



UNIVERSITY OF
TEXAS
ARLINGTON

TxDOT Report 5-7160-01-1

Implementing the ATSPM-in-the-Loop Simulation Solution with the TxDOT Statewide ATSPM System Deployment Plan

Pengfei (Taylor) Li, Ph.D., P.E.,
Stephen P. Mattingly, Ph.D.,
Peirong (Slade) Wang, Ph.D.,
Swastik Khadka
Sijan Shrestha

Report Publication Date:	Submitted: July 2026 Published: September 2026
Project:	5-7160-01
Project Title:	Implementing the ATSPM-in-the-Loop Simulation Solution with the TxDOT Statewide ATSPM System Deployment Plan

1. Report No. FHWA/TX-27/5-7160-01-1	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Implementing the ATSPM-in-the-Loop Simulation Solution with the TxDOT Statewide ATSPM System Deployment Plan		5. Report Date September 2026	
		6. Performing Organization Code	
7. Author(s) Pengfei (Taylor) Li, Ph.D., P.E., Stephen P. Mattingly, Ph.D., Peirong (Slade) Wang, Ph.D., Swastik Khadka Sijan Shrestha		8. Performing Organization Report No. Report	
9. Performing Organization Name and Address The University of Texas at Arlington Arlington, Texas, 76010		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No. Project 5-7160-01	
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Implementation Office 125 E. 11 th Street Austin, Texas 78701-2483		13. Type of Report and Period Covered Technical Report: Month Year–Month Year	
		14. Sponsoring Agency Code	
15. Supplementary Notes Project sponsored by the Texas Department of Transportation and the Federal Highway Administration. Project Title: Implementing the ATSPM-in-the-Loop Simulation Solution with the TxDOT Statewide ATSPM System Deployment Plan			
16. Abstract The research team developed step-by-step installation and troubleshooting guidelines for the Utah Department of Transportation Automated Traffic Signal Performance Measures (ATSPM) system (v4.3) and the UTA-In-Motion platform. Both systems were packaged into a single Docker image, reducing installation time from about 20 hours to 1 hour while eliminating native-host compatibility issues. Controller support for the ATSPM-in-the-loop simulation package was expanded to Q-Free, Econolite, McCain, and Yunex platforms, and a hardware-in-the-loop interface was developed to connect traffic simulation software with physical traffic signal controllers and cabinets. A corridor case study in the Austin District integrated 23 intersections across multiple controller brands using built-in, software-in-the-loop, and hardware-in-the-loop simulation methods, demonstrating a complete signal-optimization workflow with ATSPM outputs. Artificial intelligence-augmented videos and courseware support instructor-led and self-paced learning.			
17. Key Words Automated Traffic Signal Performance Measures (ATSPM) System Traffic simulation Hardware-in-the-loop simulation Connected Vehicle Data		18. Distribution Statement No restrictions. This document is available to the public through NTIS: National Technical Information Service Alexandria, Virginia https://www.ntis.gov	
19. Security Classification (of this report) Unclassified	20. Security Classification. (of this page) Unclassified	21. No. of Pages 208	22. Price

**Implementing the ATSPM-in-the-Loop Simulation Solution with the TxDOT Statewide
ATSPM System Deployment Plan**

by

Pengfei (Taylor) Li, Ph.D., P.E.

Assistant Professor, Department of Civil Engineering
The University of Texas Arlington
Arlington, Texas, 76010

Stephen P. Mattingly, Ph.D.

Professor, Department of Civil Engineering
The University of Texas Arlington
Arlington, Texas, 76010

Peirong (Slade) Wang, Ph.D.

Postdoctoral Research Associate, Department of Civil Engineering
The University of Texas Arlington
Arlington, Texas, 76010

Swastik Khadka, and Sijan Shrestha

Graduate Research Assistants, Department of Civil Engineering
The University of Texas Arlington
Arlington, Texas, 76010

Project 5-7160-01

Project Title: Implementing the ATSPM-in-the-Loop Simulation Solution with the TxDOT
Statewide ATSPM System Deployment Plan

Sponsored by the
Texas Department of Transportation
and the
Federal Highway Administration

September 2026

The University of Texas Arlington
Arlington, Texas 76010

DISCLAIMER

This research was sponsored by the Texas Department of Transportation (TxDOT) and the Federal Highway Administration (FHWA). The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of FHWA or TxDOT. This report does not constitute a standard, specification, or regulation. The United States Government and the State of Texas do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

ACKNOWLEDGMENT

This project was sponsored by TxDOT and FHWA. The authors thank Mr. Dassi Martin and Dr. Jianming Ma of TxDOT for their guidance and comments on this project.

CONTENTS

Technical Report Documentation Page.....	i
DISCLAIMER	i
ACKNOWLEDGMENT.....	ii
CONTENTS.....	iii
LIST OF FIGURES	viii
LIST OF TABLES.....	xiii
LIST OF ACRONYMS	xiv
EXECUTIVE SUMMARY	xvi
CHAPTER 1: Introduction	1
1.1. Background	1
1.2. Customizing the VISSIM Model for High-Resolution ATSPM Outputs.....	1
1.2.1. Simulated traffic signal transition events output	2
1.2.2. Simulated Detector and Vehicle Trajectory Outputs	3
1.2.3. Setting Up Pedestrian Detectors and Overlaps.....	4
1.2.4. Turn on high-resolution simulation output.....	7
1.3. Q-Free MAXTIME Signal Software-in-the-Loop (SILS) Simulation for ATSPM Data Generation	11
1.3.1. Mandatory Requirements for Automated MAXTIME Launching for Multiple Intersections.....	12
1.3.2. Software-in-the-Loop Simulation Configuration with VISSIM.....	14
1.3.3. Step-by-Step Procedure for Configuring the MAXTIME Controller	20
1.4. Software Tool for Parsing VISSIM Outputs into ATSPM Databases	34
1.4.1. Running Multiple Intersections in Automation Using VBScript.....	34
1.4.2. Parsing RBC Signal Control Events from the VISSIM Database to the UDOT ATSPM Database.....	37
1.4.3. Parsing MAXTIME SILS Signal Control Events into the UDOT ATSPM Database .	39
1.4.4. Parsing Simulated Connected Vehicle Trajectories and Uploading to the UTA-In- Motion Database.....	40
1.4.5. Transferring ATSPM Data from UDOT ATSPM to UTA-In-Motion ATSPM	42
CHAPTER 2: UDOT ATSPM Installation.....	43
2.1. Introduction	43
2.2. Prerequisites	43

2.2.1. Windows Operating System	43
2.2.2. Internet Information Services (IIS).....	43
2.2.3. SQL Server and SQL Server Management Studio (SSMS)	44
2.3. Database Configuration.....	45
2.3.1. Creating a New SQL Server Login.....	45
2.3.2. Assigning Permissions	46
2.4. Installing IIS Server and Necessary Windows Packages	47
2.4.1. Enabling IIS and Required Features	47
2.5. Installing UDOT ATSPM Applications in IIS.....	48
2.5.1. Copying ATSPM Web Application to IIS	48
2.5.2. Configuring Web.Config	49
2.5.3. Configuring Application Pool.....	50
2.6. Adding UDOT ATSPM Applications in IIS	51
2.6.1. Application Configuration Summary.....	52
2.6.2. Verification of Deployment	53
2.7. Installing Backend MOE Modules in IIS	53
2.7.1. MOE Backend Deployment Overview.....	53
2.7.2. MOE Backend Application Configuration	54
2.8. Installing Backend SPM Images in IIS	55
2.8.1. SPM Images Deployment Overview	55
2.8.2. SPM Images Application Configuration.....	56
2.9. Registering UDOT ATSPM Account and Configuring SPM Image Path.....	57
2.9.1. Account Registration and Role Assignment	57
2.9.2. SPM Image Path Configuration.....	58
2.10. Testing SPM Image and MOE Services	60
2.10.1. Verification of MOE WCF Services	60
2.10.2. Validation of Database Connectivity and Service Linking.....	61
2.11. Application Configuration.....	62
2.11.1. Webpage Title Configuration.....	62
2.11.2. Logo Configuration	63
2.11.3. Default Map Location Configuration	63
2.12. Generating Additional Intersection Data.....	65

2.13. Adding Intersection at UDOT ATSPM.....	65
2.13.1. Intersection and Controller Configuration.....	66
2.13.2. Signal Phase Configuration	66
2.13.3. Detector Configuration	68
Chapter 3: UTA-In-Motion (UTAIM) Installation Manual	69
3.1. Introduction	69
3.2. Prerequisites	69
3.3. Python Installation.....	70
3.4. Installing and Configuring XAMPP and Apache Server.....	71
3.4.1. XAMPP Installation.....	71
3.4.2. Configuring XAMPP and Apache	71
3.5. Installing the MySQL Community Version.....	72
3.5.1. MySQL Installation	72
3.5.2. Product Configuration	72
3.6. Configuring the MySQL Database in Workbench	73
3.6.1. Database Schema Configuration	74
3.6.2. Data Import and Initialization.....	74
3.7. Installing UTAIM on the Apache Server.....	75
3.8. Adding Intersection to UTAIM	77
3.8.1. Adding a User	77
3.8.2. Mapping Sensors to Users	78
3.8.3. Adding Intersections	78
3.8.4. Adding Intersection Diagrams	79
3.8.5. Configuring Intersection Connections.....	80
3.8.6. Setting Intersection Distances	80
3.8.7. Configuring Traffic Signal Phases.....	81
3.8.8. Assigning Lane and Speed Information.....	82
3.9. Generating Graphs in UTAIM.....	83
3.9.1. Purdue Coordination Diagram (PCD)	83
3.9.2. Coordination (Time–Space) Diagram.....	84
Chapter 4: TxDOT ATSPM Digital Twin Development.....	86
4.0. Introduction	86

4.1. Deploying the TxDOT ATSPM Docker Image in Windows	87
4.1.1. What is Docker and Its Benefits to This Project.....	87
4.1.2. Required Hardware, Software, and Libraries	89
4.1.3. Deploying Process and Success Verification.....	92
4.1.4. Demonstrating Operations	94
4.2. Configuring the VISSIM Model to Generate ATSPM-Compliant Data	102
4.2.1. Detector Events Output	103
4.2.2. Pedestrian Detectors	104
4.2.3. Overlaps.....	105
4.2.4. Sequential Detector Layout Setup	106
4.2.5. Enabling High-Resolution Simulation Output	107
4.3. VISSIM with Econolite EOS Control Emulator	111
4.3.1 Adding an Econolite EOS Signal Controller in VISSIM	111
4.3.2. Web Front Panel Configuration	112
4.3.3. Main Menu (MM) and Sub Menu	113
4.3.4. Controller Settings.....	115
4.3.5. Detector Setup in VISSIM and EOS (MM-6).....	129
4.3.6. VISSIM Signal Head Configuration.....	131
4.3.7. Stepwise Guideline for Setting Up a Coordination Plan (Event Plan)	132
4.4. Parsing the VISSIM and EOS Output to the ATSPM Database.....	133
4.4.1. Parsing RBC Signal Events: VISSIM to UDOT ATSPM.....	133
4.4.2. Data Generation and Parsing of EOS Events into UDOT ATSPM	134
4.4.3. Transferring ATSPM Data: UDOT ATSPM to UTAIM	136
4.4.4. Generating Figures from UDOT ATSPM and UTAIM	137
Chapter 5: Hardware-in-the-Loop Simulation Control Interface Device	139
5.1. VISSIM Setup For HILS.....	139
5.1.1. Prerequisites.....	139
5.1.2. Set up an HILS intersection in VISSIM	140
5.1.3. Set up HILS signal phases and detectors in VISSIM	141
5.2. Signal controller set up for HILS	142
5.3. HILS-CID Windows Application	143
5.3.1. Overlap Configuration.....	144

5.3.2. Parameter Management and User Workflow	145
5.3.3. Connecting to VISSIM	146
5.3.4. Summary of the Operational Sequence	147
5.4. Chapter Summary.....	147
Chapter 6: Case Study.....	149
6.1. Introduction	149
6.2. Evaluation of the Baseline Traffic Signal Performance using the ATSPM-in-the-loop Traffic Simulation Model	149
6.2.1. Setting Up HILS and SILS in VISSIM	153
6.2.2. Analysis of Performance Metrics from VISSIM	158
6.2.3. Comparison of Measure of Effectiveness (MOEs) from VISSIM and UDOT ATSPM	159
6.2.4. Analysis of performance metrics from UDOT-ATSPM	161
6.2.5. Analysis of performance metrics from UTAIM.....	168
6.3. Analyzing Traffic Signal Performance with Projected Travel Demand growth using ATSPM-in-loop simulation	170
6.3.1. Analysis of performance metrics from VISSIM.....	171
6.3.2. Analysis of performance metrics from UDOT-ATSPM	173
6.3.3. Analysis of performance metrics from UTAIM.....	178
6.4. Analyzing Traffic Signal Performance with Optimized Traffic Signal.....	180
6.5. DISCUSSION	185
APPENDIX.....	185

LIST OF FIGURES

Figure 1-1: Snapshot of Indiana traffic signal events logging enumerations.....	2
Figure 1-2: Comparison between VISSIM and ATSPM signal event formats.....	3
Figure 1-3: VISSIM's detector events output.	3
Figure 1-4: Snapshot of simulated vehicle trajectories in the WGS84 format.....	4
Figure 1-5: Pedestrian detectors in VISSIM	5
Figure 1-6: Overlaps setup in RBC.....	6
Figure 1-7: RBC detector actuation outputs and decoding.	7
Figure 1-8: Loading the UDA for the WGS84 vehicle trajectory output.	8
Figure 1-9: Enabling detector actuation outputs for each intersection.....	9
Figure 1-10: Connecting VISSIM to an SQL Server database.....	10
Figure 1-11: VISSIM simulation output evaluation configuration.	10
Figure 1-12: SQL Server database for VISSIM output.....	11
Figure 1-13: MAXTIME DLL download page for PTV VISSIM compatibility	15
Figure 1-14: Downloaded files for MAXTIME integration	15
Figure 1-15: Destination path for the DLL/WTT files.....	16
Figure 1-16: PTV VISSIM Signal Controller Interface Snapshot.	16
Figure 1-17: PTV VISSIM Signal Controller Dialog Box.....	17
Figure 1-18: PTV VISSIM Standard Signal Controller Parameters Dialog Box.....	18
Figure 1-19: VISSIM signal control group setup.....	19
Figure 1-20: VISSIM-MAXTIME integration setup.	19
Figure 1-21: Q-Free MAXTIME Application Launcher Snapshot.	20
Figure 1-22: Highlighted Database Type in Q-Free MAXTIME Application Snapshot.....	21
Figure 1-23: Snapshot Displaying List of Factory Setting Defaults in Q-Free MAXTIME.....	21
Figure 1-24: Snapshot Displaying List of User-Created Databases in Q-Free MAXTIME.	22
Figure 1-25: Q-Free MAXTIME Application Launcher with advanced settings on.	23
Figure 1-26: MAXTIME Welcome Page Display.....	23
Figure 1-27: MAXTIME User Groups Display.	24
Figure 1-28: MAXTIME profile mode Display.....	24
Figure 1-29: MAXTIME Date and Time Setup Display.....	25
Figure 1-30: Main Display (front panel) of MAXTIME Controller Window.....	25
Figure 1-31: Advanced Settings Option in MAXTIME with Multiple Inputs.....	27
Figure 1-32: MAXTIME controller status interface.	27
Figure 1-33: Window for Changing Controller Name and Unit Information in MAXTIME	28
Figure 1-34: Advanced Input-Output (Akhtar & Moridpour) Module Type Configuration in MAXTIME.....	29
Figure 1-35: Advanced Input-Output Channel Configuration Page in MAXTIME	29
Figure 1-36: Vehicle and Pedestrian Detector Layout.	30
Figure 1-37: Input Points Page for Different Input Control Types in MAXTIME	31
Figure 1-38: IO Channel Configuration for Overlaps in MAXTIME.....	31
Figure 1-39: Overlaps Configuration in MAXTIME.....	32
Figure 1-40: Intersection Layout with Different Phases for Overlaps in VISSIM.....	33
Figure 1-41: Status and Performance Tab Display in MAXTIME.	33
Figure 1-42: Multiple MAXTIME controllers coupled with multiple intersections in PTV VISSIM.....	35

Figure 1-43: VBScript for automating MAXTIME configuration for multiple controllers.....	35
Figure 1-44: VBScript automation parameters.	36
Figure 1-45: Warning indicating that user input should be avoided during VBScript execution.	37
Figure 1-46: UDOT-ATSPM and VISSIM output DBs displayed in SQL Server Management Studio.	37
Figure 1-47: Workflow of ATSPM-in-the-loop simulation based on VISSIM+RBC.	38
Figure 1-48: The local JSON file saves the parsing configuration.	38
Figure 1-49: Loading the parsing configurations from the local JSON file.....	39
Figure 1-50: Workflow of ATSPM-in-the-loop simulation based on VISSIM+MAXTIME_SILS.....	40
Figure 1-51: Retrieving the ATSPM traffic signal events from the MAXTIME SILS	40
Figure 1-52: Transforming simulated CV trajectories to match the UTAIM system.....	41
Figure 1-53: Transferring the ATSPM data from the UDOT ATSPM database to the UTAIM database. ..	42
Figure 2-1: SQL Server installation process showing the installer's initial setup screen.	44
Figure 2-2: SQL Server Management Studio installation process.....	44
Figure 2-3: Verification of the MSSQLSERVER service running.....	45
Figure 2-4: Initial connection dialog for SQL Server login configuration.....	45
Figure 2-5: Creation of the SPM SQL Server login.....	46
Figure 2-6: User mapping and database role configuration for the SPM login.	46
Figure 2-7: Confirmation of the completed SQL Server login configuration.....	47
Figure 2-8: Enabling IIS and required Windows features in the Windows Features dialog.	48
Figure 2-9: Placement of the ATSPM-Web-Rebuild folder in the IIS root directory.....	49
Figure 2-10: Folder permission configuration for IIS system accounts.....	49
Figure 2-11: Updated Web.config file showing the ATSPM database connection string.	50
Figure 2-12: Creation of a dedicated IIS application pool for ATSPM.....	50
Figure 2-13: Application pool settings with 32-bit applications enabled.....	51
Figure 2-14: Illustrates the IIS interface used to add the ATSPM application to the web server.....	52
Figure 2-15: Shows the application settings used to link the alias, application pool, and physical path...	52
Figure 2-16: Presents the ATSPM web application interface displayed after successful deployment.....	53
Figure 2-17: MOEWcfServiceLibrary folder and permissions.....	54
Figure 2-18: Shows the MOEWcfServiceLibrary application registered within IIS.....	55
Figure 2-19: SPMImages folder creation and permission configuration.	56
Figure 2-20: Shows the SPMImages directory registered as an application within IIS.....	57
Figure 2-21: Shows the user registration interface used to create a new ATSPM account.	57
Figure 2-22: Illustrates a successful login using the default administrator account.	58
Figure 2-23: Assigning roles to the newly registered ATSPM account within the admin section.	59
Figure 2-24: SPM image path and URL settings.	59
Figure 2-25: Testing MOE WCF service endpoints showing the service status page in a browser.....	60
Figure 2-26: Web.config connection-string updates for backend integration.	61
Figure 2-27: Endpoint address updated based on metric service generator.	62
Figure 2-28: Editing the homepage title to reflect the desired webpage title.	63
Figure 2-29: Illustrates the process of replacing the default logo image within the application directory.	63
Figure 2-30: Default map location configuration in Map.js.	64
Figure 2-31: Updated ATSPM map view.	64
Figure 2-32: GenerateAddDataScript execution.....	65
Figure 2-33: Illustrates the interface used to create a new signal entry.	66

Figure 2-34: Detailed view of the signal phase configuration for a new intersection signal.	67
Figure 2-35: Distance to stop bar measured from VISSIM	68
Figure 3-1: Python installer setup window initiating the installation process.	70
Figure 3-2: XAMPP Control Panel displaying Apache configuration with custom port settings.	71
Figure 3-3: MySQL Installer community version setup screen during a full installation process.	72
Figure 3-4: MySQL Installer community version password setup	73
Figure 3-5: Setting up a new connection in MySQL Workbench	73
Figure 3-6: Setting up new schemas (utaim and utaim_config) for the database.	74
Figure 3-7: MySQL Workbench interface displaying the created schemas and imported data details.	75
Figure 3-8: UTAIM application files deployed in the Apache htdocs directory.	76
Figure 3-9: UTAIM main interface is displayed after a successful login using the provided credentials. .	76
Figure 3-10: Adding a new user to the UTAIM system.	77
Figure 3-11: Mapping sensors to users for real-time travel time analysis.	78
Figure 3-12: Adding intersections in UTAIM, including assigning IDs and geographic coordinates.	79
Figure 3-13: Uploading intersection diagrams to the database	79
Figure 3-14: Configuring intersection connections to define directional relationships.	80
Figure 3-15: Setting intersection distances for accurate travel time calculations.	81
Figure 3-16: Configuring traffic signal phases with assigned detectors.	82
Figure 3-17: Assigning Lane numbers and speed limits for intersections.	83
Figure 3-18: PCD for analyzing intersection performance.	84
Figure 3-19: TSD for Northbound Cooper Street.	85
Figure 4-1: Example of a Docker Container	88
Figure 4-2: Proposed digital twin with Docker containers	89
Figure 4-3: Windows version specifications (Windows 11 Pro shown in the figure).....	90
Figure 4-4: Successfully Enabled Hyper-V and Container (input n for refusing restart)	90
Figure 4-5: Enable Windows containers when installing Docker Desktop.....	91
Figure 4-6: Installing WSL (Windows Subsystem for Linux)	91
Figure 4-7: Switch to Windows Container in Docker Desktop.....	92
Figure 4-8: Successful loading of udot spm docker tar file	93
Figure 4-9: Successfully built Docker status	93
Figure 4-10: Config Image Path and Image URL in the UDOT system.....	94
Figure 4-11: Running Docker Containers in the Docker Desktop	95
Figure 4-12: Login SQL Server of UDOT SPM	96
Figure 4-13: Login MySQL of UTAIM SPM	97
Figure 4-14: Restore Data in SQL SERVER	98
Figure 4-15: Backup Data in SQL SERVER	99
Figure 4-16: Backup Data in MySQL Workbench.....	100
Figure 4-17: Restore Data in MySQL Workbench.....	101
Figure 4-18: Signal List in UDOT SPM system	101
Figure 4-19: Successfully Run of GenerateAddDataScripts for intersection markers on the map.	102
Figure 4-20: Example of .ldp detector actuation file format used in VISSIM.....	104
Figure 4-21: Pedestrian detector placement and numbering convention in VISSIM.....	105
Figure 4-22: Example of overlap configuration using RBC in VISSIM.....	105
Figure 4-23: RBC detector actuation outputs and decoding	106

Figure 4-24: Configuring WGS84 coordinate output in VISSIM.....	107
Figure 4-25: SC Detector Record configuration in VISSIM.....	108
Figure 4-26: Database configuration interface in VISSIM for SQL Server output.....	109
Figure 4-27: Output evaluation setup.....	110
Figure 4-28: Consolidated simulation outputs stored in SQL Server for ATSPM.	110
Figure 4-29: EOS Signal Controller in PTV VISSIM.....	112
Figure 4-30: Econolite EOS Web Front Panel interface in VISSIM.....	113
Figure 4-31: Main menu navigation in Econolite EOS controller.	115
Figure 4-32: Controller Submenu in Econolite.....	115
Figure 4-33: Timing plan setup screen (MM-2-1) in EOS controller.	116
Figure 4-34: Minimum Green interval configuration (MM-2-1-1).....	117
Figure 4-35: Passage time configuration (MM-2-1-2).....	118
Figure 4-36: Maximum Green interval configuration (MM-2-1-3).....	119
Figure 4-37: Pedestrian timing configuration (MM-2-1-4).	120
Figure 4-38: Clearance interval configuration (MM-2-1-5).	121
Figure 4-39: Phase recall configuration (MM-2-1-6).	123
Figure 4-40: Overlap timing configuration (MM-2-1-7).	124
Figure 4-41: Vehicle overlap configuration and deployment.....	126
Figure 4-42: Phase Options for managing phase behaviors.....	127
Figure 4-43: Sequence submenu in the EOS controller for phase order and barriers.	127
Figure 4-44: Sequence configuration setup (MM-2-7-1).....	128
Figure 4-45: Vehicle detector assignment screen in the EOS controller.....	130
Figure 4-46: Advance detector placement and configuration in VISSIM.....	131
Figure 4-47: VISSIM signal head organization for vehicle, pedestrian, and overlap groups.	132
Figure 4-48: Event plan setup for coordination in EOS scheduler.....	133
Figure 4-49: Validation of parsed simulation events in UDOT-ATSPM.....	134
Figure 4-50: RBC parsing configuration: VISSIM to UDOT ATSPM.	134
Figure 4-51: Econolite EOS event logging configuration screen.	135
Figure 4-52: High-resolution data logging options in EOS controller.....	136
Figure 4-53: Data transfer process from UDOT ATSPM to UTAIM.....	137
Figure 5-1: License summary in VISSIM.....	139
Figure 5-2: Set up external signal control in VISSIM for HILS.....	140
Figure 5-3: Set up IP address, port, and signal phases using STDSC_GUI.dll.....	141
Figure 5-4: Set up signal heads and detectors in VISSIM	142
Figure 5-5: Signal controller set up for HILS	143
Figure 5-6: Graphic User Interface of HILS-CID.....	144
Figure 5-7: Signal Phase display.....	146
Figure 5-8: Detector and pedestrian call status in HILS-CID.....	147
Figure 6-1: Austin RM 2244 corridor	150
Figure 6-2: Flow balance using vehicle inputs and exits between intersections.....	150
Figure 6-3: RM 2244 Zone division along the corridor.....	151
Figure 6-4: RBC controller with the corresponding input parameters.....	153
Figure 6-5: Econolite EOS web front panel showing controller status.....	154
Figure 6-6: VISSIM to UDOT ATSPM parser interface for post-processing EOS event data.....	155

Figure 6-7: HILS operation using the physical controller front panel, web portal, and VISSIM model. .	156
Figure 6-8: HILS Interface for connecting the MAXTIME controller to VISSIM.	157
Figure 6-9: Real-time event parsing interface for capturing HILS controller events during simulation. .	157
Figure 6-10: Approach volume for RM 2244 at Rollingwood Town from UDOT ATSPM (A) and VISSIM (B).....	160
Figure 6-11: Approach delay for RM 2244 at Barton Creek Blvd from UDOT ATSPM (A) and VISSIM (B).....	161
Figure 6-12: Approach volume for RM 2244 at Rollingwood Town.	164
Figure 6-13: Approach delay for SB (Phase 2) and NB (Phase 6) for RM 2244 at Rollingwood Town..	165
Figure 6-14: Arrival on Red for SB (Phase 2) and NB (Phase 6) for RM 2244 at Rollingwood Town. ..	166
Figure 6-15: PCD for SB (Phase 2) and NB (Phase 6) for RM 2244 at Rollingwood Town.	167
Figure 6-16: PCDs for RM 2244 at Barton Creek Blvd from UTAIM (A) and UDOT ATSPM (B).....	169
Figure 6-17: TSD for RM 2244 corridor (Intersection 2 to 9).	170
Figure 6-18: Comparison of average queue length before and after volume increase at Intersections. ...	172
Figure 6-19: Comparison of average vehicle delay before and after volume increase at Intersections....	173
Figure 6-20: Comparison of approach volume before and after for RM 2244 at Rollingwood Town.	175
Figure 6-21: Comparison of approach delay before and after (phase 6) for RM 2244 at Rollingwood Town.	176
Figure 6-22: Comparison of arrival on red before and after (phase 2) at RM 2244 at Rollingwood Town.	177
Figure 6-23: Comparison of PCD before and after (phase 2) for RM 2244 at Rollingwood Town.	178
Figure 6-24: Comparison of PCD before and after (phase 2) for RM 2244 at Barton Creek Blvd.	179
Figure 6-25: TSD for RM 2244 corridor under increased traffic volume.....	180
Figure 6-26: Average queue length after signal timing optimization.	181
Figure 6-27: Average vehicle delay after signal timing optimization.....	182
Figure 6-28: TSD for the optimized RM 2244 corridor under increased traffic volume.....	184
Figure 7-1: Approach Volume at Barton Creek Blvd.	186
Figure 7-2: Arrival on Red for RM 2244 at Barton Creek Blvd.	187
Figure 7-3: Approach Delay at Barton Creek Blvd.	187
Figure 7-4: Approach Volume at Senna Hills.....	188
Figure 7-5: Arrival on Red at Senna Hills.	188
Figure 7-6: Approach delay at Senna Hills.....	189
Figure 7-7: PCD for RM 2244 at Senna Hills from UDOT ATSPM.....	189

LIST OF TABLES

Table 6-1: List of intersection with the representative intersection	152
Table 6-2: Results obtained from VISSIM.....	158
Table 6-3: Performance Metrics obtained from UDOT-ATSPM.	159
Table 6-4: Performance Metrics for the Chosen Intersection.	162
Table 6-5: Results obtained from VISSIM.....	171
Table 6-6: Performance Metrics for the Chosen Intersection.	173
Table 6-7: Level of Service obtained from VISSIM.	182

LIST OF ACRONYMS

AoC	Arrival on Coordination
AoG	Arrivals on Green
AoR	Arrivals on Red
AI	Artificial Intelligence
API	Application Programming Interface
ATC	Advanced Transportation Controller
ATCC	Advanced Transportation Controller Cabinet
ATSPM	Automated Traffic Signal Performance Measures
AWF	Advance Warning Flasher
BA	Band Attainability
BIU	Bus Interface Unit
BLOB	Binary Large Object
CID	Control Interface Device
CILS	Cabinet-in-the-Loop Simulation
CMU	Conflict Monitor Unit
CV	Connected Vehicle
CSV	Comma-Separated Values
DB	Database
DLL	Dynamic-Link Library
EB	Eastbound
EULA	End-User License Agreement
FHWA	Federal Highway Administration
FYA	Flashing Yellow Arrow
GT	Green Time
GUI	Graphical User Interface
HAWK	High-Intensity Activated Crosswalk
HILS	Hardware-in-the-Loop Simulation
HILS-CID	Hardware-in-the-Loop Simulation Control Interface Device
IIS	Internet Information Services
INDOT	Indiana Department of Transportation
I/O	Input/Output
IP	Internet Protocol
ITS	Intelligent Transportation Systems
JSON	JavaScript Object Notation
LPI	Leading Pedestrian Interval
LOS	Level of Service
MM	Main Menu
MMU	Malfunction Management Unit
MOE	Measure of Effectiveness

NB	Northbound
NEMA	National Electrical Manufacturers Association
NTCIP	National Transportation Communications for ITS Protocol
PCD	Purdue Coordination Diagram
PHF	Peak Hour Factor
PR	Platoon Ratio
PTV	Planung Transport Verkehr
RBC	Ring-Barrier Controller
RM	Ranch-to-Market Road
SB	Southbound
SILS	Software-in-the-Loop Simulation
SIU	Serial Interface Unit
SPM	Signal Performance Measure
SQL	Structured Query Language
SSMS	SQL Server Management Studio
TCP	Transmission Control Protocol
TOD	Time of Day
TRL	Technology Readiness Level
TSD	Time-Space Diagram
TTI	Time to Intersection
TxDOT	Texas Department of Transportation
UDA	User-Defined Attribute
UDOT	Utah Department of Transportation
UTA	University of Texas at Arlington
UTAIM	UTA-In-Motion
URL	Uniform Resource Locator
VIOT	Virtual Input/Output Trace
VISSIM	Verkehr In Städten – SIMulationsmodell
WB	Westbound
WCF	Windows Communication Foundation
WGS84	World Geodetic System 1984
WSL	Windows Subsystem for Linux
XAMPP	Cross-Platform Apache, MariaDB/MySQL, PHP, and Perl package
XML	Extensible Markup Language

EXECUTIVE SUMMARY

TxDOT RTI 5-7160: Implementing the ATSPM-in-the-Loop Simulation Solution with the TxDOT Statewide ATSPM System Deployment Plan

This implementation project builds directly on the research project TxDOT RTI 0-7160, “Improve Traffic Signal System Planning, Design and Management with Big-Data-Enhanced Automated Traffic Signal Performance Metrics (ATSPM) System.” It advances the Technology Readiness Level (TRL) of the developed solutions from TRL 7 to TRL 9, enabling operational deployment in real-world environments.

To support independent adoption by TxDOT engineers, the research team produced detailed step-by-step installation guidelines for two open-source ATSPM systems: Utah-DOT-ATSPM (v4.3) and the UTA add-on system UTA-In-Motion (UTAIM). Drawing on prior deployment experience, the guidelines include extensive troubleshooting guidance and enable reliable installation and operation on both dedicated servers and individual workstations.

A second major outcome is the adoption of modern software distribution practices. Both ATSPM systems were packaged into a single Docker image. This approach reduces fresh installation time from approximately 20 hours to about 1 hour and eliminates longstanding software-compatibility issues associated with native host installations.

The project also expanded controller support within the ATSPM-in-the-loop simulation package to include Q-Free, Econolite, McCain, and Yunex platforms. This broader compatibility allows most TxDOT districts to apply the solution. In addition, a National Transportation Communications for ITS Protocol (NTCIP)-based hardware-in-the-loop simulation control interface device (HILS-CID) was delivered, enabling connection of Verkehr In Städten – SIMulationsmodell (VISSIM), translated as “Traffic in Cities – Simulation Model,” to physical traffic signal controllers as well as National Electrical Manufacturers Association (NEMA) TS-2 and Advanced Transportation Controller (ATC) cabinets.

In collaboration with the TxDOT Austin District, a comprehensive case study was conducted along a busy corridor containing 23 intersections. The study integrated multiple controller brands using Ring-Barrier Controller (RBC), software-in-the-loop simulation (SILS), hardware-in-the-loop simulation (HILS), and cabinet-in-the-loop simulation (CILS). By combining a standard signal-optimization workflow with ATSPM outputs, the case study demonstrates a complete process for translating the research outcomes into practical applications.

Finally, the project produced a suite of Artificial Intelligence (AI)-augmented training videos that introduce new users to ATSPM-in-the-loop simulation concepts, together with complete courseware supporting both in-person instruction and self-paced learning exercises.

CHAPTER 1: Introduction

1.1. Background

Building on the findings of the Texas Department of Transportation (TxDOT) Research Project 0-7160, *Improving Traffic Signal System Planning, Design, and Management with Big-Data-Enhanced ATSPM System*, Project 5-7160 focuses on implementing an ATSPM-in-the-loop simulation framework that supports ATSPM data generation, database integration, and performance visualization. This chapter describes the configuration of Planung Transport Verkehr (PTV) VISSIM, derived from Verkehr In Städten – SIMulationsmodell (“Traffic in Cities – Simulation Model”) outputs, traffic signal controller emulators, and data parsing tools used to produce ATSPM-compatible signal event, detector actuation, and vehicle trajectory datasets.

Automated Traffic Signal Performance Measures (ATSPM) systems collect and process high-resolution traffic signal controller and detector data to support continuous evaluation of traffic signal operations. These systems have been developed and advanced through collaboration among federal and state agencies, academic institutions, and industry partners. Despite their growing adoption, agencies continue to face implementation challenges, including differences between traditional signal design practices and ATSPM-based performance evaluation, substantial deployment resource requirements, and operational and maintenance complexity.

To support ATSPM implementation, this report documents procedures for generating high-resolution traffic signal event data and vehicle trajectories from PTV VISSIM in World Geodetic System 1984 (WGS84) format. For Project 5-7160, the report describes the configuration of the built-in Ring-Barrier Controller (RBC) and the Econolite EOS controller emulator for simulated traffic signal control. Q-Free MAXTIME Software-in-the-Loop Simulation (SILS), developed and applied under Project 0-7160, is retained in this comprehensive final report for documentation and was not used for the current Project 5-7160 corridor implementation. The report also documents hardware-in-the-loop simulation (HILS) using a physical Q-Free MAXTIME controller connected to VISSIM through the TxDOT Hardware-in-the-Loop Simulation Control Interface Device (HILS-CID). In addition, it presents procedures for converting simulated outputs into ATSPM-compatible data structures and transferring data between Microsoft SQL Server and MySQL databases. Together, these procedures establish the data flow required to analyze and visualize simulated traffic signal performance using the Utah Department of Transportation ATSPM platform (UDOT ATSPM) and UTA-In-Motion (UTAIM).

1.2. Customizing the VISSIM Model for High-Resolution ATSPM Outputs

PTV VISSIM can generate records of traffic signal phase transitions, detector states, and vehicle movements. However, its native output formats differ from the event-code structure and trajectory formats required by ATSPM platforms. Therefore, the simulated outputs must be configured and converted before they can be stored and analyzed in UDOT ATSPM and UTAIM. This section describes the required VISSIM output settings and data-conversion procedures.

Traffic signal event data used by the two ATSPM platforms—UDOT ATSPM and UTAIM—must conform to the standardized event codes established by the Indiana Department of Transportation (INDOT) in collaboration with Purdue University. The standard includes more than 100 event codes, each representing a specific signal or detector state change, as illustrated in Fig. 1-1. For most events, an additional parameter identifies the associated phase or detector channel. This standardized coding structure supports consistent data processing, integration, and comparison across the two platforms.

Event Code	Event Descriptor	Parameter	Description
Active Phase Events:			
0	Phase On	Phase # (1-16)	Set when NEMA Phase On becomes active, either upon start of green or walk interval, whichever occurs first.
1	Phase Begin Green	Phase # (1-16)	Set when either solid or flashing green indication has begun. Do not set repeatedly during flashing operation.
2	Phase Check	Phase # (1-16)	Set when a conflicting call is registered against the active phase. (Marks beginning of MAX timing)
3	Phase Min Complete	Phase # (1-16)	Set when phase min timer expires.
4	Phase Gap Out	Phase # (1-16)	Set when phase gaps out, but may not necessarily occur upon phase termination. Event may be set multiple times within a single green under simultaneous gap out.
5	Phase Max Out	Phase # (1-16)	Set when phase MAX timer expires, but may not necessarily occur upon phase termination due to last car passage or other features.

Figure 1-1: Snapshot of Indiana traffic signal events logging enumerations.

In field deployments, ATSPM platforms periodically retrieve high-resolution event data from traffic signal controllers. The retrieved records are decoded and stored in the corresponding databases: Microsoft SQL Server for UDOT ATSPM and MySQL for UTAIM. These databases support subsequent processing, analysis, and visualization of traffic signal performance.

1.2.1. Simulated traffic signal transition events output

It is relatively straightforward to store simulated traffic signal group events generated by VISSIM directly in a SQL Server database; however, all simulated events must be parsed from their native VISSIM format into an ATSPM-compliant event structure, as illustrated in Fig. 1-2.

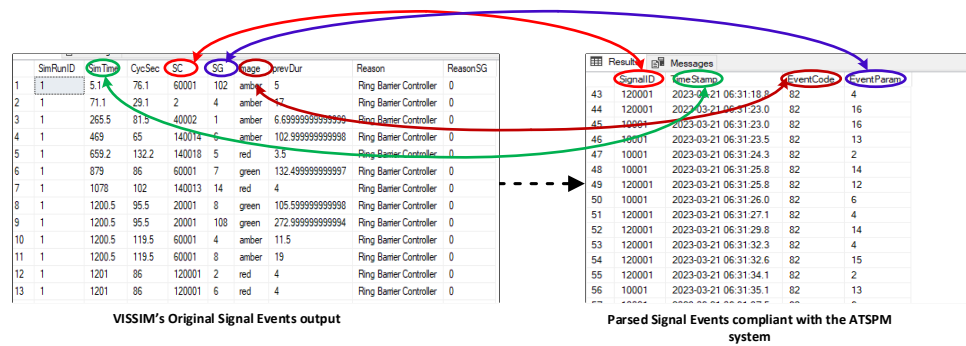


Figure 1-2: Comparison between VISSIM and ATSPM signal event formats.

1.2.2. Simulated Detector and Vehicle Trajectory Outputs

As of VISSIM version 2022, the simulation engine cannot directly output detector actuation events to a database; instead, detector data are exported as text files with the extension “.ldp”. After each simulation run, VISSIM automatically generates one detector output file per intersection. File names follow the format “XXX_YY_ZZZ.ldp”, where *XXX* denotes the VISSIM model name, *YY* represents the intersection ID within the model, and *ZZZ* indicates the simulation run number.

Within each file, detector status at each simulation time step is encoded as a character string. As illustrated in Fig. 1-3, the character “.” represents an *OFF* detector state, while all other characters indicate an *ON* state. The detector status records for each intersection are processed sequentially, with status transitions identified and translated into ATSPM-compliant detector events.

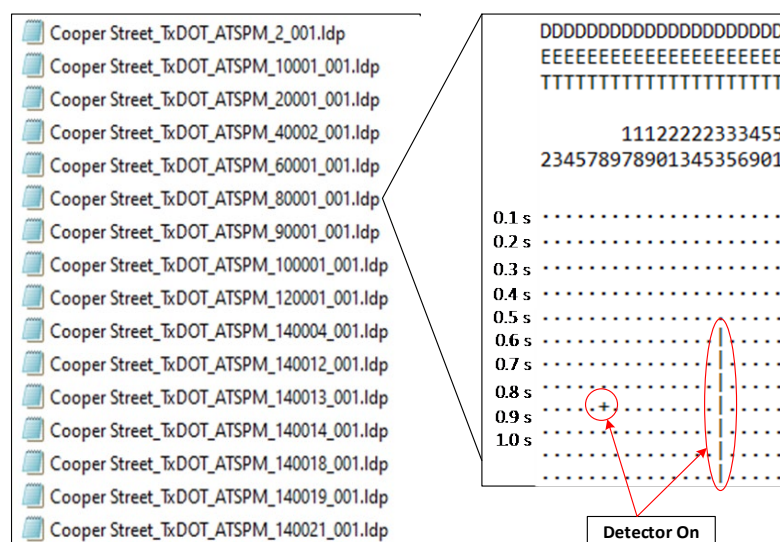


Figure 1-3: VISSIM's detector events output.

Vehicle trajectory output is a standard function in VISSIM. However, the native trajectory output is provided in the form (x, y, t) , where x and y represent a vehicle's relative position within the simulation coordinate system, and t denotes the simulation time. Because these coordinates are referenced to the simulation model rather than real-world geography, simulated vehicle trajectories cannot be directly ingested into operational ATSPM databases. Additional geographic and temporal information is required to make the simulated trajectories actionable.

Recent versions of VISSIM (version 11 and later) provide a feature referred to as User-Defined Attributes (UDA), which allows users to customize simulation outputs. PTV provides example implementations demonstrating how the UDA feature can be used to transform relative (x, y) coordinates into real-world WGS84 (latitude, longitude) coordinates based on the model's geolocation reference. In addition to spatial transformation, a real-world timestamp must be assigned to each trajectory point. While VISSIM can output simulated time steps and time-of-day values, an explicit calendar date must also be specified to ensure that simulated vehicle trajectories comply with ATSPM data requirements. An example of the resulting trajectory records in WGS84 format is shown in Fig. 1-4.

	SIMRUN	NO	SIMSEC	SIMTMOFDAY	LATITUDEFRONT	LONGITUDEFRONT	SPEED
1	1	1	0.2	07:15:00.20	32.7269243537268	-97.115990739113	9.11533486809736
2	1	1	0.3	07:15:00.30	32.7269282360769	-97.1159821353471	9.11533486809736
3	1	2	0.3	07:15:00.30	32.7283996083935	-97.1154053897013	9.55631645693473
4	1	1	0.4	07:15:00.40	32.7269321141677	-97.11597354102	9.09533486809736
5	1	2	0.4	07:15:00.40	32.7283996144147	-97.115395175956	9.55631645693473
6	1	1	0.5	07:15:00.50	32.7269359799026	-97.1159649740746	9.05731513503856
7	1	2	0.5	07:15:00.50	32.7283996204359	-97.1153849622107	9.55631645693473
8	1	3	0.5	07:15:00.50	32.7391780947863	-97.1123069496779	15.5159431431886
9	1	1	0.6	07:15:00.60	32.7269398294449	-97.1159564430136	9.01929875132546
10	1	2	0.6	07:15:00.60	32.7283996264571	-97.1153747484654	9.55631645693473
11	1	3	0.6	07:15:00.60	32.7391779317901	-97.1123235105382	15.4759431431886
12	1	4	0.6	07:15:00.60	32.6755642901307	-97.1347428699463	18.9463765247932
13	1	1	0.7	07:15:00.70	32.726943662796	-97.115947947834	8.98128574435755

Figure 1-4: Snapshot of simulated vehicle trajectories in the WGS84 format.

1.2.3. Setting Up Pedestrian Detectors and Overlaps

Configuring Pedestrian Detectors in RBC

Pedestrian detectors are used in traffic signal systems to register pedestrian calls at crosswalks and initiate pedestrian signal phases. In the simulation environment, pedestrian detectors are configured to replicate field-installed push buttons or sensing devices and must be correctly mapped to the signal controller logic to ensure proper operation.

In the RBC controller, pedestrian detectors must be assigned using the 1XX numbering convention to ensure that simulation outputs are compatible with the ATSPM event format. In this convention, XX represents the associated concurrent vehicle phases for the pedestrian movement. For example, when a pedestrian crosses the major roadway, the corresponding pedestrian detector should be numbered 104 or 108, as vehicle phases 04 and 08 represent the through movements on the minor street.

Fig. 1-5 illustrates an example pedestrian detector configuration in VISSIM and its corresponding mapping in the RBC. This mapping ensures consistency between simulated pedestrian calls and controller phase operations, enabling accurate representation of pedestrian movements and signal responses within the simulation.

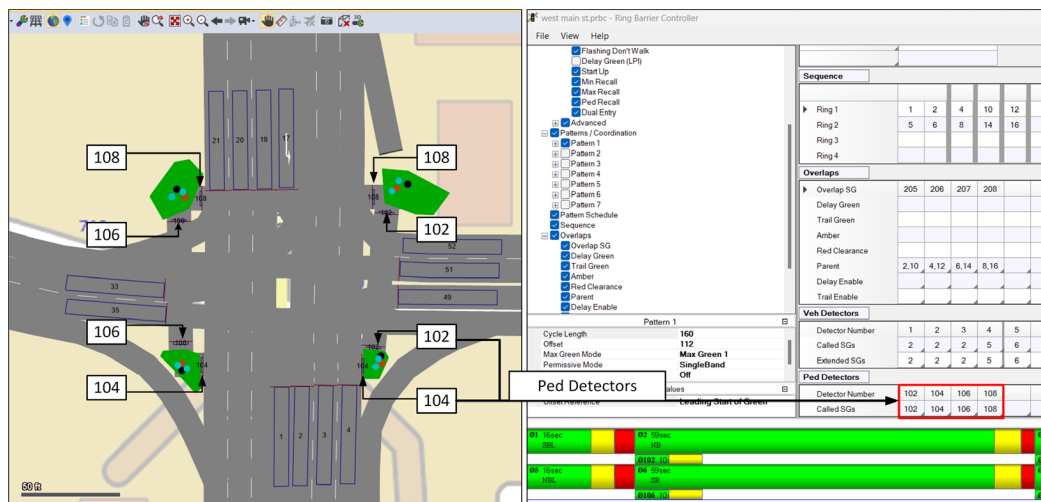


Figure 1-5: Pedestrian detectors in VISSIM

Configuring Overlaps in RBC

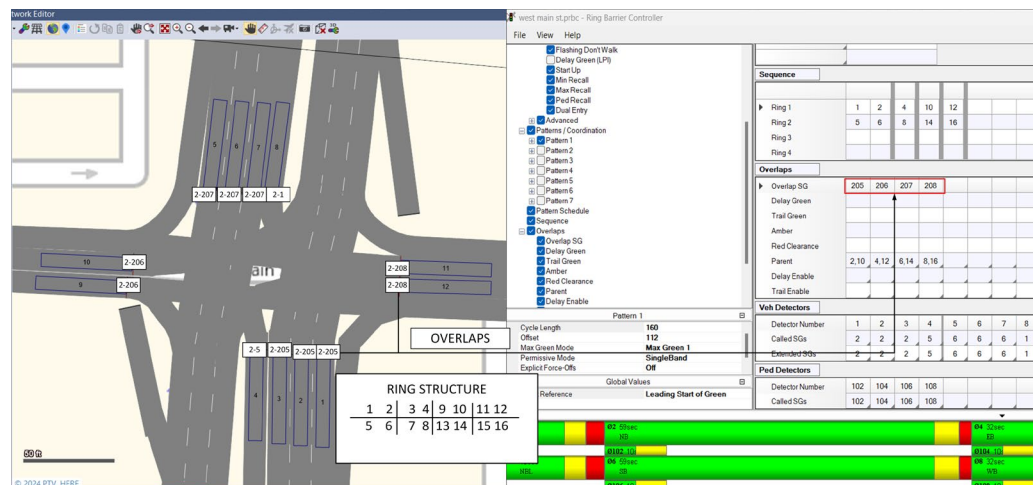
When configuring signal phases in an RBC controller, overlap definitions must be carefully specified to ensure correct phase concurrency and compatibility with ATSPM data processing. Overlaps allow multiple signal movements to operate simultaneously by combining outputs from one or more parent phases.

A double-cycle or 16-phase operation with overlaps can be configured as follows:

1. Multiple Parent Phases
 - Overlaps are defined by combining two or more parent signal phases.
2. Double-Cycle Operation

- Two complete signal cycles are executed within the duration of a single standard cycle.
3. Overlap Naming Convention
- Overlaps must be assigned using the 2XX numbering format (e.g., 205, 208).
 - This 2XX numbering convention applies to the RBC signal-group identifiers. Q-Free MAXTIME overlaps are configured separately using the controller's overlap designations.
4. Example Configuration
- As shown in Fig. 1-6, Overlap 205 represents the concurrent operation of Phase 2 and Phase 10, both serving through movements on the major roadway.
 - Phase 1 (northbound left-turn) and Phase 5 (southbound left-turn) are omitted during the second cycle.

Pedestrian detectors and overlaps must be configured using the 1XX and 2XX numbering conventions, respectively, as outlined in the previous sections and shown in Fig. 1-6. Failure to adhere to these conventions will prevent successful parsing of simulated outputs into ATSPM-compliant data formats for both UDOT ATSPM and UTAIM systems.



1. The first column specifies the intersection number.
2. The second column specifies the total number of detectors at the intersection.
3. The remaining columns list the detector port numbers in the exact order used by the RBC output.

[illegible]

1.2.4. Turn on high-resolution simulation output

7

the user. This section summarizes the procedure for activating and exporting simulated data required for ATSPM system integration.

Configuring WGS84 Coordinate System in VISSIM

Import the User-Defined Attributes (UDA) required for generating vehicle trajectories in WGS84 format using the provided VISSIM template, as shown in Fig. 1-8.

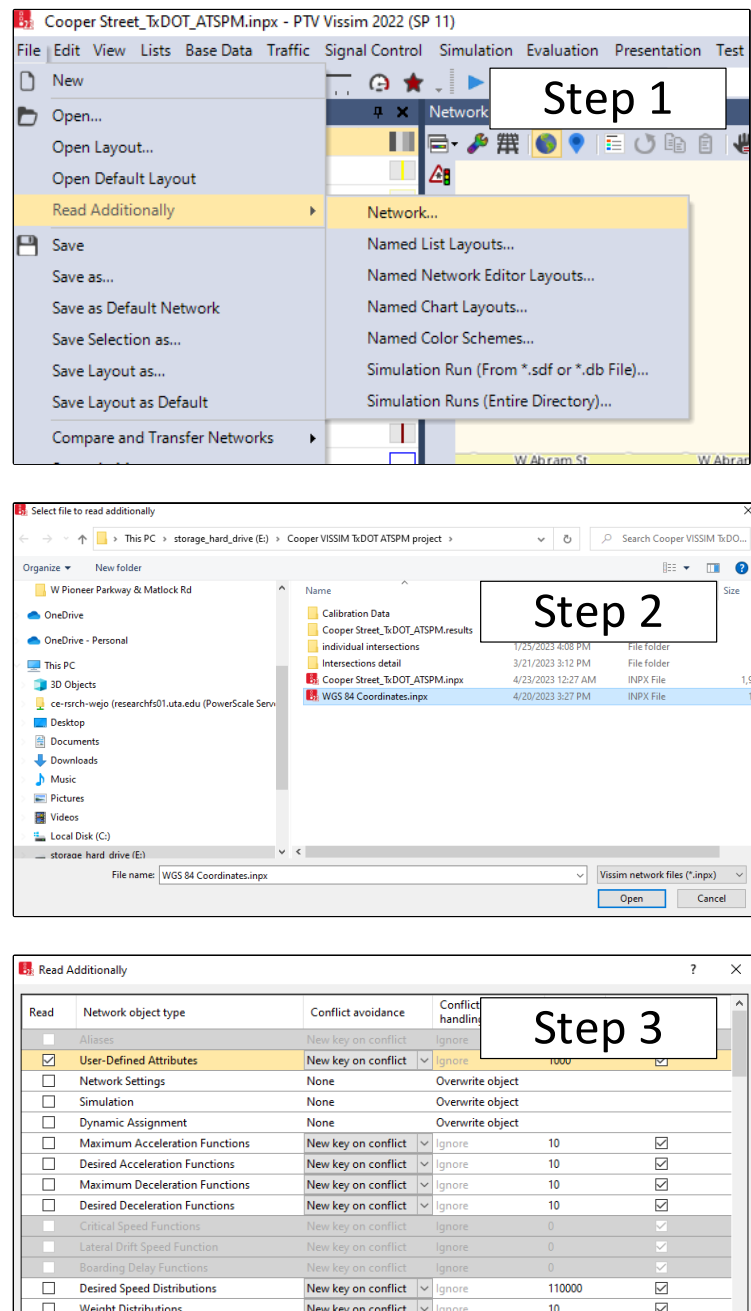


Figure 1-8: Loading the UDA for the WGS84 vehicle trajectory output.

Enable high-resolution traffic detector actuation output for each intersection. Fig. 1-9 illustrates the four sequential steps required to activate detector actuation data storage.

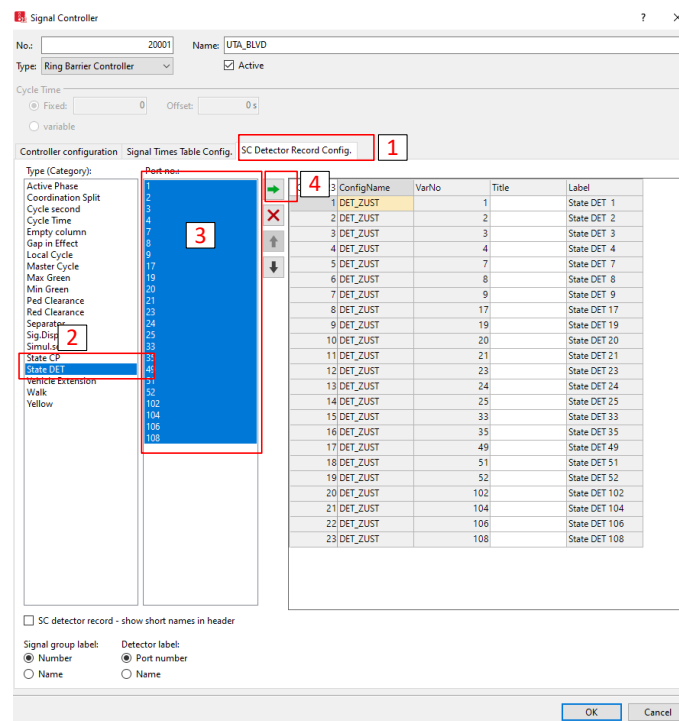


Figure 1-9: Enabling detector actuation outputs for each intersection.

Database Configuration in VISSIM

VISSIM supports storing simulation outputs in various database formats. Since the UDOT ATSPM system utilizes Microsoft SQL Server, simulation results are configured to be stored directly in an SQL Server database. Users may either create a new database on an existing SQL Server instance or install a free edition, such as SQL Server Express. It is recommended that database installation and configuration be performed with assistance from an IT professional.

After the SQL Server database is successfully set up, users should verify that VISSIM can establish a connection before running the simulation. Fig. 1-10 illustrates the procedure for connecting VISSIM to an SQL Server database. In Step 4, users must enter the appropriate server's name, username, password, and database name, and then select *Test connection*. If the connection is successful, a message box will appear confirming that the database connection has been established.

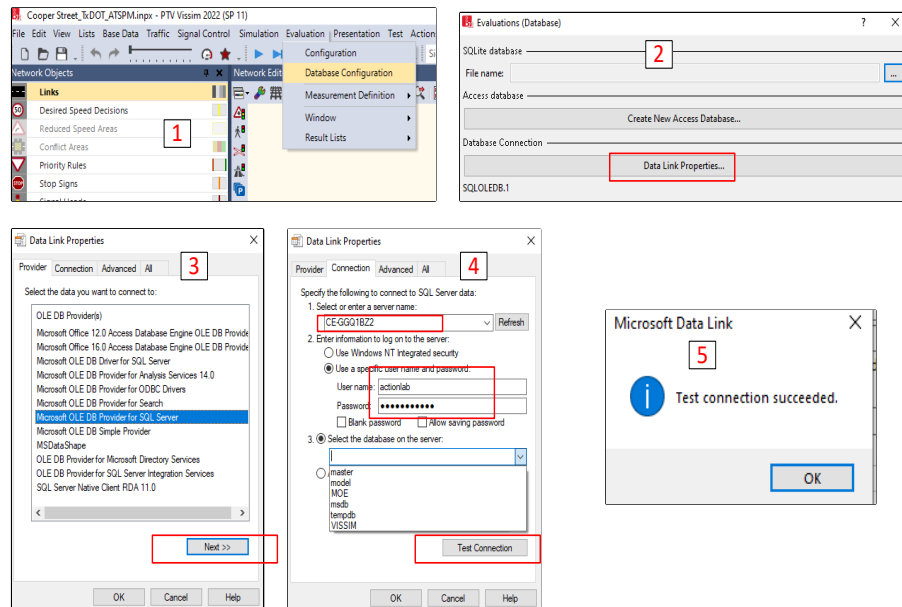


Figure 1-10: Connecting VISSIM to an SQL Server database.

To activate simulation outputs, navigate to Evaluation > Configuration in the VISSIM main menu, as shown in Fig. 1-11. In the configuration dialog, enable the three required output options to allow simulation results to be stored in the database. Next, select the Vehicle Record attributes and configure the vehicle output fields as illustrated in the third configuration step. These attribute selections ensure that the required vehicle trajectory data are correctly exported for subsequent processing and ATSPM integration.

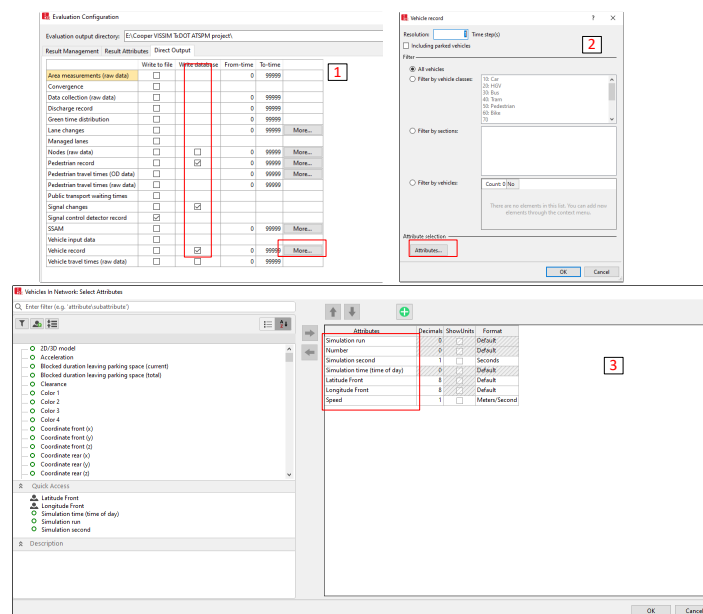


Figure 1-11: VISSIM simulation output evaluation configuration.

After completing the above four steps, VISSIM will automatically store ATSPM-related simulation outputs in three tables within the SQL Server database, containing data on pedestrian records, vehicle trajectories, and high-resolution traffic signal events. Outputs from multiple simulation runs may be stored within the same database instance. At this stage, the simulation data are ready to be transformed into ATSPM-compliant records for further processing and analysis.

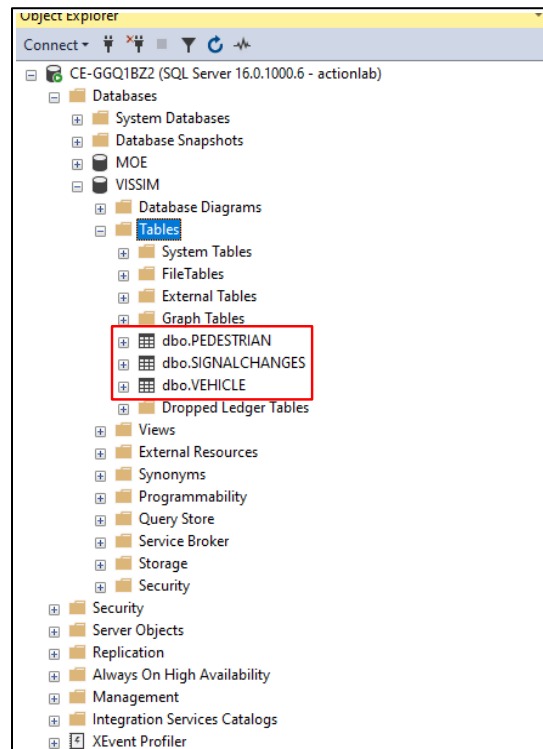


Figure 1-12: SQL Server database for VISSIM output.

1.3. Q-Free MAXTIME Signal Software-in-the-Loop (SILS) Simulation for ATSPM Data Generation

Q-Free MAXTIME Software-in-the-Loop Simulation (SILS) was developed and applied during TxDOT Research Project 0-7160 as one component of the ATSPM-in-the-loop simulation framework. As noted in Section 1.1, MAXTIME SILS was not used for the current Project 5-7160 corridor implementation, which uses the built-in VISSIM Ring-Barrier Controller (RBC), the Econolite EOS controller emulator, and hardware-in-the-loop simulation (HILS) with a physical Q-Free MAXTIME controller connected to VISSIM through the TxDOT Hardware-in-the-Loop Simulation Control Interface Device (HILS-CID). The MAXTIME SILS configuration procedure is retained in this comprehensive report to document the full development of the framework and provide guidance for future applications.

The MAXTIME SILS framework provides more detailed traffic signal event records than the standard RBC output available in VISSIM. The RBC can record signal-state transitions, such as the beginning or end of a green interval; however, it may not identify the controller logic responsible for a phase termination. For example, the RBC output may indicate that a green interval ended without specifying whether the phase terminated because of a gap-out, max-out, or another controller condition. In comparison, MAXTIME SILS records both signal-state changes and the underlying controller events associated with those changes.

Under Project 0-7160, the project team selected the Q-Free MAXTIME SILS implementation because it could be integrated directly with VISSIM and configured to generate high-resolution signal events compatible with ATSPM data structures. The additional controller-event information supports the generation of a broader range of ATSPM performance measures and provides a more complete representation of traffic signal controller operations within the simulation environment. The following sections retain the MAXTIME SILS setup and configuration procedures developed under Project 0-7160 for documentation, replication, and potential future deployment. The selection of Q-Free's SILS solution was driven by the following considerations:

- The Q-Free SILS package for VISSIM is available at no additional software cost to agencies, consultants, and academic users who have purchased a Q-Free traffic control product or who agree to the company's End-User License Agreement (EULA). In contrast, many alternative SILS packages require upfront licensing fees or recurring subscription costs.
- Q-Free provides regular updates to its MAXTIME control software, and corresponding updates were provided for the SILS module to maintain consistency with the latest controller functionality. This ensures that the SILS implementation remains current and adaptable to evolving traffic signal control practices.

While Q-Free's SILS package is recommended for the ATSPM-in-the-loop simulation framework presented in this study, the overall methodology is not limited to a single vendor. The framework also supports other SILS implementations, including Econolite's solutions, and can be extended to accommodate additional signal control platforms as needed.

1.3.1. Mandatory Requirements for Automated MAXTIME Launching for Multiple Intersections

Strict adherence to the requirements outlined in this section is essential for the successful automation and simultaneous operation of multiple MAXTIME SILS controllers. Failure to comply with these requirements may result in controller launch failures, incorrect signal event logging, or communication errors between VISSIM and MAXTIME.

Configuring a MAXTIME SILS controller typically requires numerous manual inputs, which can be time-consuming and error-prone, particularly when managing multiple intersections. To streamline this process, a dedicated VBScript-based automation tool has been developed to configure and launch MAXTIME controllers automatically. For the script to function correctly, the following mandatory conditions must be satisfied.

