

Vissim

(Version 11)

Module 5: Concepts and Tips

Georgia Institute of Technology

Latest Update: 8/10/2021

Contact:

Michael Hunter

Michael.Hunter@ce.gatech.edu

Disclaimer: The objective of these modules is to bring new users sufficiently (and quickly) up to speed so they can use the manual and other resources when they have questions on model development. These modules should not be considered as official guidance; users should always refer to official PTV Vissim or project sponsor documentation for the final word on a model feature question. Send any errors, issues, or comments to michael.hunter@ce.gatech.edu so any corrections can be made to the material.

PTV-Vissim (V11) Module 5 Overview

This is the fifth in a series of Vissim modules. [Module 1](#) guides the user through the creation of a corridor which consists of three signalized, pre-time, four-approach intersections. [Module 2](#) adds coordinated-actuated control to the corridor from [Module 1](#). [Module 3](#) adds an unsignalized intersection. [Module 4](#) familiarizes the user with the basics of designing a freeway segment with a diamond interchange. [Module 5](#) will introduce additional concepts and tips for building and reviewing a Vissim model, exploring some of the Vissim modeling issues not covered in the first four modules (allowing a user to quickly get up to speed) but critical to model development. [Module 5](#) assumes the user is familiar with the concepts in the prior four modules and will build on the final .inpx developed in [Module 4](#). The wireframe of this model is seen in *Figure 1*.

In [Modules 1](#) through [4](#) steps were generally provided “click-by-click”. [Module 5](#) relies on the skills developed in the earlier modules; thus, only the required action may be stated. For example, stating “draw the link” instead of step-by-step instructions on how to draw a link. New material and concepts will receive more detailed guidance.

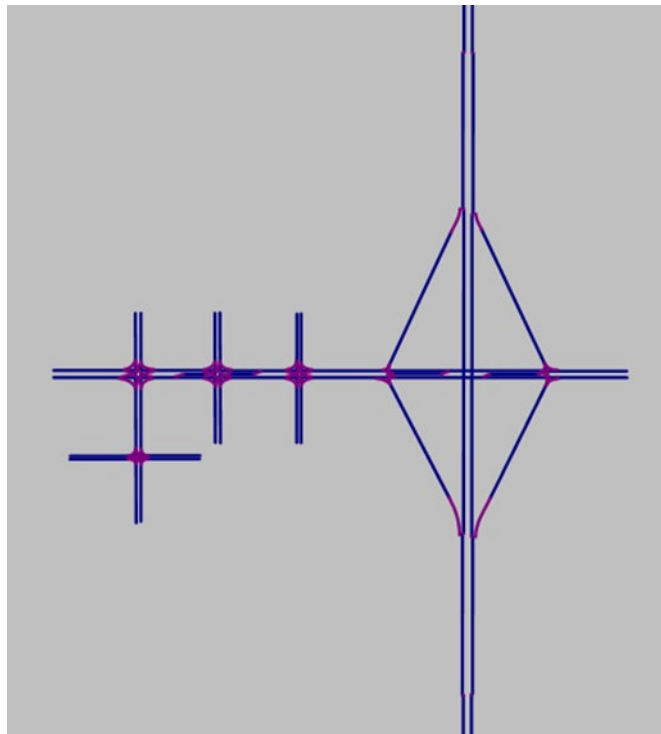


Figure 1: Module 4 final network configuration

Module 5 will step through eight primary topics:

1. Setting vehicle fleet characteristics and composition.
2. Understanding time intervals.
3. Reviewing acceleration, deceleration, and desired speed distributions.
4. Defining driver behavior, lane changing, emergency stop distance.
5. Inserting reduced speed zones and desired speed decision.
6. Utilizing an alternative intersection link configuration.
7. Utilizing an alternative intersection left-turn bay configuration.
8. Utilizing an alternative ramp merge section.

Base Knowledge

For this module, it is assumed that you have a basic understanding of the concepts covered in Modules 1, 2, 3, and 4, including:

- Starting Vissim and navigating the GUI (Graphical User Interface),
- Links and connectors,
- Signal controllers, signal groups, and signal heads,
- Ring barrier control (RBC), and
- Conflict areas.

IMPORTANT: SAVE FREQUENTLY AS YOU WORK!

Open Module 4 Vissim File

To start open the [Module 4](#) Vissim .inpx file. Save the .inpx under a new name, e.g. [Module](#)

5. Let's also update the RBC controllers to:

- RBC_Mod5_Buzz_and_Burdell_v1.rbc
- RBC_Mod5_Buzz_and_Ramblin_v1.rbc
- RBC_Mod5_Buzz_and_Pi_v1.rbc
- Mod 5 Buzz Blvd and Peach Freeway West Int.rbc
- Mod 5 Buzz Blvd and Peach Freeway East Int.rbc

Detailed instructions on opening and saving a Vissim file, as well as renaming RBC controllers, may be found in the prior modules.

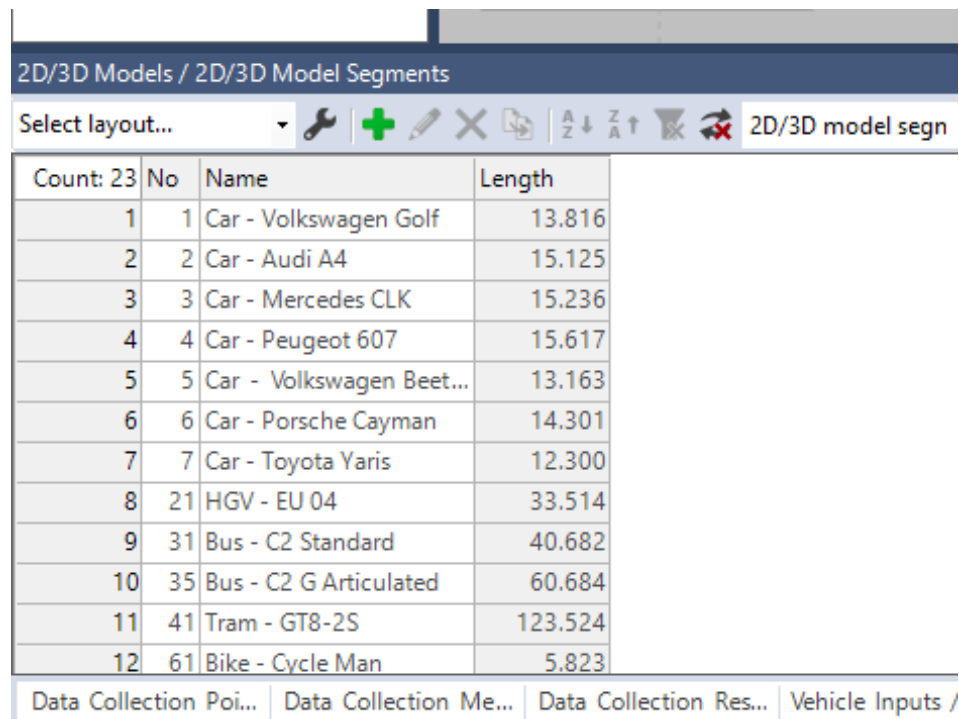
Vehicle Fleet Characteristic and Composition

The model created in [Modules 1](#) through [4](#) is a generic model in that we did not specifically consider the appropriateness of the composition and characteristics of the vehicle fleet. This includes characteristics such as vehicle length, vehicle acceleration and deceleration capability, vehicle distribution (percent cars, trucks, buses, etc.), and so on. To specify the attributes in Vissim we will need to dig into the several of the Vissim defaults:

- 1) Vehicle Make & Models in Fleet (Vissim calls this 2D/3D models),
- 2) Relative Flows,
- 3) Vehicle Types and Classes,
- 4) Vehicle Characteristics, e.g., Min and Max Accel/Decel Distributions, and
- 5) Driver behavior (the type a driving a vehicle will experience, loosely arterial vs freeway).

Vehicle Fleet Make & Model Composition

Vissim refers to the make and model of vehicles as *2D/3D models*. To see the current vehicles in your Vissim model, go to *Base Data* → *2D/3D models*. The *2D/3D Models / 2D/3D Model Segments* window should open. The window should be similar to Figure 2.



The screenshot shows the '2D/3D Models / 2D/3D Model Segments' window in Vissim. The window title is '2D/3D Models / 2D/3D Model Segments'. Below the title bar is a toolbar with icons for 'Select layout...', 'Tools', 'Add', 'Delete', 'Copy', 'Paste', 'Sort', and 'Refresh'. The main area contains a table with the following data:

Count: 23	No	Name	Length
1	1	Car - Volkswagen Golf	13.816
2	2	Car - Audi A4	15.125
3	3	Car - Mercedes CLK	15.236
4	4	Car - Peugeot 607	15.617
5	5	Car - Volkswagen Beet...	13.163
6	6	Car - Porsche Cayman	14.301
7	7	Car - Toyota Yaris	12.300
8	21	HGV - EU 04	33.514
9	31	Bus - C2 Standard	40.682
10	35	Bus - C2 G Articulated	60.684
11	41	Tram - GT8-2S	123.524
12	61	Bike - Cycle Man	5.823

At the bottom of the window, there are tabs for 'Data Collection Poi...', 'Data Collection Me...', 'Data Collection Res...', and 'Vehicle Inputs /'.

Figure 2: *2D/3D Models / 2D/3D Model Segments* default vehicles

You will notice these are all European vehicles. This is Vissim default fleet. If you are modeling a US location it is generally recommended to change the fleet to the US default, as this is more representative of the US vehicle mix. The US mix tends to contain vehicles

that are larger and have other characteristic differences, which can impact model results. For instance, queues tend to be shorter when using the European fleet. To change the fleet to the US default:

- step i. Go to *File* → *Read Additionally* → *Network...*
- step ii. You now need to find the *USA Defaults.inpx* which resides in the Vissim Training Directory. On my computer this was at the location shown in Figure 3. Depending on how your computer was configured this may be in a different location; however, from *PTV Vision* down it should be the same.



Figure 3: File location of USA Defaults.inpx

- step iii. Select *Open*.
- step iv. The *Read Additionally* window in Figure 4 will now be displayed. This is a list of all of the defaults that will be updated to US parameters. If you wish you could unselect some of the objects; however, we will leave them all selected for now.

Select all of the rows in the *Read Additionally* window. (Be sure to scroll down to get them all. They will change from white to yellow when selected.)
- step v. Click on *Edit Selected Cell*.
- step vi. In the drop-down menu on the *Edit Selected* window, *Figure 5*, select *Overwrite object*. Select *OK*.
- step vii. You should be back to the *Read Additionally* window. Unclick *Network Settings*. (If we do not do this when we overwrite the defaults the network location will move.) Select *OK*.

Read	Network object type	Conflict avoidance	Conflict handling	Offset	Discard duplicates
<input type="checkbox"/>	Aliases	New key on conflict	Ignore	0	
<input type="checkbox"/>	User-Defined Attributes	New key on conflict	Ignore	0	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Network Settings	None	Overwrite object		
<input checked="" type="checkbox"/>	Simulation	None	Overwrite object		
<input checked="" type="checkbox"/>	Dynamic Assignment	None	Overwrite object		
<input checked="" type="checkbox"/>	Maximum Acceleration Functions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Desired Acceleration Functions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Maximum Deceleration Functions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Desired Deceleration Functions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Critical Speed Functions	New key on conflict	Ignore	0	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Lateral Drift Speed Function	New key on conflict	Ignore	0	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Desired Speed Distributions	New key on conflict	Ignore	10000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Weight Distributions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Power Distributions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Time Distributions	New key on conflict	Ignore	1000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Location Distributions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Distance Distributions	New key on conflict	Ignore	0	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Color Distributions	New key on conflict	Ignore	1000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	2D/3D Model Distributions	New key on conflict	Ignore	1000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Occupancy Distributions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Vehicle Types	New key on conflict	Ignore	1000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Vehicle Classes	New key on conflict	Ignore	100	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Vehicle Compositions	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Driving Behaviors	New key on conflict	Ignore	1000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Link Behavior Types	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Pedestrian Types	New key on conflict	Ignore	1000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Pedestrian Classes	New key on conflict	Ignore	100	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Pedestrian Compositions	New key on conflict	Ignore	100	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Walking Behaviors	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Area Behavior Types	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Display Types	New key on conflict	Ignore	100	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Levels	New key on conflict	Ignore	10	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Toll Pricing Calculation Models	New key on conflict	Ignore	0	<input checked="" type="checkbox"/>

Figure 4: US defaults to be added to the model

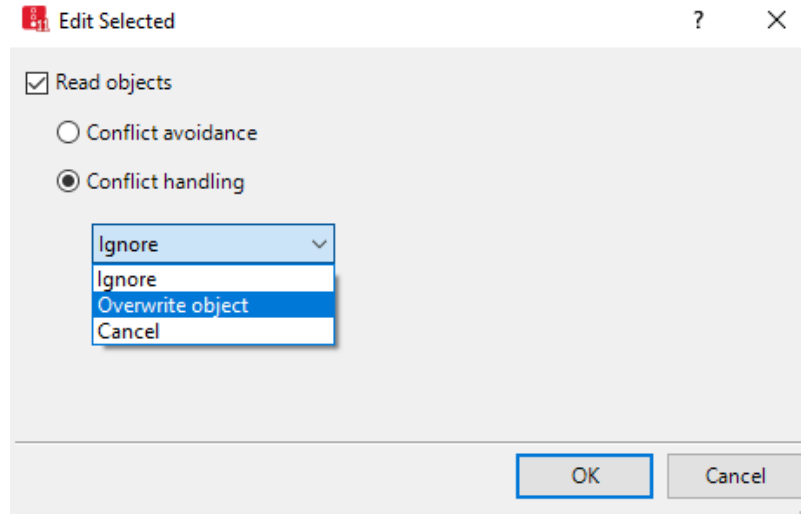


Figure 5 Edit Selected window

step viii. Go to *Base Data* → *2D/3D models* again. Scroll down the *2D/3D / 2D/3D Model Segments* window. You will now see the default US fleet represented as in *Figure 6*.

