Automatically Assessing the Need for Traffic Signal Retiming Using Connected Vehicle Data

by

Pirooz Ghaleh

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Civil Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Traffic signal controllers are a critical element in managing urban traffic systems effectively. The emergence of Automated Traffic Signal Performance Measures (ATSPM), aided by recent technological advancements, allows for continuous traffic performance monitoring, and supports traffic agencies in taking proactive measures. High-resolution trajectory data from connected vehicles (CVs) has surfaced as a cost-effective method for assessing ATSPM. Although various metrics have been developed to measure traffic signal performance, none have been specifically designed to predict the benefits of signal retiming. This study devises a novel metric using CV data to estimate the potential reduction in overall intersection delay that could result from signal retiming. Hence, this measure uniquely estimates the potential avoidable delay rather than simply the observed signal performance. This new metric could enable traffic agencies to predict the benefits of potential signal retiming without the need for conducting costly traffic surveys. Such a tool would help these agencies prioritize locations and times of day for signal retiming. This study outlines the process of calculating this index and employs the VISSIM microsimulation software to demonstrate and evaluate the index under various traffic scenarios and CV market penetration rates. In our experiments, the suggested metric successfully detected signal retiming needs in situations involving an imbalanced degree of saturation, traffic demand fluctuations on competing movements, and changes in traffic direction, even with CV penetration rates as low as 10%.

Acknowledgements

My heartfelt thanks are extended to Dr. Christian Bachmann and Dr. Liping Fu, whose mentorship and scholarly guidance have been the cornerstone of my research pursuits. Their encouragement and profound insights have been fundamental to the realization of this thesis.

I am deeply thankful for the enduring support and love from my wife, Baharak, who has been a pillar of strength and encouragement.

I extend my appreciation to my colleagues and friends at the University of Waterloo and the iTSS group. Their camaraderie and support have greatly enriched my research experience.

A special note of gratitude to my parents, whose unconditional love and nurturing of my curiosity have shaped my academic journey. Your belief in me has been a source of constant inspiration.

Dedication

To my dearest Melorin and my loving wife, Baharak,

This thesis is a testament to the future I yearn to build for us. Each page bears the weight of time we've sacrificed, a small step towards a shared dream of a better tomorrow. To you both, I dedicate not only my work but every stride we take together toward the promise of our family's brighter future. With all my love and devotion.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The transportation system is a crucial element of the economy, enabling the movement of people and goods across different locations. In urban areas, the road network plays a vital role in accommodating a significant share of traffic and meeting transportation demands. To ensure the smooth flow of vehicles within this network, signalized intersections are strategically positioned to regulate conflicting movements that occur on shared road segments.

Signalized intersections are essential for effective urban traffic management. These intersections allocate time to different road users, enabling them to make their movements safely without conflicting with each other. By sharing the space over time, signalized intersections enhance safety by preventing simultaneous green signals for conflicting movements. However, this sharing of space over time comes at a cost, reducing road capacity and causing delays. Increased delays at intersections can negatively impact mobility, quality of life, fuel consumption, and emissions.

Traffic signal control plays a pivotal role in urban transportation management. Over the years, the methodologies governing signal control have undergone considerable evolution. These methodologies can be broadly classified into three predominant categories: fixed time, actuated, and adaptive.

1. **Fixed-Time Control:** This conventional approach operates using predetermined signal timings, irrespective of the real-time traffic demand. These timings are set based on historical traffic data or established design standards. While its implementation is straightforward, the method's rigidity and inability to adjust to fluctuating traffic conditions can result in operational inefficiencies.

2. **Actuated Control:** Actuated control methods employ real-time vehicular detection, typically facilitated by sensors, to adjust signal timings accordingly. While it offers more flexibility than fixed-time control, it operates within predefined minimum and maximum time thresholds.

3. **Adaptive Control:** As the most sophisticated of the three methodologies, adaptive control continuously modifies signal timings in response to real-time traffic conditions. Utilizing advanced sensors and algorithms, this method aims to proactively optimize traffic flow and alleviate congestion.

In many urban environments, notwithstanding the availability of these advanced methodologies, a substantial portion of traffic signals operate using fixed-time plans. Such a preference can be attributed to reasons ranging from cost considerations and infrastructure constraints to the straightforward nature of fixed-time control. Nonetheless, with the dynamic nature of traffic patterns, there emerges a pressing need for periodic reassessment and adjustment of these fixed timings to ensure congruence with contemporary demands. This process is termed "signal retiming".

Signal retiming is a commonly employed practice to maintain the efficiency of fixed time signalized intersections. With changing traffic patterns, it becomes necessary to assess the performance of signal timing and make necessary adjustments. However, traffic agencies often face resource limitations, resulting in infrequent retiming cycles of 3 to 5 years or reliance on citizen complaints to prompt action (Federal Highway Administration, 2022).

This study focuses on harnessing Connected Vehicle (CV) trajectory data to estimate the need for retiming fixed-time signals. Leveraging this data can provide valuable insights for traffic agencies, enabling them to prioritize their resources and conduct traffic surveys where the most significant benefits are predicted. By moving away from relying solely on citizen complaints or periodic retiming schedules, this approach aims to enhance traffic signal efficiency and improve urban traffic management. Ultimately, such improvements have the potential to yield social, economic, and environmental benefits in the long run.

1.1 Traffic congestion impacts

Traffic congestion is a pressing issue in urban areas, significantly impacting various aspects of transportation systems. It has far-reaching effects on the economy, environment, and society at large. The consequences of traffic congestion include extended travel times, reduced productivity, increased fuel consumption, and elevated greenhouse gas emissions, all of which negatively impact the economy and air quality (Eckert, 2022). The Canadian Automobile Association (CAA) reported that major Canadian cities experience substantial traffic bottlenecks, leading to a substantial rise in travel time, with increases of up to 50 percent compared to free flow conditions (CAA, 2021a). Remarkably, three of these bottlenecks rank among the most severe in North America and are comparable to major U.S. cities such as New York and Los Angeles. Consequently, Canadian drivers consume an additional 22 million litres of fuel annually, resulting in extra 58 million kilograms of CO_2 emissions released into the atmosphere (CAA, 2021a).

In 2022, data from TomTom, a location technology company, revealed significant delays for drivers in major Canadian cities during rush hour. In the seven most congested city centers, drivers typically spend an average of 135 to 199 hours per year in rush hour traffic. Toronto, Vancouver, and Winnipeg emerged as the two worst cities in Canada and the third and fourth worst in North America, respectively. Globally, they ranked 30th and 31st in terms of time spent driving in rush hour per year. The impact of this traffic congestion is evident when considering environmental and financial implications. TomTom's analysis encompassed various measurements, including the yearly cost of driving 10 km during rush hour, total driving time, fuel prices, and CO2 emissions for different cities. For instance, in Toronto, annual CO2 emissions reached 990 kg, with 203 kg attributed to congestion, and drivers incurred a yearly fuel cost of CA$755, with CA$155 resulting from congestion. Similarly, in Vancouver, yearly CO2 emissions amounted to 830 kg, with 120 kg caused by congestion, and drivers faced a yearly fuel cost of CA$633, with CA$92 due to congestion (Traffic Index Ranking, 2022). These figures underscore the substantial impact of congestion on carbon emissions and financial expenses.