1. VISSIM Signal Controller Numbering

In PTV VISSIM, signal controller numbers must start from 1 and increase sequentially by 1 for each intersection. If the simulation model was created using non-sequential or non-zero-based numbering, the controller IDs must be adjusted accordingly. This renumbering can be performed directly within VISSIM. Once updated, all associated components (e.g., detectors) will be automatically reindexed to reflect the new controller numbering.

2. Detector Type Configuration

All detectors must be configured using the “Standard” detector type. Failure to use this detector type may result in incorrect detector state recognition by the MAXTIME SILS emulator.

3. MAXTIME Intersection Database Naming Convention

When creating user databases in MAXTIME, the intersection name must begin with the intersection ID that matches the corresponding VISSIM signal controller number. For example, if the intersection “*W Division St & Cooper St*” has a VISSIM controller ID of *N*, the MAXTIME database name must be formatted as:

“N: W Division St & Cooper St”

This naming convention must be applied consistently across all intersections, where *N* = 1, 2, 3,

4. MAXTIME Web Port Assignment Rules

MAXTIME outputs ATSPM signal event data through its web portal and XML server. The web port number for each MAXTIME controller is automatically assigned by the VBScript. The last one or two digits of the web port number must match the VISSIM intersection ID.

For example, if intersection IDs range from 1 to 5, the corresponding web ports will be assigned as 1001 through 1005. If the highest intersection ID is 18, the web port will be 1018.

The VBScript allows users to define the base port number within the range of 1000 to 9000, if needed. Users should avoid commonly occupied ports (e.g., 8000, 8080), as conflicts may prevent the XML server from initializing correctly.

5. VISSIM-MAXTIME TCP Port Configuration

The TCP port serves as the communication channel between VISSIM and each MAXTIME SILS controller. These ports must follow a strict numbering rule: the last one or two digits of the TCP port must match the intersection ID.

In this project, TCP ports are defined using the format $10000 + X$, where X is the VISSIM intersection ID. For example, if “*W Division St & Cooper St*” is Intersection 1, the TCP port must be 10001. For Intersection 18, the TCP port must be 10018.

Users must also verify that the TCP port number configured in VISSIM matches the TCP port defined in the corresponding MAXTIME intersection database.

The automation script includes built-in port conflict handling. If a port in the $1000 + X$ range is already occupied, the script will automatically attempt $2000 + X$, $3000 + X$, and continue incrementally up to $9000 + X$, without requiring any user input.

1.3.2. Software-in-the-Loop Simulation Configuration with VISSIM

This section provides a step-by-step procedure for configuring the MAXTIME SILS controller for each intersection within VISSIM

Step-by-Step Procedure for Configuring VISSIM

Before configuring the MAXTIME control software within VISSIM, users must ensure that the VISSIM environment is properly prepared to support Software-in-the-Loop Simulation (SILS). The following steps outline the required setup procedure.

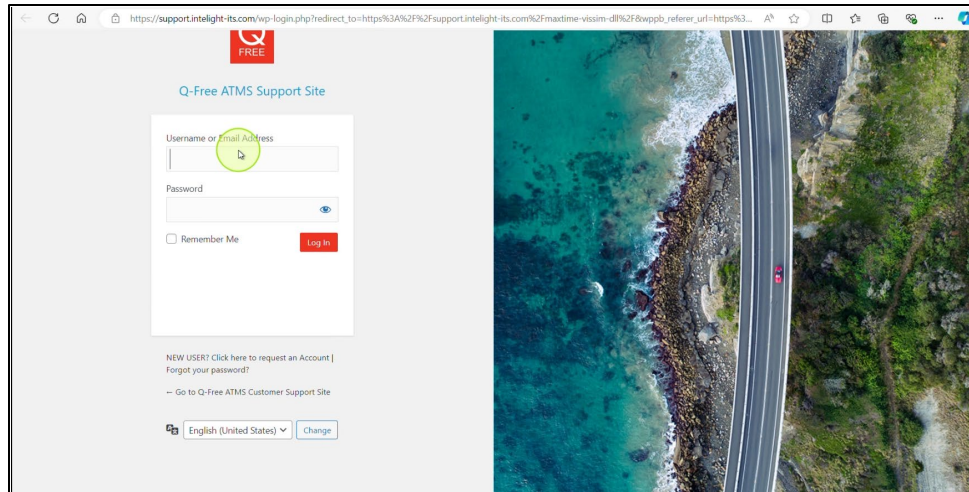


Figure 1-13: MAXTIME DLL download page for PTV VISSIM compatibility

1. Download Required MAXTIME Integration Files

Download the MAXTIME SILS files compatible with the installed PTV VISSIM version from the Q-Free support portal: <https://support.inteligh-its.com/maxtime-vissim-dll/>

After logging in and obtaining the necessary permissions, download the following files:

- *MaxTime.dll*
- *STDSC_GUI.dll*
- *Maxtime.wtt*

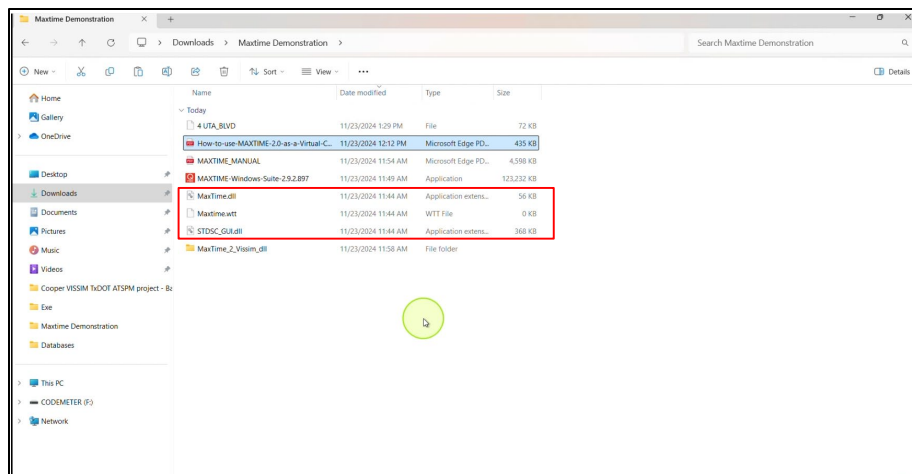


Figure 1-14: Downloaded files for MAXTIME integration

Copy all downloaded DLL and WTT files into the VISSIM executable directory. The default installation path is typically: *C:\Program Files\PTV Vision\YOUR_VISSIM_VERSION\Exe*

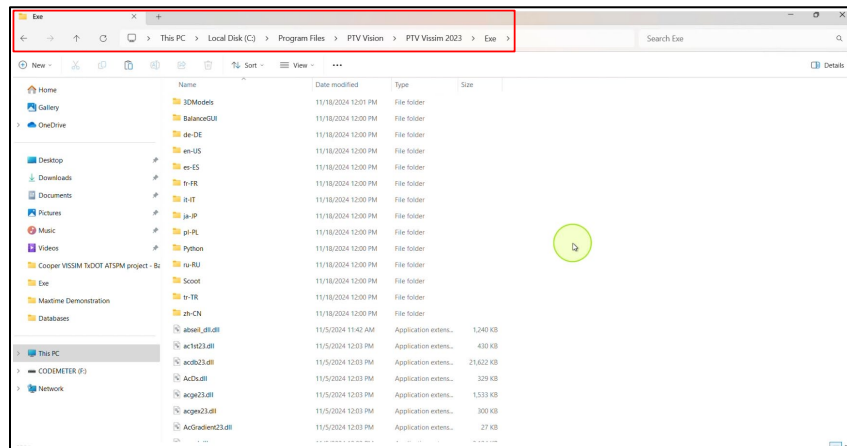


Figure 1-15: Destination path for the DLL/WTT files

2. Set Signal Controller Type to External

In the VISSIM network, select the target intersection and change the signal controller type to External.

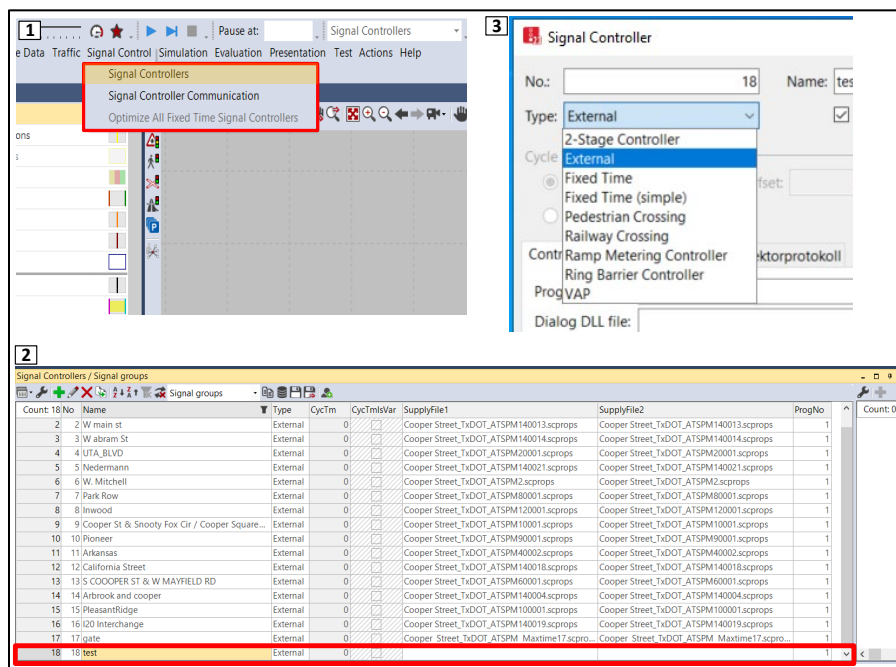


Figure 1-16: PTV VISSIM Signal Controller Interface Snapshot.

3. Configure External Signal Controller Files

Upon selecting the external signal controller, a configuration dialog will appear. Verify that the Program File and Dialog DLL File fields reference the files downloaded in Step 1.

For the WTT file, click the green “+” icon and add the downloaded *Maxtime.wtt* file.

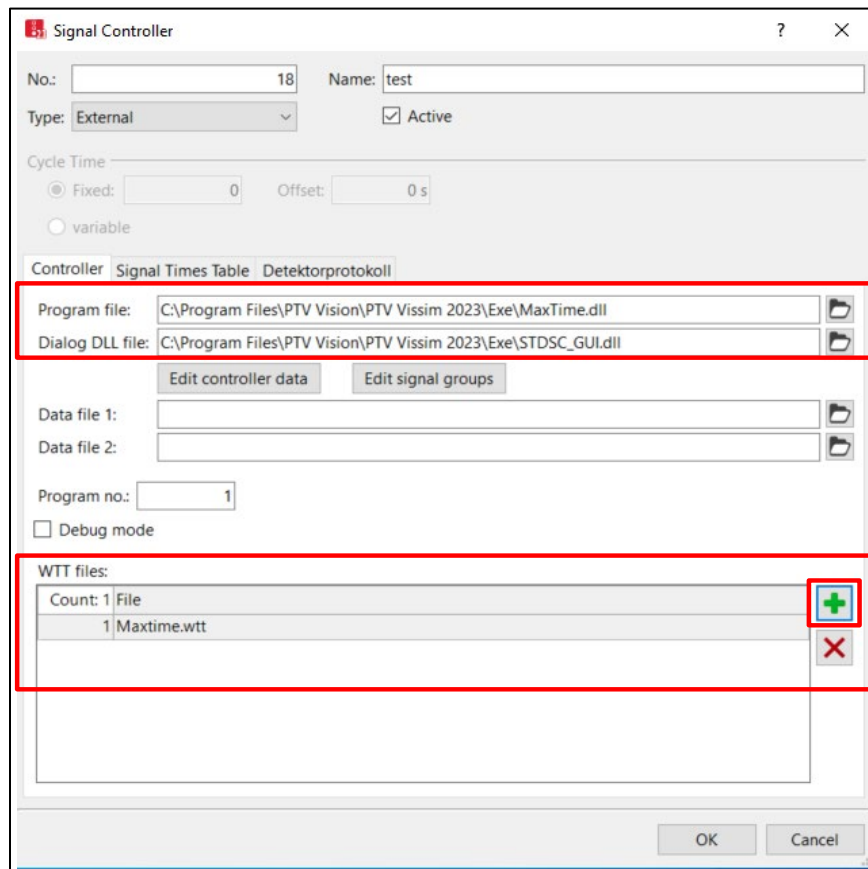


Figure 1-17: PTV VISSIM Signal Controller Dialog Box.

At this stage, the signal controller is ready for signal group configuration. It is important to note that, within the MAXTIME SILS framework, signal groups represent input/output (I/O) channels, not traffic signal phases. As a result, signal group numbers do not necessarily correspond directly to vehicle or pedestrian phase numbers.

- I/O channels 1–8 correspond to vehicle phases 1–8.
- I/O channels 9–12 is commonly used to represent pedestrian phase outputs (e.g., pedestrian phases 2, 4, 6, and 8).
- I/O channels 13–16 is commonly used to represent overlap phase outputs (e.g., overlaps 1–4).

Users must therefore model MAXTIME SILS controllers in VISSIM using the concept of I/O channels, rather than assuming a direct phase-to-signal-group mapping.

4. Configure Signal Group TCP Port and Control Channels

Navigate to Edit Signal Groups > Global Settings. By default, a standard TCP port value (e.g., 1234) is assigned. This value must be updated to the required port number corresponding to the intersection ID (e.g., 10001 for Intersection 1, 10018 for Intersection 18). The same TCP port number must later be configured within the MAXTIME controller database.

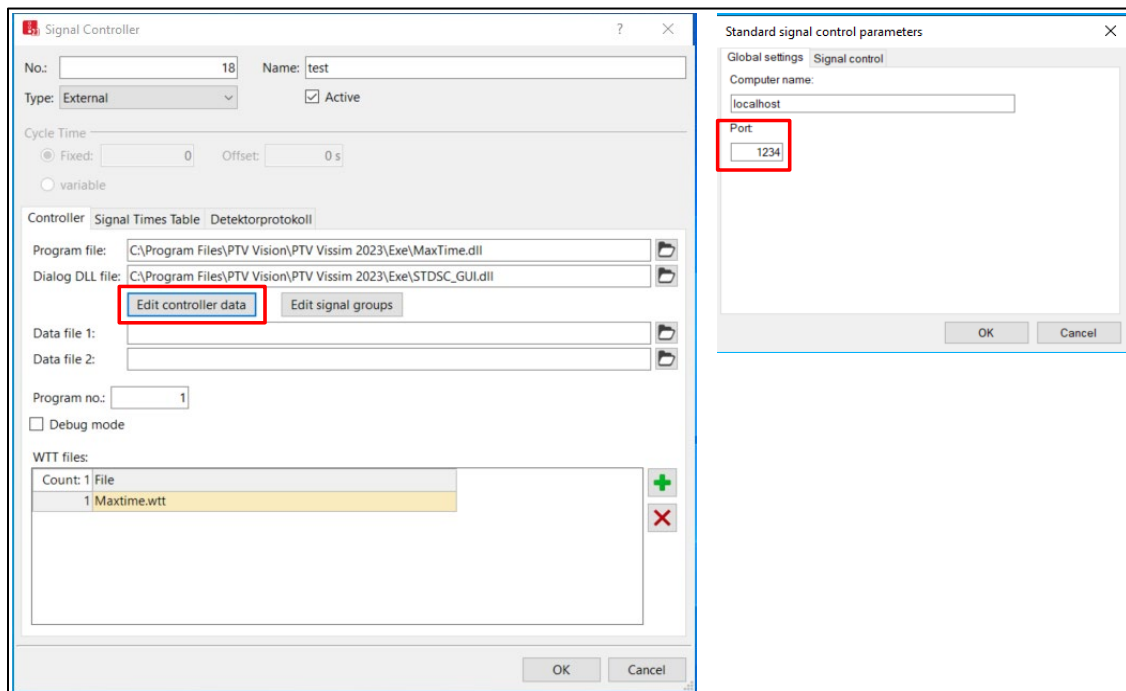


Figure 1-18: PTV VISSIM Standard Signal Controller Parameters Dialog Box.

Under Signal Control, add signal control group numbers corresponding to the MAXTIME I/O channels.

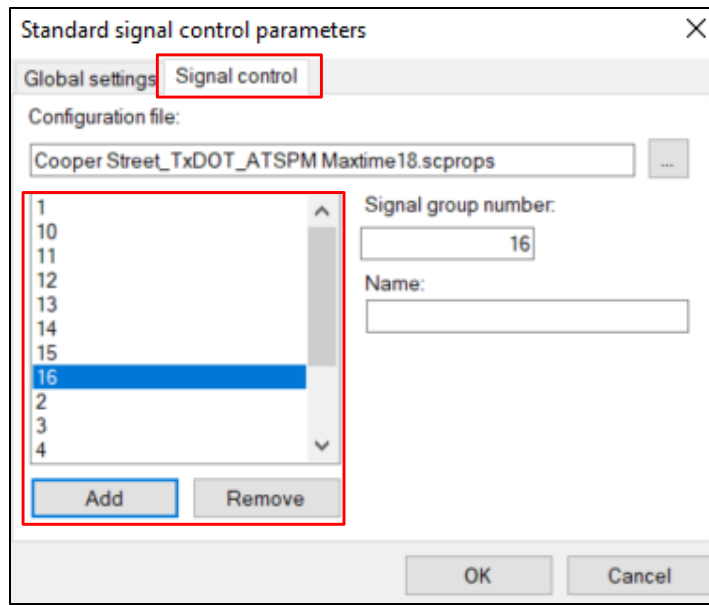


Figure 1-19: VISSIM signal control group setup.

Upon completing the above steps, VISSIM is fully prepared to couple with the virtual MAXTIME SILS controller. This configuration establishes the communication interface required for proper data exchange between the simulation engine and the signal control emulator.

The next section provides detailed instructions for configuring the Q-Free MAXTIME controller to complete the SILS integration.

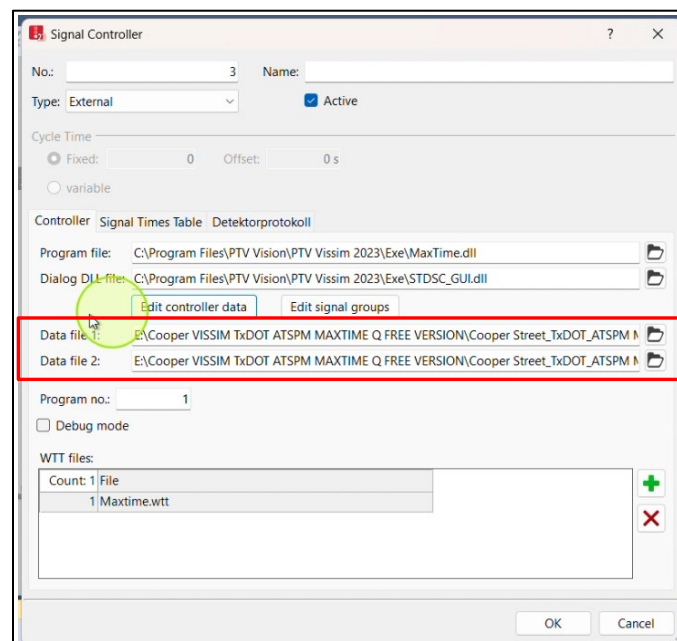


Figure 1-20: VISSIM–MAXTIME integration setup.

Note: Access to the Q-Free software download portal requires a privileged account and approval at Q-Free's discretion.

1.3.3. Step-by-Step Procedure for Configuring the MAXTIME Controller

This section describes the detailed procedure for configuring and launching the Q-Free MAXTIME SILS controller and coupling it with PTV VISSIM. Proper configuration is required to ensure correct signal operation, detector recognition, and ATSPM event generation.

Configure and Launch the MAXTIME Application

1. Launch the MAXTIME application using Run as Administrator to enable configuration edits.
2. Navigate to the MAXTIME tab for controller configuration (see Fig. 1-21).
3. Select the appropriate MAXTIME mode (e.g., Standard, Ramp Meter, CVITS, Test Developer).

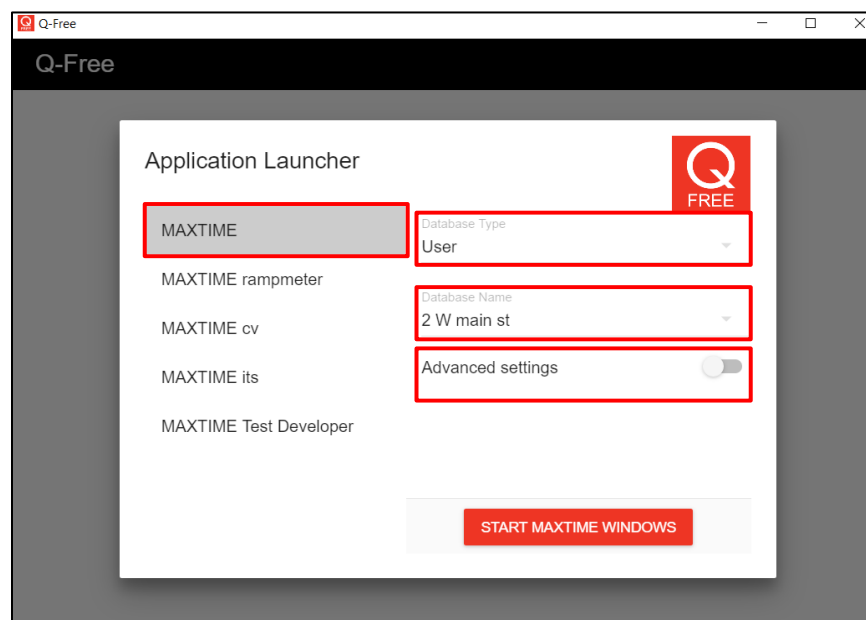


Figure 1-21: Q-Free MAXTIME Application Launcher Snapshot.

Select the Database Type:

- User Database – allows creation of project-specific signal timing databases.
- Factory Database – provides preconfigured timing plans supplied by Q-Free.

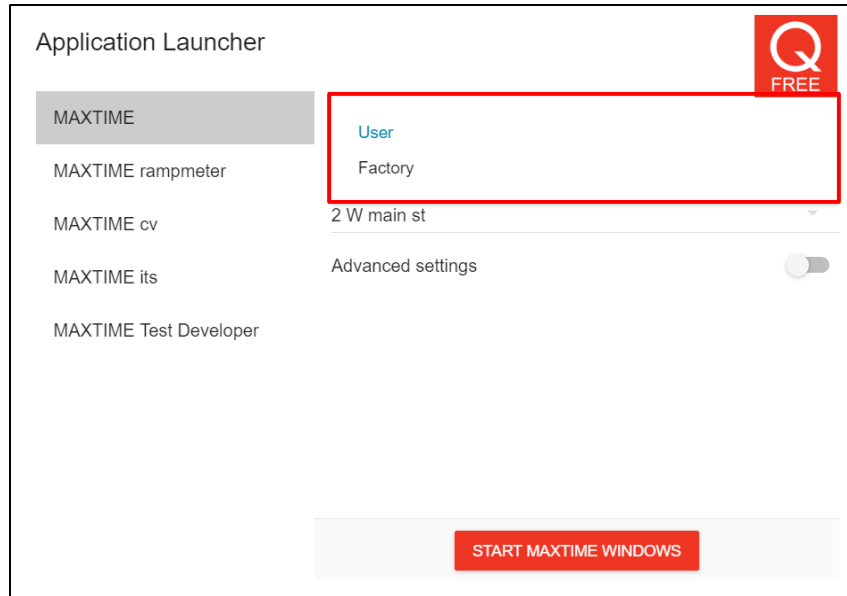


Figure 1-22: Highlighted Database Type in Q-Free MAXTIME Application Snapshot.

For first-time users, the User database will be empty. It is recommended to initially select a Factory database, which automatically creates the database structure. This database can then be modified or reprogrammed as needed. Database names must follow the intersection numbering rules defined in Section 1.3.1. Click Start MAXTIME Windows to launch the controller.

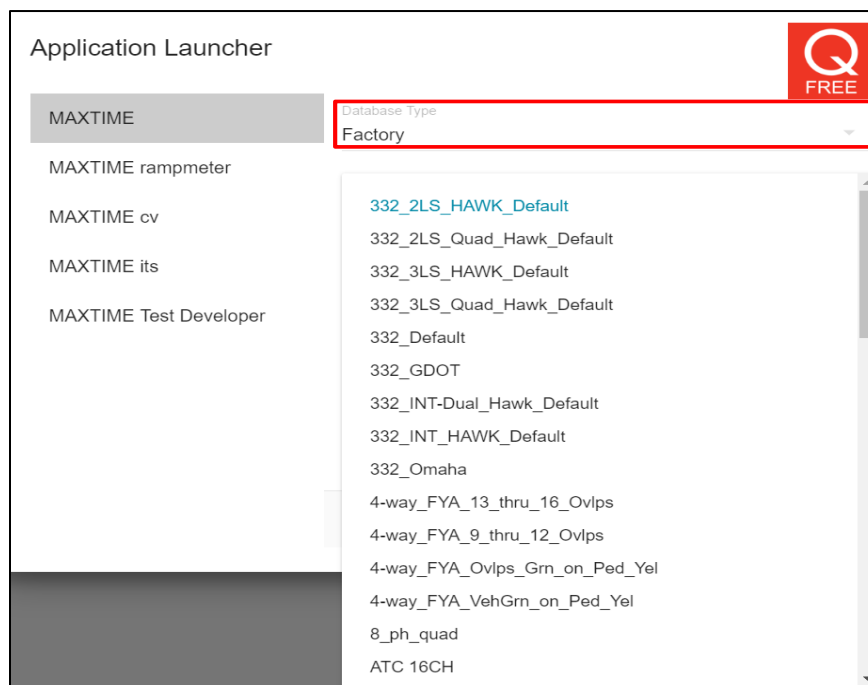


Figure 1-23: Snapshot Displaying List of Factory Setting Defaults in Q-Free MAXTIME.

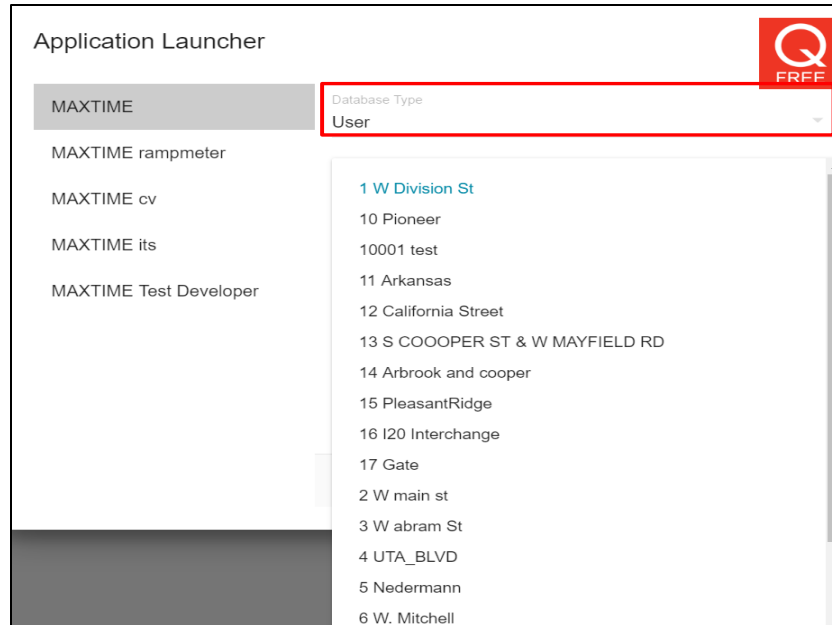


Figure 1-24: Snapshot Displaying List of User-Created Databases in Q-Free MAXTIME.

Controller Startup Options

After selecting the database, users may proceed in one of the following ways:

- Standalone Operation

Click *Start MAXTIME Windows* without configuring Advanced Settings. In this mode, MAXTIME runs independently and is not connected to VISSIM.

- SILS Coupling with VISSIM (Recommended)

Configure Advanced Settings to establish communication between MAXTIME and VISSIM.

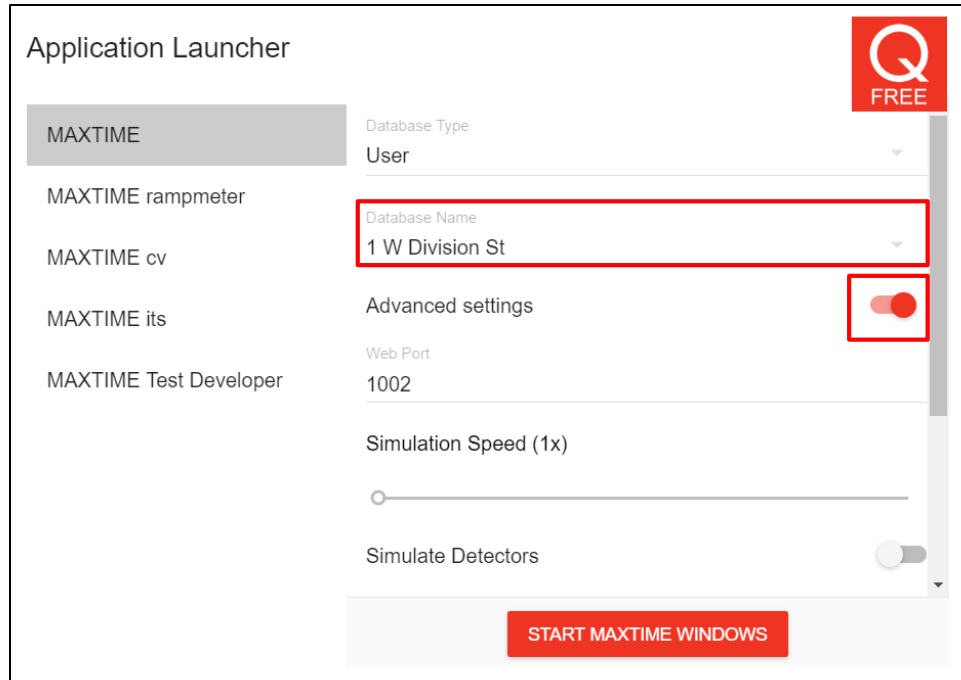


Figure 1-25: Q-Free MAXTIME Application Launcher with advanced settings on.

Initial MAXTIME Setup (Standalone or Coupled Mode)

When MAXTIME is launched for the first time:

1. The Welcome page will appear.
2. User profiles are stored locally at: C:\ProgramData\intelight\Profile

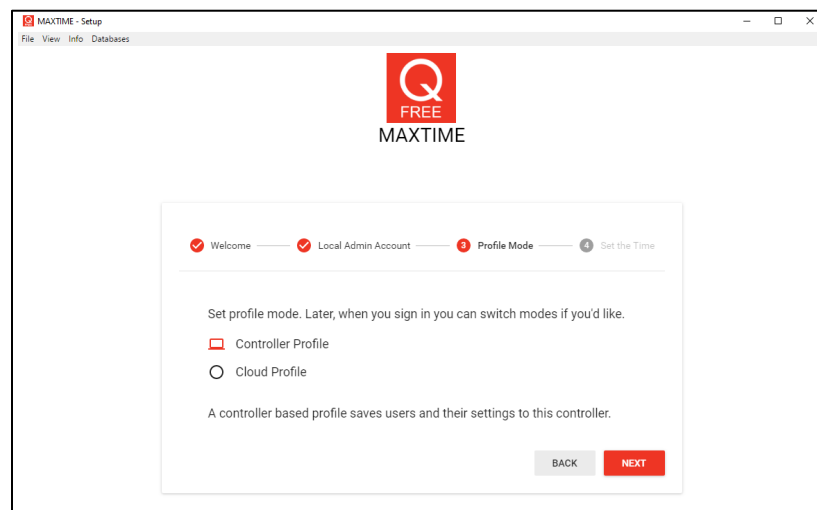


Figure 1-26: MAXTIME Welcome Page Display.

3. Configure user accounts. MAXTIME supports three user roles (see Fig. 1-27):
- System Administrator
 - User
 - Guest

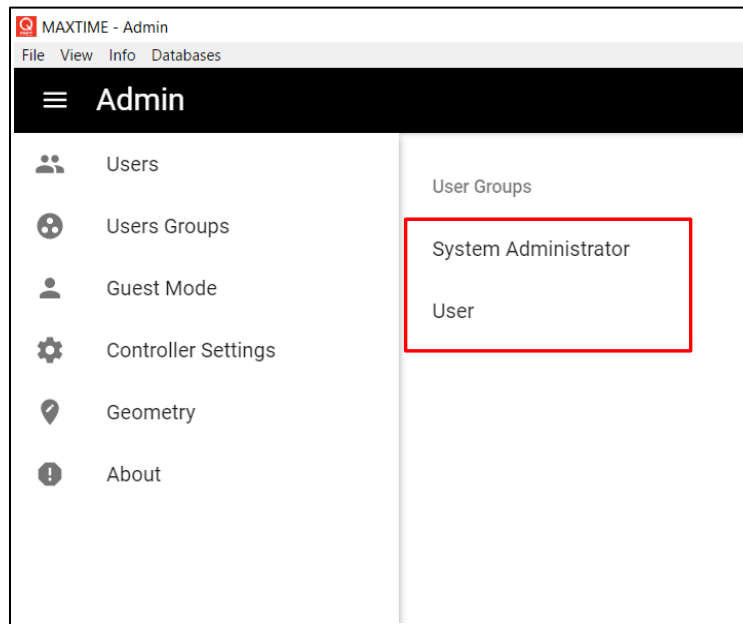


Figure 1-27: MAXTIME User Groups Display.

Select a Profile Mode:

- Controller Profile – saves settings locally.
- Cloud Profile – saves settings remotely.

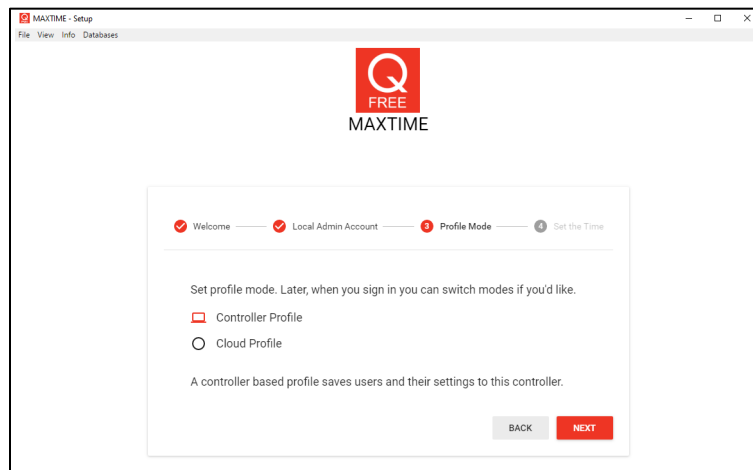


Figure 1-28: MAXTIME profile mode Display

Set the controller Date, Time, Time Zone, and Daylight-Saving parameters.

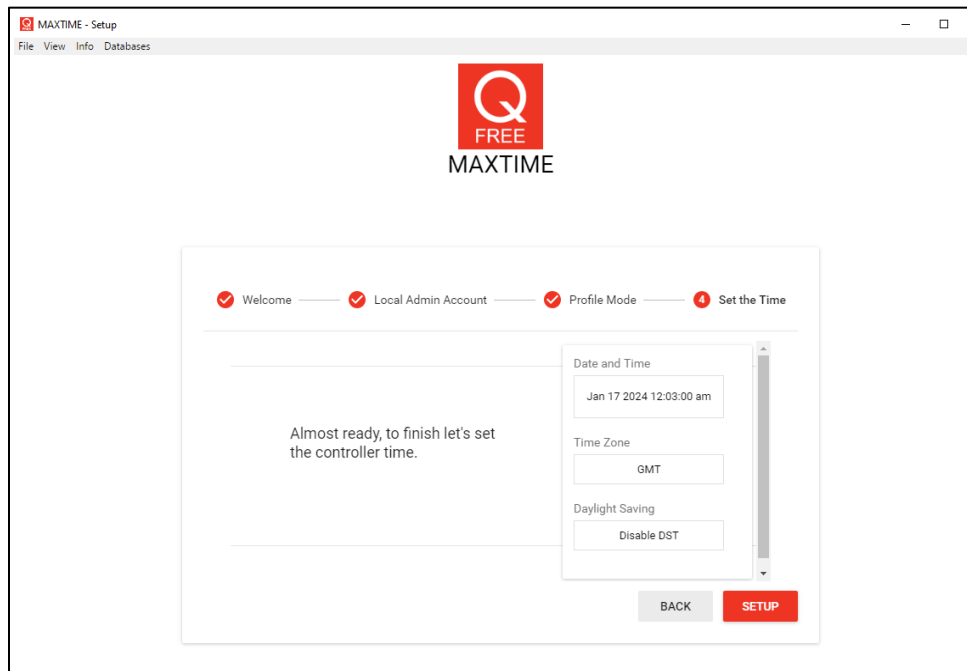


Figure 1-29: MAXTIME Date and Time Setup Display.

Once time configuration is complete, the MAXTIME controller is operational (see Fig. 1-30).

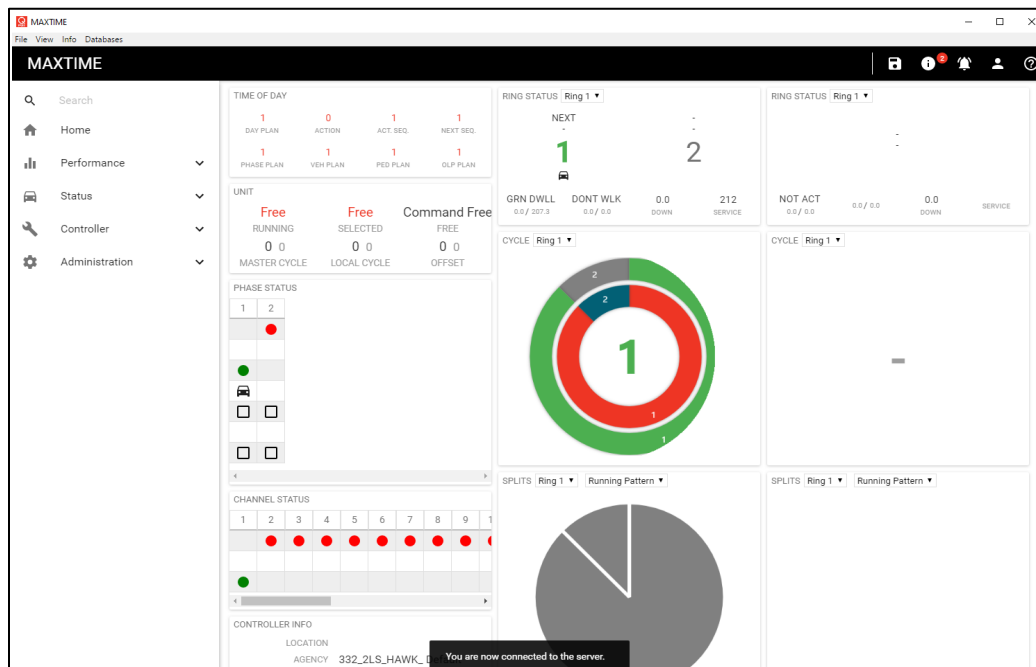


Figure 1-30: Main Display (front panel) of MAXTIME Controller Window.

Advanced Settings for VISSIM–MAXTIME Integration

Advanced Settings are required to couple MAXTIME with VISSIM. Each parameter must be configured correctly.

- Web Port

Each MAXTIME controller must use a unique web port. The port must not be in use by another application. Example access URL: <http://127.0.0.1:1002/maxtime>

Avoid commonly occupied ports (e.g., 8000, 8080). Even after closing MAXTIME, ports may remain temporarily reserved.

- Simulation Speed

This setting has no effect when MAXTIME is coupled with VISSIM, as simulation speed is controlled by VISSIM.

- Simulation Detectors

Detectors are defined in VISSIM; this option should remain unchanged or disabled.

- PTV VISSIM IP Address

Retain the default value of 127.0.0.1 when VISSIM and MAXTIME are operating on the same computer.

- PTV VISSIM TCP Port

Must match the TCP port configured in VISSIM for the corresponding intersection (see Section 1.3.1). Each controller requires a unique TCP port.

- PTV VISSIM ID

Enter the signal controller number assigned in VISSIM. This value must match the controller “NO.” field in VISSIM.

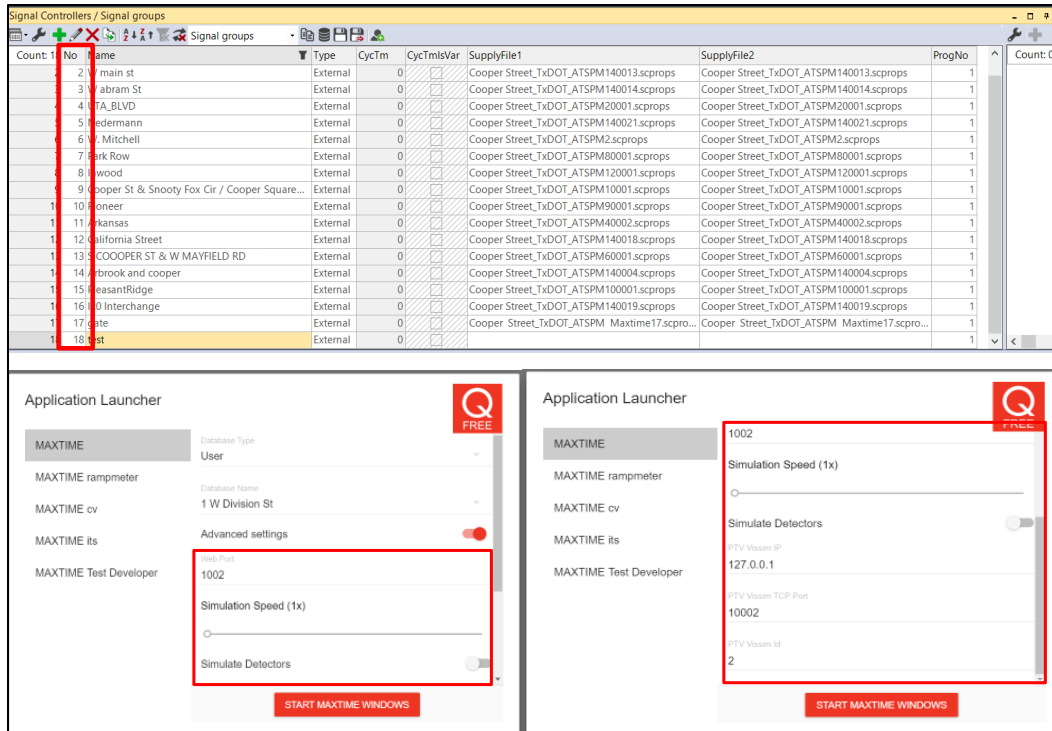


Figure 1-31: Advanced Settings Option in MAXTIME with Multiple Inputs.

After entering all parameters, click Start MAXTIME Windows. If the TCP port or controller ID does not match VISSIM, MAXTIME may launch, but the simulation will not operate correctly.

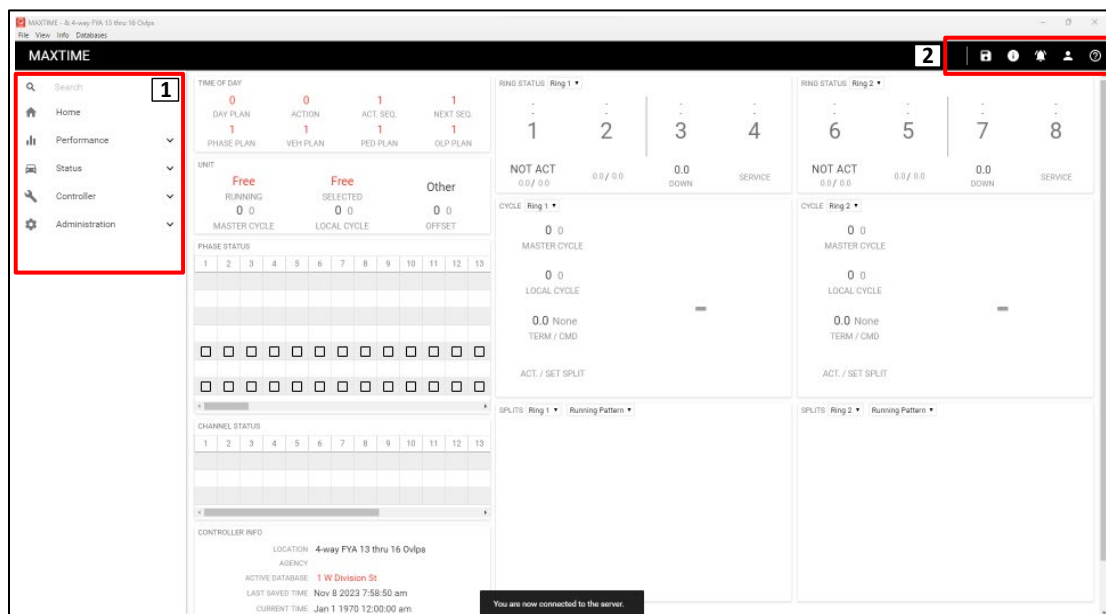


Figure 1-32: MAXTIME controller status interface.

Controller Identification and Cabinet Information

Navigate to Administration > Unit Information to update:

- Controller ID
- Main Street (major street)
- Side Street (minor street)

This information ensures proper identification of each controller.

MAXTIME - Cooper St & UTA BLVD

File View Info Databases

MAXTIME

Search

Home

Performance

Status

Controller

Administration

Unit Information

Communication Settings

Date & Time Settings

Event Logs

Database Management

Unit Information

Controller ID: 1

Main Street: Cooper St

Side Street: UTA BLVD

Agency:

Database Description

4-way FYA 13 thru 16 Ovips

Version Information

Module	Version	Make	Model
1	2.9.2	Q-Free	MaxTime
2	Windows	Q-Free	Microsoft

Figure 1-33: Window for Changing Controller Name and Unit Information in MAXTIME

Configure I/O Modules for Simulation

To enable simulation operation:

1. Navigate to Controller > Advanced I/O > Cabinet Configuration > I/O Modules.
2. Set the I/O Module Type to Simulation.

If the I/O module type is not set to Simulation, MAXTIME will not connect to VISSIM, and no outputs will be produced.

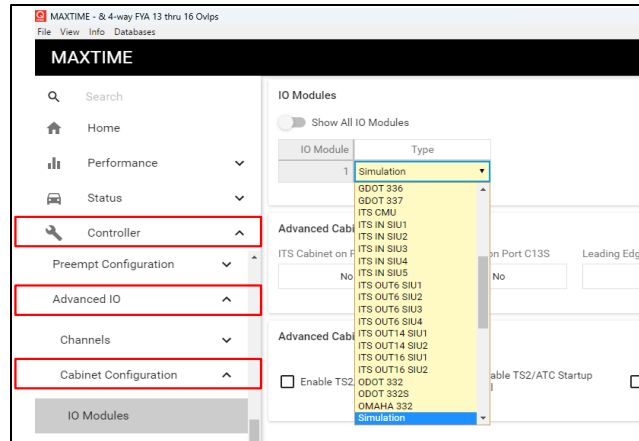


Figure 1-34: Advanced Input-Output (Akhtar & Moridpour) Module Type Configuration in MAXTIME

Configure I/O Channels and Phase Mapping

In MAXTIME, VISSIM signal phases are mapped to I/O channels, not directly to controller phase numbers. As described in Section 1.3.2, the I/O-channel assignments must be configured according to the signal timing plan and the required vehicle, pedestrian, and overlap outputs. Users must configure I/O channels according to the signal timing plan rather than assuming direct phase equivalence.

Examples:

- VISSIM signal group representing Vehicle Phase 1 → I/O Channel 1
- VISSIM signal group representing a pedestrian phase → the corresponding pedestrian I/O channel

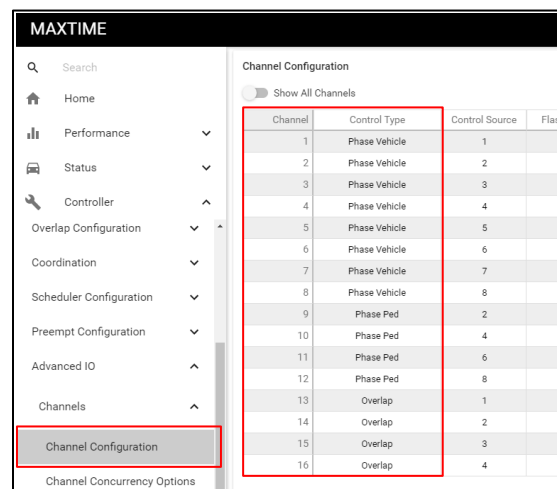


Figure 1-35: Advanced Input-Output Channel Configuration Page in MAXTIME

Configure Detector Inputs

Detectors in VISSIM generate vehicle, pedestrian, and special calls. To ensure correct detection:

- In VISSIM:
 - Assign correct detector port numbers.
 - Use only Standard detector types.
- In MAXTIME:
 - Navigate to Controller > Advanced I/O > Cabinet Configuration > Input Points.
 - Assign control types based on port number: Example:
 - Port 35 → Veh Detector Call
 - Port 67 → Ped Detector Call

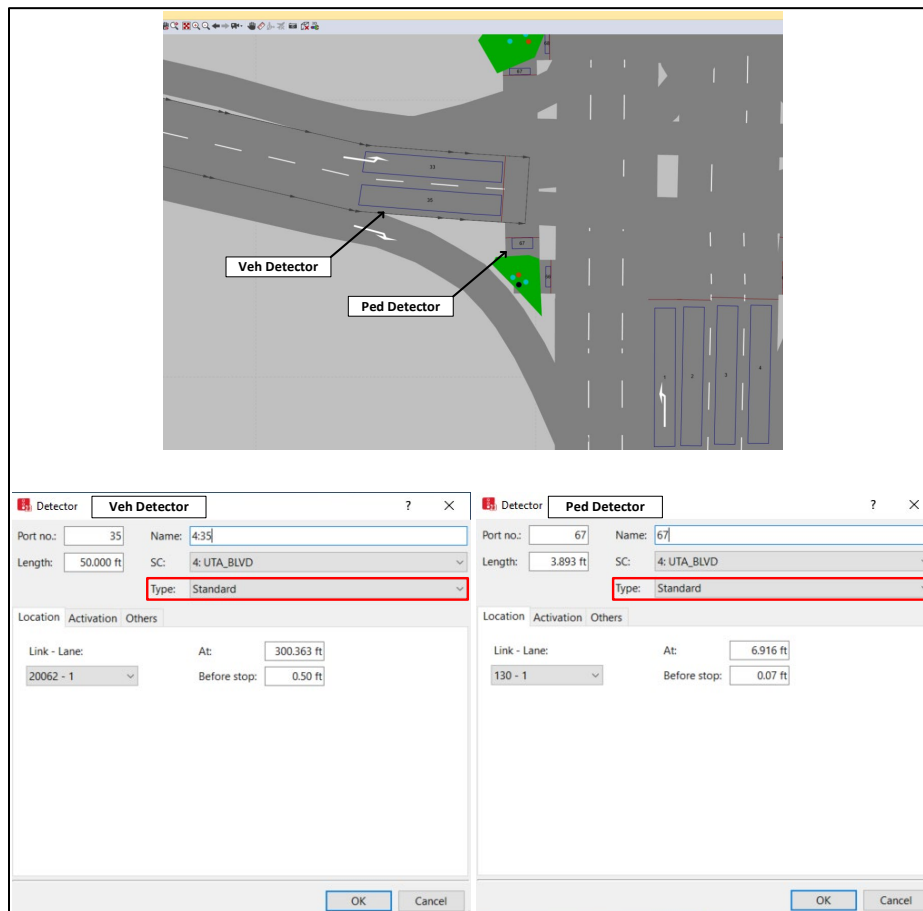


Figure 1-36: Vehicle and Pedestrian Detector Layout.

Note: Incorrect port-to-input mapping will prevent MAXTIME from recognizing detector calls

IO Module 1				Show All Input Points			
Input Point	Description	Input Control Type	Index				
32	I-32	Veh Detector Call	32				
33	I-33	Veh Detector Call	33				
34	I-34	Veh Detector Call	34				
35	I-35	Veh Detector Call	35				
36	I-36	Veh Detector Call	36				
37	I-37	Veh Detector Call	37				
38	I-38	Veh Detector Call	38				
39	I-39	Veh Detector Call	39				
40	I-40	Veh Detector Call	40				
41	I-41	Veh Detector Call	41				
42	I-42	Veh Detector Call	42				
43	I-43	Veh Detector Call	43				
57	I-57	Veh Detector Call	57				
58	I-58	Veh Detector Call	58				
59	I-59	Veh Detector Call	59				
60	I-60	Veh Detector Call	60				
61	I-61	Veh Detector Call	61				
62	I-62	Veh Detector Call	62				
63	I-63	Veh Detector Call	63				
64	I-64	Veh Detector Call	64				
65	I-65	Ped Detector Call	65				
66	I-66	Ped Detector Call	66				
67	I-67	Ped Detector Call	67				
68	I-68	Ped Detector Call	68				
69	I-69	Veh Detector Call	69				

Figure 1-37: Input Points Page for Different Input Control Types in MAXTIME

Configure Overlaps in MAXTIME

In MAXTIME, overlaps are associated with I/O channels:

- Overlap 1 → Channel 13
- Overlap 2 → Channel 14
- Overlap 3 → Channel 15
- Overlap 4 → Channel 16

MAXTIME																																																																							
Search Home Performance Status Controller Overlap Configuration Coordination Scheduler Configuration Preempt Configuration Advanced IO Channels Channel Configuration Channel Concurrency Options Cabinet Configuration Phase Intervals Channel Configuration Show All Channels <table> <tr> <th>Channel</th><th>Control Type</th><th>Control Source</th><th>Flash</th></tr> <tr><td>1</td><td>Phase Vehicle</td><td>1</td><td></td></tr> <tr><td>2</td><td>Phase Vehicle</td><td>2</td><td></td></tr> <tr><td>3</td><td>Phase Vehicle</td><td>3</td><td></td></tr> <tr><td>4</td><td>Phase Vehicle</td><td>4</td><td></td></tr> <tr><td>5</td><td>Phase Vehicle</td><td>5</td><td></td></tr> <tr><td>6</td><td>Phase Vehicle</td><td>6</td><td></td></tr> <tr><td>7</td><td>Phase Vehicle</td><td>7</td><td></td></tr> <tr><td>8</td><td>Phase Vehicle</td><td>8</td><td></td></tr> <tr><td>9</td><td>Phase Ped</td><td>2</td><td></td></tr> <tr><td>10</td><td>Phase Ped</td><td>4</td><td></td></tr> <tr><td>11</td><td>Phase Ped</td><td>6</td><td></td></tr> <tr><td>12</td><td>Phase Ped</td><td>8</td><td></td></tr> <tr><td>13</td><td>Overlap</td><td>1</td><td></td></tr> <tr><td>14</td><td>Overlap</td><td>2</td><td></td></tr> <tr><td>15</td><td>Overlap</td><td>3</td><td></td></tr> <tr><td>16</td><td>Overlap</td><td>4</td><td></td></tr> </table>				Channel	Control Type	Control Source	Flash	1	Phase Vehicle	1		2	Phase Vehicle	2		3	Phase Vehicle	3		4	Phase Vehicle	4		5	Phase Vehicle	5		6	Phase Vehicle	6		7	Phase Vehicle	7		8	Phase Vehicle	8		9	Phase Ped	2		10	Phase Ped	4		11	Phase Ped	6		12	Phase Ped	8		13	Overlap	1		14	Overlap	2		15	Overlap	3		16	Overlap	4	
Channel	Control Type	Control Source	Flash																																																																				
1	Phase Vehicle	1																																																																					
2	Phase Vehicle	2																																																																					
3	Phase Vehicle	3																																																																					
4	Phase Vehicle	4																																																																					
5	Phase Vehicle	5																																																																					
6	Phase Vehicle	6																																																																					
7	Phase Vehicle	7																																																																					
8	Phase Vehicle	8																																																																					
9	Phase Ped	2																																																																					
10	Phase Ped	4																																																																					
11	Phase Ped	6																																																																					
12	Phase Ped	8																																																																					
13	Overlap	1																																																																					
14	Overlap	2																																																																					
15	Overlap	3																																																																					
16	Overlap	4																																																																					

Figure 1-38: IO Channel Configuration for Overlaps in MAXTIME.

To configure overlaps:

1. Navigate to Controller > Overlap Configuration > Overlaps.
2. Switch from *Item* view to *Table* view for easier editing.
3. Define included phases and modifier phases. Example:
 - Included Phase: 2
 - Modifier Phase: 1
 - Represents a permissive left-turn (flashing yellow).

Overlap	1	2	3	4
Enabled	Enabled	Enabled	Enabled	Enabled
Description				
Type	FYA - 4 Section	FYA - 4 Section	FYA - 4 Section	FYA - 4 Section
Included Phases	2	4	6	8
Modifier Phases	1	3	5	7
Modifier Overlaps				
Negative Phases				
Trail Green	0	0	0	0
Trail Yellow	0.0	0.0	0.0	0.0
Trail Red	0.0	0.0	0.0	0.0
Walk	0	0	0	0
Ped Clear	0	0	0	0
Delay	3.0	3.0	3.0	3.0
Flash	On	On	On	On
Negative Overlaps				
Negative Peds	2	4	6	8
Neg Ped Overlaps				
Min Green	0	0	0	0
Red Revert	0.0	0.0	0.0	0.0
Startup Call	☐	☐	☐	☐
Recall	☐	☐	☐	☐

Figure 1-39: Overlaps Configuration in MAXTIME.

Place Overlaps in VISSIM

After overlap configuration:

1. Identify the corresponding I/O channel
2. In VISSIM, assign the overlap to the “Or signal group”.
 - Example: Overlap 1 → I/O Channel 13 placed at Phase 1 signal head for the permissive left-turn indication.

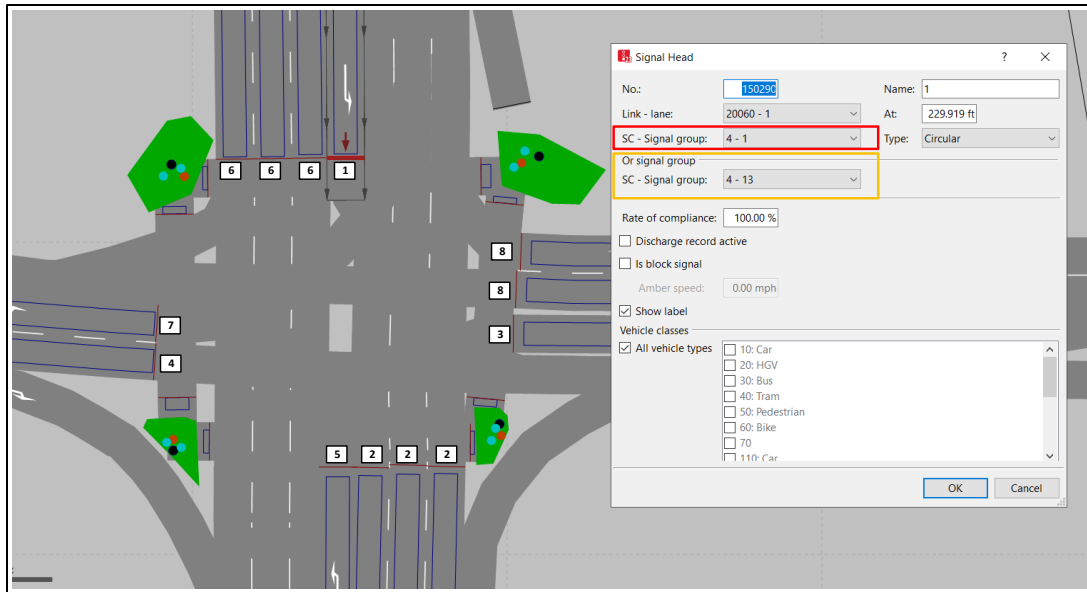


Figure 1-40: Intersection Layout with Different Phases for Overlaps in VISSIM.

Verification and Status Monitoring

Verify detector calls, phase status, and overlap operation by navigating to Main Screen > Status in MAXTIME. Performance reports are automatically generated; however, report duration and size may be limited.

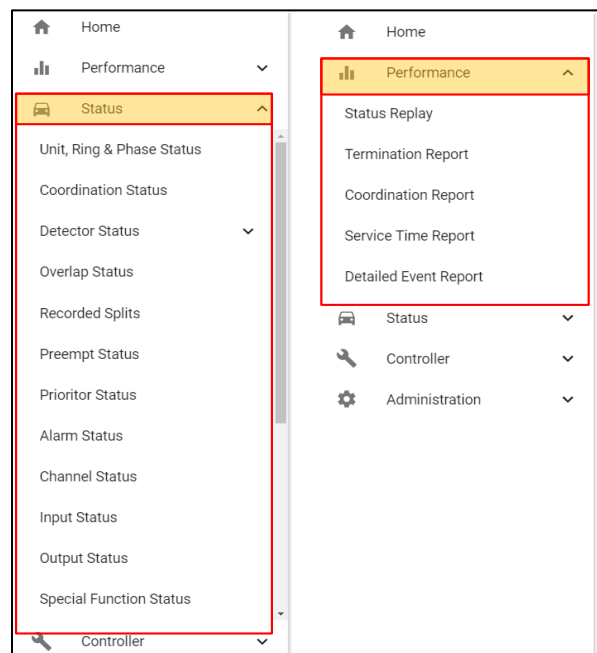


Figure 1-41: Status and Performance Tab Display in MAXTIME.

1.4. Software Tool for Parsing VISSIM Outputs into ATSPM Databases

The previous sections described how VISSIM, RBC, and MAXTIME SILS are configured to generate ATSPM-related simulation outputs. This section describes the software tools used to convert those outputs into ATSPM-compliant database formats. The tools support four main functions: parsing RBC and MAXTIME SILS signal events into the UDOT ATSPM database, processing simulated connected vehicle (CV) trajectories for use in UTAIM, transferring ATSPM records from UDOT ATSPM to UTAIM, and automating multiple MAXTIME SILS controller launches when needed.

To bridge the gap between simulated outputs and operational ATSPM databases, a dedicated software tool was developed to parse simulated traffic signal event data and connected vehicle (CV) trajectory outputs into ATSPM-compliant formats. This tool consists of multiple functional modules, each designed to support a specific stage of the data processing workflow. The following software components were developed as part of this effort:

1. ATSPM Data Parser

A parsing tool that converts simulated ATSPM datasets into the UDOT ATSPM database format, supporting both RBC-generated and MAXTIME SILS-generated traffic signal event data.

2. ATSPM Database Transfer Tool

A data transfer utility that migrates ATSPM data from the UDOT ATSPM database to the UTA-In-Motion (UTAIM) ATSPM database.

3. MAXTIME SILS Automation Script

A VBScript-based automation tool that launches and configures multiple MAXTIME SILS controllers simultaneously for SILS-enabled intersections in VISSIM.

The following subsections describe the functionality and implementation of each software component in detail.

1.4.1. Running Multiple Intersections in Automation Using VBScript

The configuration procedures described in Section 1.3 focus on setting up a single signal controller for one intersection. When a project involves multiple intersections, each intersection requires a separate controller database and a distinct instance of the MAXTIME Windows application. As specified in Section 1.3.1, each intersection must be assigned a unique web port, a unique VISSIM TCP port, and a corresponding signal controller number in VISSIM. Fig. 1-42

illustrates an example configuration with multiple MAXTIME controllers operating alongside multiple signal controllers in PTV VISSIM.

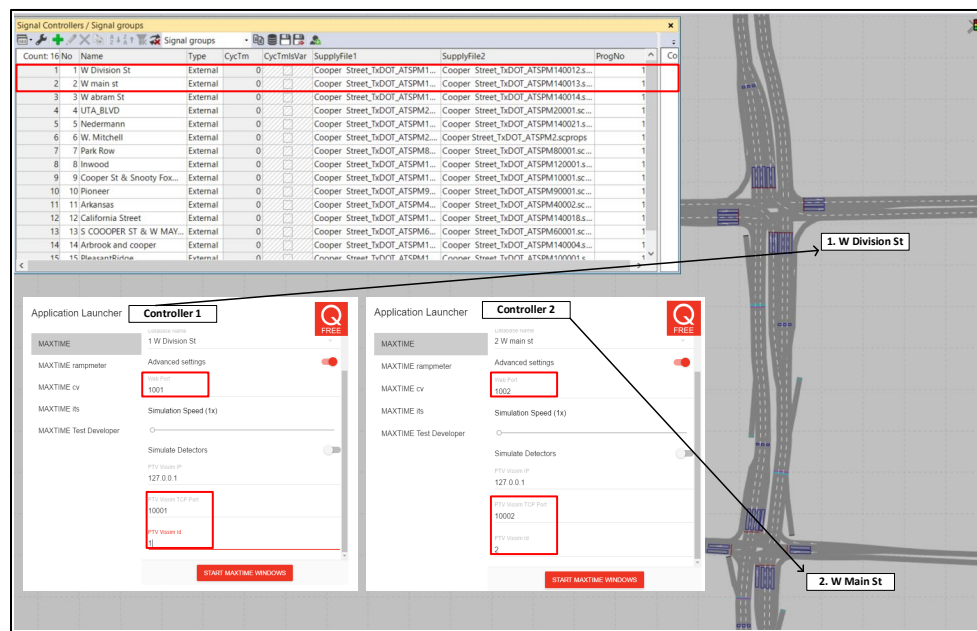


Figure 1-42: Multiple MAXTIME controllers coupled with multiple intersections in PTV VISSIM.