2D/3D Models / 2D/3D Model Segments			
Count: 31	No	Name	Length
1	1	Car - Honda Accord	15.620
2	2	Car - Nissan Altima	16.024
3	3	Car - Nissan Quest	15.764
4	4	Car - Plymouth Voyager	16.014
5	5	Car - Toyota Avensis	15.404
6	6	SUV - Ford Explorer	16.047
7	7	SUV - GMC Yukon	17.831
8	8	SUV - Jeep Grand Che...	15.226
9	11	LtTruck - Chevrolet Sil...	21.887
10	12	LtTruck - Ford F150	17.753
11	21	HGV - US AASHTO W...	45.724
12	22	HGV - US AASHTO W...	55.000
13	23	HGV - US AASHTO W...	73.575
14	24	HGV - US AASHTO W...	73.244
15	25	HGV - Flatbed	32.577

Figure 6: Default USA vehicle fleet

If we had not set the conflict handling to overwrite then both the European and US defaults would be present. While this would introduce more vehicles you could use, it also complicates the effort, so for now we will stick to the single default set.

It is also possible to change the characteristics of individual vehicles or create new vehicles, although this is beyond the scope of this module. While this will not be required in most modeling efforts there may be certain circumstances where this may be necessary, such as exploring the impact of a transit vehicle or heavy vehicle that has characteristics not found in the default vehicles. The Vissim manual provides guidance on generating new *2D/3D models* and a number of additional vehicle models are available with the Vissim software.

step ix. To see how these vehicles are distributed within the fleet go to *Base Data* → *Distributions* → *2D/3D Models*. Click on *10 Car NA*. This stands for the *Vehicle Type - Car* (we will talk about *vehicle types* in the next section), North American Fleet. You should see something similar to Figure 7. This shows you the distribution of cars that will be used (under *Share* in the right window). For instance, 12.9 percent of cars will be Honda Accords, 6.0 percent will be Nissan Altimas, and so on.

Count	No	Name
1	10	Car NA
2	20	HGV NA
3	30	Bus
4	40	Tram
5	60	Bike
6	61	Bike Man
7	62	Bike Woman
8	100	Man
9	200	Woman
10	250	Woman & Child
11	300	Wheelchair

Count	Share	Model2D3D
1	0.129	1: Car - Honda Accord
2	0.060	2: Car - Nissan Altima
3	0.064	3: Car - Nissan Quest
4	0.055	4: Car - Plymouth Voyager
5	0.135	5: Car - Toyota Avenis
6	0.106	6: SUV - Ford Explorer
7	0.050	7: SUV - GMC Yukon
8	0.058	8: SUV - Jeep Grand Cherokee
9	0.192	12: LtTruck - Ford F150
10	0.151	11: LtTruck - Chevrolet Silvera...

Figure 7: Composition of USA Vehicle Type car

step x. You should also be able to see that the default vehicle types in the model have been updated. For instance, go to *Vehicle Inputs* in the *Network Objects* and open the list, Figure 8. You should now see that the Vehicle Composition (*VehComp(0)*) is reset to *Default NA*, where previously it was *Default*.

Count	No	Name	Link	Volume(0)	VehComp(0)
1	1	Buzz Blvd, EB inp	1: Buzz Blvd EB	300.0	1: Default NA
2	3	Burdell Dr., SB inp	3: Burdell Dr. SB	300.0	1: Default NA
3	4	Ramblin St., NB inp	6: Ramblin St. NB	300.0	1: Default NA
4	5	Ramblin St., SB inp	5: Ramblin St. SB	300.0	1: Default NA
5	6	Pi Pkwy, NB inp	8: Pi Pkwy NB	300.0	1: Default NA
6	7	Pi Pkwy, SB inp	7: Pi Pkwy SB	300.0	1: Default NA
7	8	Buzz Blvd, WB inp	2: Buzz Blvd WB	600.0	1: Default NA
8	9	Varsity Way EB	11: Varsity Way EB	150.0	1: Default NA
9	10	Varsity Way WB	12: Varsity Way WB	150.0	1: Default NA
10	11	Peach Freeway SB	13: Peach Freeway SB	1500.0	1: Default NA
11	12	Peach Freeway NB	14: Peach Freeway NB	1500.0	1: Default NA

Count	Cont	TimeInt	Volume	VehComp	VolType
1	<input type="checkbox"/>	0-MAX	1500.0	1: Default...	Stocha...

Figure 8: Vehicle Inputs list

- step xi. However, not all default settings are the same between the European and US fleets. Go to *Traffic* → *Vehicle Compositions*. In the pop-up window, you should see *Freeway default* and *Default NA*. We had selected a desired speed (*DesSpeedDistr*) of 50 kph (30 mph) for vehicles entering Buzz Blvd or the cross streets. If you click on *Default NA* you will see on the right side that the desired speed is set to 40 mph for the US fleet. Use the pull down and reset both the car and HGV desired speed to 30 mph.
- step xii. Next check the freeway mainline as well as the on- and off-ramp *desired speed decisions* that were created in [Module 4](#). The freeway mainline *desired speed* was set to 120 kph, on-ramp to 120 kph, and the off-ramp to 50 kph. Set these to 70 mph, 70 mph, and 30 mph, respectively. When you make these updates be sure you change the *desired speeds* for all *vehicle types* used (*Car* and *HGV* in our case). (The conversion from 120 kph is actually 72 mph, however, we would need to create a new *desired speed distribution*. For now, we will go with 70 mph. Later on, we will see how to create or change the underlying desired speed distribution.)

CRITICAL CONCEPT

When you updated the defaults, we used the overwrite feature. As seen in Steps xi and xii not all values of the new defaults may convert to the equivalent of the previous value. When reading in data from other sources always be sure to confirm that model objects have updated as you expect. In many instances the update will be ok, e.g., we want to use the acceleration distributions for the US fleet rather than European. However, in other cases you must fix the values. For instance, for *DesSpeedDistr* when changing from the European to US defaults the speeds often do not convert correctly, e.g., updating from 40 kph to 40 mph!

This will be particularly true if you added the new default values rather than overwriting. In that case both the European and US defaults would be present in the model. Model objects such as *Vehicle Inputs* would continue to reference the European defaults and would need to be updated. Or, if you open the Detectors list you will see the detectors are set to *AllVehTypes*, this is the default setting. If we made the detectors specific to certain vehicle types then you would need to update the detector to the new US composition. This is also true of routing decisions, signal heads, and many other model objects. Always confirm that model objects have updated their references as you expect. These are good items to check when reviewing someone else's model.

Helpful Hints

In order to jump right into the developing a Vissim model *the 2D/3D models and vehicle categories, types, and classes* (discussed next) were skipped in [Module 1](#). However, when starting a new model it is best to start from the USA default .inpx as this will set the various default values at the start of your model development. When working from a model that is already created you will need to go through the process as described above (and below) if the model has not been set to the US defaults. In such cases be sure to check the update on all objects where the vehicle type is been directly referenced.

It is critical that **you and all future model users, reviewers, etc.** are aware of the chosen default definitions. Different users having different assumptions about the underlying model construction and defaults is situation ripe for error. It is good practice to develop model construction protocols for each model, and as possible, provide consistency across modeling efforts. This document should be provided whenever a model is reviewed or updated.

Vehicle Categories, Types, and Classes

Vissim uses a hierarchy to classify vehicles, define various vehicle/driver characteristics, and assign vehicles to Vissim objects. At the base is categories, each vehicle type is assigned a category, and each vehicle class is assigned vehicle types. In the current default setting you will notice that *Categories* are numbered using ones, *Vehicle Classes* are number using tens, and *Vehicle Types* using hundreds, e.g., the *Car Category* is labeled *No. 1*, the *Car Vehicle Class* is labeled *No. 10*, and the *Car Vehicle Type* is labeled *No. 100*.

Vehicle Categories

Vissim groups vehicles with “similar technical driving properties” into *vehicle types*. Each *vehicle type* is assigned a *category*. There are six default categories in Vissim: *Car*, *HGV* (i.e., heavy vehicle), *Bus*, *Tram*, *Pedestrian*, and *Bike*. The *category* provides the basic *vehicle type* behavior, such as weight and power distribution, special rules in the application of car-following models (e.g., a Tram drives at exactly the desired speed), special lane changing rules (e.g., a Tram cannot change lanes), etc. Occasionally you may wish to adjust some of these parameters.

For example, the *maximum acceleration*, *desired acceleration*, *maximum deceleration*, and *desired deceleration* are all *category characteristics*. You can see these distributions under

Base Data → *Functions*. Later in this module we will learn how to open and adjust these underlying distributions.

Vehicle Types

To understand *vehicle types*, go to *Base Data* → *Vehicle Types*. The *vehicle types* window, similar to Figure 9, should be displayed.

Count	No	Name	Category	Model2D3DDistr	ColorDistr1	OccupDistr	Capacity
1	100	Car	Car	10: Car NA	1: Default	1: Single Occupancy	0
2	200	HGV	HGV	20: HGV NA	1: Default		0
3	300	Bus	Bus	30: Bus	1: Default	1: Single Occupancy	110
4	400	Tram	Tram	40: Tram	1: Default	1: Single Occupancy	215
5	510	Man	Pedestrian	100: Man	101: Shirt Man		0
6	520	Woman	Pedestrian	200: Woman	201: Shirt Woman		0
7	600	Bike	Bike	60: Bike	101: Shirt Man		0
8	610	Bike Man	Bike	61: Bike Man	101: Shirt Man		0
9	620	Bike Woman	Bike	62: Bike Woman	201: Shirt Woman		0

Figure 9: Vehicle Types

Double click on the row for *100 Car*, the USA default. A pop-up window similar to Figure 10 should now be displayed.

Vehicle type

No.: 100 Name: Car

Static Functions & Distributions Special External Driver Model

Category: Car

Vehicle Model: 10: Car NA

Length: from 15.23 ft to 21.89 ft

Width: from 6.65 ft to 8.52 ft

Colors

Color 1: 1: Default

Color 2:

Color 3:

Color 4:

OK Cancel

Figure 10: Vehicle Type pop-up window

Under the *Static* window you can see that the *Vehicle Type* is named *Car*, has the *Category Car*, and the *Vehicle Model 10: Car NA*. If you select *Functions and Distributions* the window should now be similar to Figure 11.

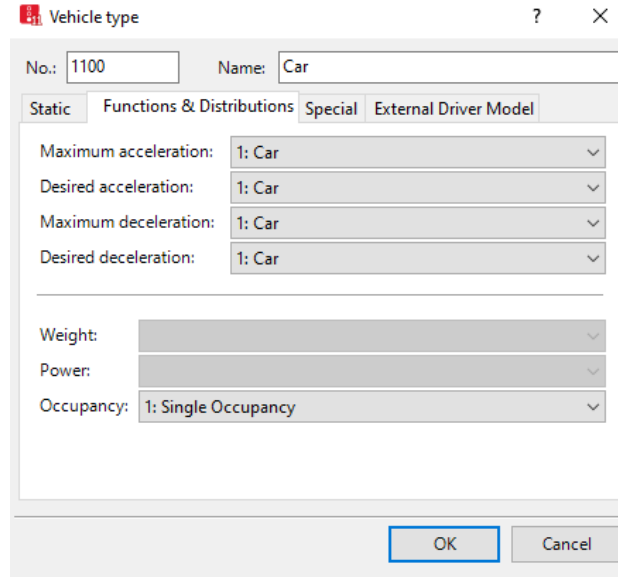


Figure 11: *Vehicle Type, Functions and Distributions*

The *maximum acceleration*, *desired acceleration*, *maximum deceleration*, and *desired deceleration* are all set to the *category car*. Using the pull-down menu associated with each of these you can set the *vehicle type* to follow the *acceleration* or *deceleration distribution* for any of the categories. We will leave them all set to the *Car category* for now. Later in this module we will learn how to open and adjust these underlying distributions.

Helpful Hints

In most typical model applications, you will not need to adjust the underlying *Vehicle Type* properties for *Static* or *Functions and Distributions*. However, in certain situations such adjustments may be necessary. It is critical that such adjustment be justified, seeking to match the characteristics of known field vehicles. For instance, matching a transit vehicles acceleration characteristic to those observed in the field.

Vehicle Classes

You can further group *vehicle types* into *Vehicle Classes*. A *vehicle class* may contain one or more *vehicle types*. Each default *vehicle class* will always contain a *vehicle type* of the same name. *Vehicle classes* are used to associate *vehicle types* to *desired speed decisions*, *routing decisions*, and *signal heads*, as well as numerous other model objects. (For instance, open the list for *desired speed decisions*. You will see these are assigned to *vehicle classes*, not *vehicle types*.)

Select *Base Data* → *Vehicle Classes*. You will see a Window similar to Figure 12. Look at the pulldown list under *VehTypes* for *10 Car*. You will see that only *100 Car*,

Model2D3DDistr - 10:Car NA is selected, the USA default car type. However, you could select additional vehicle types to add to this class or create a new class that contains numerous vehicle types. For example, in the left window, for the *10 Car* row, click on the *VehTypes* column. You should see a list of the *vehicle types* available. *VehType 100: Car* will already be selected. Go ahead and select *200: HGV*. You should now see the *HGV vehicle type* row in the right window as well. Thus, the *Vehicle Class - 10 Car* now includes *vehicle types 100 Car* and *200 HGV*. (Before exiting this list let's go ahead and unclick *HGV* from the *vehicle type* for *10 Car*, so this class will not include trucks.)

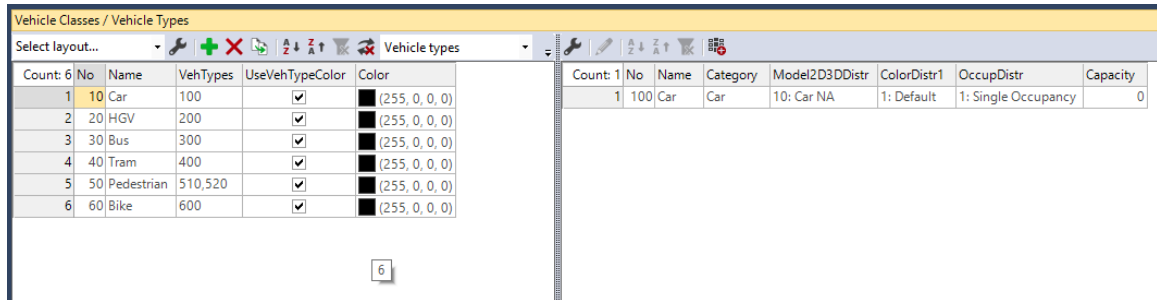


Figure 12: Vehicle Classes / Vehicle Types list

In our current model we always defaulted to *all vehicle types* when working with the various model objects (*detectors, routing decisions, etc.*) so we did not need to directly address *vehicle class*. However, by way of example let's consider *routing decisions*. Open the routing decision list. It should be similar to Figure 13. For one of the arterial routes uncheck *AllVehTypes*. The *VehClasses* box will become editable. You would now need to select the classes that will follow this routing decision.

