User Manual for INC-ZONE and RESP-STG VISSIM Applications

Introduction

This user manual documents the simulated versions of the following applications within the Response, Emergency Staging and Communications, Uniform Management, and Evacuation (R.E.S.C.U.M.E.) bundle of DMA Applications:

1. Incident Scene Work Zone Alerts for Drivers and Workers (INC-ZONE)
2. Incident Scene Pre-Arrival Staging Guidance for Emergency Responders (RESP-STG)

The application also initiates an incident at a requested location, time and link which will enable a short lane-closure for specified number of lanes by using a stopped vehicle in the simulation system. The applications demonstrated were developed as a part of the US Department of Transportation, Intelligent Transportation Systems Joint Program Office's Project on the Impact Assessment of INC-ZONE and RESP-STG applications. This document uses a simple 2-node network to demonstrate the application functionalities. The users are expected to add minimal modifications so as to use it on larger networks. The applications demonstrated are specific to VISSIM micro-simulation software developed by PTV Vision and uses Component Object Modeling in Python to initiate and manage the simulation.

Prerequisites

The application is designed to run with the following installations in place:

1. VISSIM 6 or 7 microsimulation program with a registered COM server.
2. Python 2.6 or 2.7
3. VISSIM network with following vehicle classes and types:
    a. Vehicle Type 101 added using base class of HGV vehicles. This vehicle type initiates the incident by stopping on the requested lane at the requested time and location.
    b. Vehicle Type 102 added using base class of HGV vehicles. This vehicle serves as the emergency vehicle being deployed as part of the RESP-STG application.

The application (and the simulation) can be initiated by running the file: `manager.py` which contains all the application functionalities and a simulation manager. The inputs for the simulation and applications are defined in this function and can be changed inline.

Application Description

The application provided in manager.py runs out of box for the included sample network, `network.inpx`. This network consists of a 2-node freeway with 5-lanes on both directions with 55 miles per hour speed limit. The length of the links are kept longer so that it can entail most of the vehicle commands. The application consists of the following functions:

1. `IncidentCreate()` – creates the incident by adding a stopped vehicle at a specific location.
2. `IncidentRemove()` – clears the incident by removing the stopped vehicle at the specified location.
3. `IncZone()` – Queries the vehicles using their link, lane and location to generate 'threatened' vehicles and provide them with speed and lane guidance to avoid the incident lane. Distances are computed using MUTCD guidelines for work zone. `IncZone()` function is used at market penetration levels 10, 25, 50 and 100 percent.
4. `IncZone2()` – Similar to `IncZone()`, but for 75 percent market penetration.
5. `ClearIncZone()` – Defaults vehicle behavior after the incident is cleared.
6. `RespStgDispatch()` – Dispatch emergency vehicles to the incident scene, based on the number of lanes blocked.
7. `RespStgRouting()` – Routes the emergency vehicle to the incident scene by avoiding overcrowded lanes.
8. `ClearRespStg()` – Emergency vehicles leave the location of the incident when incident is cleared.
9. `runSimulation()` – This is the simulation manager which initiates different functions mentioned previously
10. `startup()` – This function contains all the input variables required for the simulation. A brief description of inputs is provided in this chapter.

Input Variables:

All the input variables required for the simulation are provided below. The input variables are defined in the `startup()` function and describes four types of variables:

1. Simulation Variables:
    a. `Random_Seed` = starting seed for the simulation.
    b. `End_of_simulation` = simulation duration in seconds.
    c. `Sim_Resolution` = resolution of the simulation in steps per second.
    d. `Speed_Limit` = speed limit of the link containing the incident.
2. Incident Variables:
    a. `Incident` = Boolean input on whether to initiate an incident or not.
    b. `Crash_Start` = Start time of the crash in seconds.
    c. `Crash_End` = End time of the crash in seconds.
    d. `Crash_Link` = Link on which crash needs to be initiated as a string.
    e. `Blocked_Lanes` = Number of lanes blocked during the crash (1 or 2).
    f. `Crash_Loc` = Location of the crash on the link in feet.
3. INC-ZONE Variables:
    a. `INC_ZONE` = Boolean input on whether to initiate INC-ZONE or not.
    b. `Pen_Rate` = Market penetration rate or percentage of vehicles to be given vehicle commands.
    c. `IZDelay` = Number of seconds between the crash initiation and INC-ZONE application initiation.
    d. `IZFreq` = Rate of INC-ZONE application messages in terms of simulation steps.
    e. `Zone_1_speed` = Desired speed of vehicles in the farthest zone of the incident.

    f. `Zone_2_speed` = Desired speed of vehicles in the nearest zone of the incident.
    g. `Zone_3_speed` = Desired speed of vehicles beyond the incident.

4. RESP-STG Variables:
    a. `RESP_STG` = Boolean input on whether to initiate RESP-STG or not.
    b. `RespDelay` = Number of seconds between crash initiation and emergency response using RESP-STG.

Sample Files

Three sample files are provided with this document:

1. Network file (`network.inpx`): This VISSIM file consist of a 2-node 5-lane freeway network with both directions with around 4000 vehicles/hour traffic volume on each direction.
2. Layout file (`network.layx`): This file supports the network file.
3. Application file (`manager.py`): This python file consists of the sample applications for incident management as well as INC-ZONE and RESP-STG applications along with ability to batch run simulations.

The simplified network will demonstrate the application capabilities well and color coding of vehicles are used to demonstrate the functionalities. Zone 1 vehicles are represented using Blue color, Zone 2 using Yellow and Zone 3 using White when INC-ZONE application is activated. RESP-STG enabled responder vehicles are shown in Red.

Please note that simple modifications to the code are required if it is being used for other networks including:

1. INC-ZONE application modification: If the incident link length is shorter than the messaging extent (computed by `dist_one` in the code), then the `IncZone()` and `IncZone2()` functions should be modified to include vehicles on upstream links, until `dist_one` limit is included.
2. RESP-STG application modification: If the emergency response vehicle has to travel a few links, they should be added to the `RespStgRouting()` function. If there are more routing options to choose from, users are encouraged to use Data Collection Points to determine the faster route.