Contents

[1. Simulation Models 1](#_Toc429998590)

[1.1. Car following model 2](#_Toc429998591)

[1.2. Lane change model 3](#_Toc429998592)

[1.2.1. Lane-changing motivation generation 4](#_Toc429998593)

[1.2.1.1. Mandatory lane-changing motivation 4](#_Toc429998594)

[1.2.1.2. Discretionary lane-changing motivation 4](#_Toc429998595)

[1.2.1.3. Combining discretionary and mandatory lane-changing motivations 5](#_Toc429998596)

[1.2.2. Lane-changing gap acceptance model 5](#_Toc429998597)

[1.2.3. Car following model specifications during lane changing process 6](#_Toc429998598)

[1.2.3.1. Yielding Car Following 6](#_Toc429998599)

[1.2.3.2. Receiving Car Following 7](#_Toc429998600)

[1.2.3.3. After Lane-changing Car Following 7](#_Toc429998601)

[1.3. Traffic generation model 7](#_Toc429998602)

[2. Other simulation models and settings 8](#_Toc429998603)

[2.1. Speed limit distribution 8](#_Toc429998604)

[2.2. Speed friction across lanes 8](#_Toc429998605)

[2.3. Notes on simulation parameters 8](#_Toc429998606)

[2.3.1. Smoothing factor 8](#_Toc429998607)

[2.3.2. Reaction time 8](#_Toc429998608)

[2.3.3. Gap reduction factor 8](#_Toc429998609)

[2.4. Batch run with different combinations of flows 9](#_Toc429998610)

[3. Experiments 9](#_Toc429998611)

[3.1. Traffic breakdown 9](#_Toc429998612)

[3.1.1. Without ACC and CACC vehicles 9](#_Toc429998613)

[4. Codes 18](#_Toc429998614)

[References: 19](#_Toc429998615)

# Manual Simulation Models

The simulation model is built upon the basic framework of the oversaturated traffic model proposed by Yeo. Some important extensions and modification are made in order to represent detailed car following and lane changing behavior that were ignored in the original model.

## Car following model

The car following model is essentially the Newell’s simplified car following model with safety and acceleration limitation constraints ([Newell 2002](#_ENREF_1)). The acceleration equation reads:

Where:

= acceleration [m/s^2]

= speed of the follower [m/s]

= spacing [m]

= leading vehicle’s length [m]

= jam gap [m]

= headway parameter [s]

In Newell’s model, the equilibrium speed-spacing relationship is:

The capacity is then:

Where:

= capacity [veh/s]

For example, if the free flow capacity is 2100 vehicles per hour per lane, the jam gap is 1.5 meter, the length of the vehicle is 4.5 meters and free flow speed is 113 km/h, then the headway parameter is 1.5 second.

The Gipps’ deceleration component [[2]](https://paperpile.com/c/SWNuYU/evJT) is used here to place a safety margin on Newell’s simplified equation, which is formulated as:

Where:

= reaction time [s]

= speed of the follower after reaction time [s]

= buffer time in Gipps model [s]

= speed of the leader [s]

= maximum deceleration of the follower (<0) [m/s^2]

= estimation factor regarding the maximum deceleration of the leader (>1)

The free flow acceleration is implemented by the following equation:

Where:

= Free flow speed [m/s]

= Maximum acceleration [m/s^2]

The final acceleration equation reads:

To smooth the transition between different car-following modes, the following transition treatment is adopted:

Where:

= Smoothing factor (. Suggested values are between 1 and 2.

## Lane change model

The following figure shows the overarching structure of the proposed lane changing model based on Yeo’s oversaturated model ([Yeo, Skabardonis et al. 2008](#_ENREF_2)).



Figure 1 Lane change behavior structure

RCF: Receiving car following mode;

BLC-Sync: Car following mode before lane-changing - I (synchronizing the speed with the leader on the target lane);

BLC-Slow: Car following mode before lane-changing - II (applying a comfortable deceleration to skip the current gap);

ALC: Car following mode after lane-changing;

YCF: Yielding (cooperative) car following mode;

LC: Lane change.