For example, you may wish to model a bus route on the arterial. In all likelihood the bus route would not follow the current routing decisions. (That is, we don't want the busses to be randomly assigned as left, though, or right at each intersection based on the entered volume splits!). To handle the bus, you would create a route specific to the *Bus vehicle class* (this is one of the default classes, or you could always create your own transit class!) that contains the *Bus vehicle type*. You would then need to assigned all of the other vehicle classes to the existing arterial routes and the bus to the new *Bus* route.

Before continuing click *AllVehTypes* on any routes you may have unclicked, to return to the original route assignments.

Count: 28	No	Name	Link	Pos	AllVehTypes	VehClasses	RouteChoiceMeth
1	1	Buzz and Burdell, EB App	1: Buzz Blvd EB	83.499	<input checked="" type="checkbox"/>		Static
2	2	Buzz and Burdell, WB App	2: Buzz Blvd WB	2738.016	<input checked="" type="checkbox"/>		Static
3	3	Buzz and Burdell, NB App	4: Burdell Dr. NB	598.818	<input checked="" type="checkbox"/>		Static
4	4	Buzz and Burdell, SB App	3: Burdell Dr. SB	100.468	<input checked="" type="checkbox"/>		Static
5	5	Buzz and Ramblin, EB App	1: Buzz Blvd EB	644.896	<input checked="" type="checkbox"/>		Static
6	6	Buzz and Ramblin, WB App	2: Buzz Blvd WB	2141.863	<input checked="" type="checkbox"/>		Static
7	7	Buzz and Ramblin, NB App	6: Ramblin St. NB	147.484	<input checked="" type="checkbox"/>		Static
8	8	Buzz and Ramblin, SB App	5: Ramblin St. SB	97.876	<input checked="" type="checkbox"/>		Static
9	9	Buzz and Pi, EB App	1: Buzz Blvd EB	1180.467	<input checked="" type="checkbox"/>		Static
10	10	Buzz and Pi, WB App	2: Buzz Blvd WB	1768.511	<input checked="" type="checkbox"/>		Static
11	11	Buzz and Pi, NB App	8: Pi Pkwy NB	150.740	<input checked="" type="checkbox"/>		Static
12	12	Buzz and Pi, EB App	7: Pi Pkwy SB	74.541	<input checked="" type="checkbox"/>		Static

Figure 13: Routing Decision list

CRITICAL CONCEPTS

The method described above to address bus routing is somewhat brute force in that we would not use *AllVehTypes* for any of the routes, specifically assigning the correct vehicle classes to each *routing decision*. Alternatively, you could leave the existing routes assigned to *AllVehTypes* and only assign the bus class to the new bus *routing decision* if you ensure that the bus crosses a bus *routing decision* before crossing any of the existing *routing decision* points. The bus must never cross an existing *routing decision* when it is not already assigned to a bus *routing decision*. In this approach we are taking advantage of the property that a vehicle will ignore all *routing decisions* it crosses while assigned to a route. Remember, all other vehicles will ignore the bus routing decision as their vehicle class is not assigned to that route.

The takeaway from this discussion is that there are often multiple ways to achieve the same goal in Vissim; however, each method will have its own potential benefits and drawbacks. When reviewing or debugging a model it is critical to know the modeling methods that have been implemented so you can understand expected behaviors and potential sources of error.

Helpful Hints

A helpful item in *vehicle class* is that you can assign a specific color to a *vehicle class*. To do this in the *Vehicle Classes / Vehicle Types list* (Figure 13) uncheck *UseVehTypeColor* and select the desired *Color*. This will result in model using the color you selected rather than the color under *vehicle type*, which could be different for each *2D/3D model* in the fleet. Having all vehicles in a class the same color can be extremely useful with watching a simulation animation and seeking to observe the behavior of a specific *vehicle class*.

For example, in the *Vehicle Classes / Vehicle Types list* window deselect *UseVehTypeColor* and change the color of 10 *Car* to red and 20 *HGV* to yellow, as in the below image. (To do this double click in the *Color* column for the correct row and select the *Color* you want from the color wheel.) Run the model. You should now be able to see the *HGVs* as yellow and the *Cars* as red!



Count: 6	No	Name	VehTypes	UseVehTypeColor	Color
1	10	Car	100	<input type="checkbox"/>	■ (255, 241, ...)
2	20	HGV	200	<input type="checkbox"/>	■ (255, 243, ...)
3	30	Bus	300	<input checked="" type="checkbox"/>	■ (255, 0, 0, 0)
4	40	Tram	400	<input checked="" type="checkbox"/>	■ (255, 0, 0, 0)

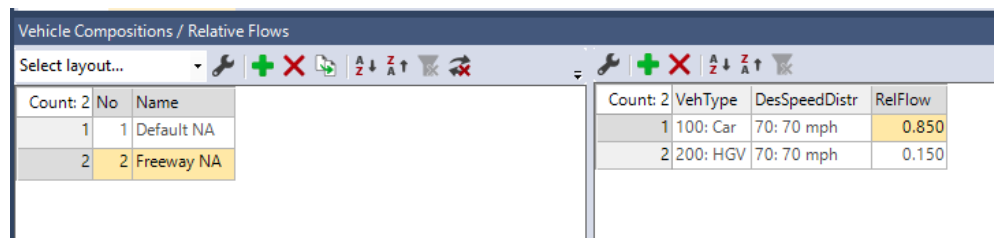
When you are done reselect the *UseVehTypeColor* checkbox if you want to return to the original color scheme. (You don't need to change the *Color* column back to black as this will be ignored while *UseVehTypeColor* is selected.)

Distribution of Vehicle Types within Vehicle Composition – Relative Flow

A critical item that we have not yet considered is distribution of *vehicle types* within a *composition*. Go to *Traffic* → *Vehicle Compositions*. Select *Default NA*. *RelFlow* defines the percentage of the total composition assigned to each *vehicle type*. Thus, for the arterial (we are using *Default NA* for the arterial inputs) we have currently assigned (by default) 90 percent of the vehicles as type *Car* and 10 percent as *HGV*. As the percent of each *vehicle type* in the traffic stream can significantly influence traffic it is critical that you assign the correct values to the vehicle composition.

Currently all *Vehicle Inputs* utilize the *Default NA composition*. However, let's assume you need a different ratio of cars and trucks on the freeway. To do this we need to create a new *Vehicle Composition* that we may then assign to the freeway *Vehicle Inputs*.

- step i. Go to the *Vehicle Compositions* list (if you are not already there).
- step ii. Click the green plus  (above the left side window). *Name* the new *composition* Freeway NA. In the right window the *vehType* should be *100 Car*. Change the speed to 70 mph.
- step iii. In the right side of the window select the . The new row should have *VehType* to *200: HGV*. Confirm the speed is 70 mph.
- step iv. Change the *RefFlow* to 0.85 for *100: Car* and 0.15 for *200: HGV*. The composition should now look similar to Figure 14.



Count	No	Name
1	1	Default NA
2	2	Freeway NA

Count	VehType	DesSpeedDistr	RelFlow
1	100: Car	70: 70 mph	0.850
2	200: HGV	70: 70 mph	0.150

Figure 14: *Vehicle Compositions Window with new Freeway NA*

- step v. Go to the *Vehicle Inputs* Window (in *Network Objects* menu) and update the *Vehicle Composition (VehComp)* for the Peach Freeway inputs to the Freeway NA, *Figure 15*.

Vehicle Inputs / Vehicle Volumes By Time Interval					
Count	No	Name	Link	Volume(0)	VehComp(0)
11	1	Buzz Blvd, EB inp	1: Buzz Blvd EB	300.0	1: Default NA
2	3	Burdell Dr., SB inp	3: Burdell Dr. SB	300.0	1: Default NA
3	4	Ramblin St., NB inp	6: Ramblin St. NB	300.0	1: Default NA
4	5	Ramblin St., SB inp	5: Ramblin St. SB	300.0	1: Default NA
5	6	Pi Pkwy, NB inp	8: Pi Pkwy NB	300.0	1: Default NA
6	7	Pi Pkwy, SB inp	7: Pi Pkwy SB	300.0	1: Default NA
7	8	Buzz Blvd, WB inp	2: Buzz Blvd WB	600.0	1: Default NA
8	9	Varsity Way EB	11: Varsity Way EB	150.0	1: Default NA
9	10	Varsity Way WB	12: Varsity Way WB	150.0	1: Default NA
10	11	Peach Freeway SB	13: Peach Freeway SB	1500.0	2: Freeway NA
11	12	Peach Freeway NB	14: Peach Freeway NB	1500.0	2: Freeway NA

Figure 15: Updated Vehicle Inputs list

Through the preceding steps you have added a new vehicle composition consisting of 85% *Vehicle Type Car* and 15% *Vehicle Type HGV*. Notice, vehicles are assigned to a *Vehicle Composition* by *Vehicle Type*, rather than by *Vehicle Class* as with *Desired Speed Decisions*. Also, the Freeway composition utilizes a *DesSpeedDistr* of 70 mph. This means the *Desired Speed Decision* points that have been placed at the freeway inputs are no longer necessary. You could delete them if you wish, for now we will leave them.

Acceleration and Deceleration Functions

Previously when considering *Vehicle Types* we saw that each *vehicle type* is assigned a *maximum acceleration, desired acceleration, maximum deceleration, and desired deceleration* based on a *Vehicle Category*. In this section we will see how to review and edit these distributions.

Go to *Base Data* → *Functions* → *Maximum Acceleration*. You should see a window similar to *Figure 16* below. In the window you will see the distribution of *Maximum Acceleration* for each vehicle category.

Maximum Acceleration Functions / Data Points						
Count	No	Name	X	Y	yMin	yMax
6	1	Car	0.00	11.48	6.43	11.48
2	2	HGV	6.21	10.50	4.90	11.48
3	3	Bus	12.43	9.14	4.27	11.48
4	4	Tram	18.64	8.10	3.78	11.48
5	5	Pedestrian	24.85	7.22	3.37	11.48
6	6	Bike	31.07	6.44	3.01	10.74
7	7		37.28	5.74	2.68	9.57
8	8		43.50	5.10	2.38	8.50
9	9		49.71	4.50	2.10	7.50
10	10		55.92	3.94	1.84	6.56
11	11		62.14	3.41	1.59	5.68

Figure 16: Maximum Acceleration parameters.

From this table we can read that, when accelerating from a stop (i.e., a speed of 0 mph, the value in the X column), a vehicle in the *category Car* will be assigned a *maximum acceleration* between 6.43 ft/s^2 ($yMin$) to 11.48 ft/s^2 ($yMax$), with a distribution median of 11.48 ft/s^2 (Y). Likewise, a vehicle traveling at 6.2 mph will be assigned a maximum acceleration between 4.9 ft/s^2 and 11.48 ft/s^2 , with a distribution median of 10.50 ft/sec^2 . To get a better sense of the data, double click on row 1 for *Car* in the left window. You should see the pop-up window in *Figure 17*. This is a graphical form of the table on the right side of the *Maximum Acceleration* window.

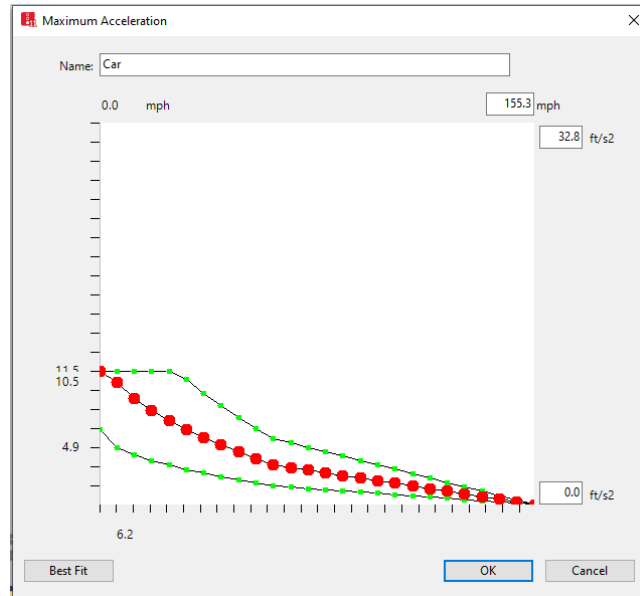


Figure 17: Car Maximum Acceleration, acceleration range and median vs vehicle speed

The green lines represent the minimum and maximum values of *maximum acceleration* that will be assigned, while the red dots are the median value. Every *Car* in the model will be assigned its maximum accelerations from this distribution. Similarly, if you open the *desired acceleration*, *maximum deceleration*, and *desired deceleration functions* you may see the distributions used for these parameters.

While beyond this module, the exact application of maximum vs desired acceleration (or deceleration) is discussed in detail in the Vissim manual. Loosely, the maximum represents the maximum possible but in many of the typical situations the use of desired is preferred. *Car* maximum acceleration and decelerations are also adjusted for gradient. *HGV* and other vehicle types also have other specific difference that users are encouraged to review in the manual if this behavior is critical to your results interpretation.

Helpful Hints

A modeler may change the values directly in the table or by selecting one of the points in the graph and dragging it to the new desired values. (Go ahead and try it!) At times you may find it necessary to adjust the acceleration and deceleration parameters. However, such adjustments must be justified and clearly documented as they can significantly impact model results. These parameters should not be changed simply to obtain desired results. For instance, a strong temptation may exist to increase accelerations to improve performance at a merge location. However, such a change has numerous drawbacks, such as the vehicle behavior may not be realistic and the acceleration change is applied *everywhere* in the model. Likely, a better model would be achieved by first checking for errors (always start with error checking!), then considering site specific calibrations such as adjusting lane change and emergency stop distances.

CRITICAL CONCEPTS

Why are there distributions for *maximum acceleration*, *desired acceleration*, etc. rather than fixed values?

We will see later on that when running Vissim we must set a “random seed”. (We are currently using the default). If we change this seed, and nothing else in the model, we will get different model results. These differences, in part, come from randomness introduced into the model through parameters such as *maximum acceleration*. When we change the random seed value the vehicles will be randomly (according to the parameters of the distribution) assigned new maximum acceleration parameters. By trying different random seeds, we can explore the impact of different user and vehicle characteristics on same relative traffic. One rough way to think of this is that even while different days on a roadway may have similar volumes (and weather conditions, lighting, etc.) we might find different travel times, delays, queues, etc. This occurs, in part, because on these different days the vehicles may be different, drivers may be different, the order of vehicles may differ, etc. From one day to the next even the same driver may exhibit different levels of aggressiveness, inattention, etc. That is, there is randomness in the real-world system! By using distributions to allow some randomness and variability it becomes possible for the simulation to capture, at least somewhat, the variability we see in the real world. Throughout Vissim there are a number of “random” choices, such as setting a vehicle’s *maximum acceleration* or the selection of which vehicle will turn at an intersection that allow the simulation to reflect randomness. However, it is important to notice that this variability is *not* due to changes in day-to-day volume, but changes in the characteristics of the same absolute volume.