1.2 Urban traffic management and traffic signals

Efficient urban traffic management necessitates a comprehensive and integrated approach to effectively address the challenges posed by traffic congestion. It is imperative to recognize that no universal solution can single-handedly alleviate the problem. Merely constructing additional roads is insufficient in tackling congestion, and even the adoption of Electric Vehicles, while beneficial for reducing emissions, does not resolve traffic congestion (Hristova, 2023). To successfully mitigate the impact of congestion, a multifaceted strategy that encompasses various measures and strategies is essential. By implementing a holistic approach, traffic congestion can be proactively managed.

Traffic management systems encompass various technologies and strategies aimed at enhancing traffic flow and safety, ranging from traffic signals to automated enforcement and variable speed limits. These systems offer a cost-effective approach to optimizing the capacity of existing infrastructure, minimizing the need for expensive investments in new roads or transit. By improving traffic signals, congestion can be reduced through faster traffic movement, increased reliability, and fewer collisions. Advanced signal systems can detect traffic patterns and adjust timing, potentially reducing delays by up to 40% (CAA, 2021a). Notable examples include the City of Toronto's signal re-timing program, which allocated CA$ 850,000 annually from 2012 to 2015, resulting in a cost-benefit ratio of CA$ 64.2 saved for every dollar spent (Livey, 2015).

1.3 Traffic signal retiming and limitations

As indicated previously, for fixed-time traffic signals, periodic signal retiming is a commonly employed practice to address the dynamic nature of traffic patterns and changes in demand. Through periodic adjustments of signal timings, transportation agencies aim to optimize signal phasing and timings to accommodate the varying traffic conditions. Signal retiming is a cost-effective method to improve traffic flow. It reduces delays, enhances safety, and decreases fuel consumption and emissions. It has been shown that the benefit-to-cost ratio for signal retiming could reach as high as 40:1 (Sunkari, 2004). Traditional periodic signal retiming approaches have limitations that stem from resource constraints and inefficiencies. Studies have shown that signal retiming is a cost-effective investment, with an average cost of US$3,700 per signal and 26 person-hours of work for most agencies (Gordon, 2010). However, resource limitations remain a significant factor contributing to suboptimal signal retiming. Although guidelines recommend reviewing and retiming signals every 30 months to 3 years, most agencies exceed this interval and typically retiming is done at a 5-year period (Gordon, 2010). To improve the signal retiming process, clearer guidance is needed on coordination boundaries, controller timing plan parameters, and the optimal number of timing plans and their durations of use. Additionally, investigating specific design necessities for stored data user services and management systems could cut the expenses associated with assessing current timing plans and elevate their role in identifying retiming needs (Gordon, 2010).

1.4 Traffic signal performance measurement

Performance measurement serves two primary functions in signal retiming: performance monitoring and supporting signal retiming process. Performance monitoring involves assessing the impact of signal timing on various intersection users, such as drivers, pedestrians, cyclists, and public transit vehicles. These measures are typically aligned with the agency's objectives for retiming and are employed to evaluate signal performance pre- and post-retiming, providing insights into the current conditions. On the other hand, supporting signal retiming process is closely linked to performance monitoring but specifically aimed at assisting traffic engineers during the modelling and design phase of retiming (Gordon, 2010).

To achieve an in-depth insight into traffic signal performance measurement, it is valuable to examine the evolution of signal timing procedures over time, particularly through the analysis of NCHR manuals. The initial edition of the traffic signal timing manual was published in 2008, with its primary focus on optimizing signal plans to minimize vehicle delays within a model framework. However, the second edition, published in 2015, introduced a critical realization that the optimized signal plan for minimizing vehicle delays does not always yield the most effective solution for signal retiming challenges. In this edition, the importance of traffic signal performance measurement was emphasized, expanding beyond solely considering vehicle delays. The outcome-based signal timing procedure presented various performance measurement methods that could be selected based on the operational objectives. Building upon these advancements, the third edition of the signal timing manual was released in 2020 under the title "Performance-Based Management of Traffic Signals." This edition was motivated by the increased availability of data to traffic agencies through smart city technologies, specifically focusing on the utilization of Automated Traffic Signal Performance Measures (ATSPM) technology. ATSPM offers extensive performance metrics. It is evident that there exists a significant relationship between traffic signal performance measurement and signal timing procedures, warranting a comprehensive review of their evolution in the following sub-sections.

1.4.1 Traditional signal timing practice

The first edition of the ‘Signal Timing Manual’, STM, was published in 2008. The purpose of the STM is to guide managers, supervisors, and practitioners in improving signal timing through clear direction and sound practices. By emphasizing proper training, proactive operation, and effective maintenance of traffic signals, the manual aims to achieve optimized signal timings that reduce congestion and fuel consumption (Koonce, 2008).

This concise and practical manual offers comprehensive guidance on signal timing, covering various aspects from policy and financial factors to the creation, evaluation, and maintenance of timing plans. It serves as a valuable reference for practitioners involved in the management, operation, and maintenance of traffic signals, as well as those responsible for planning and designing these systems (Koonce, 2008). In this edition, performance measures were introduced as the best way to assess the effectiveness of the policy and implementation of signal timing. These measures capture user-perceived operational performance, such as delay per person or vehicle, travel time, queue lengths, and air quality or vehicle emissions. This recognition stems from the fact that the capacity and demand of the signal are not easily perceived by users, making these performance measures crucial for evaluating signal efficiency and effectiveness (Koonce, 2008).

In STM, several performance metrics were introduced to assess the effectiveness of signal timing. Intersection-level field performance measurements include stopped delay and control delay, which provide insights into vehicle delays at intersections and the impact of control devices on overall delay. When dealing with signal timing for a series of signalized intersections, such as coordinated signals, it is crucial to consider performance measures that consider the interaction between adjacent intersections. STM proposes metrics such as the number of stops, travel speed, and bandwidth, specifically designed for arterials and networks. These metrics can be described as follows:

Stopped Delay: An intersection delay study is a commonly used approach to estimate stopped delay. This method involves observing a specific movement at the intersection over a designated period. Two data points are recorded: the total volume of vehicles and the number of vehicles that come to a stop within a given time interval. The sum of the stopped vehicles during the time interval is divided by the total entering volume to calculate the average stopped delay. However, in the HCM, this method has been replaced by the control delay estimation method.

Control Delay: The control delay can be observed directly in the field by noting the entry and exit times of vehicles for a particular movement or direction. The HCM provides a detailed description of this method in Appendix A of Chapter 16, including a methodology and worksheet. Although this method does not capture all aspects of deceleration and acceleration associated with control delay, it is considered to provide a reasonable estimate. However, it should be noted that during oversaturated conditions, queue formation can pose challenges for this method. Queues may extend beyond the measurement area and spill into other intersections, which can affect the accuracy of the measurements. In such cases, estimating travel times for selected origin-destination pairs may be a more useful approach.