Running simulations with a large number of MAXTIME controllers, particularly for corridors with 16 or more intersections, can be time-consuming and prone to manual configuration errors. To streamline this process, a VBScript-based automation tool was developed to automatically populate required configuration parameters and launch multiple MAXTIME controller instances simultaneously. This automation significantly reduces setup time and minimizes the potential for user input errors.

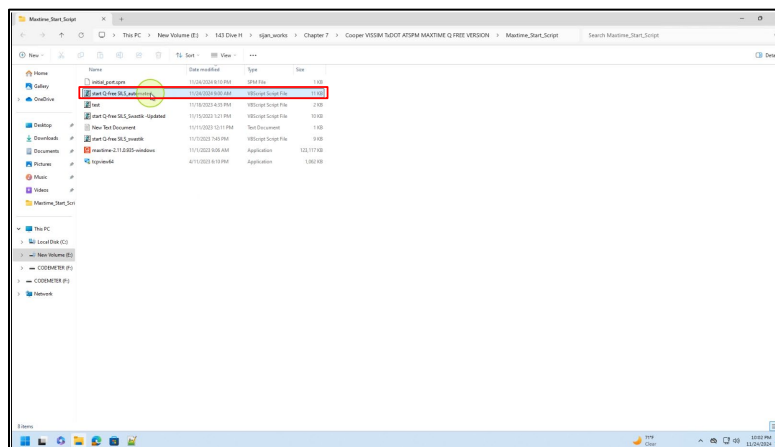


Figure 1-43: VBScript for automating MAXTIME configuration for multiple controllers.

Before executing the VBScript, users must verify and modify the following parameters within the script:

1. WEB_PORT_START

Specifies the starting web port number for the first intersection (e.g., 3001).

2. Number_of_controllers

Defines the total number of intersections included in the simulation (e.g., 10 for ten intersections).

3. MAXTIME Executable Directory

Confirms the file path to the MAXTIME executable (e.g.,

C:\AppData\Local\Programs\MAXTIME\MAXTIME.exe).

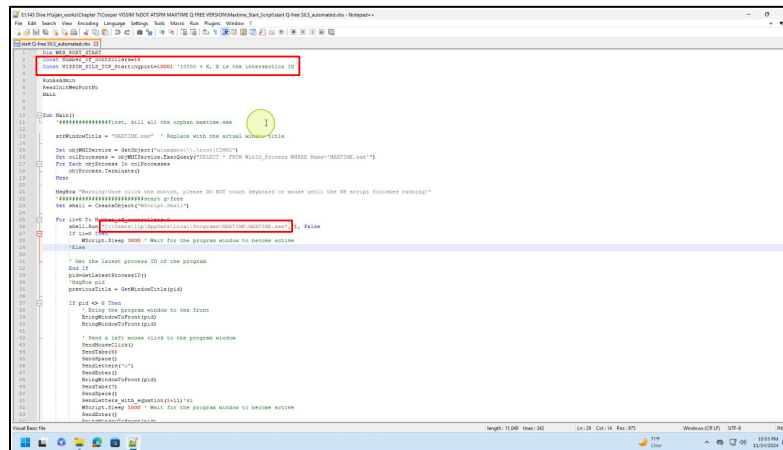


Figure 1-44: VBScript automation parameters.

Once the VBScript is launched, users should avoid interacting with the computer, including mouse or keyboard input, as cursor movement may interrupt or disrupt the automation process.

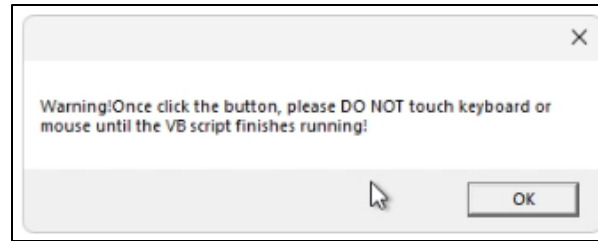


Figure 1-45: Warning indicating that user input should be avoided during VBScript execution.

1.4.2. Parsing RBC Signal Control Events from the VISSIM Database to the UDOT ATSPM Database

Before converting RBC-simulated traffic signal events into UDOT ATSPM-compliant data, users must ensure that the VISSIM simulation outputs have been successfully stored in a SQL Server database (DB) and that UDOT ATSPM is properly installed. As illustrated in Fig. 1-46, the VISSIM output DB operates in parallel with the UDOT ATSPM MOE DB. Fig. 1-47 presents an overview of the data-processing workflow when using RBC within VISSIM.

The initial step of the parsing process requires users to specify database connection parameters, including database locations, usernames, passwords, and the simulation start time. All required tables in both the VISSIM output database and the UDOT ATSPM database are pre-defined by their respective systems, and table names are fixed. As a result, users are not required to manually enter or modify table names within the parsing software.

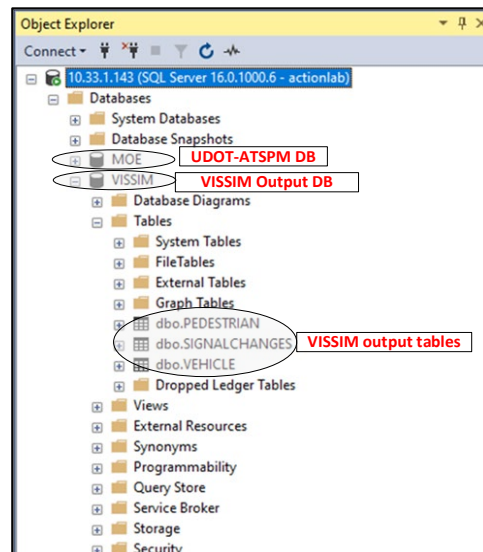


Figure 1-46: UDOT-ATSPM and VISSIM output DBs displayed in SQL Server Management Studio.

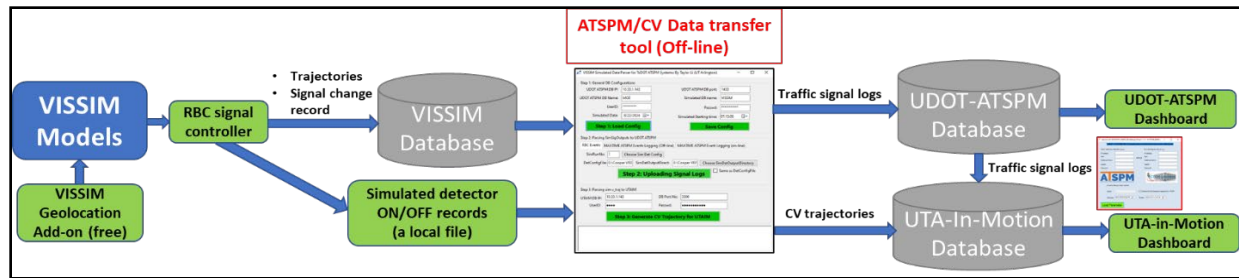


Figure 1-47: Workflow of ATSPM-in-the-loop simulation based on VISSIM+RBC.

To reduce repetitive input and streamline the parsing workflow, the parsing software is designed to load all required parameters automatically. Parsing configurations are stored locally in a JSON-formatted text file, as shown in Fig. 1-48. Users may customize parsing parameters by either editing the JSON file directly using a text editor or importing the file into the GUI, modifying the parameters as needed, and saving the updated configuration. The revised JSON file is then retained locally for use in subsequent parsing operations (see Fig. 1-49).

```

{
  "UDOT_DB_IP": "10.33.1.143",
  "UDOT_DB_PORT": "1433",
  "UDOT_DB_NAME": "MOE",
  "UDOT_DB_USERID": "actionlab",
  "UDOT_DB_PASSWD": "ENGINEERING",
  "SIM_DB_NAME": "VISSIM",
  "SIM_DATE": "08/22/2024",
  "SIM_START_TIME": "07:15:00",
  "SIM_RUN_NO": "1",
  "DET_CONFIG_FILE": "E:\\Cooper VISSIM TxDOT ATSPM project - Swastik's Paper Version\\List of Detector.csv",
  "SIM_DET_OUTPUT_DIR": "E:\\Cooper VISSIM TxDOT ATSPM project - Swastik's Paper Version",
  "UTAIM_DB_IP": "10.33.1.143",
  "UTAIM_DB_PORT": "3306",
  "UTAIM_DB_USERID": "root",
  "UTAIM_DB_PASSWD": "ENGINEERING",
  "MAXTIME_DB_FOLDER": "E:\\Cooper VISSIM TxDOT ATSPM MAXTIME Q FREE VERSION\\Maxtime DB Files",
  "SIM_Date": "08/22/2024"
}

```

Figure 1-48: The local JSON file saves the parsing configuration.

Figure 1-49: Loading the parsing configurations from the local JSON file

1.4.3. Parsing MAXTIME SILS Signal Control Events into the UDOT ATSPM Database

Fig. 1-50 illustrates the workflow for converting simulated traffic signal events generated by MAXTIME SILS into a format compatible with UDOT ATSPM. Unlike RBC, which generates traffic signal events directly within VISSIM, MAXTIME SILS produces signal event logs through dedicated signal control emulators. Each simulated intersection is paired with a separate MAXTIME SILS controller instance.

As described in Sections 1.3.1 and 1.3.2, several supporting elements must be established for each intersection, including web server configuration, intersection identification, and synchronization of TCP socket ports between VISSIM and MAXTIME. These requirements are essential to ensure correct communication and data consistency across the simulation and control environments.

For multi-intersection simulations, the VBScript-based automation process described in Section 1.4.1 can be used to launch and configure the required MAXTIME SILS controller instances. The intersection-numbering and port-assignment conventions remain those specified in Section 1.3.1.

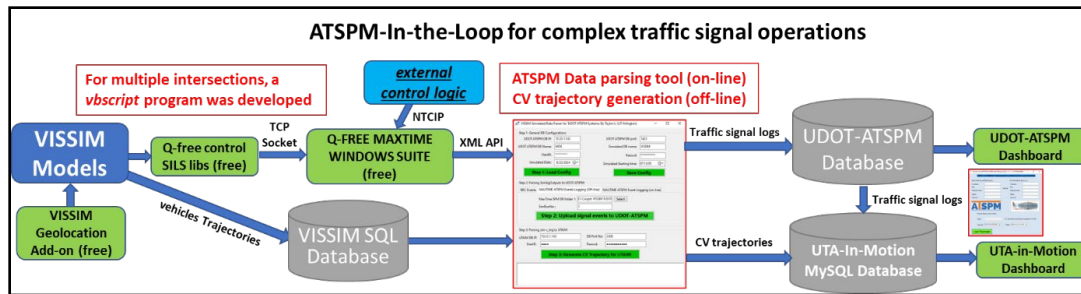


Figure 1-50: Workflow of ATSPM-in-the-loop simulation based on VISSIM+MAXTIME_SILS

Once a MAXTIME SILS controller instance is fully configured, it launches an embedded web server that allows users to access the MAXTIME user interface through a standard web browser for monitoring and validation of control operations. The web server publishes ATSPM-compliant traffic signal events in Extensible Markup Language (XML) format. The developed parsing tool retrieves these events through the MAXTIME XML application programming interface (API) and uploads them directly into the UDOT ATSPM database, as illustrated in Fig. 1-51.

Traffic signal event retrieval from MAXTIME SILS must occur while the VISSIM simulation is running due to inherent data-archiving limitations within the MAXTIME control software.

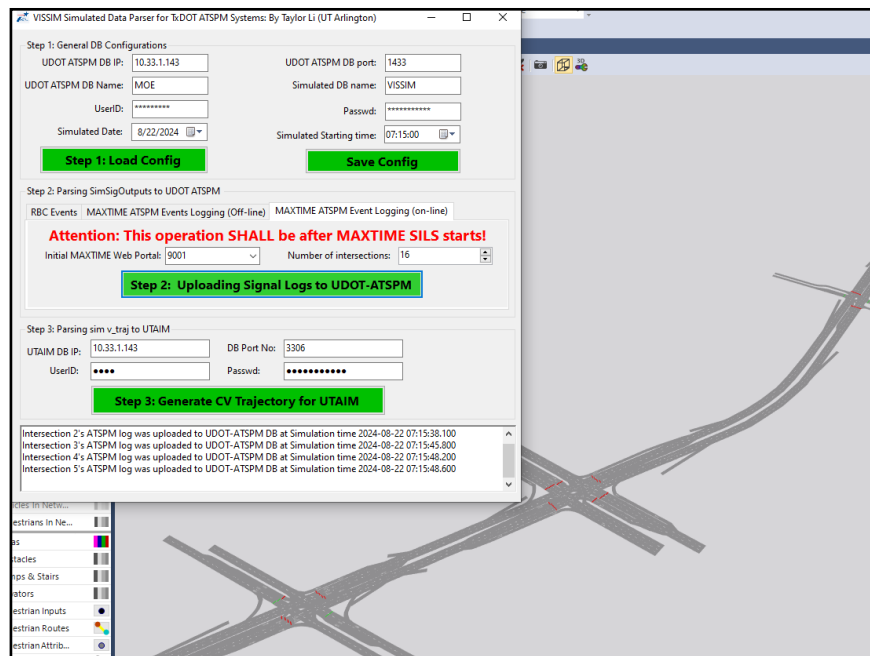


Figure 1-51: Retrieving the ATSPM traffic signal events from the MAXTIME SILS

1.4.4. Parsing Simulated Connected Vehicle Trajectories and Uploading to the UTA-In-Motion Database

One of the ATSPM platforms used in this project is UTA-In-Motion (UTAIM), developed by the project team at The University of Texas at Arlington. UTAIM functions as an extension of the UDOT ATSPM system by incorporating emerging traffic big data sources, such as commercial connected vehicle (CV) datasets, to generate additional signal performance measures (SPMs) across signalized intersections.

Commercial CV trajectory data are typically provided in WGS84 format, represented as (*latitude, longitude, timestamp*). Building upon the WGS84 vehicle trajectories generated and stored within the VISSIM output database, the parsing tool recalculates real-world timestamps for each trajectory waypoint based on the simulated time steps (e.g., 0.1, 0.2, 0.3 s) and a user-specified simulation start time (e.g., 7:15:00 AM on August 22, 2025). To further align simulated trajectories with real-world CV data characteristics, the parsing tool resamples trajectory timestamps from high-frequency simulation intervals (0.1 s) to more realistic reporting intervals ranging from 1 to 3 seconds.

Following these spatial and temporal transformations, the simulated vehicle trajectories conform to the structure and resolution of operational commercial CV datasets and can be directly ingested by the UTAIM database. The transformation of the simulated trajectories into the required UTAIM format is illustrated in Fig. 1-52.

Figure 1-52: Transforming simulated CV trajectories to match the UTAIM system.

1.4.5. Transferring ATSPM Data from UDOT ATSPM to UTA-In-Motion ATSPM

UTAIM operates on a database architecture that differs from the UDOT ATSPM SQL Server backend. UTAIM utilizes MySQL Community Edition and maintains a database separate from the UDOT ATSPM database, even when both platforms are hosted on the same computer. As a result, ATSPM data stored in the UDOT ATSPM SQL Server database are not automatically propagated to the UTAIM MySQL database.

To enable data migration between the two systems, a dedicated ATSPM data transfer tool was developed. Similar to the parsing tools described in Sections 1.4.2 through 1.4.4, users must load a configuration file specifying the source UDOT ATSPM SQL Server database and the destination UTAIM MySQL database, including the required authentication credentials.

Users must also define a From Time and To Time window to restrict the transfer to ATSPM records whose timestamps fall within the specified interval, as shown in Fig. 1-53. The simulation time period must align with this defined window for the transfer operation to execute successfully.

Once the data transfer process is completed, the UTAIM database contains the required ATSPM signal records, along with the simulated CV trajectory data uploaded through the trajectory parsing workflow. This enables UDOT ATSPM and UTAIM to generate, analyze, and visualize the corresponding performance measures across the simulated network.

Figure 1-53: Transferring the ATSPM data from the UDOT ATSPM database to the UTAIM database.

CHAPTER 2: UDOT ATSPM Installation

2.1. Introduction

Chapter 1 established the simulation-based data generation and integration workflow for ATSPM analysis. The workflow included generating high-resolution traffic signal events and simulated connected vehicle trajectories in PTV VISSIM, converting these simulation outputs into ATSPM-compatible formats, and transferring applicable signal records from the UDOT ATSPM SQL Server database to the UTAIM database.

Building on that workflow, this chapter presents the installation and configuration of UDOT ATSPM. It provides a step-by-step procedure for deploying the required software infrastructure on Windows 10 or Windows 11. The installation process includes configuring Microsoft SQL Server or SQL Server Express and SQL Server Management Studio (SSMS), enabling Internet Information Services (IIS) to host the UDOT ATSPM web application, and installing the backend services required to compute measures of effectiveness (MOEs).

The chapter also documents procedures for establishing database connections, deploying the web application, configuring access permissions and system settings, verifying system functionality, and generating traffic signal performance visualizations. Upon completion, the installed environment provides the database, web application, and processing services required to store simulated high-resolution signal data and generate MOEs.

2.2. Prerequisites

Before deploying the UDOT ATSPM system, a stable software and operating system environment must be established. The prerequisites outlined in this section describe the computing platform and supporting services used for system installation and operation.

2.2.1. Windows Operating System

The UDOT-ATSPM system was deployed on a Windows 10 (or Windows 11) operating environment with all available system updates applied. Administrative-level privileges are required to install and configure core system components, including database services, web hosting services, and supporting backend modules.

2.2.2. Internet Information Services (IIS)

The UDOT ATSPM web application is hosted using Internet Information Services (IIS). IIS provides the web server framework required to serve the ATSPM user interface and backend

services. The configuration and activation of IIS components relevant to ATSPM deployment are documented in Section 2.4.

2.2.3. SQL Server and SQL Server Management Studio (SSMS)

UDOT ATSPM relies on Microsoft SQL Server or SQL Server Express 2012 or later to store and manage high-resolution traffic signal event data and computed performance measures. SSMS is used to administer database objects, configure user access, and verify database connectivity during system deployment. SQL Server 2022 was used in this implementation. The installation process and initial setup are illustrated in Fig. 2-1.

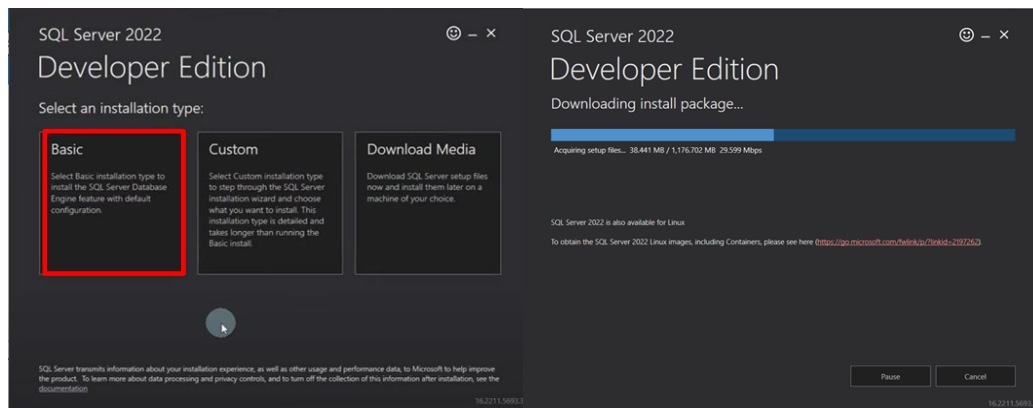


Figure 2-1: SQL Server installation process showing the installer's initial setup screen.

Fig. 2-2 shows the installation of SSMS Studio from the official Microsoft page.

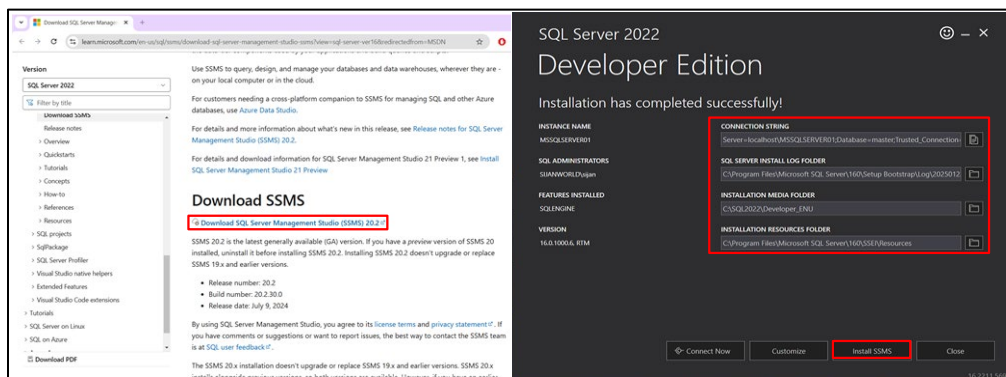


Figure 2-2: SQL Server Management Studio installation process.

Following installation, the SQL Server instance was verified to be operational by confirming that the MSSQLSERVER service was running, as shown in Fig. 2-3.

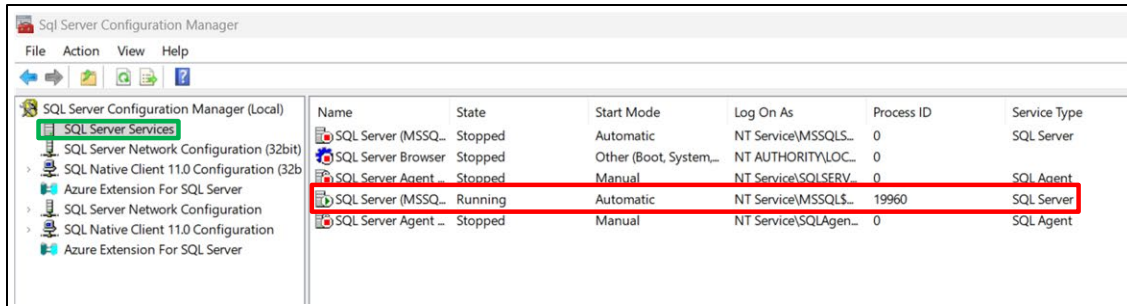


Figure 2-3: Verification of the MSSQLSERVER service running.

2.3. Database Configuration

Proper database configuration is required to support reliable storage, processing, and retrieval of ATSPM traffic signal data. This section documents the database access model implemented for the UDOT ATSPM system, including user authentication and permission assignment.

2.3.1. Creating a New SQL Server Login

To support ATSPM data ingestion and processing, a dedicated SQL Server login was created. This account, referred to as SPM, is used by the ATSPM backend services and associated parsing tools to interact with the SQL Server database, as shown in Fig. 2-4.

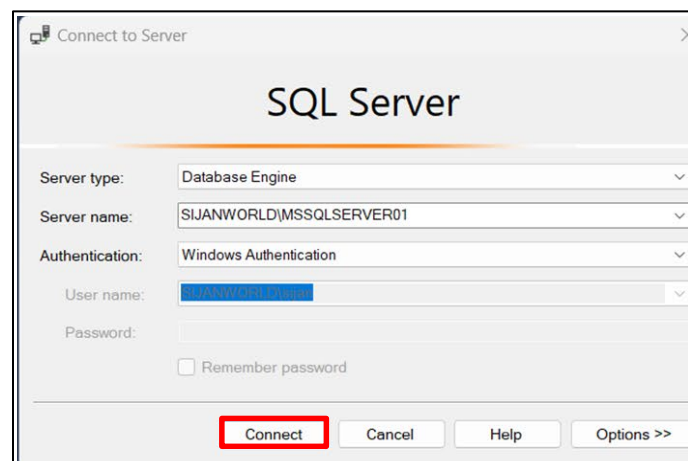


Figure 2-4: Initial connection dialog for SQL Server login configuration.

The login was created using SQL Server Authentication to allow controlled access independent of Windows user accounts. For this implementation, password policy enforcement was disabled to avoid unintended service disruptions caused by automatic password expiration. The login creation process is illustrated in Fig. 2-5.

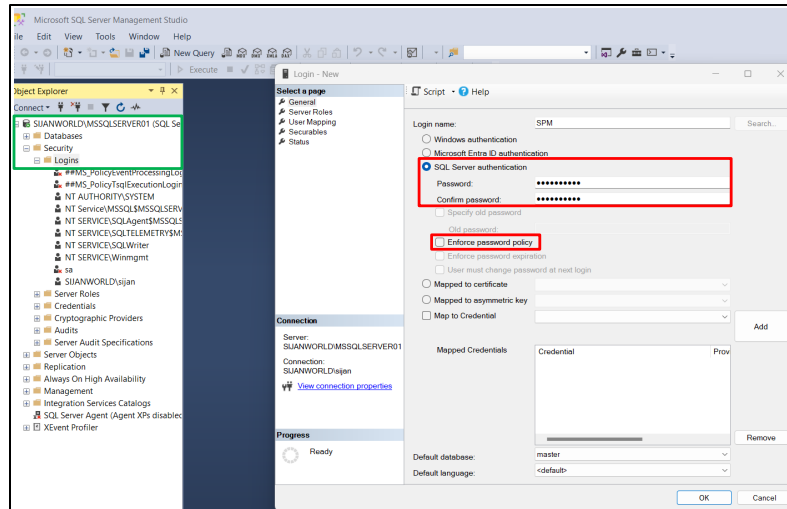


Figure 2-5: Creation of the SPM SQL Server login.

2.3.2. Assigning Permissions

Appropriate server-level and database-level permissions were assigned to the SPM login to enable ATSPM operations. At the server level, the account was granted the dbcreator role to allow database creation and management without requiring full administrative privileges.

At the database level, the SPM user was assigned roles necessary for ATSPM data access and processing, including:

- db_datareader, enabling read access to all database tables;
- db_datawriter, enabling insertion and update of traffic signal event records;
- db_owner, providing full control over database objects to support schema updates and automated processing tasks.

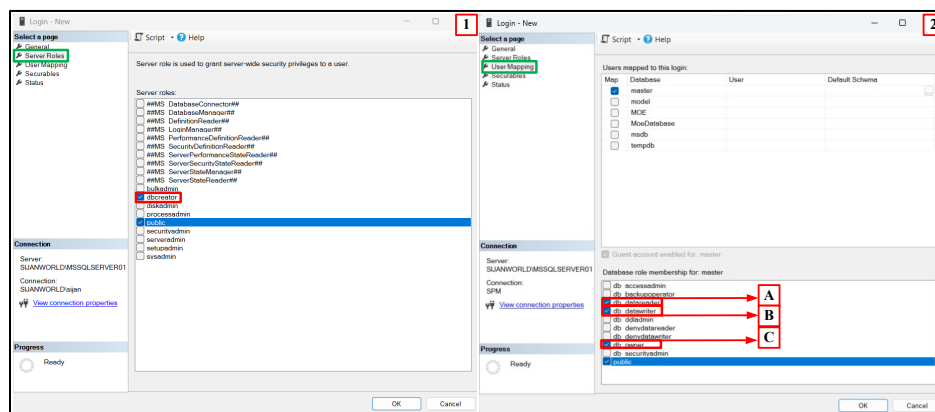


Figure 2-6: User mapping and database role configuration for the SPM login.

The user mapping and role assignments are shown in Fig. 2-6, and successful completion of the configuration is confirmed in Fig. 2-7.

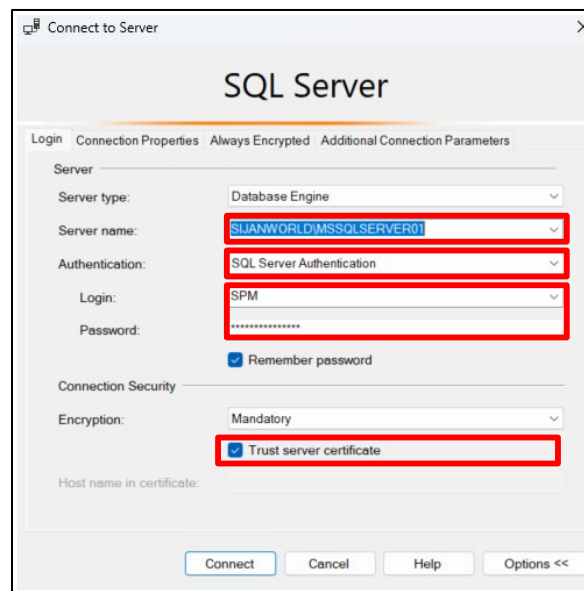


Figure 2-7: Confirmation of the completed SQL Server login configuration.

2.4. Installing IIS Server and Necessary Windows Packages

As noted in Section 2.2.2, IIS is required to host the ATSPM web application and support server-side functions.

2.4.1. Enabling IIS and Required Features

The following IIS components and Windows features must be enabled to support UDOT-ATSPM deployment:

1. .NET Framework (Advanced Services): enable ASP.NET 4.8 and required WCF Services (HTTP Activation and TCP Port Sharing).
2. IIS Management Tools: enable IIS Management Console, IIS Management Scripts and Tools, and IIS Management Service to support configuration and maintenance.
3. World Wide Web Services: enable Static Content (required for hosting ATSPM web resources) and HTTP Logging for request tracking and diagnostics.
4. Security: enable Basic Authentication and Request Filtering; enable Windows Authentication where available to support enterprise access control.

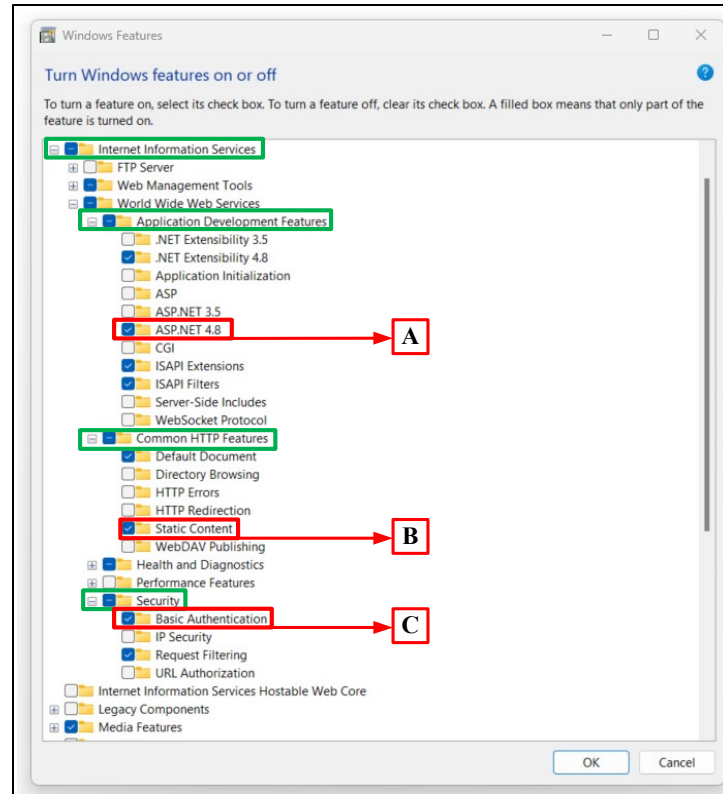


Figure 2-8: Enabling IIS and required Windows features in the Windows Features dialog.

Important: After enabling IIS features, the installation must be applied at the Windows level (a restart may be required). IIS availability should then be verified through the IIS management interface.

2.5. Installing UDOT ATSPM Applications in IIS

After IIS is configured, the UDOT ATSPM web application must be deployed to enable traffic signal performance monitoring and visualization. This section documents the approach to deploying the ATSPM web application, configuring database connectivity, and selecting an appropriate IIS application pool.

2.5.1. Copying ATSPM Web Application to IIS

The ATSPM web application package, provided as the ATSPM-Web-Rebuild directory, was deployed within the IIS web root to enable hosting by the IIS server.

- The ATSPM-Web-Rebuild folder was placed under the IIS root directory (*C:\inetpub\wwwroot*).

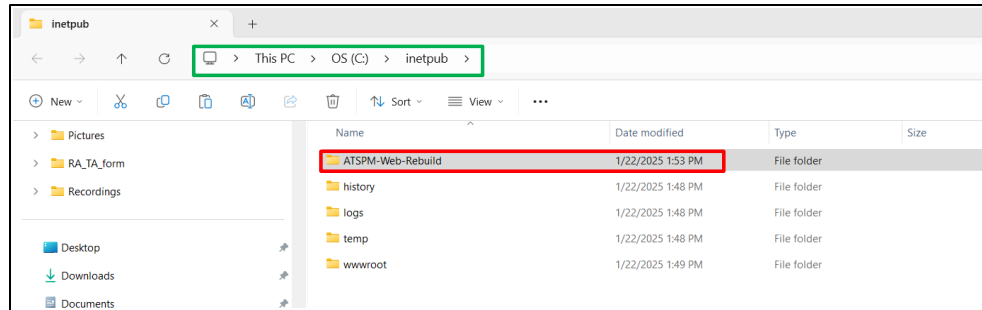


Figure 2-9: Placement of the ATSPM-Web-Rebuild folder in the IIS root directory.

- Folder-level security permissions were configured to ensure proper execution of the web application.
- The following IIS system accounts were granted Full Control access to the application directory:
 - IUSR
 - IIS_IUSRS

These permissions allow IIS to read, write, and execute the application files required for normal ATSPM operation, as shown in Fig. 2-10.

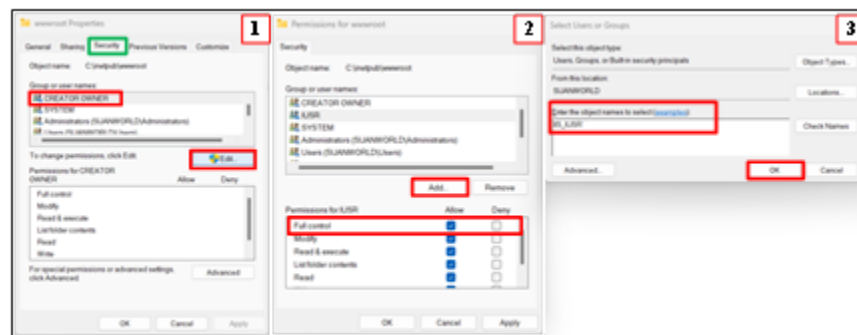


Figure 2-10: Folder permission configuration for IIS system accounts.

2.5.2. Configuring Web.Config

The Web.config file defines database connectivity parameters for the ATSPM web application. To connect the application to the local UDOT ATSPM SQL Server database, the following configuration elements were updated:

- Data Source was set to match the local SQL Server instance name.
- Initial Catalog was set to MOE, corresponding to the UDOT-ATSPM database.
- User ID was set to SPM, the dedicated SQL Server login created for ATSPM operations.

- The corresponding SQL Server authentication password was provided as shown in Fig. 2-11.

These settings enable the ATSPM web application to communicate with the backend SQL Server database and retrieve traffic signal event data and performance measures.



Figure 2-11: Updated Web.config file showing the ATSPM database connection string.

2.5.3. Configuring Application Pool

A dedicated IIS application pool was configured to isolate the ATSPM web application and ensure stable runtime behavior.

- A new application pool named .Net V4.5 was created.

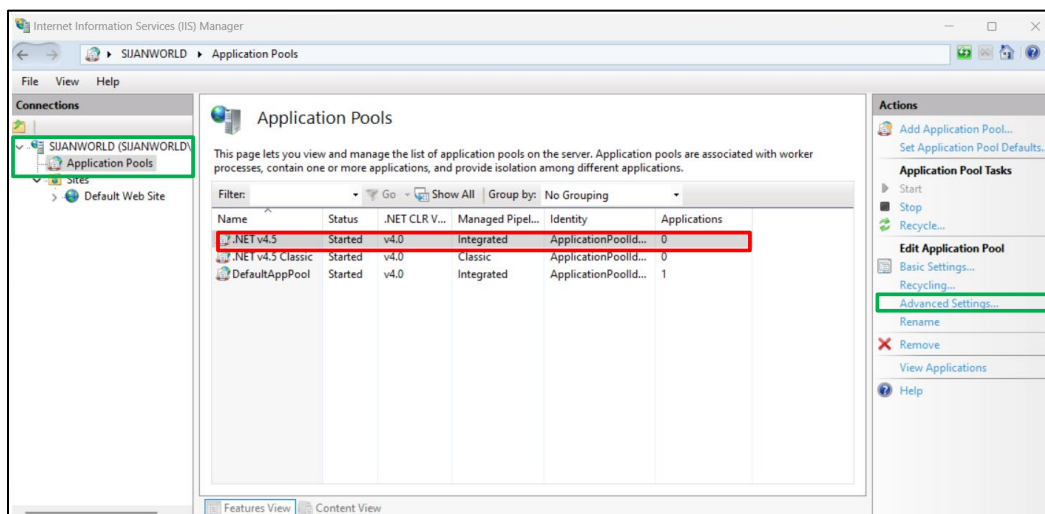


Figure 2-12: Creation of a dedicated IIS application pool for ATSPM.

- The application pool was configured to use .NET CLR Version v4.0, consistent with ATSPM application requirements.
- 32-bit application support was enabled to ensure compatibility with dependent libraries.

This configuration allows the ATSPM web application to operate independently of other IIS-hosted services and supports reliable execution under varying workloads as of Fig. 2-13.

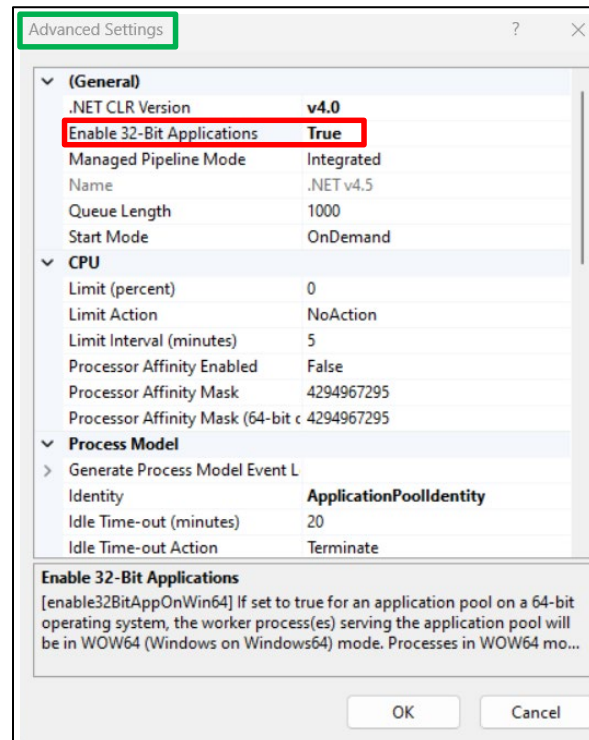


Figure 2-13: Application pool settings with 32-bit applications enabled.

2.6. Adding UDOT ATSPM Applications in IIS

After deploying the web application files and configuring the application pool as described in Section 2.5, the UDOT ATSPM application must be registered within IIS to enable web-based access. This section documents how the application was added to the IIS environment and verified for operation.

- The ATSPM application was registered under the Default Web Site (or an equivalent IIS site) to make it accessible through a browser.
- A dedicated application alias and application pool were assigned to ensure proper isolation and runtime compatibility.
- The physical path to the deployed ATSPM web application files was explicitly linked to the IIS application as shown in Fig. 2-14.

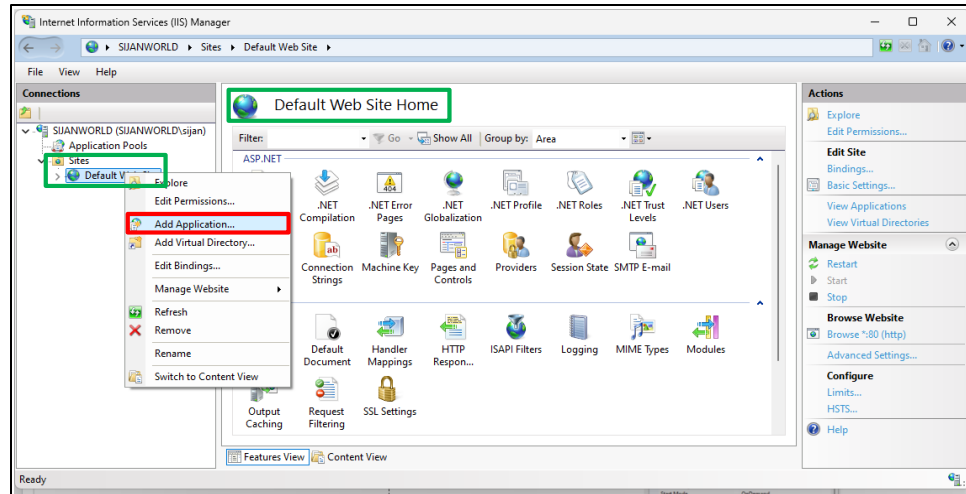


Figure 2-14: Illustrates the IIS interface used to add the ATSPM application to the web server.

2.6.1. Application Configuration Summary

The following configuration parameters were used when adding the ATSPM application in IIS:

- Application Alias: ATSPM
 - This alias defines the URL path used to access the application
 - Example: [*http://localhost/ATSPM*](http://localhost/ATSPM)
- Application Pool: .Net V4.5
 - Corresponds to the application pool configured in Section 2.5.3.
 - Ensures compatibility with the ATSPM web application framework
- Physical Path:
 - C:\inetpub\wwwroot\ATSPM-Web-Rebuild
 - Points to the directory containing the deployed ATSPM web application files

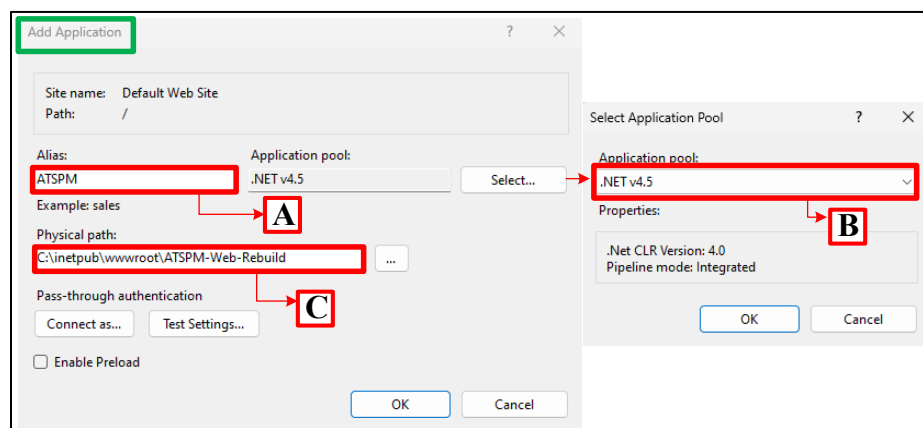


Figure 2-15: Shows the application settings used to link the alias, application pool, and physical path.

2.6.2. Verification of Deployment

After the application was added to IIS, successful deployment was verified by accessing the ATSPM web interface through a local web browser, as in Fig. 2-16.

- The application was accessed using the URL: [**http://localhost/ATSPM**](http://localhost/ATSPM)
- Successful loading of the ATSPM web interface confirmed:
 - Correct IIS application registration
 - Proper application pool assignment

Database connectivity and backend service operation must be verified separately using the procedures provided later in this chapter.

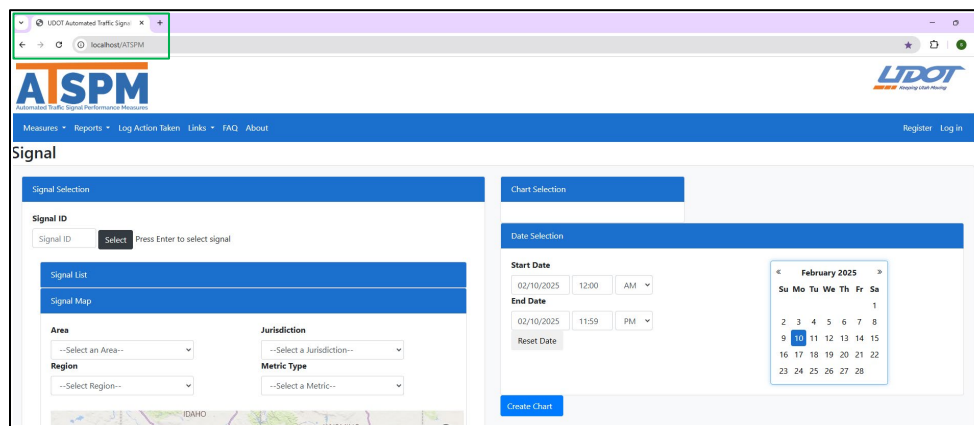


Figure 2-16: Presents the ATSPM web application interface displayed after successful deployment.

2.7. Installing Backend MOE Modules in IIS

In addition to the UDOT ATSPM web application, the MOE backend modules must be deployed to support the computation and processing of ATSPM performance measures. These modules are implemented as Windows Communication Foundation (WCF) services and are hosted within IIS, as shown in Fig. 2-17.

2.7.1. MOE Backend Deployment Overview

The MOE backend service, provided as the MOEWcfServiceLibrary package, was deployed and registered in IIS using the following configuration approach:

- The MOEWcfServiceLibrary directory was placed under the IIS web root

(*C:\inetpub\wwwroot*) to enable hosting by the IIS server.

- Folder-level security permissions were configured to allow proper execution of backend services.
- The service was added as an IIS application and assigned an appropriate application pool.

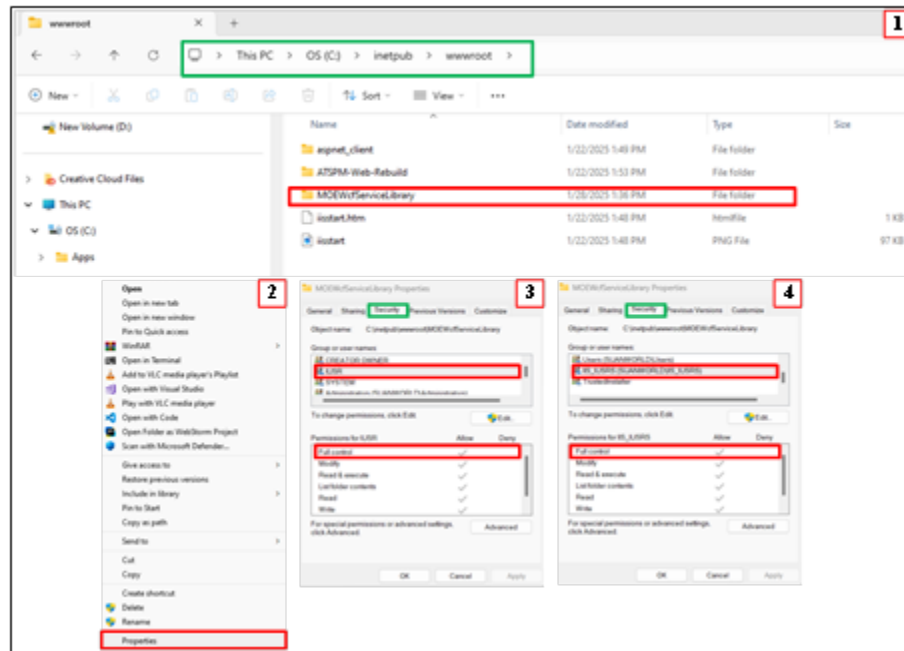


Figure 2-17: MOEWcfServiceLibrary folder and permissions.

2.7.2. MOE Backend Application Configuration

To enable IIS hosting of the MOE backend services, the following configuration parameters were applied:

- Application Alias: MOEWcfServiceLibrary
 - Defines the service endpoint used by ATSPM backend components.
- Application Pool: .NET Framework 4.5
 - Ensures compatibility with the WCF service framework.
- Physical Path:
 - C:\inetpub\wwwroot\MOEWcfServiceLibrary
 - Points to the directory containing the MOE backend service files.
- Folder Permissions:
 - The following IIS system accounts were granted Full Control access:
 - IUSR
 - IIS_IUSRS

These settings ensure that IIS can execute the backend WCF services and allow the ATSPM system to compute and retrieve MOE results as shown in Fig. 2-18.

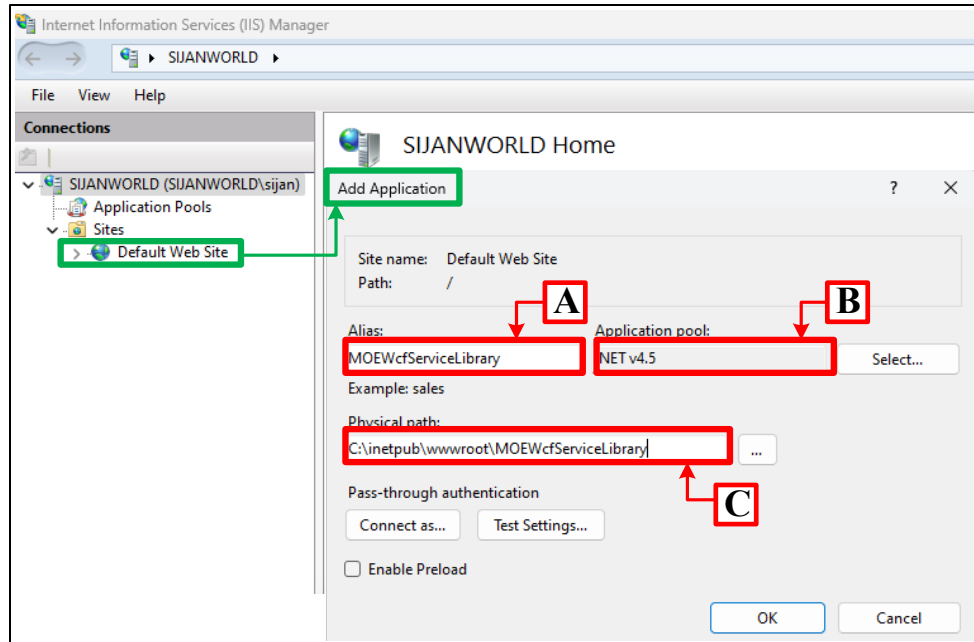


Figure 2-18: Shows the MOEWcfServiceLibrary application registered within IIS.

2.8. Installing Backend SPM Images in IIS

In addition to the ATSPM web application and MOE backend services, a dedicated directory is required to store and serve SPM image outputs generated by the ATSPM system. These images are accessed by the ATSPM web interface and must therefore be hosted through IIS.

2.8.1. SPM Images Deployment Overview

The SPMImages directory was created and registered in IIS using the application settings described in Section 2.8.2.

- A dedicated directory named SPMImages was created at C:\ATSPMroot\SPMImages.
- Folder-level security permissions were configured to allow IIS to read, write, and serve image files generated by backend processing modules.
- The directory was registered as an IIS application to enable web-based access.

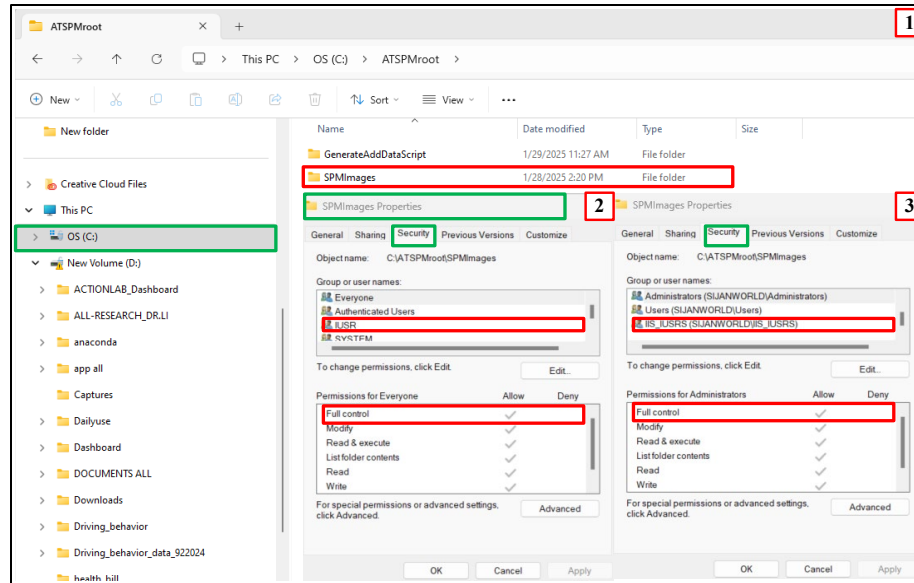


Figure 2-19: SPMImages folder creation and permission configuration.

2.8.2. SPM Images Application Configuration

To enable IIS hosting of SPM image outputs, the following application settings were applied:

- Application Alias: SPMImages
 - Defines the URL path used by the ATSPM web application to retrieve image files.
- Application Pool: .NET Framework 4.5
 - Consistent with the application pool used for the ATSPM web application and backend services.
- Physical Path:
 - C:\ATSPMroot\SPMImages
 - Points to the directory where SPM image files are stored.
- Folder Permissions:
 - The following IIS system accounts were granted Full Control access:
 - IUSR
 - IIS_IUSRS

These settings ensure that backend modules can generate image files and that IIS can serve them to the ATSPM web interface without access restrictions, as in Fig. 2-20.

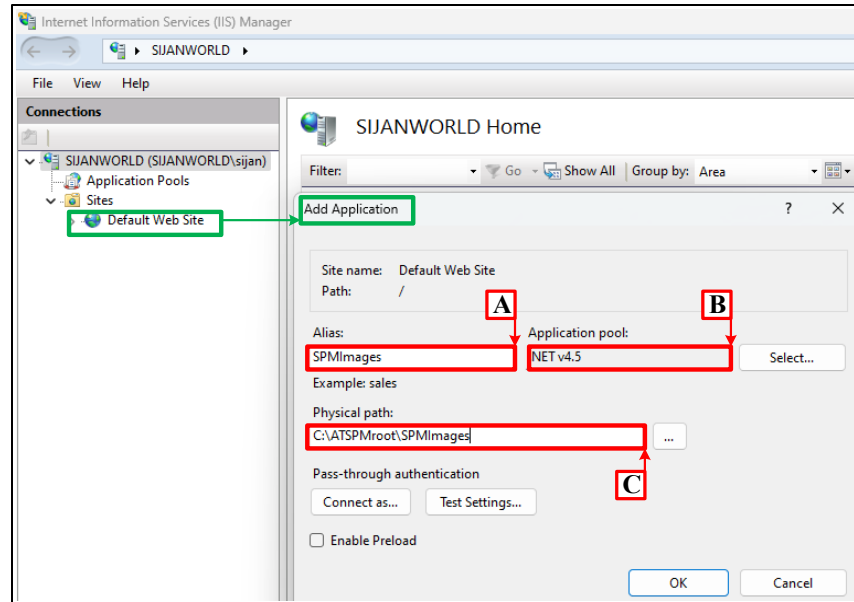


Figure 2-20: Shows the SPMImages directory registered as an application within IIS.

2.9. Registering UDOT ATSPM Account and Configuring SPM Image Path

After deploying the UDOT ATSPM web application and backend services as described in Sections 2.5 through 2.8, user access and system paths must be configured to enable full functionality of the platform. This section documents the approach to registering user accounts, assigning permissions, and configuring the image storage path needed to display SPM outputs.

2.9.1. Account Registration and Role Assignment

- A new UDOT-ATSPM user account was created through the ATSPM web interface.
- Following registration, the account was assigned all available user roles to ensure full access to system features, including administration and performance visualization.
- Role assignment was verified through the administrative user management interface.

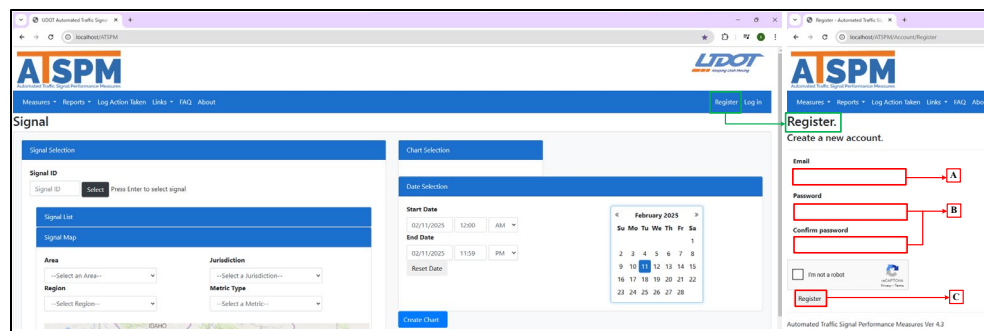


Figure 2-21: Shows the user registration interface used to create a new ATSPM account.

To perform administrative configuration tasks, the system was accessed using the default administrator credentials:

- Username: defaultadmin@spm.gov
- Password: L3tM3in!

These credentials provide full administrative privileges and are used to manage users and system-level settings, as in Fig. 2-22.

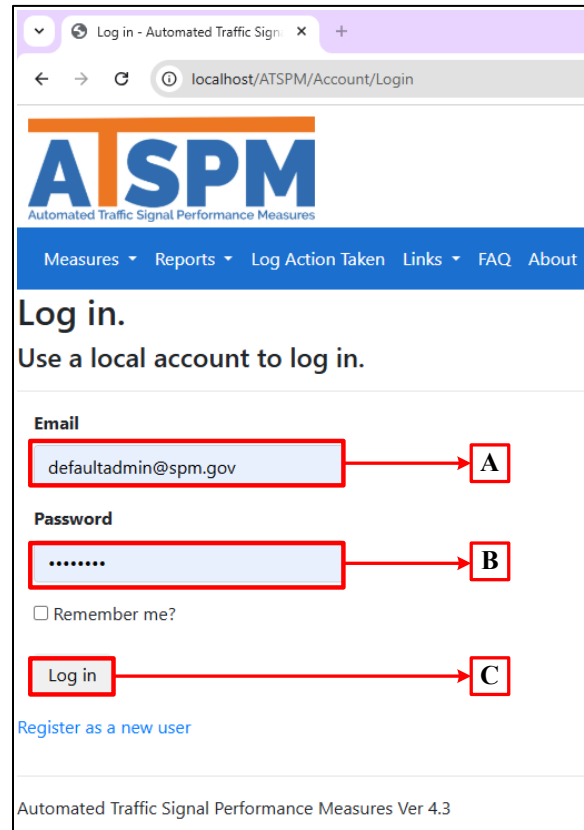


Figure 2-22: Illustrates a successful login using the default administrator account.

2.9.2. SPM Image Path Configuration

To enable proper storage and retrieval of SPM-generated images, the system-wide image settings were updated within the ATSPM administrative configuration.

The following parameters were configured under General Settings:

- Image Path:
 - Set to ATSPMroot/SPMImages/
 - Defines the local directory used to store generated SPM images.

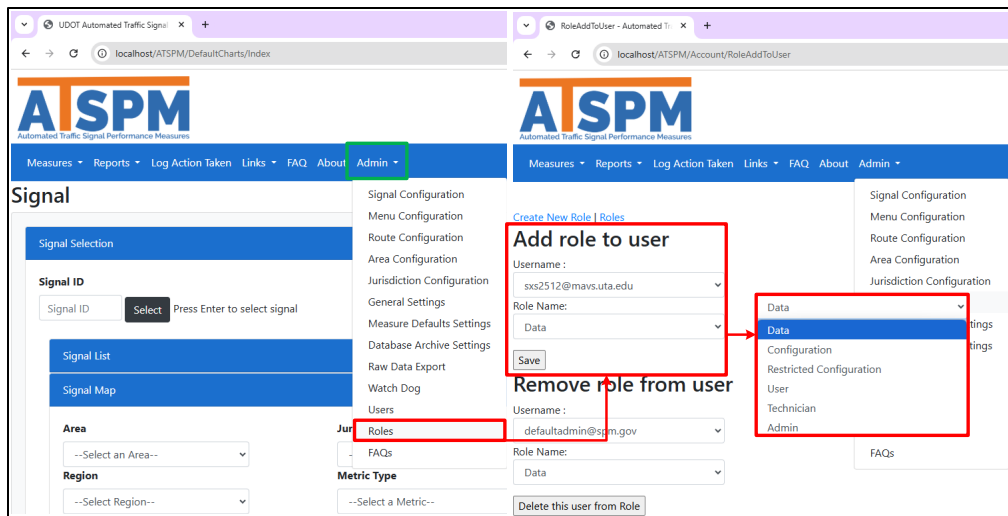


Figure 2-23: Assigning roles to the newly registered ATSPM account within the admin section.

- Image URL:
 - Set to the server-accessible URL corresponding to the SPMImages application.
 - The server address was derived from the system's network configuration (IPv4 address).

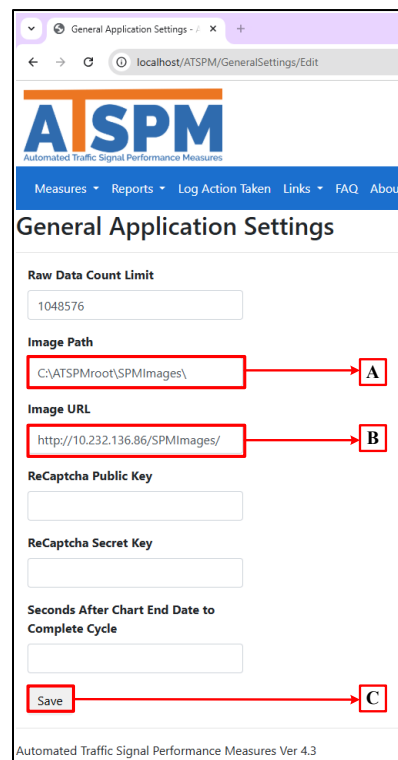


Figure 2-24: SPM image path and URL settings.

2.10. Testing SPM Image and MOE Services

After deployment of the UDOT ATSPM web application, backend MOE services, and SPM image components, system functionality must be verified to confirm correct service hosting, configuration of database connectivity, and front-end integration. This section documents the verification procedures used to test the MOE WCF services and SPM image-generation workflow.

2.10.1. Verification of MOE WCF Services

Fig. 2-25 shows that the MOE backend services were first validated at the file and service levels.

- The MOE service directory C:\inetpub\wwwroot\MOEWcfServiceLibrary was verified to contain the required WCF service files:
 - MOEWcfServiceLibrary.LinkPivotService.svc
 - MOEWcfServiceLibrary.MetricGenerator.svc

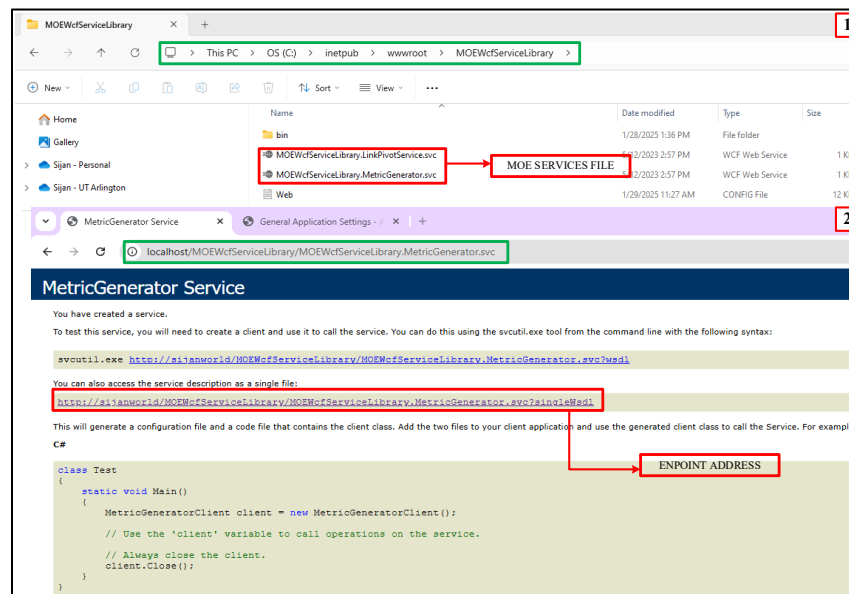


Figure 2-25: Testing MOE WCF service endpoints showing the service status page in a browser.

These files represent the backend services responsible for link-based performance calculations and MOE generation.

Service availability was then validated through a web browser using the following endpoints:

- LinkPivotService: [*http://localhost/MOEWcfServiceLibrary/MOEWcfServiceLibrary.LinkPivotService.svc*](http://localhost/MOEWcfServiceLibrary/MOEWcfServiceLibrary.LinkPivotService.svc)

- MetricGenerator:

[*http://localhost/MOEWcfServiceLibrary/MOEWcfServiceLibrary.MetricGenerator.svc*](http://localhost/MOEWcfServiceLibrary/MOEWcfServiceLibrary.MetricGenerator.svc)

Successful access to these URLs confirmed that the WCF services were correctly hosted in IIS and operational.

