

2016

STRIDE | Southeastern Transportation Research,
Innovation, Development and Education Center

Final Report

Signalized Intersection Simulation Program for Education

Project No. 2012-076S



Author: Scott S. Washburn, Ph.D., P.E., University of Florida

March 2016



DISCLAIMER

“The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation’s University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.”

TABLE OF CONTENTS

INTRODUCTION	1
BACKGROUND	1
OBJECTIVES.....	1
PROGRAM DEVELOPMENT	2
SOFTWARE ARCHITECTURE OVERVIEW	2
PEDAGOGICAL FEATURES	2
<i>Geometric Input Data</i>	2
<i>Traffic Input Data</i>	3
<i>Signal Phasing/Timing Input Data</i>	3
SUMMARY	4
REFERENCES.....	5
APPENDIX: USER GUIDE	6

LIST OF AUTHORS

University of Florida, Department of Civil and Coastal Engineering

Scott Washburn, Associate Professor

Acknowledgments

The author thanks Drs. Michael Kyte (University of Idaho) and Dimitra Michalaka (The Citadel) for their constructive software review comments, as well as undergraduate students Megan McGinley and Ethan Stoop for their software testing efforts.

ABSTRACT

In the “Introduction to Transportation Engineering” class, two of the most difficult topics for students to learn are often queuing theory and signal operation and analysis. It is well documented in that the current generation of students are largely ‘visual’ learners and that ‘active learning’ exercises are likely to be a more effective teaching method than the traditional lecture. This learning paradigm for students lends itself well to the use of computer software, particularly traffic simulation programs, to significantly enhance student education in transportation engineering.

Software programs can be valuable learning tools, if the program is designed with the focus on education. With a well-designed signalized intersection simulation program that is focused on education and considers pedagogical issues, student interaction with the program through active-learning exercises will allow them to easily visualize the relationship between the various inputs and performance measures, which should lead to more efficient and effective student learning.

The result of this STRIDE project is a signalized intersection microscopic simulation program that can lead to more efficient and effective learning of signalized intersection operation and analysis than traditional commercial simulation programs.

Introduction

Background

In the “Introduction to Transportation Engineering” class, two of the most difficult topics for students to learn are often queuing theory and signal operation and analysis. It is well documented in the education literature (e.g., Nychka, 2011; Richards and McNally, 2011) that the current generation of students, and likely all generations to follow, are largely ‘visual’ learners. Additionally, ‘active learning’ is gaining considerable momentum in academia as recent studies have shown this approach to be a more effective teaching method than the traditional lecture (see Hanford [2012] for more information on this topic). This learning paradigm for students lends itself well to the use of computer software, particularly traffic simulation programs, to significantly enhance student education in transportation engineering.

Nearly all traffic simulation software packages, however, are designed for application (i.e., intended to be used by practitioners or researchers for specific applications), rather than illustrating the concepts and principles that lead to the outputs/results that are normally the primary interest of users of a software program. Most, if not all, commercial traffic simulation software programs have a significant learning curve for their use, particularly in the area of network coding and inputs specification. In an introductory-level class, there is little to no time available to devote to learning software operation. The focus needs to be on learning the concepts and theory of the subject matter, rather than network coding details, which usually vary from program to program.

Software programs can be valuable learning tools, if the program is designed with the focus on education. With a well-designed signalized intersection simulation program that is focused on education and considers pedagogical issues, student interaction with the program through active-learning exercises will allow them to easily visualize the relationship between the various inputs (traffic demand, phase sequence, phase times, etc.) and performance measures (delay, queue length, etc.), which should lead to more efficient and effective student learning.

Objectives

The objective of this project was to develop a signalized intersection microscopic simulation program that will lead to more effective and efficient learning of queuing theory and signalized intersection operation and analysis concepts.

Program Development

Software Architecture Overview

The signalized intersection capability was added to an existing traffic simulation framework under development by Dr. Washburn, called SwashSim.

SwashSim is a traffic micro-simulation tool that employs state-of-the-art software architecture. This architecture is object-oriented and built on the C#.NET framework programming model, which allows for a high level of extensibility and modularity. The new architecture also supports a high level of fidelity with respect to temporal and spatial modeling resolution.

One of the key features of SwashSim is the ability to model individual vehicle characteristics and dynamics in great detail; thus, being able to account for the effects of roadway design features on vehicle performance with more accuracy than other simulation tools. SwashSim is intended for both research and education applications; however, the signalized intersection module is focused primarily on education applications.

Much of the vehicle-movement logic in SwashSim is the same as that employed in CORSIM 6, with some notable exceptions:

- SwashSim uses the Modified Pitt car-following model (Cohen, 2002) as opposed to CORSIM 6's Pitt car-following model.
- SwashSim uses a 0.1-second simulation time resolution instead of 1 second for CORSIM 6.
- Explicit modeling of vehicle paths from system entry to system exit. CORSIM 6 does not explicitly model vehicle movements through an intersection area--the animation component of CORSIM 6, TrafVu, interpolates vehicle positions through the intersection areas based on estimated intersection vehicle entry and exit times from CORSIM 6.

Pedagogical Features

A complete overview of the software program is contained within the user guide, which is included in the appendix of this report. In this section, a brief overview is given of specific program features intended to make the program more efficient and effective for educational purposes:

Geometric Input Data

Rather than the user having to go through a time-consuming, and sometimes complex, network coding process, the SwashSim signal module includes two pre-configured geometry templates, as follows:

- Four approach legs, with each leg consisting of two through lanes and an exclusive left-turn bay
- Four approach legs, with each leg consisting of only two through lanes

Other geometric templates will be added to the program over time. Configurations with an exclusive left turn bay can be used to illustrate permissive and/or protected left-turn phasing. Configurations without an exclusive left turn bay could be used to illustrate a split-phasing scenario. Through the use of the pre-configured geometry templates, the network set up time, which does not contribute to student learning of transportation engineering, is negligible.

Traffic Input Data

The traffic data inputs are very simple and intuitive. Users enter the traffic demand in just one location, which consists of a graphical representation of the movement orientations at the intersection. The signal module is currently restricted to just passenger cars/trucks, so the user does not have to enter any vehicle type percentage data. However, the signal module does use an equal percentage of 11 different passenger vehicles, as indicated in the user guide.

Signal Phasing/Timing Input Data

Signal operation and signal analysis are obviously intrinsically linked; however, the details of signal operation, particularly the concept of dual-ring signal controller operation, are often overlooked or highly abstracted in undergraduate-level class.

One issue that can often cause confusion for students when discussing signal operation and analysis is the use of the term “phase”. For signal analysis purposes, this term is often used to refer to a group of vehicle movements that simultaneously receive the green indication. However, the term “phase” is also used to refer to an individual movement (left, right, through, or through/right combination) in the context of signal controller operation (i.e., timing-ring logic). To eliminate this confusion in teaching students, Kyte and Tribelhorn (2014) introduced the term “timing stage” to refer to a group of movements (phases) that move simultaneously. This software program makes use of the timing stage concept.

The use of the timing stage concept makes the specification of the phasing sequence very intuitive. Once the timing inputs (green, yellow, all-red intervals) have been specified, the program will automatically create a dual-ring timing plan. Presenting students with both the timing stage inputs and dual-ring signal controller inputs makes it easy for students to visualize and understand the relationship between the two. Implementing a signal phasing/timing input screen similar to the signal controller emulators that are available would not have served the pedagogical interests of this program, as these emulators are generally intended to teach signal engineers and technicians the “button-ology” aspects of a specific controller, not facilitate the learning of signal control theory.

Summary

This work is aimed at the education and workforce development components of STRIDE's mission. In particular, the results of this project will ultimately contribute to:

- Enhanced student learning in the Introduction to Transportation Engineering course for the topics of signalized intersection operation and analysis.
- Enhance general public understanding of signal operation through use of the software program at functions such as the E-Fair, Family Engineering Night, and the like.
- Assist with recruitment by getting students interested in transportation engineering through the use of a visually stimulating and engaging tool.