Stops: The number of stops is an important measure for assessing the effectiveness of signal systems, despite not being widely standardized. There are two key reasons for their significance. Firstly, stops have a greater impact on emissions and fuel consumption compared to delays, as accelerating vehicles emit more pollutants and consume more fuel than idling vehicles. Secondly, stops act as an indicator of the smoothness of traffic flow on an arterial road, where frequent stops can result in driver dissatisfaction. In the perception of signal timing effectiveness, number of stops can often have a more significant impact than delays, particularly in arterial applications where minimizing stops is a desired objective.

Travel Time, Travel Speed, and Arterial Level of Service: The assessment of arterial traffic progression often relies on measures such as travel time and travel speed. These metrics consider both delays at intersections and travel times between intersections. The Highway Capacity Manual (HCM) provides a framework for determining the arterial level of service (LOS) based on travel speed. Arterial LOS is defined by the class of the arterial and the travel speed along it, considering factors such as intersection spacing, running time between intersections, and control delay for through vehicles.

Bandwidth, Bandwidth Efficiency, and Bandwidth Attainability: Bandwidth, a measure used to evaluate the effectiveness of coordinated signal timing plans, is distinct from other measures as it focuses solely on the temporal and spatial arrangement of signal timings, without explicit consideration of traffic flow. Bandwidth represents the maximum green time allocated to a designated movement within a corridor and is typically expressed in seconds. It can be assessed at the intersection level (link bandwidth) or across the entire arterial (arterial bandwidth). To normalize bandwidth, two related measures, namely bandwidth efficiency and bandwidth attainability, are employed. Bandwidth efficiency accounts for cycle length and is used to quantify the quality of progression along the arterial. Bandwidth attainability, on the other hand, assesses how effectively the available green time is utilized at the most critical intersection. Notably, bandwidth is influenced by demands on non-coordinated phases, offsets along the arterial, and traveler origin-destination patterns. It is also interconnected with travel time and stops, as a larger bandwidth contributes to shorter travel times and fewer stops for vehicles traversing the corridor.

Emissions and Fuel Consumption: Emissions and fuel consumption are key measures used to evaluate the effectiveness of transportation systems. Emissions play a vital role in assessing air quality and are particularly significant in programs aimed at improving air quality. Software packages enable the estimation of emissions and fuel consumption. Additionally, some software packages offer a performance index that combines multiple effectiveness measures, allowing for the optimization of cycle lengths, offsets, and splits. Although these performance indexes may not be directly observable, their component measures can be observed in the field.

In STM, the evaluation techniques for measuring performance metrics relied on the use of analysis tools such as software and simulations, as well as direct field measurements conducted by manual observers or floating car runs. However, these field measurement techniques were found to be demanding and inefficient. As a result, signal timing practices became increasingly reliant on simulations and analysis, while actual field measurements were lacking. This imbalance between simulation-based approaches and field measurements created a gap in accurately assessing signal timing performance.

1.4.2 Technological advances in performance measurements

The STM second edition places a greater emphasis on traffic signal system users and their priorities, aiming to address the limitations of current signal timing models that often adopt a one-size-fits-all approach. These models typically assume that the optimized solution provided by the model meets the needs of the operating environment and users. Nevertheless, such approaches might not consistently meet the varied requirements of users like pedestrians, cyclists, and public transport. On the other hand, the STM2 offers an outcome-focused method for signal timing, enabling experts to devise tailored timing plans grounded in the operational setting, user preferences, and community goals. Subsequently, performance indicators assess the success in achieving these aims. The process involves selecting timing strategies and values, implementing them, and monitoring their impact to ensure they align with the operational objectives. Recognizing that signal timing cannot be approached uniformly, this process acknowledges the need for tailored solutions (Urbanik, 2015).

The second edition of STM emphasized the importance of utilizing various performance measurements and incorporating field measurements instead of solely relying on model results. This shift in approach was driven by advancements in technology, which have improved the process of performance measurement for traffic signals. In recent years, different data sources and methods, such as Bluetooth and Wi-Fi tracking technologies, have emerged as valuable tools for measuring performance metrics like travel time, delay, and queue length. These technologies enable the tracking of vehicles equipped with Bluetooth or Wi-Fi devices, allowing for real-time data collection and analysis to assess travel times and speeds across different sections of the road network. This innovative approach reduces the effort required and facilitates more frequent data gathering compared to traditional methods such as running special floating vehicles. By embracing these advancements, practitioners can make more informed decisions and continually improve signal timing practices (Gordon, 2010).

Every year, traffic engineers develop a range of techniques and tools to facilitate the measurement of specific metrics in the field. In 2010 a significant breakthrough occurred with the development of ATSPMs. ATSPM had such a profound impact that it led to the publication of a new manual titled "Performance-based Management of Traffic Signals". This manual, which can be considered the 3rd version of the STM, provided comprehensive guidance on utilizing performance-based approaches for managing traffic signals.

1.4.3 ATSPMs

As previously stated, although guidelines recommend a cycle of 30 months to 3 years for retiming traffic signals, traffic agencies typically manage to do so every 3 to 5 years due to their constraints. The effectiveness of this retiming is often evaluated using limited comparisons of before-and-after travel time data. In the absence of continuous performance measurement capabilities, agencies heavily rely on citizen complaints to identify maintenance or operational issues, leading to compromised safety, reduced efficiency, and increased congestion. The emergence of ATSPMs is modernizing signal timing and operations practices by integrating high-resolution data logging with existing traffic signal infrastructure and analysis techniques. This cost-effective technology equips agencies with the necessary information to proactively manage signal performance, aligning with safety and mobility objectives (FHWA, 2019).

Figure 1-1 shows the ATSPM technology implementation concept in a simple diagram. The process of gathering, transferring, and interpreting ATSPMs involves four main steps. Firstly, time-stamped data is recorded at intersections, capturing vehicle and pedestrian presence as well as signal timing intervals. This data provides insights into intersection operations, such as driver and pedestrian wait times and frequency of vehicle stops. Secondly, the high-resolution signal data is moved to a centralized database, requiring network engineering and adherence to security protocols. Thirdly, the data is interpreted using ATSPM software to normalize and calculate metrics, generating visual reports for staff and public use. Finally, agencies can act based on the data, leveraging ATSPMs to improve safety, efficiency, and the overall experience of signalized intersections. Traffic signal engineers can make timing changes and equipment repairs, managers can prioritize projects, policymakers can evaluate investments, and transportation planners can inform designs (Lonzer, 2019).