2.10.2. Validation of Database Connectivity and Service Linking

To ensure that the MOE services and ATSPM web application were correctly connected to the backend SQL Server database, configuration files were reviewed and updated as needed.

- The Web.config file within C:\inetpub\wwwroot\MOEWcfServiceLibrary was configured to reference the correct SQL Server instance, using the appropriate SPM database user credentials.

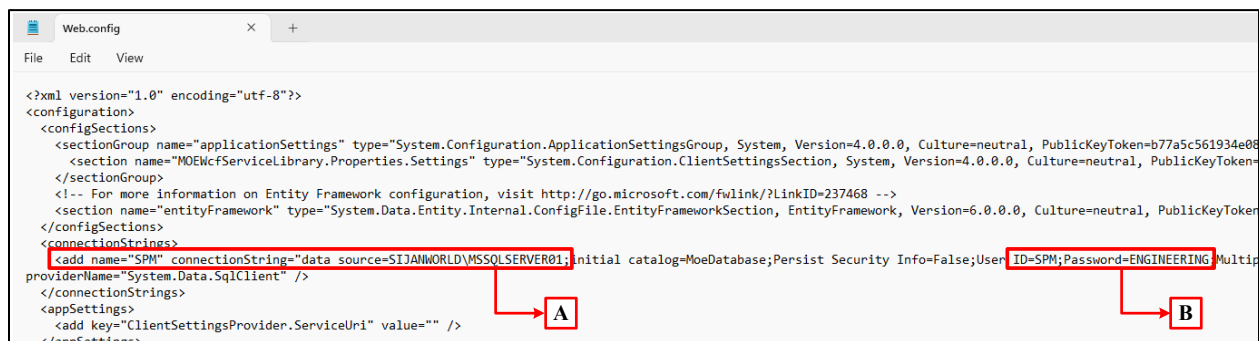


Figure 2-26: Web.config connection-string updates for backend integration.

- The Web.config file within C:\inetpub\wwwroot\ATSPM-Web-Rebuild was configured to reference the correct MOE service endpoints, including:
 - LinkPivotService
 - MetricGenerator

These endpoint URLs were aligned with the validated service addresses to ensure proper communication between the ATSPM front-end application and the backend MOE services.



Figure 2-27: Endpoint address updated based on metric service generator.

2.11. Application Configuration

After successful deployment and verification of the UDOT ATSPM web application and backend services as described in Sections 2.5 through 2.10, several application-level configurations were applied to customize the user interface and default visualization settings. This section documents the configuration changes implemented for branding and map initialization.

2.11.1. Webpage Title Configuration

- The ATSPM webpage title was customized by updating the ViewBag.Title parameter in the application view file located under: C:\inetpub\wwwroot\ATSPM-Web-Rebuild\Views\DefaultCharts.
- This modification allows the displayed webpage title to reflect the hosting organization or specific deployment context of the ATSPM system.

The title update process is illustrated in Fig. 2-28.

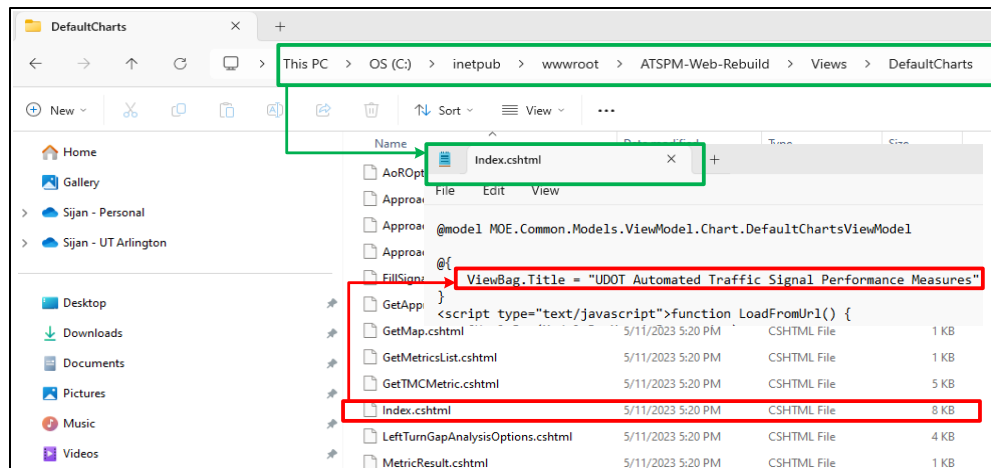


Figure 2-28: Editing the homepage title to reflect the desired webpage title.

2.11.2. Logo Configuration

- The default ATSPM logo was replaced to support organization-specific branding.
- The logo image stored under: C:\inetpub\wwwroot\ATSPM-Web-Rebuild\Images was updated with a customized logo file.
- The replacement image was configured to match the expected filename (NewUDOTLogo) or, alternatively, the application reference was updated accordingly.

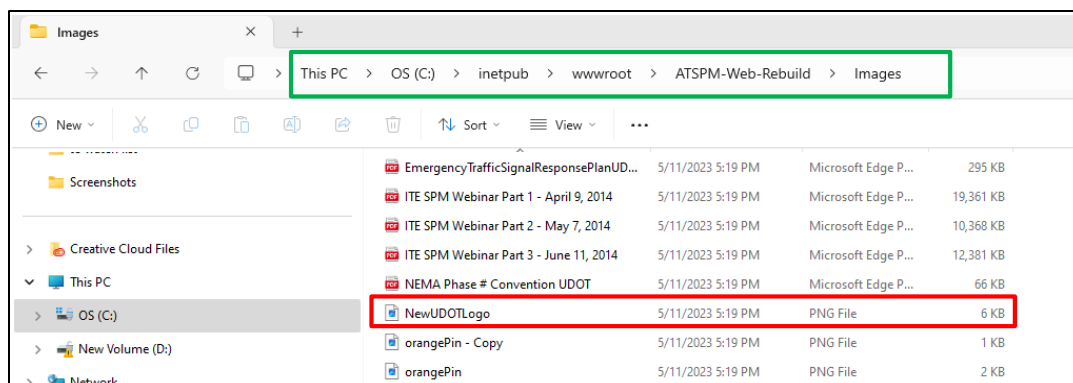


Figure 2-29: Illustrates the process of replacing the default logo image within the application directory.

2.11.3. Default Map Location Configuration

- The default map view displayed by the ATSPM web interface was updated by modifying the map initialization parameters in Map.js, located under: C:\inetpub\wwwroot\ATSPM-Web-Rebuild\Scripts.
- The latitude and longitude values within the GetMap() and GetRouteMap() functions were updated to reflect the desired default geographic location.

- The map zoom level was also adjusted to control the initial view extent, with a zoom level of 12 used in this implementation.

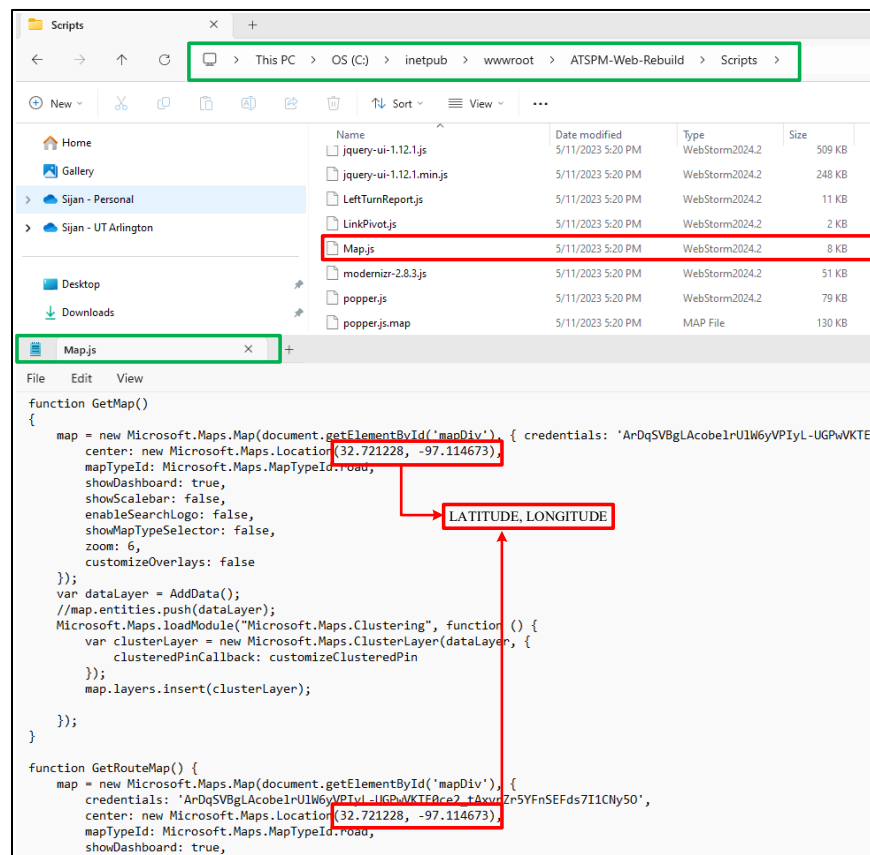


Figure 2-30: Default map location configuration in Map.js.

Figs. 2-30 and 2-31 show the code modifications and the resulting changes in the UDOT ATSPM web interface before and after the configuration update.

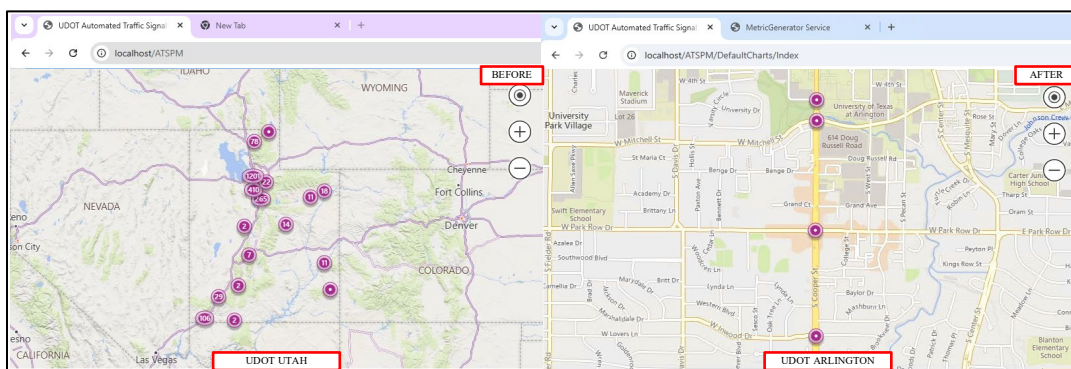


Figure 2-31: Updated ATSPM map view.

2.12. Generating Additional Intersection Data

To support visualization of intersections within the ATSPM web interface, additional intersection data were generated using a dedicated utility provided with the installation package. This process produces intersection markers that are subsequently displayed within the ATSPM system, as illustrated in Fig. 2-32.

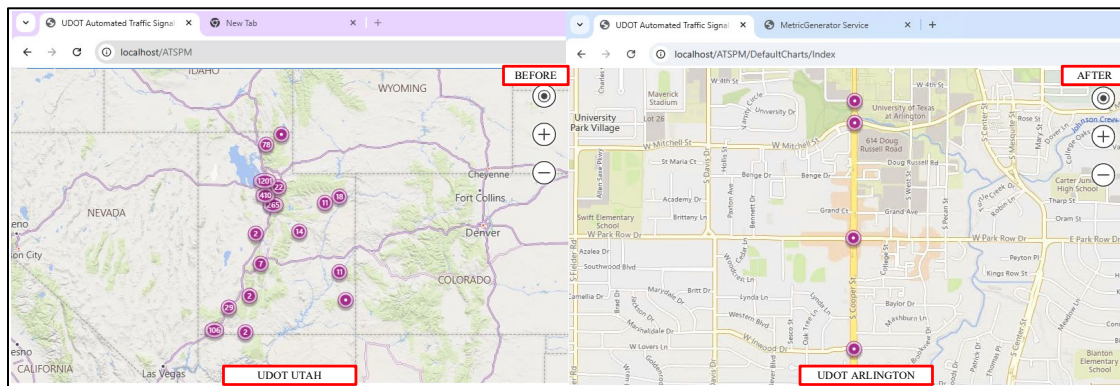


Figure 2-32: GenerateAddDataScript execution.

- The GenerateAddDataScript directory was placed under C:\ATSPMroot to establish a local execution environment for the data generation utility.
- The script configuration file located under C:\ATSPMroot\GenerateAddDataScript\bin\Release was updated to reflect the local system environment, including:
 - connectionString, configured to reference the appropriate ATSPM SQL Server database.
 - Path1, set to C:\inetpub\wwwroot\ATSPM-Web-Rebuild\Scripts\ which defines the destination directory for the generated intersection data.
- After configuration, the GenerateAddDataScript.exe utility was executed to generate the required intersection marker files.

Successful execution of the script resulted in the creation of intersection marker data, enabling intersections to be properly displayed and referenced within the ATSPM web interface.

2.13. Adding Intersection at UDOT ATSPM

After completing system installation and backend configuration, intersections must be defined within the UDOT-ATSPM system to enable signal performance monitoring and visualization. This section documents the configuration elements used to register intersections and associate signal, phase, and detector information.

- Intersections were added through the Signal Configuration module within the administrative interface.
- Each intersection was assigned a unique Signal ID to ensure unambiguous identification within the ATSPM database, as in Fig. 2-33.

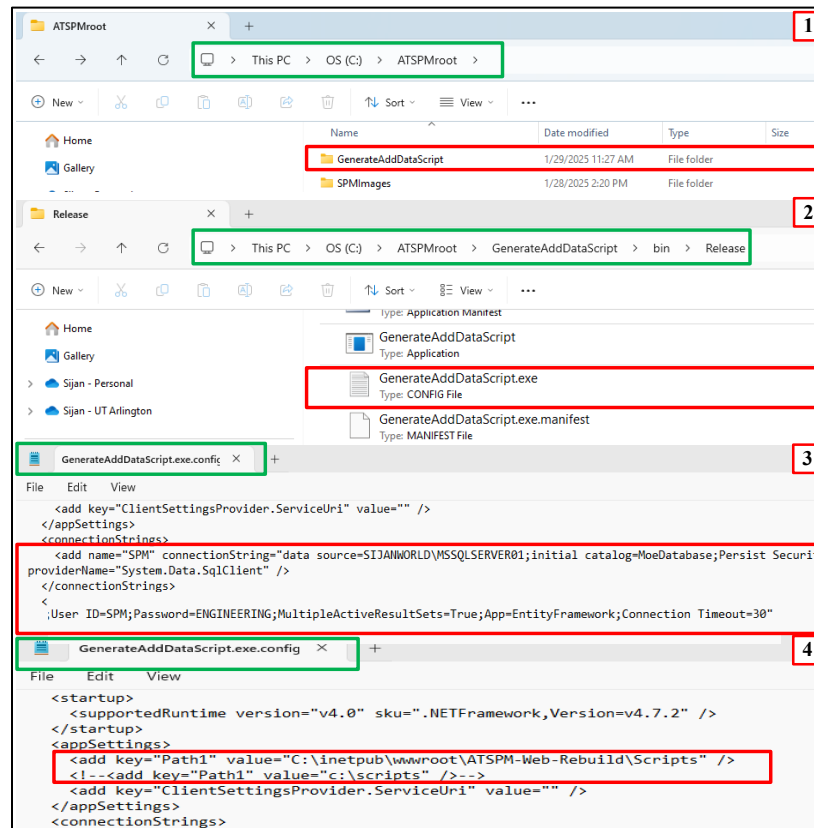


Figure 2-33: Illustrates the interface used to create a new signal entry.

2.13.1. Intersection and Controller Configuration

The following intersection attributes were configured to establish spatial and operational context:

- Intersection naming
 - Primary and Secondary Street names were defined to reflect field conditions.
- Geographic location
 - Latitude and longitude coordinates were entered to support accurate map-based visualization.
- Controller type
 - The ASC/3 controller type was selected, consistent with commonly deployed traffic signal controllers supported by ATSPM.

2.13.2. Signal Phase Configuration

Signal phase information was defined to represent traffic movements and operational logic at each intersection.

- Each phase was configured with:
 - Movement direction (e.g., northbound, southbound)
 - Movement type (e.g., left turn, through)
 - Operation mode (protected or permissive)
 - Approach speed, used in performance calculations
- Additional elements, such as pedestrian phases and overlaps, were configured where applicable.

Accurate phase configuration is essential for the correct generation of signal performance measures and intersection images, as in Fig. 2-34.

The screenshot displays the 'Signal 1' configuration interface. At the top, there are buttons for 'Delete This Version', 'Add New Version', 'Copy Signal', 'Delete This Signal', and 'Save'. The main form is divided into several sections:

- Version Information:** Includes 'Version' (2/13/2025 - Create New), 'Version Start' (2025-02-13), 'Version Label' (Create New), and 'Version Label' (B).
- Intersection Name:** 'INTERSECTION NAME' (A) is highlighted. It includes 'Primary Name' (North Cooper St), 'Secondary Name' (W Division St), 'Region' (Region 2), 'Areas' (No Areas Selected), and 'Jurisdiction' (Default).
- Coordinates:** 'COORDINATES' (B) is highlighted. It includes 'Latitude' (32.73912) and 'Longitude' (-97.11444).
- Controller Type:** 'ASC3' (C) is highlighted.
- Configuration Table:** A table with columns for 'Phase/Direction', 'Description', 'Protected Phase', 'Permissive Phase', 'Pedestrian Phase', and 'Ped. Detector(s)'. It includes a 'New Phase/Direction' button and a 'Copy Delete' button.
- Detector 126:** A section for configuring a specific detector. It includes 'Det Channel' (25), 'Detection Types' (Advanced Count, Advanced Speed, Lane-by-lane Count, Lane-by-lane with Speed Restriction, Stop Bar Presence, Advanced Presence), 'Detection Hardware' (Wavetronix Matrix), 'Latency Correction' (0), 'Lane Number (Lane-by-lane Count)' (3), 'Movement Type (Lane-by-lane Count)' (Thru), 'Lane Type (Lane-by-lane Count)' (Vehicle), and 'Distance To Stop Bar (Advanced Count)' (221). A note 'SEE FIGURE BELOW' is present.
- Configuration Table (Bottom):** A table listing phases and their associated detectors. It includes a 'Phase/Direction' column and a 'Copy Delete' column. A note 'ALL THE PHASES, ALONG WITH THE DETECTORS, NEED TO BE ADDED' is present.

Figure 2-34: Detailed view of the signal phase configuration for a new intersection signal.

2.13.3. Detector Configuration

Detector information was defined to support vehicle detection, phase actuation, and performance metric computation.

Each detector was configured with:

- Detector channel name
- Detection type (e.g., presence, count)
- Detection hardware type
- Associated lane number
- Movement type
- Distance to stop bar

For advanced count detectors, the distance to the stop bar was measured in the VISSIM network using the model scale tool. The VISSIM scale tool was used to measure the distance between the stop-bar detector and the advance detector location. Accurate specification of detector attributes is critical for:

- Correct generation of intersection images,
- Reliable traffic flow representation, and
- Proper operation of advanced features such as preemption, priority, and freight signal priority, as illustrated in Fig. 2-35.

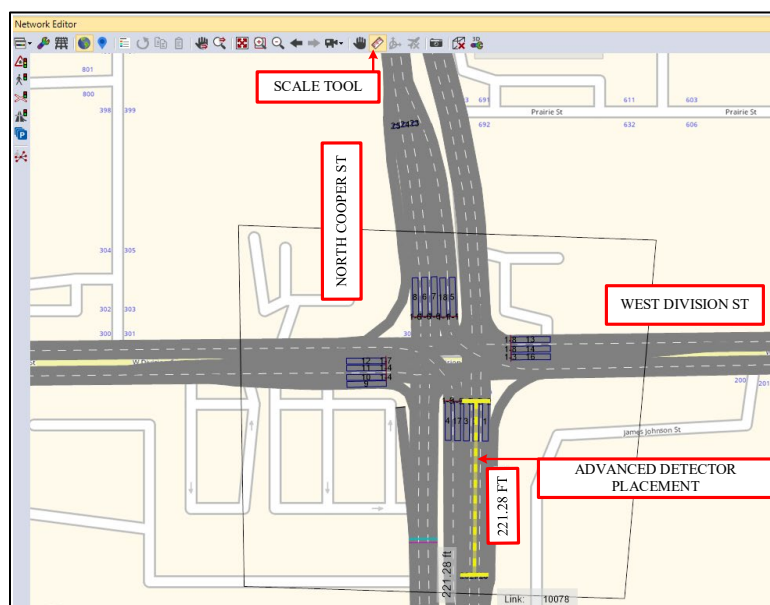


Figure 2-35: Distance to stop bar measured from VISSIM

Chapter 3: UTA-In-Motion (UTAIM) Installation Manual

3.1. Introduction

UTAIM is an advanced add-on module developed by UTA to extend the capabilities of UDOT ATSPM through the integration of emerging traffic big data sources, including CV trajectory data. UTAIM processes CV trajectories in WGS84 format, enabling alignment between simulated vehicle trajectories and real-world commercial CV datasets through a structured data-parsing workflow.

Unlike UDOT ATSPM, which uses Microsoft SQL Server, UTAIM uses a separate MySQL database architecture. As described in Section 1.4.5, a dedicated data transfer tool transfers applicable ATSPM signal records from the UDOT ATSPM SQL Server database to the UTAIM MySQL database, where they can be combined with simulated CV trajectory data for additional analysis and visualization.

By leveraging scalable data-archiving mechanisms and visualization tools, UTAIM supports the generation of enhanced MOEs, including advanced analytics such as time–space diagrams (TSDs). These capabilities enable more comprehensive traffic signal performance evaluation and support data-driven decision-making.

This chapter documents the installation and configuration framework for deploying UTAIM on Windows 10 or Windows 11 systems. The deployment environment includes supporting software components such as Python, XAMPP, Apache, and MySQL, which together provide a local server environment for database management and web-based visualization. Following installation, UTAIM operates on a local Apache–MySQL platform, enabling development, testing, user configuration, and generation of performance visualizations without reliance on external servers.

3.2. Prerequisites

Prior to installing and configuring UTAIM, the following system requirements were verified to ensure compatibility with the software environment and supporting tools:

- Operating System:
 - Windows 10 or Windows 11
- Code Editor:
 - Visual Studio Code (VS Code) is installed for code review and configuration tasks
- Python Environment:
 - Python version 3.10 or later installed to support UTAIM backend scripts and data processing utilities

These prerequisites establish the baseline environment required for the successful installation and execution of UTAIM components.

3.3. Python Installation

Python is required to support UTAIM backend processing and data parsing workflows. In this implementation, Python was installed using the official Windows installer.

- The Python installer package (*python-3.13.0-amd64.exe*) was used to deploy the Python environment.
- During installation, the option to add Python to the system PATH was enabled to allow command-line access from any directory, as shown in Fig. 3-1.

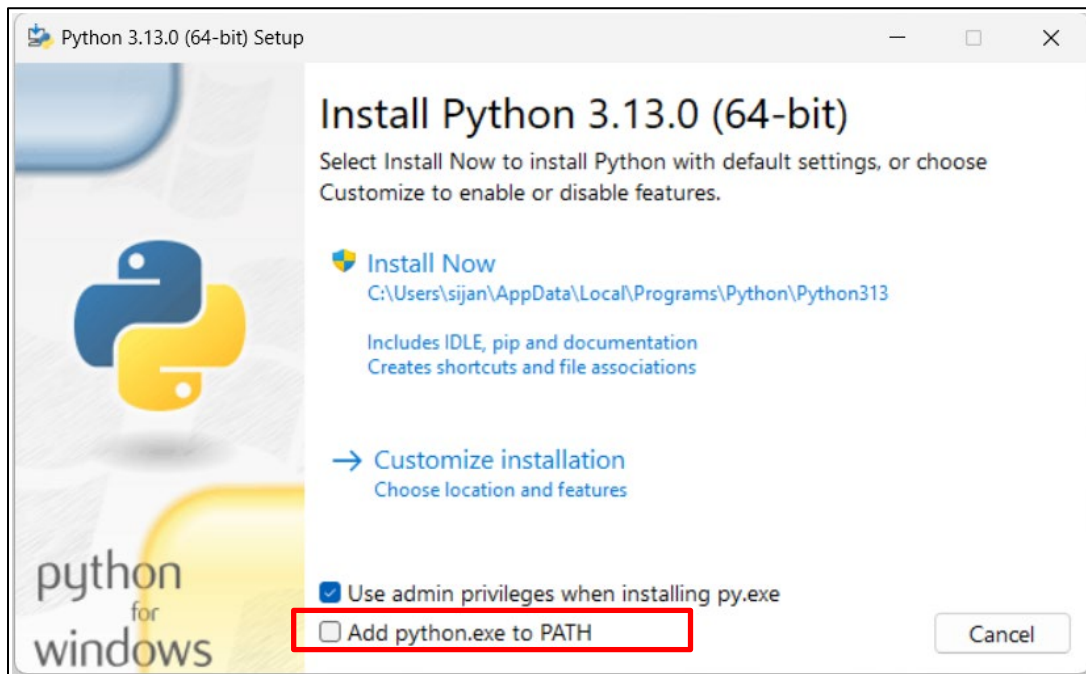


Figure 3-1: Python installer setup window initiating the installation process.

- The default installation configuration was selected to ensure compatibility with UTAIM scripts.
- After installation, the system was restarted to finalize environment variable updates.
- Successful installation was verified by checking the installed Python version from the command-line interface.

Completion of this step ensured that the Python runtime environment was properly configured and ready for use by UTAIM processing modules.

3.4. Installing and Configuring XAMPP and Apache Server

To support local deployment and testing of UTAIM, a local web server environment was established using XAMPP, an open-source software package that provides Apache and PHP components for local web hosting. This configuration enables UTAIM to operate on a local machine without reliance on an external server.

3.4.1. XAMPP Installation

- The XAMPP Windows (64-bit) installer was used to deploy the local server environment.
- Only the Apache and PHP components were retained, as these are required for hosting the UTAIM web interface.
- A custom installation directory (e.g., *D:\Xampp*) was selected instead of the default system drive to improve performance and reduce potential conflicts with antivirus software.
- The installation was completed using the default configuration settings.

3.4.2. Configuring XAMPP and Apache

Following installation, Apache server settings were updated to ensure compatibility with other services running on the system and to support the stable operation of UTAIM.

- The Apache server ports were reassigned to avoid conflicts with commonly used system ports:
 - Main Port: 8080
 - SSL Port: 8443
- Apache was configured to run as a Windows service, allowing the server to start automatically upon system reboot. This ensures the continuous availability of the UTAIM application without requiring manual intervention.

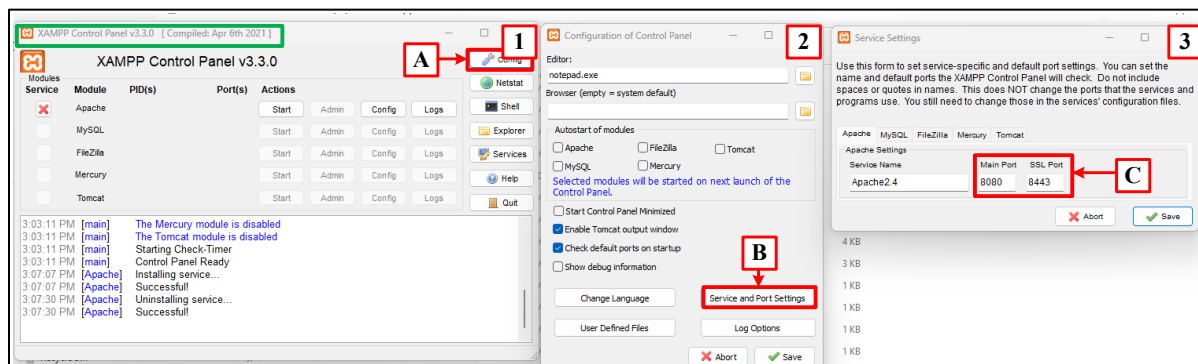


Figure 3-2: XAMPP Control Panel displaying Apache configuration with custom port settings.

3.5. Installing the MySQL Community Version

The MySQL Community Edition was installed to serve as the database backend for the UTAIM system. MySQL provides a lightweight, open-source database management platform that integrates effectively with local web servers such as Apache and supports scalable data storage for CV and ATSPM-related datasets.

3.5.1. MySQL Installation

- The MySQL Community Installer (64-bit) was used to deploy the database environment.
- MySQL Community Server 8.0.32 was required for this implementation.
- A full installation configuration was selected to ensure that all required MySQL components and supporting tools were available, as in Fig. 3-3.
- During the installation process, prerequisite checks were reviewed. Unmet prerequisites other than Python were determined not to affect UTAIM operation and were therefore not installed.

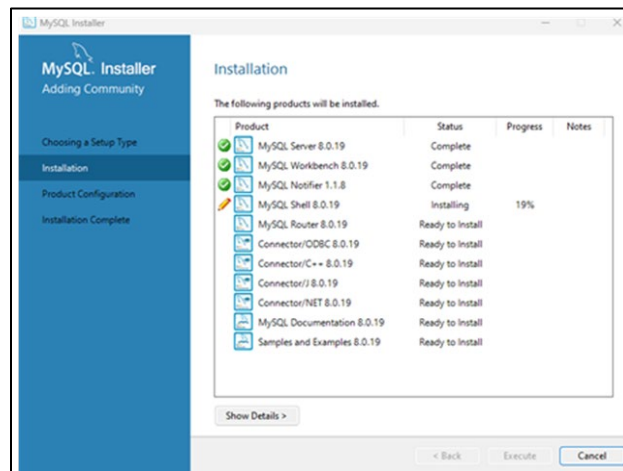


Figure 3-3: MySQL Installer community version setup screen during a full installation process.

3.5.2. Product Configuration

Following installation, the MySQL server was configured to ensure compatibility with UTAIM.

- The Legacy Authentication Method was selected to retain compatibility with MySQL 5.x-based authentication, which is required by the UTAIM application framework.
- A MySQL root database password was defined in accordance with the password requirements specified by the UTAIM system.

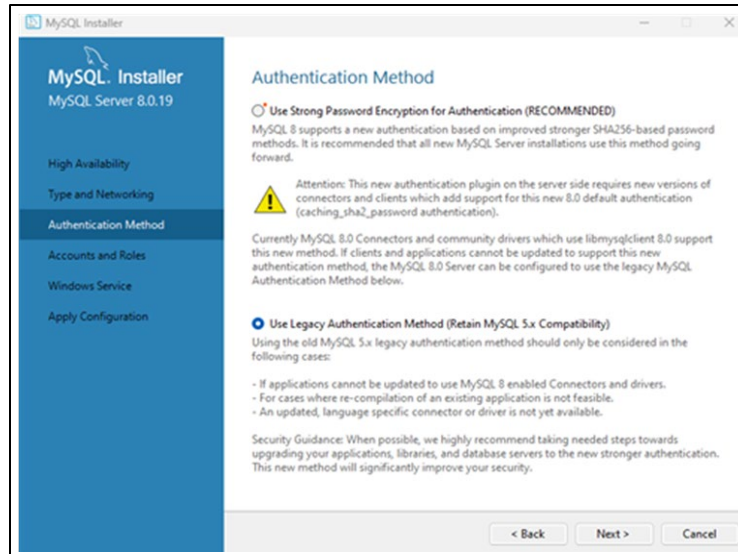


Figure 3-4: MySQL Installer community version password setup

3.6. Configuring the MySQL Database in Workbench

MySQL Workbench was used as the primary graphical interface for configuring and managing the MySQL databases required by UTAIM. Proper database configuration is essential to ensure integration between UTAIM applications and the MySQL backend.

- MySQL Workbench 8.0 Community Edition was launched and connected to the local MySQL server instance (MYSQL80) using the configured root credentials, as shown in Fig. 3-5.

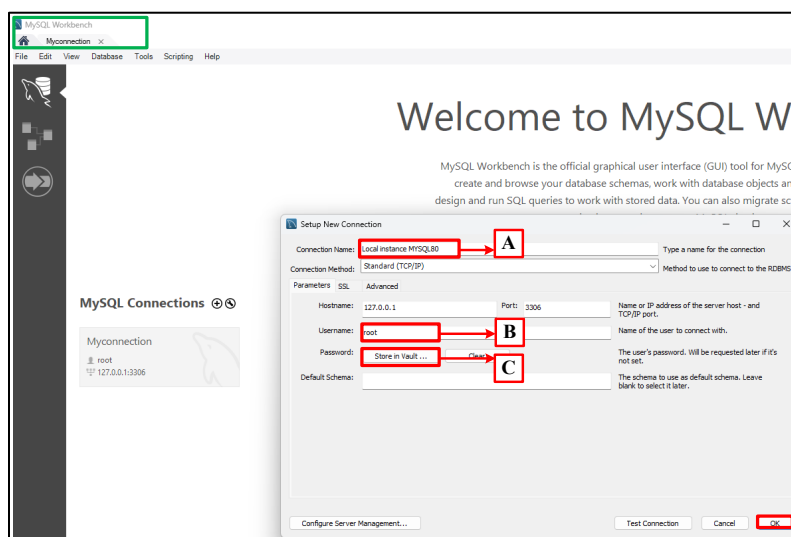


Figure 3-5: Setting up a new connection in MySQL Workbench

3.6.1. Database Schema Configuration

To support UTAIM operation, two dedicated database schemas were created using default character set and collation settings:

- utaim: Stores connected vehicle trajectories, processed data, and analysis outputs.
- utaim_config: Stores system configuration parameters and application-level settings.

After schema creation, the schema list was refreshed to confirm successful deployment, , as shown in Fig. 3-6.

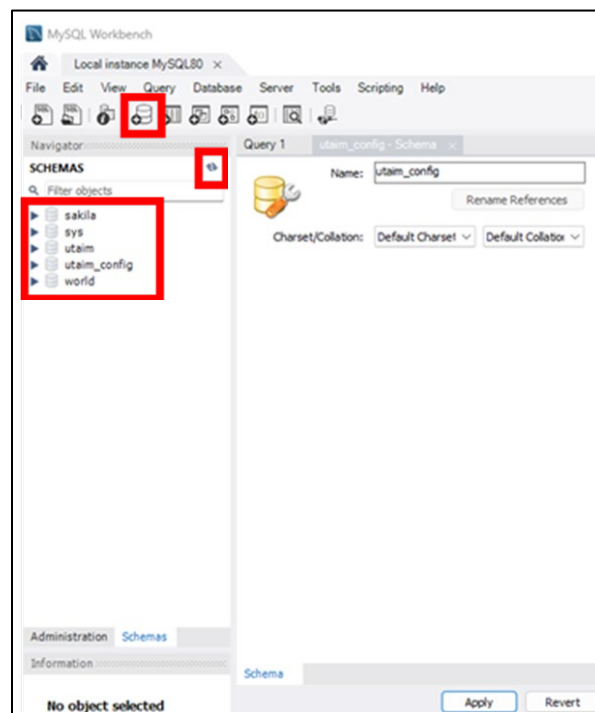


Figure 3-6: Setting up new schemas (utaim and utaim_config) for the database.

3.6.2. Data Import and Initialization

Following schema creation, the initial database content was populated using a provided backup dataset.

- The backup directory containing UTAIM database files was loaded through the MySQL Workbench data import utility.
- Data were imported into the utaim schema to initialize the database structure and populate required tables.

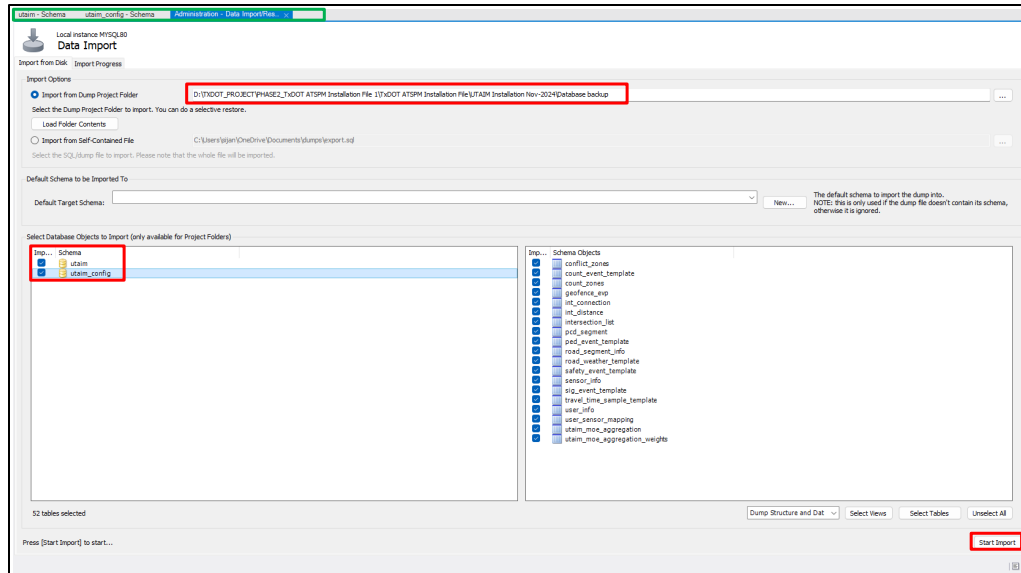


Figure 3-7: MySQL Workbench interface displaying the created schemas and imported data details.

3.7. Installing UTAIM on the Apache Server

To enable local deployment and testing, UTAIM was installed on the Apache web server provided through the XAMPP environment. This configuration allows UTAIM to be accessed locally through a web browser and connected to the MySQL backend configured in the previous sections.

- A dedicated utaim directory was created within the Apache document root (D:\Xampp\htdocs) to host the application files.
- Fig. 3-8 shows the complete set of UTAIM application files that were extracted and placed in the D:\Xampp\htdocs\utaim directory, making the application accessible through Apache.

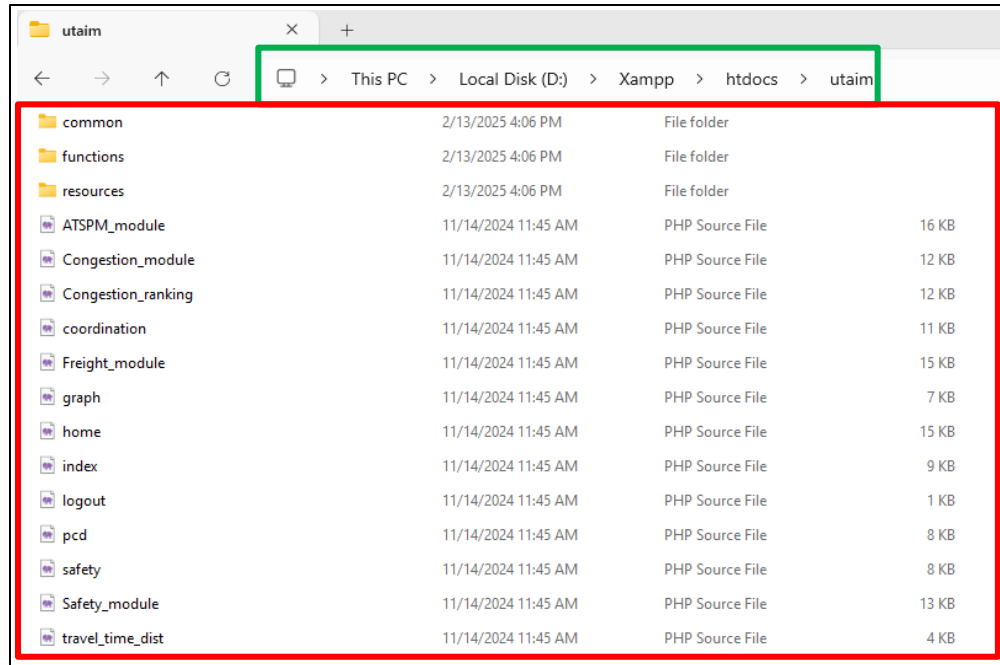


Figure 3-8: UTAIM application files deployed in the Apache htdocs directory.

Following deployment, the UTAIM web interface was accessed through the local Apache server using the configured port.

- The application was launched via the local URL: [**http://localhost:8080/utaim**](http://localhost:8080/utaim)
- Successful loading of the UTAIM login interface confirmed that the Apache server was correctly serving the application.

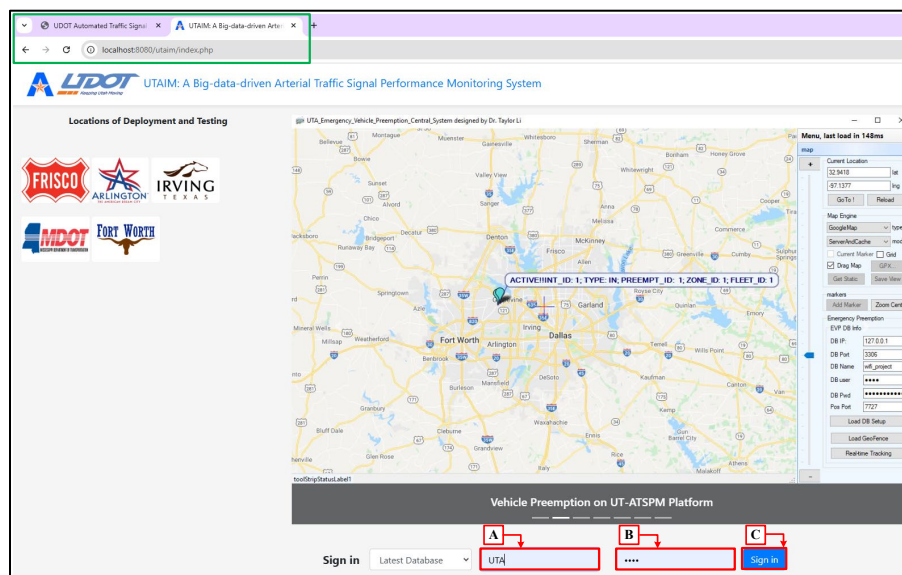


Figure 3-9: UTAIM main interface is displayed after a successful login using the provided credentials.

3.8. Adding Intersection to UTAIM

This section documents the configuration steps required to integrate intersections into the UTAIM system. The process involves defining users, associating sensors, configuring intersections and network relationships, and assigning traffic signal and roadway attributes to enable travel-time and performance analysis.

3.8.1. Adding a User

UTAIM user access is managed through a dedicated user information table within the MySQL database. Each user record defines authentication credentials used to access the system.

- A new user entry was added by inserting a record containing the username and password.
- After the record was added, the table was refreshed to confirm successful registration of the user.

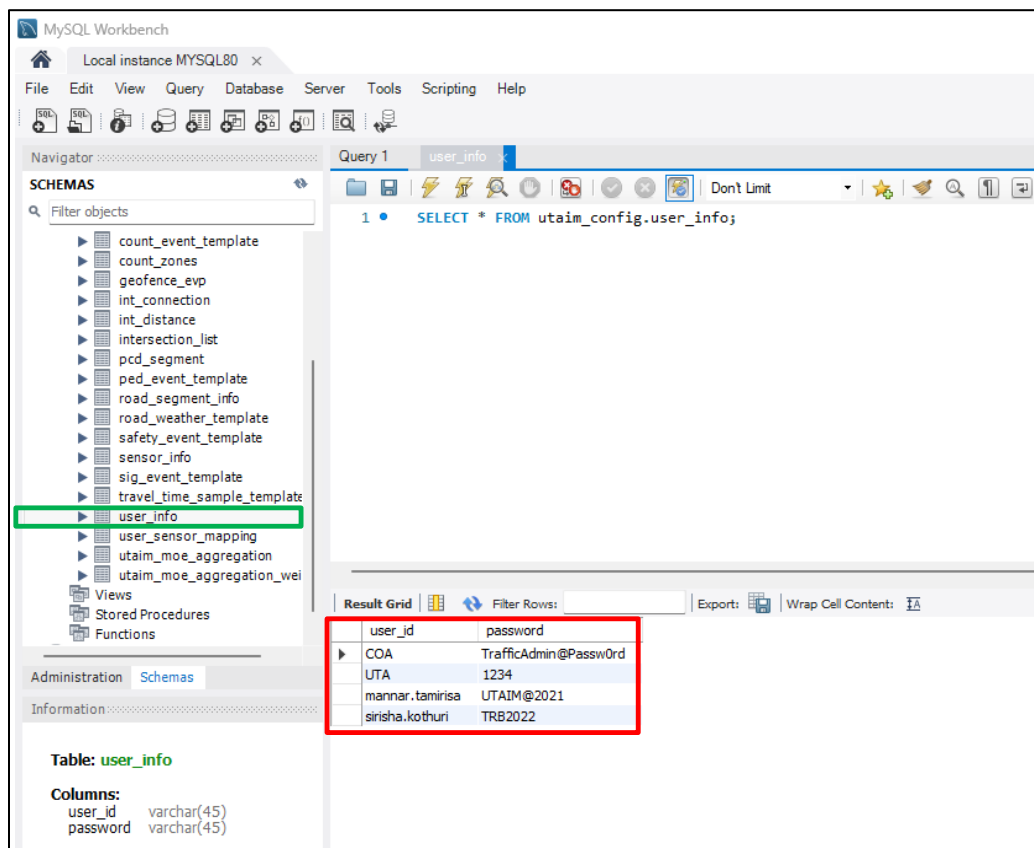


Figure 3-10: Adding a new user to the UTAIM system.

User records may be removed in the future by deleting the corresponding database row if access is no longer required.

3.8.2. Mapping Sensors to Users

Access to connected vehicle and travel-time data is controlled through the user–sensor mapping table, which links users to specific sensors.

- Required sensors were associated with the user account to enable real-time travel-time analysis.
- Existing legacy sensor entries were retained to preserve compatibility with potential future data requirements.

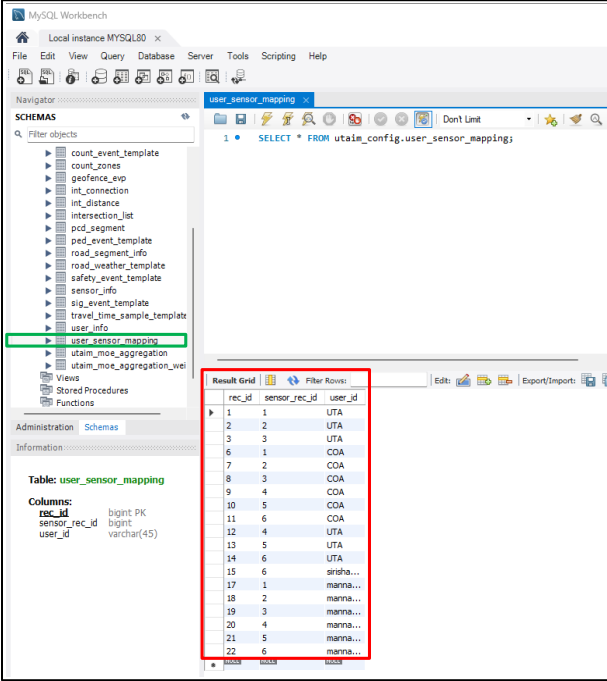


Table: user_sensor_mapping

Columns:

- rec_id: bigint PK
- sensor_rec_id: bigint
- user_id: varchar(45)

rec_id	sensor_rec_id	user_id
1	1	UTA
2	2	UTA
3	3	UTA
6	1	COA
7	2	COA
8	3	COA
9	4	COA
10	5	COA
11	6	COA
12	4	UTA
13	5	UTA
14	6	UTA
15	6	srisha...
17	1	manna...
18	2	manna...
19	3	manna...
20	4	manna...
21	5	manna...
22	6	manna...

Figure 3-11: Mapping sensors to users for real-time travel time analysis.

3.8.3. Adding Intersections

Intersections were registered in UTAIM using the intersection list table, which serves as the primary reference for network configuration.

- Each intersection was assigned a unique, sequential record ID, as shown in Fig. 3-12.
- A corresponding intersection identifier was defined.
- Geographic coordinates (latitude and longitude) were obtained from Google Maps and entered to support spatial visualization.
- Ownership attributes were specified to control user-level access to each intersection.

3.8.5. Configuring Intersection Connections

Directional relationships between intersections were defined using the intersection connection table.

- Each intersection ID was linked to upstream and downstream intersections.
- Northbound (NB) and southbound (SB) connections were specified explicitly.
- Missing connections were assigned a value of zero to indicate no linkage.

Table: int_connection

Columns:

- rec_id int PK
- int_id int
- direction varchar(4)
- downstream_int_id int
- downstream_int_direction varchar(4)

rec_id	int_id	direction	downstream_int_id	downstream_int_direction
1	1	NB	8	NB
2	1	SB	2	SB
3	2	NB	1	NB
4	2	SB	3	SB
5	3	NB	2	NB
6	3	SB	4	SB
7	4	NB	3	NB
8	4	SB	5	SB
9	5	NB	4	NB
10	5	SB	0	
11	6	NB	9	NB
12	6	SB	7	SB
13	7	NB	6	NB
14	7	SB	8	SB
15	8	NB	7	NB
16	8	SB	1	SB
17	9	NB	0	
18	9	SB	6	SB

Figure 3-14: Configuring intersection connections to define directional relationships.

3.8.6. Setting Intersection Distances

Intersection spacing was defined using the intersection distance table, which supports free-flow travel-time calculations.

- Distances were measured using either:
 - The VISSIM simulation model, or
 - Google Maps distance measurement tools
- Measured distances were recorded in feet and matched to the corresponding intersection IDs.

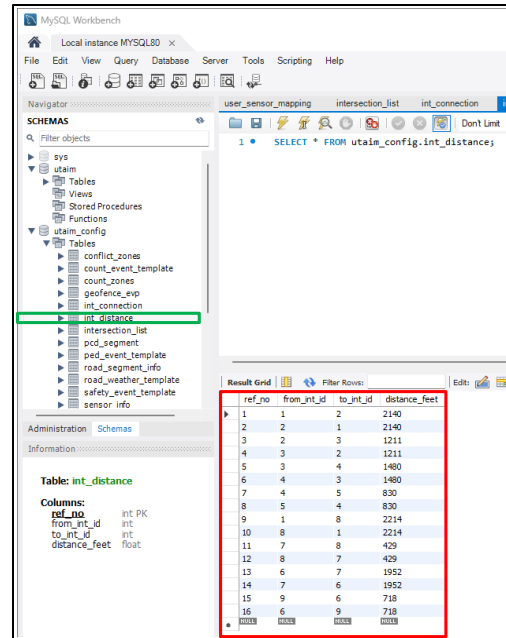


Figure 3-15: Setting intersection distances for accurate travel time calculations.

3.8.7. Configuring Traffic Signal Phases

Traffic signal phase configuration links signal movements with detectors and approach attributes.

- The total number of phases for each intersection was defined.
- Phase directions (e.g., northbound, westbound) were assigned.
- Detector channels were mapped using a semicolon-separated format (e.g., 23;24;25 for advance detectors).

Detector channel formatting must follow the specified convention; otherwise, phase detection will fail.

- Time-to-Intersection (TTI) values were calculated using detector-to-stop-line distances and posted speed limits.
- Additional detector types (red-light, presence, and queue detectors) were configured as applicable.
- The Wi-Fi link ID was set to 99999 to retain compatibility with legacy system logic as illustrated in Fig. 3-16.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema for 'utalm_config', with 'pcd_segment' selected. The main window shows the table's structure and a list of 20 records. The table has 15 columns: rec_id, int_id, phase_id, direction, advance_dets, tti_sec, red_light_dets, presence_dets, queue_dets, wf-link_id, lane_no, capacity_vphpl, reference_link, and ffs_mph. The records represent different traffic signal phases and their associated detectors.

rec_id	int_id	phase_id	direction	advance_dets	tti_sec	red_light_dets	presence_dets	queue_dets	wf-link_id	lane_no	capacity_vphpl	reference_link	ffs_mph
1	1	2	NB	7;8;9	2	0	3;4;5	0	12	3	1200	2	41
2	1	6	SB	23;24;25	2	0	19;20;21	0	11	3	1200	1	41
3	2	2	NB	0	0	0	2	0	12	3	1200	4	41
4	2	6	SB	0	0	0	4	0	11	3	1200	3	41
5	3	2	NB	0	0	0	2	0	12	3	1200	6	41
6	3	6	SB	0	0	0	6	0	11	3	1200	5	41
7	4	2	NB	23;24;25	2	0	19;20;21	0	12	3	1200	8	41
8	4	6	SB	7;8;9	2	0	3;4;5	0	11	3	1200	7	41
9	5	2	NB	0	0	0	1	0	12	3	1200	10	41
10	5	6	SB	0	0	0	2	0	11	3	1200	9	41
11	1	4	WB	39;40	2	0	35;36	0	99999	2	1200	999	41
12	1	8	EB	55;56	2	0	51;52	0	99999	2	1200	999	41
13	4	4	WB	55;56;57	2	0	51;52;53	0	99999	3	1200	999	41
14	4	8	EB	35;36;37	2	0	35;36;37	0	99999	3	1200	999	41
15	6	1	SBL	0	0	0	17	0	99999	1	1200	999	41
16	6	2	NB	7;8;9	2	0	2;3;4	0	99999	3	1200	999	41
17	6	3	WBL	0	0	0	49	0	99999	1	1200	999	41
18	6	4	EB	0	0	0	35;39	0	99999	2	1200	999	41
19	6	5	NBL	0	0	0	1	0	99999	1	1200	999	41
20	6	6	SB	23;24;25	2	0	19;20;21	0	99999	3	1200	999	41

Figure 3-16: Configuring traffic signal phases with assigned detectors.

3.8.8. Assigning Lane and Speed Information

Roadway characteristics were assigned to each approach to support performance analysis.

- The number of lanes was determined using:
 - Field observations, or
 - Satellite imagery from Google Maps.
- Vehicle capacity values were defined based on local traffic conditions.
- Reference link identifiers were assigned.
- Free-flow speeds were entered using posted speed limits

link_id	from_sensor_ID	to_sensor_ID	one_way_flag	speed_limit_mph	distance_feet	travel_time	lane_no	capacity_phpl
1	1	5	0	40	5700	97	3	1500
11	1001	1002	0	40	5700	97	3	1500
12	1003	1004	0	40	5700	97	3	1500
13	1005	1006	0	70	2155	100	4	1800
14	1007	1008	0	70	2200	100	4	1800
2	5	1	0	40	5700	97	3	1500
99999	1	5	0	40	1	100	2	1500

Figure 3-17: Assigning Lane numbers and speed limits for intersections.

3.9. Generating Graphs in UTAIM

The final stage of UTAIM configuration involves generating visual analytics to evaluate traffic signal operations and corridor-level performance. Two primary visualizations are supported: the Purdue Coordination Diagram (PCD) and the Coordination Time–Space Diagram (TSD).

3.9.1. Purdue Coordination Diagram (PCD)

The PCD is a core visualization for assessing intersection-level performance by depicting the relationship between vehicle arrivals, signal phase timing, and detector activity. It supports the evaluation of signal coordination quality and identification of operational issues.

- The PCD visualizes vehicle arrivals relative to signal phases, enabling assessment of arrivals on green (AoG) performance.
- Detector actuations are shown as discrete points, providing insight into demand patterns.
- The diagram incorporates multimodal and operational elements, including:
 - arrival profiles and signal cycles;
 - AoG percentage and green time (GT);
 - pedestrian calls, preemption, and priority events; and
 - weather summaries, when available, represented by a color bar indicating precipitation conditions.

An example PCD for intersection-level performance analysis is shown in Fig. 3-18.

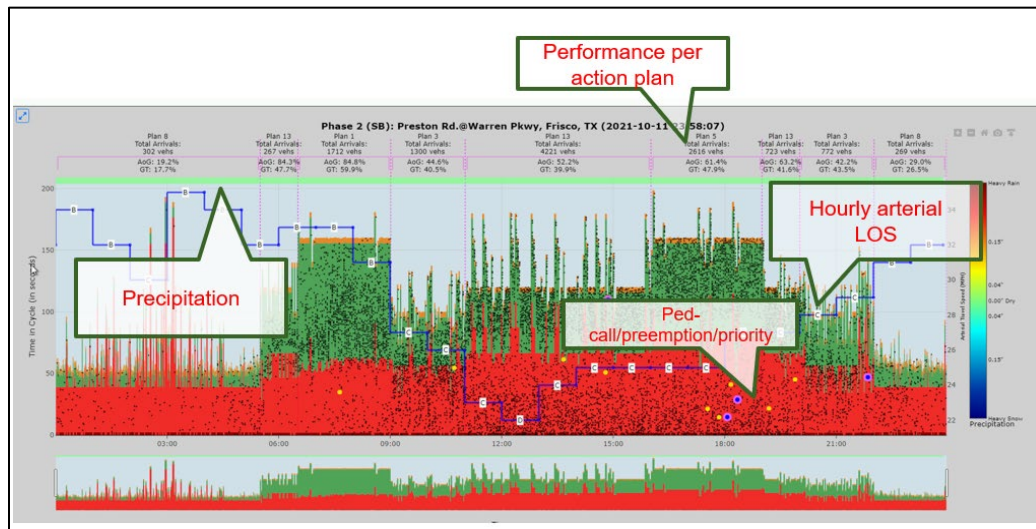


Figure 3-18: PCD for analyzing intersection performance.

The interactive interface allows zooming for detailed inspection across cycles and time periods.

Note: PCDs can only be generated for dates corresponding to the simulation run. Additionally, the parsed dataset must also cover the full target analysis period.

3.9.2. Coordination (Time–Space) Diagram

The TSD is a distinctive output of UTAIM that illustrates vehicle trajectories across multiple intersections, supporting corridor-level analysis of progression and coordination.

- The TSD displays sampled vehicle trajectories traveling between adjacent intersections, revealing stop-and-go behavior and progression quality.
- Special operational events, including rail and emergency vehicle preemptions, pedestrian calls, and freight signal priority (Prioritor) events, are explicitly reflected in the trajectory patterns.
- The visualization helps identify:
 - locations with recurring vehicle stoppages;
 - impacts of preemption and priority events on downstream flow; and
 - opportunities for signal-timing and coordination optimization.

An example TSD for the northbound Cooper Street corridor is shown in Fig. 3-19.

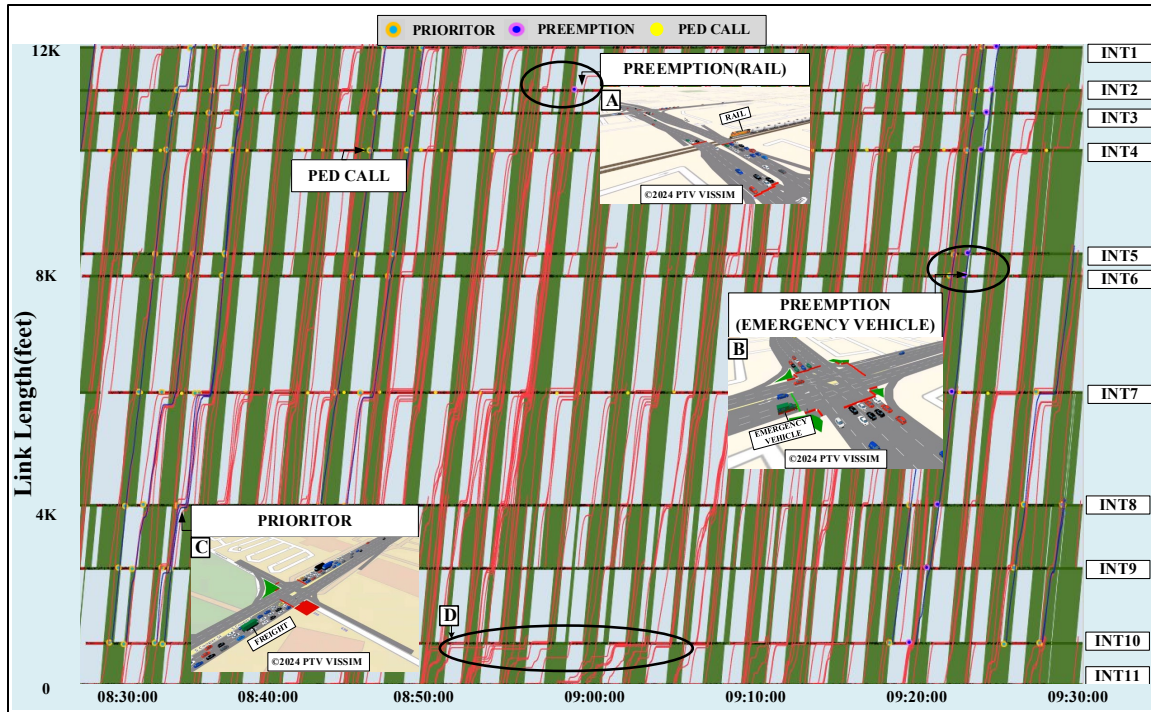


Figure 3-19: TSD for Northbound Cooper Street.

For computational efficiency and interpretability, the analysis is typically limited to a one-hour time window, and only a subset of vehicle trajectories (approximately 10%) is used to generate the diagram.

Note: Generating TSDs for periods longer than one hour is not advised due to performance limitations and visual congestion.

Chapter 4: TxDOT ATSPM Digital Twin Development

4.0. Introduction

The TxDOT ATSPM Digital Twin is a packaged deployment framework designed to enable rapid implementation of multiple ATSPM platforms. Once deployed, the system allows users to either load preconfigured ATSPM databases or generate new ATSPM datasets through traffic simulation. This chapter builds upon the installation and configuration procedures presented in Chapters 2 and 3 and documents the corridor-scale deployment and operational use of the digital twin.

The primary focus of this chapter is the implementation of UDOT ATSPM and UTAIM for the Ranch-to-Market Road 2244 (RM 2244) corridor in Austin, Texas. The study corridor includes 23 signalized intersections and provides the simulation network used to demonstrate the controller-integration and ATSPM data-generation workflows developed under Project 5-7160.

Three traffic signal control configurations were established in PTV VISSIM:

- The built-in RBC was used to represent signal operations directly within the VISSIM environment.
- The Econolite EOS controller emulator was integrated through SILS to reproduce external controller logic and generate high-resolution signal event data.
- A physical Q-Free MAXTIME controller was connected to VISSIM through the TxDOT HILS-CID to support hardware-in-the-loop simulation and controller-based ATSPM event generation.

These configurations represent different levels of controller integration. The RBC provides a fully simulation-based environment for configuring and testing signal operations. The Econolite EOS SILS configuration incorporates external controller logic and supports functions such as coordination, detector processing, and preemption. The HILS-CID configuration extends the framework by allowing a physical Q-Free MAXTIME controller to exchange signal indications, detector actuations, and pedestrian calls with VISSIM in real time. The application and comparison of these configurations within the RM 2244 corridor case study are presented in Chapter 6.

In addition to controller integration, the digital twin uses a Docker-based architecture to streamline deployment of UDOT ATSPM and UTAIM. This approach reduces system setup time while improving portability, reproducibility, and scalability. The framework provides a structured workflow encompassing traffic-demand preparation, simulation development, controller integration, ATSPM-compliant data generation, database deployment, and initial performance visualization using PCDs and TSDs.

The objectives of the TxDOT ATSPM Digital Twin are threefold:

1. Provide a realistic training and exploration platform
2. Deliver an up-to-date TxDOT traffic signal inventory
3. Facilitate rapid deployment for smaller agencies

This chapter is organized into three primary components:

1. System Deployment with Docker – Description of the Docker-based deployment strategy for the TxDOT ATSPM Digital Twin.
2. Traffic Simulation with VISSIM – Configuration of corridor-level traffic simulation models using multiple signal control strategies to generate high-resolution ATSPM data.
3. Data Parsing and Integration – Procedures for converting simulated traffic signal and vehicle trajectory data into formats compatible with operational ATSPM systems.

4.1. Deploying the TxDOT ATSPM Docker Image in Windows

Manual installation of UDOT ATSPM and UTAIM can be time-consuming and sensitive to differences in host-system hardware and software. To address these limitations, the project team at UTA adopted a prebuilt Docker-based deployment. The Docker image encapsulates all required system components, including web services, databases, backend processing modules, and configuration settings, into a standardized runtime environment. This eliminates dependency conflicts and minimizes host-system compatibility issues.

Using the Docker image reduces deployment time from hours to minutes and ensures consistent system behavior across different user environments. Once deployed, both UDOT ATSPM and UTAIM are immediately operational without repeating the manual installation and configuration procedures described in Chapters 2 and 3, providing a reliable and scalable solution for corridor-level ATSPM digital twin deployment.

4.1.1. What is Docker and Its Benefits to This Project

Docker is an open-source containerization platform that enables applications and their dependencies to be packaged into lightweight, portable containers. These containers run consistently across different computing environments, allowing software to be deployed, managed, and executed without dependency conflicts or system-specific configuration issues.

Docker is built on several core components:

- Docker Engine, which manages container execution and resources.

