12-746 Fundamental Python Prototyping for Infrastructure Systems

Project Proposal:

Monte-Carlo Generator for Bridge Traffic Configurations Data

Name: Jingxiao Liu **Andrew ID**: jingxial

Date: Sept 2016 **Email**: jingxial@andrew.cmu.edu

Introduction

The transportation system is regarded as blood system in human's body. An efficient and safe transport infrastructure, especially bridges, that is responsible for moving goods and people. Recently, the Weigh-in-Motion (WIM) systems measure vehicle weights accurately for many months or even years of real traffic. Statistical analyses of the WIM data provides an efficient way to predict future load effects. However, the cost of time and money for collecting and measuring limits the amount of traffic data. In order to extend the quantity of traffic data, artificial traffic data is generated depending on analyses of traffic characteristics by using Monte-Carlo simulation. Then, the generated traffic data is used for estimate extreme load effects during the infrastructure lifetime.

During last 8 months, I have almost finished the probability descriptions and modelling of traffic physical configurations, including gross vehicle weight (GVW), axle weights, axle spacings and number of axles. That model developed by the MATLAB uses multimodal distribution (Tri-modal Lognormal, Normal, and Weibull distribution) and Copula correlation functions (for more detail, please contact me for my thesis). In this project, I would like to use the parameters calculated from the mentioned model to generate bridge traffic configurations data by Python.

Objectives

This Monte-Carlo generator should:

- Be a packaged objected-oriented programming project;
- Have legible and reasonably structured inputs;
- Output artificial traffic efficiently.

Potential users

If this project can be achieved, the main customers should be bridge designers and structural researchers. Because this project can efficiently generate traffic configurations data for more than 10 years or even 100 years, also it can do some predictions of future traffic composition by changing those distribution parameters. Users can apply these generated data to bridge finite-element models with scientific traffic flow model for monitoring during bridges' operation time.

Main Features

For this work, the project should be written using object-oriented (OO) programming. In this project, it should contain three classes: the *Vehicle* class, the *Pattern* class, and the *Time* class. As an example of OO approach, properties of the physical vehicle (Gross Vehicle Mass (GVM),

number of axles, axle spacings, axle weights etc.) are programmed into the *Vehicle* class. In order to reduce errors and protect original data, *Vehicle* only allows the rest of the program access to these class members through public functions. Also, *Time* and *Pattern* are included in *Vehicle*, *Vehicle* can return its time and pattern class of arrival on the bridge by calling functions. Input function will be used to read distribution and function parameters from *.csv files, and one output function will be written for writing BeDIT file (Please see my thesis for more details).

The main class of this project is *Vehicle*, which contains 12 properties: head, time, speed, GVW, length, number of axles, direction, lane, transverse length in lane, axle weights, axle spacings, and pattern. Some of them are the main considered attributes:

- Time: the *Time* class will be developed to indicate arrival time of Vehicles on the bridge.
- GVW: one function should be developed to generate tri-modal mixed GVW distribution (Lognormal, Normal, and Weibull distribution) samples efficiently.
- Axle Weights: Axle Weights property is a 20 length array for each vehicle. For each pattern class vehicles, axle group weights percentages will be generated firstly using Lognormal, Normal, and Weibull distributions and copula function. In order to ensure that the sum of axle group weight percentages is equal to the GVW, they will be scaled pro-rata.

Another main challenging feature of this project is the input traffic files. Three input files will define GVW, axle group weight percentage, axle spacing, number of certain type vehicles and pattern name. The values are separated by commas. For example, Pattern file will be used to store the GVW distribution parameters, axle group weight percentage model names, correlation parameters, axle spacing model names, pattern name and the proportion for each pattern. Explanations of Pattern files for a four-axle group vehicle pattern viewed in tabular form are:

p_1	μ_L	σ_L	AGWP model number	AGWP model number	AGWP model number	AGWP model number
p_2	μ_N	σ_N	Copula parameter	Copula parameter	Copula parameter	Pattern
c	a	b	Axle spacing model number	Axle spacing model number	Axle spacing model number	Proportion

Estimated Schedule

The following table is a planning of this project.

Task	Period
Review and establish project's structure	21st Sept – 24th Sept
Develop main object-oriented classes	25th Sept-1st Oct
Develop main functions	2nd Oct – 10th Oct
Develop I/O functions	11th Oct – 17th Oct
Finish report and video	18th Oct – 24th Oct