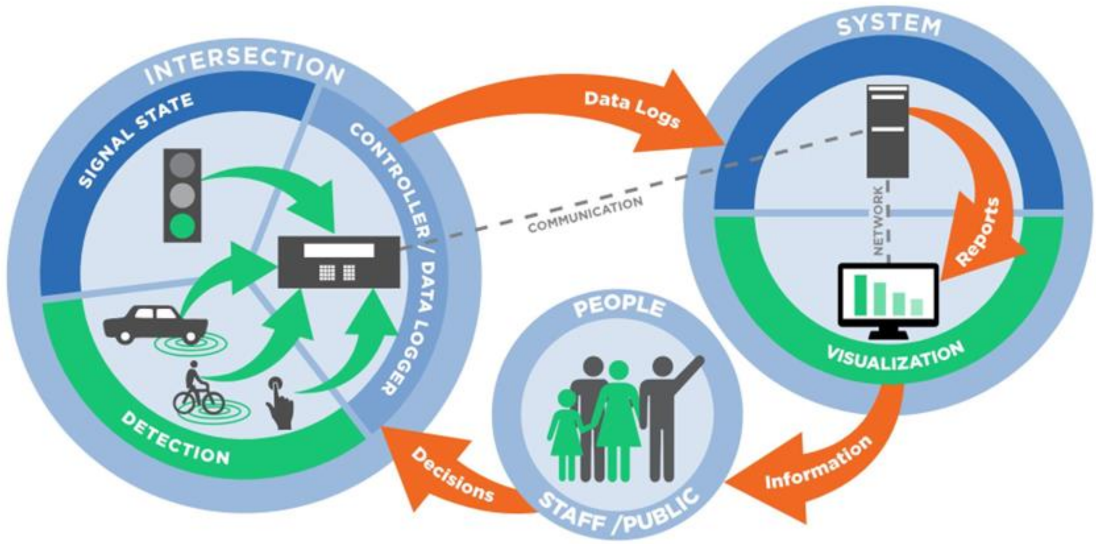


Figure 1-1. ATSPM Technology: intersection data logging, data communication, analysis, and visualization (Lonzer, 2019)

In Figure 1-2, the fundamental process of generating ATSPMs is depicted. At an intersection, signal states (such as green, yellow, and red) and detector events are initially captured and logged by a data logger. These logs are then transferred to data storage, which can be either server-based or cloud-based. Subsequently, the ATSPM software processes the data, enabling the generation of diverse reports (Nevers, 2020).

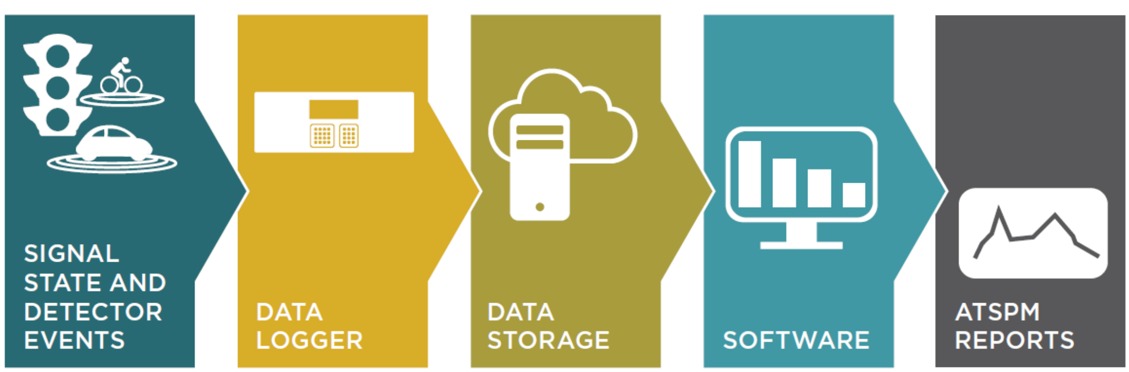


Figure 1-2. The flow of information to produce ATSPMs (Nevers, 2020)

The implementation of signal performance measures based on high-resolution data relies on key components, including existing detection systems, a high-resolution data logger integrated into or added to the signal controller, and a communication infrastructure. This comprehensive approach enables agencies to gather precise data and analyze it effectively for performance measurement. While agencies may face challenges in achieving accurate detection, the availability of high-resolution data helps identify and address any issues. The procurement of a data system and the establishment of a communication infrastructure are crucial steps, particularly for smaller agencies with limited IT resources. Potential solutions include packaged products or services offered by vendors or consultants, as well as cooperative efforts among agencies to procure regional systems with shared data. By overcoming these challenges, agencies can successfully implement ATSPM technology and leverage its benefits for improved traffic signal management (Day, 2015).

1.4.4 Performance-based management of traffic signals

Traditionally, the initiation of signal retiming projects was either based on a set schedule or a request from the public. In such cases, a manager would delegate a specific intersection or corridor to an engineer for retiming adjustments. The engineer would gather traffic data and conduct field observations, subsequently developing a model to emulate those specific conditions. Adjustments to improve operations would be made within this model. The updated signal timings would then be implemented in the actual location by field staff, with further refinements based on onsite observations. Often, a before-and-after analysis would be executed to evaluate the results of the retiming effort. Such conventional methods confined the engineer's insights to a limited timeframe of data and observations, though some aspects, like onsite observations, remained crucial. However, with the advent of signal performance metrics, the process has become more streamlined. This approach offers staff a more extensive dataset, enriching their grasp of traffic dynamics (Nevers, 2020).

ATSPM technology has modernized the approach to traffic signal timing, offering traffic agencies comprehensive and continuous performance measurement capabilities. With access to a wealth of data provided by smart city technologies, agencies can now proactively improve the efficiency of their signal retiming procedures. The significance of this advancement is evident, leading to the publication of the guidebook “Performance-based Management of Traffic Signals” in 2020. This guidebook serves as a valuable resource for agencies looking to incorporate signal performance measures into their comprehensive approach to performance-based management. Smart city technologies such as ATSPM have transformed the management and operation of transportation systems, enabling agencies to collect extensive data at signalized intersections. This data has facilitated the development of numerous performance measures, allowing agencies to gain insights into equipment health, intersection operations, and cycle-by-cycle events. By utilizing dashboard reporting and automated alerts, traffic signal managers and staff can make informed, data-driven decisions throughout all aspects of traffic signal operations, from planning to design and implementation. The guidebook aims to support agencies in harnessing the power of signal performance measures as part of a holistic approach to managing traffic signals (Nevers, 2020).

The “Performance-based Management of Traffic Signals” introduces various data sources for measuring performance metrics, classifying them into internal and external categories. External data sources measure traffic performance independently of traffic signal controllers, and internal data sources capture events directly from the signal controllers. However, the manual predominantly relies on internal data sources, which include high-resolution data from signal controllers or low-resolution data aggregated from traffic centers. Implementing ATSPM heavily depends on acquiring advanced signal controllers with data logging capabilities, installing detectors at intersections, and establishing reliable communication infrastructure. Unfortunately, these requirements are not always readily available, resulting in a capital investment constraint that significantly limits the widespread application and adoption of ATSPM by agencies, thereby slowing down their implementation process.