To obtain a copy of the program, contact Dr. Scott Washburn at swash@ce.ufl.edu.

References

Cohen, S. L. Application of Car-Following Systems to Queue Discharge Problem at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*, TRR 1802, Washington, D.C., 2002.

Cohen, S. L. Application of Car-Following Systems in Microscopic Time-Scan Simulation Models. *Journal of the Transportation Research Board*, TRR 1802, Washington, D.C., 2002.

Hanford, Emily. "Physicists Seek To Lose The Lecture As Teaching Tool". Web article. Posted January 1, 2012. <http://www.npr.org/2012/01/01/144550920/physicists-seek-to-lose-the-lecture-as-teaching-tool>

Kyte and Tribelhorn. "Operation, Analysis, and Design of Signalized Intersections: A Module for the Introductory Course in Transportation Engineering". CreateSpace Independent Publishing Platform. 2014

Nychka, John A., "Room with A VUE," *American Society of Engineering Education Annual Conference Jun 26-29, 2011 Vancouver, B.C.*

Richards, Grant P and Helen A McNally, "An Examination of Learning Styles and its Impact on Curriculum Development," *American Society of Engineering Education Annual Conference Jun 26-29,2011 Vancouver, B.C.*

Appendix: User Guide

SWASHSIM

Traffic Simulation Program

User Guide for the Signalized Intersection Module

Scott Washburn, Ph.D., P.E.
Department of Civil and Coastal Engineering

UF | **Transportation Institute**
UNIVERSITY of FLORIDA



March 2016

Contents

Introduction to SwashSim	1
Getting Started.....	2
Main Input Screen.....	3
Roadway Data	3
Traffic Data.....	3
Signal Phasing and Timing Inputs.....	4
Simulation Control Panel	7
Random Number Seeds	8
Warm-up Time	8
Simulation Duration	9
Output Options	9
Multi-Run	9
Run Simulation.....	9
Stop Simulation.....	9
Close.....	9
Animation.....	9
Animation Controls	11
Signal Phasing and Timing Window	12
Vehicle Properties	13
Results	13
Saving a Project File	15
Current Limitations	15

Introduction to SwashSim

SwashSim is a traffic micro-simulation tool that employs state-of-the-art software architecture. This architecture is object-oriented and built on the C# / .NET framework programming model, which allows for a high level of extensibility and modularity. The new architecture also supports a high level of fidelity with respect to temporal and spatial modeling resolution. One of the key features of SwashSim is the ability to model individual vehicle characteristics and dynamics in great detail; thus, being able to account for the effects of roadway design features on vehicle performance with more accuracy than other simulation tools. SwashSim is intended for both research and education applications; however, the signalized intersection module is focused primarily on education applications. This user guide documents the features of the signalized intersection module.

Getting Started

After you have installed the program¹, double-click on the SwashSim icon on your desktop or within the 'Start' menu. Alternatively, you can navigate to the installation folder and double-click on 'SwashSim.exe' to start the program.

From the start screen (Figure 1), select 'Signalized Intersection' with the mouse, and then click on the 'Load Facility' button. Alternatively, you can double-click with the left mouse button on the 'Signalized Intersection' label. The main input screen will then be loaded.

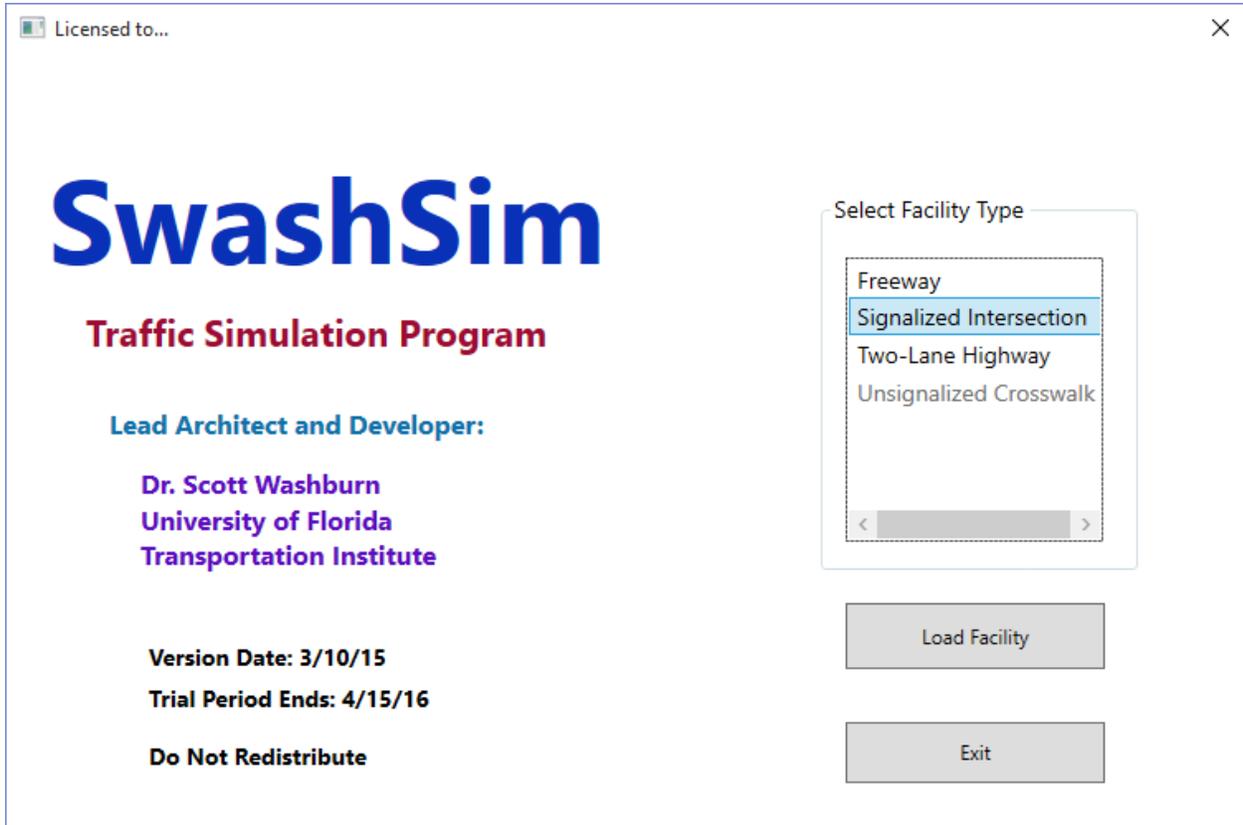


Figure 1

¹ You should have full administrative rights to the installation folder you choose for the program. See the section on the simulation control panel for more information.

Main Input Screen

The main input screen (Figure 2) is where the roadway and traffic data are specified.