- Docker Images, which contain the application code, dependencies, and runtime configuration.
- Docker Containers, which are executable instances of Docker images.
- Docker Hub, a centralized repository for storing and distributing images.
- Docker Compose, which supports defining and running multi-container applications.

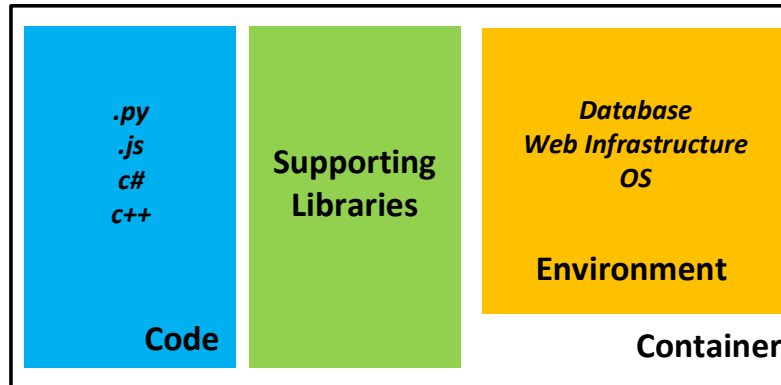


Figure 4-1: Example of a Docker Container

For this project, Docker provides a standardized and reliable deployment mechanism for the TxDOT ATSPM Digital Twin. By encapsulating the UDOT SPM v4.3 and UTAIM systems within containers, Docker eliminates installation issues caused by missing libraries, incompatible software versions, or operating system differences. This approach ensures consistent system behavior across users and significantly reduces deployment complexity.

The Docker-based digital twin architecture used in this project is illustrated in Fig. 4-2. In this framework, the PTV VISSIM simulation communicates with the UDOT SPM v4.3 Docker container through open network ports. Simulated high-resolution traffic signal events are first stored in the SQL Server database within the UDOT SPM container under the VISSIM schema. The parsing software then reprocesses these events and stores them under the MOE schema. Subsequently, the parsed data are transferred to the UTAIM Docker container, where they are saved in the UTAIM schema and converted into signal event tables.

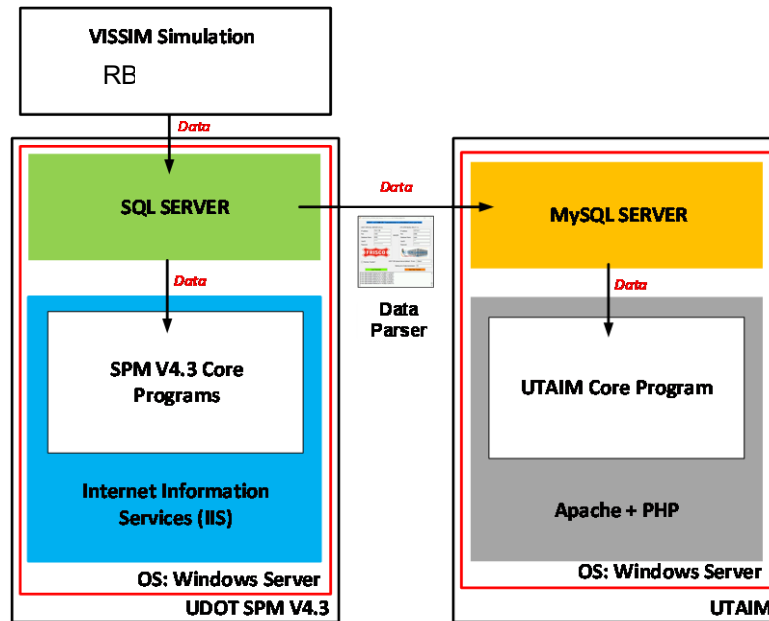


Figure 4-2: Proposed digital twin with Docker containers

After this data exchange process, both the UDOT SPM and UTAIM systems contain valid, synchronized signal event datasets, enabling the generation of a wide range of Signal Performance Measures (SPMs) within the digital twin environment.

4.1.2. Required Hardware, Software, and Libraries

Deployment of the TxDOT ATSPM Digital Twin using Docker requires a compatible Windows environment with containerization support enabled.

- Operating System Requirements
 - 64-bit Windows Pro, Enterprise, or Education editions are required to support Windows containers.
 - System-version information can be verified under Settings > System > About > Windows specifications.

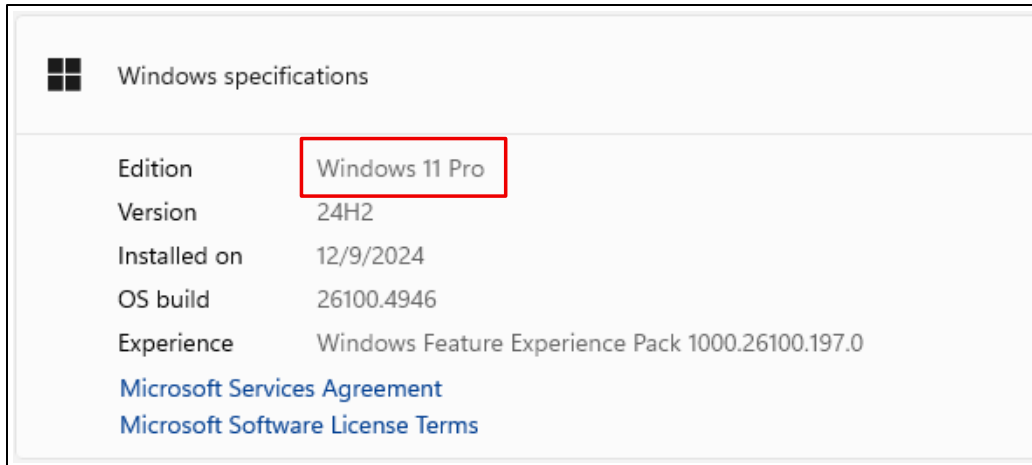


Figure 4-3: Windows version specifications (Windows 11 Pro shown in the figure)

- Windows Containerization Features
 - Both Hyper-V and Windows Containers must be enabled to support Docker-based deployment.
 - These features allow Docker Desktop to run Windows containers and manage virtualized resources.
 - The following commands must be executed in PowerShell with administrator privileges:
- Enable-WindowsOptionalFeature -Online -FeatureName Containers -All
- Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All

If prompted to restart after executing each command, enter n to postpone the restart until Docker Desktop has been installed, as shown in Fig. 4-4.

```
PS C:\WINDOWS\system32> Enable-WindowsOptionalFeature -Online -FeatureName Containers -All
Do you want to restart the computer to complete this operation now?
[Y] Yes [N] No [?] Help (default is "Y"): n

Path      :
Online    : True
RestartNeeded : True

PS C:\WINDOWS\system32> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
Do you want to restart the computer to complete this operation now?
[Y] Yes [N] No [?] Help (default is "Y"): n

Path      :
Online    : True
RestartNeeded : True
```

Figure 4-4: Successfully Enabled Hyper-V and Container (input n for refusing restart)

- Docker Desktop Installation
 - Docker Desktop for Windows must be installed with Windows Containers enabled during installation.
 - During installation, select “Allow Windows Containers to be used with this installation,” as shown in Fig. 4-5.
 - After installation, a system restart is required to activate Docker services.
 - On first launch, acceptance of the Docker Subscription Service Agreement is required as in Fig. 4-5.

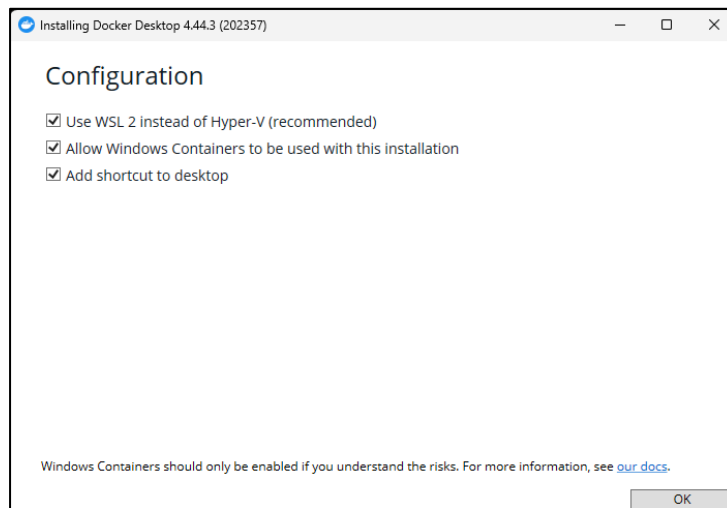


Figure 4-5: Enable Windows containers when installing Docker Desktop

- Windows Subsystem for Linux (WSL)
 - If WSL is not already installed, Docker Desktop prompts for installation during initial startup.
 - WSL is required for Docker Desktop operation, even when running Windows containers.



Figure 4-6: Installing WSL (Windows Subsystem for Linux)

- Switching to Windows Containers
 - Docker Desktop must be explicitly switched to Windows Containers mode.
 - The Privileged Helper Service must be enabled to allow container networking and system-level operations, as shown in Fig. 4-7.

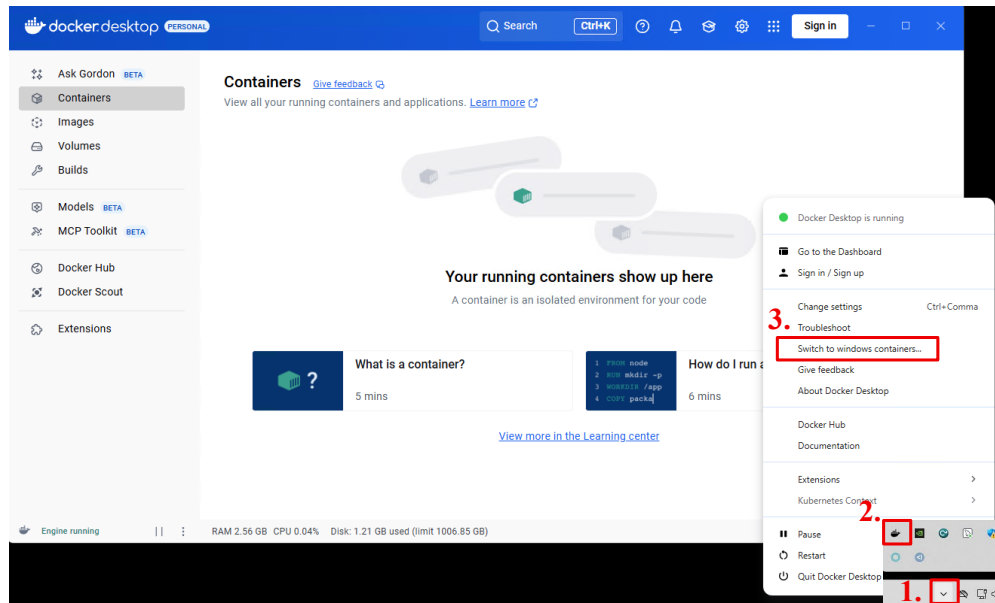


Figure 4-7: Switch to Windows Container in Docker Desktop

Once these requirements are satisfied, the system environment is prepared for deployment of the UDOT SPM v4.3 and UTAIM Docker containers.

4.1.3. Deploying Process and Success Verification

The extracted package includes the following components:

- `udot-app-v4.3.tar` and `utaim-app.tar`, which contain the prebuilt Docker images for the UDOT SPM and UTAIM systems, respectively. These images are loaded directly into Docker Desktop without extraction.
- `Database_backup/SQLSERVER` and `Database_backup/UTAIM`, which are bind-mounted directories used to store SQL Server and MySQL database files. These folders enable persistent data storage and support database backup and restoration.
- `SPM Databackup`, which contains example signal event datasets generated from VISSIM using both RBC and Econolite EOS controllers. These datasets can be restored to demonstrate SPM functionality.
- `UDOT_SPM_Images`, which stores performance diagrams generated by the UDOT SPM system in PNG format.

- docker-compose.yml, which defines container services, port mappings, and bind-mounted volumes.
- Supporting documentation and installation utilities.

TxDOT5-7160 Docker System Setup

1. Unzip the folder: *TxDOT5-7160 Docker Files*.
2. Open the folder in terminal:
 - Right-click → Open in Terminal, OR
 - Type cmd in the File Explorer address bar.
3. Verify Docker installation using: `docker-compose—version`
4. Confirm that the terminal displays the installed Docker Compose version, such as:
Docker Compose version v2.38.2-desktop.1
5. Load the Docker images using:
`docker load -i udot-app-v4.3.tar`
`docker load -i utaim-app.tar`

After loading, the terminal should show as in Fig. 4-8: Loaded image: txdotv43-udot-app:latest

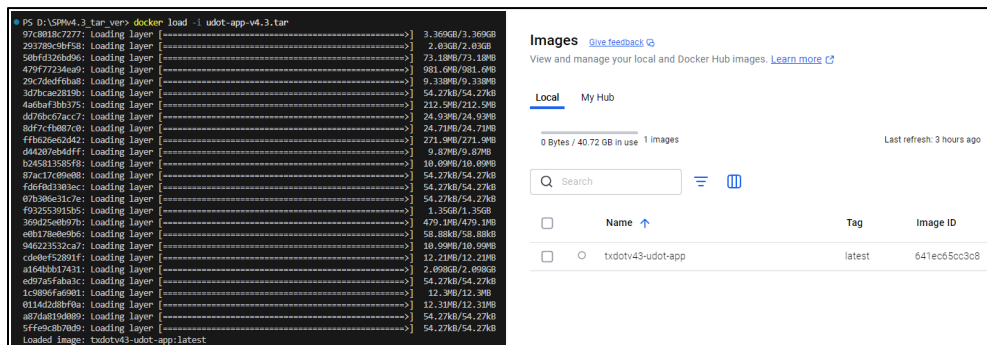


Figure 4-8: Successful loading of udot spm docker tar file

After loading the UDOT SPM and UTAIM Docker images, the system is initialized using the docker-compose configuration. The docker-compose file exposes required services, including web interfaces and database connections, through predefined host ports. If port conflicts exist on the host machine, port mappings can be modified directly in the docker-compose file prior to deployment.

```
D:\TxDOT5-7160 docker files>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
09695f6deb2d	txdotv43-utaim-app	"powershell -Command..."	6 minutes ago	Up 6 minutes	0.0.0.0:82->80/tcp, 0.0.0.0:8446->443/tcp, 0.0.0.0:3308->3306/tcp
517c76e9a148	txdotv43-udot-app	"cmd /S /C powershell..."	6 minutes ago	Up 6 minutes	0.0.0.0:81->81/tcp, 0.0.0.0:8443->443/tcp, 0.0.0.0:1434->1433/tcp

Figure 4-9: Successfully built Docker status

Successful deployment is verified as in Fig. 4-9 by confirming that all containers are running and accessible. When the system is operating correctly:

- The UDOT ATSPM web interface is accessible through the assigned host port.
- The UTAIM web interface loads successfully in a browser.
- Docker container status confirms active and healthy services.

General Application Settings

Raw Data Count Limit
1048576

Image Path
C:\ATSPMroot\SPMImages\

Image URL
http://localhost:81/spmimages/

ReCaptcha Public Key

ReCaptcha Secret Key

Seconds After Chart End Date to Complete Cycle

Save

Figure 4-10: Config Image Path and Image URL in the UDOT system

Stop and Rebuild Docker Container

1. Open the terminal in the project folder.
2. Stop the container: `docker-compose down`
3. Before rebuilding, delete all files inside: TxDOT5-7160 docker files / Database_backup / SQLSERVER (Prevents old MOE database from interfering with UDOT SPM initialization.)
4. Rebuild the container: `docker-compose up -d`

This command recreates the containers using the previously loaded images referenced in the `docker-compose.yml` file. It does not reload the `.tar` files or rebuild the Docker images.

4.1.4. Demonstrating Operations

After successful deployment, the operational status of the TxDOT ATSPM Digital Twin can be verified through Docker Desktop, where container status, exposed ports, and control actions (start, stop, restart) are displayed as in Fig. 4-11.

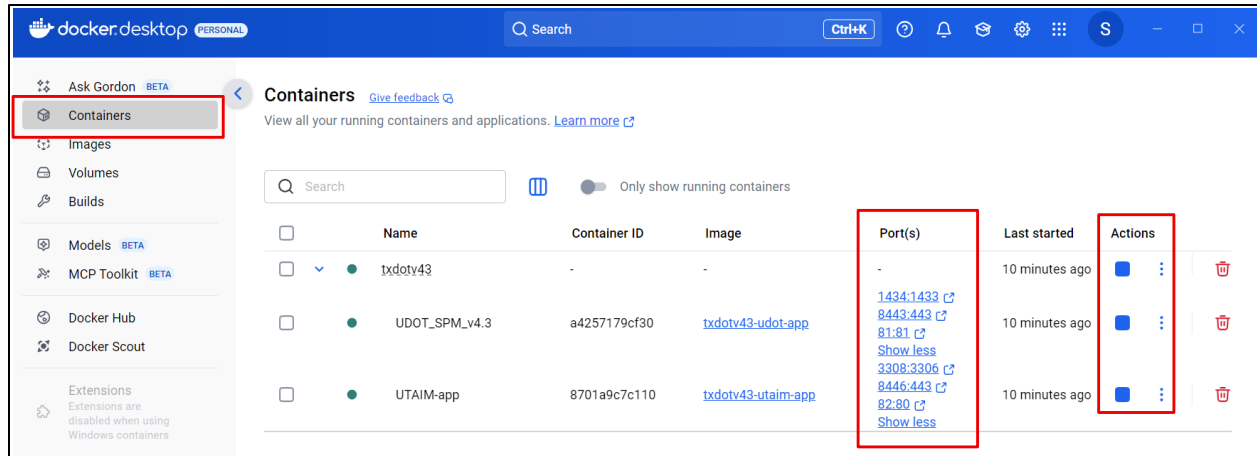


Figure 4-11: Running Docker Containers in the Docker Desktop

Database Connectivity Verification

Operational verification includes confirming database connectivity for both ATSPM platforms.

UDOT SPM – SQL Server (SSMS Required)

1. Open SQL Server Management Studio (SSMS).
2. Server Type: Database Engine
3. Server Name: 127.0.0.1,1434
4. Authentication: SQL Server Authentication
 - Login: sa
 - Password: Engineering123!
5. Connection Security:
 - Encryption: Optional
 - Check the Trust server certificate

After connecting, you should see the MOE database.

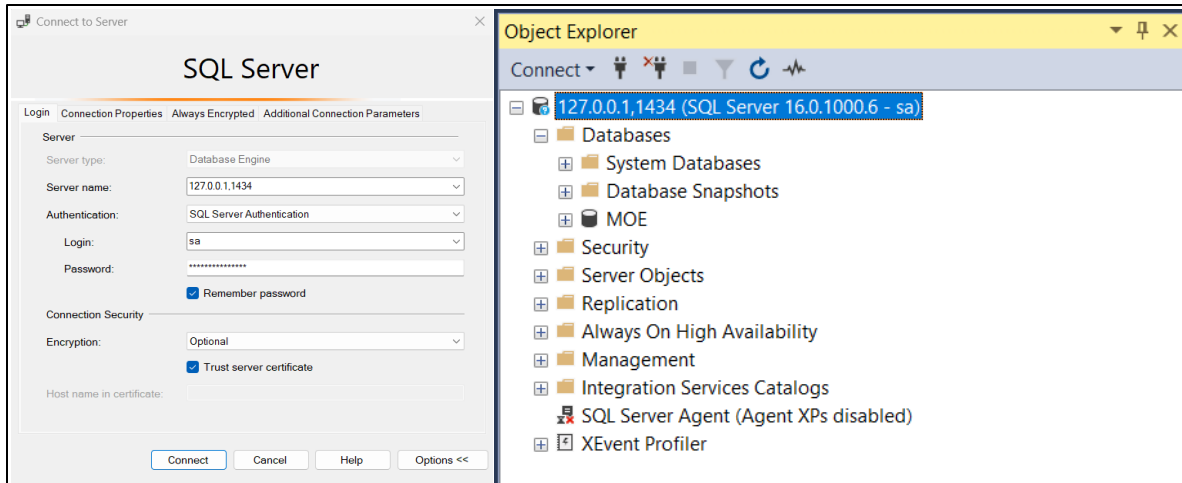


Figure 4-12: Login SQL Server of UDOT SPM

UTAIM SPM – MySQL (MySQL Workbench Required)

1. Open MySQL Workbench → Click “+” (New Connection).
2. Configure:
 - Connection Name: Docker Mysql
 - Port: 3308 (must match Docker port)
 - Username: root
3. Click “Store in Vault...”
 - Password: ENGINEERING
4. Click Test Connection → Should return: Successfully made the MySQL connection
5. Click OK, then double-click Docker Mysql to connect.

After login, you should see:

- utaim
- utaim_config

under the SCHEMAS tab

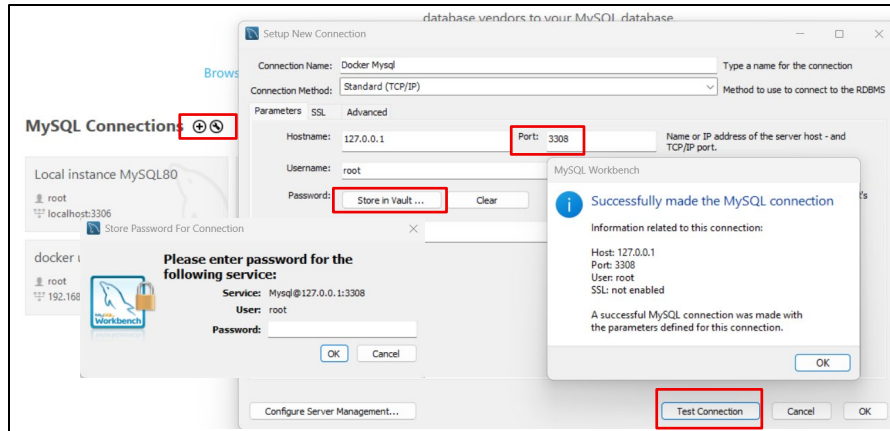


Figure 4-13: Login MySQL of UTAIM SPM

Data Backup and Restoration – UDOT SPM

Note: The MOE database is created only after first visiting: <http://localhost:81/ATSPM>

Therefore, data must be restored manually.

Restore EOS Data

1. Copy MOE_eos.bak from:

SPM Databackup / EOS data / UDOT data

→ Paste into:

Database_backup / SQLSERVER

(Do not modify existing files.)

2. Open SQL Server Management Studio (SSMS)

Login:

- User: sa
- Password: Engineering123!

3. Click New Query and run:

```
RESTOREDATABASE [MOE]
FROM DISK ='C:\sql-data\MOE_eos.bak'
WITH REPLACE;
```



```
USE [MOE];
CREATE USER [SPM] FOR LOGIN [SPM];
ALTER ROLE [db_owner] ADD MEMBER [SPM];
```

Click Execute and wait for the completion message as in Fig. 4-14.

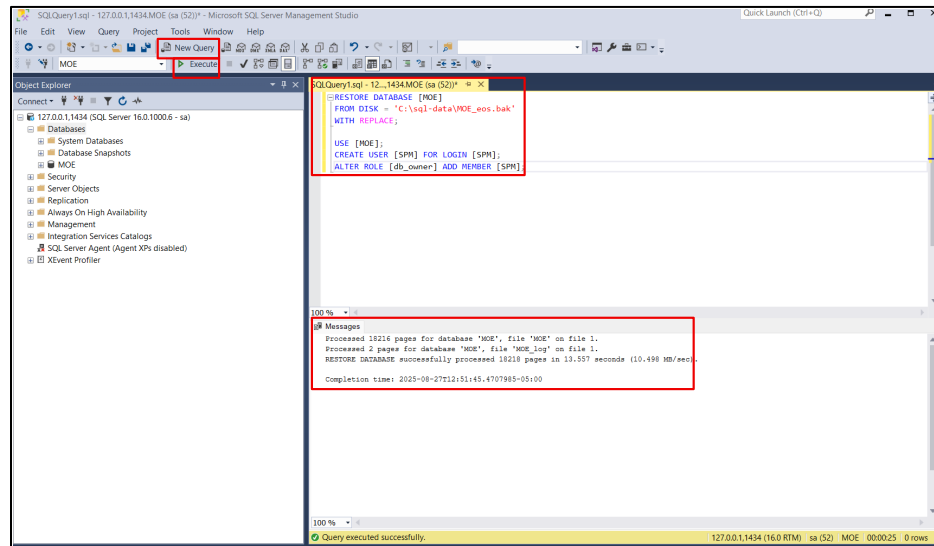


Figure 4-14: Restore Data in SQL SERVER

Backing Up the MOE Database

1. In SSMS, right-click MOE → Tasks → Back Up...
2. In Back Up Database – MOE:
 - General tab: set backup path to C:\sql-data
 - (Use Add... to change location/name, and remove the old path.)
 - Media Options: check Verify backup when finished
 - Click OK to start backup.

The backup file will be saved under: TxDOT5-7160 docker files / Database / SQLSERVER (see Fig. 4-15).

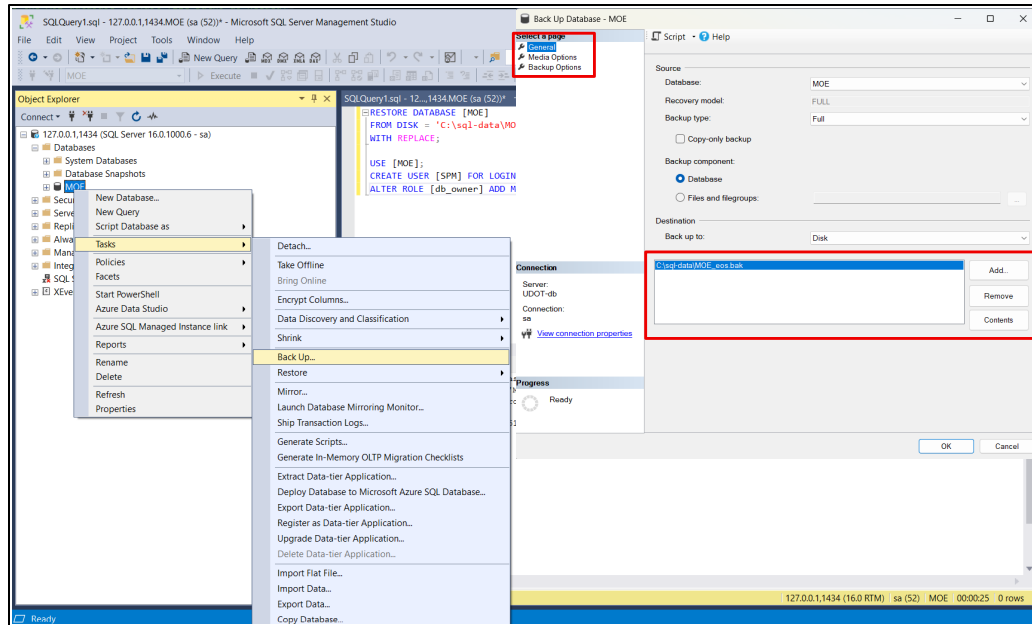


Figure 4-15: Backup Data in SQL SERVER

Data Backup and Restoration – UTAIM

Unlike UDOT, the UTAIM Docker is delivered with EOS data preloaded. Backup/restore steps are provided for future use.

Back Up UTAIM Database

1. Open MySQL Workbench and log in.
2. Go to Server → Data Export:
 - Select schemas: utaim and utaim_config
 - Ensure Dump Structure and Data is checked
 - Choose the backup file save location (MySQL allows saving directly to your computer.)
3. Click Start Export and wait for completion.

Backup dump file will be created at the selected location as shown in Fig. 4-16.

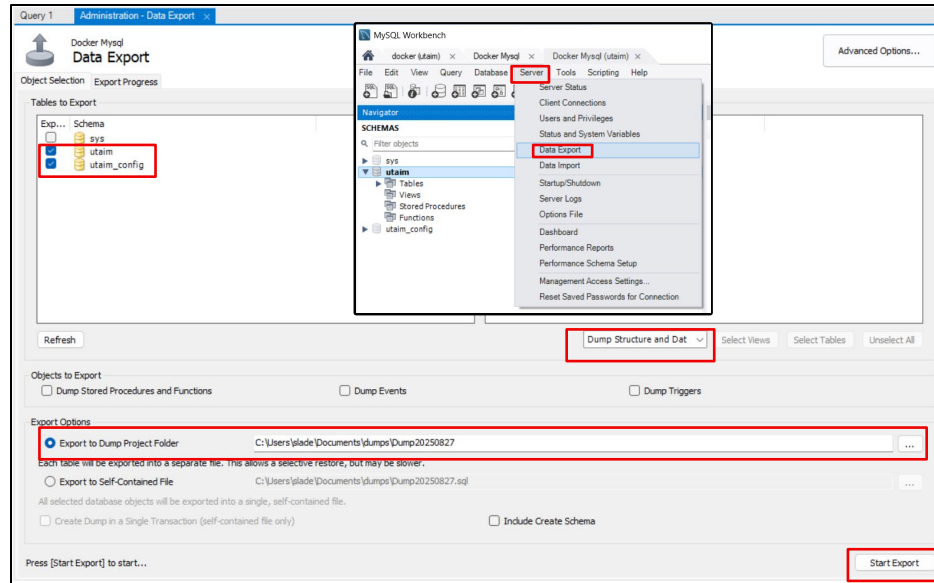


Figure 4-16: Backup Data in MySQL Workbench

Restoring the UTAIM Database

1. In MySQL Workbench, first delete existing schemas:
 - Right-click utaim → Drop Schema
 - Right-click utaim_config → Drop Schema

Then recreate them:

- Right-click in Schemas panel → Create Schema
 - Name: utaim → Click Apply
 - Repeat for utaim_config
2. Go to Server → Data Import:
 - Select Import from Dump Project Folder
 - Choose the backup folder
 - Select schemas: utaim and utaim_config
 - Ensure Dump Structure and Data
 - Click Start Import

After the restoration is complete, the imported tables and records should appear under the corresponding schemas, as shown in Fig. 4-17.

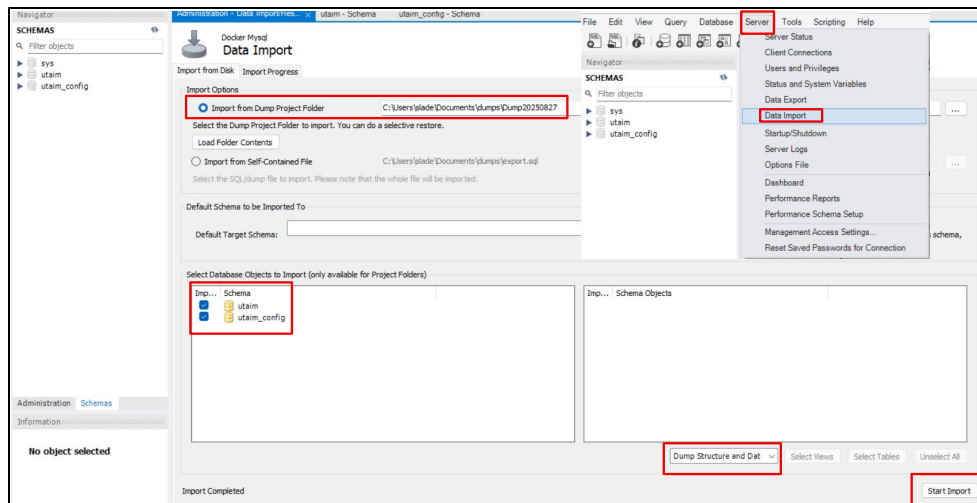


Figure 4-17: Restore Data in MySQL Workbench

Updating Intersection Markers in UDOT SPM

1. After restoring the MOE database, confirm intersections appear correctly in UDOT SPM (See Fig. 4-18 – Signal List in UDOT SPM system).
2. Open PowerShell and run: `dockerexec -it UDOT_SPM_v4.3 powershell`
3. Within the container PowerShell session, execute the intersection-marker generation utility.
4. Refresh the UDOT ATSPM webpage. The generated intersection markers should now be visible and interactive on the map.

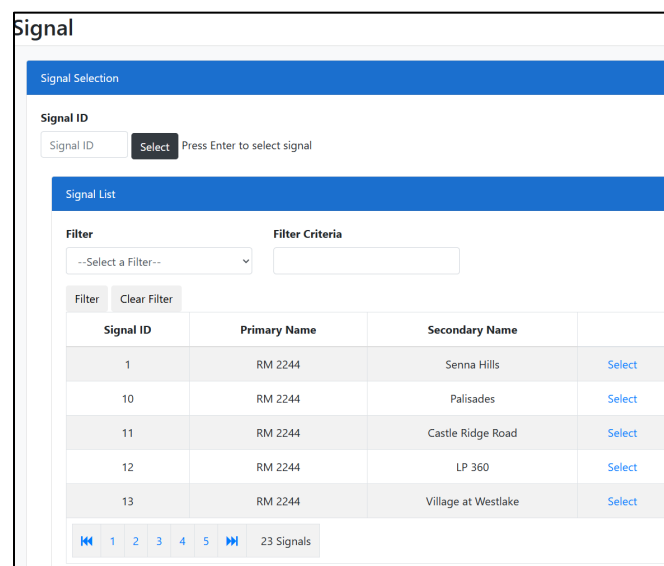
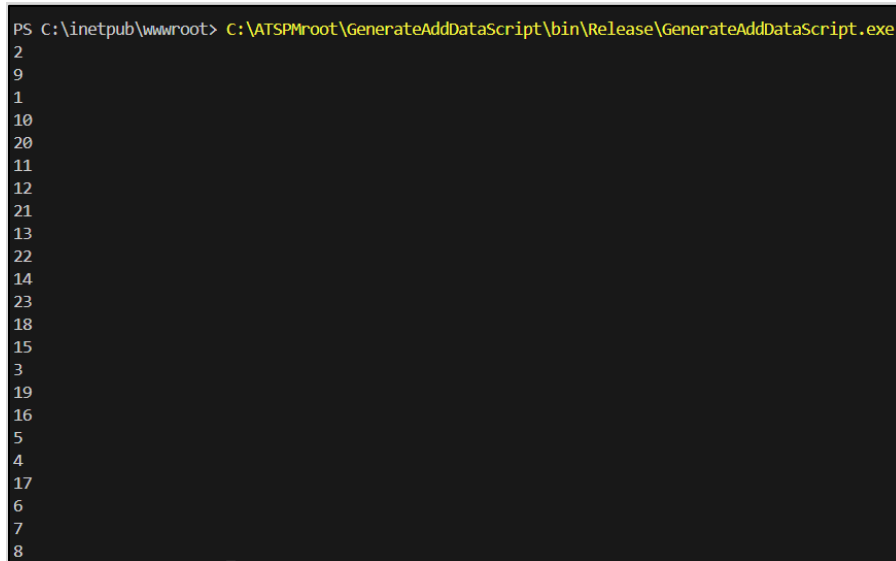


Figure 4-18: Signal List in UDOT SPM system

The executable name must be verified before finalizing this command. Section 2.12 identifies it as GenerateAddDataScript.exe, whereas the current section identifies it as GeneratedAddDataScript.exe. Use the exact filename present in:

C:\ATSPMroot\GenerateAddDataScript\bin\Release

Once confirmed, the command should follow this format: Start-Process "C:\ATSPMroot\GenerateAddDataScript\bin\Release\GenerateAddDataScript.exe". This utility generates map markers for intersections defined in the MOE database.