Many agencies have incorporated ATSPM into their traffic signal management systems. However, the implementation of ATSPM presents challenges that include substantial investments in vehicle detection technology, communication devices, and data systems. Due to these complexities, numerous agencies anticipate that it will take several years to achieve complete deployment of ATSPM across their systems (Saldivar-Carranza et al., 2021).

1.4.5 Trajectory-based ATSPMs

A new approach has emerged that allows similar metrics to ATSPMs to be measured without using local sensors and traffic signal controllers. This approach utilizes trajectory GPS data from vehicles on the road to estimate performance metrics like average speed and vehicle delays. Probe vehicles provide trajectory data with time stamps that can be used to measure performance metrics. Researchers have successfully used trajectory data for ATSPMs. For example, Zhao et al. (2019) suggested leveraging probe vehicle trajectories to estimate queue lengths and traffic counts at signalized intersections.

Saldivar-Carranza et al. (2021) used high-fidelity vehicle trajectory data to measure delay, level of service, and split failure at the intersection level, as well as downstream blockage and quality of progression for corridors. Saldivar-Carranza et al. (2022) centered on assessing the effects of the signal timing plan changes on a signalized corridor using CV data. These studies demonstrate the potential of CV data for signal performance measurement and provide insights into the development of new performance measures and methodologies.

CVs are considered the ideal source for trajectory-based ATSPM as they allow for the collection and sharing of real-time traffic data. However, the current CV environment requires expensive instrumentation at every intersection. While this ideal situation may not yet be feasible, there are alternative ways to acquire trajectory data from CVs. For example, using cell coverage as a communication backbone to collect data from vehicles equipped with onboard GPS units can provide trajectory data with adequate accuracy at the lane level. According to CAA website, by 2022, 70-95% of new cars in Canada were expected to have had wireless technology that reports extensive vehicle information. Although this approach may not be as comprehensive as a fully CV environment, it still provides valuable data that can be used for traffic monitoring and management. Despite these advances, current metrics derived from these data do not allow for an assessment of the improvement potential from signal retiming.

1.5 Problem statement

Traffic signal retiming is a critical aspect of traffic management, but agencies often face challenges due to limited resources, leading to infrequent retiming efforts occurring every 3-5 years. The evaluation of signal retiming effectiveness relies on limited before-after travel time data, which hinders comprehensive performance measurement. Without ongoing monitoring, agencies heavily depend on citizen complaints to identify issues, compromising safety, and efficiency, and increasing congestion (FHWA, 2019).

Moreover, existing metrics used to assess traffic signal performance primarily focus on current performance levels and identifying congestion, such as vehicle delays and level of service (LOS). However, these metrics have limitations in estimating the potential improvements achievable through signal retiming. They fail to provide insights into how signal retiming could effectively enhance traffic flow and reduce congestion.

For instance, considering the LOS D for a left turn movement at a busy intersection where competing movements also have high traffic demands, this LOS of D might be deemed acceptable if there is limited room to change signal splits and improve the overall situation. However, if the same LOS D were measured at an intersection with unoccupied capacity in other movements, there is a high possibility that an optimized signal plan could reduce such congestion. Unfortunately, the current performance metrics like vehicle delay or LOS do not adequately capture these nuances and specific situations. Therefore, there is a need to develop more comprehensive metrics that consider the potential benefits of signal retiming and accurately reflect how it can optimize traffic flow and alleviate congestion.

1.6 Research objectives

The goal of this research is to propose a method to automatically evaluate the need for traffic signal retiming utilizing CV data. This suggested approach accounts for existing conditions as well as the potential for congestion reduction. The research can be structured in terms of the following research objectives:

- **Development of a Method for Predicting Signal Retiming Benefits:** The primary aim is to formulate a method that forecasts the advantages of signal retiming. Leveraging the abundant CV data available, this method intends to guide traffic agencies in pinpointing areas where retiming would be most beneficial.
- **Evaluation of the Developed Method's Capability:** Once the predictive method is established, its efficacy must be evaluated. This could involve employing actual CV data or utilizing microsimulations to scrutinize the method's strengths and weaknesses.
- **Accuracy and Sensitivity Analysis of the Method:** A crucial objective is to assess the accuracy of the developed method and its responsiveness to varying parameters. This analysis will offer insights into the method's merits, potential shortcomings, and possibly suggest mitigating strategies for identified challenges.
- **Evaluation of Results and Proposal for Practical Applications:** The culmination of this research will involve a comprehensive review of the findings, discussions, and conclusions. The outcome

provides recommendations on how traffic agencies could employ the method and suggests prospective research avenues that facilitate its practical application.

In fulfilling these objectives, this thesis seeks to enhance the domain of traffic signal optimization, offering critical perspectives and tools for improving signal management in urban contexts. The complexity of traffic signal optimization is widely recognized, leading to the necessity for certain assumptions to be adopted for simplification. In line with previous studies that have made use of such simplifications, specific assumptions have been incorporated in this research to aid in the prediction of signal retiming benefits. It should be noted that the general applicability of the method might be constrained by these assumptions, suggesting avenues for future enhancements and broader interpretations. Furthermore, while there are countless ways to simulate traffic scenarios, this study only tackles a limited set of them. These limitations, among others, will be further discussed in the Conclusion chapter.

1.7 Thesis organization

This dissertation is systematically structured into five distinct chapters:

- **Chapter 2:** A comprehensive review of the current literature surrounding traffic signal retiming and the role of traffic performance metrics.
- **Chapter 3:** A comprehensive description on the developed method that utilizes CV data to predict traffic signal retiming at intersections.
- **Chapter 4:** Presentation and discussion of the research results.
- **Chapter 5:** A summation of the key findings, conclusions, and suggestions for potential future research in the area.

Each chapter builds upon the previous, guiding the reader through the research process from existing knowledge to the contribution this dissertation makes to the field.

Chapter 2: Literature Review

Chapter 2 provides a concise review of the literature on traffic signal retiming techniques for fixed time signal control systems, utilization of CV data for signal optimization, and performance measurement. The chapter covers the traditional Highway Capacity Manual (HCM) methodology for signal retiming, performance-based approaches, and the challenges involved. It also explores studies on CV data-based signal optimization, examines different algorithms used, and discusses their effectiveness. Furthermore, it examines studies utilizing CV data for assessing signal performance and highlights the challenges faced by traffic agencies in real-time signal control and performance monitoring. The chapter concludes with the identification of research gaps and the potential benefits of developing a metric using CV data for predicting signal performance improvement through retiming.

2.1 Traffic signal timing approaches

This section provides an overview of various traffic signal timing techniques, approaches, and associated challenges. It begins by introducing the traditional Highway Capacity Manual (HCM) methodology for signal timing, emphasizing its conventional practices. The section then explores performance-based traffic signal management approaches, which offer a proactive approach to retiming signals. In contrast to the conventional Automated Traffic Performance Measures (ATSPM) technology that requires significant capital investments, a comparison is made between trajectory-based and conventional ATSPM techniques, highlighting the advantages of each approach.