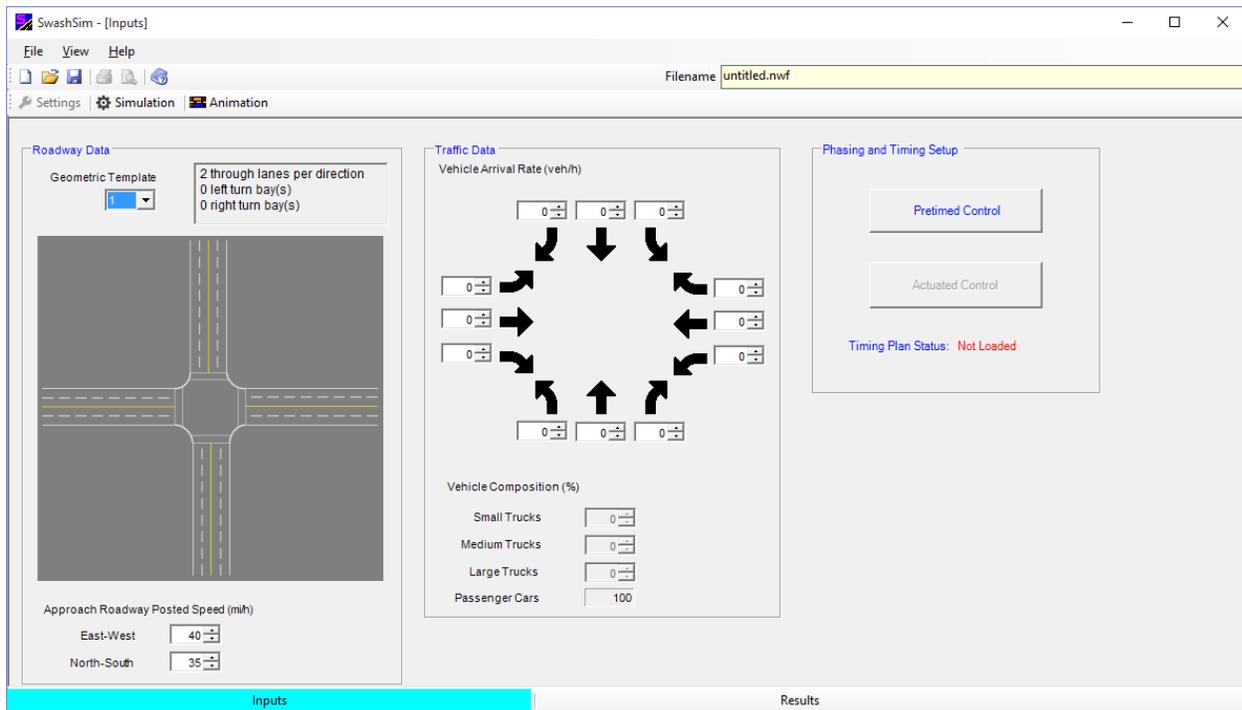


Figure 2

Roadway Data

- Configuration Template: one of two intersection configurations can be selected by clicking on the dropdown box under 'Configuration Template'. Configuration 1 consists of two thru lanes only per approach. Configuration 2 consists of two thru lanes a left-turn bay per approach.
- The posted speed limit for the E/W and N/S approaches are also specified in this section. These values are used to set the mean vehicle desired speed. The minimum value is 20 mi/h and the maximum value is 45 mi/h.

Traffic Data

- Enter the traffic demand, in units of veh/h, for each of the movements in this section.
- The vehicle type distribution is also specified in this section. The signal module of SwashSim is currently limited to passenger cars/trucks. However, it does consider an equal percentage of 11 different passenger vehicles:
 - 2006 Honda Civic Si
 - 2005 Mazda 6
 - 2008 Chevrolet Impala LS
 - 2004 Pontiac Grand Am GT
 - 2001 Volvo S40
 - 2009 Honda Civic
 - 1998 Buick Century

- 2002 Chevrolet Silverado
- 2011 Ford F150
- 1998 Chevrolet S10
- 2004 Chevrolet Tahoe

To specify the signal timing data, in the “Phasing and Timing Setup” group box, press the “Pretimed Control” button, which is described in the next section.

Signal Phasing and Timing Inputs

Pressing the “Pretimed Control” button on the main inputs screen will load the “Signal Phasing and Timing Inputs” input screen (Figure 3).

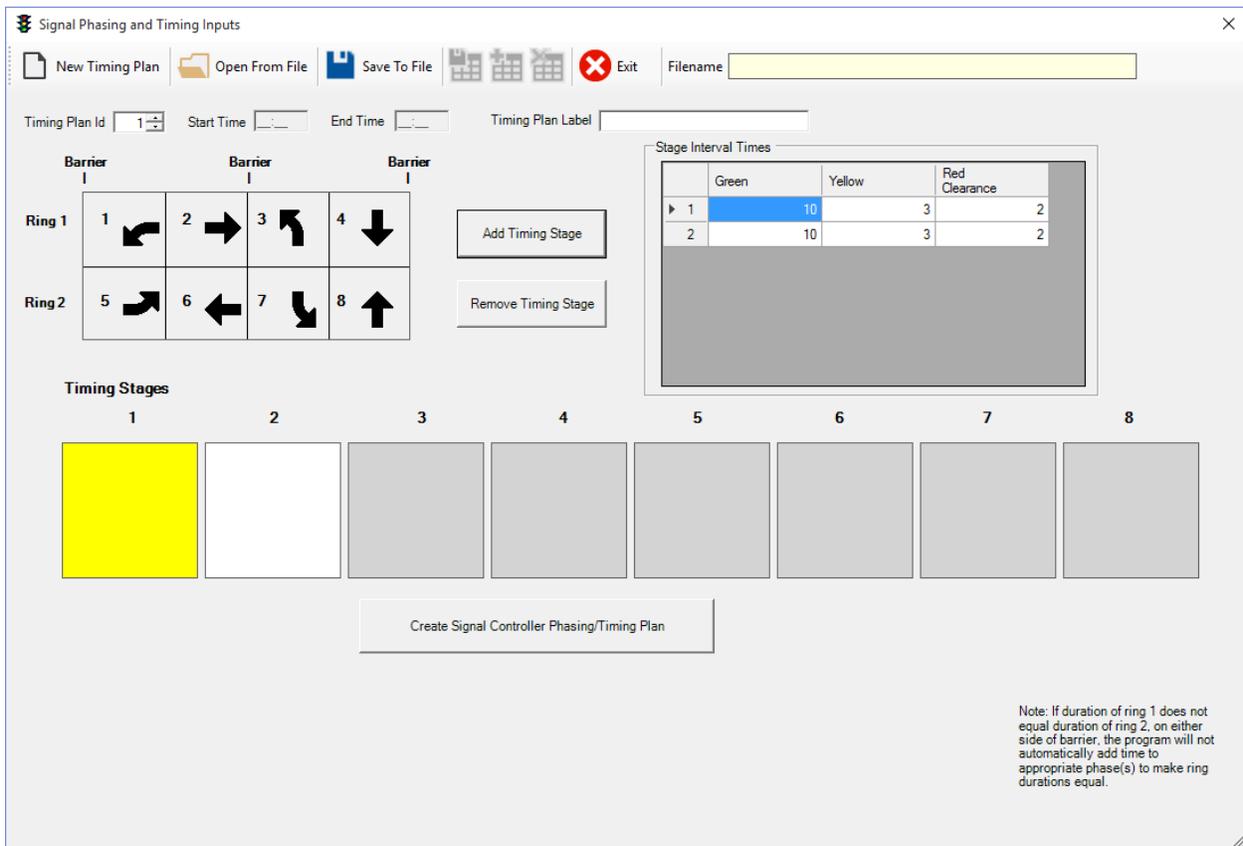


Figure 3

To add a timing stage, press the “Add Timing Stage” button. This will highlight the next available Timing Stage box in yellow. To edit a Timing Stage, click the desired stage so that it is highlighted

in yellow. Timing stages² are used to specify the phases (1-8) that will move (i.e., receive green) simultaneously. To add a traffic movement to the timing stage, click on the desired movement arrow(s) in the ring/barrier diagram. To remove a movement, click on the corresponding arrow in the ring/barrier diagram. If movement is selected that conflicts with another movement in that timing stage, an error message will appear and the movement arrow will not be added to the timing stage.

For each timing stage, the green, yellow, and all-red times need to be specified. To edit a timing interval, click on the appropriate table cell to highlight it in blue and type in the desired number. Press the 'Tab' or 'Enter' key to save edits. Note that the number in the left-most column corresponds to the timing stage number. Figure 4 shows an example timing stage plan.