```
PS C:\inetpub\wwwroot> C:\ATSPMroot\GenerateAddDataScript\bin\Release\GenerateAddDataScript.exe
2
9
1
10
20
11
12
21
13
22
14
23
18
15
3
19
16
5
4
17
6
7
8
```

Figure 4-19: Successfully Run of GenerateAddDataScripts for intersection markers on the map.

4.2. Configuring the VISSIM Model to Generate ATSPM-Compliant Data

Traffic Demand Preparation and Volume Balancing

Traffic demand inputs were prepared for each direction and simulation time interval before configuring the VISSIM signal controllers and output settings. Base traffic volumes at the external network entry points were entered directly into the corresponding VISSIM vehicle inputs. At intermediate intersections and access points, the entering and exiting corridor volumes were compared to determine the change in traffic demand along the corridor.

The downstream corridor volume was balanced using the following relationship:

$$\text{Downstream volume} = \text{Upstream volume} + \text{Entering volume} - \text{Exiting volume}$$

For locations where the entering volume exceeded the exiting volume, the positive difference represented additional traffic entering the corridor. This net volume was assigned to the

corresponding internal vehicle input in VISSIM. When the difference was zero or negative, the vehicle input was set to zero because VISSIM does not permit negative vehicle-input volumes. Traffic leaving the corridor was represented through the vehicle-routing decisions and was therefore not entered as a negative demand.

The balanced volumes were transferred into the existing VISSIM volume input spreadsheet using the corresponding roadway direction, link, and simulation time interval. The original row order, time-interval columns, and vehicle-composition assignments were retained to ensure compatibility with the VISSIM network input structure. This procedure maintained continuity between traffic entering, traveling through, and exiting the modeled corridor without introducing invalid negative vehicle inputs.

VISSIM with the RBC Traffic Control Emulator

The RBC is VISSIM's built-in traffic signal controller module used to simulate signalized intersection operations. In this project, the RBC served as the built-in controller configuration for generating high-resolution signal events and detector-actuation records that were later converted into ATSPM-compliant datasets. Unlike external SILS controllers, the RBC derives events directly from simulated signal-color transitions, allowing efficient modeling of multiple intersections within a single VISSIM network without external controller software. To ensure ATSPM compatibility, the configuration followed strict structural requirements, including standardized detector naming, proper overlap settings, sequential detector-layout definitions, and consistent database-output connections. These elements ensure reliable parsing of signal and detector events into ATSPM-compliant formats. Detailed implementation procedures are provided in Section 1.2, with the following discussion summarizing the key configuration principles applied to the RM 2244 digital twin deployment.

4.2.1. Detector Events Output

The RBC generates two types of records:

- Signal transition records → Stored directly in the SQL Server database (e.g., Green → Yellow → Red phase changes).
- Detector actuation records → Saved as .ldp text files, logging ON/OFF detector status at each time step.

In VISSIM 2024, detector actuations cannot be written directly to the database and are instead saved as .ldp files.

As shown in Fig. 4-20, the .ldp file format uses simple characters:

- . (period) = OFF

- Any other character = ON

These files are later parsed into ATSPM event logs for signal performance analysis.

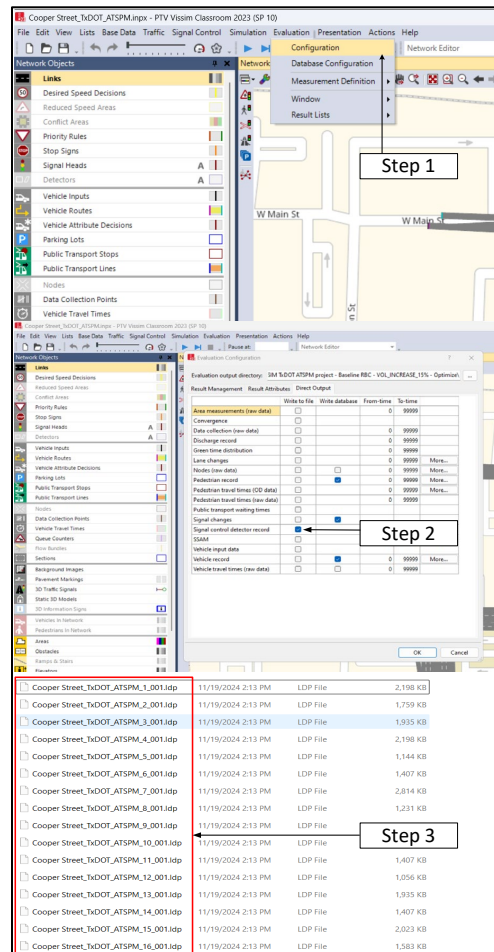


Figure 4-20: Example of .ldp detector actuation file format used in VISSIM.

4.2.2. Pedestrian Detectors

Pedestrian detectors must follow ATSPM-compatible configuration. In the RBC, they use the IXX format, where XX corresponds to the concurrent vehicular phase.

Examples:

- Pedestrian crossing concurrent with Phase 04 → 104
- Pedestrian crossing concurrent with Phase 08 → 108

This numbering ensures correct phase mapping during parsing. Pedestrian detector placement is illustrated in Fig. 4-21.

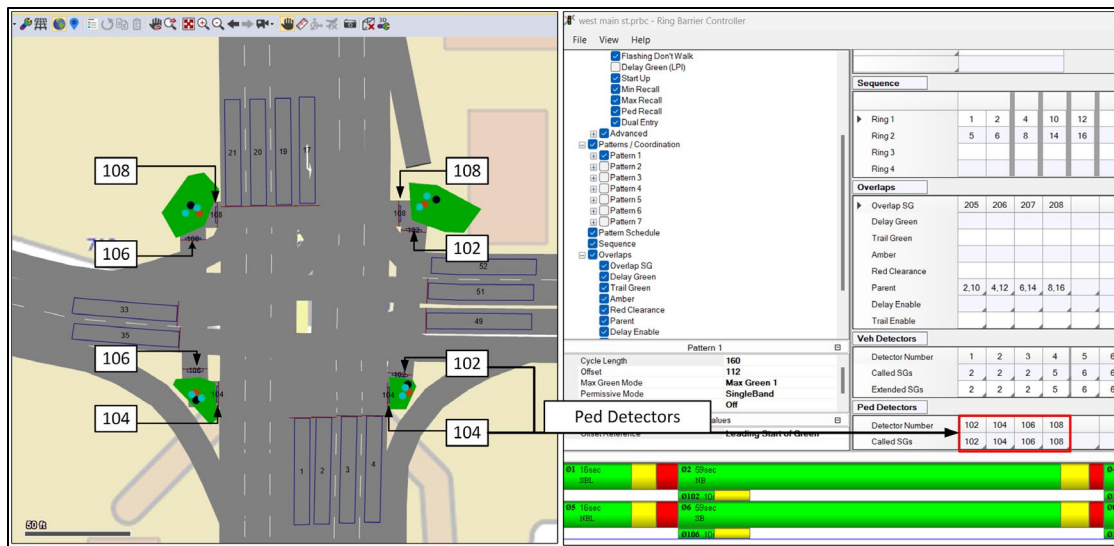


Figure 4-21: Pedestrian detector placement and numbering convention in VISSIM.

4.2.3. Overlaps

Overlaps allow movements to run simultaneously with parent phases. In the RBC, overlaps must follow the 2XX format as in Fig. 4-22. Incorrect numbering prevents proper parsing into UDOT ATSPM and UTAIM.

Example:

- 205 → Overlap between Phase 2 and Phase 10, allowing shared green time for through movements.

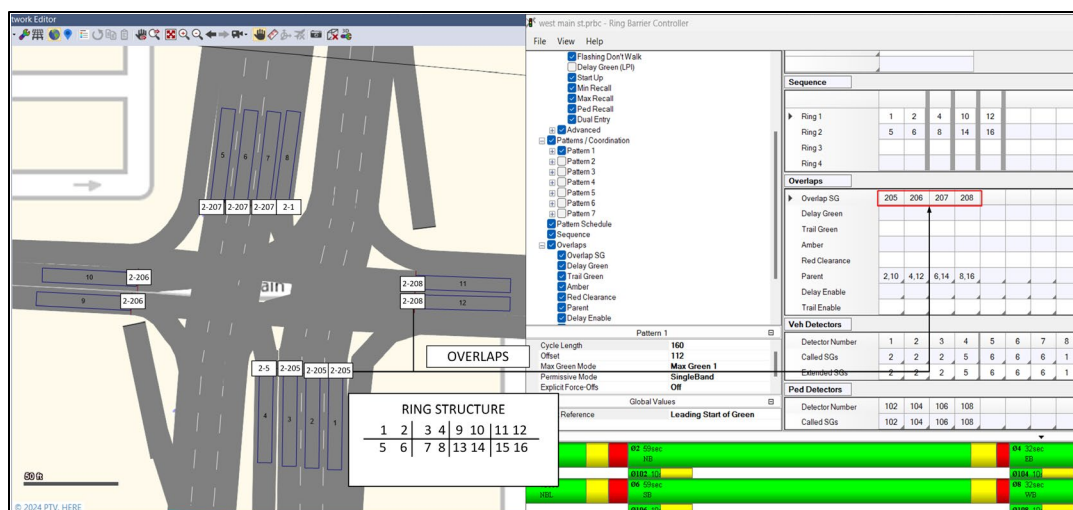


Figure 4-22: Example of overlap configuration using RBC in VISSIM.

4.2.4. Sequential Detector Layout Setup

The RBC requires a consistent detector numbering layout to correctly interpret actuation records. This layout is defined in the CSV file “List of Detectors.csv.”

File structure:

- Column 1 $\rightarrow$ Intersection number
- Column 2 $\rightarrow$ Total number of detectors
- Remaining Columns $\rightarrow$ Detector port numbers (in sequential order)

Example: Intersection 9 with 12 detectors $\rightarrow 1, 2, 3, 4, 5, 6, 8, 12, 13, 14, 15, 16, 17$ (see Fig. 4-23).

In Step 2 of the parsing GUI, users must select the correct List of Detectors.csv file containing detector ports in the required order and the proper VISSIM simulation folder containing .ldp files; otherwise, parsing will be incorrect. The parser reads this CSV file alongside the generated .ldp files to decode detector status accurately (see Section 1.2.2).

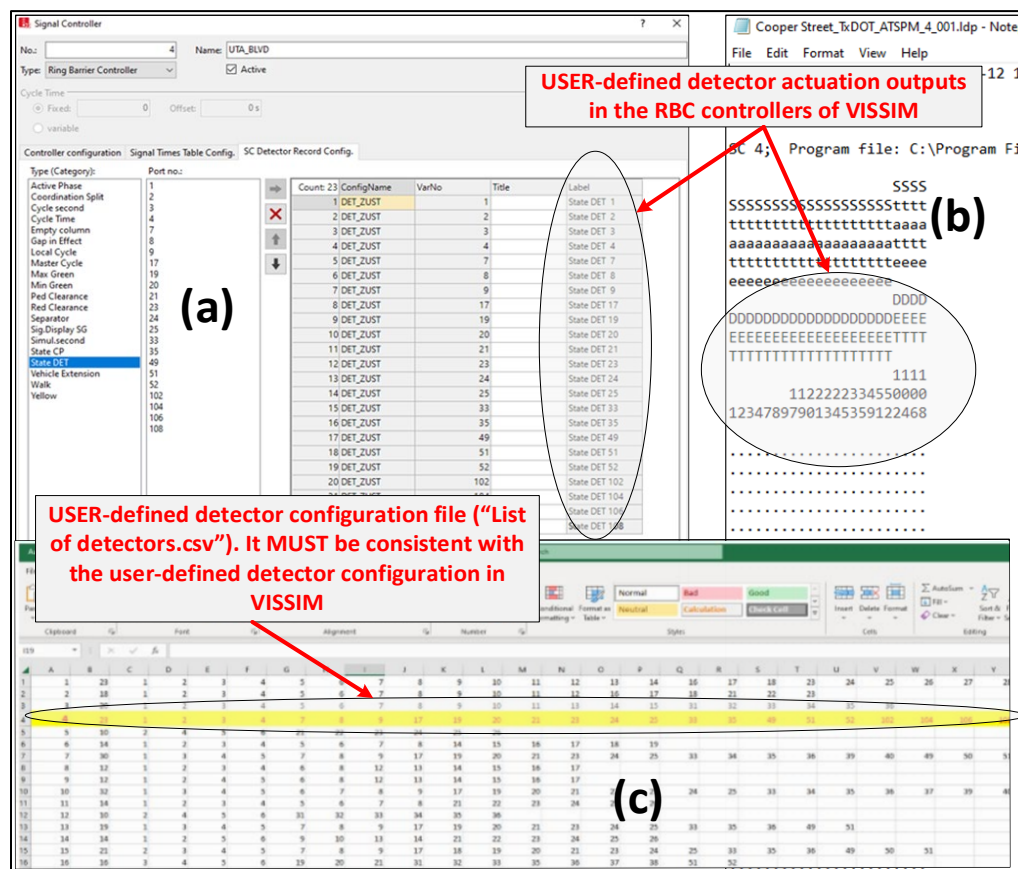


Figure 4-23: RBC detector actuation outputs and decoding

4.2.5. Enabling High-Resolution Simulation Output

By default, VISSIM does not output data at ATSPM-required resolution. Users must enable high-resolution logging to capture detailed detector states and vehicle records.

Configuring WGS84 in VISSIM

- Convert relative coordinates (x, y) to WGS84 (lat, lon).
- Import User-Defined Attributes (UDAs) for WGS84 trajectories via the VISSIM template.
- In VISSIM: Files → Read Additionally → Network
- Select WGS84 Coordinate input.
- Enable the UDA by checking the Read column (see Fig. 4-24).

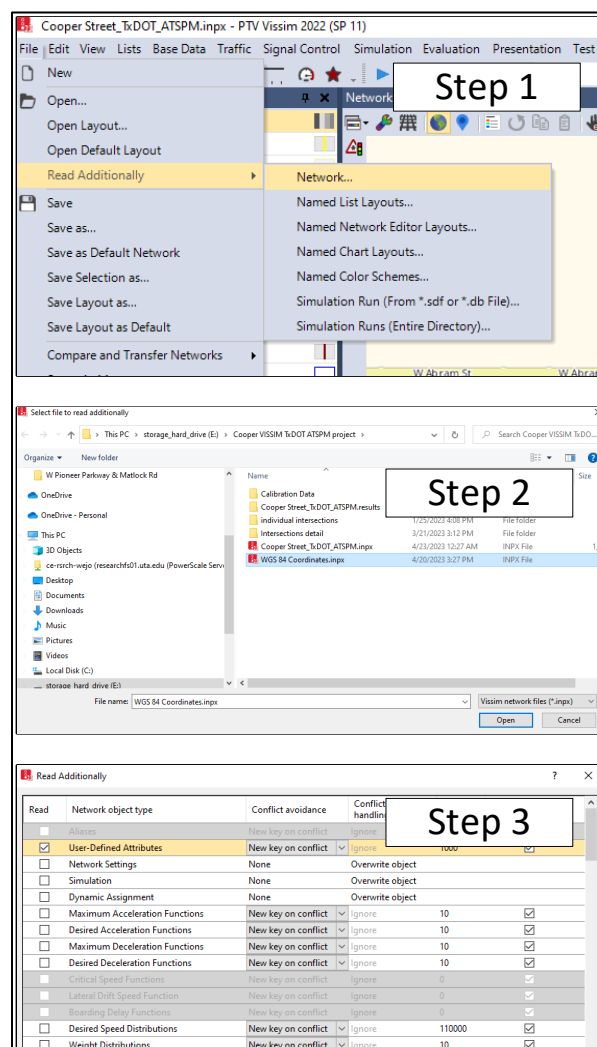


Figure 4-24: Configuring WGS84 coordinate output in VISSIM.

Configuring SC Detector Records in VISSIM

- Open the SC Detector Record Config tab (see Fig. 4-25).
- Select the State DET category.
- Choose the relevant detector port numbers.
- Click Add to apply the detector configuration.

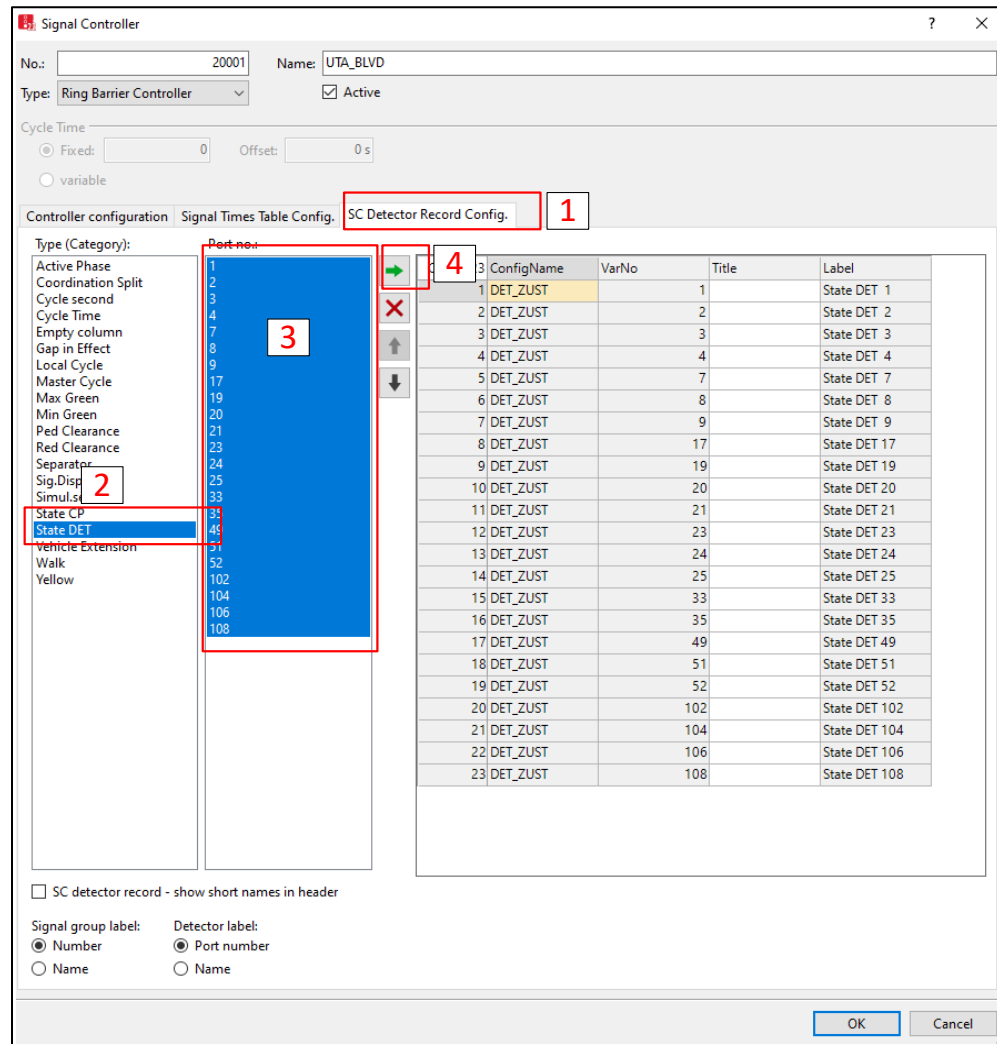


Figure 4-25: SC Detector Record configuration in VISSIM.

Database Output Configuration

VISSIM can save outputs locally or directly to a SQL Server database. For UDOT ATSPM, results must be written to Microsoft SQL Server. The database must be created beforehand, and SSMS can be used for its setup.

Configuration Steps:

1. Go to Evaluation → Database Configuration → Data Link Properties.
2. Select the correct OLE DB Provider.
3. Enter database details:
 - Server name
 - Username
 - Password
 - Database name
4. Click Test to verify the connection (see Fig. 4-26).

A confirmation message indicates VISSIM can access the database successfully.

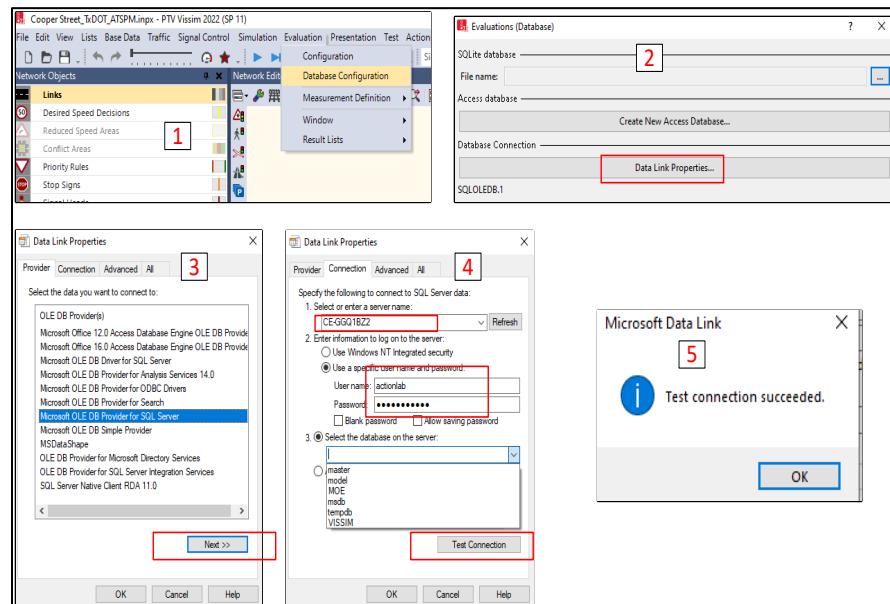


Figure 4-26: Database configuration interface in VISSIM for SQL Server output.

4. Configuring Output Evaluation in VISSIM

Configure VISSIM to export vehicle and traffic data in the required ATSPM format.

- Go to Evaluation → Configuration.
- Select attributes: Simulation run, Number, Simulation second, Simulation time, Latitude front, Longitude front, Speed.
- Ensure the exact sequence shown in Fig. 4-27.
- Set resolution to 1 time step.

Trajectories can be exported for selected vehicle classes; in this case, a new “CV” class (10% of vehicles) is defined to output trajectory data in the required format.

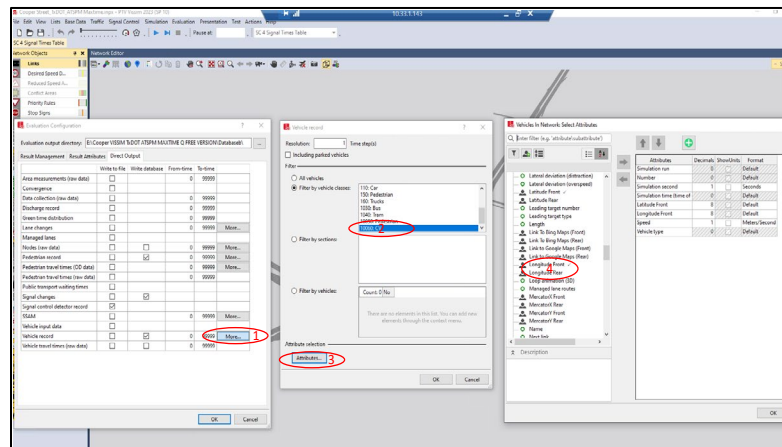


Figure 4-27: Output evaluation setup

To store simulation results in an ATSPM-compatible format, VISSIM must be connected to a SQL Server database. The RBC outputs are automatically written to three primary tables:

- `dbo.PEDESTRIAN` → pedestrian calls and activations
- `dbo.SIGNALCHANGES` → signal phase transitions
- `dbo.VEHICLE` → simulated vehicle trajectories

As shown in Fig. 4-28, these outputs are consolidated within the SQL Server database. Multiple simulation runs can be stored in the same database, ensuring all signal and trajectory data remain accessible for ATSPM analysis.

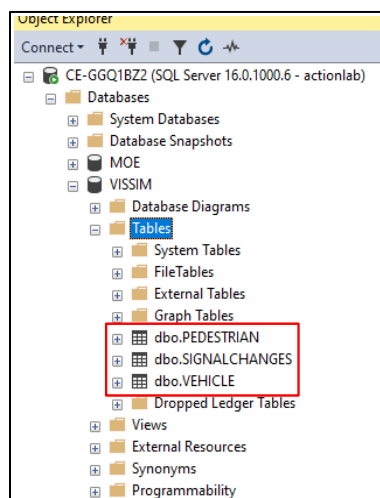


Figure 4-28: Consolidated simulation outputs stored in SQL Server for ATSPM.

4.3. VISSIM with Econolite EOS Control Emulator

The Econolite EOS controller emulator is an external, software-based traffic signal controller that replicates field-controller operations within VISSIM. It enables testing of advanced features such as coordination, preemption, pedestrian timing, and detection in a simulated environment.

In this setup, EOS is integrated with VISSIM using a SILS architecture, allowing dynamic interaction between the traffic network and the controller before field deployment. VISSIM 2024 or later is required, as earlier versions do not support EOS integration

4.3.1 Adding an Econolite EOS Signal Controller in VISSIM

Step 1 – Create the Controller

1. Go to the Signal Control tab.
2. Select Signal Controllers → Add a Controller.
3. Choose Econolite EOS as the controller type.

Use a clear controller name matching the intersection.

Step 2 – Configure the Controller

1. A warning will appear: “*Controller configuration file doesn’t exist.*”
2. Click Yes to create it.
3. A .cfg configuration file will be generated in the working folder.
4. Update the signal groups (EOS creates 48 signal groups by default; see Fig. 4-29).

The initial warning is normal during the first setup.

Step 3 – Simulation & Web Front Panel

1. In controller settings, locate “Automatically open web front panel during simulation.”
 - Leave this unchecked initially to avoid errors.
2. Run the simulation.
3. To access EOS: Right-click controller → Open Web Front Panel.
4. A browser window will open for monitoring and adjustments.

Use the web front panel only after all controllers are configure.

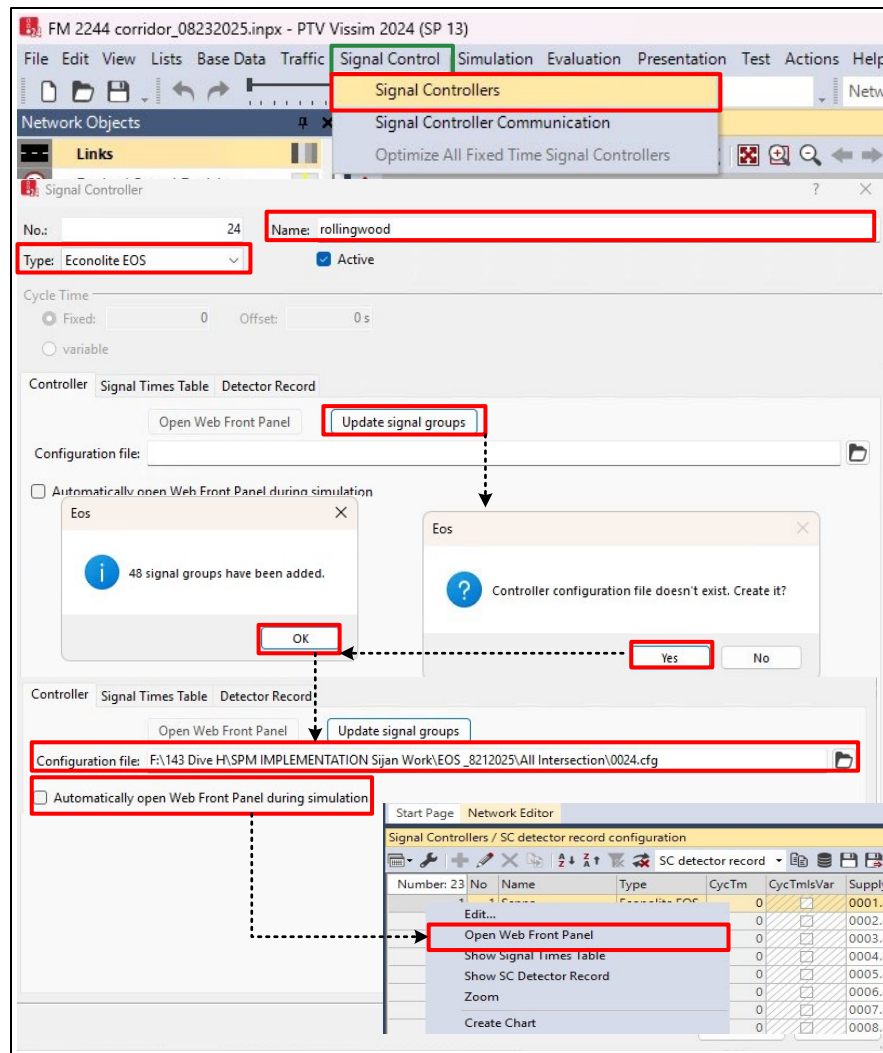


Figure 4-29: EOS Signal Controller in PTV VISSIM

4.3.2. Web Front Panel Configuration

The Web Front Panel allows browser-based interaction with the Econolite EOS controller, opening locally on assigned ports (default starting at 8081) for real-time control (see Fig. 4-30).

Step 1 – Launch Web Front Panel

1. After creating the controller and running the simulation, the Web Front Panel can open automatically in a browser.
2. Each intersection is assigned a unique localhost port, starting at 8081 (e.g., 8082, 8083, etc.).

The parsing software (VISSIM → UDOT) works correctly only when the intersection number begins with “1.”

Step 2 – Manage Port Conflicts

1. If the assigned port (e.g., 8083) is already in use, the panel may not display.
2. In this case:
 - Create a new VISSIM file.
 - Edit the controller .cfg file to assign an available port.
 - Reload the updated configuration into the original network. Enable the Web Front Panel only when needed for testing. Keep it off during standard runs to prevent conflicts.

Step 3 – Simulation without Web Front

1. The simulation can run normally without the Web Front Panel.
2. Data output continues based on the configured .cfg setup.
3. Disabling the Web Front prevents port conflicts while maintaining full simulation functionality.

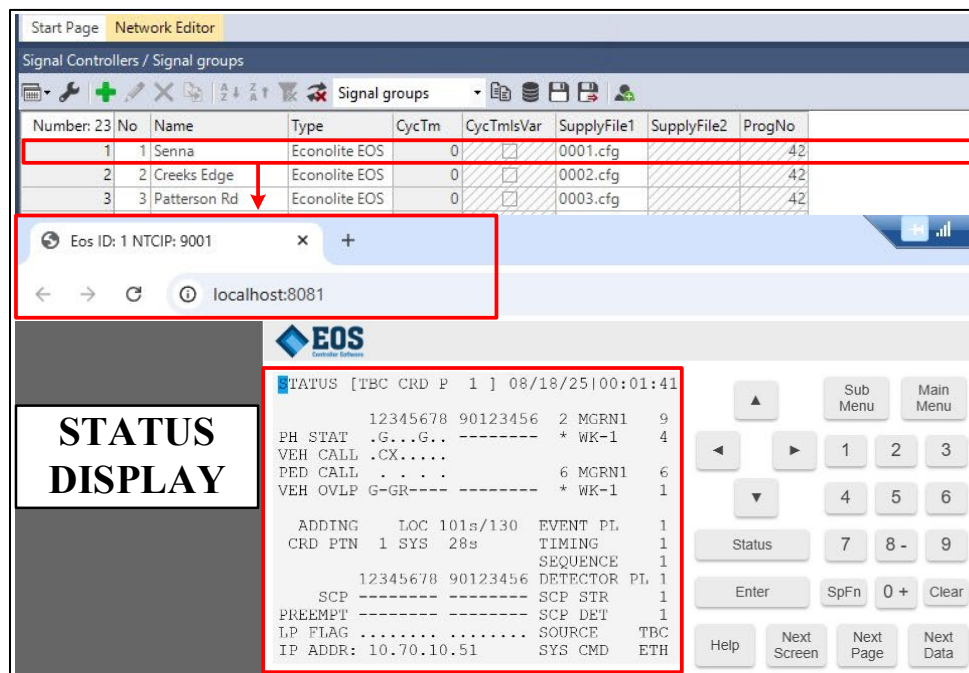


Figure 4-30: Econolite EOS Web Front Panel interface in VISSIM.

4.3.3. Main Menu (MM) and Sub Menu

The Main Menu (MM) is the starting point for navigating all EOS controller functions, including configuration, timing, detector settings, coordination, and diagnostics.

Step 1 – Access Main Menu

1. Press the Main Menu button on the keypad.

If already inside a submenu, pressing MM returns you here.

Step 2 – Select Desired Function

1. Use the arrow keys to move the highlight.
2. Press Enter to select.

You can also press the corresponding number key (1–9) to select a function directly.

Available Main Menu Options

1. Configuration – System parameters and defaults
2. Controller – Signal timing plans and phase operations
3. Coordinator – Coordination patterns and offsets
4. Preemptor/SCP – Preemption settings (e.g., emergency, rail)
5. Scheduler – Day plans and time-of-day schedules
6. Detector – Detector setup and calibration
7. Status Display – Live controller monitoring
8. Utility – Memory, firmware, and file tools
9. Diagnostic – Troubleshooting and performance checks

After selection, the corresponding submenu loads automatically. Changes are saved upon pressing Enter.

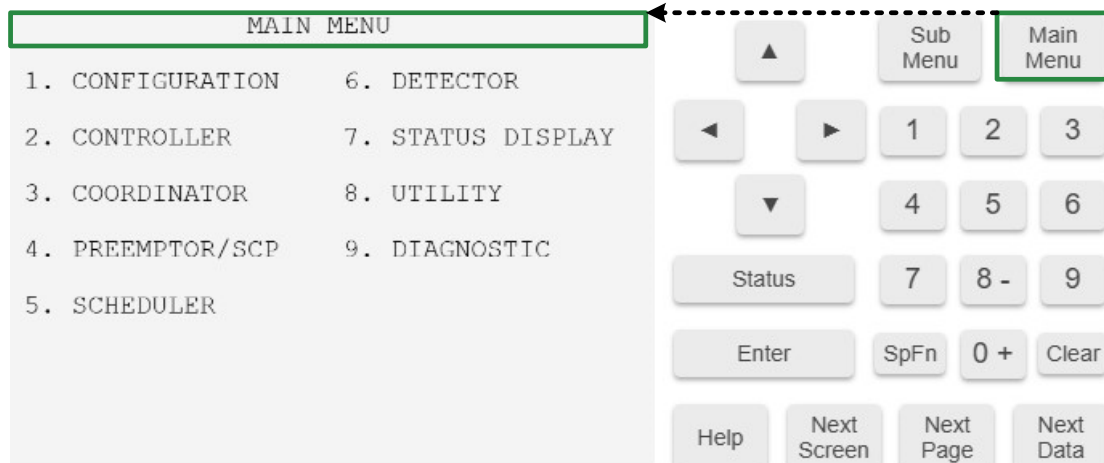


Figure 4-31: Main menu navigation in Econolite EOS controller.

4.3.4. Controller Settings

The Controller Submenu also includes the following settings, as shown below in Fig. 4-32. The principal settings used in this implementation are described in the following procedures.

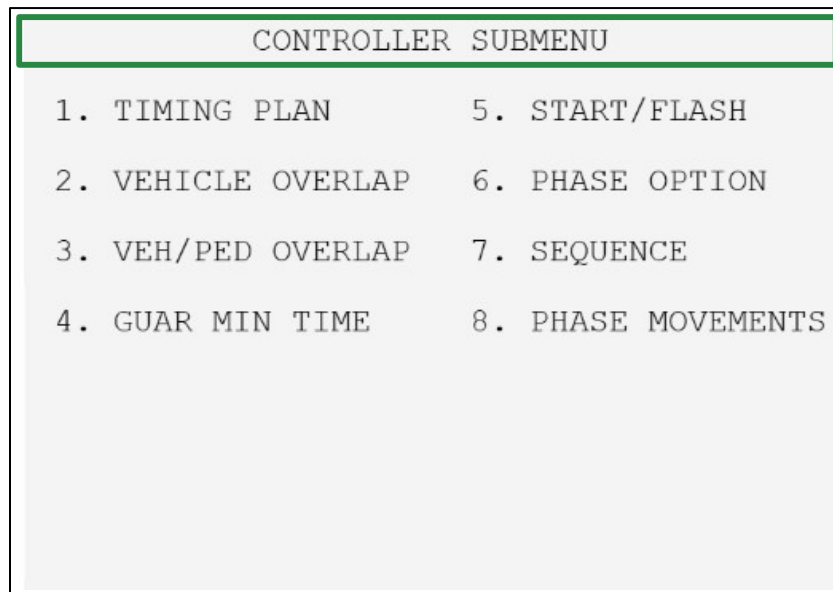


Figure 4-32: Controller Submenu in Econolite

A) Timing Plans (MM-2-1)

Timing Plans define basic signal timing parameters for uncoordinated operation, including green, yellow, red, and pedestrian intervals. Each controller supports nine programmable timing options, which should align with traffic demand and agency policies.

Step 1 – Open Timing Plans

1. From the Controller Submenu, select Timing Plans.
2. Choose the appropriate Timing Plan based on time of day (select the desired action plan).

Multiple plans allow different timing sets for varying traffic conditions or special events.

Step 2 – Select Programming Option

1. Within the selected plan, choose the timing parameter to program.
2. Select one of the following options (each programmed separately):
 - Minimum Green
 - Passage
 - Maximum Green
 - Pedestrian
 - Clearance
 - Phase Recall
 - Overlap
 - Phase Output

Ensure timing parameters are coordinated with corridor-level strategies to maintain smooth progression.

TIMING PLANS SUBMENU	
1. MIN GREEN	6. PHASE RECALL
2. PASSAGE	7. OVERLAP
3. MAX GREEN	8. PHASE OUTPUT
4. PEDESTRIAN	9. TIMING SUMMARY
5. CLEARANCE	

Figure 4-33: Timing plan setup screen (MM-2-1) in EOS controller.

1) Minimum Green (MM-2-1-1)

Minimum Green is the initial portion of the green phase and represents the shortest guaranteed green time to serve vehicles. It ensures queued vehicles can begin clearing before the phase terminates.

Configuration:

1. From the Controller Submenu, select the phase to program.
2. Enter the Minimum Green (initial green) duration in seconds.

The actual maximum green time may be influenced by added initial, vehicle extension, bike minimum green, guaranteed minimum green, and pedestrian walk + clearance intervals.

TIMING PLANS SUBMENU		TIMING PLAN [1] >							
1. MIN GREEN	6. PHASE RECALL	MINIMUM GREEN							
2. PASSAGE	7. OVERLAP	PHASE	1	2	3	4	5	6	7 8
3. MAX GREEN	8. PHASE OUTPUT	GREEN	5	15	5	0	5	15	0 0
4. PEDESTRIAN	9. TIMING SUMMARY	BIKE	0	0	0	0	0	0	0 0
5. CLEARANCE		COND SRV	0	0	0	0	0	0	0 0
		VARIABLE INITIAL							
		PHASE	1	2	3	4	5	6	7 8
		LANE	NO	NO	NO	NO	NO	NO	NO NO
		#ACT B4	0	0	0	0	0	0	0 0
		SEC/ACT	0.0	0.0	0.0	0.0	0.0	0.0	0.0 0.0
		MAX	0	0	0	0	0	0	0 0

Figure 4-34: Minimum Green interval configuration (MM-2-1-1).

2) Passage (MM-2-1-2)

The Passage interval allows the green phase to extend based on vehicle detection, ensuring efficient service without unnecessary delay.

Configuration:

1. Define the volume or occupancy thresholds that trigger green extension, consistent with traffic demand and detector placement.
2. Set EXTEND 1 (see Fig. 4-35), which specifies the number of seconds the green phase is extended per valid actuation.

A phase may gap out only after the minimum green has expired, a conflicting call exists, and the passage timer has expired.

TIMING PLAN [1] >								
VEHICLE PASSAGE								
PHASE	1	2	3	4	5	6	7	8
EXTEND1	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
EXTEND2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
VOLUME OCCUPANCY								
PHASE	1	2	3	4	5	6	7	8
BEFORE	0	0	0	0	0	0	0	0
#CARS	0	0	0	0	0	0	0	0
#STEP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
REDUCE	0	0	0	0	0	0	0	0
MIN GAP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 4-35: Passage time configuration (MM-2-1-2).

3) Maximum Green (MM-2-1-3)

Maximum Green sets the upper limit a phase can remain green to prevent overextension.

Setup:

- Enter MAX1 / MAX2 / MAX3 for the active time plan.
- Behavior varies by cycle activity.

Dynamic Max (Optional):

- Two consecutive max-outs → max increases by a Dynamic Step (up to Dynamic Max).
- Two consecutive gap-outs → max decreases toward the original setting.
- If a phase alternates between a gap-out and a max-out, the maximum-green value does not change.

Frequent Dynamic Max changes can affect cycle stability and corridor coordination.

TIMING PLANS SUBMENU								
1. MIN GREEN	6. PHASE RECALL	TIMING PLAN 1						
2. PASSAGE	7. OVERLAP	MAX GREEN DATA						
3. MAX GREEN	8. PHASE OUTPUT	PHASE	1	2	3	4	5	6
4. PEDESTRIAN	9. TIMING SUMMARY	MAX1	15	80	20	0	0	90
5. CLEARANCE		MAX2	0	0	0	0	0	0
		MAX3	0	0	0	0	0	0
		DYNAMIC MAX						
		PHASE	1	2	3	4	5	6
		DYM MAX	0	0	0	0	0	0
		DYM STP	0.0	0.0	0.0	0.0	0.0	0.0

Figure 4-36: Maximum Green interval configuration (MM-2-1-3).

4) Pedestrian (MM-2-1-4)

This screen is used to configure WALK, pedestrian-clearance, and pedestrian-extension intervals.

Basic Settings

- WALK → Time the WALK signal is displayed after a pedestrian call
- PED CLR → Flashing DON'T WALK interval after WALK
- Ped Extension → Additional time if applicable

The system uses the greater of the programmed and guaranteed WALK or pedestrian-clearance values.

Optional Settings

- WALK2 / CLEAR2 → For secondary pedestrian phases
 - Set via TBEP, detector assignment, or cabinet input mapping
- HAWK Mode (if applicable) → Configure ingress phase and travel time

HAWK operation differs from standard pedestrian phasing and requires the corresponding controller and signal-display configuration.

TIMING PLAN [1] > v								
PEDESTRIAN								
PHASE	1	2	3	4	5	6	7	8
DELAY	0	0	0	0	0	0	0	0
WALK	0	10	0	10	0	10	0	10
CLEAR	0	16	0	16	0	16	0	16
ADVANCED	0	0	0	0	0	0	0	0
ALTERNATE								
PHASE	1	2	3	4	5	6	7	8
WALK2	0	0	0	0	0	0	0	0
CLEAR2	0	0	0	0	0	0	0	0
PEDESTRIAN CARRY OVER								
PHASE	1	2	3	4	5	6	7	8
TO PHASE	0	0	0	0	0	0	0	0
MAX EXTENSION								
PHASE	1	2	3	4	5	6	7	8
WALK	0	0	0	0	0	0	0	0
CLEAR	0	0	0	0	0	0	0	0
HAWK								
PHASE	1	2	3	4	5	6	7	8
INGRES PH	0	0	0	0	0	0	0	0
TRAVL TM	0	0	0	0	0	0	0	0

Figure 4-37: Pedestrian timing configuration (MM-2-1-4).

5) Clearance (MM-2-1-5)

This screen is used to configure yellow, red, and warning intervals to provide separation between conflicting movements.

Step 1 – Open Clearance Menu

- Navigate to: Main Menu → M-2-1-5 Clearance
- Options include: Clearance, Max Extension, AWF, HAWK Flash

5.1 Clearance

- Set Yellow and Red Clearance times
- Set Red Revert (minimum red before phase can re-service)

Follow MUTCD guidance to avoid intervals that are too short.

5.2 Max Extension

- Enable Red Extend to allow clearance to reach RED MAXIMUM when a detector is configured with RED EXTEND.

5.3 Advance Warning Flasher (AWF) – Optional

- Configure Pre-Clearance duration
- AWF outputs may include:
 - Pre-Clearance
 - Yellow Clearance
 - Red Clearance
 - Red Revert
 - When-not-timing
 - AUTO Flash

AWF should align with agency policies for warning systems.

5.4 HAWK Flash – Optional

- Configure pedestrian hybrid beacon intervals:
 - Yellow Flash (before solid yellow clearance)
 - Red Delay (after ped clearance, before vehicle red flash)

Changes are saved when you press Enter, but verify the settings to confirm the updates.

TIMING PLAN [1] >									
CLEARANCE									
PHASE	1	2	3	4	5	6	7	8	
PRECLEAR	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
YELLOW	5.0	5.0	4.0	0.0	0.0	5.0	0.0	0.0	0.0
RED	2.0	2.0	2.0	0.0	0.0	2.0	0.0	0.0	0.0
REVERT	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
MAX EXTENSION									
PHASE	1	2	3	4	5	6	7	8	
RED	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HAWK FLASH									
PHASE	1	2	3	4	5	6	7	8	
YELLOW	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
RED DLY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 4-38: Clearance interval configuration (MM-2-1-5).

6) Phase Recall (MM-2-1-6)

This screen is used to configure recall options for Phases 1 through 16, including Lock Detector, Vehicle Recall, Pedestrian Recall, Max Recall, Soft Recall, and No Rest. The settings also apply to Timing Plans 2 through 4.

Lock Detector

- Retains a vehicle call after the detector input is no longer active.
- Use this option only where detector operation is sufficiently reliable to avoid unintended calls.

Vehicle Recall

- Automatically places a vehicle call (even during yellow/red). Recommended for low-volume approaches with inconsistent detection.

Pedestrian Recall

- Automatically places a ped call when not in WALK.
- May increase cycle length if service isn't always needed.

Max Recall

- Phase always runs to maximum green.
- Dynamic Max is disabled.
- Useful in heavy traffic; avoid on minor approaches.

Soft Recall

- Places a call when the controller rests in another phase. Maintains service consistency in coordinated systems.

No Rest

- Prevents the controller from resting in the current phase.
- In the absence of detector calls, the controller proceeds to the next phase permitted to rest.

TIMING PLAN [1]	
PHASE	12345678 90123456
LOCK DETECTOR	
VEH RECALL	.2...6..
PED RECALL	
MAX RECALL	
SOFT RECALL	
NO REST	

Figure 4-39: Phase recall configuration (MM-2-1-6).

07) Overlap (MM-2-1-7)

This screen is used to configure advance, delay, pedestrian interaction, and trailing timing for vehicle overlaps.

Access:

- Main Menu → M-2-1-7 Overlap

Overlap Options

Advance

- Starts overlap during the yellow of the terminating phase.
- May hold all-red if Advance Green exceeds clearance.
- Improves progression but may add delay.

Delay

- Adds delay before overlap begins.
- Useful where added clearance or pedestrian protection is needed.

Advance Ped

- Delays vehicle overlap when the pedestrian phase is active.
- Can function like an LPI for overlaps.
- Must be coordinated to avoid excessive vehicle delay.

Trailing

- Extends overlap after parent phase ends.
- Common for shared lanes or smoother transitions.

A OVERLAP [A] TYPE[OVERLAP]	v B OVERLAP [A] TYPE[PPLT/FY]
12345678 90123456	PROTECTED TURN PHASE 1
INCLUDED PHASE .2.....	PERMISSIVE THROUGH PHASE 2
PROTECTED PHASE	OUTPUT MODE CH13 GRN OLP
INHIBIT WITH PED	INHIBIT WITH PED
INHIBIT WITH PHASE	EARLY FYA ENABLE NO
INHIBIT WITH GREEN	SPECIAL FUNCTION BIT DISABLE 0
TRAILING WHEN EXIT	OVERLAP [A] TYPE[PPRT/FY]
TRAILING WHEN NEXT	v
TRAILING PHASE ALLOW	PROTECTED TURN PHASE 0
GREEN WHEN NEXT	PERMISSIVE THROUGH PHASE 0
ABCDEF GH IJKLMNO P	INHIBIT WITH PED
INHIBIT WITH OVERLAP	EARLY FYA ENABLE NO
	SPECIAL FUNCTION BIT DISABLE 0

Figure 4-40: Overlap timing configuration (MM-2-1-7).

08) Phase Output (MM-2-1-8)

- Map phase outputs per cabinet requirements.
- Verify outputs match field wiring and detector mapping.

B) Vehicle Overlap (M-2-2)

A vehicle overlap allows one controller output to be associated with more than one phase and can support progression or movements requiring shared phase operation.

Access: Main Menu → M-2-2 Vehicle Overlap

Overlap Types

- Protected
- Pedestrian Protected
- Not Overlap
- Trailing
- Leading
- Flashing
- Advance Green

The selected overlap type defines how the overlap interacts with its included parent phases.

Flashing Yellow Arrow (FYA)

Protected/permissive left turn (PPLT)/FYA

- Configure flashing yellow arrow.
- Outputs: flashing yellow, yellow clearance, red clearance.
- Protected green controlled by PROTECTED TURN output.
- Improves safety by distinguishing permissive vs. protected turns.

Protected/permissive right turn (PPRT)/FYA

- Provides a similar operation for right-turn movements using an applicable four-section signal head.

Early FYA Enable

- Disables red-clearance timing when the display transitions from protected to permissive operation.
- This setting should be used only when it is consistent with the approved signal-display and timing design.

Special Function Bit Disable

- Disable FYA by time-of-day in Event Plan.
- Useful for restricting permissive turns during peak or safety-critical periods.

C) Veh/Ped Overlap (MM-2-3)

The Veh/Ped Overlap screen displays all vehicle and pedestrian overlaps that have been deployed within the controller configuration, as illustrated in Fig. 4-41.

VEH/PED OVERLAPS																	v
INCLUDED	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	
VEH OL A	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL B	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL C	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.
VEH OL D	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL E	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL F	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL G	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL H	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL I	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL J	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL K	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL L	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL M	X	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
VEH OL N	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Figure 4-41: Vehicle overlap configuration and deployment.

D) Phase Options (MM-2-6)

Used to configure phase behavior and intersection control logic.

Access: Main Menu → Phase Submenu → Controller Phase Options

Available Settings include options for Phases in Use, Dual Entry, Simultaneous Gap, Lock Phase Next, and conditional services.

Dual Entry

- Allows two phases to run simultaneously when both are called.
- Common for parallel movements (e.g., opposing through phases).

Review Other Key Options

- Phases in Use → Active phases
- Simultaneous Gap → Whether two phases gap out together
- Lock Phase Next → Holds a phase until the paired phase is ready
- Conditional Service / LRT / BRT / Phase / ADW WARN → Advanced coordination and special operation settings

T CONTROLLER PHASE OPTIONS V		
PHASES IN USE	12345678	90123456
DUAL ENTRY	12345678	
SIMULTANEOUS GAP		
LOCK PHASE NEXT	12345678	90123456
COND SERVICE TO		
COND SERVICE FROM		
LRT/BRT PHASE		
PRE GREEN CLEAR		
2 SECTION SIGNAL		
ADV WARN WIG/WAG		

Figure 4-42: Phase Options for managing phase behaviors.

E) Sequence Configuration (MM-2-7)

Used to configure phase order and timing logic, including:

- Controller Sequence
- Backup Prevent Plan
- No Serve features

Always place the controller in flash or bench (off-street) mode before making sequencing changes. Do not modify sequencing while the controller is active in the field.

SEQUENCE SUBMENU
1. SEQUENCE
2. BACKUP PREVENT
3. NO SERVE

Figure 4-43: Sequence submenu in the EOS controller for phase order and barriers.

Step 1 – Open Sequence Menu

- Navigate to: Main Menu → Sequence

Options:

- Sequence → Configure ring assignments and barriers
- Backup Prevent → Manage backup phases/plans
- No Serve → Define unserved phase sequences

Step 2 – Configure Sequence (MM-2-7-1)

2.1 Sequence

- Set ring assignments, barrier placement, and phase order
- Required for Lead-Lag configuration

Verify ring/barrier compatibility to avoid phase conflicts.

2.2 Independent Ring Group (Global Option)

- Allows Ring 3 and Ring 4 to cross barriers independently of Rings 1 and 2.

2.3 Sequence Commands

- Copy to another sequence
- Set or override default sequence
- Apply custom sequencing as needed

CONTROLLER SEQUENCE [1] >																
	B	-	B	-	B	-	B	-	B	-	B	-	B	-	B	-
RING 1		1	2		3	4		9	10		13	14		.	.	
RING 2		5	6		7	8		11	12		15	16		.	.	
RING 3		.	.		.	.		.	.		.	.		.	.	
RING 4		.	.		.	.		.	.		.	.		.	.	
INDEPENDENT RING GROUP NO																
HARDWARE ALT SEQUENCE NO																
SEQUENCE COMMANDS .																

Figure 4-44: Sequence configuration setup (MM-2-7-1).

Step 3 – Configure Backup Prevent

- Define backup phases and manage backup plans.

Ensures controller reliability if the primary sequence fails. Test backup plans in bench mode before deployment.

Step 4 – Configure No Serve

- Define sequences where certain phases are intentionally skipped.

This setting may be used for off-peak or special-event operation. However, the excluded phase combinations should be verified against the applicable signal-timing plan and expected demand.

4.3.5. Detector Setup in VISSIM and EOS (MM-6)

Detector setup in EOS includes assigning detectors to phases and configuring their operation.

1. Vehicle Detector Assignment Plan (MM-6-1)

Used to assign detectors to phases for accurate vehicle detection and proper phase operation.

Step 1 – Open Vehicle Detector Assignment

- Main Menu → Detector Submenu → Vehicle Assignment

The screen shows detector assignments for each phase, including Primary, Additional, and TS2 detectors.

Step 2 – Assign Detectors

- Assign a Primary detector for each phase (PH1 columns 1–8).

The primary detector is the main source for calling the phase.

- Assign Secondary detectors if additional coverage is needed (PH2 columns 1–8).

Useful for upstream detection or multi-lane coverage.

- Enable TS2 detectors if advanced detection logic is required (e.g., pedestrian or transit priority).

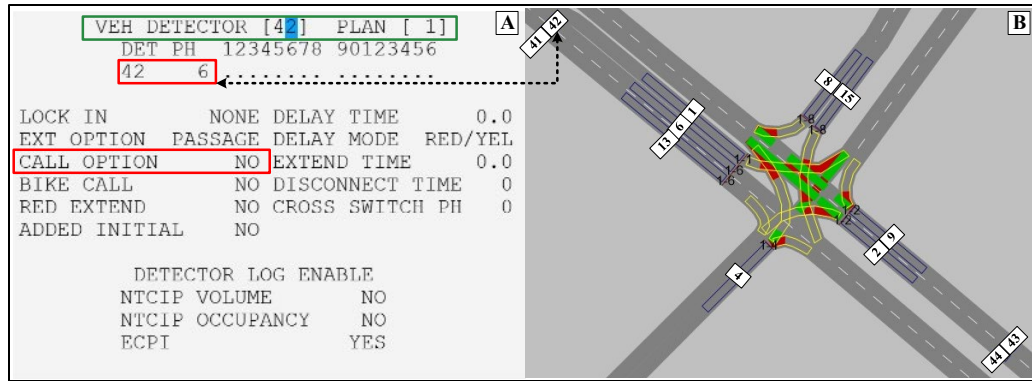


Figure 4-46: Advance detector placement and configuration in VISSIM.

4.3.6. VISSIM Signal Head Configuration

Signal heads configured in Econolite EOS must also be mapped correctly in VISSIM.

1. Signal Group / Vehicle Phase

- Format: xx – PhaseNumber: Phase Label
- Example: 19 – 5: Phase 5

Links each vehicle signal head to its corresponding controller phase (green, yellow, red timing).

2. Pedestrian Signals

- Format: xx – PedID: Pedestrian Label
- Example: 19 – 106: Pedestrian 6

This format assigns a unique identifier to each pedestrian movement while associating it with the applicable controller phase.

3. Overlap Signals

- Format: xx – OverlapID: Overlap Label
- Example: 19 – 213: Overlap M

This format is used for shared or special movements, such as protected-permissive turning movements. Each overlap must have a unique identifier to prevent incorrect signal-head mapping.

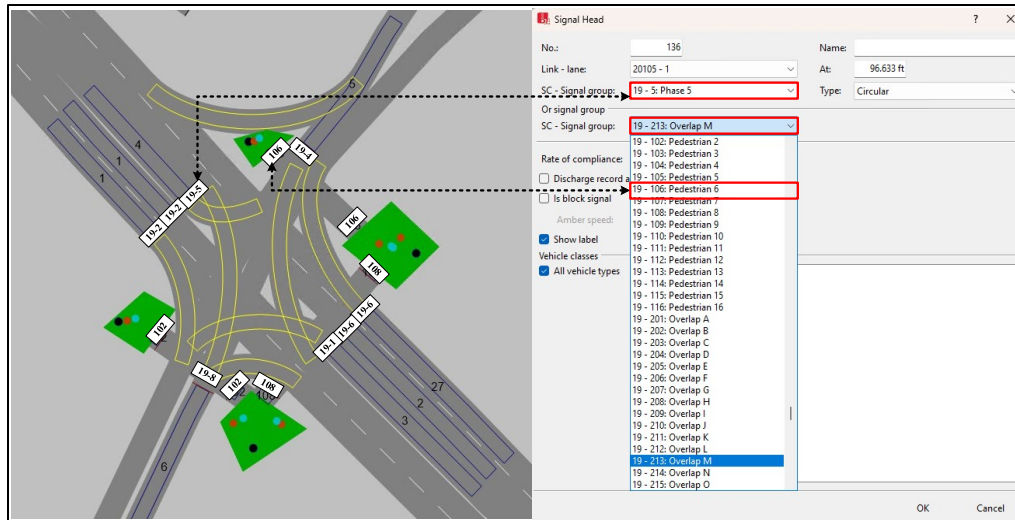


Figure 4-47: VISSIM signal head organization for vehicle, pedestrian, and overlap groups.

4.3.7. Stepwise Guideline for Setting Up a Coordination Plan (Event Plan)

This procedure is used to configure cycle length, offset, splits, and coordinated phases.

Step 1 – Access Event Plan

- Main Menu → Scheduler (MM-5) → Event Plan (Option 2)

Step 2 – Enable Coordination

- Set Event Plan Type = Coordination (COORD)

Step 3 – Enter Coordination Parameters

- Cycle Length (e.g., 120 s)
- Offset (e.g., 89 s)
- Sequence (e.g., 1)
- Actuated Coordination = YES (if required)

Step 4 – Assign Splits

- In Splits, allocate green percentages per phase.
- Ensure splits match the corridor timing plan.

Step 5 – Identify Coordinated Phases

- Locate Coord Phase in the Event Plan.

- Mark included phases (e.g., 2 and 6).
- These phases will follow the defined cycle and offset.

SCHEDULE SUBMENU	EVENT PLAN [1]								TYPE[COORD]	v
1. CLOCK OPTION	CYCLE LENGTH 120s								ACTUATED COORD	NO
	OFFSET VAL 89s								SEQUENCE	1
2. EVENT PLAN	1	2	3	4	5	6	7	8		
	SPLITS	15	80	0	25	0	95	0	25	
3. DAY PLAN	= 120s									
		9	10	11	12	13	14	15	16	
4. SCHEDULE	SPLITS	0	0	0	0	0	0	0	0	
5. EXCEPTION DAYS	COORD PHASE								12345678 90123456	
	FIXED FORCE OFF									
	ADAPTIVE SPLIT									
	VEHICLE EXTENSION 2									
	VEHICLE RECALL									
	WALK 2									

Figure 4-48: Event plan setup for coordination in EOS scheduler.

4.4. Parsing the VISSIM and EOS Output to the ATSPM Database

A custom parsing tool was developed to convert simulated traffic signal events into the database structure required by UDOT ATSPM. The built-in RBC supports efficient simulation setup and produces phase-transition and detector-actuation records that can be parsed for ATSPM analysis. Econolite EOS more closely represents an external controller environment and produces controller-originated high-resolution event records. However, EOS requires more detailed controller configuration and familiarity with signal-control operations than the RBC configuration.

4.4.1. Parsing RBC Signal Events: VISSIM to UDOT ATSPM

This process converts RBC-generated signal events into the UDOT ATSPM MOE database.

Step 1 – Prepare & Access

- Ensure VISSIM outputs are saved to SQL Server.
- Confirm UDOT ATSPM is installed.
- Identify both the VISSIM output database and UDOT ATSPM (MOE) database.
- Enter database credentials and simulation start time in the parser.

Both databases must be accessible from the system running the parser. The required database-table names are predefined.

Step 2 – Configure Parsing Parameters

- Load the parsing settings from the JSON configuration file described in Section 1.4.2.
- Modify the settings in the parser GUI or directly in the JSON file, as needed.
- Save the updated configuration for future parsing runs.

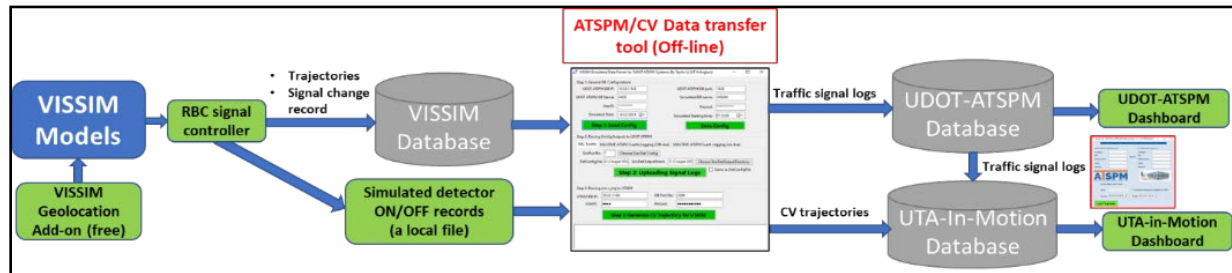


Figure 4-49: Validation of parsed simulation events in UDOT-ATSPM.

Step 3 – Execute & Validate

- Run the parsing software to convert RBC events into UDOT-ATSPM format.
- Verify results in the MOE database.

If issues occur, check database connections, CSV configuration, and VISSIM data integrity.

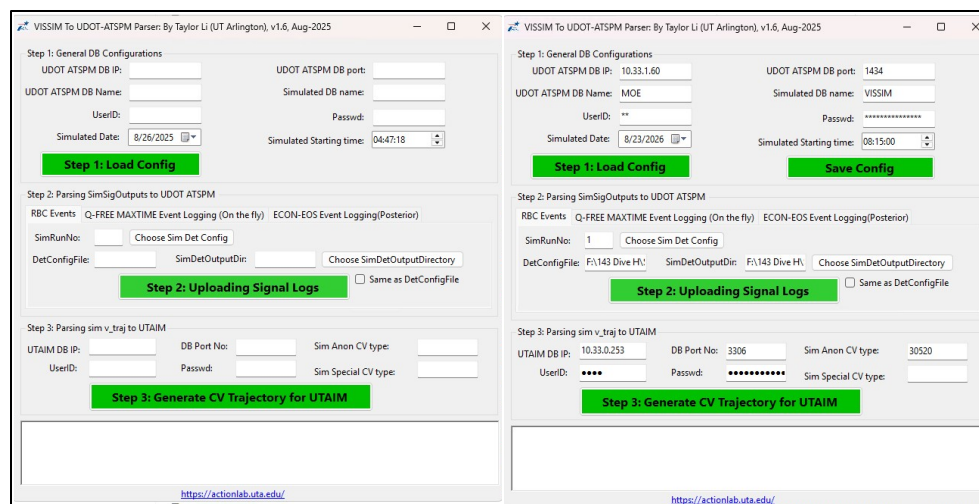


Figure 4-50: RBC parsing configuration: VISSIM to UDOT ATSPM.

4.4.2. Data Generation and Parsing of EOS Events into UDOT ATSPM

Using Econolite EOS in SILS mode, high-resolution signal events are generated for full ATSPM MOE extraction.

Step 1 – Enable Event Logging

- Main Menu → Logging → Event Logging
- Enable required logs (e.g., VIOL, MMU Faults, Detector Errors, SCP, Preemption, Alarms)
- Set each to YES/NO as needed (see Fig. 4-51)

Step 2 – Configure Data Logging Parameters

1) Virtual Input/Output Trace (VIOT)

- Captures all I/O events in real time (fixed record size).
- Existing trace files may be overwritten when the simulation is rerun; therefore, required files should be copied or archived before starting another run.

2) High-Resolution MOE Logging (HI-RES MOE)

- Options: 1MIN, 15MIN, 1HR
 - 15MIN → Recommended (balanced detail and file size)
 - 1MIN → Too large for efficient storage/processing
 - 1HR → Too aggregated; may hide short-term variations

MOE logs are generated only at the selected interval. If set to NO, no MOE data is recorded.

EVENT LOGGING							
VIOT	YES	HI-RES MOE	15MIN	ECON_10.70.10.51_2025_08_30_0800.datZ	8/30/2025 2:44 PM	DATZ File	12 KB
RFEs (MMU/TF)	YES	ACCESS	YES	ECON_10.70.10.51_2025_08_30_0815.datZ	8/30/2025 2:49 PM	DATZ File	11 KB
MMU FL FAULTS	YES	DATA CHANGE	YES	ECON_10.70.10.51_2025_08_30_0830.datZ	8/30/2025 2:55 PM	DATZ File	11 KB
3 RFEs > 24 H	YES	LOW BATTERY	YES	ECON_10.70.10.51_2025_08_30_0845.datZ	8/30/2025 3:01 PM	DATZ File	12 KB
RFE (DET/TEST)	YES	CTR DOWNLOAD	YES	eos.log	8/30/2025 3:06 PM	Text Document	6 KB
DETECTOR ERROR	YES	POWER ON/OFF	YES	NVRAM.bin	8/30/2025 2:40 PM	BIN File	1,595 KB
SCP	YES	LOCAL FLASH	YES	snmp_port	8/30/2025 2:38 PM	File	1 KB
PREEMPTION	YES	ONLINE/OFFLINE	YES	websocket_port	8/30/2025 2:38 PM	File	1 KB
ALARM 1	YES	ALARM 2	YES				
ALARM 3	YES	ALARM 4	YES				
ALARM 5	YES	ALARM 6	YES				
ALARM 7	YES	ALARM 8	YES				
ALARM 9	YES	ALARM 10	YES				
ALARM 11	YES	ALARM 12	YES				

Figure 4-51: Econolite EOS event logging configuration screen.

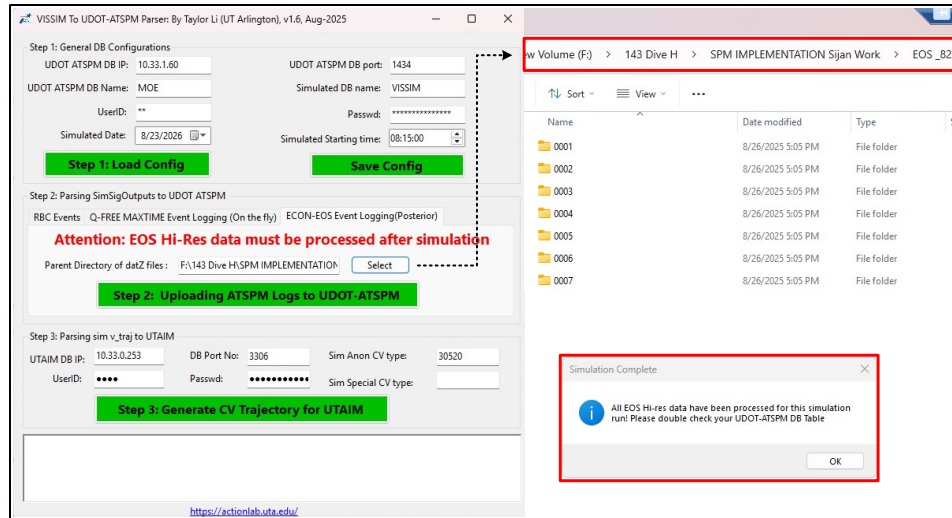


Figure 4-52: High-resolution data logging options in EOS controller.

4.4.3. Transferring ATSPM Data: UDOT ATSPM to UTAIM

Transfers ATSPM data within a selected time range for MOE visualization in UTAIM.

Step 1 – Prepare Databases

- Confirm ATSPM data exists in the UDOT ATSPM SQL Server database.
- Verify the UTAIM MySQL database is accessible.
- Open the transfer tool and enter:
 - Source SQL Server credentials
 - Destination MySQL credentials
 - From / To time range
 - Load settings from DBconfig.csv
 - Use the Save Parameter if updating the configuration

Ensure the selected simulation time falls within the specified transfer range.

Step 2 – Execute Transfer

- Click Start Data Transfer to move data to UTAIM.
- Use Flush UTAIM DB if clearing existing records is required.

After completion, the UTAIM database contains the transferred ATSPM signal records. Simulated CV trajectories are added through the separate trajectory-parsing workflow described in Section 1.4.4. UTAIM does not automatically synchronize with UDOT ATSPM; therefore, the transfer must be performed for each required dataset, as shown in Fig. 4-53.

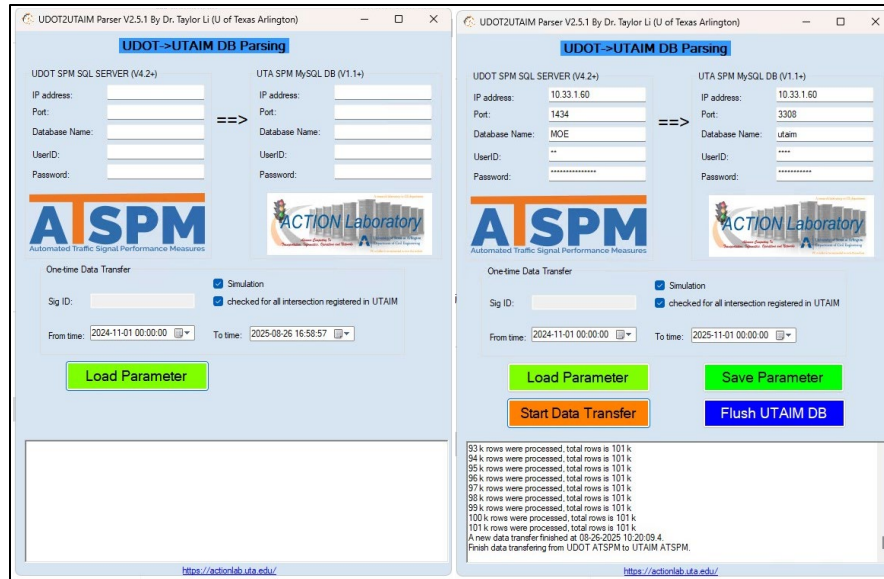


Figure 4-53: Data transfer process from UDOT ATSPM to UTAIM

4.4.4. Generating Figures from UDOT ATSPM and UTAIM

Simulations were conducted from 6:30 to 7:30 AM. The complete one-hour simulation period was used for the analysis, with no warm-up or cool-down period excluded.

- RBC outputs → Parsed into UDOT MOE database, then transferred to UTAIM
- EOS outputs (.datz files) → Stored directly in UDOT, then transferred to UTAIM

Performance Measures

UDOT ATSPM generates standard MOEs, including:

- Split Monitor
- Purdue Coordination Diagram (PCD)
- Approach volume
- Split failure
- Arrivals on red/green
- Left-turn gap
- Pedestrian delay
- Wait time
- Approach delay
- Preemption details

UTAIM provides additional outputs such as:

- Time–space diagrams (TSD)
- Arrivals on green/red (with trajectory integration)

CV-based outputs were not included in the representative analysis conducted for this implementation. Detailed corridor-specific performance figures and their interpretation are presented in Chapter 6.

Chapter 5: Hardware-in-the-Loop Simulation Control Interface Device

5.1. VISSIM Setup For HILS

This chapter describes the procedure for establishing a HILS model using the TxDOT HILS-CID. The system operates through NTCIP communication and uses VISSIM's External Signal Control interface to connect a physical traffic signal controller to the simulation environment. For the Project 5-7160 demonstration, the physical controller was a Q-Free MAXTIME controller.

5.1.1. Prerequisites

The HILS-CID is built upon VISSIM's External Signal Control function, which enables the simulation to interface with embedded external control logic. In this application, the external logic originates from the physical traffic signal controller hardware. Several prerequisites must be satisfied to ensure the HILS-CID operates correctly.

1. External Signal Control License in VISSIM: The VISSIM software is licensed for the external signal control module (see Fig 5-1).
2. Traffic signal controllers must support NTCIP 1202 v02 at least.

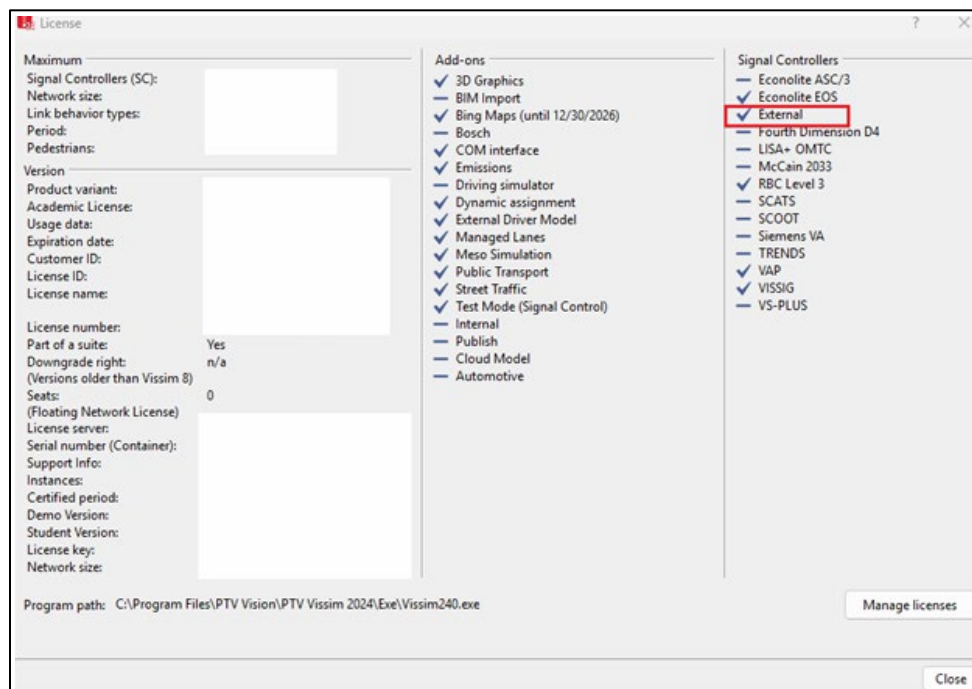


Figure 5-1: License summary in VISSIM

5.1.2. Set up an HILS intersection in VISSIM

To add a new HILS intersection, the user must first set the control type to “External”. Once selected, VISSIM will display the dialog shown in Fig. 5-2. For the *Program File*, select the provided HILS_TxDOT_DLL.dll. For the *Dialog DLL File*, choose STDSC_GUI.dll. For the *WTT File*, create an empty WTT file and add it to the configuration.

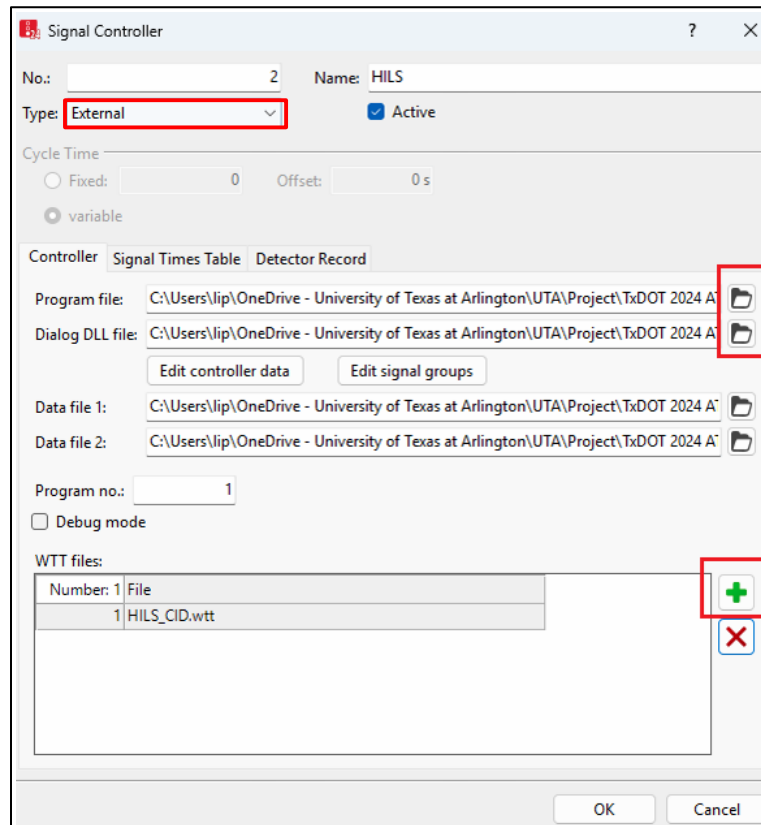


Figure 5-2: Set up external signal control in VISSIM for HILS