Furthermore, this section identifies, and delves into, the challenges related to signal retiming techniques, underscoring the need for improved methodologies to address these challenges effectively. The section concludes by discussing the potential benefits of finding a metric that can predict signal retiming impacts. The metric could be used by traffic agencies to focus their resources on running the traffic survey procedure for signal retiming where it is predicted to be most effective. These benefits include optimizing the traffic agencies' resources through targeted traffic surveys and evaluating any intersection of interest using CV data, without solely relying on traditional methods or requiring substantial capital investments. This could potentially increase efficiency in the agency’s efforts and help with proactive and optimal signal retiming.

2.1.1 Traditional HCM method

The signal timing framework is built on two primary elements: the signal timing policy and the signal timing procedure. The signal timing policy outlines the execution of the signal timing procedure, which in turn should be employed to enhance the process. Figure 2-1 illustrates the connection between transportation guidelines and the signal timing procedure (Administration & Transportation, 2008).

Signal timing policy plays a crucial role in controlling and defining priorities within the transportation system and the application of signal timing. It should be a direct reflection of a region's transportation policies, mirroring the emphasis on the operation and safety of the transportation system. It is important to note that the term "policy" in this context supports strategic objectives and may not have universal applications in all cases. Engineering judgment should be employed alongside policy development and application in the realm of signal timing (Administration & Transportation, 2008).

The procedure for formulating a signal timing strategy is commonly solidified in numerous agencies, focusing primarily on optimization mechanics. However, factors such as time, funding, and resources can influence the process. Signal timing involves distinct procedures that are interrelated, including data management, signal optimization, performance evaluation, and deployment. To expand beyond these specific activities, additional steps are incorporated to address other important considerations. This comprehensive signal timing environment, illustrated in Figure 2-1 offers a structure to produce outcomes consistent with overall policies (Administration & Transportation, 2008).

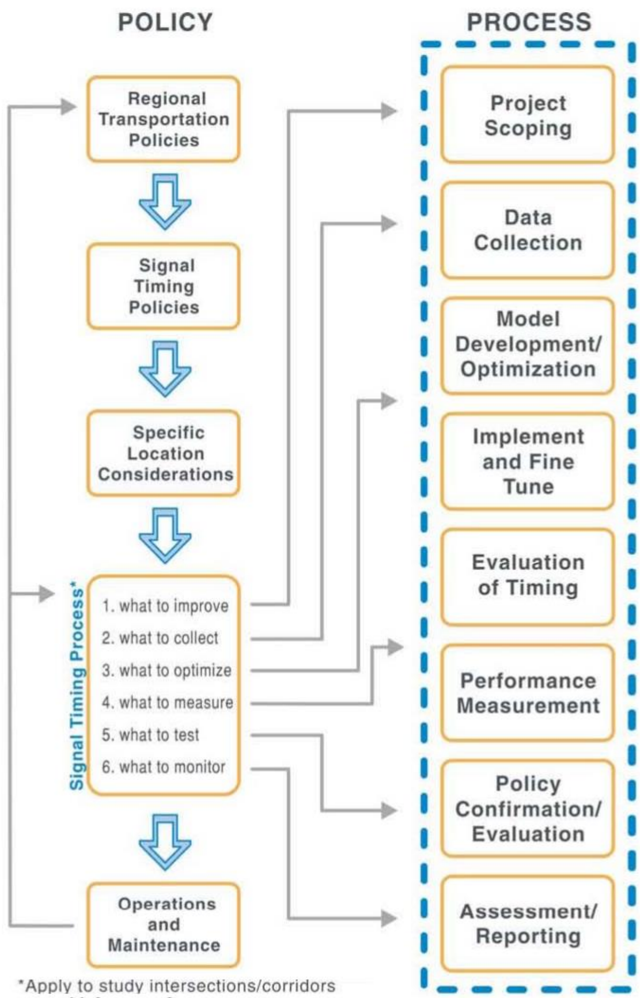


Figure 2-1. Summary of Signal Timing Environment (Administration & Transportation, 2008)

Figure 2-2 presents a diagram illustrating the signal-timing design and maintenance process. The process begins with the initial steps of defining objectives, assessing, and prioritizing activities. This is followed by the assembly of appropriate data to aid in timing and documentation goals. Subsequently, software packages are employed to design the cycle, offsets, and splits for intersections based on performance measures. Once the timing plans are documented, they undergo deployment in the field, and the subsequent stage involves evaluation and maintenance (Day et al., 2010).

In this process, feedback plays a crucial role in improving performance. Figure 2-2 illustrates two feedback arrows, FB1 and FB2. FB1 represents the evaluation of timing plans given by the signal timing software to decide if modifications are necessary. FB2 represents the evaluation of deployed timing plans through field measurements, motorist complaints, and safety records. Traditionally, there is significant quantitative feedback in the modelling and design stages (FB1 arrow) but limited quantitative feedback in the deployment stages (FB2 arrow). This might result in operational shortcomings if not addressed (Day et al., 2010).

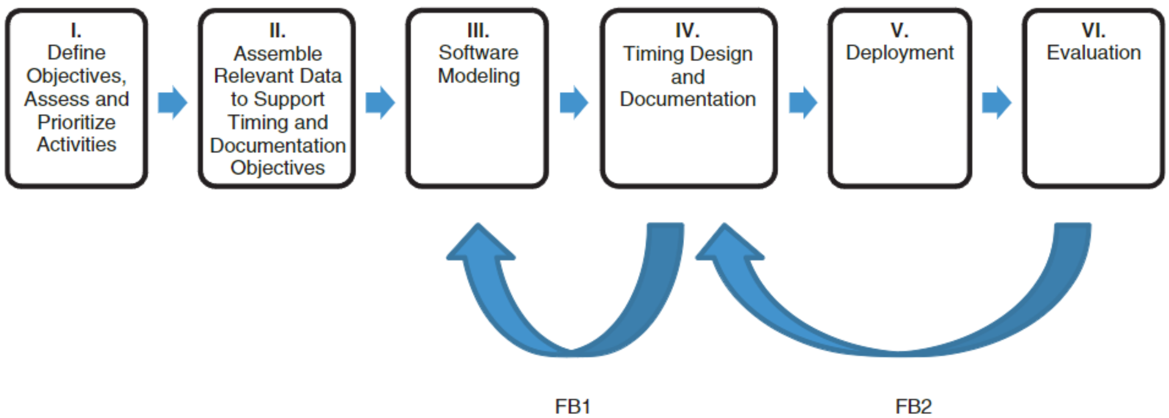


Figure 2-2. Feedback in the traditional signal timing design and maintenance process (Day et al., 2010)

Several weaknesses contribute to these deficiencies. Data collection for the timing process often neglects off-peak and weekend conditions. Timing plans, which are static, do not account for the dynamic nature of the network. These plans are often re-evaluated infrequently, sometimes after several years or in response to user complaints. However, the information provided by system users is often imprecise. Additionally, when new data is collected, performance measures are often estimated using the same tools used for designing signal timings, leading to recurring deficiencies in the settings returned by the software (Day et al., 2010).

