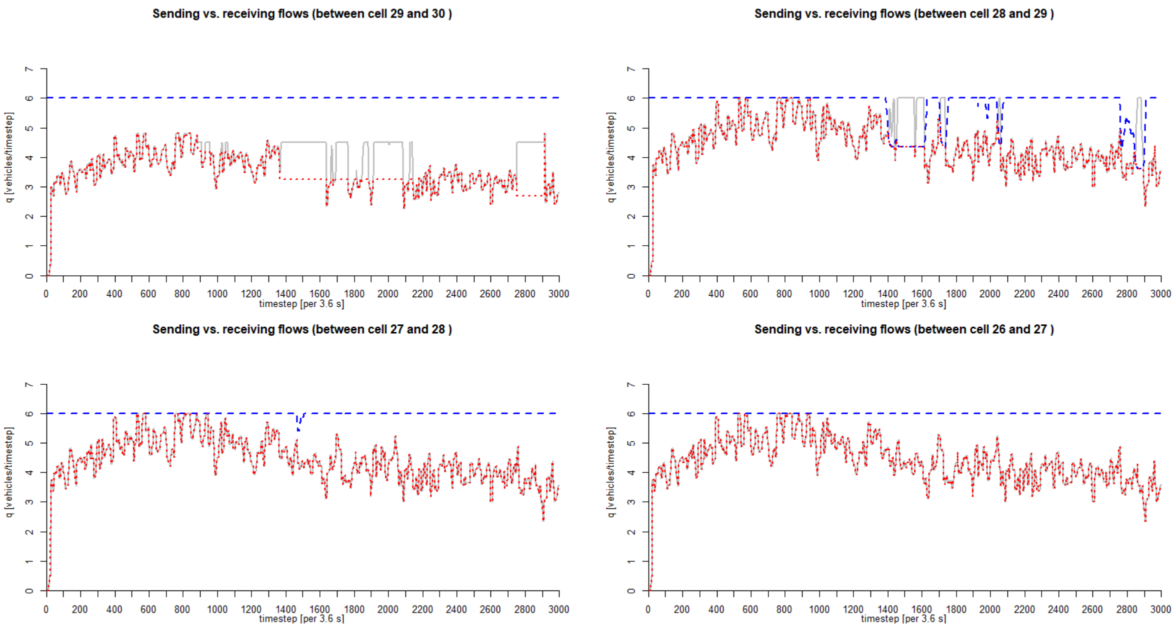


Figure 15: Flows at S11

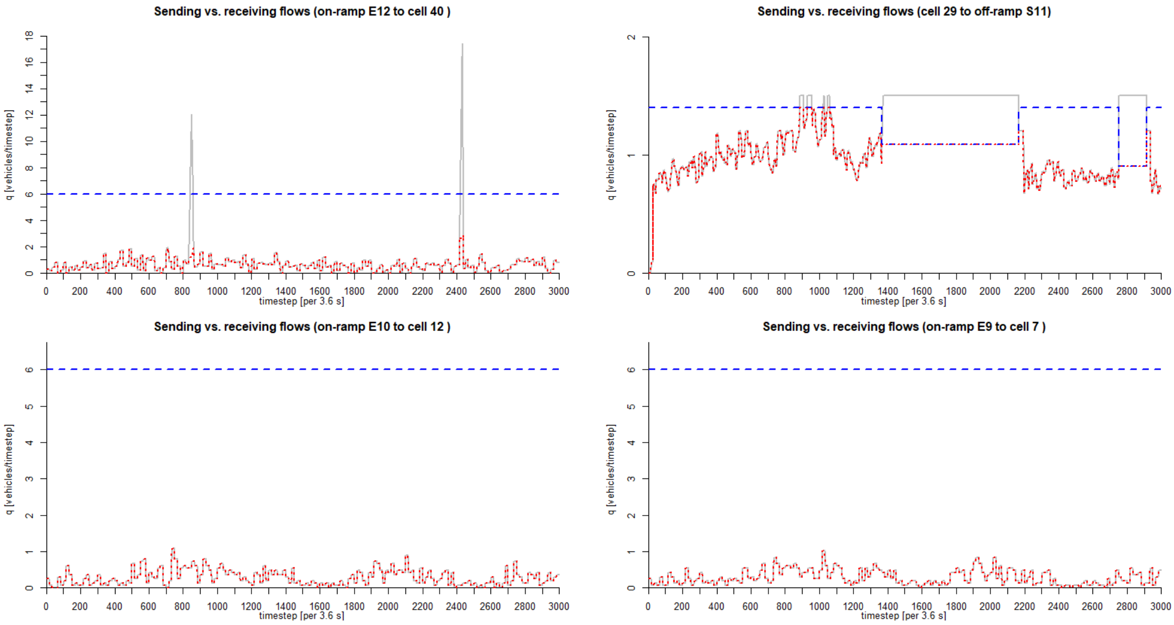


Figure 16: On - and off ramp flows

Table 7: Changing flows on freeway at bottleneck S11

Section	Start time	End time	Flow $[veh/h]$
1	07:00:04	07:11:06	2789.391
2	07:11:06	07:53:02	4037.23
3	07:53:02	07:54:25	4200
4	07:54:25	07:55:52	3861.25
5	07:55:52	07:57:25	4200
6	07:57:25	08:01:34	3728.478
7	08:01:34	08:01:44	4200
8	08:01:44	08:02:49	3756.667
9	08:02:49	08:03:32	4200
10	08:03:32	08:22:26	3932.651
11	08:22:26	09:11:38	3307.756
12	09:11:38	09:45:11	3273.281
13	09:45:11	09:56:13	2939.674
14	09:56:13	10:00:00	3021.968