As a side note: It is acknowledged that the critical concept text above is playing loose with the term “random”. There is a lot that could be unpacked here. The intent of the preceding is to give an intuitive sense in a single paragraph of stochastic transportation simulation to people without extensive statistical backgrounds. Those with further interest are encouraged to seek out some of the many references on stochastic simulation. While not a Vissim text one of my go to references is *Simulation Modeling and Analysis* by Averill M. Law.

Desired Speed Distributions.

In a manner similar to the *acceleration* and *deceleration distributions*, in the previous sections we have assigned *desired speed decision* distributions. Desired speed is one of the more critical parameters utilized by Vissim, influencing travel time, delay, flow rates, capacity, etc. The desired speed of a vehicle is the speed at which a vehicle will travel at free flow, that is, when a vehicle’s speed is not influenced by neighboring vehicles. Desired speed should represent the speed a vehicle would travel on the roadway in uncongested conditions. In many instances this may be represented by the measured 85% speed on the roadway.

Helpful Hints

There is a long philosophical discussion beyond the scope of this module that could be had whether desired speed should be allowed to be set higher than the speed limit. Regardless of the decision it is important that the model developer, supporting agency, and any others involved in the process come to agreement on setting of desired speed.

To see a *Desired Speed Distribution*, go to *Base Data* → *Distributions* → *Desired Speed*. The *Desired Speed Distributions* list should open. Scroll down and double click on *30 mph*. You should see a window similar to *Figure 18*. *Figure 18* represents the *desired speed distribution* for vehicles assigned a *desired speed* of 30 mph. Earlier we saw that *desired speed* is assigned in *Vehicle Compositions* and updated using *Desired Speed Decision* points.

Similar to the *acceleration* discussion each vehicle is assigned a *desired speed* from the given *desired speed distribution*. By default, the *desired speed distributions* are linear (*Figure 18*), which means the selection of desired speeds will be uniform across the given range. Thus, a vehicle from a *Vehicle Composition* with a *desired speed* of 30 mph will be

assigned a desired speed between 29 mph and 33 mph, with equal likelihood of any value being selected.

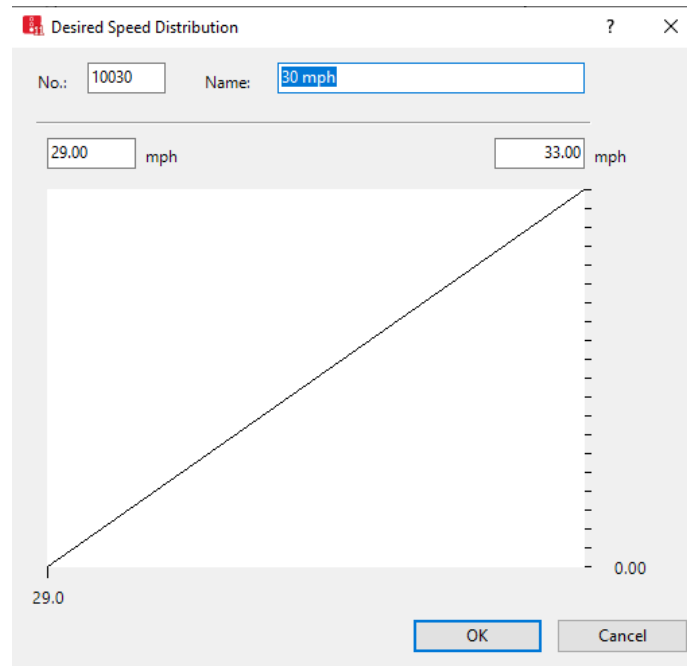


Figure 18: Default Desired Speed Distribution of 30 mph

A user may change the maximum and minimum values of the *desired speed distribution* as well as change the shape of the distribution. For example, on the open *30 mph desired speed* window right click on the line at approximately one-third and two-thirds the distance from the left side of the graph. Red dots should appear at these points. Slide each of the dots up a bit. Your distribution should now look similar to Figure 19. In this example we are creating a distribution where the lower speeds represent a higher proportion of the desired speeds. (That is, the figure represents the cdf of the desired speeds.) If you don't wish to keep this edit hit cancel on the *Desired Speed Distribution* window.

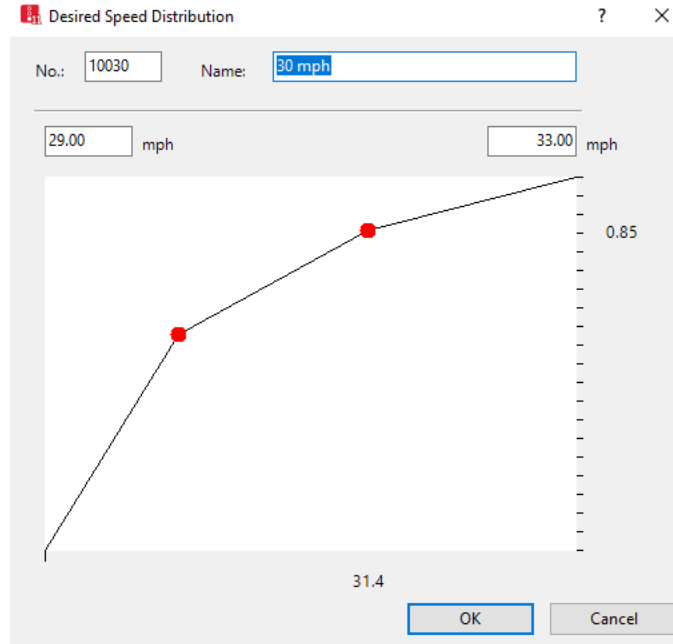


Figure 19: Adjusted 30 mph Desired Speed Distribution

CRITICAL CONCEPTS

The *desired speed distribution* underlies the performance of the model and can dramatically influence results. While it may be reasonable to adjust *desired speed distributions*, such changes should not be taken lightly. It is best that changes to the *desired speed distributions* be based on speed data collected in the field. Additionally, this data should be collected such that it represents the free flow speed of vehicles. **YOU CAN NOT USE SPEED DATA DURING CONGESTED PERIODS TO SET THE DESIRED SPEED DISTRIBUTION!**

Also, it is critical when comparing between scenarios, e.g., no-build vs build, that all scenarios utilize the same desired speed distributions, where a roadway has the same speed limit (or other measure of expected free flow speed) between scenarios. If *desired speed distributions* are altered between scenarios you will not know if model performance is based on geometric, signal, etc. changes between scenarios or due to changes in the underlying *desired speed distribution*. According to the old colloquialism, you would be “comparing apples and oranges”. This guidance is generally true for most of the underlying distributions utilized by Vissim. Changing of any underlying distributions between scenarios must be justified. For example, if part of a build scenario implements a higher or lower speed limit than changing the *desired speed distribution* is clearly justified.

As a final *desired speed* discussion point we will consider the performance measure delay. The vehicle delay measured and reported by Vissim is based on the given vehicle's desired speed. For example, consider two vehicles traveling as a platoon at a speed of 55 mph. If both vehicles have a desired speed of 55 mph then no delay will be recorded. If the first vehicle has a desired speed of 55 mph and the second has a desired speed of 58 mph, the second vehicle will accrue delay to reflect that it is traveling slower than its desired speed. Thus, the first scenario has no delay and the second has a positive delay, even though the vehicle behavior is exactly the same in both cases! While we may debate which delay is "correct" what is important in this context is to know this is how Vissim calculates delay. One method to avoid this issue is to use travel time rather than delay, then where delay is needed post-process the actual-travel-time minus ideal-travel-time to determine delay.

Reduced Speed Areas

We have seen how a vehicle's *desired speed* is initially set (*Vehicle Composition*) and how it may be updated (*Desired Speed Decision* point) as the vehicle traverses the model. However, there may be instances where it is desirable to slow a vehicle over a short section of roadway, and then the vehicle returns to its original speed. For instance, to reflect a lower speed around a curve, such as a loop ramp or intersection turn movement. One way to accomplish this speed reduction could be using a pair of *desired speed decision* points, the first upstream of the roadway segment to lower the speed and the second immediately after to return the desired speed to the original value. However, a better approach is often to use a *Reduced Speed Area*. A *reduced speed area* will cause a vehicle to reduce its speed only in the vicinity of the zone and then return to its original speed.

To demonstrate, let's add a *reduced speed zone* to a right turn on Buzz Blvd.

- step i. Zoom in on the intersection of Buzz Blvd. and Burdell Dr.
- step ii. Select *Reduced Speed Area* in the *Network Objects* Window on the left side of the Vissim window.
- step iii. Select the *connector* for the NB right turn (i.e., from Burdell Dr. NB to Buzz Blvd. EB).
- step iv. Right click on the *connector* a few feet after it diverges from the Burdell Dr. link. Holding the right click button down drag the mouse to a few feet prior to the merge point with Buzz Blvd. You should see a box along the *connector* and the *Reduced Speed Area* pop-up window, Figure 20.

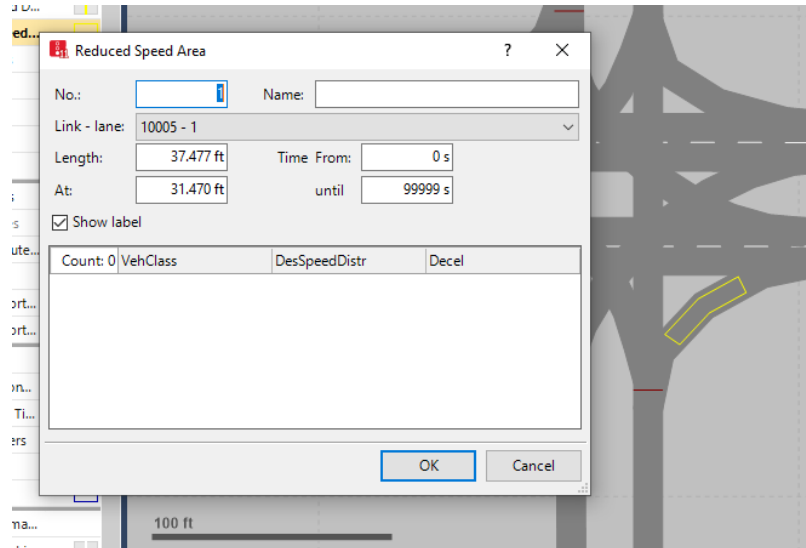


Figure 20: Buzz Blvd and Burdell Dr. NB right turn Reduced Speed Area

- step v. In the *Reduced Speed Area* pop-up window edit *Name* as Buzz Dr. and Burdell Blvd NB RT RSA.
- step vi. Add: *VehClass* to 10 Car and the *DesSpeedDistr* to 15 mph, Figure 21.
- step vii. Add: *VehClass* to 20 HGV and the *DesSpeedDistr* to 15 mph, Figure 21.
- step viii. Select *OK*.

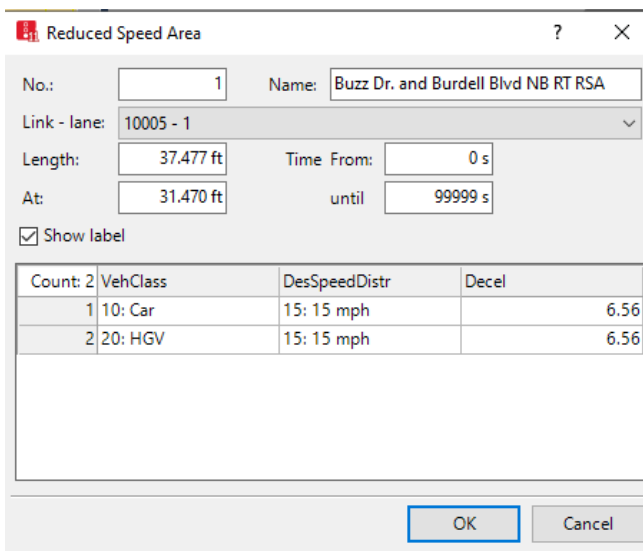


Figure 21: Buzz Blvd and Burdell Dr. NB right turn Reduced Speed Area pop-Up Window

Run the model. You should see that vehicles traversing the right turn are now slowing. We have selected 15 mph for the *Reduced Speed Area*; however, if the radii were lower (sharper) you could select a lower speed.

Repeat Steps 1 through 8 for the right turns on the remaining intersections.

While we will not worry about it for this model you could also add reduced speed areas to the left turns or any other geometric feature that would result in lower speeds in the field.

CRITICAL CONCEPTS

Desired Speed Decision Point vs Reduced Speed Area

There is a subtle but significant difference between how vehicles react to *desired speed decision points* and *reduced speed areas*. As we have seen, for *desired speed decision points* the vehicle will not begin changing speeds until it reaches the decision point. For *reduced speed areas* the vehicle will begin reducing speed upstream of the area such that it will be at the lower speed when it reaches the start of the area. When placing *desired speed decision points* and *reduced speed areas* it is important that you consider this difference in behavior when selecting placement locations. A nice trick can be to place a short *reduced speed area* immediately upstream of a *desired speed decision*. This will result in vehicle reducing speed prior to reaching the desired speed decision. This can be particularly handy in roundabout, loop ramp, etc., designs.

Impact of Geometry on Speed

In the discussion it was stated that a *reduced speed area* could be used for any geometric feature that would influence speeds. Don't miss the implication of this statement! Curve radius, by itself, does not influence the speed of vehicles in Vissim. For example, if a vehicle is traveling at 45 mph on arterial and it takes a sharp right turn it will not slow for the turn, it will maintain 45mph!!! Likewise, a vehicle with a 70 mph desired speed will maintain that speed on a loop ramp, regardless of the ramp radii. It is critical that speed reductions needed for geometric features be addressed through either *desired speed decision points* or *reduced speed areas*. This is a particularly significant issue in network locations where speed reductions due to a geometric feature may result in a bottleneck. This bottleneck would not be reflected in the model if *desired speed decision points* or *reduced speed areas* are not appropriately utilized. This is a common issue in Freeway modeling.