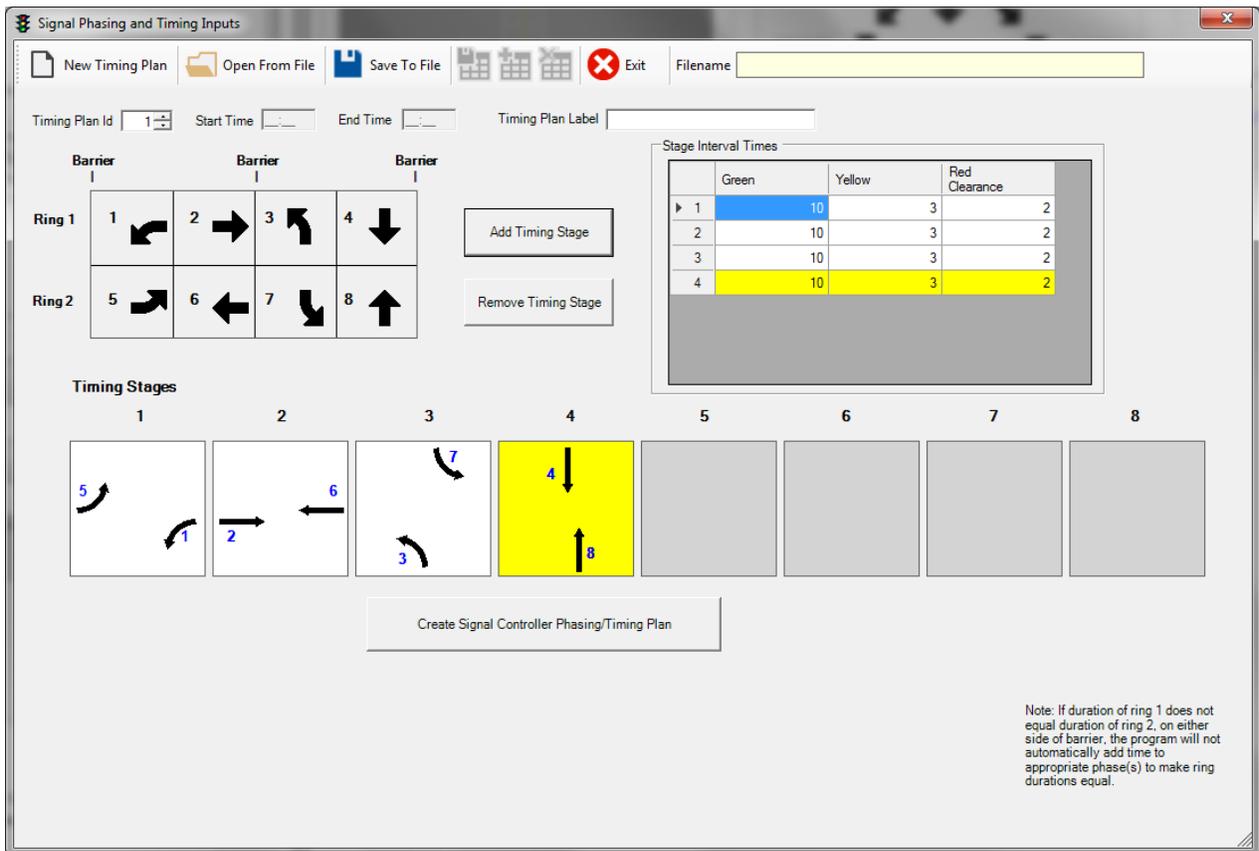


Figure 4

When the timing stage movements and corresponding interval times have been specified, the last step is to select "Create Signal Controller Phasing/Timing Plan". This will generate a timing plan that is used by the dual-ring signal controller (Figure 5).

² For more information on the term 'timing stage', please refer to "Operation, Analysis, and Design of Signalized Intersections: A Module for the Introductory Course in Transportation Engineering", by Michael Kyte and Maria Tribelhorn. CreateSpace Independent Publishing Platform. 2014.

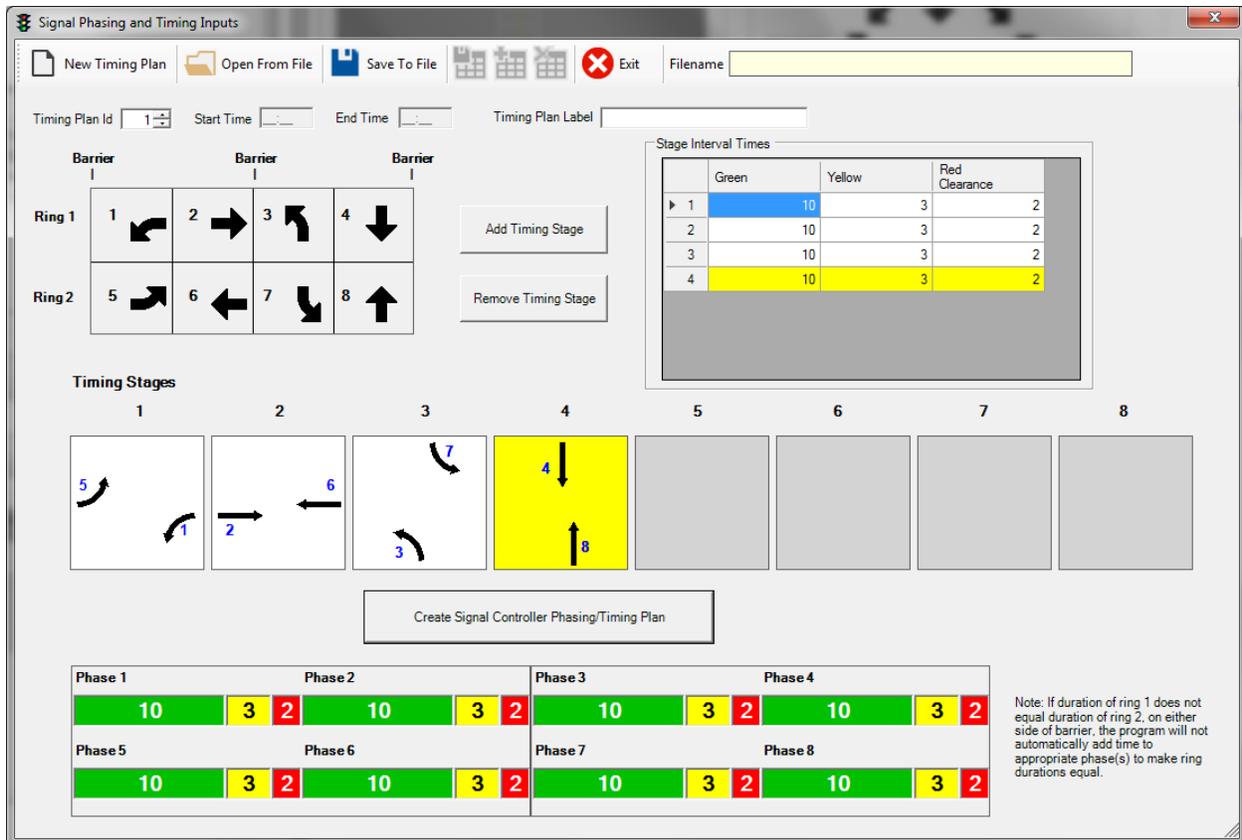


Figure 5

If revisions are made to the timing stages and/or interval times, press the “Create Signal Controller Phasing/Timing Plan” again to create a new signal controller timing plan. The last timing plan that is generated is the one that is stored in memory. The simulation program will continue to store this timing plan in memory for the duration of the current session. If you want to save the timing plan to disk, press the “Save To File” button. With this, the timing plan can be reopened in a different program session.

Simulation Control Panel

Once the roadway, traffic, and signal timing inputs have been specified, click on the “Simulation” button found on the toolbar (Figure 6).