Next, the user must select Edit Controller Data or Edit Signal Groups to configure the port number and signal phase assignments. The IP address and port must match the parameters specified later in the HILS-CID setup. VISSIM and the HILS-CID communicate through a TCP socket using this designated IP-port pair.

The system supports 16 vehicle phases, 16 pedestrian phases, and 16 overlap phases, ensuring compatibility with all TxDOT-approved traffic signal controllers. The phase numbering must follow the required convention:

- Vehicle phases: 1–16
- Pedestrian phases: 101–116
- Overlap phases: 201–216

After configuring the IP address, port, and phase assignments, click OK. The STDSC_GUI.dll will automatically embed this information into the VISSIM network (INP file) and generate a configuration file named “x.scpprops”, where *x* corresponds to the intersection ID in VISSIM (e.g., for intersection 2, the file name is *2.scpprops*).

Important:

Users must not rename the *scpprops* file. It must retain the exact format “x.scpprops”, where *x* is the VISSIM intersection ID. Additionally, users should not add or remove phases by manually editing the file in a text editor. Only modifications made through the VISSIM dialog will be correctly written into the VISSIM network and reflected in simulation behavior.

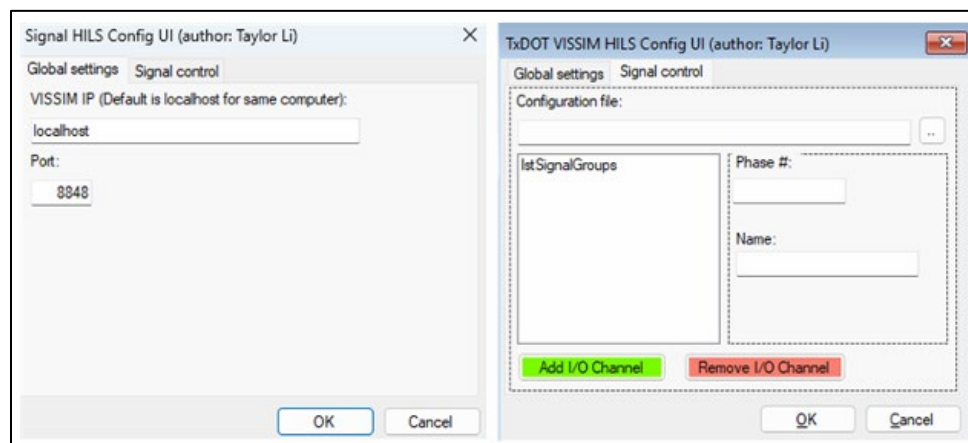


Figure 5-3: Set up IP address, port, and signal phases using STDSC_GUI.dll

5.1.3. Set up HILS signal phases and detectors in VISSIM

After configuring the controller as described in Section 5.1.2, users may place signal heads and detectors in the same manner as for any standard signal controller (e.g., an RBC). The label on each signal head (e.g., 2-4) denotes *intersection 2, phase 4*.

If a phase operates as a protected–permissive left-turn (protected green plus flashing yellow arrow), users must assign the corresponding overlap rather than the vehicle phase. For example, if Overlap A (overlap 1) provides the FYA output in the controller, then phase 201 should be selected for the associated signal head.

While VISSIM imposes no intrinsic restrictions on detector numbering, actual traffic signal controllers do. To maintain consistency across all controller brands supported by the HILS-CID, the following numbering rules must be used:

- Vehicle detector numbers: 1–64 per intersection

- Pedestrian detector numbers: 101–116 per intersection

These conventions ensure compatibility with TxDOT-approved controllers and standardized interpretation of detector actuations within the HILS-CID environment.

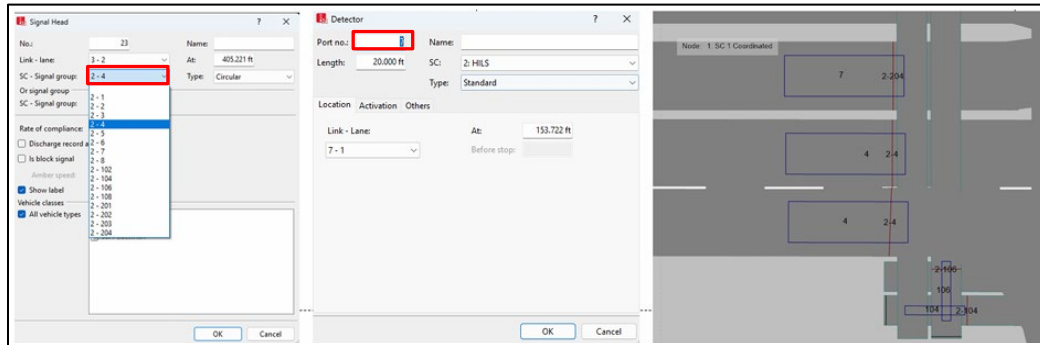


Figure 5-4: Set up signal heads and detectors in VISSIM

After configuring the signal heads and detectors, VISSIM is ready to launch the simulation. When the user clicks the Run button, VISSIM will appear to freeze, and the status bar will display the message “Initialize signal control program SC x”, where x is the ID of the first HILS intersection. This behavior is expected. It indicates that the HILS intersection is awaiting a connection from the HILS-CID tool and is prepared to begin hardware-in-the-loop communication.

5.2. Signal controller set up for HILS

Traffic signal controllers generally contain two categories of configuration:

1. System setup, which defines the number of channels, the conflict monitors unit (CMU) or malfunction management unit (MMU), and the bus interface unit (BIU) or serial interface unit (SIU) modules required to operate within a traffic signal cabinet; and
2. Timing parameters, including phases, intervals, coordination settings, and time-of-day (TOD) plans, can be used as is because they operate independently of the cabinet hardware modules disabled for simulation.

For a HILS simulation that interfaces only with the signal controller hardware, users must disable the following components:

- MMU/CMU,
- All T/F BIUs (for TS-2) or Output SIUs (for ATCC-type controllers),
- All DET BIUs (TS-2) or Input SIUs (ATCC), and
- The MMU Stop Time function.

Fig. 5-5 illustrates an example configuration using Econolite EOS. Disabling these modules prevents the signal controller from attempting to communicate with cabinet hardware that is not present in the HILS environment.

In contrast, users may configure or upload real-world traffic signal timing plans without modification. Timing parameters (e.g., phases, intervals, coordination settings, TOD plans) can be used *as is*, since they are independent of the cabinet hardware modules disabled for simulation.

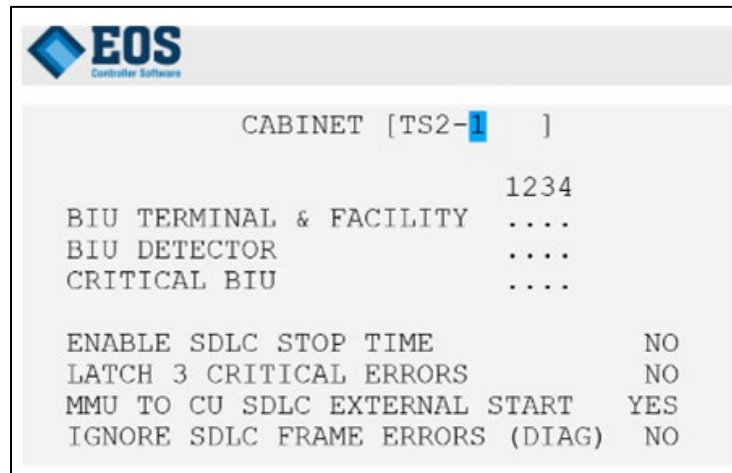


Figure 5-5: Signal controller set up for HILS

Note: If the controller is already fully configured for the traffic signal cabinet and operating normally, then you do not need to disable the modules listed above. In essence, this scenario couples the *entire* traffic signal assembly with VISSIM. This configuration is often referred to as Cabinet-in-the-Loop Simulation (CILS), a more advanced and comprehensive form of traffic signal hardware-based simulation.

5.3. HILS-CID Windows Application

The HILS-CID is a single-page software tool. When launched, users are required to enter the controller's IP address and NTCIP port, both of which can be viewed or modified on the traffic signal controller. Most controllers use port 161 by default, while Yunex SEPAC typically uses port 501.

The VISSIM connection port must match the value defined in the `STDSC_GUI.dll` configuration, as described in Section 5.1.2. This assumes that VISSIM and the HILS-CID are running on the same computer. Users must also select the controller brand from the drop-down menu. This selection is critical because each controller brand may respond differently to the NTCIP query

commands issued by the HILS-CID. For the Project 5-7160 implementation, Q-Free MAXTIME was selected as the controller brand.

The HILS-CID GUI, including the signal-controller connection, VISSIM connection, controller-brand selection, and overlap-configuration fields, is shown in Fig. 5-6.

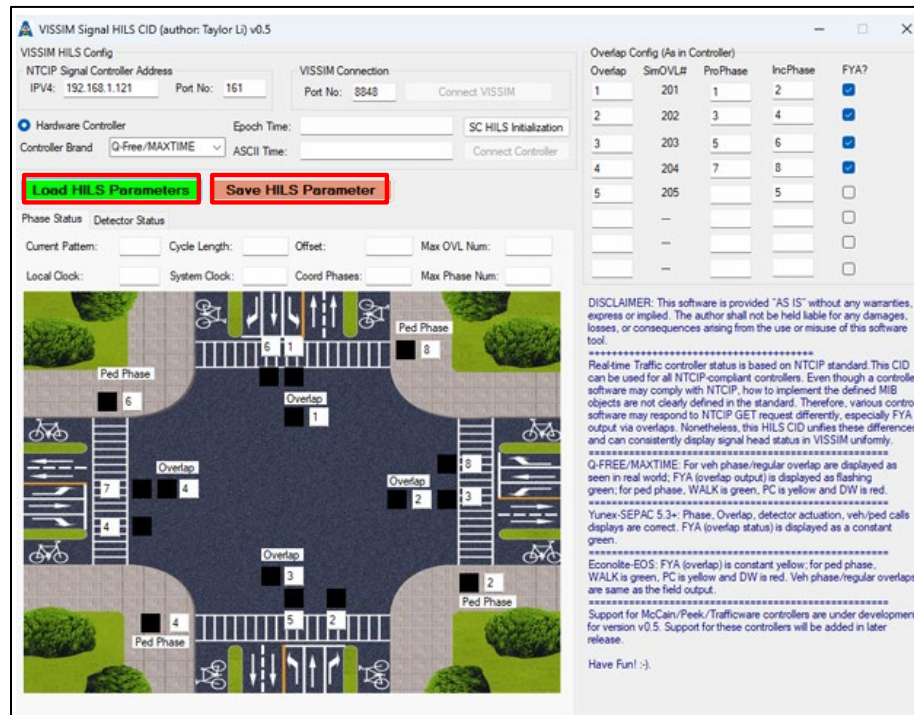


Figure 5-6: Graphic User Interface of HILS-CID

5.3.1. Overlap Configuration

If overlaps are defined in the physical traffic signal controller, they must also be specified in the HILS-CID. This configuration allows the HILS-CID to translate the controller's overlap indications into the corresponding phase identifiers used by VISSIM.

The VISSIM phase-numbering convention reserves identifiers 201–216 for overlap phases. However, the current HILS-CID Windows application supports the configuration of up to eight controller overlaps, identified as Overlap A through Overlap H. These correspond to controller overlap IDs 1–8 and VISSIM overlap phase identifiers 201–208, respectively. For example, Overlap A is entered as overlap ID 1 and is represented in VISSIM as phase 201.

For each overlap, the user must first specify the overlap ID from 1 to 8, corresponding to Overlap A through Overlap H. The HILS-CID then automatically determines the associated overlap phase number used within VISSIM.

- If an overlap represents a flashing yellow arrow (FYA) indication, select the FYA? option and enter the protected phase, which provides the green-arrow indication, and the included phase, which represents the opposing through movement.
- If the overlap is a standard overlap, enter the two parent phases. The order of the parent phases does not affect the configuration.

The overlap information entered in the HILS-CID must match the configuration stored in the physical traffic signal controller to ensure that the correct overlap indications are transmitted to VISSIM during the hardware-in-the-loop simulation.

5.3.2. Parameter Management and User Workflow

Although the initial setup requires multiple inputs, users generally need to enter them only once. After completing all required fields, click Save HILS Parameters to store the configuration in a JSON file. During future simulation sessions, the saved file can be loaded to automatically populate the configuration fields.

The graphical user interface contains the buttons required to load and save the parameters, initialize the signal controller, connect to the controller, and connect to VISSIM. When the HILS-CID is first launched, only Load HILS Parameters is enabled. This design helps ensure that users follow the required sequential workflow.

After loading the parameters, the Save HILS Parameters and SC HILS Initialization buttons become enabled. If the controller IP address and NTCIP port are correct, clicking SC HILS Initialization initiates the NTCIP handshake. A message box will then appear:

- If the message reads, *“Initialization is successful! Click ‘Connect Button’,”* the user can proceed by clicking Connect Controller, which becomes enabled.
- If an error message appears, the user should verify the controller IP address, NTCIP port, and network connectivity before attempting the initialization again.

Once communication with the physical traffic signal controller has been established, the Phase Status tab displays the signal indications received from the controller in real time.

The Phase Status tab provides visual confirmation of the vehicle-phase, pedestrian-phase, and overlap indications received from the physical Q-Free MAXTIME controller, as shown in Fig. 5-7.

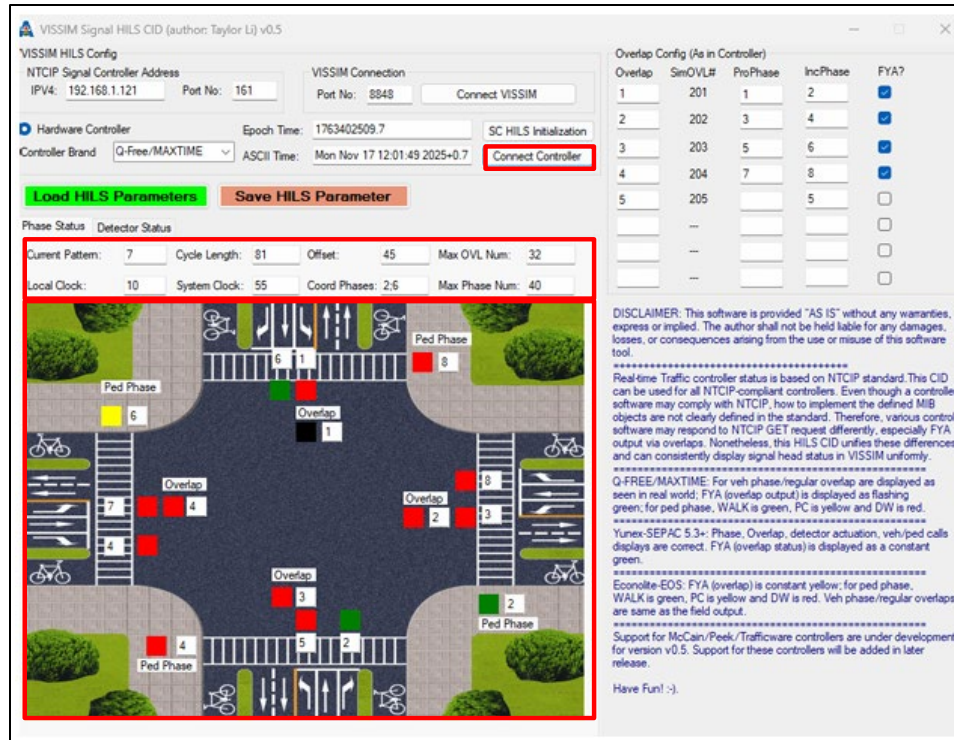


Figure 5-7: Signal Phase display

5.3.3. Connecting to VISSIM

After clicking “Connect Controller”, the final step is to connect to VISSIM. The VISSIM network must contain an active HILS intersection configured to communicate through the same port specified in the HILS-CID.

After the connection is established, the HILS-CID performs the following functions:

- Sends the real-time signal-phase status from the physical traffic signal controller to VISSIM.
- Receives vehicle-detector actuations and pedestrian calls from VISSIM and forwards them to the physical traffic signal controller.

This establishes a complete hardware-in-the-loop feedback cycle between VISSIM and the physical controller.

The Detector Status tab displays the vehicle-detector actuations and pedestrian calls exchanged between VISSIM and the physical controller, as shown in Fig. 5-8. This display allows the user to verify that the two-way HILS communication is operating correctly.

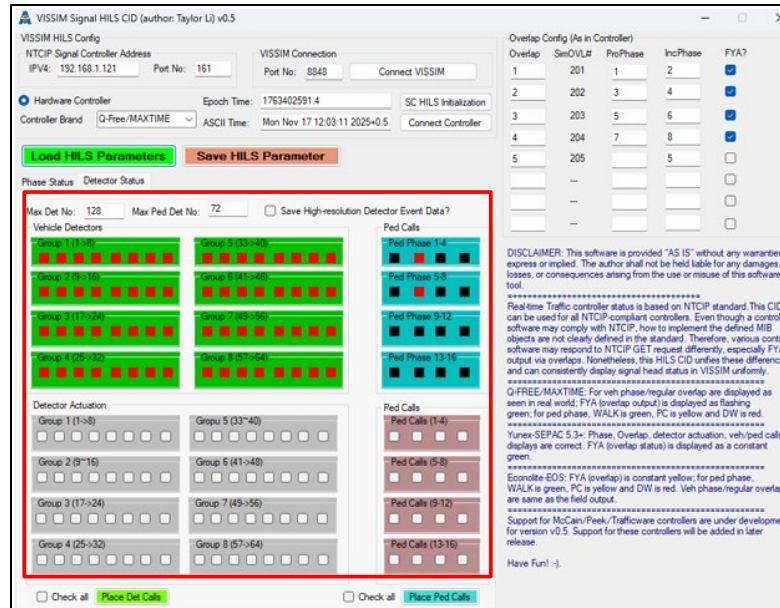


Figure 5-8: Detector and pedestrian call status in HILS-CID.

5.3.4. Summary of the Operational Sequence

The HILS-CID operational sequence is summarized as follows:

1. Load the saved HILS parameters.
2. Review or save the HILS parameters, if necessary.
3. Initialize communication with the physical signal controller.
4. Connect the HILS-CID to the physical signal controller.
5. Connect the HILS-CID to VISSIM.
6. Verify the signal-phase, pedestrian-phase, overlap, detector-actuation, and pedestrian-call displays.

Figures 5-6 through 5-8 illustrate the complete HILS-CID workflow, including the application configuration, the signal indications received from the physical controller, and the detector and pedestrian-call feedback transmitted from VISSIM.

5.4. Chapter Summary

The HILS-CID configuration integrates a physical Q-Free MAXTIME traffic signal controller with VISSIM through NTCIP-based communication. The physical controller sends real-time vehicle-phase, pedestrian-phase, and overlap indications to VISSIM, while VISSIM returns detector actuations and pedestrian calls to the controller. This two-way exchange creates a closed-loop hardware-in-the-loop simulation environment.

The configuration procedures presented in this chapter include the VISSIM external-controller setup, phase and detector numbering, physical-controller configuration, HILS-CID parameter management, overlap mapping, and connection verification. The Phase Status and Detector Status displays allow users to confirm that controller indications and simulated detector inputs are being exchanged correctly. Together, these components enable high-resolution ATSPM event generation using the operational logic of the physical Q-Free MAXTIME controller and support the case-study application presented in Chapter 6.

HILS-CID supports Q-FREE/MAXTIME, Yunex/SEPAC, McCain/Omni-Ex, Econolite/EOS. The current version of HILS-CID (v0.7 as of this report) does not support Cubic/SCOUT and Peek controllers.

Chapter 6: Case Study

6.1. Introduction

The project team conducted a case study under a controlled environment using a VISSIM simulation model, the developed software tools, and the ATSPM platform (UDOT ATSPM and UTA-In-Motion). This chapter provides guidance on configuring model outputs and performing before and after analyses. The case study was conducted along the Ranch to Market Road 2244 (RM 2244) corridor, locally known as Bee Cave Road, in the Austin District. The corridor consists of 23 signalized intersections. A baseline model representing existing 2023 conditions was first developed, and key performance measures, including travel time, approach delay, and queue length, were evaluated. Traffic volumes were then projected to represent increasing demand over a two-year period, creating the 2025 future scenario. The same performance measures were evaluated for the future condition and compared with the baseline to quantify operational changes. Finally, signal optimization was performed at selected intersections, and the resulting improvements were assessed. This chapter documents the complete workflow used to conduct the before and after analysis. Its objective is to provide users with a clear, repeatable methodology that can be applied to replicate similar studies for other corridors and scenarios.

6.2. Evaluation of the Baseline Traffic Signal Performance using the ATSPM-in-the-loop Traffic Simulation Model

The Ranch to Market Road 2244 (RM 2244) corridor, locally known as Bee Cave Road, was selected for VISSIM based digital twin development due to its role as an approximately 11.3 mile east to west arterial in Travis County, Texas, extending from SH 71 in Bee Cave to Loop 1 (MoPac Boulevard) in Austin and traversing West Lake Hills and Rollingwood. As shown in Fig. 6-1, the corridor includes major signalized intersections at SH 71, Loop 360, and Loop 1, making it well suited for corridor level ATSPM evaluation. Based on traffic volume and turning movement analysis, the 6:30 to 7:30 AM period was selected as the representative AM peak hour. The implementation focused on configuring the VISSIM simulation framework to generate ATSPM compliant data by integrating the Ring Barrier Controller (RBC), the Econolite EOS controller through a Software in the Loop Simulation (SILS) architecture, and a Hardware in the Loop Simulation (HILS) controller directly connected to the simulation environment. This configuration enabled real time interaction between the simulated traffic environment and the external signal control logic. Due to the limited availability of field data, the baseline model was partially calibrated to represent existing 2023 traffic conditions using available traffic volumes and turning movement counts. This baseline environment provides a foundation for generating high-resolution signal and trajectory data to support MOE evaluation and before-and-after operational analyses along the RM 2244 corridor.

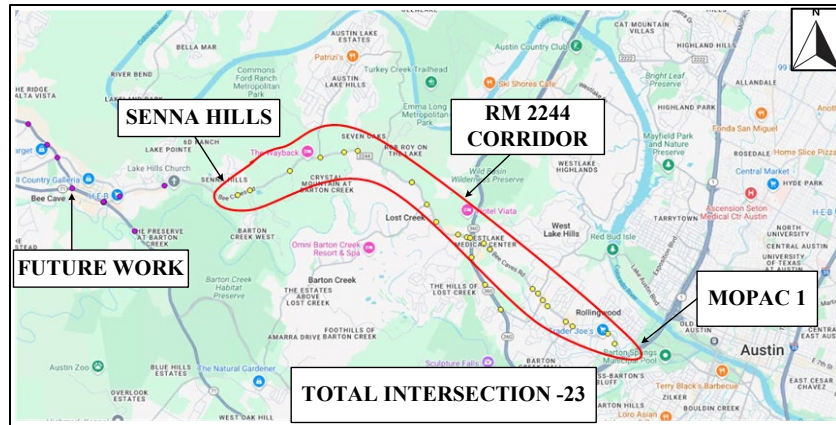


Figure 6-1: Austin RM 2244 corridor

Due to the availability of field data, only traffic volumes and turning movement counts were calibrated to match the observed 2023 conditions. Signal timing plans obtained from the local agency were also implemented in the corresponding Econolite EOS controllers for each intersection. Because traffic counts at individual intersections were collected on different dates, the observed volumes were not always consistent between adjacent intersections. To maintain realistic traffic flow throughout the corridor, vehicle volumes were balanced by strategically adding or removing vehicles within the network. An example of this volume balancing process is shown in Fig. 6-2, where additional input and output links are introduced to inject or remove vehicles based on the required traffic demand during specific time intervals. The volume balancing methodology is described in greater detail in Section 4.2.

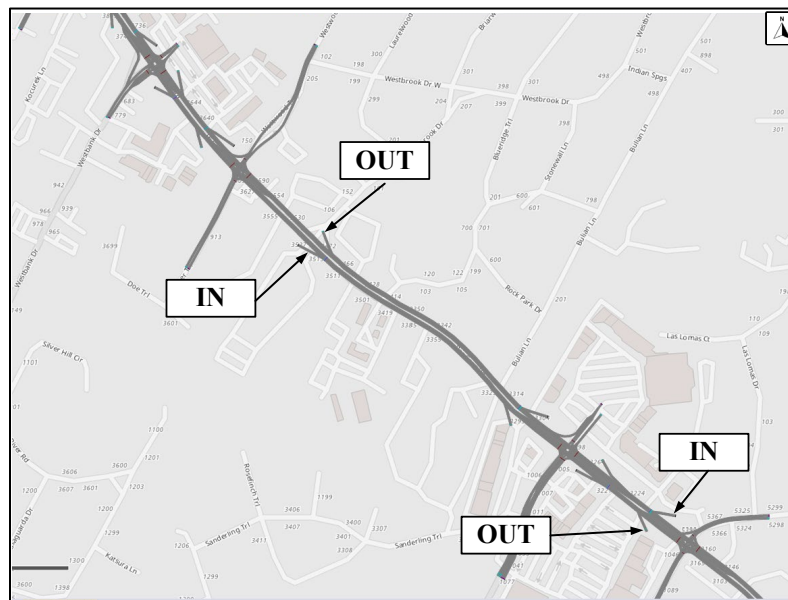


Figure 6-2: Flow balance using vehicle inputs and exits between intersections.

Because the study corridor consists of 23 signalized intersections, evaluating the performance of every intersection individually is impractical. Therefore, a corridor level approach was adopted for the overall performance evaluation. Although the analysis was summarized at the corridor level, traffic volumes and turning movement counts were calibrated at individual intersections using the available field data, and the agency-provided signal timing plans were implemented in the corresponding controllers. The corridor level approach was used solely to summarize and assess the overall operational performance of the corridor. For analysis purposes, the RM 2244 corridor was divided into three sections to facilitate the evaluation of vehicle delay and other corridor level performance measures. Zone 1 extends from RM 2244 at Senna Hills to RM 2244 at Weston Lane, Zone 2 extends from RM 2244 at Barton Creek Blvd to RM 2244 at Camp Craft Road, and Zone 3 extends from RM 2244 at Camp Craft Road to RM 2244 at Rollingwood Town Center. This segmentation was adopted to simplify the presentation of corridor performance and to provide a consistent framework for comparing operational conditions across different sections of the corridor, as shown in Fig. 6-3.

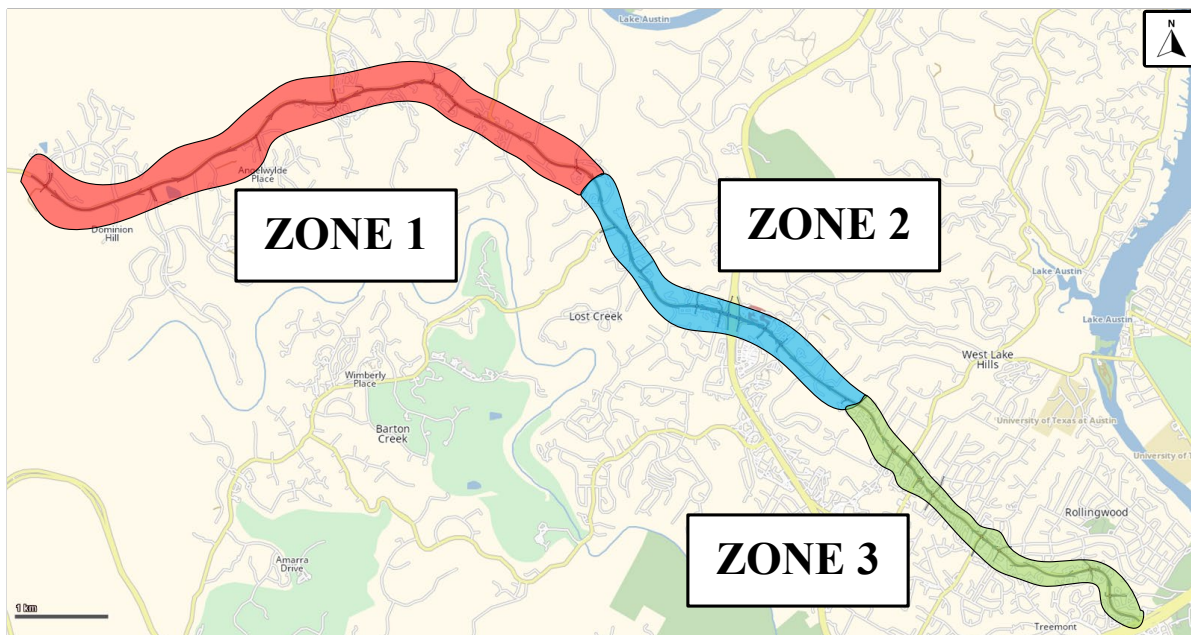


Figure 6-3: RM 2244 Zone division along the corridor

To represent each zone, one intersection was selected as a representative location from each segment for performance evaluation. The representative intersections were chosen based on relatively high traffic demand and the presence of all eight traffic movement approaches, making them suitable for assessing corridor performance. Table 6-1 lists all intersections along the corridor and identifies the representative intersections in the representative column.

Table 6-1: List of intersection with the representative intersection

S. N	Division	Signal Controller	Representative intersection
1	RM 2244 at Senna Hills	HILS	YES
2	RM 2244 at Creeks Edge Parkway	EOS	NO
3	RM 2244 at Patterson Rd	EOS	NO
4	RM 2244 at Cuernavaca Dr	EOS	NO
5	RM 2244 at Marly Way	EOS	NO
6	RM 2244 at Seven Oaks	EOS	NO
7	RM 2244 at Weston Lane	EOS	NO
8	RM 2244 at Barton Creek Blvd	EOS	YES
9	RM 2244 at Overlook at Rob Roy	EOS	NO
10	RM 2244 at Palisades/Dimensional Pl	EOS	NO
11	RM 2244 at Castle Ridge Road	EOS	NO
12	RM 2244 at Loop 360 (Diamond)	EOS	NO
13	RM 2244 at Village at Westlake	EOS	NO
14	RM 2244 at Red Bud	EOS	NO
15	RM 2244 at the Hills	EOS	NO
16	RM 2244 at Camp Craft Rd	EOS	NO
17	RM 2244 at Westlake Dr	EOS	NO
18	RM 2244 at Westbank Dr	EOS	NO
19	RM 2244 at Beaver Trail	EOS	NO
20	RM 2244 at Walsh Tarlton Ln	EOS	NO
21	RM 2244 at Rollingwood Dr	EOS	NO
22	RM 2244 at Edgegrove Dr	EOS	NO
23	RM 2244 at Rollingwood Town Ctr	RBC	YES

After evaluating all 23 intersections, one representative intersection was selected from each corridor zone: RM 2244 at Senna Hills for Zone 1, RM 2244 at Barton Creek Blvd for Zone 2, and RM 2244 at Rollingwood Town Ctr for Zone 3. These intersections were selected based on their relatively high traffic demand. Since traffic operations along the corridor are highly interconnected, evaluating representative intersections distributed throughout the corridor provides an effective means of validating the overall operational performance while avoiding the need to analyze every intersection individually. Accordingly, these representative intersections served as checkpoints for validating the simulation results. Among the three representative intersections, RM 2244 at Senna Hills was controlled using a HILS controller, RM 2244 at Rollingwood Town Center was operated using an RBC, while the remaining representative intersection, along with all other intersections in the corridor, was controlled through the Econolite EOS simulation environment. This configuration demonstrates the integration of different controller types within the same simulation framework and provides an example implementation of both approaches. Once the simulation environment was fully configured, the VISSIM model was executed for the 6:30 to 7:30 AM analysis period. To ensure that only stable traffic conditions were included in the analysis, a 15-minute buffer was added before and after

the analysis period. The initial buffer allowed vehicles to enter and circulate throughout the network before data collection began, while the ending buffer provided sufficient time for vehicles to exit the network. This approach minimizes startup and termination effects and ensures that the collected performance measures accurately represent steady state traffic conditions.

6.2.1. Setting Up HILS and SILS in VISSIM

Once the VISSIM simulation model was developed and the traffic volumes and turning movement counts were incorporated, the signal controllers were configured accordingly. The detailed signal timing plans were obtained from the Austin District. These plans were carefully reviewed to extract essential information, including signal timing parameters, timing patterns, coordination plans, detector layouts, and overlap settings. The extracted information was then implemented in VISSIM. Since the objective of this study was to evaluate different controller types, three different signal controller configurations were implemented within the simulation model. The first controller was the standard RBC, which is the default controller available in VISSIM and therefore required minimal additional configuration. Only the necessary signal timing and operational parameters were entered into the controller. Fig. 6-4 illustrates the configured RBC controller with the corresponding input parameters.

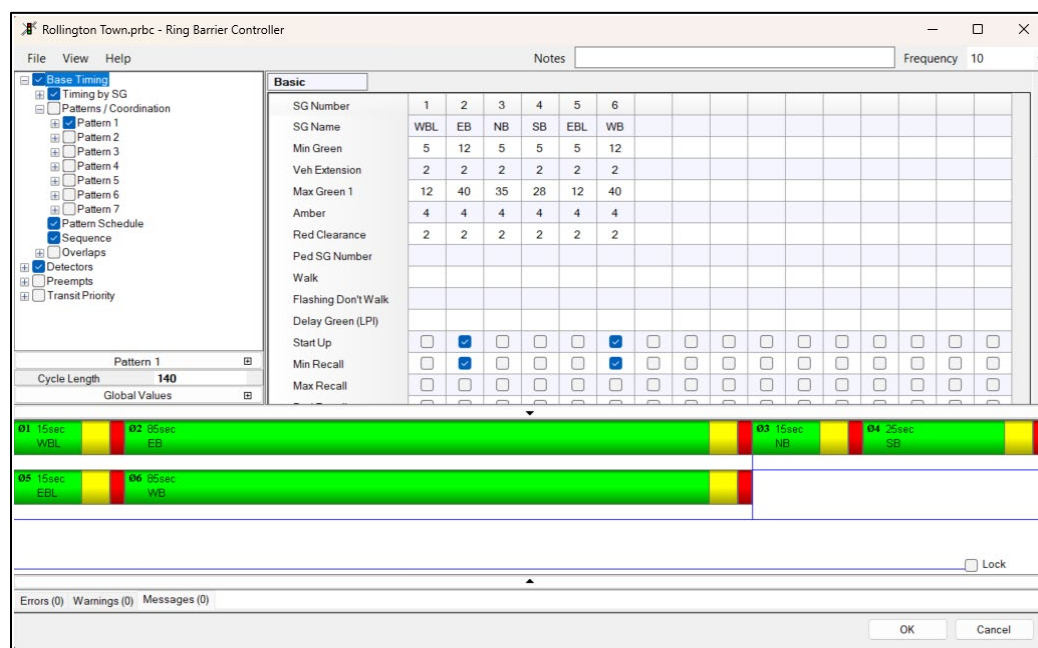


Figure 6-4: RBC controller with the corresponding input parameters

The second controller type implemented in this study was the SILS controller using the EOS. EOS is an advanced traffic signal controller emulator that is fully supported by VISSIM under the standard license. The detailed procedure for configuring the EOS controller is provided in Section 4.3 and should be referred to for implementation details. After the signal timing plans

were configured in the Econolite EOS and the VISSIM simulation was started, the emulator's front panel could be accessed directly within the VISSIM interface. This feature allows users to monitor controller operation and verify signal status during simulation. Fig. 6-5 illustrates the front panel of the Econolite EOS controller as displayed in VISSIM.

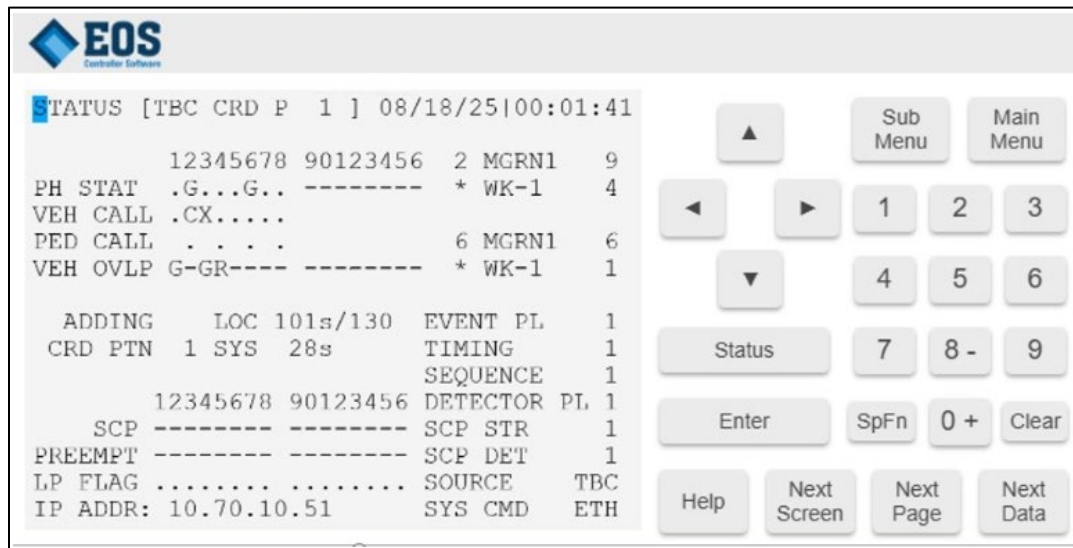


Figure 6-5: Econolite EOS web front panel showing controller status.

One of the primary advantages of the EOS controller is its ability to directly generate the controller event data required for the ATSPM system and store it in the **.DataZ** file format. As a result, the controller events can be readily extracted through post processing of these files. To automate this process, a parsing program was developed that reads the .DataZ files and imports the controller event data into the database. Once the simulation has been completed, the user only needs to execute the parsing program to process the generated files. Fig. 6-6 presents the interface of the parsing software. The parsing process should be performed only after the simulation has finished, as the .DataZ files are finalized only upon completion of the simulation in VISSIM.

VISSIM To UDOT-ATSPM Parser: By Taylor Li (UT Arlington), v2.1.0, Jul-2026

Step 1: General DB Configurations

UDOT ATSPM DB IP: 10.33.1.60 UDOT ATSPM DB port: 1434

UDOT ATSPM DB Name: MOE Simulated DB name: VISSIM

User ID: ** Password: *****

Simulated Date: 8/23/2026 Simulated Starting time: 06:30:00

Step 1: Load Config **Save Config**

Step 2: Parsing SimSigOutputs to UDOT ATSPM

RBC Events Q-FREE MAXTIME Event Logging (On the fly) **ECON-EOS Event Logging(Posterior)**

Attention: EOS Hi-Res data must be processed after simulation

Parent Directory of datZ files: F:\143 Dive H\SPM IMPLEMENTATION Select

Step 2: Uploading ATSPM Logs to UDOT-ATSPM

Step 3: Parsing sim v_traj to UTAIM

UTAIM DB IP: 10.33.1.60 DB Port No: 3308 Sim Anon CV type: 30520

User ID: **** Password: ***** Sim Special CV type:

Step 3: Generate CV Trajectory for UTAIM

<https://actionlab.uta.edu/>

Figure 6-6: VISSIM to UDOT ATSPM parser interface for post-processing EOS event data.

Finally, the HILS controller was configured. Unlike the RBC and EOS controllers, VISSIM does not natively communicate with external hardware controllers. Therefore, the HILS configuration requires additional setup to establish communication between the simulation model and the physical traffic signal controller. The complete installation and configuration procedure is provided in Sections 5.1 through 5.3 and should be followed carefully to ensure proper communication and synchronization between VISSIM and the external controller. Once the HILS connection is established, the signal timing parameters can be configured either through the controller's front panel or its web-based management portal. Fig. 6-7 illustrates (a) the front panel of the controller, (b) the web-based configuration portal, and the VISSIM simulation model operating simultaneously with the external controller.

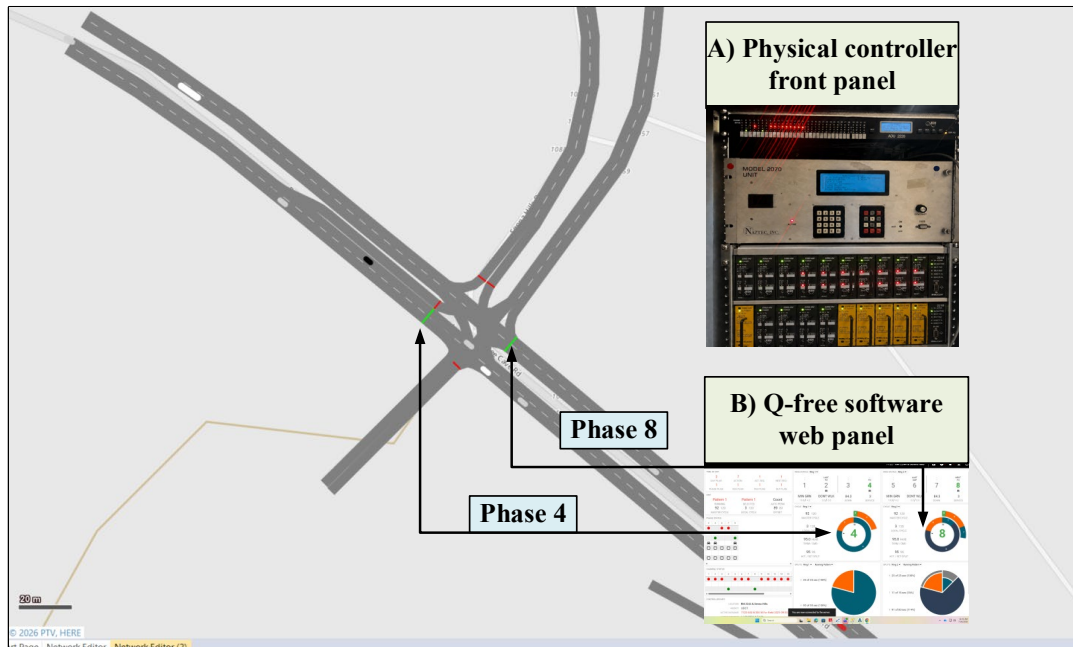


Figure 6-7: HILS operation using the physical controller front panel, web portal, and VISSIM model.

An additional software application, referred to as the HILS Interface, was also developed to establish communication between VISSIM and external traffic signal controllers. The application allows users to specify the controller type and IP address, after which it initializes the communication and synchronizes the hardware controller with the VISSIM simulation. Once connected, both the controller and the simulation operate simultaneously, enabling real time hardware in the loop testing. Although the interface supports communication with multiple controller types, the MAXTIME controller was used for this study. Fig. 6-8 illustrates the HILS Interface software. The HILS framework provides a realistic testing environment by allowing actual controller hardware to operate with simulated traffic conditions. This enables the evaluation of controller performance and the collection of authentic controller event data under repeatable and controlled traffic scenarios, making it particularly valuable for ATSPM research and controller validation.

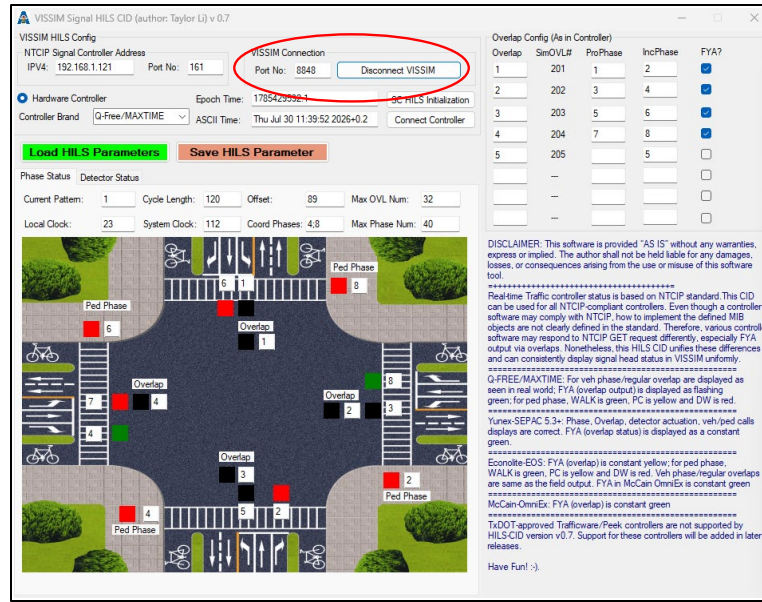


Figure 6-8: HILS Interface for connecting the MAXTIME controller to VISSIM.

A key limitation of the HILS controller, compared with the Econolite EOS emulator, is that it does not automatically generate a controller event log file after the simulation. Instead, controller events are produced in real time while the controller is operating. Consequently, these events must be captured continuously during the simulation. To address this limitation, a custom event parsing application was developed to monitor controller communication and extract controller events in real time, storing them directly into the database for subsequent ATSPM analysis. Fig. 6-9 presents the event parsing software used to capture controller events during simulation.

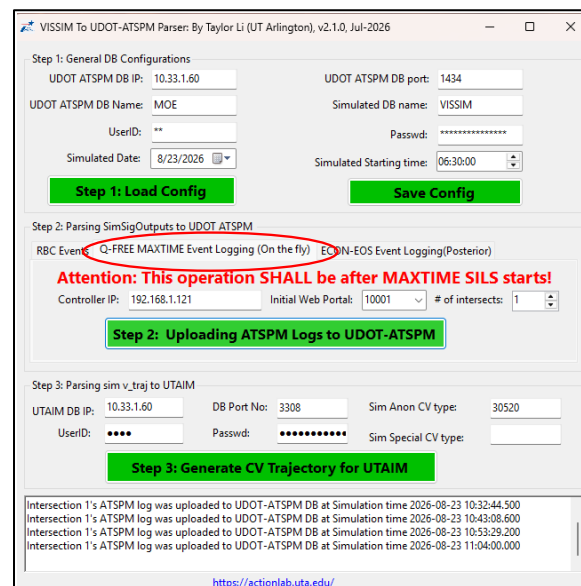


Figure 6-9: Real-time event parsing interface for capturing HILS controller events during simulation.

6.2.2. Analysis of Performance Metrics from VISSIM

The VISSIM simulation was configured using a standard set of simulation parameters to ensure consistent and reproducible results. The simulation date was set to August 23, 2026, with a simulation resolution of 10-time steps per simulation second. A random seed of 42 was used, and only one simulation run was performed. The simulation speed was set to Maximum, and all available processor cores were utilized to minimize computational time. Although VISSIM allows the random seed to be changed to generate different stochastic traffic conditions across multiple simulation runs, this study focused on evaluating a single baseline scenario. Therefore, the default random seed value of 42 was retained to ensure that the simulation results are reproducible and can be consistently replicated in future analyses.

Node evaluation areas were created using the Node object in VISSIM. The Node object allows users to define polygons of any shape for collecting intersection level performance measures. Since most intersections along the RM 2244 corridor are four leg intersections, rectangular node polygons were created around each intersection to capture their operational performance. These node evaluation areas were used to extract traffic volumes, average delay, queue length, and Level of Service (LOS). In cases where intersections had unconventional layouts, the node polygons were adjusted accordingly to ensure that all relevant traffic movements were included within the evaluation area. Although all approaches at every intersection were calibrated and evaluated, only the eastbound through (EBT) and westbound through (WBT) movements were used for the corridor level Measures of Effectiveness (MOE) evaluation. These movements represent the primary traffic flow along the RM 2244 corridor and therefore provide the most appropriate basis for assessing overall corridor performance. Focusing on the EBT and WBT approaches ensures that the evaluation reflects the operational conditions experienced by the major through traffic traveling along the corridor. Table 6-2 summarizes the performance measures extracted from the VISSIM simulation for each corridor zone.

Table 6-2: Results obtained from VISSIM.

Intersection	ZONE 1		ZONE 2		ZONE 3	
Movement	EBT	WBT	EBT	WBT	NBT	SBT
Average Queue Length	4.6	9.8	273	3.5	2	150
Average Vehicle Delay	5.8	9.1	37	7	4	84
LOS	A		C		C	

6.2.3. Comparison of Measure of Effectiveness (MOEs) from VISSIM and UDOT ATSPM

Two ATSPM platforms were evaluated in this study: UDOT ATSPM and UTA In Motion (UTAIM). Both systems generate a variety of signal performance visualizations; however, they require the input data to be formatted according to the ATSPM data structure. The native output generated by VISSIM is not directly compatible with either platform. Therefore, an intermediate data processing step was required before the simulation data could be analyzed. During the simulation, VISSIM output data were continuously extracted at 15-minute intervals and written to a database. The extracted data was then parsed into the ATSPM data format using a software tool developed by the UTA Action Lab. Additional details regarding the software architecture and data processing workflow are provided in Section 4.3. Once the data were successfully parsed, they were compatible with both the UDOT ATSPM and UTAIM platforms for visualization and performance analysis. Although UDOT ATSPM provides a comprehensive suite of performance charts, not all of these visualizations can be generated from VISSIM because certain field controller events are not available within the simulation environment. Nevertheless, several key performance metrics can be evaluated, including approach volumes, phase operations, cycle lengths, delays, and signal timing characteristics. For validation purposes, the total approach volumes reported by the ATSPM system were also compared with the corresponding VISSIM outputs. Table 6-3 lists the ATSPM performance charts generated using the UDOT ATSPM platform. These performance measures are generally displayed with time (hours and minutes of the day) on the x axis and variables such as traffic volume, gap-out events, cycle length, phase duration, and signal indications on the y axis. The following sections provide a brief description of each performance metric and its interpretation:

Table 6-3: Performance Metrics obtained from UDOT-ATSPM.

SN	Performance Metrics	X-axis	Y-axis	Output
1	Approach Delay	Time (Hours/minutes)	Delay per Vehicle (second)	Approach delay per vehicle, Total delay per vehicle
2	Approach Volume	Time (Hours/minutes)	Volume (Vehicles per hour)	Northbound and Southbound approaches, D-factor
3	Arrivals on red	Time (Hours/minutes)	Volume (Vehicles per hour)	AoR, Percent arrivals on Red, Total vehicles
4	PCD	Time (Hours/minutes)	Cycle time (seconds)	AoG, GT, and PR

Before relying on the performance charts generated by the UDOT ATSPM platform, the outputs were first validated against the corresponding results obtained directly from VISSIM. Since the VISSIM model served as the source of all simulation data and represented the calibrated baseline scenario, the performance measures extracted directly from VISSIM were treated as the reference for validation. The simulation data were first parsed from VISSIM into the UDOT ATSPM format using the developed parsing software and were subsequently transferred to UTAIM platform.

Because both ATSPM platforms use the same parsed dataset, validating the UDOT ATSPM outputs also validates the data used by UTAIM. Not all ATSPM performance charts could be validated. Both UDOT ATSPM and UTAIM require controller event data that are typically generated by field signal controllers but are not directly available from VISSIM. For example, VISSIM does not produce detailed controller event logs needed to generate several ATSPM visualizations. As a result, the validation was limited to performance measures that could be derived directly from the simulation outputs, namely approach volume and approach delay. To perform the validation, the required data were extracted directly from VISSIM and plotted in Microsoft Excel. These graphs were then compared with the corresponding charts generated by the UDOT ATSPM platform. Fig. 6-10 presents the comparison of approach volumes obtained from VISSIM and UDOT ATSPM for intersection at RM 2244 at Rollingwood Town. As shown in the figure, both datasets exhibit nearly identical temporal trends, demonstrating that the parsing process accurately transfers VISSIM data into the UDOT ATSPM framework and validating the approach volume results generated by the ATSPM system.

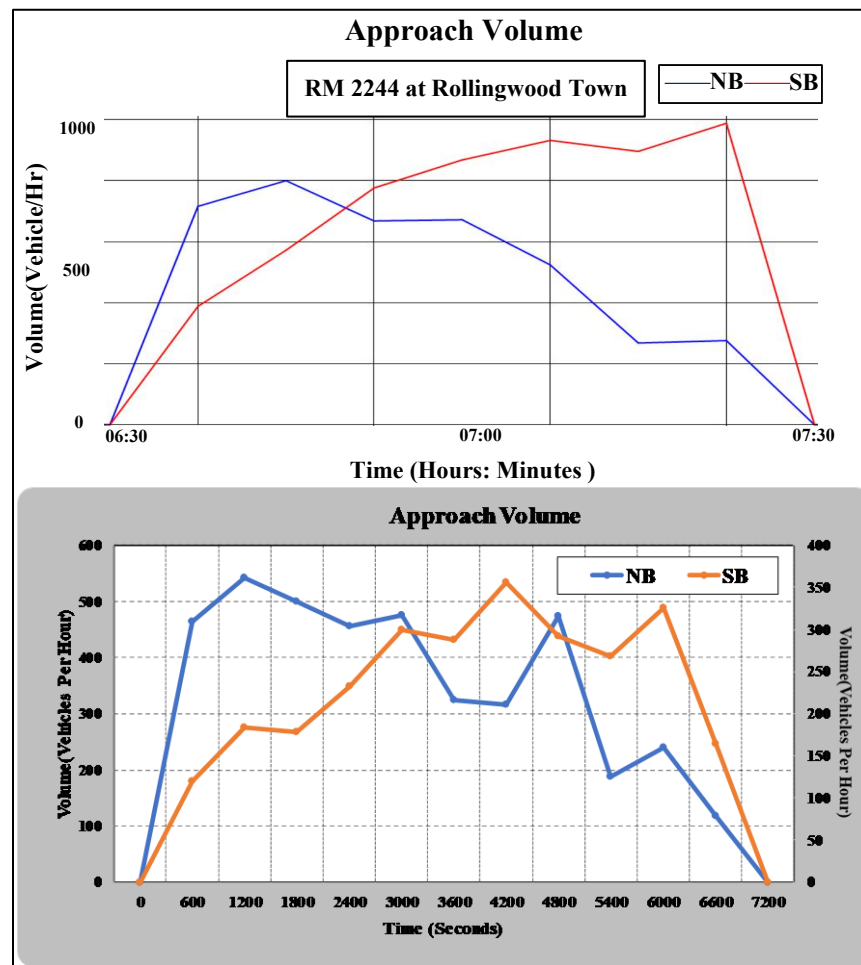


Figure 6-10: Approach volume for RM 2244 at Rollingwood Town from UDOT ATSPM (A) and VISSIM (B).

Some differences are observed when comparing approach delay between VISSIM and UDOT ATSPM. These differences are expected because UDOT ATSPM estimates delay using detector actuation data and does not explicitly account for startup delay, vehicle deceleration, or queues extending beyond the detector coverage area, whereas VISSIM directly simulates these traffic behaviors. As a result, UDOT ATSPM generally reports lower approach delay values than VISSIM, even though both systems exhibit similar temporal trends. Fig. 6-11 illustrates this relationship, where the delay profiles follow comparable patterns despite differences in the reported delay values. It should also be noted that UDOT ATSPM provides several additional performance charts that cannot be reproduced using VISSIM output alone because the required controller event data are not available within the simulation environment. Therefore, the validation presented in this study focuses primarily on approach volume and approach delay, which are the performance measures that can be directly compared between the two systems.

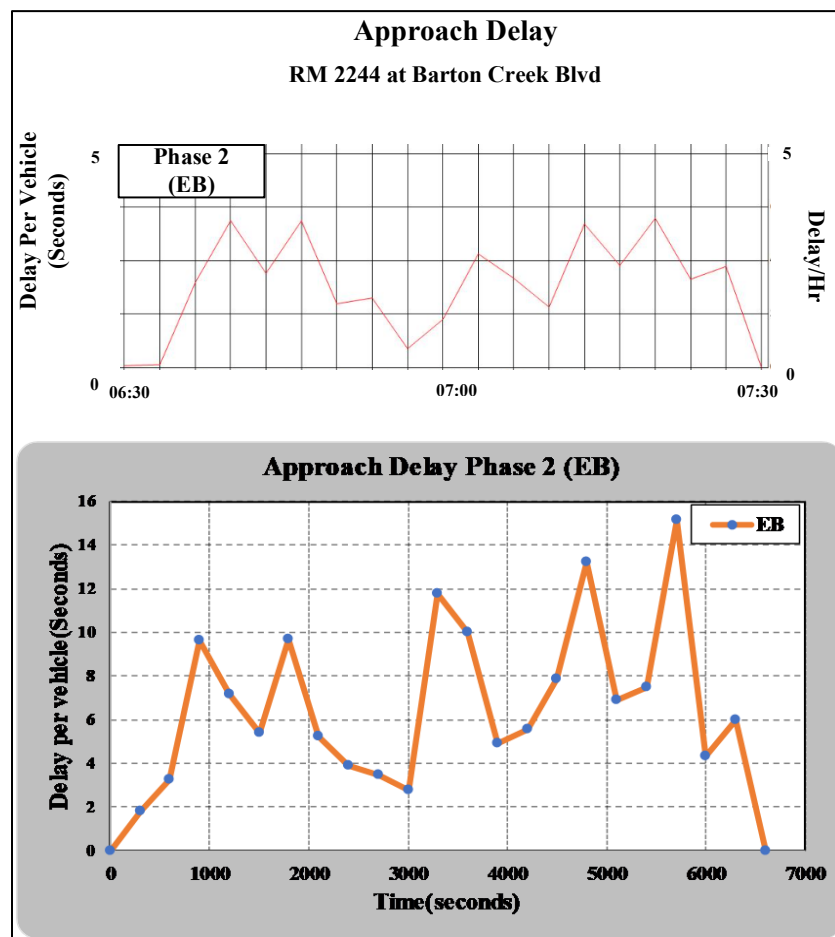


Figure 6-11: Approach delay for RM 2244 at Barton Creek Blvd from UDOT ATSPM (A) and VISSIM (B).

6.2.4. Analysis of performance metrics from UDOT-ATSPM

As discussed in Table 6-3, only a subset of the MOEs available in the UDOT ATSPM system was used in this study. Not all performance charts were applicable to the simulation-based analysis, as

some require controller event data that are not generated by VISSIM, while others were incomplete due to insufficient supporting data. Consequently, the evaluation focused on four key performance measures: Approach Volume, Arrivals on Red (AOR), Approach Delay, and Purdue Coordination Diagram (PCD).

For demonstration purposes, Zone 3 was selected to illustrate the ATSPM outputs because it is representative of the overall corridor operations. The complete set of UDOT ATSPM performance charts generated for the study is presented in Table 6-4. As shown in the table, these visualizations provide valuable insights into traffic demand, signal progression, vehicle delay, and corridor coordination, enabling a comprehensive assessment of signalized intersection performance. Zone 3 includes the intersection controlled by the RBC. The two primary approaches are the northbound and southbound movements. As shown in Table 6-4, the average approach delay for both directions was approximately 1 second. The total delay was 0.2 hours for the northbound approach and 0.5 hours for the southbound approach, indicating that the southbound movement experienced greater overall delay. This difference may be attributed to factors such as traffic demand, signal timing, or the operational characteristics of the roadway. However, because the simulation model was not fully calibrated due to limited field data, these values should be interpreted as demonstration results rather than validated operational measures. They serve as the baseline scenario and will be compared with the optimized or future scenario in subsequent analyses. The recorded approach volumes were 981 vehicles for the northbound approach and 1,355 vehicles for the southbound approach. The southbound approach carried a higher traffic volume and experienced greater total delay than the northbound approach during the analysis period. The AoR percentage was 11% for the northbound approach and 10% for the southbound approach. Since this is the final intersection in the corridor, these values are reasonable because a larger proportion of vehicles have already progressed through the coordinated signal system before reaching this location. Similarly, the Arrival on Green (AoG) percentages were comparable between the two approaches, with 85% for the northbound movement and 88% for the southbound movement. The green time percentages were 85% and 86% for the northbound and southbound approaches, respectively. Finally, the platoon ratios were 1.00 for the northbound approach and 1.02 for the southbound approach, indicating similar levels of platoon progression in both directions.

Table 6-4: Performance Metrics for the Chosen Intersection.

Intersection	Zone 1		Zone 2		Zone 3	
Performance Metrics	Phase 2 (EB)	Phase 6 (WB)	Phase 2 (EB)	Phase 6 (WB)	Phase 6 (NB)	Phase 2 (SB)
Approach Delay (sec)	3	0.2	6	1	1	1
Total Delay (hr.)	2	2	4.8	0.5	0.2	0.5
Approach Volume	2188	365	2805	2233	981	1355
Arrivals on Red (AoR-%)	33	32	27	14	11	10
Arrivals on Green (AoG-%)	67	68	75	73	85	88

Intersection	Zone 1		Zone 2		Zone 3	
Green Time (GT- %)	66	69	63.5	64	85	86
Platoon Ratio (PR)	1.02	0.99	1.4	0.99	1.0	1.02