2.1.2 Performance-based traffic signal management

Technological advancements have facilitated the measurement of performance metrics in the field more efficiently and frequently. This has led to a shift towards utilizing the feedback from field performance metrics in signal timing practices. The introduction of Automated Traffic Signal Performance Measures

(ATSPMs) has modernized traffic signal management, as highlighted in NCHRP research report 954 on Performance-Based Management of Traffic Signals. This section provides an overview of this innovative approach and examines the challenges associated with its implementation within regular traffic agency practices.

Figure 2-3 illustrates the contrasting steps between traditional signal retiming and ATSPM signal retiming. Performance-based management, in contrast to the traditional approach, empowers staff to proactively monitor the traffic signal system using ATSPMs instead of relying on reactive adjustments. This proactive approach allows staff to monitor trends over time and identify issues before they are reported. When public service requests arise, ATSPMs can be used for rapid information verification and troubleshoot issues without extensive field monitoring. Furthermore, real-time data provided by ATSPMs can be leveraged to assess the impact of signal timing adjustments and determine the need for further modifications. By embracing ATSPMs, transportation agencies can enhance operational efficiency, improve traffic flow, and ensure a safer and more seamless experience for both motorists and pedestrians (National Academies of Sciences, Engineering, and Medicine, 2020).

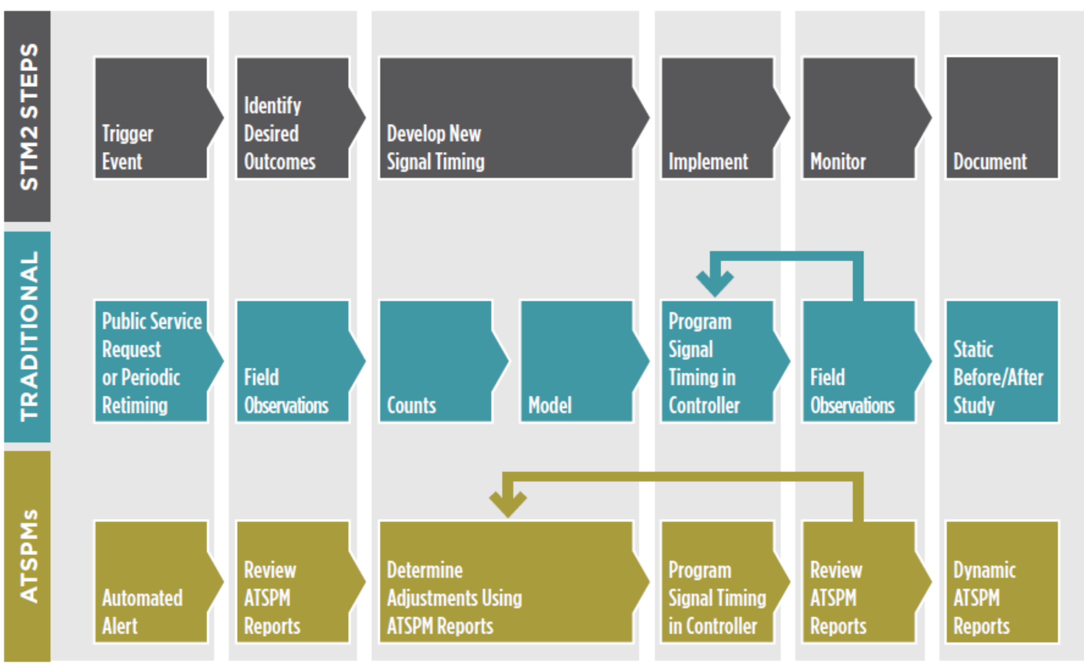


Figure 2-3. Traditional versus performance-based signal timing process (National Academies of Sciences, Engineering, and Medicine, 2020)

The NCHRP report offers a detailed examination of the responsibilities, goals, and advantages associated with every group of stakeholders engaged in the execution of signal performance metrics. It provides clear guidelines for the necessary qualifications of personnel and the required equipment. The activities related to signal performance measures are organized based on staff hierarchy, while the equipment encompasses both physical hardware and software for data collection and analysis. Choosing the right equipment and software for signal performance measures requires careful consideration of the available options, which vary in terms of costs and functionalities (National Academies of Sciences, Engineering, and Medicine, 2020).

The Performance-Based Management of Traffic Signals guidebook focuses on the original ATSPM metrics which are developed using high-resolution controller data. In this method, by recording events up to 10 times per second, ATSPMs enable agencies to consistently monitor maintenance and operations at traffic signals with great precision. Incorporating signal performance measures into the traffic signal system management aligns with the outcome-driven process presented in the earlier NCHRP report, Signal Timing Manual second edition. As shown in Figure 2-4, eight essential steps engaged in the deployment of signal performance metrics as outlined in the guidebook (National Academies of Sciences, Engineering, and Medicine, 2020):

1- **Select performance measures:** The first step focuses on selecting the most important measures based on agency goals and objectives. This should align with an agency's goals, objectives, and signal system management methods. A focused approach is necessary due to the abundance of available metrics. Detailed information on 26 signal performance measures including required inputs, resulting outputs, example applications, and additional references is provided in the guidebook.

2- **Determine implementation scale:** Next, the implementation scale is determined, with options ranging from system-wide deployment to incremental approaches.

3- **Conduct system needs gap assessment:** Once performance measures are chosen, conducting a system needs gap assessment helps identify necessary changes to system components, workforce resources, and business processes.

4- **Procure resources:** Procuring additional resources based on gap assessment is crucial for successful deployment and long-term operations.

5- **Configure system:** Configuring the system involves setting up the hardware and software for data collection and processing at the intersection and system level.

6- **Verify system:** System verification ensures data accuracy and measures calculations. After the installation of the system, it is essential to perform a verification process to ensure consistent and accurate data collection, as well as precise calculation of performance measures. This verification process may involve incorporating external data from separate sensor networks or conducting specific field studies to enhance the accuracy of the measurements.

7- **Apply performance measures:** The valuable insights can be used to trigger signal retiming, detect miss-programmed signal parameters, and identify equipment malfunctions within the traffic signal system.

8- **Integrate into agency practice:** The guidebook offers strategies for fully integrating signal performance measures into agency management practices, fostering collaboration among various stakeholders and effectively communicating the benefits of these measures to executive staff, elected officials, and the public.

This guidebook presents a collection of 26 signal performance measures grouped into five categories: Communication, Detection, Intersection/Uncoordinated Timing, System/Coordinated Timing, and Advanced Systems and Applications. Each measure is accompanied by a detailed description that provides insights into its calculation and display.

The guidebook goes beyond description and explores the practical applications of these measures. It demonstrates how they can be implemented in various contexts to enhance traffic signal systems. The guidebook also identifies the stakeholder groups, including organizational, planning, design and construction, operations, and maintenance teams, who can benefit from utilizing each measure.