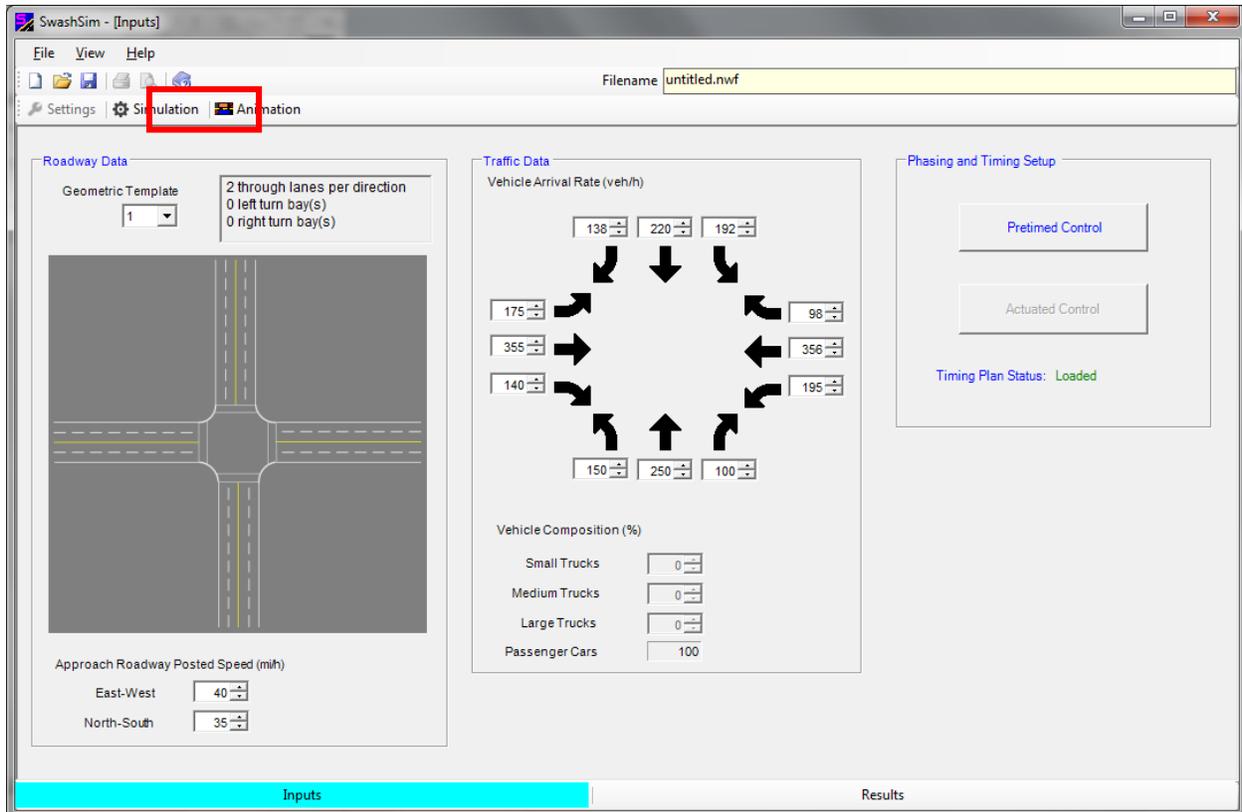


Figure 6

The “Simulation Control Panel” screen is shown in Figure 7.

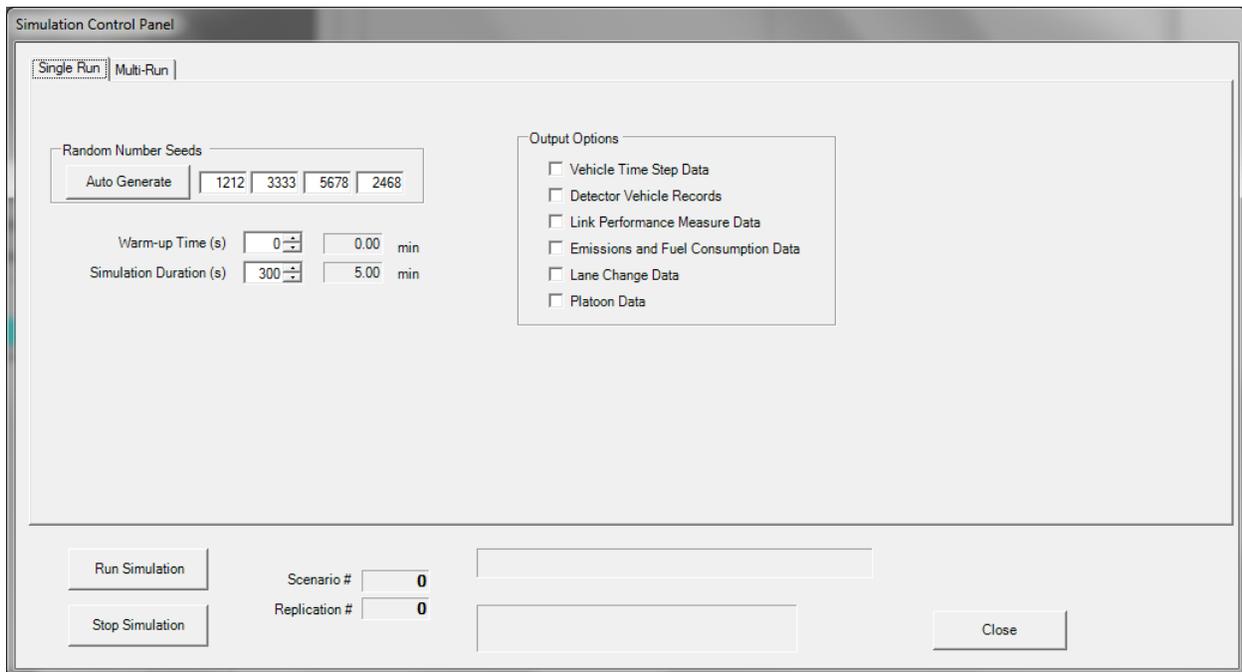


Figure 7

The “Simulation Control Panel” settings consist of the following:

Random Number Seeds

The random number seeds affect the various items within the simulation program that rely on random numbers. More specifically:

- 1st Random Number Seed – Applies to vehicle entry generation
- 2nd Random Number Seed – Applies to vehicle type generation
- 3rd Random Number Seed – Applies to driver type generation
- 4th Random Number Seed – Applies to all other randomly generated inputs

For the same inputs, a simulation run results will be at least slightly different if any of the random number seeds are varied. The seed numbers can be entered manually (any integer value), or they can be automatically created by pressing the “Auto Generate” button. If you desire to recreate the results exactly of a previous simulation run, use the same random number seeds. Note: When a simulation is run, a “RandSeedNumbers” text file is outputted. This file allows for previous seed combinations to be accessed. These values are also saved in the project file (discussed later in this guide).

Warm-up Time

This field dictates how long the traffic simulation will run before performance measurement data are collected. Enter a number between 1 and 900 seconds. Anything entered above 900 seconds will be defaulted back to 900. Press the ‘Tab’ or ‘Enter’ key to save the duration value before pressing the ‘Run Simulation’ button.

Simulation Duration

This field dictates how long the traffic simulation will run after the warm-up time. Enter a number between 1 and 3600 seconds. Anything entered above 3600 seconds will be defaulted back to 3600. Press the 'Tab' or 'Enter' key to save the duration value before pressing the 'Run Simulation' button.

Output Options

Time Step Data (TSD) File

Clicking this checkbox will generate a CSV file titled "TSD_int_0_0_1" upon running a simulation. This file is provided in case the user wants to view the detailed time-step by time-by-time vehicle properties in a spreadsheet program. This file gets written to the program installation folder. If file writing is not allowed to the installation folder under your user account, the program may crash during the file writing process.

The other output options are not applicable to the signalized intersection module.

Multi-Run

Multi-Run is available to advanced users. Please contact Dr. Washburn for instructions on how to use the multi-run options.

Run Simulation

Once all settings have been specified, click the 'Run Simulation' button. The textbox below that button will then display either "Running..." or "Finished". Running the simulation collects and prepares all data for the intersection animation discussed in the next section.

Stop Simulation

If you wish to stop the running of the simulation before it completes, press the 'Stop Simulation' button.

Close

Once the simulation is finished, press the 'Close' button to return to the main screen.

Animation

Once the simulation run has been successfully completed, press the "Animation" button on the toolbar (Figure 8).