## Lane-changing motivation generation

The lane-changing motivation is measured by a desire index between zero and one. Zero means the driver has no intention for lane changing at all, while one indicates the driver has the highest intention.

## Mandatory lane-changing motivation

If the driver is on the acceleration lane to merge onto the freeway, or the current lane is not an exit lane for the next turning, a mandatory lane-changing desire is generated by using the following equation:

Where:

= distance to the end of the ramp [m]

= time to the end of the ramp [s]

= Parameter [m]

= Parameter [s]

= Number of lane changes needed

The parameters, E and T, are differentiated between on-and-off-ramp mandatory lane changes.

## Discretionary lane-changing motivation

The discretionary lane-changing motivation is generated based on the average speed ahead on the current lane and the adjacent lanes. The anticipatory speed is determine as:

Where

= anticipatory speed on the target lane

= average speed of the target lane (left or right) ahead

= speed of the leader on the target lane

Where

= average speed of the current lane ahead

= parameter (=1 for left lane changing and <1 for right lane-changing)

= distance to the end of the ramp or the point of exit [m]

= constraint parameter. If , . If but the target lane is not one of the exit lanes, equals a positive constant less than . Otherwise, . is an parameter.

## Combining discretionary and mandatory lane-changing motivations

In each time step, the simulation generates three motivation values for each driver: a mandatory lane changing motivation , a left-lane discretionary lane-changing motivation , and a right-lane discretionary lane-changing motivation . These three desires are combined together with the priority on the mandatory lane changing motivation.

* Case 1: the desire of mandatory lane changing is larger than zero ()

If the driver has a mandatory lane changing motivation for a target lane (left/right), then the discretionary lane change desire for the opposite target lane (right/left) is set to zero. That is, if , then . The final desire is then the sum of desires of both mandatory and discretionary lane changes to the same target lane.

Where

= weighting parameter for discretionary lane change.

* Case 2: the desire of mandatory lane changing is zero ().

In this case, the desire is determined by:

A random variable is generated at the beginning of the simulation that follows a normal distribution with the mean as the driver’s average lane changing desire threshold. If , the driver decides to make a lane change at the current time step and begins scanning gaps on the target lane, otherwise, the driver remains on the current lane.

## Lane-changing gap acceptance model

Lane-changing gap acceptance model was differential for mandatory and optional lane changing scenarios. For mandatory lane changing, safety is the primary concern. For optional lane changing, besides safety, the conformability is also taken into account.

## Discretionary lane-changing gap acceptance

Once the driver decides to make a lane change, he/she will scan the gaps on the target lane. The gap acceptance is based on the minimum gap that may happen after the lane-changing maneuver.

* Forward gap:

Where:

= anticipated minimum forward gap that may happen after lane change.

* Backward gap:

Where:

= anticipated minimum backward gap that may happen after lane change.

## Discretionary lane-changing gap acceptance

Besides safety, gap acceptance for discretionary lane changing also considers the anticipatory acceleration.

* Forward gap:

and

* Backward gap:

Because the forward gap is only accepted when the lane-changer anticipates an acceleration, the disturbance caused to the follower on the target lane is also reduced. Through this way, the discretionary lane change is likely to cause less disturbances on the traffic flow.

= forward gap reduction factor

= backward gap reduction factor

## Car following after gap acceptance decision

If the gap is accepted, then the driver will make the lane-changing maneuver immediately. Otherwise, depending which gap is rejected, the driver will adopt different car following behavior.

For mandatory lane changing, if the forward gap is rejected and the backward gap is accepted, then the driver will start synchronizing its speed with respect to the leader on the target lane, which means the driver will follow both leaders on the target lane and the current lane. If the backward gap is rejected, the driver will slow down with a comfortable deceleration, , to skip the current gap, where is the comfortability factor (0<).