Alternate Intersection Link-Connector Layout

So far, we have modeled intersections by using a single *link* that extends through the intersection for through movements and *connectors* to model the tuning movements. However, a common modeling approach is to use *connectors* for all movements through the intersection, including the through movements. For example, *Figure 22(a)* shows the current layout of Buzz Blvd. and Burdell Dr. and *Figure 22(b)* show the layout with all movements using a *connector*.

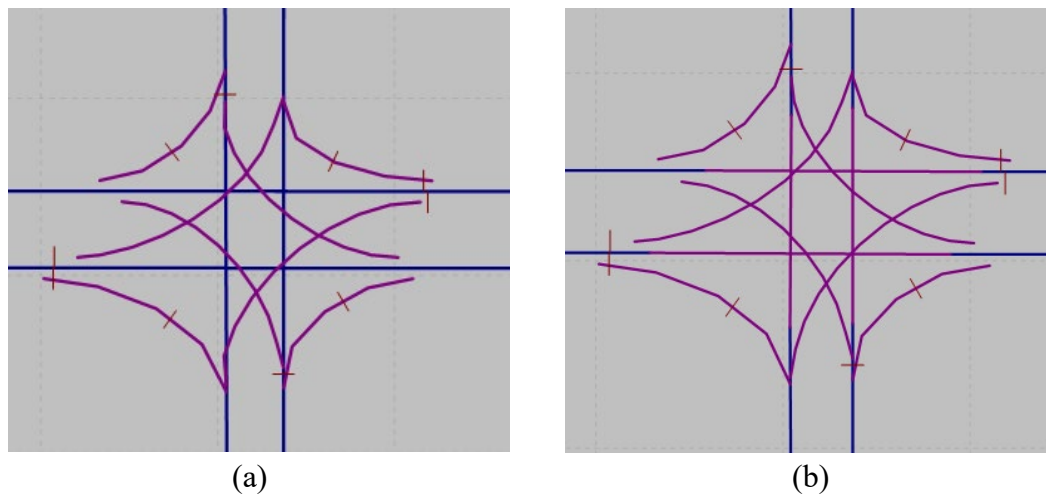


Figure 22: Buzz Blvd and Burdell Dr. (a) with continuous links for through movements (b) with connectors for through movements.

To implement this alternate design at Buzz Blvd and Burdell Dr.

- step i. Split the EB *link* at the stop bar location. Be sure the split location is downstream of the existing left and right turn *connectors*. To do this:
- Left click to select the *link*.
 - Press Ctrl-Right-Click at the location you wish to split the *link*.
 - Select *Split Link Here.... Figure 23*.

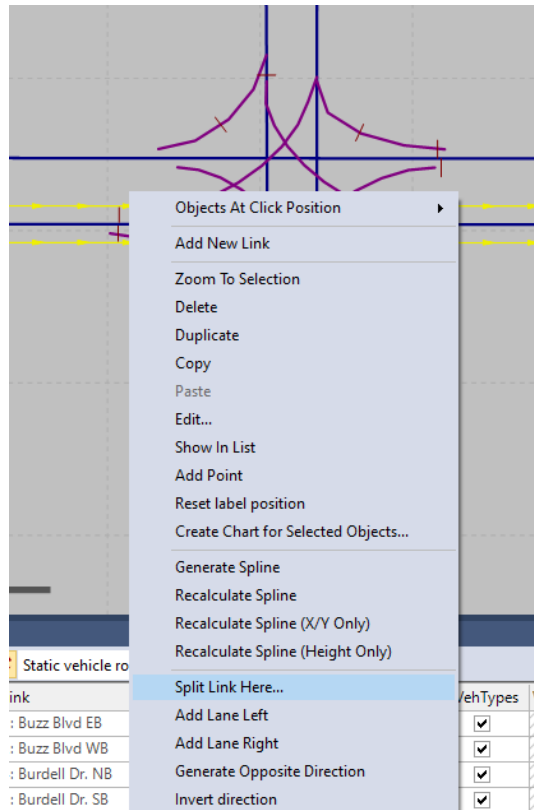


Figure 23: Split Link function

- step ii. The *Split link* pop-up window should be visible, Figure 24. Select *OK*. Vissim will split the *link* and place a short *connector* at that location. (The *Generate Automatically* check box under *Connector* in the *Split Link* window will be selected by default).

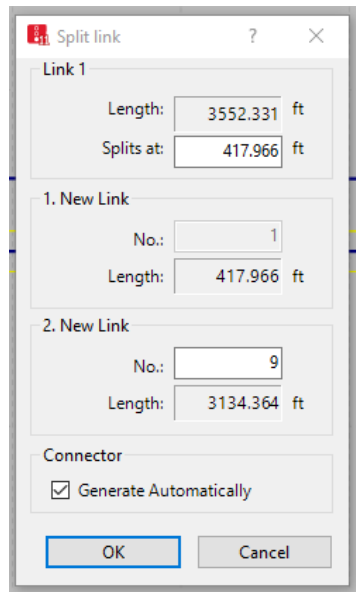


Figure 24: Split Link pop-up window

- step iii. Select the downstream *link*. (Be sure you have link selected in the *Network Objects* menu, otherwise the *link* endpoint will not highlight.) Drag the beginning of the *link* to the far side of the intersection. Be sure the new *link* start position is upstream of the SB left and NB right turn *connectors*. The *connector* will automatically lengthen. See *Figure 25*. When dragging the *link* it is critical that you are shortening it and not moving the entire *link*. If you slide the link instead of shortening it the connections to *connectors* along the *link* will be moved as well, creating unrealistic traffic movements.

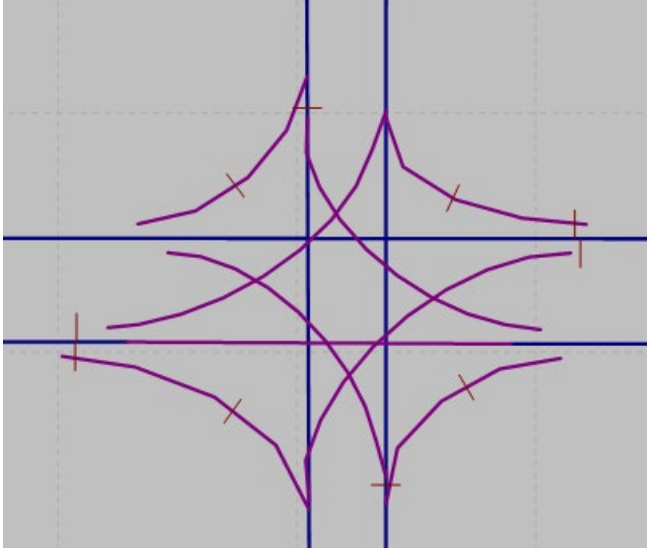


Figure 25: Buzz Blvd. and Burdell Dr. with EB Through Movement Connector

A common potential source of error when splitting *links* is accounting for the location of the existing *connectors* to the *link* being split. For instance, if you split the *link* upstream of the EB left turn *connector* then the *connector* will be tied to the *link* downstream of the split. When you drag the start of the downstream *link* to the far side of the intersection this will also drag the EB left *connector*, thus your *connector* will be connected to the wrong side of the intersection! If this happens you need to delete the EB left *connector* and replace it with one that connects to the *link* upstream of the through *connector*. **After splitting a *link* always make sure that your *connectors* remain properly assigned.**

- step iv. Repeat Steps 1 through 3 for the WB, NB, and SB *links*, *Figure 26*.

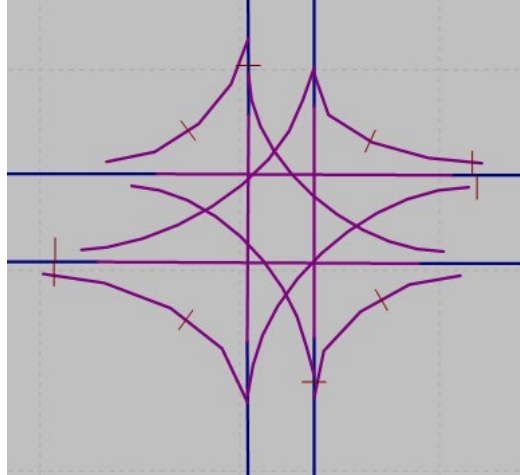


Figure 26: Buzz Blvd and Burdell Dr. with through movement connectors

step v. After splitting *links* it is good practice to make sure the routes are still correct. Particularly if you had to delete and replace a *connector*, there is a high likelihood that a *vehicle route* may have been changed or deleted. There should be routes for the left turn, though, and right turn movements for EB, WB, NB, and SB. Check that each are still correct. (If you don't recall how to check routes please see [Module 1](#).)

step vi. By splitting the *links* and adding a *connector* you deleted the existing *conflict areas*. The only *conflict areas* that remain are those where the *links* or *connectors* in the conflict were not altered by the split, in our case this will be the turn movement *connectors* with the through *links*, *Figure 27*. You need to confirm you have the following conflicts, adding in those that are no longer present:

- Four crossing conflicts, one for each of the left turn conflicting with opposing through movement. The through movements receive priority.
- Two merging conflicts, one for each left turn and opposing right turn movements entering into the same lane of the NB and SB receiving lanes, give right turn priority. (There is not a conflict on the EB and WB approaches as the left and right turns are received by different lanes.)
- Four merging conflicts, one for each through movement with the merging right turn on red, give through movement priority.

(Recall, we are not concerned with all potential conflicts, such as the crossing through movements, as these are addressed by the traffic signal control.)

The final *conflict areas* should look similar to *Figure 27(b)*. (If you do not recall how to set the *conflict areas* see [Module 1](#).)

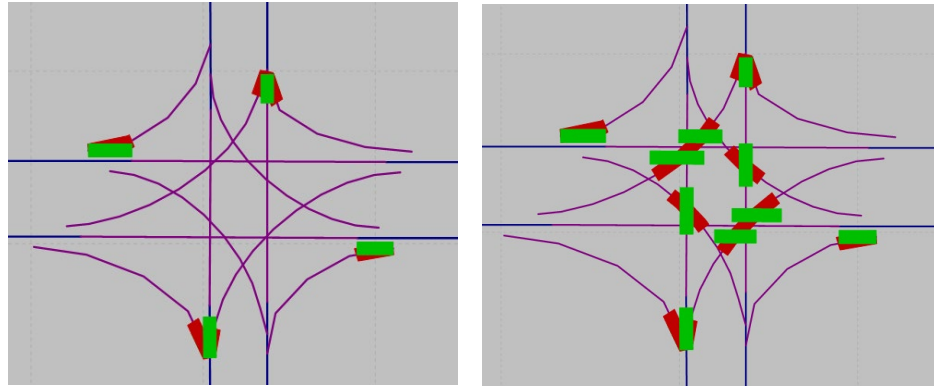


Figure 27: Buzz Blvd. and Burdell Dr. Conflict Areas (a) after splitting links, (b) updated to reflect all conflicts

Alternate Intersection Left Turn Configuration

In addition to utilizing *connectors* for through movements there is an alternate design that may be used for left turn bays, where the through movement and left turn bay lanes are on the same *link*. To demonstrate this, we will convert the intersection of Buzz Blvd and Ramblin St. to use *connectors* for all through movements and combine the EB and WB left turn bay and through lane *links*.

step i. Delete the EB left turn lane *link*: i.e., Ctrl-Right-Click on the *link* then select *Delete*, Figure 28.

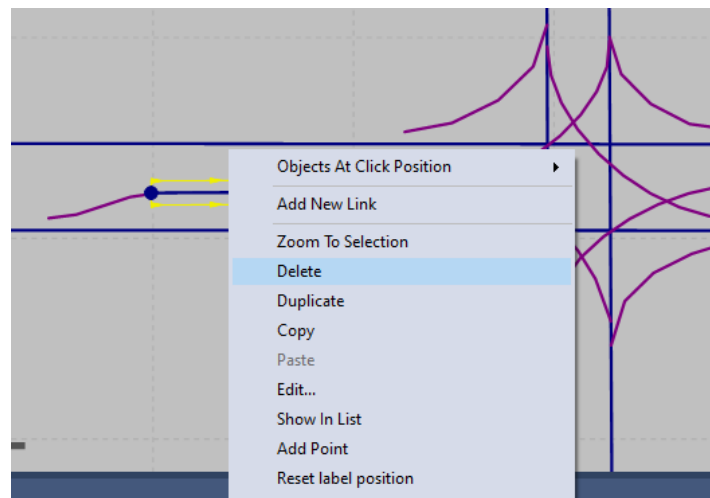


Figure 28: Delete Left Turn Bay Link

step ii. Select *Continue* in the *Delete Link* pop-up window. You will see in the pop-up that you will also be deleting the associated *connectors* and *signal head*, Figure 29.

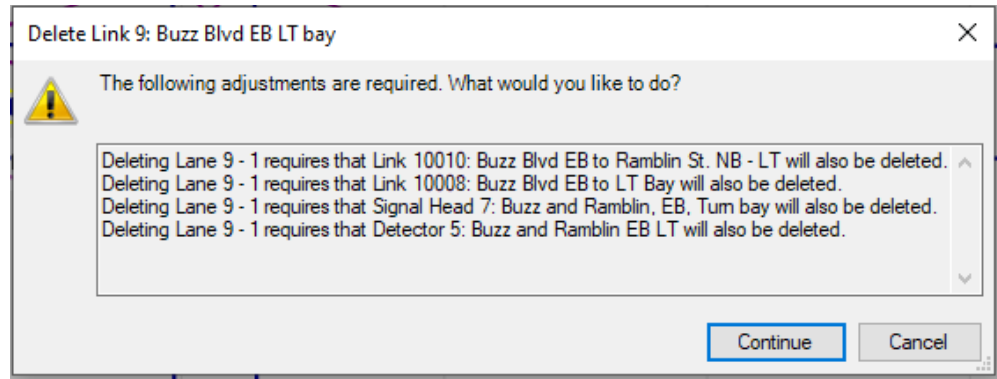


Figure 29: Delete Link pop-up window

- step iii. In a similar manner delete the WB left turn bay *link*. Your intersection should now look similar to *Figure 30*. Notice, the EB and WB left *connectors* have been deleted as well. We will replace these in a subsequent step.