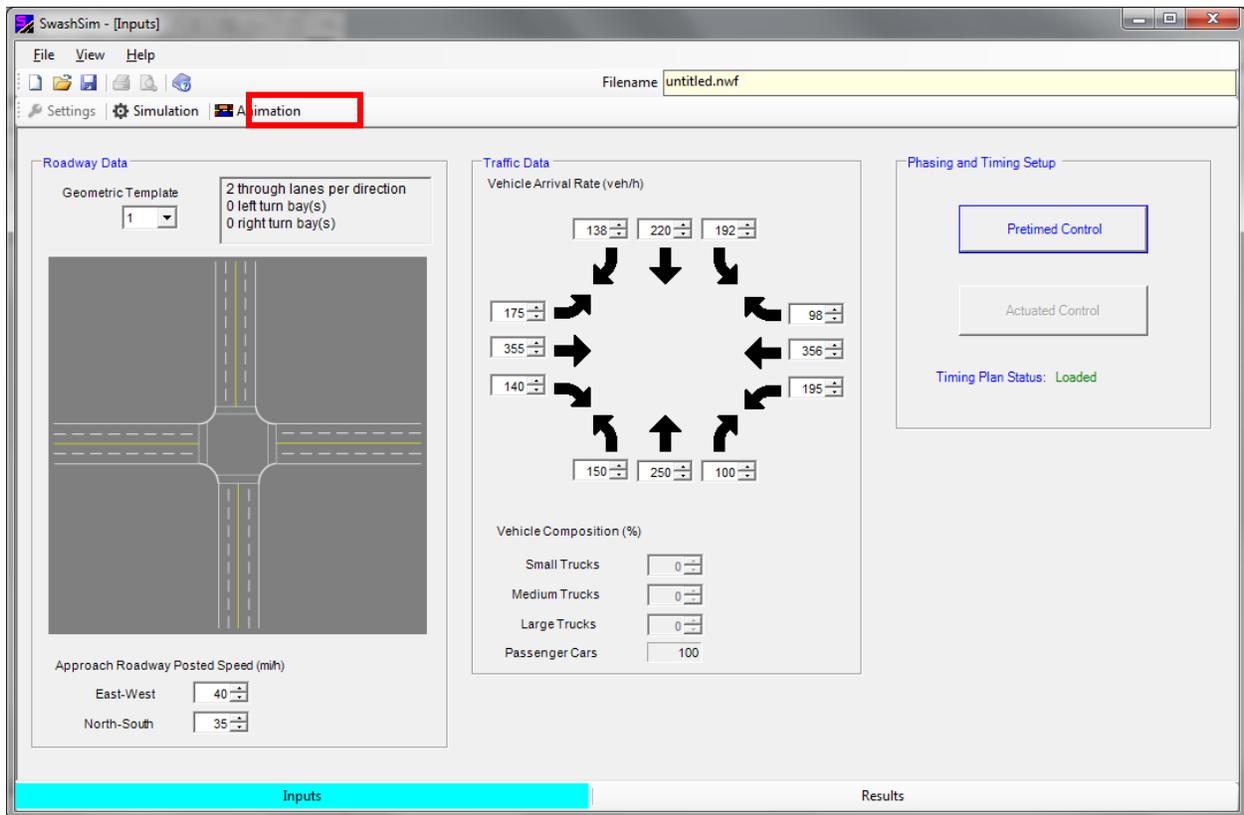


Figure 8

This will load the animation window (Figure 9).

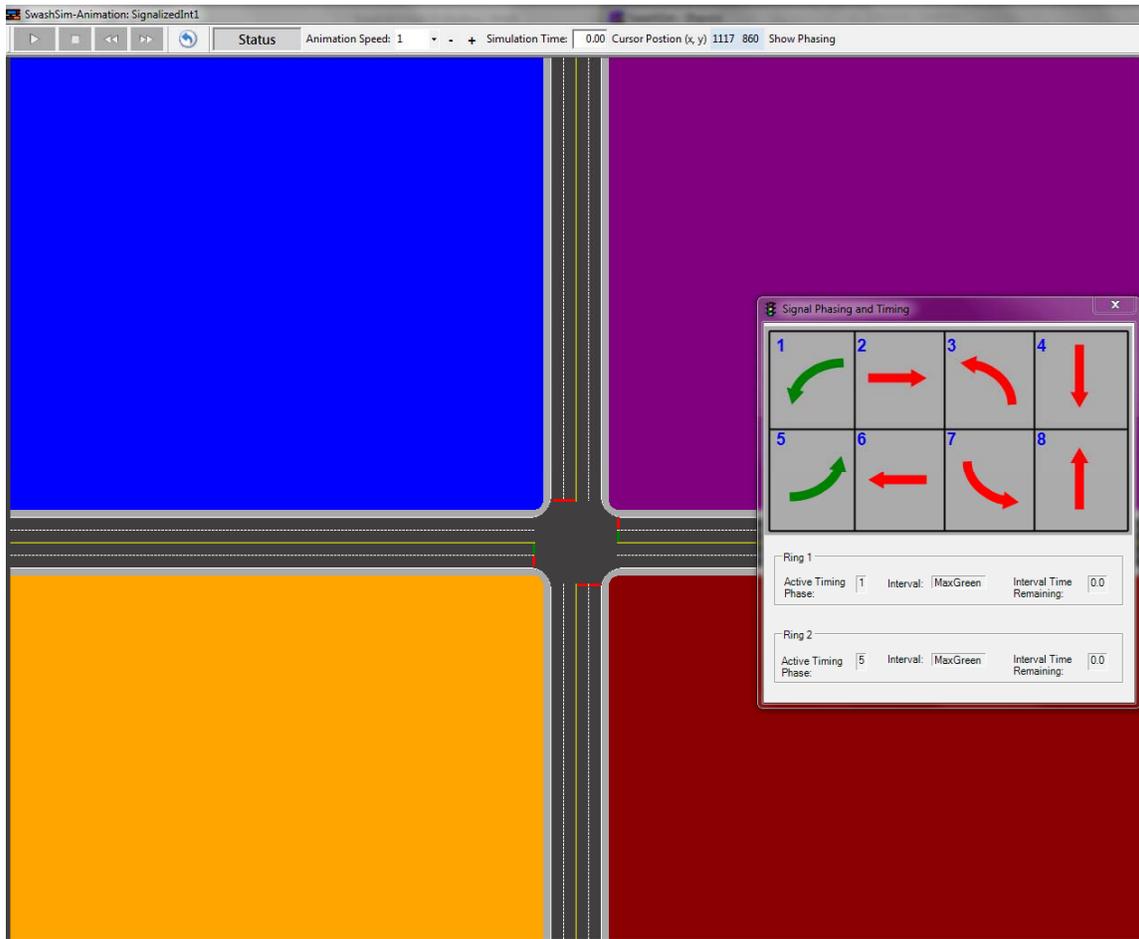


Figure 9

The intersection shown should match with the configuration template selected in the main inputs screen. The stop bars will change color to match with the active timing interval.

On the right side of the screen, a moveable window, “Signal Phasing and Timing”, is displayed.

Animation Controls

The animation controls (Figure 10) are contained in a toolbar at the top of the main screen. These controls are used to start, stop, and pause the traffic simulation animation. From this window, the animation speed can also be changed from 0.1X to 10X, using the ‘+’ and ‘-’ minus buttons. The animation can be played forward or in reverse, by using the left/right arrow buttons to the right of the stop button. When the animation has been stopped or is finished, it can be restarted by clicking on the button to the right of the ‘forward’ button.

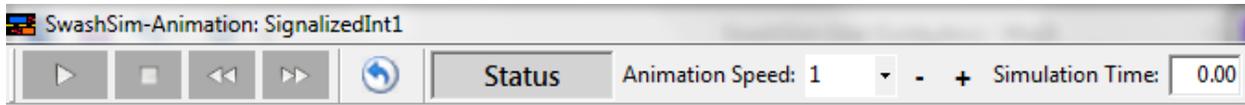


Figure 10

Signal Phasing and Timing Window

This window (Figure 11) displays the dual-ring phasing diagram. During the animation, the movement arrow colors will match with the current signal indication (i.e., green, yellow, red) for that movement. This display is coordinated with the stop bar colors displayed on the intersection (Figure 9).