For discretionary lane changing, if the backward gap is rejected, the driver will then adopt the normal car following behavior. This is different from mandatory lane changing behavior where the driver would slow to skip the current gap. On the other hand, if the backward gap is accepted but the forward gap is rejected, the driver will synchronize its speed with the leader on the target lane.

## Car following model specifications during lane changing process

During the lane-changing process, both the lane changer and the follower on the target lane adopt some specific car following behavior, including Yielding Car Following (YCF), Receiving Car Following (RCF), Before Lane Changing Synchronizing (BLC-Sync), Before Lane Changing Slowing (BLC-Slow).

## Yielding Car Following

In each simulation step, drivers monitor leaders on the adjacent lanes. If a leader has an intent for a lane change to the current lane, the driver will decide if YCF should be applied based on a cooperative factor . This cooperative factor is generated for each driver at the beginning of the simulation that follows a normal distribution with user specified mean and variation . When a vehicle needs to make a YCF decision, the simulation will generate a random variable between 0 and 1. If this variable is smaller than the cooperative factor , then the driver will apply YCF.

In YCF mode, smaller jam gap, reaction time, and time headway are adopted as follows:

Where:

= reduced time headway in a lane-changing process

= reduced jam gap in a lane-changing process

= reduced reaction time in a lane-changing process

= reduction factor of jam gap (between 0 and 1)

= reduction factor of reaction time (between 0 and 1)

= reduction factor of time headway (between 0 and 1)

## Receiving Car Following

Once a vehicle finish a lane-changing maneuver, the new follower will apply RCF mode. The RCF is the same as the regular CF except that reduced time headway, jam gap, and reaction time are adopted. Once the RCF mode is activated, it will apply the same set of parameters in YCF mode, and then gradually increase the reduction factor, and finally return to the regular CF mode. Specifically, assuming the transition period is denoted by (unit: simulation time step), the reduction parameters are determined as:

Where

= th time step after RCF activates;

, , = reduction factors at th time step after RCF activates.

## After Lane-changing Car Following

When a vehicle successfully merges into the target lane, it will start a transition from the Before Lane-changing CF mode to After Lane-changing CF mode. Similar to the RCF mode, the same transition of reduction factors are applied.

## Traffic generation model

The purpose of traffic generation model is to generate vehicles on the source section given the input flow, destination distribution, and type distribution.

The headway distribution follows the shift negative-exponential distribution as:

Where

= minimum time headway [s]

= distribution parameter

= average time headway where is the lane-based hourly volume [s]

Once a vehicle is generated, then based on the type and destination distribution, the simulation will assign its type and destination. An important part in the traffic generation model is the “holding” function in order to ensure the vehicle will be in steady state (with an acceleration of zero and the speed equals the free flow speed) when it arrives at the origin section.

# ACC and CACC simulation models

## Assumptions

* Manual driving models are applied when ACC/CACC vehicles are on ramps;
* Once a lane changing desired is generated, the manual driving model is applied for ACC/CACC vehicles.

# Other simulation models and settings

## Speed limit distribution

In the real world, the slow traffic moves on the right lanes. To mimic this phenomenon, we apply a lane-based speed limit distribution.

Where

= redistributed speed limit [m/s]

= the section speed limit [m/s]

(Lane one is the rightmost lane)

## Speed friction across lanes

In the fast lanes, when the driver does not observe any leaders on the adjacent lanes who have intention of lane changing, he/she still tries to synchronize its speed with the slower traffic on the adjacent lane and drive cautiously with a lower speed. This friction effect is modeled as follows.

Where

= free-flow speed from AIMSUN [m/s]

= speed ahead on the left/right lane [m/s]

## Notes on simulation parameters

## Smoothing factor

Do not use very high smoothing factor (less than 3). High smoothing factors will lead to crashes or a special case that the fictitious leader at end of ramp is surpassed.

## Reaction time

Before selecting a reaction time, please use the *CompareEqu* in Matlab code to make sure the equilibrium speed profile is higher than the Newell’s model. So that, the underlying model in CF is indeed following Newell’s model. Normally, the value should be low than 1s.