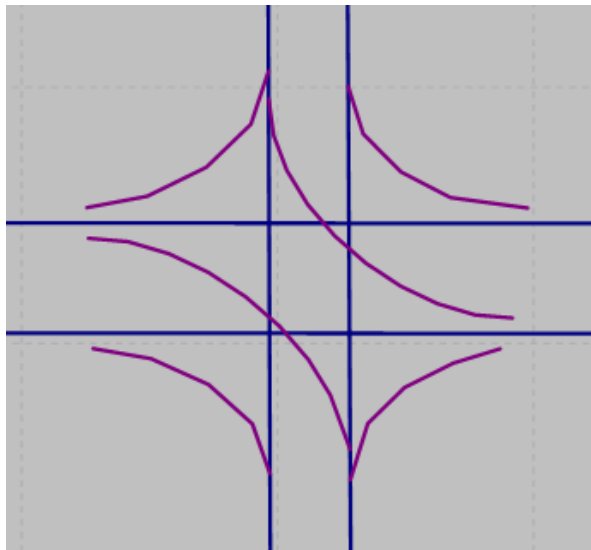


Figure 30: Buzz Blvd. and Ramblin St. with left turn bay links deleted

- step iv. In a manner similar to the previous Buzz Blvd. and Burdell Dr. example split all of the through *links* (EB, WB, NB, and SB) and use *connectors* for the through movements. Your intersection should be similar to *Figure 31* after this step.

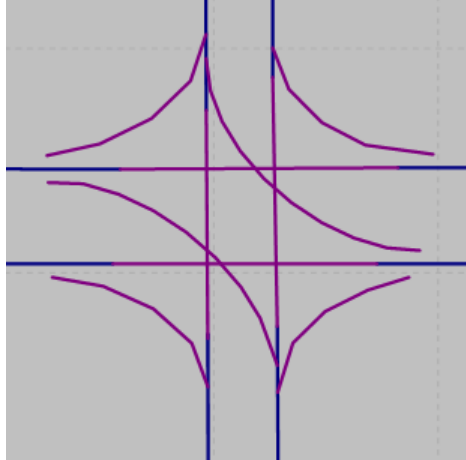


Figure 31: Buzz Blvd and Ramblin St. with through movement connectors

step v. Split EB *link* approximately where the left turn bay would start. Select *OK* in the *Split Link Pop-Up* window, Figure: 32. We will leave the new *connector* at its default length (approximately 3 ft.).

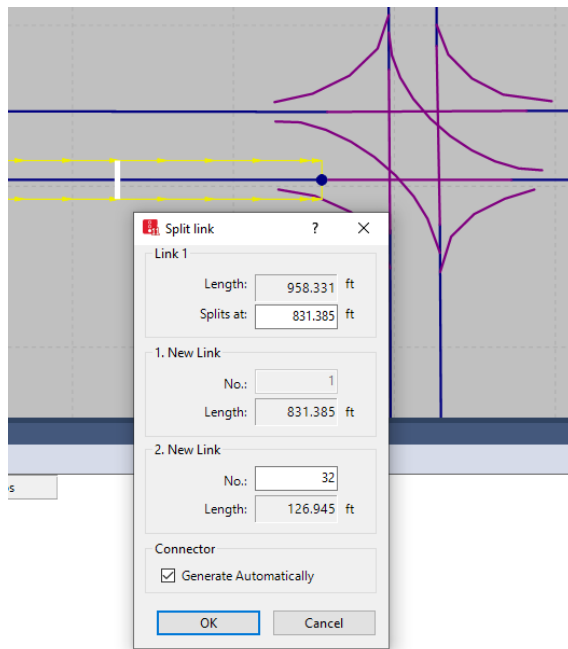


Figure: 32 Split Link pop-up window

step vi. Now we will add the left turn bay lane to the *link* immediately upstream of the intersection. Ctrl- Right-Click on the *link* then select *add lane left*, Figure 33. The EB approach should now look similar to Figure 34(a) (*link* view), (b) lane view. (It is a good idea to double click on the EB *connector* for the intersection through movement and confirm that the *from link* lanes are identified as *lane 1* and *lane 2*. This should be correct if you used *add lane left*, Figure 35.)

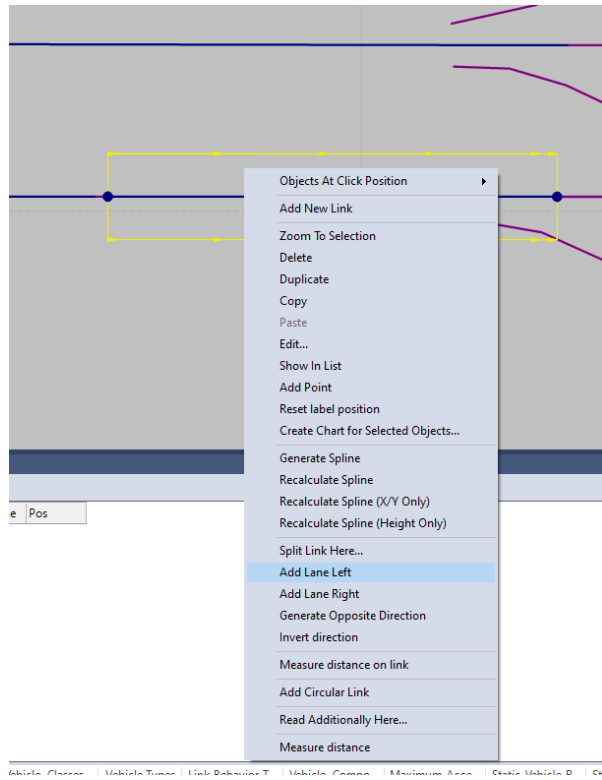
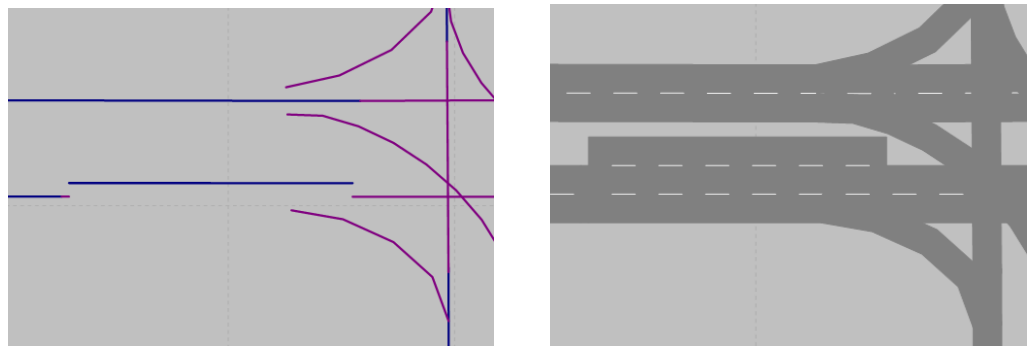


Figure 33: Add Lane Left



(a)

(b)

Figure 34: Addition of Lane for Left Turn Bay: (a) Link Diagram and (b) Lane Diagram

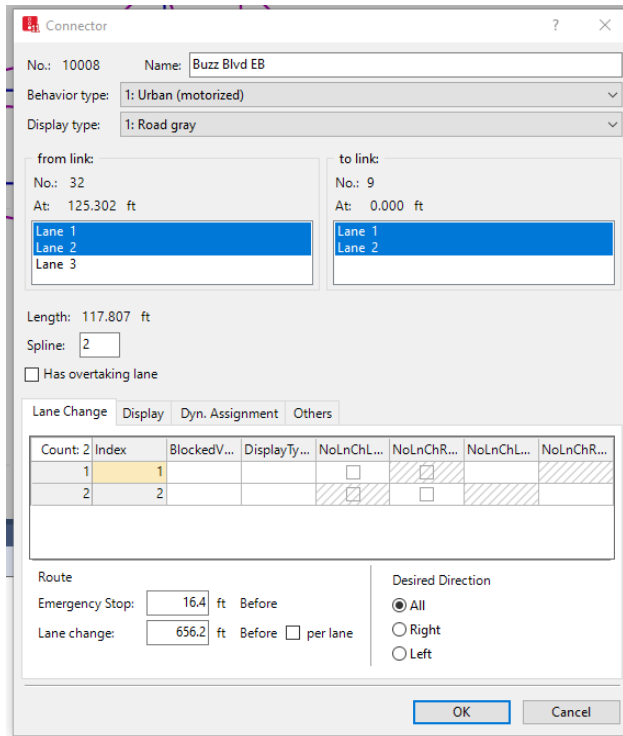


Figure 35: EB Through Movement Connector Window

step vii. Add the EB left turn connector, confirm from link is Lane 3 and to link is lane 1, update the spline to 10 and Name to Buzz Blvd. and Ramblin St. WB Left, Figure 36.

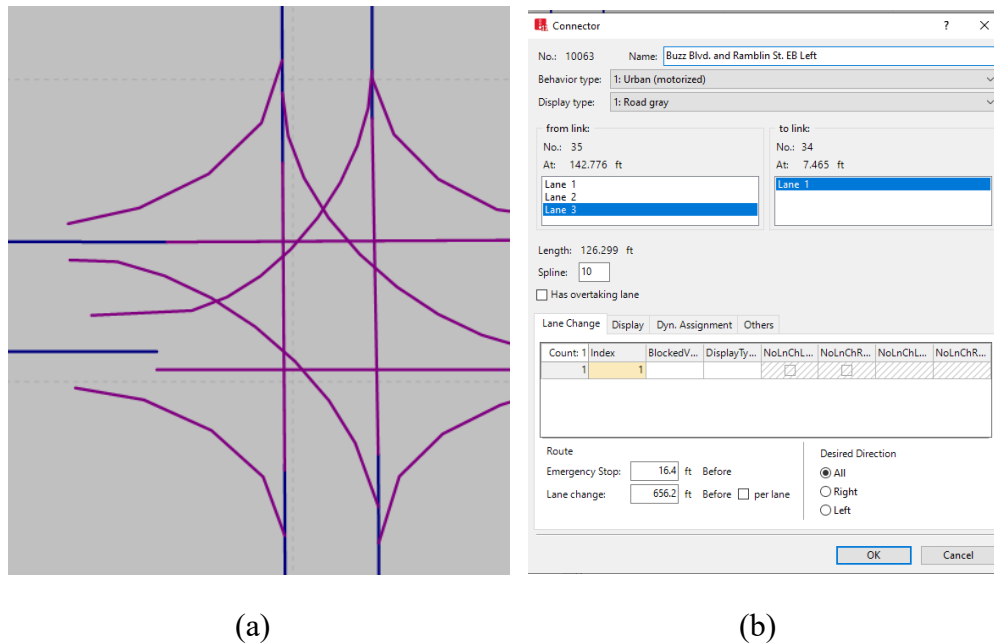


Figure 36: EB left turn (a) Connector Placement and (b) Connector pop up Window

- step viii. Go to the *vehicle route* for the EB approach to Buzz Blvd. and Ramblin Dr. You should find the right turn and through movement routes should be ok. If you check the left turn route it will be incorrect, likely indicating some strange path through the network. This is because when we deleted the turn bay *link* we deleted the path for the left turn. Vissim is attempting to find another path through the network. If you select the destination point of the EB left turn movement (indicated by a green line at the end of the route on Ramblin St.) and move it slightly this will cause VISSIM to recalculate the route. It should now consider the new *connector* we added and provide the correct route. (If you are having trouble resetting the left-turn route go ahead and delete it, select the route starting bar, and drag a new left turn route.) Check that the *RelFlow* values remain correct.
- step ix. When we deleted the left turn bay *link* we also deleted the left turn *signal head*. Select the *Signal Heads* icon in the *Network Objects* Menu and add a new *signal head* on the left turn *connector* (i.e., with the *Signal Heads* Network Object selected right click on the correct location on the *connector*). Set the *SC – Signal Group* to 222-5 and the *Name* to Buzz and Ramblin, EB, Lane 3 LT. *Figure 37*.

It is important that the *signal head* be placed on the *connector*. If placed on lane 3 of the *link* it is possible that a vehicle could try to “cheat”, avoiding the left turn *signal head* by staying in the through lane until it is past the left turn *signal head* (this assumes the *signal head* on the through lane is green) and then changing lanes to the *link* lane connected to the left turn *connector*. You may find it informative if, in your working with the model, you intentionally make this mistake and watch the subsequent behavior. (Later we will see how the *Emergency Stop Distance* parameter for a *connector* may be used to eliminate this potential error as well.)

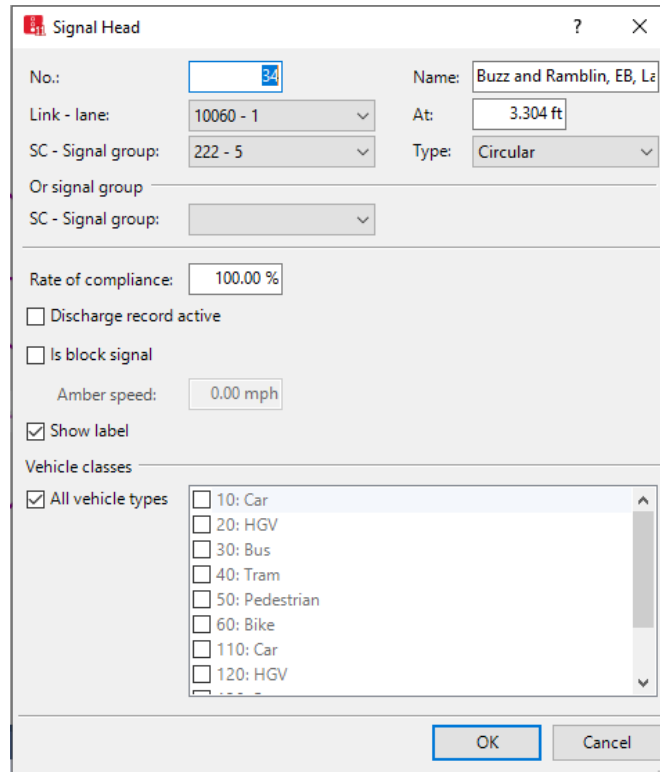
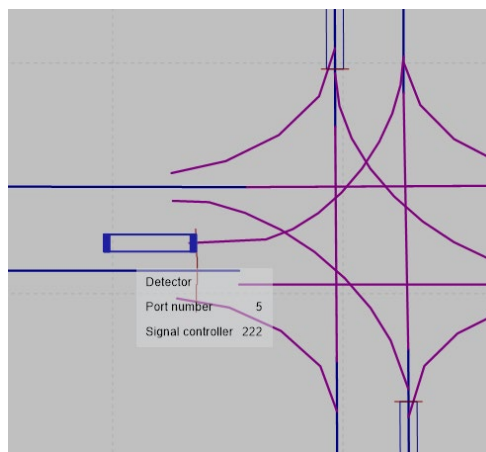
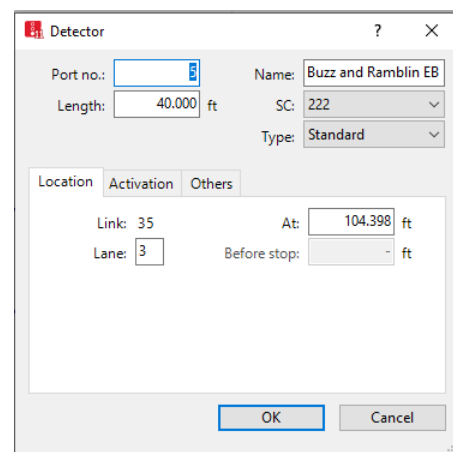


Figure 37: EB left signal head pop-up window

step x. Finally, we need to add the left turn *detector*. Select *Detectors* in *Network Objects* menu and add a 40 ft *detector* at the Stop Bar for lane 3. Name the *detector* Buzz and Ramblin EB Left, change SC to 222, and *Port No:* to 5, Figure 38.