Using data obtained from the UDOT ATSPM system, approach volume graphs were generated for the study intersection. One of the key advantages of these graphs is that they provide not only traffic volume but also several additional performance measures, including the Peak Hour Factor (PHF), directional distribution factor (D factor), and K factor. Fig. 6-12 presents the approach volumes for the northbound and southbound approaches, each represented by a different color. As shown in the figure, the southbound approach consistently carries a higher traffic volume than the northbound approach, as indicated by the higher red curve compared with the blue curve. It is important to note that the y-axis represents the equivalent hourly traffic volume rather than the actual vehicle counts observed during each time interval. Therefore, additional calculations are required to determine the actual traffic volume for a specific analysis period. The supplementary statistics provided by the UDOT ATSPM system offer further insights into traffic characteristics. The peak hour occurred at different times for the two approaches, with the northbound approach reaching its peak between 6:30 AM and 7:30 AM, while the southbound approach peaked between 7:15 AM and 8:15 AM. For the entire intersection, the Peak Hour Factor (PHF) was 0.944, the K factor was 0.622, the peak hour volume was 1,453 vehicles, and the total recorded volume was 2,336 vehicles. The UDOT ATSPM system also reports approach specific traffic characteristics. The K factor was 0.584 for the northbound approach and 0.500 for the southbound approach. The directional distribution (D) factor was 0.523 for the northbound approach and 0.679 for the southbound approach, confirming that the southbound movement carried a larger share of the total traffic demand. These traffic measures are fundamental for traffic engineering analyses, as they support signal timing optimization, intersection design, and operational evaluation. Consequently, the approach volume graph, together with the associated ATSPM performance measures, provides a comprehensive understanding of traffic demand and directional characteristics at the study intersection.

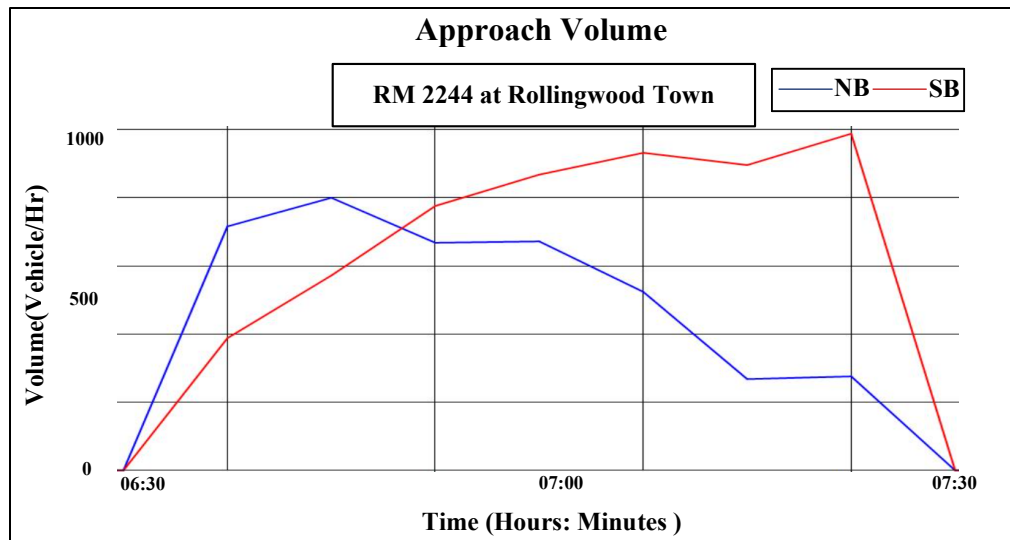


Figure 6-12: Approach volume for RM 2244 at Rollingwood Town.

Similarly, using the UDOT ATSPM system, approach delay graphs were generated for both the northbound and southbound approaches. The system provides both total approach delay and average approach delay per vehicle. As shown in Fig. 6-13, the southbound approach experiences higher total delay than the northbound approach. The data presented correspond to the period from 6:30 AM to 7:30 AM. During this interval, the northbound approach recorded a total delay of approximately 0.2 hours, whereas the southbound approach recorded a total delay of about 0.5 hours. The average approach delay per vehicle was approximately 1 second for both approaches. The higher southbound total delay is consistent with the greater southbound traffic volume during the analysis period. This suggests that factors other than traffic volume influenced the observed delay. Possible reasons include more favorable signal timing, such as longer effective green time or better signal coordination, as well as differences in vehicle arrival patterns, turning movement demand, or queue dissipation between the two approaches. These factors can reduce delays even under higher traffic volumes.

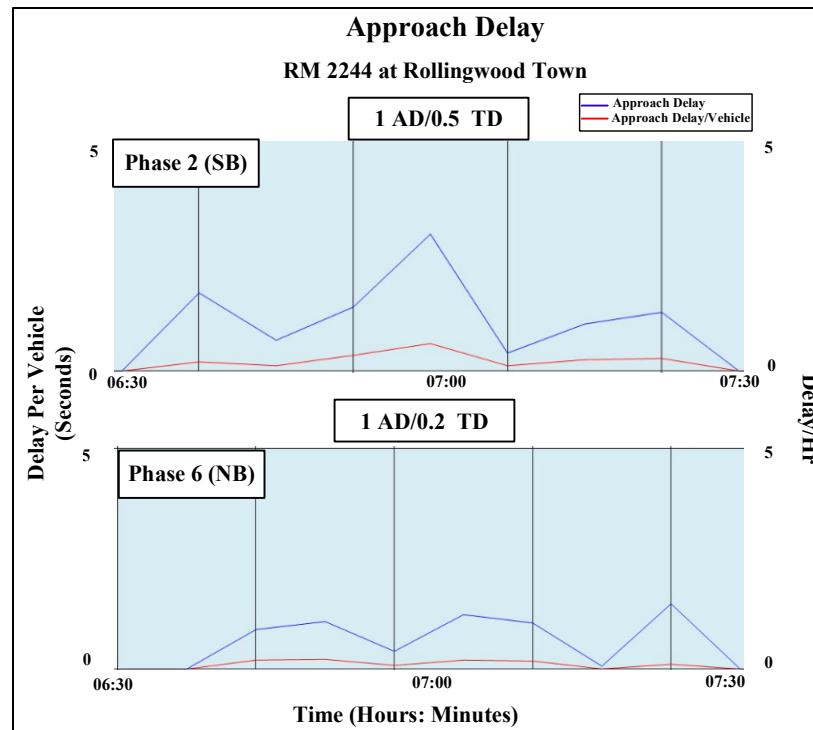


Figure 6-13: Approach delay for SB (Phase 2) and NB (Phase 6) for RM 2244 at Rollingwood Town.

AoR is an important performance measure used to evaluate signal coordination along a corridor. It represents the percentage of vehicles that arrive at an intersection during the red phase. A lower AoR indicates better signal progression, allowing more vehicles to pass through the intersection without stopping, whereas a higher AoR suggests poorer coordination and increased delay. Therefore, the objective of coordinated signal operation is to minimize AoR and maximize arrivals on green. Similar to the UDOT ATSPM system, AoR graphs were generated for both the northbound and southbound approaches at RM 2244 and Rollingwood Town Center, as shown in Fig. 6-14. The northbound approach recorded an AoR of approximately 11%, while the southbound approach recorded an AoR of about 10%. Although the southbound approach carried a higher traffic volume during the study period, both approaches exhibited nearly identical AoR percentages, indicating comparable signal progression in both directions. The relatively low AoR values suggest that nearly 90% of vehicles arrived during the green interval, reflecting satisfactory coordination of the signal timings. Minor fluctuations in AoR throughout the analysis period are expected because of variations in traffic demand and vehicle arrival patterns. These results represent the baseline scenario for the corridor. In the later sections, the AoR values will be compared with those obtained under the future traffic demand scenario. As traffic volumes increase while the signal timing plans remain unchanged, a higher percentage of vehicles are expected to arrive during the red phase, resulting in increased AoR values. This hypothesis will be evaluated by comparing the baseline and future scenarios.

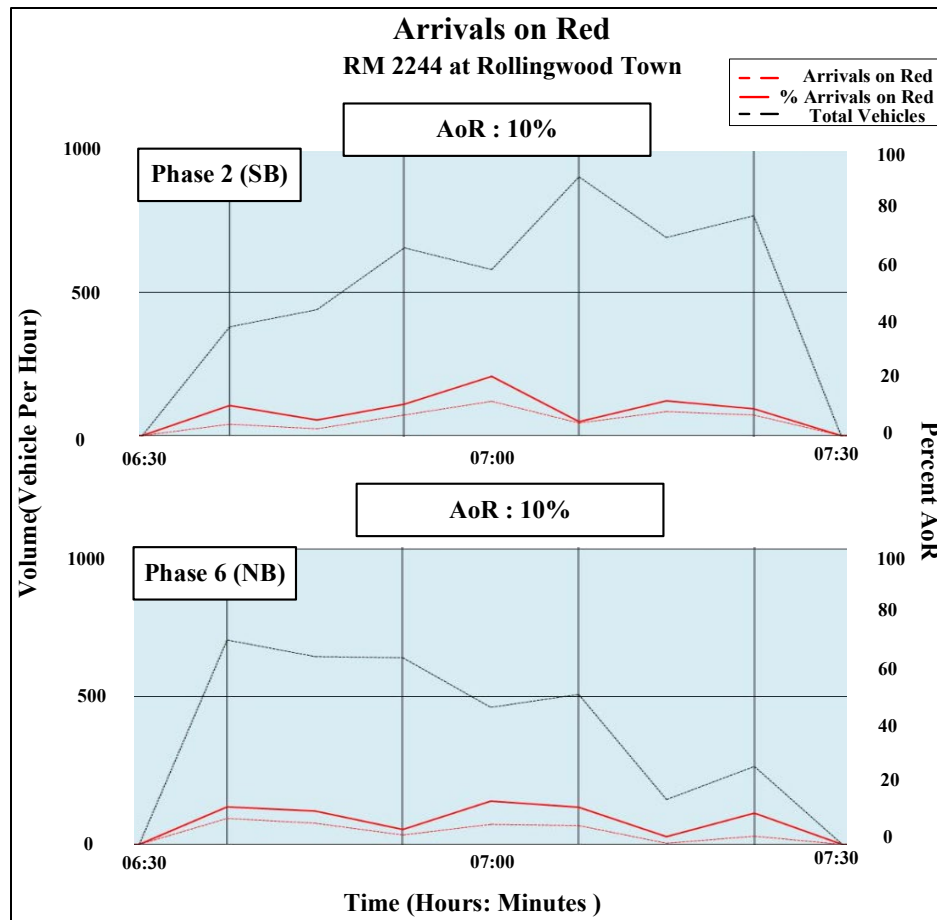


Figure 6-14: Arrival on Red for SB (Phase 2) and NB (Phase 6) for RM 2244 at Rollingwood Town.

Another graph generated using the UDOT ATSPM system is the PCD. The PCD is an important tool for evaluating signal coordination because it illustrates the relationship between vehicle arrivals and signal timing. It provides information on vehicle arrival times, cycle length, green intervals, AoG, and the time of day. An example PCD for the reference intersection in Zone 3 is shown in Fig. 6-15.

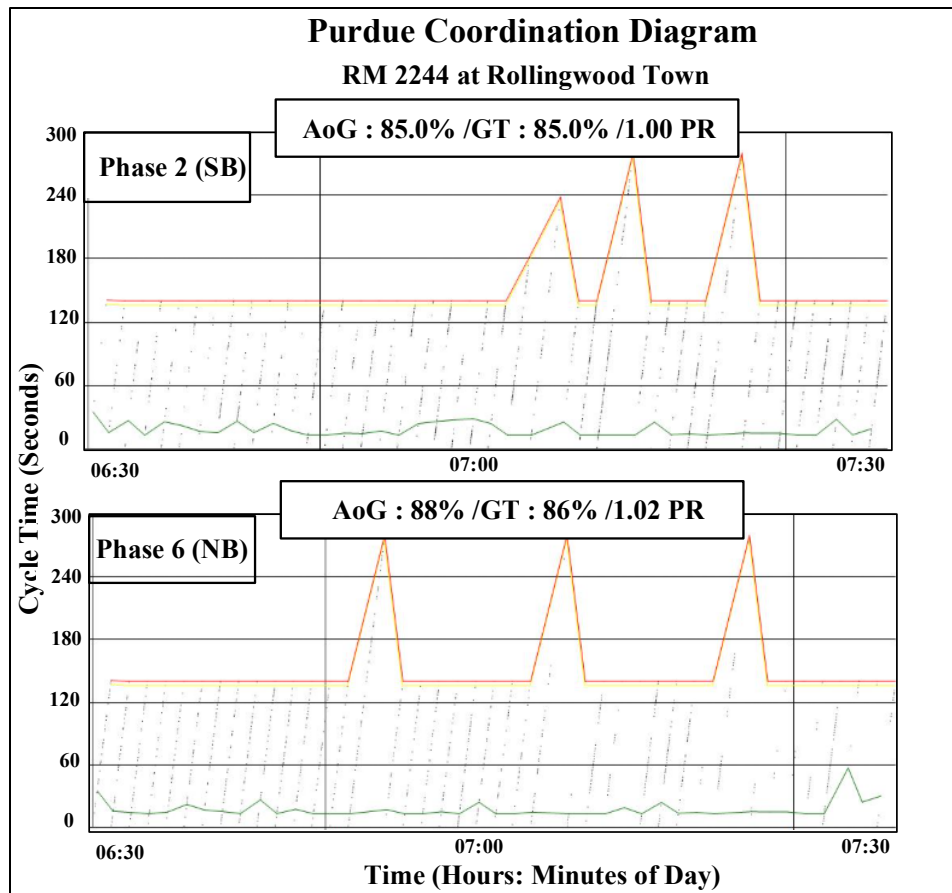


Figure 6-15: PCD for SB (Phase 2) and NB (Phase 6) for RM 2244 at Rollingwood Town.

Based on Fig. 6-15, the signal cycle length is approximately 120 seconds under normal operating conditions. However, occasional spikes in the cycle length can be observed. These occur when no vehicles are detected on the competing phases. The figure represents Phase 6, which serves the southbound through movement. Since this phase operates with minimum recall, it continues to receive green until a vehicle is detected on another phase. This behavior was also confirmed through the VISSIM simulation. Around 7:00 AM, no vehicles were detected on the side street approaches, allowing the green interval for Phase 6 to continue and increasing the cycle length to nearly 250 seconds. The PCD also indicates that approximately 86% of the cycle was green time, while 88% of vehicles arrived during the green interval. These values suggest that the intersection operates efficiently and that most vehicles are able to pass through the intersection without stopping. In addition, the platoon ratio was 1.02. The platoon ratio compares the proportion of vehicles arriving during the green interval with the proportion of green time available. A value close to 1.0 indicates random vehicle arrivals, while values greater than 1.0 indicate that vehicle arrivals are well coordinated with the green interval. Therefore, the observed platoon ratio of 1.02 suggests that the signal progression is slightly better than random, indicating satisfactory coordination along the corridor. These results represent the baseline operating condition of the corridor. In the future scenario, traffic volumes are expected to increase while the signal timing

remains unchanged. Consequently, lower arrivals on green, a reduced platoon ratio, and increased stops and delays are expected. These changes will be evaluated and compared with the baseline scenario in a later section.

To improve the readability of this case study, detailed figures and discussions are presented only for reference intersection in Zone 3. The corresponding figures for the representative intersections in Zone 1 and Zone 2 are provided in the Appendix. These intersections exhibited similar trends and performance characteristics, and therefore only the reference intersection is discussed in detail to avoid repetition.

6.2.5. Analysis of performance metrics from UTAIM

Similar to the UDOT ATSPM system, the UTAIM platform also requires simulation data to be parsed into a standardized format before analysis. Dedicated software was developed to convert the VISSIM output into a format compatible with the UTAIM system. Since UTAIM utilizes the same parsed event data stored in the validated UDOT database, and the UDOT database had already been verified against the VISSIM ground truth, no additional validation of the UTAIM database was necessary. One of the primary differences between the UDOT ATSPM and UTAIM systems is that UTAIM can generate a trajectory-based time space diagram in addition to the PCD. This provides a more comprehensive visualization of signal progression by illustrating vehicle trajectories and the effective green band along the corridor. Similar to UDOT ATSPM, the UTAIM system also reports performance measures such as AoG, Green Time (GT), and other coordination metrics. A comparison of the PCDs generated by the two ATSPM systems is presented in Fig. 6-16, both diagrams correspond to the intersection of RM 2244 and Barton Creek Boulevard during the analysis period from 6:30 AM to 7:30 AM. The first PCD, generated using UTAIM, displays green, yellow, and red intervals, while the black dots represent individual vehicle arrivals. The UTAIM PCD reports an AoG of 75% and a GT of 63.5%. The corresponding UDOT ATSPM PCD reports an AoG of 73% and a GT of 64%. These values are very similar, indicating strong agreement between the two systems. Minor differences are expected because each platform computes and displays performance measures using slightly different processing methods. Examination of both PCDs shows that the intersection generally operates with a cycle length of approximately 120 seconds, with occasional spikes occurring around 7:00 AM. These extended cycles occur because no vehicle calls were received from the competing phases. Since the Eastbound and Westbound movements operate under minimum recall, the controller continues to serve these phases until a call is received from another approach. This behavior was confirmed in the VISSIM simulation, where no vehicle arrivals were observed on the side street approaches during that period, resulting in the extended cycle lengths visible in both PCDs. Overall, the PCDs generated by UDOT ATSPM and UTAIM exhibit highly consistent trends and performance measures, demonstrating that the UTAIM system accurately reproduces the key coordination metrics while providing the additional benefit of trajectory-based time space diagrams for enhanced corridor level analysis.

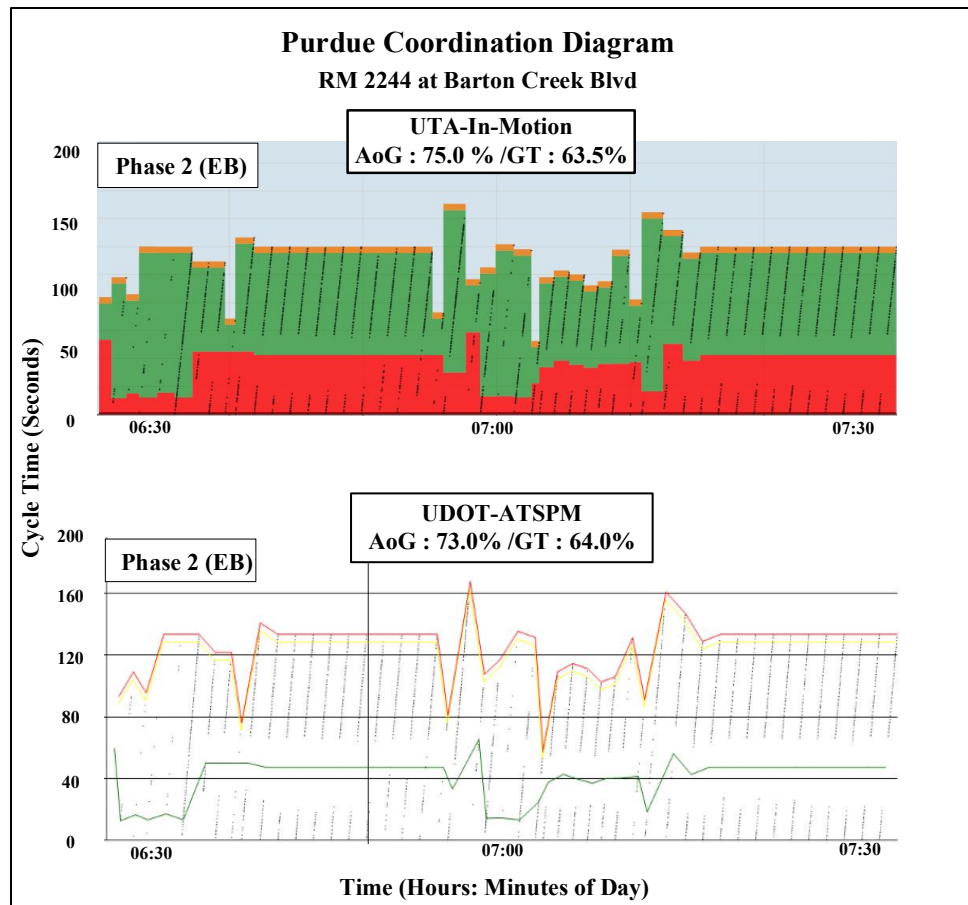


Figure 6-16: PCDs for RM 2244 at Barton Creek Blvd from UTAIM (A) and UDOT ATSPM (B).

In addition to the PCD, the UTAIM system can generate a trajectory-based Time Space Diagram (TSD), which provides a comprehensive visualization of vehicle progression along a signalized corridor. Fig. 6-17 presents the eastbound TSD for the corridor extending from RM 2244 at Creeks Edge Parkway (Intersection 2) to RM 2244 at Overlook at Rob Roy (Intersection 9) during the 6:30 AM to 7:30 AM analysis period. To improve the readability of the figure, only 5% of the vehicle trajectories are displayed, corresponding to the connected vehicle penetration rate assumed in this study. The TSD illustrates the trajectory of individual vehicles together with the signal indications at each intersection. It also displays several performance measures, including AoG, Arrivals on Coordination (AoC), GT, and Band Attainability (BA). The green band shown in the diagram represents the time window during which a vehicle traveling at the progression speed can pass through consecutive intersections without stopping. A wider and more continuous green band indicates better signal coordination, whereas interruptions to the green band often result in vehicle stops and reduced progression efficiency. In Fig. 6-17, the red lines represent individual vehicle trajectories. It is evident that many trajectories exhibit repeated stop and go movements, indicating the presence of queues and interrupted progression along the corridor. As queues discharge from one intersection, some vehicles are unable to clear the downstream intersection before the green interval ends, resulting in split failures. Region A of Fig. 6-17 highlights this behavior, where

frequent stop and go movements and multiple split failures can be observed. These patterns indicate that the available green time is insufficient to serve all arriving vehicles during a single signal cycle, causing vehicles to remain queued for the subsequent cycle.

As discussed previously, the PCD alone provides information on signal coordination at an individual intersection but does not reveal how vehicles progress between intersections. By combining the PCD with the trajectory based TSD, both intersection level and corridor level operations can be evaluated simultaneously. This integrated visualization helps identify locations where vehicle progression deteriorates, queues propagate upstream, and split failures occur, thereby providing valuable guidance for selecting intersections and signal timing parameters for optimization.

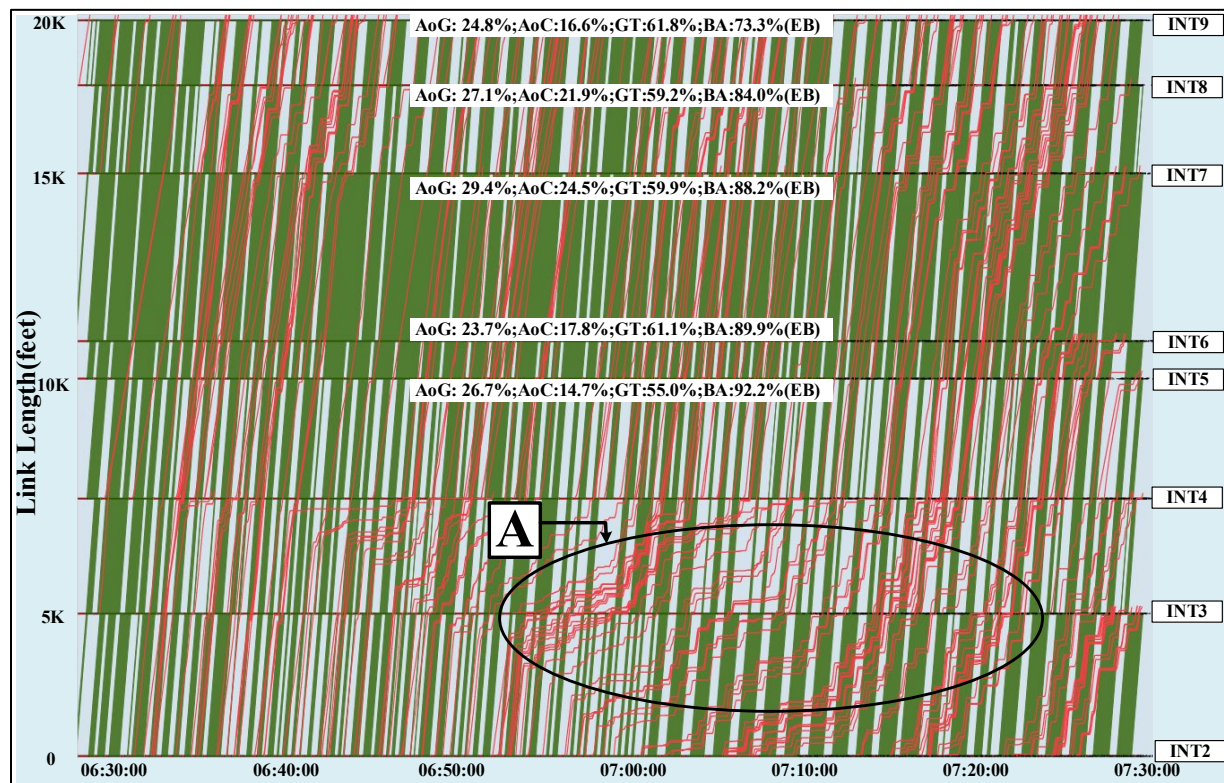


Figure 6-17: TSD for RM 2244 corridor (Intersection 2 to 9).

6.3. Analyzing Traffic Signal Performance with Projected Travel Demand growth using ATSPM-in-loop simulation

One of the primary objectives of this study was to evaluate future traffic conditions and examine how projected traffic growth affects the ATSPM performance measures. Although the baseline VISSIM model was only partially calibrated against field conditions because of limited data availability, forecasting future traffic demand is a common practice in transportation planning. Many transportation agencies assume an annual traffic growth rate of approximately 3% for planning and operational analyses. Since the available traffic volume and turning movement

counts were collected in 2023, future demand was estimated for the 2025 scenario by applying a compound annual growth rate of 3%. Accordingly, all traffic volumes and turning movements were multiplied by a growth factor of $(1 + 3\%)^2 = 1.06$, representing the projected increase in travel demand over the two-year period.

After the traffic volumes and turning movement counts were increased to represent the 2025 scenario, the VISSIM model was rerun using the same simulation settings as those used for the baseline scenario. The resulting simulation outputs were then extracted and parsed into the formats required by the two ATSPM systems using the custom parsing software developed for this study. Once the parsed data were stored in their respective databases, the ATSPM systems generated updated performance measures and visualization graphs for evaluation. The following sections discuss how the projected increase in traffic demand affected the operational performance of the corridor. To maintain consistency with the baseline analysis, the same representative intersections from each of the three corridor zones were selected for comparison. These intersections also represent the different controller configurations used in the study, including one HILS controller, one RBC, and the remaining intersections operating with the Econolite EOS controller. This approach enables a consistent comparison of the baseline and future scenarios while evaluating the impact of increased traffic demand across different controller types.

6.3.1. Analysis of performance metrics from VISSIM

To evaluate the impact of the projected traffic growth on intersection performance, MOEs were extracted directly from the VISSIM node analysis. Since these values are obtained directly from the simulation, they are considered the ground truth for this study. The extracted MOEs include maximum queue length, average queue length, total vehicle delay, average vehicle delay, and LOS for the three representative intersections located in the three different corridor zones. These performance measures were then compared with those from the baseline scenario to quantify the operational impacts of the increased traffic demand. The output after the project growth for future scenarios is presented in Table 6-5.

Table 6-5: Results obtained from VISSIM.

Intersection	ZONE 1		ZONE 2		ZONE 3	
Movement	EBT	WBT	EBT	WBT	NBT	SBT
Average Queue Length	52	0	204.4	4.8	2.2	121
Average Vehicle Delay	65	1.5	24.6	5.8	4.7	79.5
LOS	B		C		C	

After increasing the traffic volumes, selected intersections were evaluated using the VISSIM Node Analysis module. The average queue length and average vehicle delay were compared between the baseline (before) and future (after) scenarios. Although most intersections exhibited only minor changes, three intersections experienced a noticeable degradation in performance. Fig. 6-18 presents the average queue length for Intersections 2, 3, and 4. As shown, the future scenario resulted in longer queues at all three intersections compared with the baseline scenario, indicating the impact of the increased traffic demand. These intersections will be further evaluated using time space diagrams to better understand the progression quality and identify the causes of the increased congestion.

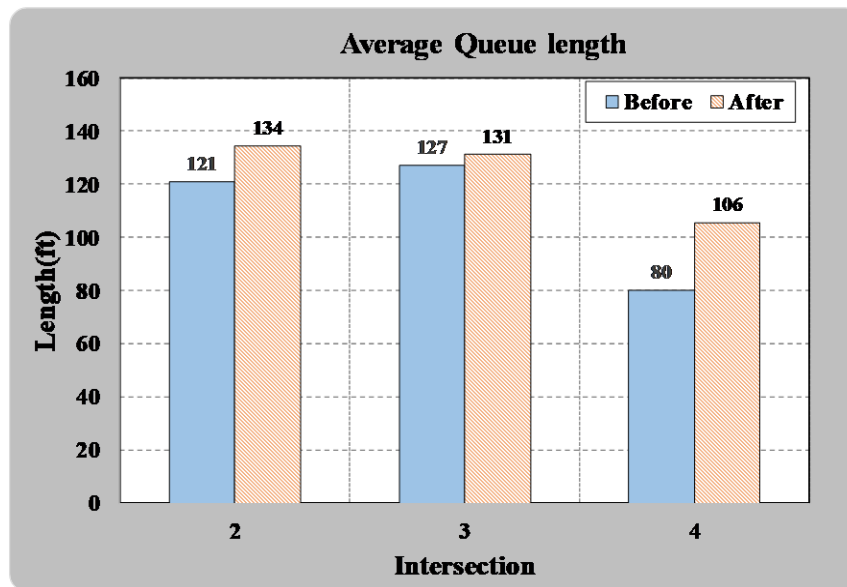


Figure 6-18: Comparison of average queue length before and after volume increase at Intersections.

Fig. 6-19 presents the average vehicle delay for the same three intersections. Similar to the queue length results, the average delay increased in the future scenario, confirming the reduction in operational performance under higher traffic volumes. It should be noted that these values were obtained from the VISSIM Node Analysis module. Since node boundaries are manually defined by the user, the reported queue lengths and delays depend on the size and location of the node surrounding each intersection. Consequently, the absolute values may not exactly match those reported by the ATSPM system. However, they provide a reliable indication of the relative change in intersection performance and clearly demonstrate the increasing trend in delay and congestion across the evaluated intersections.

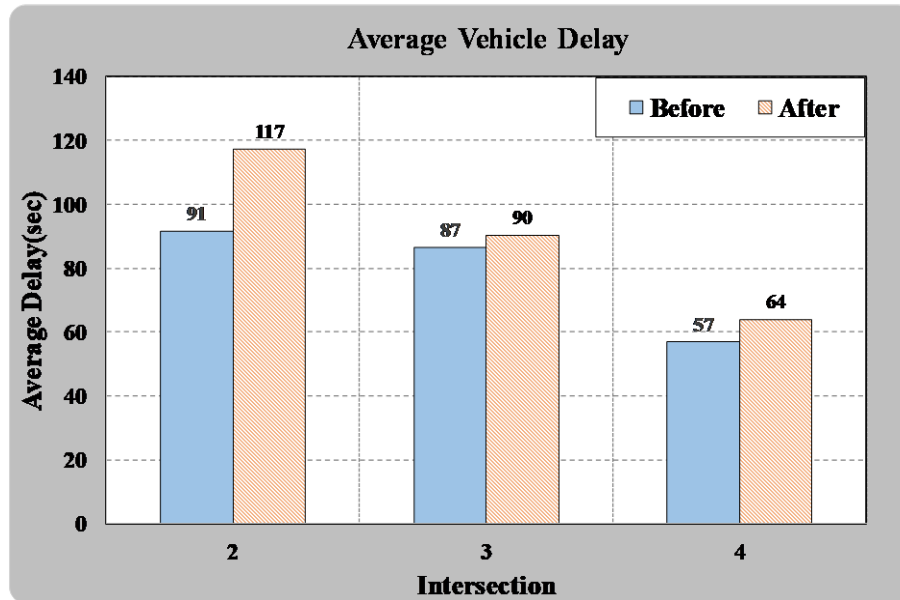


Figure 6-19: Comparison of average vehicle delay before and after volume increase at Intersections.

6.3.2. Analysis of performance metrics from UDOT-ATSPM

Similar to the baseline analysis, the simulation outputs from the future scenario were parsed into the UDOT ATSPM system, from which a variety of performance graphs were generated. For consistency, the same reference intersection, RM 2244 at Rollingwood Town Center (Intersection 23), was selected to illustrate the future scenario results. The complete set of UDOT ATSPM performance charts generated for the study is presented in Table 6-6.

Table 6-6: Performance Metrics for the Chosen Intersection.

Intersection	Zone 1		Zone 2		Zone 3	
Performance Metrics	Phase 2 (EB)	Phase 6 (WB)	Phase 2 (EB)	Phase 6 (WB)	Phase 6 (NB)	Phase 2 (SB)
Approach Delay (sec)	3	2	6	1	1	1
Total Delay (hr.)	1.6	0.2	5.5	0.5	0.3	0.5
Approach Volume	1858	383	3169	2666	1007	1399
Arrivals on Red (AoR-%)	34	31	26	11	11	15
Arrivals on Green (AoG-%)	66	72	74	89	85	89
Green Time (GT- %)	66	69	64	88	85	86
Platoon Ratio (PR)	1.00	1.04	1.16	1.01	1	1.03

The first performance measure evaluated was the approach volume. Fig. 6-20 presents the approach volume graph generated by the UDOT ATSPM system for the projected 2025 traffic demand. The graph was compared with the baseline scenario to examine the impact of increased traffic demand on the intersection approaches. As shown in Fig. 6-20 the traffic volume increased on both approaches compared with the baseline scenario. The increase is particularly noticeable for the southbound approach, where the peak hourly volume increased from approximately 1,000 vehicles per hour to nearly 1,200 vehicles per hour. Despite the increase in traffic demand, the peak hour remained unchanged, occurring between 6:45 AM and 7:45 AM, indicating that the temporal distribution of demand remained relatively consistent while the overall traffic volume increased. The total peak hour volume at the intersection increased from 1,453 vehicles in the baseline scenario to 1,559 vehicles in the future scenario, reflecting the projected traffic growth. The overall K-factor increased slightly from 0.622 to 0.648, indicating that a larger proportion of the daily traffic is concentrated within the peak hour. This suggests that traffic demand becomes slightly more peaked, increasing operational pressure during the busiest period of the day. The PHF decreased from 0.944 to 0.882. A lower PHF indicates that traffic flow within the peak hour became less uniform, with greater fluctuations in demand during the four consecutive 15-minute intervals. In other words, vehicles arrive in more concentrated bursts rather than being evenly distributed throughout the hour. Such traffic patterns generally result in longer queues and increased delays because short periods of high demand can exceed the available intersection capacity. The directional K-factors also experienced minor changes. The northbound K-factor decreased slightly from 0.584 to 0.579, while the southbound K-factor increased from 0.580 to 0.597. These changes indicate that the additional traffic growth is more concentrated on the southbound approach than on the northbound approach, which is consistent with the larger increase in southbound traffic volume observed in Fig. 6-20. Consequently, the southbound approach is expected to experience greater operational impacts, including increased delays and queue lengths, compared with the baseline scenario.

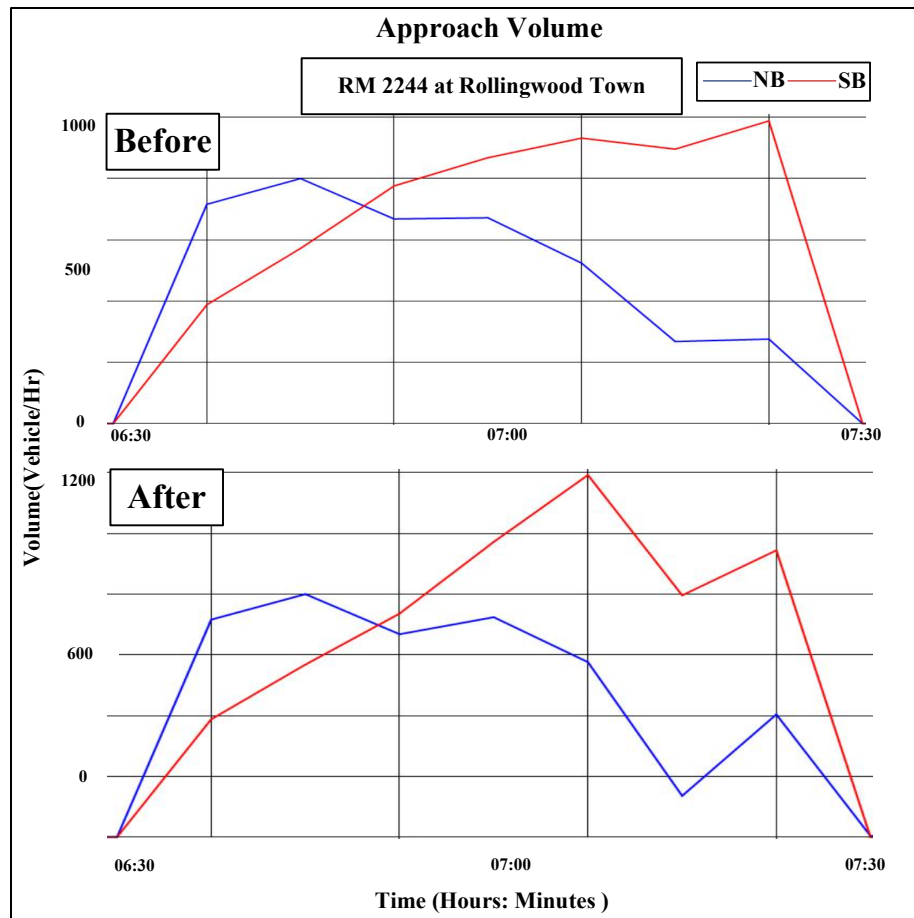


Figure 6-20: Comparison of approach volume before and after for RM 2244 at Rollingwood Town.

Following the approach volume analysis, the approach delay graphs were evaluated for both the baseline and future traffic growth scenarios. Fig. 6-21 compares the approach delay for Phase 6 (northbound through movement) before and after the projected traffic growth. To improve readability, only the graph for the reference intersection is presented in the main text, while the remaining intersections and phases are included in the Appendix. As shown in Fig. 6-21, the overall delay pattern remains similar; however, a slight increase in delay is observed after the projected traffic growth. The total approach delay increased from approximately 0.2 hours to 0.3 hours, while the average delay per vehicle remained at about 1 second. The higher peaks observed during the peak period indicate that the increased traffic demand resulted in additional delays, although the intersection continues to operate efficiently. Overall, the graph suggests that the projected traffic growth has begun to affect intersection performance, but the operational impact remains relatively small.

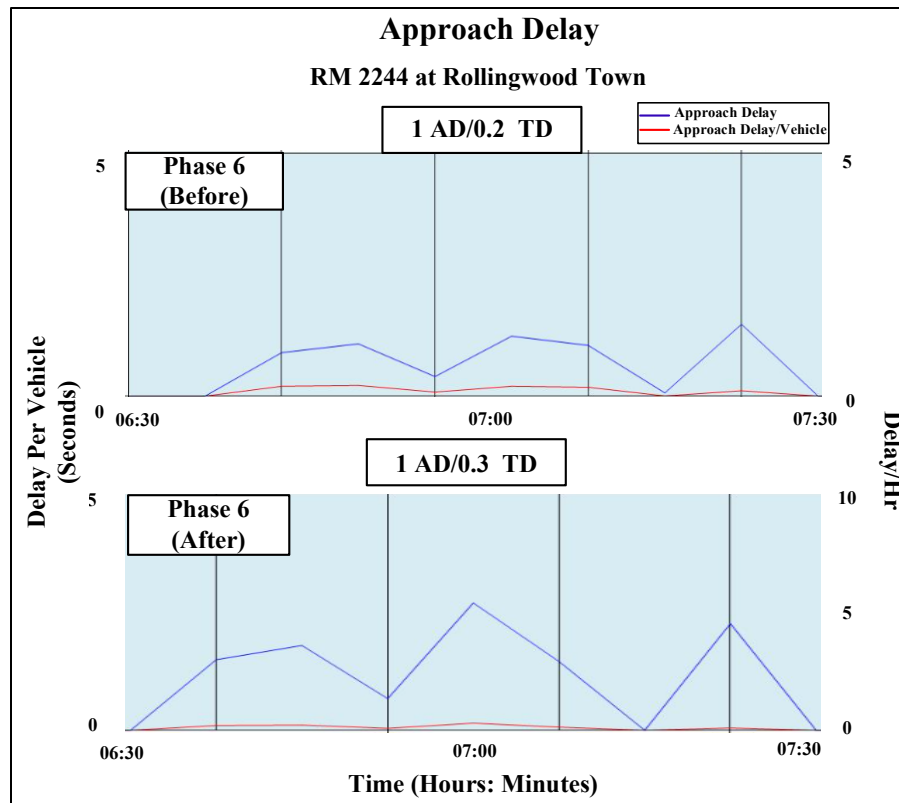


Figure 6-21: Comparison of approach delay before and after (phase 6) for RM 2244 at Rollingwood Town.

Following the approach delay analysis, the AoR graphs were compared between the baseline and future traffic growth scenarios. As shown in Fig. 6-22, the overall AoR increased from 10% in the baseline scenario to 15% in the future scenario. This increase indicates that a larger proportion of vehicles arrived during the red interval, suggesting a deterioration in signal coordination. The signal timing plan and coordination settings were originally developed using the 2023 traffic volumes and turning movements. As traffic demand increased in the projected 2025 scenario, vehicle arrival patterns also changed, reducing the effectiveness of the existing coordination plan. Consequently, fewer vehicles were able to progress through the corridor without stopping, resulting in more arrivals during the red phase. This increase in AoR indicates that the corridor would benefit from retiming or re-optimizing the signal coordination plan to accommodate the projected traffic demand.

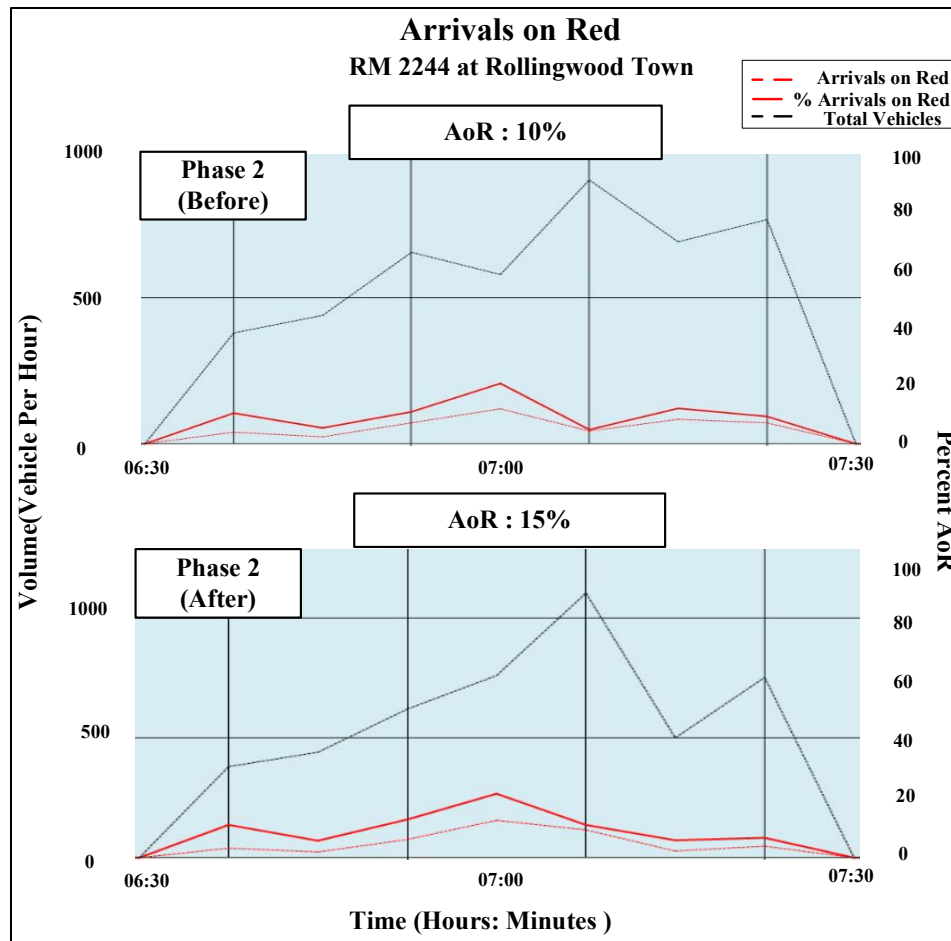


Figure 6-22: Comparison of arrival on red before and after (phase 2) at RM 2244 at Rollingwood Town.

Finally, the PCD was evaluated to compare the baseline and future traffic growth scenarios. Fig. 6-23 presents the PCD for Intersection 23 (RM 2244 at Rollingwood Town Center). As shown in the figure, there is little difference between the two scenarios. Similar to the baseline condition, the cycle length remains approximately 120 seconds, with occasional spikes caused by the absence of vehicle calls on the competing approaches. Under these conditions, the controller extends the green time for the coordinated phase, resulting in longer cycle lengths. Since the signal timing plan and controller settings were kept unchanged for the future scenario, the overall characteristics of the PCD remained nearly identical. Although traffic volumes increased, the controller continued to operate using the same coordination parameters, producing a similar cycle length pattern with only minor variations. This indicates that the increased traffic demand had little effect on the controller operation itself, although other performance measures, such as AoR and approach delay, indicate a gradual deterioration in corridor performance.

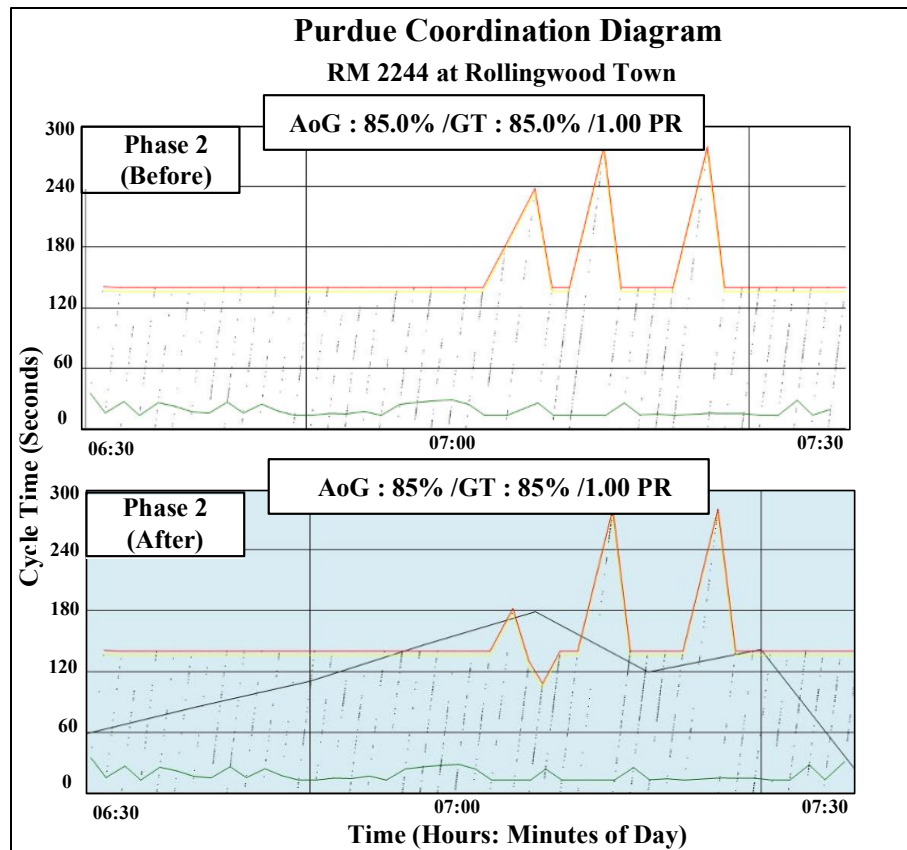


Figure 6-23: Comparison of PCD before and after (phase 2) for RM 2244 at Rollingwood Town.

6.3.3. Analysis of performance metrics from UTAIM

The simulation outputs for the projected traffic growth scenario were also parsed into the UTAIM ATSPM system. Similar to the baseline analysis, the PCD was first evaluated. Fig. 6-24 compares the PCD for Intersection 8 (Zone 2) under the baseline and future traffic scenarios. Similar to the UDOT ATSPM results, only minor differences are observed between the two scenarios. The cycle length remains close to 120 seconds throughout most of the analysis period, with occasional spikes caused by the absence of vehicle calls on the competing approaches. Since the controller settings and signal timing plans remained unchanged, the overall PCD pattern is nearly identical despite the increase in traffic demand. This indicates that the controller continued to operate using the same timing plan while serving a higher traffic demand. The corridor performance metrics derived from the PCD also show only minor changes. The AoG increased slightly from 75% in the baseline scenario to 76% in the future scenario, while the GT increased from 63.5% to 64.3%. Although these values suggest that the controller operation remained largely unchanged, the AoR increased from 10% to 15%, indicating that a greater proportion of vehicles arrived during the red interval under the higher traffic demand. This increase suggests that the existing coordination plan is becoming less effective as traffic volumes increase, even though the signal timing parameters themselves remain unchanged.

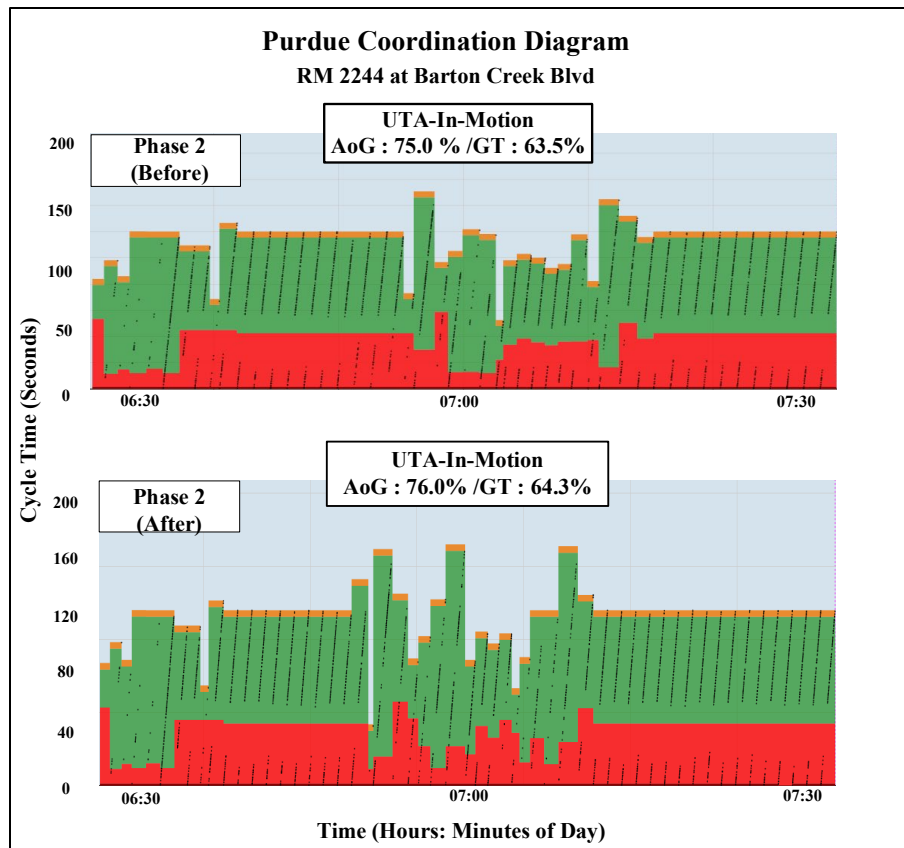


Figure 6-24: Comparison of PCD before and after (phase 2) for RM 2244 at Barton Creek Blvd.

Fig. 6-25 presents the corresponding Trajectory TSD for the future traffic growth scenario. Compared with the baseline scenario, a greater number of vehicle trajectories can be observed because the increased traffic demand resulted in more connected vehicle trajectories being recorded while maintaining the same connected vehicle penetration rate. The red trajectory lines exhibit more frequent stop and go movements, particularly near the downstream intersections, indicating increased queue formation and more vehicles waiting through one or more signal cycles. Although the progression pattern remains generally similar to the baseline scenario, the denser trajectories and longer horizontal sections of the trajectories indicate that additional vehicles are experiencing delays as traffic demand increases. These observations are consistent with the increases in AoR and approach delay discussed in the previous sections, demonstrating that the existing coordination plan is becoming less effective under the projected traffic demand.

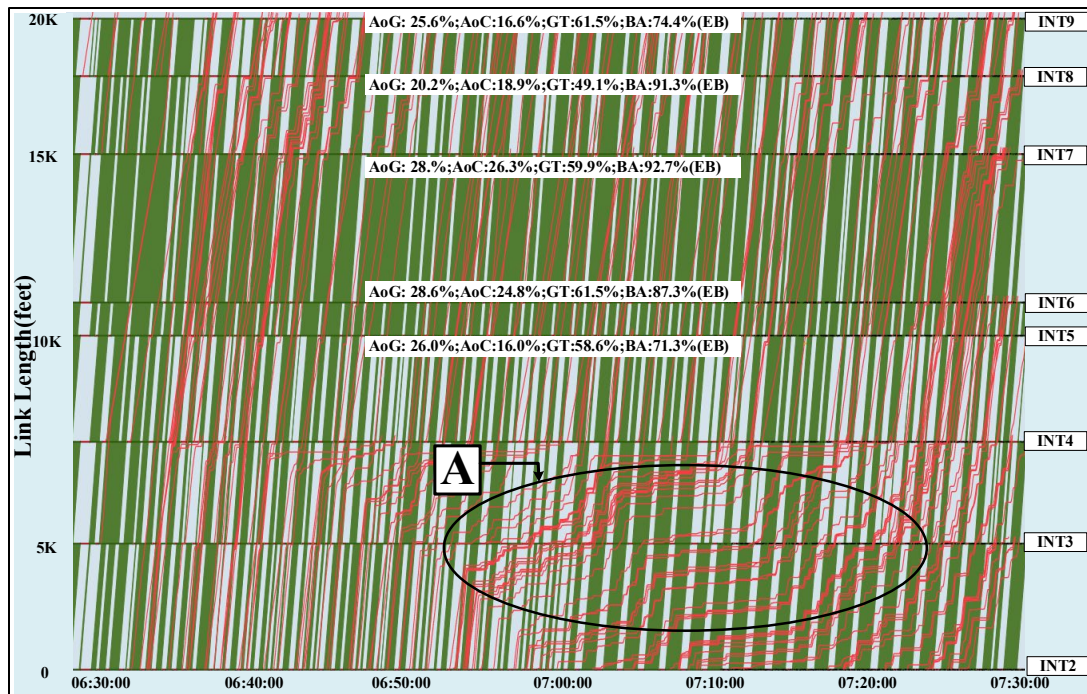


Figure 6-25: TSD for RM 2244 corridor under increased traffic volume.

6.4. Analyzing Traffic Signal Performance with Optimized Traffic Signal

Based on the comparison between the baseline and future traffic growth scenarios, it is evident that the increase in traffic demand resulted in higher delays, increased AoR, and more frequent stop and go movements along the corridor. To demonstrate the effectiveness of signal optimization together with the ATSPM visualization tools, the intersections exhibiting significant stop-and-go behavior in the Trajectory TSD were selected for optimization. The optimized TSD was then compared with the previous scenario to evaluate the impact of the signal timing modifications.

In the existing timing plan, all intersections operated using a common cycle length of 120 seconds. For the selected intersections exhibiting frequent stop and go movements, the signal timing parameters, primarily the cycle length, were adjusted to improve vehicle progression. It should be noted that this optimization serves as a demonstration of how modifying controller parameters can influence corridor performance and the corresponding ATSPM visualizations, rather than as a comprehensive signal retiming study. After increasing the traffic volumes, selected intersections were evaluated using the VISSIM Node Analysis module. The average queue length and average vehicle delay were compared among the baseline (before), future (after), and optimized scenarios. Although most intersections exhibited only minor changes, three intersections experienced a noticeable degradation in performance after the traffic volumes increased. Fig. 6-26 presents the average queue length for Intersections 2, 3, and 4. As shown, the future scenario resulted in longer queues at all three intersections compared with the baseline scenario, indicating the impact of the increased traffic demand. After implementing the signal timing optimization, the average queue

lengths were reduced at all three intersections compared with the future scenario. Although the optimized queue lengths remained slightly higher than those observed in the baseline scenario, the results demonstrate that the optimization was effective in mitigating the additional congestion caused by the increased traffic demand. These intersections will be further evaluated using time space diagrams to better understand the progression quality and identify the causes of the increased congestion.

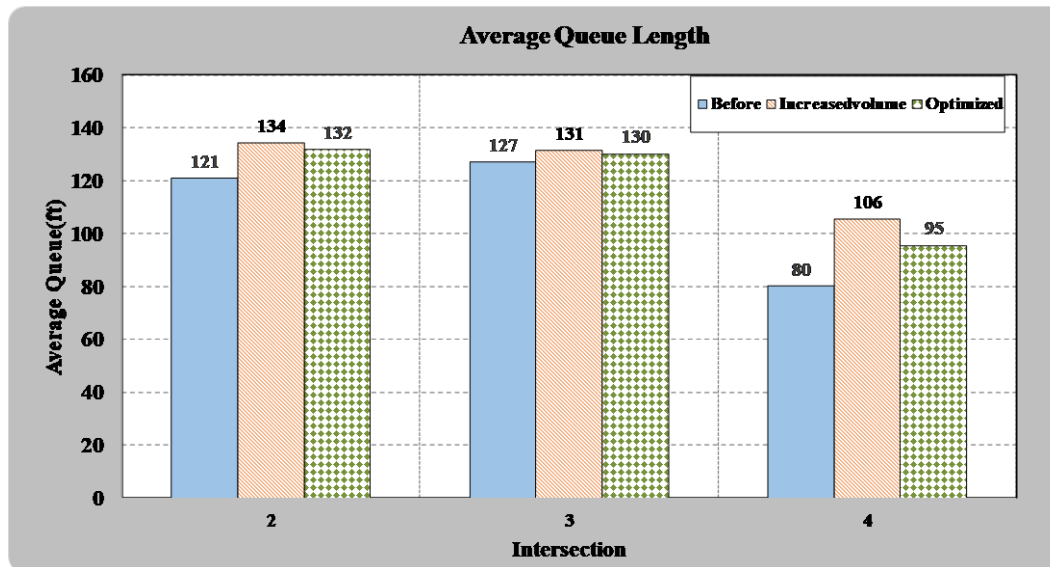


Figure 6-26: Average queue length after signal timing optimization.

Fig. 6-27 presents the average vehicle delay for the same three intersections. Similar to the queue length results, the average vehicle delay increased in the future scenario, confirming the reduction in operational performance under higher traffic volumes. Following the optimization, the average vehicle delay decreased at all three intersections, indicating that the revised signal timing improved traffic operations. While the optimized delays did not fully return to the baseline values, they represent a substantial improvement over the future scenario, demonstrating that the optimization successfully reduced the impact of the increased traffic demand. It should be noted that these values were obtained from the VISSIM Node Analysis module. Since node boundaries are manually defined by the user, the reported queue lengths and delays depend on the size and location of the node surrounding each intersection. Consequently, the absolute values may not exactly match those reported by the ATSPM system. However, they provide a reliable indication of the relative change in intersection performance and clearly demonstrate the increase in delay and congestion after the traffic volume increase, as well as the improvement achieved through the optimized signal timing.

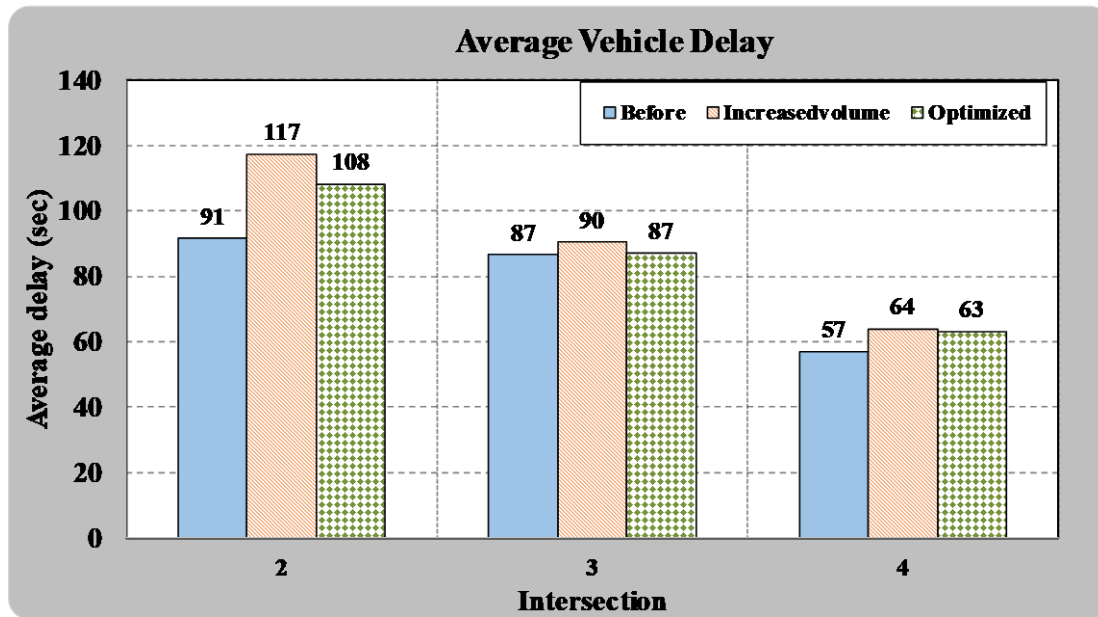


Figure 6-27: Average vehicle delay after signal timing optimization.

To quantify the operational improvements, VISSIM node analysis was used to compare the LOS and other MOEs between the future traffic growth scenario and the optimized scenario. The comparison is presented in Table 6-7. Several intersections experienced a deterioration in LOS under the future traffic growth scenario, indicating increased delays and reduced operational performance. Specifically, Intersections 5, 6, 14, 18, and 23 exhibited poorer LOS than the baseline condition. After signal timing optimization, however, the LOS either remained unchanged or improved. For example, the LOS improved from B to A at Intersections 6, 7, 14, and 17, from E to D at Intersection 20, and from C to A at Intersections 22 and 23. These improvements demonstrate that the optimization strategy effectively reduced delays and enhanced the operational performance of the selected intersections.

Table 6-7: Level of Service obtained from VISSIM.

S. N	Division	Baseline Scenario	Increased Volume Scenario	Optimized Scenario
1	RM 2244 at Senna Hills	A	B	B
2	RM 2244 at Creeks Edge Parkway	D	D	D
3	RM 2244 at Patterson Rd	D	D	D
4	RM 2244 at Cuernavaca Dr	D	D	D
5	RM 2244 at Marly Way	B	C	C
6	RM 2244 at Seven Oaks	A	B	A
7	RM 2244 at Weston Lane	B	B	A
8	RM 2244 at Barton Creek Blvd	C	C	C
9	RM 2244 at Overlook at Rob Roy	A	A	A

S. N	Division	Baseline Scenario	Increased Volume Scenario	Optimized Scenario
10	RM 2244 at Palisades/Dimensional Pl	B	B	B
14	RM 2244 at Red Bud	A	B	A
15	RM 2244 at the Hills	A	A	A
16	RM 2244 at Camp Craft Rd	A	A	A
17	RM 2244 at Westlake Dr	B	B	A
18	RM 2244 at Westbank Dr	A	B	B
20	RM 2244 at Walsh Tarlton Ln	E	E	D
22	RM 2244 at Edgegrove Dr	C	C	A
23	RM 2244 at Rollingwood Town Ctr	B	C	A

Finally, a TSD was generated for the optimized scenario and compared with the future traffic growth scenario to visually assess the impact of the signal timing optimization. As shown in Fig. 6-28, both scenarios exhibit stop and go traffic patterns along the corridor. However, the future traffic growth scenario displays considerably more severe congestion. In particular, Region A, located between Intersections 3 and 4, contains numerous long horizontal trajectories, indicating that vehicles remained stationary for extended periods while waiting in queues. These prolonged horizontal segments are clear indicators of excessive queuing and increased delay under the higher traffic demand. In contrast, the optimized scenario still exhibits some stop and go behavior within the same region, but the frequency and duration of vehicle stops are noticeably reduced. The horizontal segments are shorter and less dense, suggesting that queues dissipate more quickly and vehicles progress through the corridor more efficiently. This improvement is consistent with the reductions in delay and queue length observed from the VISSIM node analysis. It should be noted that this optimization does not represent a comprehensive signal retiming plan for the entire corridor. Instead, only the cycle lengths and splits were modified at selected intersections to demonstrate the potential operational benefits of relatively simple signal timing adjustments. Even with these limited changes, the TSD clearly illustrates an improvement in traffic progression, providing engineers with a useful visual tool for evaluating signal performance and supporting engineering judgment when considering future optimization strategies.

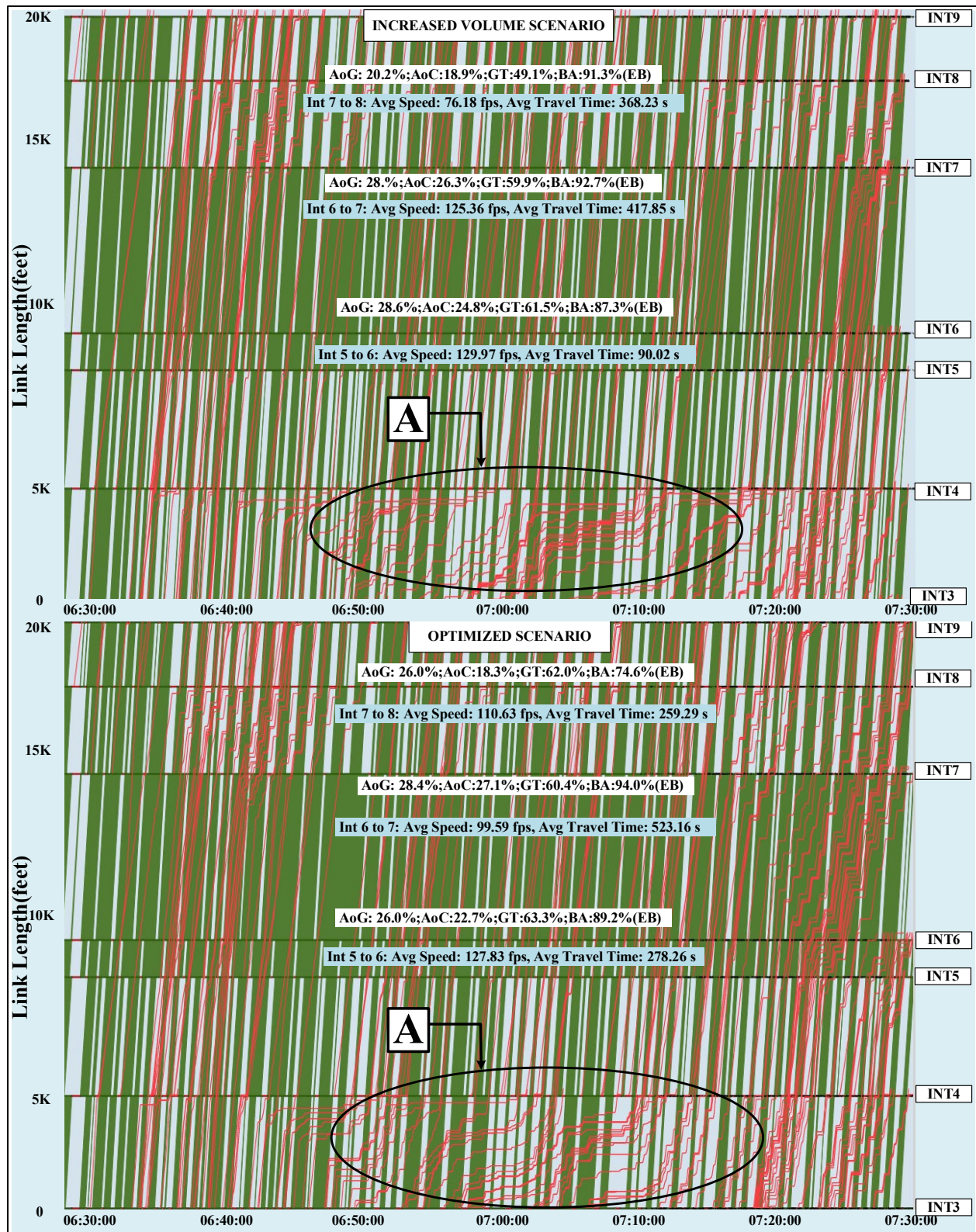


Figure 6-28: TSD for the optimized RM 2244 corridor under increased traffic volume.

6.5. DISCUSSION

This case study demonstrated the application and evaluation of two ATSPM systems, UDOT ATSPM and UTAIM ATSPM, using a VISSIM digital twin of the Ranch to Market Road 2244 (RM 2244) corridor. The simulation model incorporated field traffic volumes, turning movement counts, and actual signal timing plans to establish the baseline scenario. To evaluate future operational performance, traffic volumes were increased using a projected annual growth rate, and the simulation was repeated under the same controller settings. The corridor also included three different controller types, namely Maxtime HILS, VISSIM RBC, and Econolite EOS, demonstrating that the proposed framework can be applied to multiple controller platforms within a single corridor. To support the two ATSPM systems, separate data parsing software packages were developed to convert the raw VISSIM outputs into the formats required by UDOT ATSPM and UTAIM ATSPM. Using the VISSIM node analysis as the ground truth, the generated performance measures and visualization graphs from both systems were evaluated and compared. The results showed that the projected increase in traffic demand led to higher delays, longer queues, and increased AoR at several intersections. Although the PCDs exhibited only minor changes because the controller settings and signal timing plans remained unchanged, the Trajectory TSDs clearly illustrated the operational impacts of the increased traffic demand. The denser vehicle trajectories, more frequent stop and go movements, and additional split failures observed in the TSDs provided a more intuitive visualization of deteriorating corridor performance than could be inferred from the PCDs alone. To demonstrate the practical use of the visualization tools, a simple signal optimization was performed by adjusting the cycle lengths at selected intersections exhibiting significant stop and go behavior. The optimized scenario showed improved corridor progression, with fewer stop and go movements and reduced queuing near the optimized intersections. This case study demonstrates that combining ATSPM visualizations with simulation-based optimization provides transportation engineers with an effective framework for identifying operational deficiencies, evaluating alternative signal timing strategies, and assessing their impacts before field implementation.

APPENDIX

For this case study, only selected intersections were analyzed and discussed in detail to demonstrate the evaluation methodology and the effectiveness of the proposed optimization. However, the developed system automatically generated the same performance figures and analyses for all intersections along the study corridor. To avoid redundancy within the main body of the chapter while still providing supporting results, the remaining figures are included in this appendix. These figures correspond to the 2023 baseline scenario before the projected traffic-volume increase.

Figures 7-1 through 7-3 present the supplementary baseline results for RM 2244 at Barton Creek Blvd, which was operated using the Econolite EOS controller within the SILS environment. Figures 7-4 through 7-7 present the corresponding results for RM 2244 at Senna Hills, where the HILS-CID was used to connect VISSIM with a physical traffic signal controller. The results generally follow the trends discussed for the representative intersections in Chapter 6 and further demonstrate that both controller configurations can generate usable data for ATSPM analysis.

These supplementary results are important because they demonstrate that the same workflow can be applied to other intersections along the corridor without requiring a separate analysis procedure. The developed system can automatically generate the required performance outputs for Econolite EOS, RBC, and HILS-CID controlled intersections, providing users with a consistent method for reviewing baseline conditions and comparing them with future-demand or optimized scenarios.

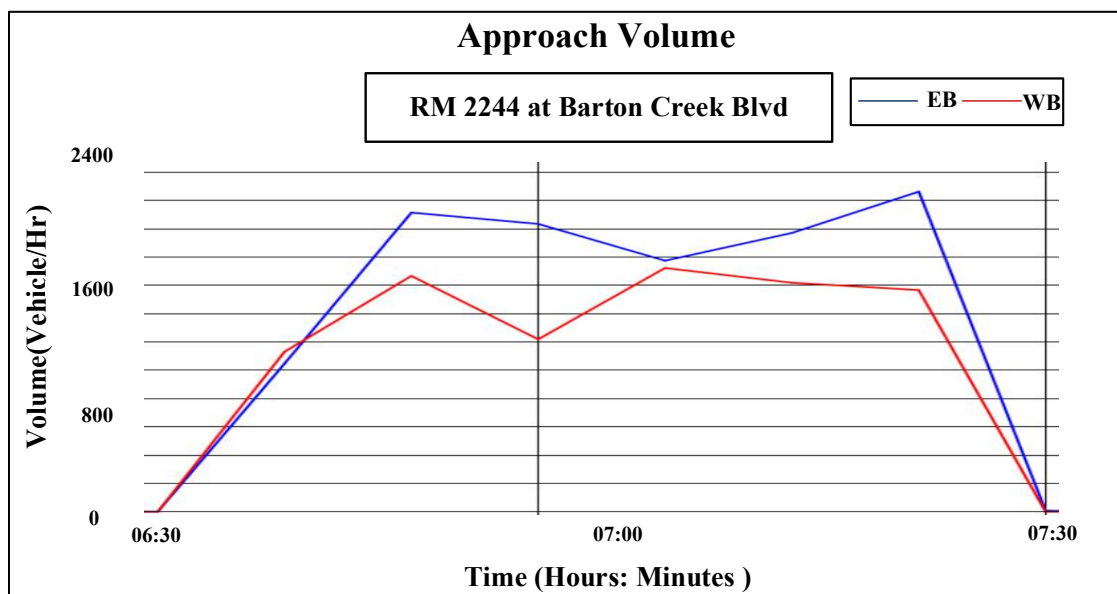


Figure 7-1: Approach Volume at Barton Creek Blvd.

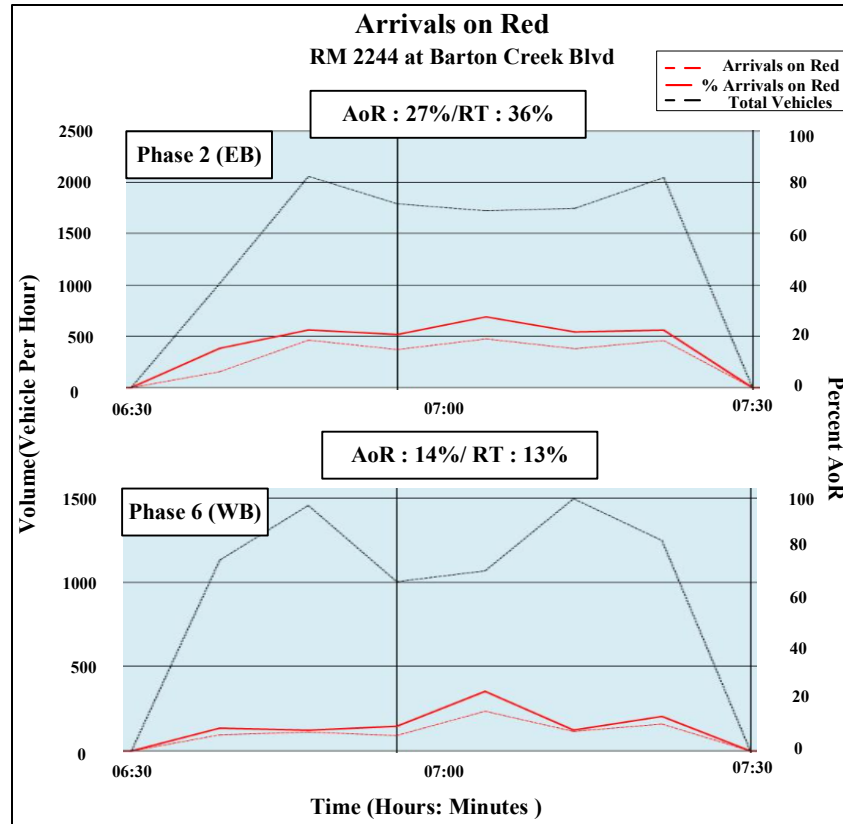


Figure 7-2: Arrival on Red for RM 2244 at Barton Creek Blvd.

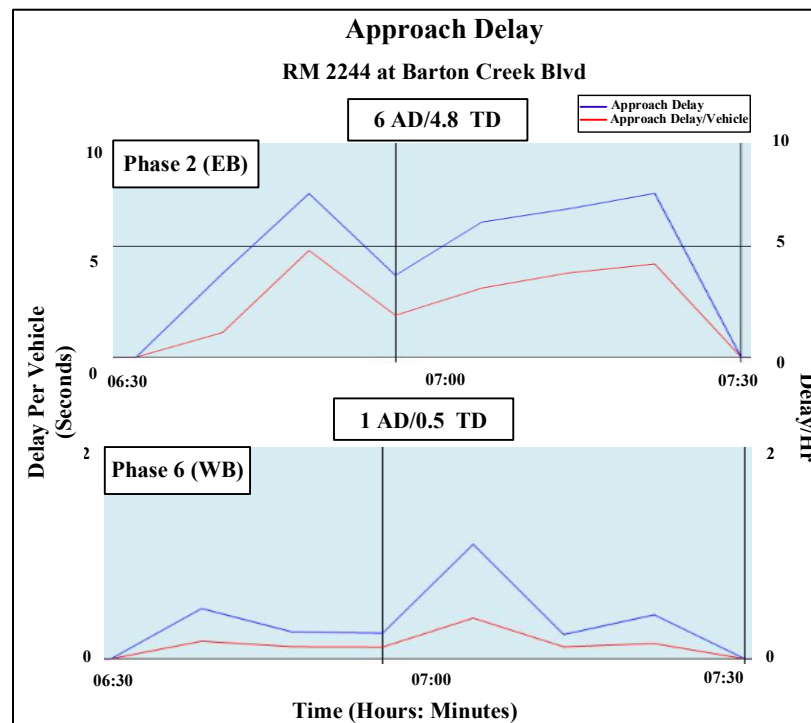


Figure 7-3: Approach Delay at Barton Creek Blvd.

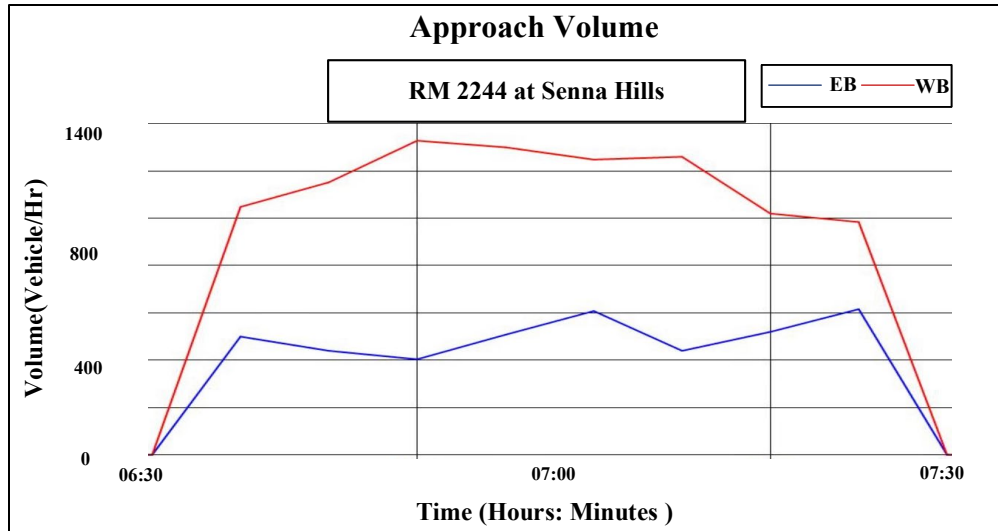


Figure 7-4: Approach Volume at Senna Hills.

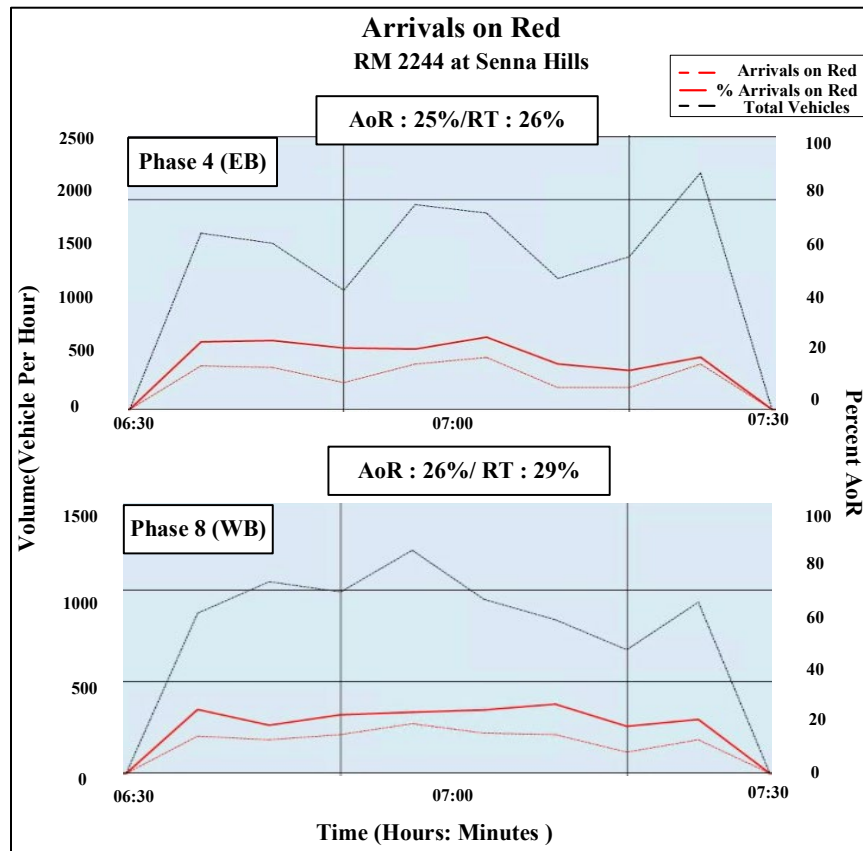


Figure 7-5: Arrival on Red at Senna Hills.

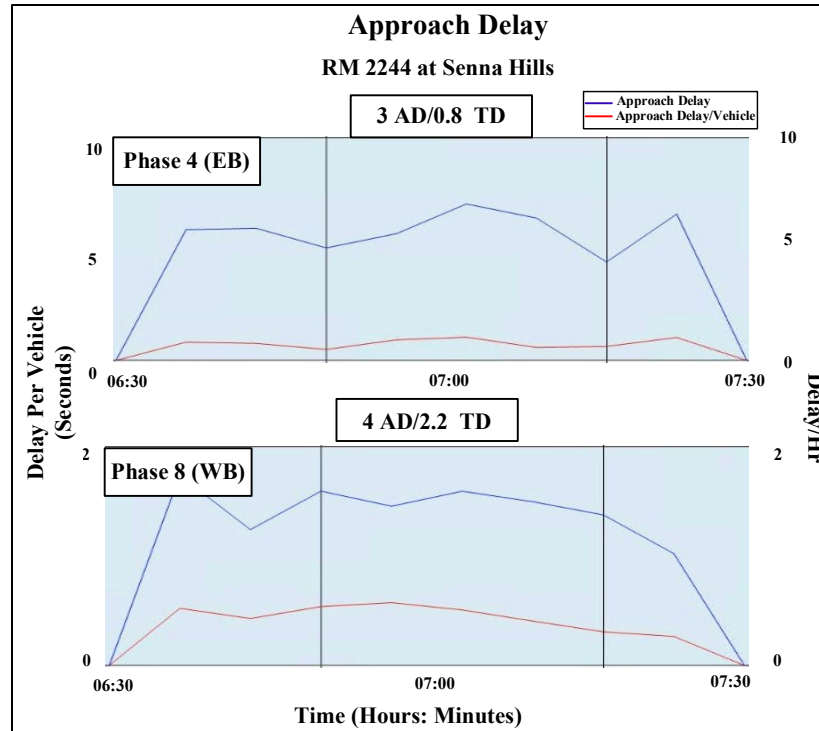


Figure 7-6: Approach delay at Senna Hills.

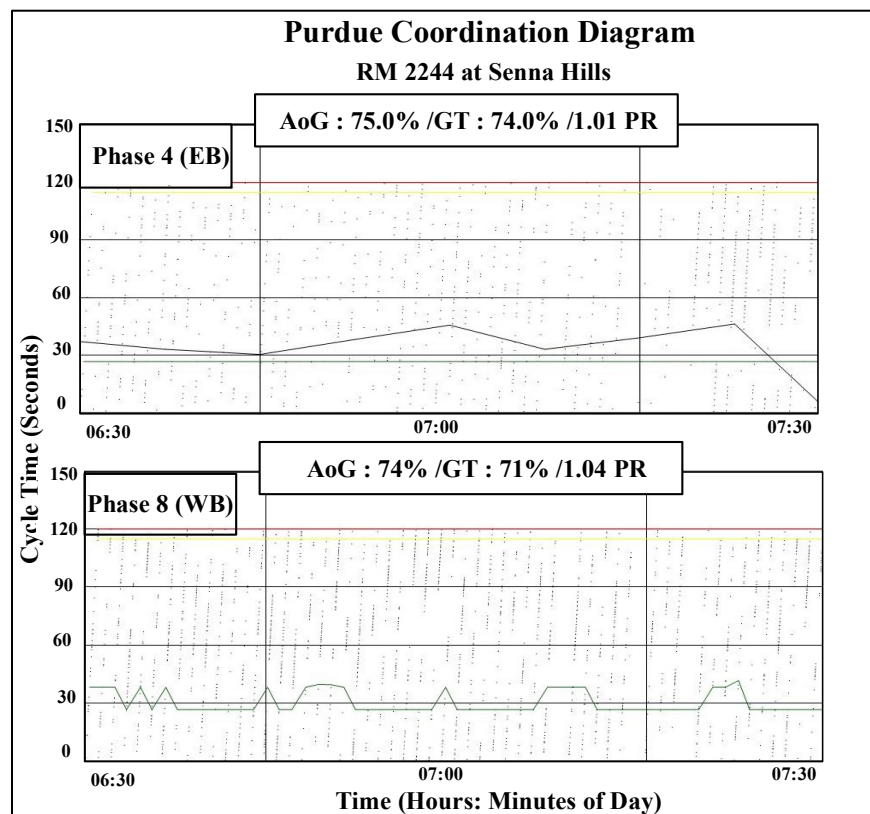


Figure 7-7: PCD for RM 2244 at Senna Hills from UDOT ATSPM.