## Gap reduction factor

For the forward gap reduction factor, it should be larger than 0.5. Otherwise, crashes may happen. The backward gap reduction factor is suggested to be larger than 0.6.

## Batch run with different combinations of flows

This function is designed specifically for the on-and off-ramp network. Users should specify the combinations of on-ramp traffic, through traffic and off-ramp traffic volumes, as well as replication IDs in the file “Batch\_Run.txt” in the directory of “C:\CACC\_Simu\_Data”. A small application was written in C# to run multiple simulations. The workflow of the application is as follows:

1. Read Batch\_Run.txt file to determine the number of runs;
2. For each replications, write a temp file named “Batch\_Volume.txt” that will be read by the python script; Write “aconsole.exe” command line into the .bat file and execute it.

A python script is coded to call AIMSUN to run simulations for combinations. Its workflow is shown below.



Figure 2 Workflow of the batch simulations with different volume combinations

# Experiments

## Traffic breakdown

## Without ACC and CACC vehicles

* Parameter Setting

|  |  |  |
| --- | --- | --- |
| Parameters | Value [mean(min, max, dev)] | Unit |
| Car following parameters | | |
|  | 1.5 (1,2,0.5) | m/s^2 |
|  | 110 (100, 120, 10) | km/h |
|  | 1 (0.5, 2, 0.5) | m |
|  | 1.3 (1.1, 1.5, 0.3) | s |
|  | 4.5 (4, 5, 0.5) | m |
|  | 0.5 (0.4, 0.7, 0.2) | s |
|  | 0.3 | s |
|  | -3 (-3.5, -2, 0.5) | m/s^2 |
|  | 1.1 | N/A |
|  | 2 | N/A |
| Lane-changing parameters | | |
|  | 300 (250, 400, 50) | m |
|  | 30 (20, 60, 20)) | s |
|  | 0.7 | N/A |
|  | 0.8 | N/A |
|  | 0.6 | N/A |
|  | 0.75 | N/A |
|  | 0.4 | N/A |
|  | 0.6 | N/A |
|  | 0.2 | N/A |
|  | 0.5 | N/A |
|  | 0.25 (0.2, 0.5, 0.2) | N/A |
|  | 0.8 (0.5,0.9,0.2) | N/A |
| Section parameters | | |
| On-ramp speed limit | | 90 km/h |
| Freeway speed limit | | 104 km/h |

* Results
  + Part 1: Increasing through traffic volume:

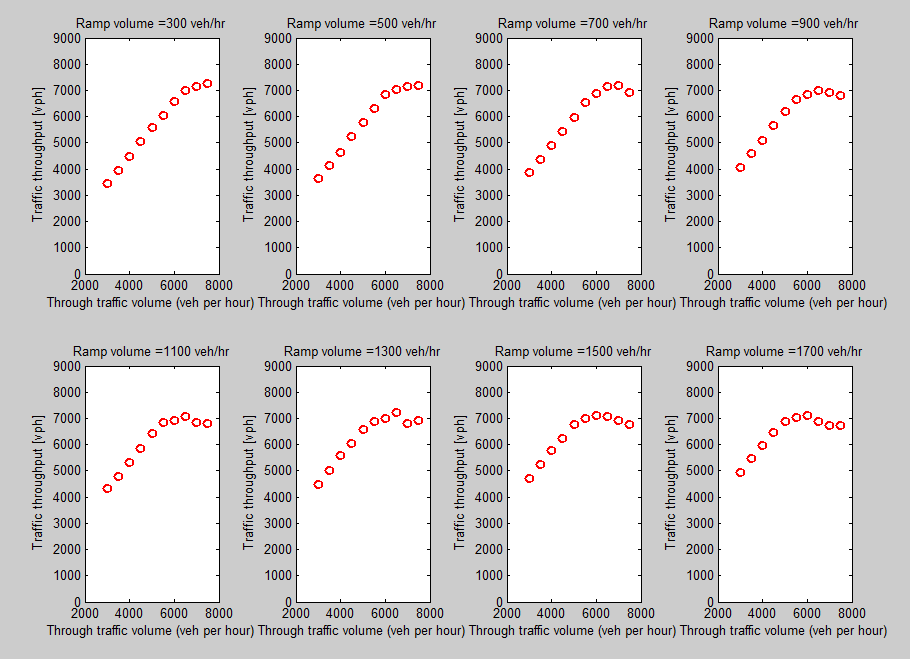


Figure 4 Traffic throughput with respect to through and on-ramp traffic volume

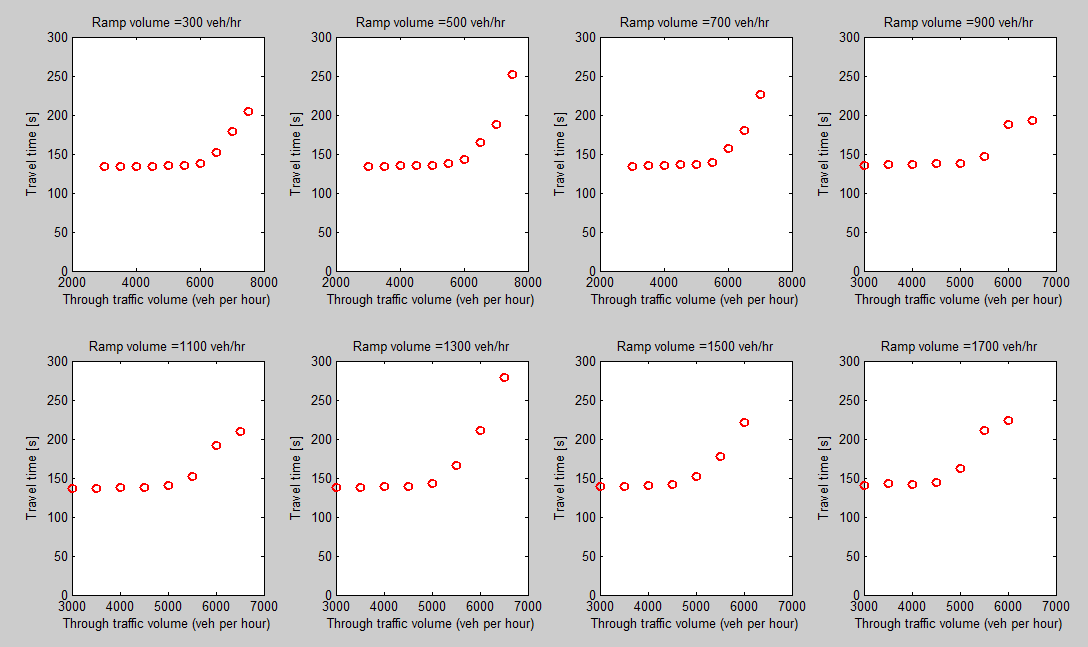


Figure 5 Travel time with respect to through and on-ramp traffic volume