(a)



(b)

Figure 38: EB left turn (a) Detector placement and (b) Detector pop-up window

step xi. Repeat steps 5 through 12 for the EB approach. You will again use *add link left* as left and right are defined by the direction of travel on the *link*. The EB left turn is phase 1, so be sure to update the steps according, i.e., use *Signal Head, SC - Signal group 222-1* and *Detector Port 1*.

step xii. Finally, the intersection conflict areas need to be updated. The following conflicts should be set (*Figure 39*):

- Four crossing conflicts, one for each of the left turn conflicting with opposing through movement. The through movements receive priority.
- Two merging conflicts, one for each left turn and opposing right turn movements entering into the same lane of the NB and SB receiving lanes, give right turn priority. (There is not a conflict on the EB and WB approaches the left and right turns are received by different lanes.)
- Four merging conflicts, one for each through movement with the merging right turn on red, give through movement priority.

(Strictly speaking the crossing conflict between the EB or WB through and the opposing left is not required if the left is protected only. However, it does not change the protected movement behavior if this conflict is defined and simplifies the process of adding a permitted left turn later, if desired.)

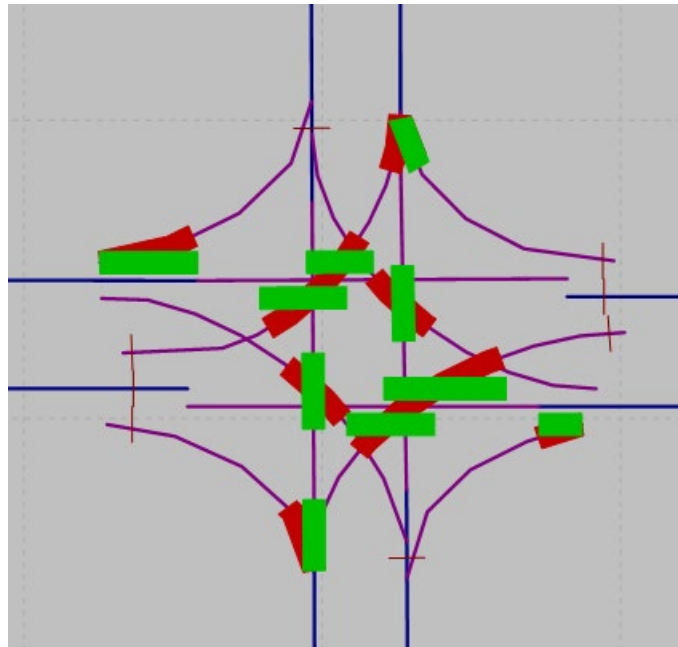


Figure 39: Buzz Blvd. and Ramblin Dr. Conflict Areas

This would be a good time to run the model. Carefully watch the WB and EB left turns to make sure they are operating as expected. You may find it helpful to temporarily increase the percentage of traffic going to the EB and WB left turns (this may be found in the routing decisions.) Also look at any warnings or errors generated by the model.

Separate left turn bay *link* vs combined left turn bay with through lane *link*

A reasonable question is what is the difference between vehicle behavior in these two left turn lane modeling methods. In our original method (Module 1) we utilize a separate *link* for the left turn bay. In this method a left turning vehicle must enter the left turn bay using the *connector* at the upstream location of the left turn *link*. In the alternate method the left turn lane and the through lanes are part of the same *link*. In this method a left turn vehicle may change lanes into the left turn lane anywhere along the *link*. (This is not strictly true based on the set *connector Emergency Stop Distance* which we will discuss later; however, it is an acceptable assumption for the current discussion.) Thus, a vehicle may enter the *link* at the start of the left turn lane or may wait until it is further down the *link*.

There are a number of arguments for both methods; however, from a high-level perspective, it is important to understand that Vissim is a flexible modeling platform and with a reasonable understanding by the model developer either method can be used in most circumstances. The choice of method is often driven by model developer preference. In addition, many software applications that automate the creation of a Vissim link network from another model use the alternative method. It is rarely worth the effort to switch from one method to another when starting the development process from an existing model.

An advantage of the alternate method is that it provides a more realistic model of left turn behavior, in that, in the field vehicles may change lanes anywhere along the left turn lane. In addition, if there are driveways adjacent to the left turn bay this method will allow vehicles that enter the through lane after the left turn bay has started to still change into the left turn lane. However, the alternate method can be more difficult to calibrate. For instance, a left turning vehicle may stay in the through lane until it is closer to the stop bar and then get “stuck”, having difficulty merging into the left turn bay due to queuing (Figure 40, the orange vehicle is blocking the center lane while seeking to merge into the left turn lane). This can cause blocking of the through lanes that is not observed in the field. The original method’s avoidance of this problem is often a reason for its preference.

To address this challenge in the alternative method there are three readily available calibration techniques that are commonly utilized, *connector Emergency Stop* and *Lane Change Distances* and vehicle *Diffusion Time*. (There are of course other calibration techniques and parameter adjustments that could be undertaken; for instance, Vissim 2020 introduced an ability to set the distribution of lane changes along the turn bay. However,

we will focus on these three as they will be found to be critical not only here but in modeling many conflicts in Vissim.)

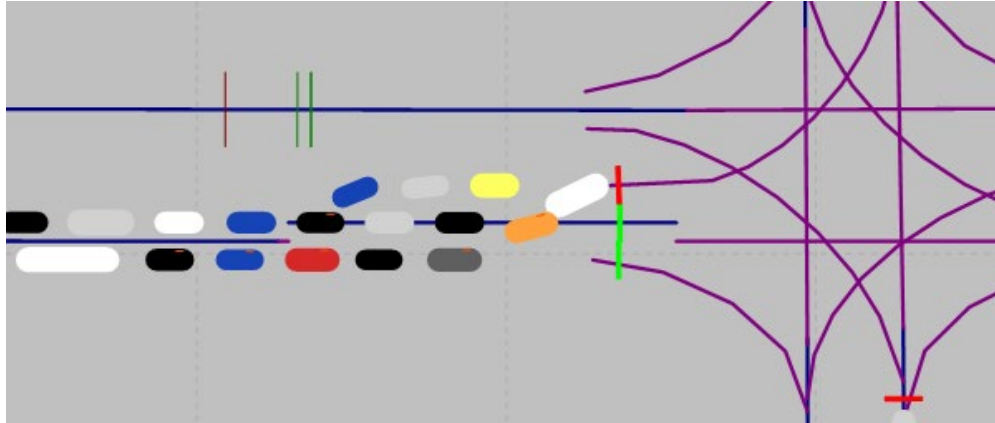


Figure 40 Through Lane Blockage by Left Turning Vehicle

Emergency Stop Distance.

If you open the EB Left Turn *connector* pop up window, *Figure 36(b)*, near the bottom of the window you will see *Emergency Stop Distance*. This powerful calibration parameter sets the distance upstream of the *connector*, by which a vehicle on a route that uses the *connector*, must be in a lane that connects to the *connector*, without requiring any lane changes. The default value is 16.4 ft. Thus, in our example, a vehicle that is assigned a left turn route must be in lane 3 (the *link* lane to the *connector*) by the time it reaches within 16.4 ft of the *connector*. If not, the vehicle will stop and wait for an opportunity to merge into lane 3. By setting the *Emergency Stop Distance* higher we can eliminate late lane changes. For instance, set the value for the EB left *connector* *Emergency Stop Distance* to 75 ft and run the model. You should not see any vehicles seek to change into lane 3 within 75 ft of the *connector*. If you set the *Emergency Stop Distance* value to a length slightly less than the distance from the *connector* to the end of the *link* the vehicle behaviors will become similar to that of the separate turn bay *link* method, as we are not allowing vehicles to change into the left turn lane anywhere along the turn bay length except at the very beginning.

CAUTION: If the Emergency Stop Distance exceeds the length of the lane connected to the *connector* (lane 3 in our example) vehicles will never be able to make the left turn as they cannot switch into a lane that requires no additional lane changes, i.e., lane 3. Left turning vehicles will stop on the preceding upstream *link*, never able to move forward! Give it try, set the *Emergency Stop distance* longer that the distance from the *connector* to the end of the *link* and observe what happens.

Helpful Hints

Here and in other sections of the modules it is recommended that you enter parameters that you know will not work. It is a good idea to do this kind of testing. It will give you a sense of what the different failures look like when observing a model. This can be very helpful when reviewing your work or that of others, where the errors are not intentional!

One last comment on the use of *Emergency Stop distance*. While we have focused on the left turn lane change there are other behaviors we may wish to control. For example, in the alternative left turn design (i.e., through and left lanes on the same link) it is possible that a through vehicle might choose to change into the left turn lane to “get around” a slower vehicle. This changing in and out of the left turn lane by through vehicles can be disruptive, and lead to unrealistic behaviors and congestion not reflective of field conditions. (Another example of this behavior can be seen on freeway models, where through vehicles may change into off-ramp deceleration lanes to pass slower through vehicles). To eliminate this behavior, increase the *emergence stop distance* to greater than the distance from the *connector* to the end of the left turn bay. With this change a vehicle on a through route will not be allowed to change into the left turn lane, as the left turn lane does not connect to the through *connector* without the need for lane changes. (This distance can be greater than the distance to the end of the *link* with the left turn bay as a vehicle on the preceding upstream *link* will require no lane changes to reach the through *connector*.)

Lane Change Distance

A second parameter related to *connectors* is the *lane change distance*. This is the distance upstream of a *connector* where a vehicle on a route that crosses the *detector* may first start seeking to change lanes to reach the *connector*. Prior to this the vehicle is unaware of the need to change lanes to reach the connector. The default parameter length is 656.2 ft, which in most arterial instances will be sufficient; however, longer distances are often desirable on freeways. An example of the need for longer *lane change distances* on arterials may be where a route contains closely spaced intersections or extensive congestion and early lane changing is required for a vehicle to properly preposition to be in the correct lane(s) to

enter the *connector*. As the lane change distance is associated with vehicles having a route crossing the *connector* the lane change distance may not exceed the distance to the route starting point. (The entered value may be higher but in the model execution it will be limited to the distance to the route start, as it is not possible for a vehicle to start responding to a route it is not yet on.) While beyond this module, in situations where additional distance beyond the vehicle route is needed for a vehicle to successful preposition for a *connector* a modeler may explore the use of *combining static routes*. Details on this may be found in the Vissim manual.

Time to Diffusion

When considering *Emergency Stop* and *Lane Change distance* in combination it becomes apparent that these parameters have the ability to significantly influence vehicle behavior. However, in some instances, vehicles may continue to appear “stuck”. For example, the vehicle in Figure 40, that stayed in the through lane, will not be able to change lanes until the left turn queue clears. In other circumstances it is possible that a vehicle may be caught in a conflict for a much longer, possibly indefinite period, resulting in lane blockage that is not realistic. Vissim addresses these vehicles using *Diffusion Time*. *Diffusion Time* is the maximum amount of time that a vehicle may wait at the *Emergency Stop Distance*. *Diffusion time* defaults to 60 seconds but may be changed by the modeler. *Diffusion time* may be found in:

Base Data → *Driving Behaviors* → (*Select a Behavior*) → *Lane Change* →
Waiting Time before Diffusion.

Once the vehicle waits for this length of time it will be removed from the network and a warning written to the error file. In many larger networks or networks with congestion it is not unreasonable for a few vehicles to be diffused; however, instances of such diffusion should be checked to make sure there is not a modeling error. If a significant number of diffusions are occurring at a location than a modeling error is likely. Modelers should avoid lowering the *diffusion time* to address conflicts that are most likely better served through other calibration parameters. Ultimately a vehicle being diffused is a failure of the model and should be limited as much as possible.

Alternate Freeway On-Ramp Configuration

Next, we will consider an alternate freeway on-ramp link configuration to that demonstrated in [Module 4](#). In this alternate model design, we will explicitly capture the taper of the acceleration lane. For the demonstration we will update the SB On-Ramp from Buzz Blvd to Peach Freeway.

- step i. Split the SB freeway *link* near the end of the taper. This will be near the end of the current four lane *link* you are using to model the deceleration lane. This split should be close (say 20 or 30 ft) to the *connector* that connects the 4-lane *link* to the 3-lane *link*, [Figure 41](#). (See also [Figure 43](#))

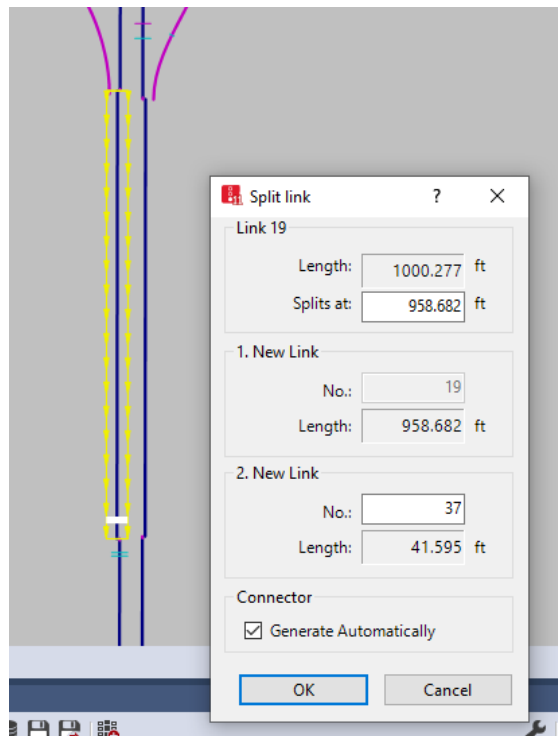


Figure 41: Acceleration link split location

- step ii. For the short *link* you just created at the end of the taper set the lane 1 (the acceleration lane) width to 1-ft. Slide the new short *link* over slightly so the *link* aligns with the downstream *link*, [Figure 42\(a\)](#). Or, from the lane view the left edge of the short *link* and the downstream *link* align, [Figure 42\(b\)](#). (See also [Figure 43](#).)