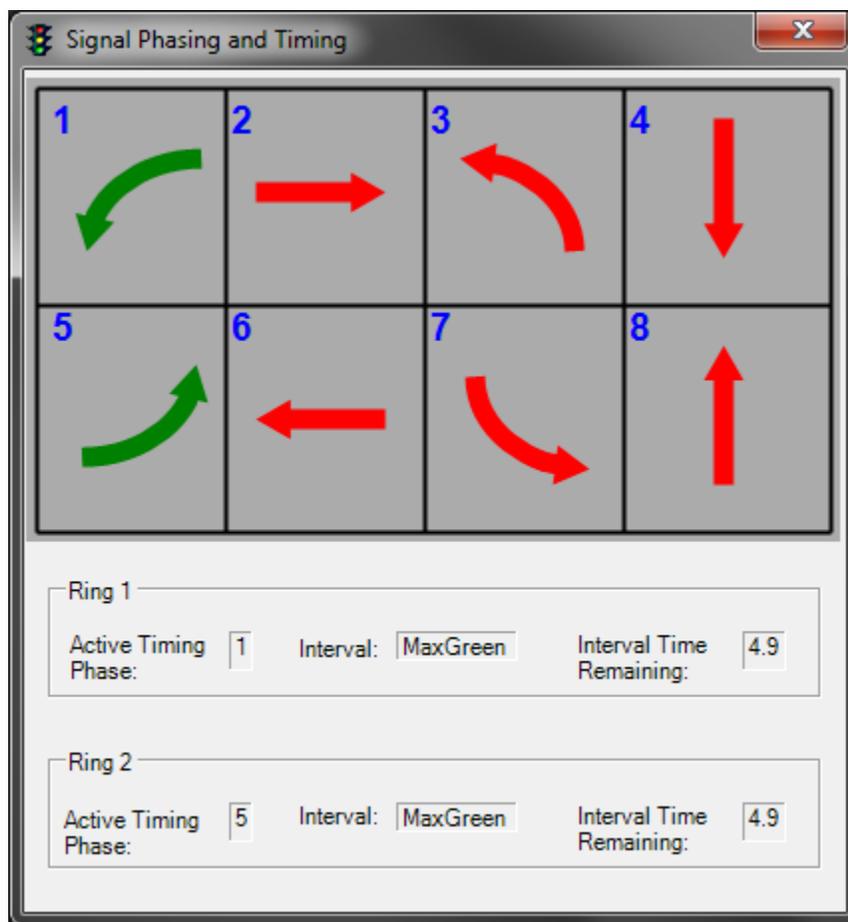


Figure 11

- Active Timing Phase – This represents the current active movement, colored in green. Recall that in the dual-ring diagram, phases 1 through 4 run from left to right in the top row and phases 5 through 8 run from left to right in the bottom row.
- Interval – Displays the active timing interval (i.e., green, yellow, all-red).

- Interval Time Remaining – Displays the time remaining in the active interval.

Vehicle Properties

While the animation is running, individual vehicles can be clicked on (use the left mouse button) to display a window that provides various details about that status of the vehicle (Figure 12). If you experience difficulty in clicking on a vehicle, try pausing the animation first.

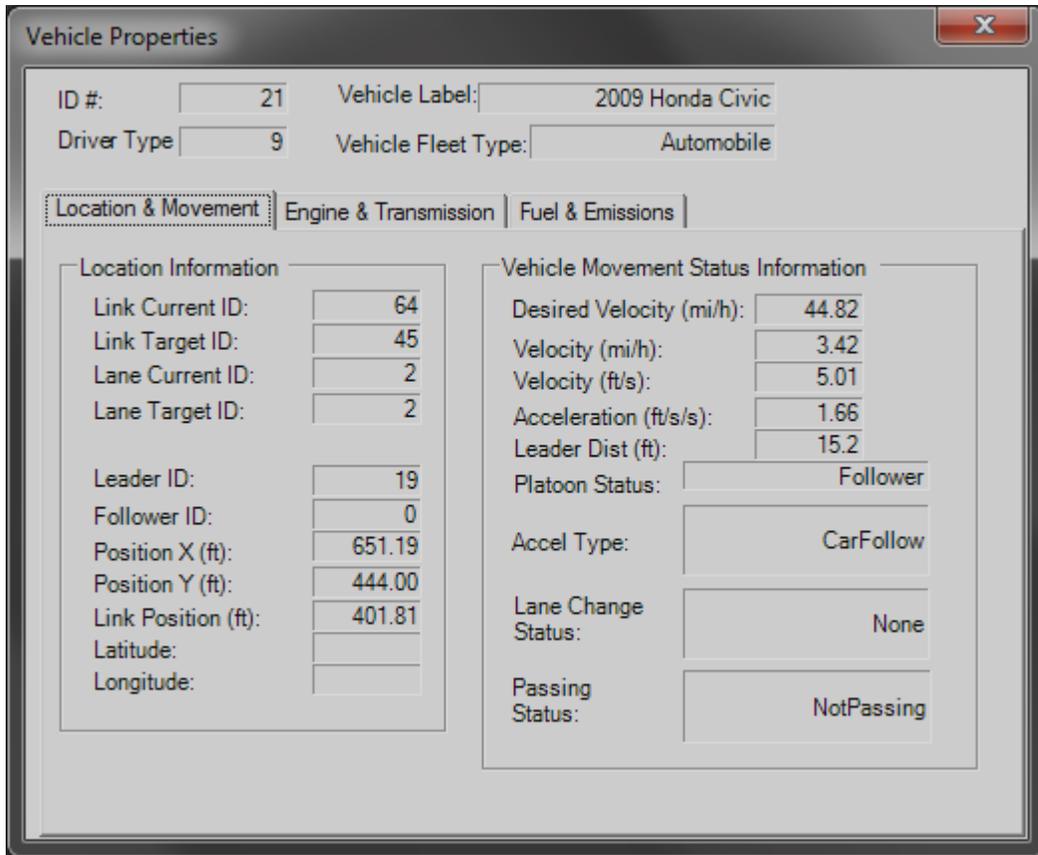
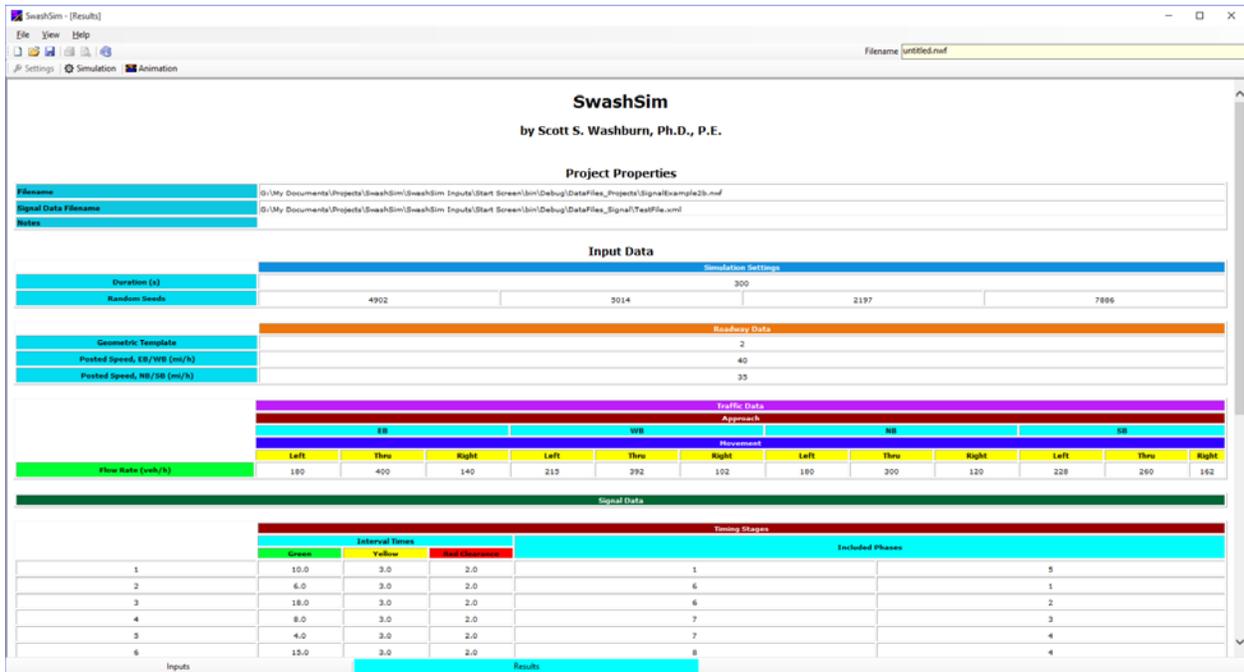