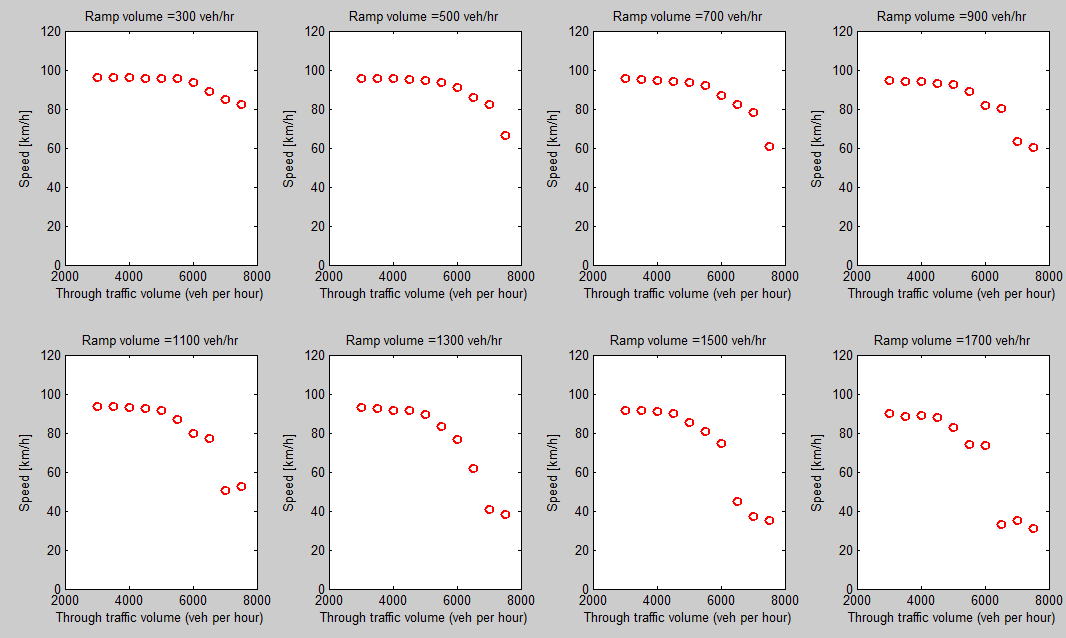


Figure 6 Travel speed with respect to through and on-ramp traffic volume

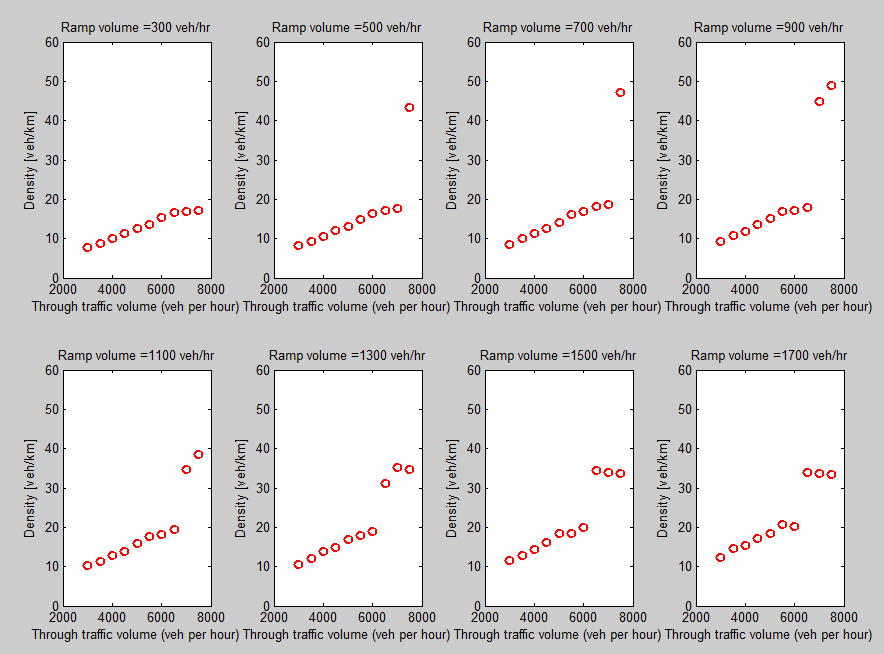


Figure 7 Density with respect to through and on-ramp traffic volume

* + Part 2: Increasing through traffic volume:

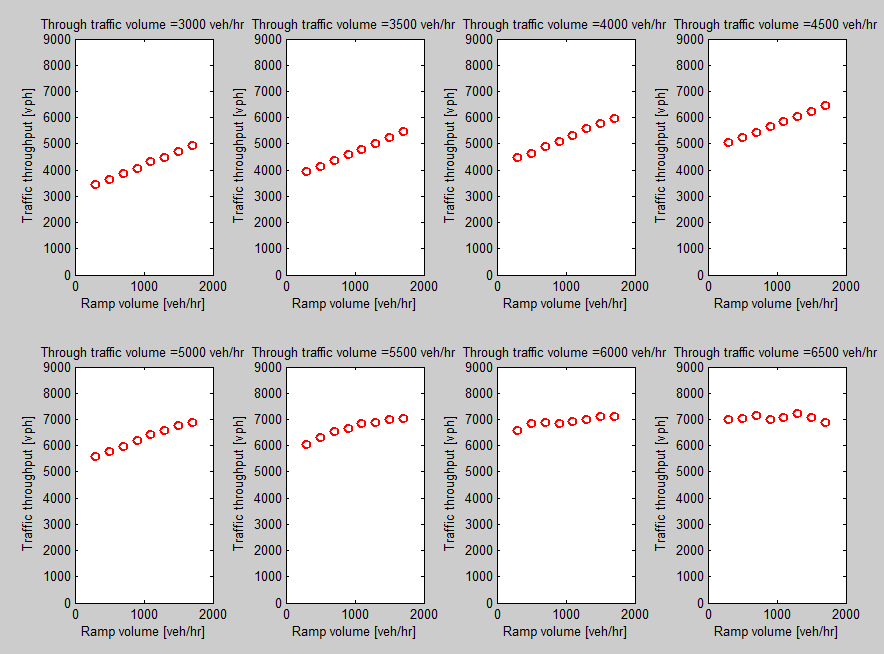


Figure 8 Traffic throughput with respect to through and on-ramp traffic volume

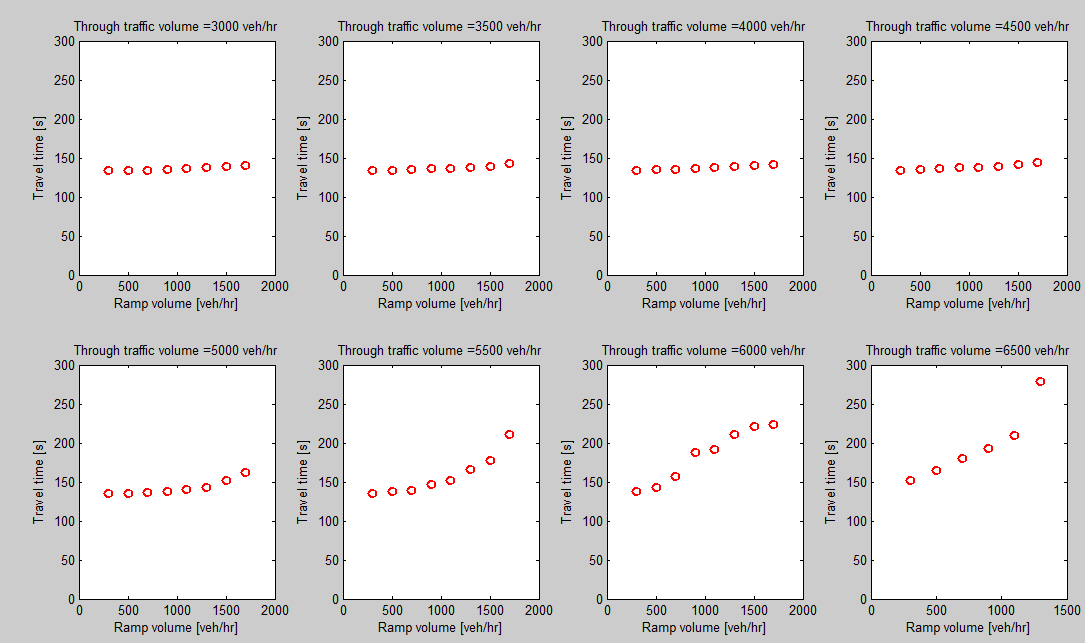


Figure 8 Travel time with respect to through and on-ramp traffic volume

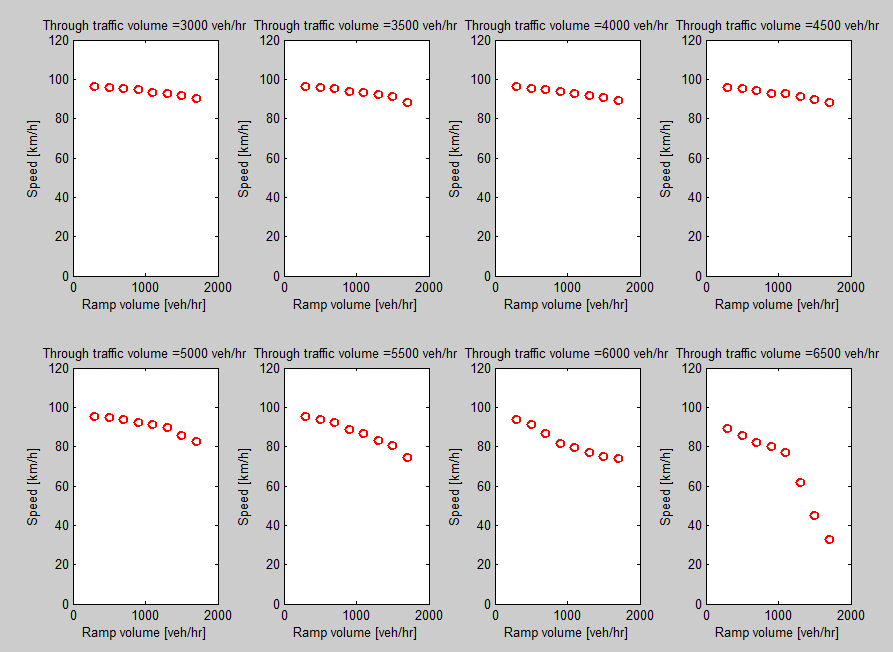


Figure 10 Speed with respect to through and on-ramp traffic volume

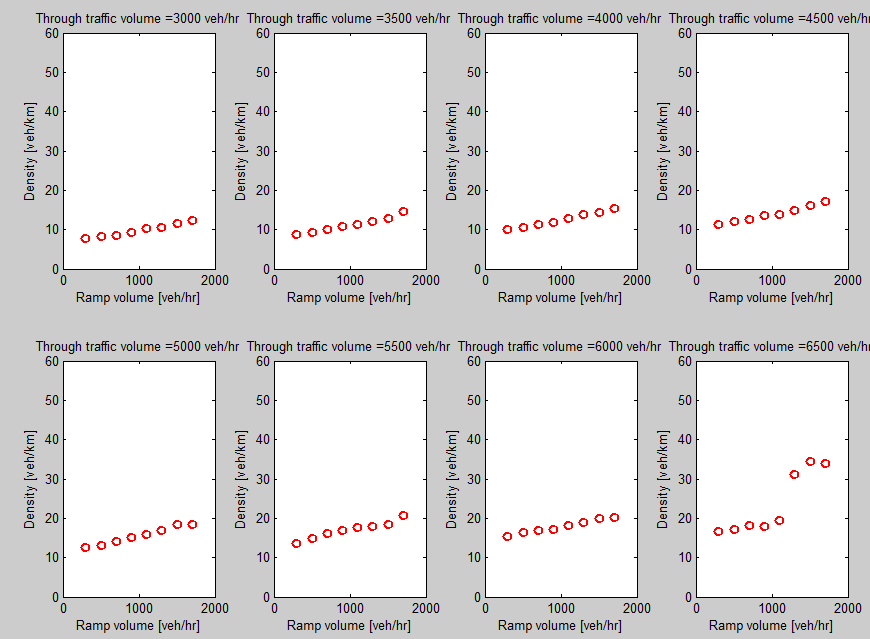


Figure 11 Density with respect to through and on-ramp traffic volume

# Codes

All the codes and network files are available at the following Github repository:

<https://github.com/daliwei/CACC_ACC_MODEL_AIMSUN>

# References:

Newell, G. F. (2002). "A simplified car following model."

Yeo, H., et al. (2008). "Oversaturated freeway flow algorithm for use in next generation simulation." Transportation Research Record: Journal of the Transportation Research Board(2088): 68-79.