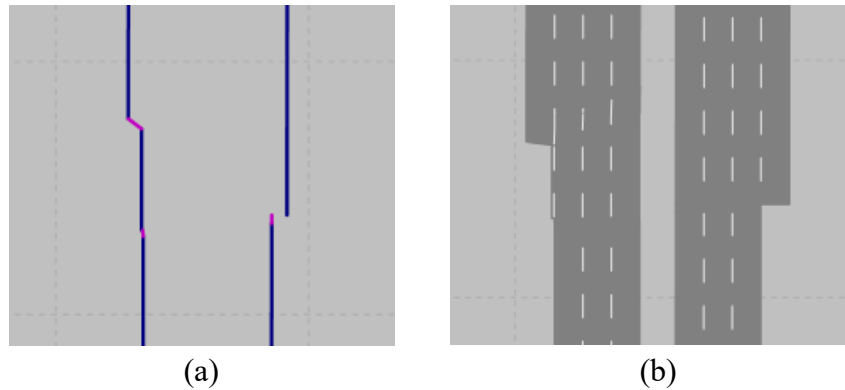


Figure 42: Realigned Link with 1-ft Lane 1 (a) link / connector view, (b) lane view

- step iii. Drag the downstream end of the *link* that connects to the upstream ramp gore (i.e., the link upstream of the short link you just set the 1 ft lane 1 width on) to the start of the taper, or the earliest point where vehicles begin to change from the acceleration lane to the freeway lane, *Figure 43*. This will result in a long connector between this link and the short link you just created.

Often you will have an aerial to judge the location where merging starts, for now just assume this point is downstream of the off-ramp *connector*. I used approximately 600ft for the long connector length, which would be the taper length for a 12 ft lane removed at at 50:1. Thus, the *connector* will essentially be the length of the acceleration lane taper. For this *connector* we will leave the *Emergency Stop* distance at the default value (16.4 ft.) as the upstream end of this 4-lane *connector* connects to a 4-lane *link* and no lane changes are required.

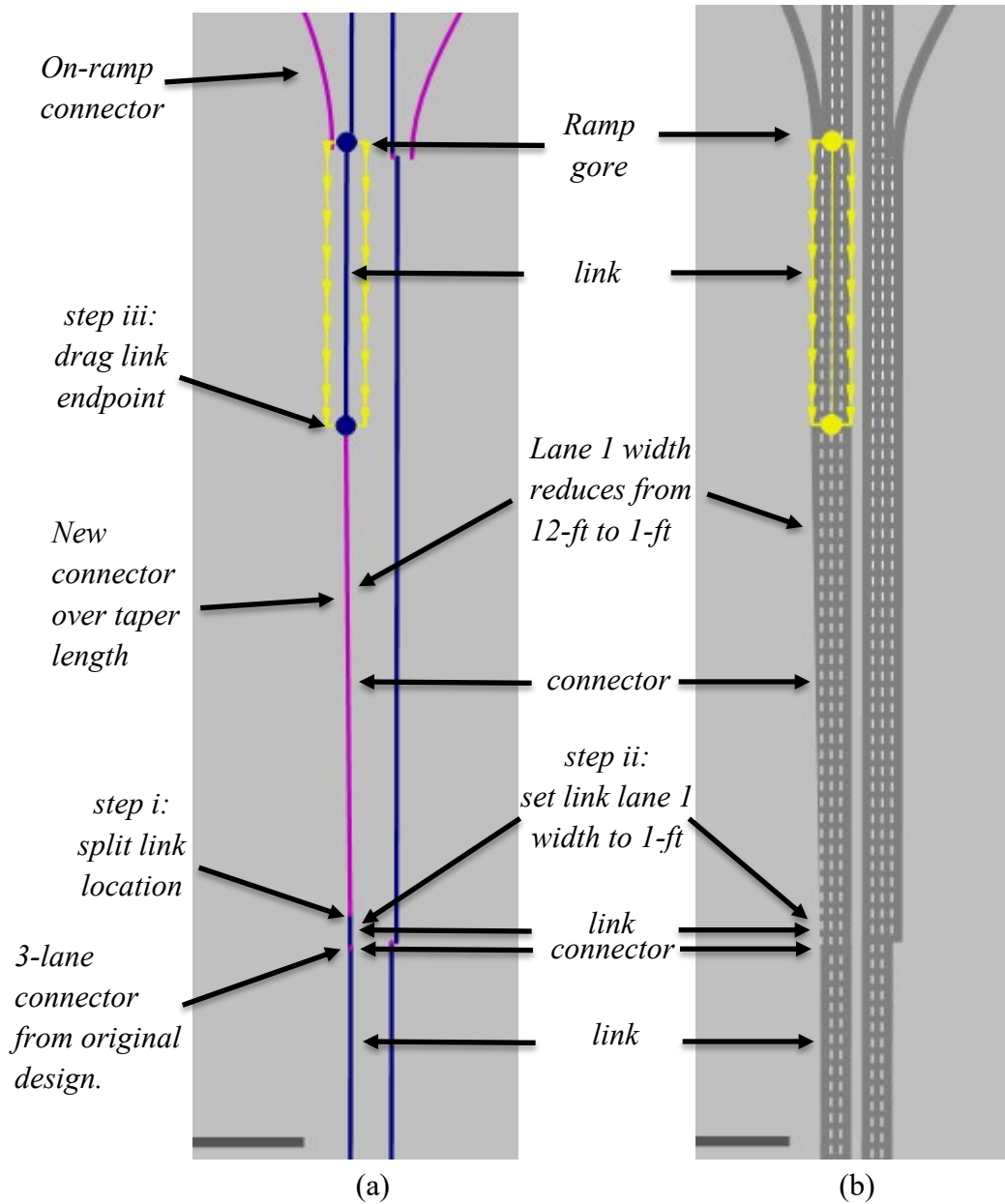

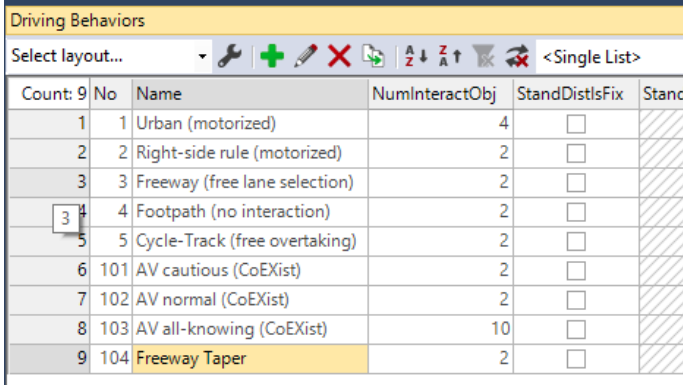


Figure 43: Updated Link and Connector Layout for Alternate Ramp Design

As seen in Figure 43(b) these three steps will create the visual appearance of the taper. We are taking advantage of a *connector* feature, where if the lanes at the upstream and downstream link connection points are of different widths the *connector* will linearly adjust the lane width over the *connector* length. However, as currently designed the behavior of vehicles will not change from traditional on-ramp design. In the next few steps we will show how to make vehicles in the acceleration lane respond to the taper and vehicles in the adjacent lane respond to acceleration lane vehicle encroachment.

- step iv. Go to *Base Data* → *Driving Behaviors*. As discussed earlier each *link* will have an assigned *driving behavior* and these *driving behaviors* define many of the behaviors exhibited by the vehicle.
- step v. Select the *Freeway (Free Lane Selection)* behavior. Click the *Duplicate Objects* Button. .
- step vi. You have now created a duplicate of the *Freeway (Free Lane Selection)* driver behavior. Rename the duplicate as *Freeway Taper*, *Figure 44*.



Count:	No	Name	NumInteractObj	StandDistIsFix	Stanc
1	1	Urban (motorized)	4	<input type="checkbox"/>	
2	2	Right-side rule (motorized)	2	<input type="checkbox"/>	
3	3	Freeway (free lane selection)	2	<input type="checkbox"/>	
4	4	Footpath (no interaction)	2	<input type="checkbox"/>	
5	5	Cycle-Track (free overtaking)	2	<input type="checkbox"/>	
6	101	AV cautious (CoEXist)	2	<input type="checkbox"/>	
7	102	AV normal (CoEXist)	2	<input type="checkbox"/>	
8	103	AV all-knowing (CoEXist)	10	<input type="checkbox"/>	
9	104	Freeway Taper	2	<input type="checkbox"/>	

Figure 44: Updated Driving Behaviors list

- step vii. Double click on the *Freeway Taper* behavior. The *Driving Behavior* pop-up window should open.
- step viii. Go to the *Lane Change* tab. Click on *Cooperative Lane Change* (if it is not already selected.)

Cooperative lane change

Maximum speed difference:

Maximum collision time:

Figure 45: Updated Lane Change Behavior in freeway taper Driving Behavior

- step ix. Go to the *Lateral* tab. Select *Observe adjacent lane(s)*, *Figure 46*.

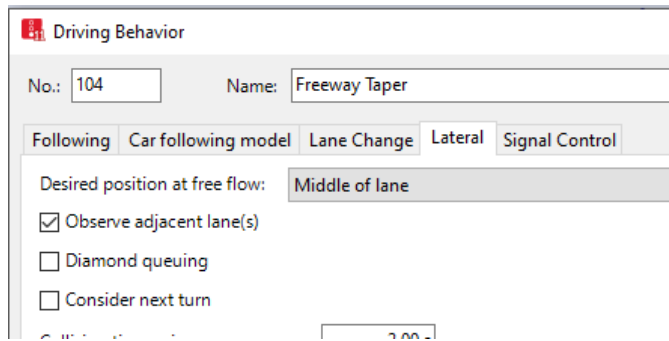


Figure 46: Updated Lateral in freeway taper Driving Behavior

- step x. Select *Ok*.
- step xi. Go to *Base Data* → *Link Behavior Types*.
- step xii. Select the *Freeway (Free Lane Selection)* behavior. Click the *Duplicate Objects* Button.
- step xiii. You have now created a duplicate of the *Freeway (Free Lane Selection)* driver behavior. Rename the duplicate as *Freeway Taper*.
- step xiv. Change the *link* behavior *Freeway Taper DrivBehavDef* to *Freeway Taper*, Figure 47.

Count	No	Name	DrivBehavDef
1	1	Urban (motorized)	1: Urban (motorized)
2	2	Right-side rule (motorized)	2: Right-side rule (motorized)
3	3	Freeway (free lane selection)	3: Freeway (free lane selection)
4	4	Footpath (no interaction)	4: Footpath (no interaction)
5	5	Cycle-Track (free overtaking)	5: Cycle-Track (free overtaking)
6	6	Freeway Taper	104: Freeway Taper

Figure 47: Updated Link Behavior Types list

- step xv. Double click on the *connector* you created that contains the acceleration lane. Change the *Behavior type* to *Freeway Taper*. You can do the same for the *links* upstream and downstream of this *connector*, although this is not critical.

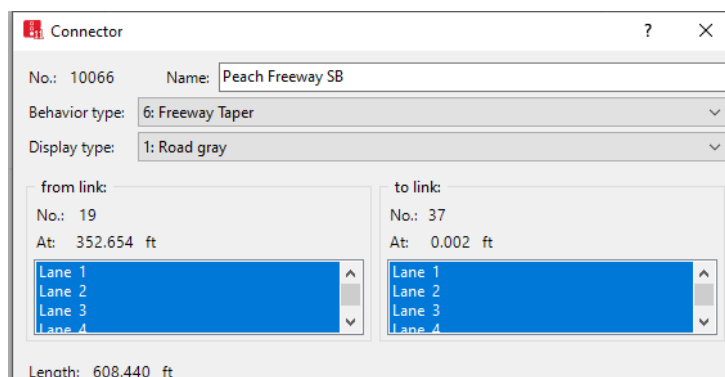


Figure 48: Updated Connector Behavior type

- step xvi. Click *OK*.
- step xvii. Finally, double click on the short *connector* to open the connector dialogue box. Reduce to *Emergency Stop Distance* to a lower value, say 50'.

What is happening in steps iv through xvii? By default, Vissim tracks a vehicle position as being in the center of the vehicle's current lane. Thus, for a 12-ft lane a vehicle's position is at 6-ft from either edge. In the taper *connector* we created the acceleration lane physically narrows as the vehicle proceeds down the lane. So, when the lane width is reduced to 10-ft the vehicle position is 5-ft from both edges, at an 8-ft lane width the vehicle is 4-ft from both edge lines, etc. Now, a vehicle clearly also has a width and at some point, that vehicle in the narrowing lane will begin to encroach into the adjacent lane. However, by default, Vissim does not consider this encroachment as it ignores the vehicle width. Thus, a vehicle is not considered to impact vehicles in the adjacent lane until the center of the vehicle crosses the lane line. That is, a vehicle can only occupy one lane at a time. By selecting *observe adjacent lanes* in the *driving behavior* a vehicle in the adjacent lane **will** respond to the encroachment of vehicles from a neighboring lane, i.e., the acceleration lane. The acceleration lane taper that we inserted in steps i through iii has the effect of forcing this encroachment. Finally, we reduced the *emergency stop distance* as the taper will help force the lane merge. We do this as prior to the *emergency stop distance* a vehicle must begin deceleration, so they may stop if necessary. By moving the *emergency stop distance* further downstream we lessen the impact of this deceleration and allow a more natural merge to occur based on the taper.

Overall, the taper and *emergency stop distance* combined with *observe adjacent lane* can improve the flow at the merge ramp section, in many cases eliminating stops and creating more of a zipper type merging behavior. When implementing, you may need to spend some time calibrating the length of taper and the *emergency stop distance* (as well as *lane change distance*) to achieve operations similar to the field. This is often a good technique to attempt where the standard method results in unexpected vehicle stopping and congestion in the ramp acceleration lane. Currently such challenges are often overcome by lowering the diffusion time. However, as discussed, vehicle diffusion is an indication of a model not properly working. This alternative approach may prove a better way to resolve this issue.

CONGRATULATIONS!!!

**You have completed
Concepts and Tips!**

Way to go!!!



https://commons.wikimedia.org/wiki/File:Mr._Smiley_Face.svg#file