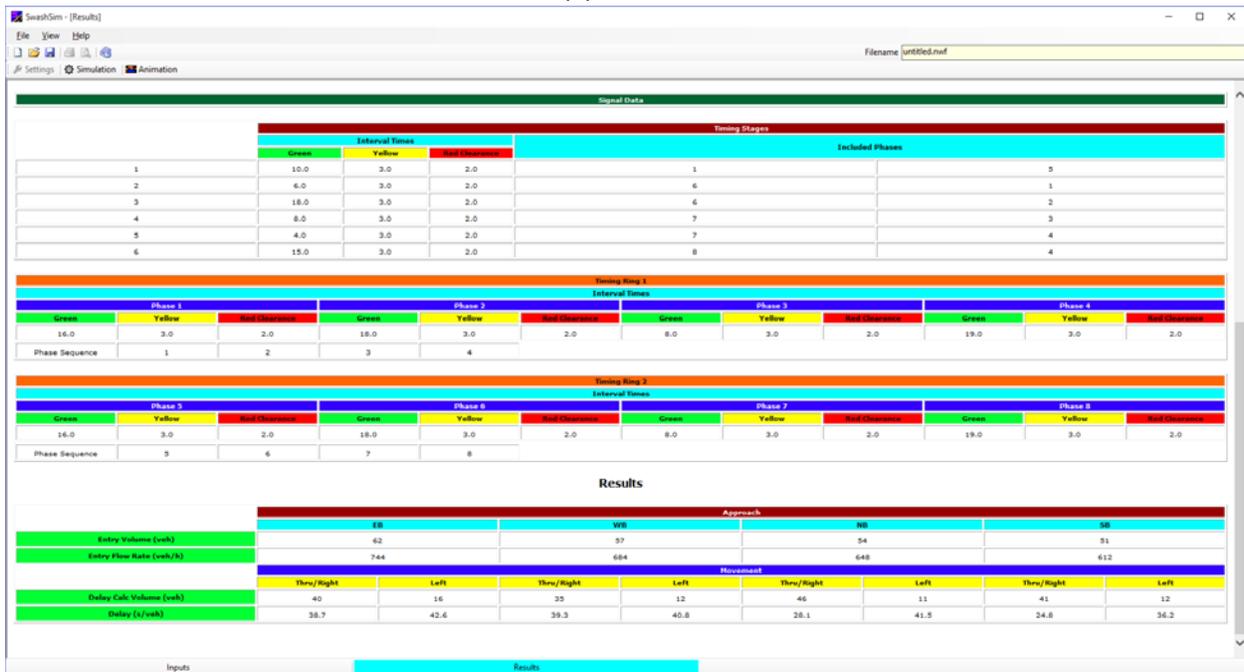
Figure 12

Results

Clicking on the 'Results' tab at the bottom of the main screen will load summary of the input values and performance measures (Figure 13).



(a)



(b)

Figure 13

Vehicle delay is simply measured as the difference in desired travel time (based on the posted speed limit) and actual travel time through the link preceding the intersection. Accordingly, this delay is calculated from just the vehicles that enter and exit the approach link during the

simulation period. Thus, the 'Delay Calc Volume' will be less than volume of vehicles that entered the network within the simulation period.

Saving a Project File

To save your project input values to a file, select 'Save' from the 'File' menu. In addition to saving the roadway, traffic, and simulation settings, a reference to the signal timing settings file will be saved, if such settings were saved to a file. The name of the saved file will be displayed in the upper-right corner of the main inputs screen (Figure 14). If the full filename is not visible in the textbox, you can click in this box with your mouse and navigate to the right with the keyboard 'right arrow' key or 'End' key. The default file extension is ".nwf".

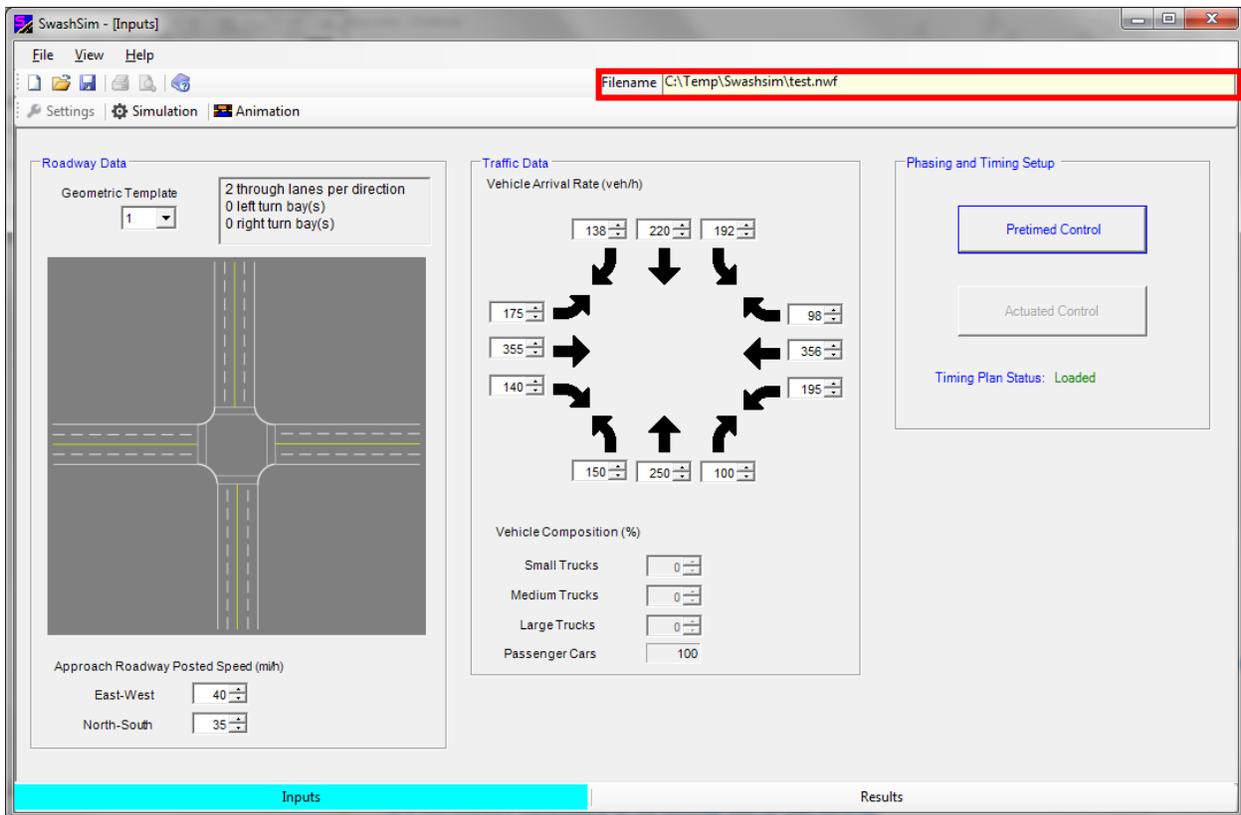


Figure 14

Current Limitations

The program currently has the following limitations:

- Permitted movements are not accommodated (this will be included in a future update).
- Only passenger cars/trucks can be modeled.

- The maximum simulation duration is 75 minutes (15 of which can be warm-up time).
- If oversaturated conditions are modeled, and vehicles queue up to the entry point of the network, the program will not restrict future vehicle entries. Thus, the simulation may crash if oversaturated conditions are modeled.
- The program does not warn you if your stage timings result in the timing rings not finishing at the same time. Thus, one timing ring may cross the barrier before the other timing ring. This issue will be addressed in a future update.
- Only pretimed control is modeled at this time, but actuated control will be added in a future update.