To align the performance measures with traffic signal system objectives, the guidebook categorizes them into 10 areas, such as equipment health, vehicle delay, vehicle progression, pedestrians, bicycles, rail, emergency vehicles, transit, trucks, and safety. This categorization enables transportation agencies to assess and improve signal system performance effectively.

In terms of data sources, the guidebook considers six types: controller high-resolution data, central system low-resolution data, vendor-specific data, automated vehicle identification (AVI) data, probe vehicle segment speed data, and automated vehicle location (AVL) data. These data sources provide valuable insights for evaluating performance and making informed decisions.

For each performance measure, the guidebook defines the detection needs, including unmapped, stop bar presence, stop bar count, advance, and radar speed detector. Figure 2-4 from the NCDOT ATSPM Implementation Plan depicts these detector locations for full detection at an intersection. Some measures may require calibration, and the guidebook provides relevant considerations for calibration purposes. References are provided to guide practitioners to additional resources for further information on each measure. Throughout the guidebook, illustrative examples are included to demonstrate the real-world applications of the performance measures.

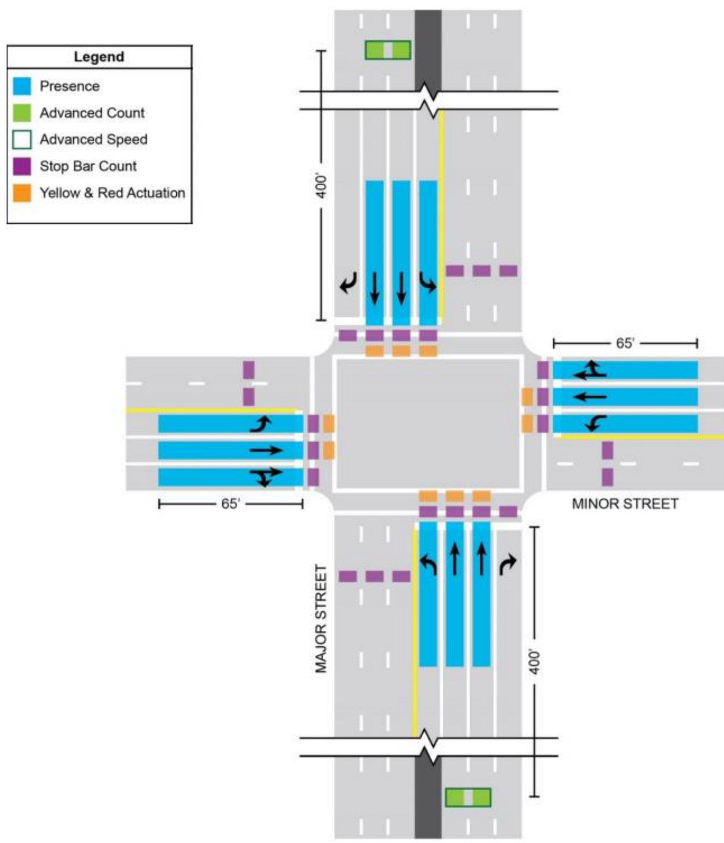


Figure 2-4. Full detection hardware as needed for traditional ATSPM **(State of North Carolina, 2019)**

Detectors have traditionally been used as the primary method for collecting ATSPM data at intersections. However, using detectors for data collection poses challenges. One major challenge is the significant infrastructure investment required. Different types of detectors are needed for different ATSPMs, and they must be installed at each intersection and lane, resulting in high costs and potential disruptions. Ongoing maintenance is also necessary to ensure proper functionality and prevent degradation over time. Moreover, achieving city- or county-wide coverage with detection-communication infrastructures is often impractical due to the extensive requirements (SMATS, 2021).

2.2 Trajectory-based ATSPMs

The significance of monitoring traffic signal performance has been acknowledged for over two decades. Day et al. (2010) critiqued the conventional design process for traffic signal timing, proposing the initial ATSPMs as a proactive approach to signal management. However, the widespread adoption of the original ATSPMs is constrained by the need for extensive physical infrastructure, including detection systems on every approach and communication technology. The costs and maintenance efforts associated with this infrastructure can be substantial (Waddell et al., 2020).

To overcome these challenges, researchers are exploring alternative approaches that leverage emerging technologies like CVs and third-party data providers. These methods enable vehicle trajectory data collection without relying on costly physical equipment (SMATS, 2021).

CVs, including trucks, fleets, personal vehicles, and mobile devices, serve as probe vehicles by providing real-time location and movement data, enabling the measurement of key metrics such as speed and travel time. The common adoption of navigation apps and GPS devices has significantly increased the availability of this probe vehicle data through crowdsourcing. Transportation agencies and commercial entities, such as INRIX, TomTom, HERE, and Traction, archive and provide access to data from probe vehicles, categorized into vehicle trajectory and segment data. Vehicle trajectory data consists of raw GPS locations and timestamps, while segment data aggregates this raw data into speed and performance measures for different road segments, including freeways, arterials, and corridors (Leitner et al., 2022).

Unlike fixed sensors, CVs with GPS devices can gather high-resolution position, travel time and speed information throughout the network, irrespective of traffic flow measurements. CVs have the capability to continuously collect data, providing valuable insights at any time and on any day when GPS-equipped vehicles are in operation (Jenelius, 2013).

Multiple studies have confirmed the viability of using vehicle trajectory data to measure traffic signal performance metrics comparable to traditional ATSPMs. For instance, Saldivar-Carranza et al. (2021) utilized high-fidelity vehicle trajectory data to derive operational performance measurements, including the level of service, split failure, downstream blockage, and quality of progression. However, the accuracy and reliability of these methods depend on the quality and precision of the crowd-sourced data and the extent to which vehicles are equipped with the necessary technology. Therefore, the trajectory data source plays a crucial role in achieving reliable results.

To acquire vehicle trajectory data, various data sources can be considered. Probe vehicles, traditionally used by traffic agencies, are one source for measuring performance metrics. GPS-equipped fleets, such as buses or goods vehicles, can also provide trajectory data. Additionally, navigation software on mobile devices can generate extensive datasets of trajectory data. However, the most precise and accurate data can be obtained from modern private vehicles equipped with GPS units and cellular connectivity, referred to as high-resolution connected car (HRCC) data. HRCC data offers a higher level of accuracy compared to other data sources. The pinnacle of accuracy is achievable through the utilization of true CV technology, which requires dedicated infrastructure on roadways.

With the emergence of connected cars and the recognition of their vast potential in providing comprehensive traffic data, HRCC trajectory data has become a more accurate and cost-effective method for measuring ATSPMs. HRCC data captures precise vehicle location, speed, and heading at frequent intervals, making it an enriched data source for analyzing signal performance. It eliminates the need for expensive hardware and communication infrastructure, making it a cost-effective choice for organizations seeking real-time and historical traffic data. As connected car technology becomes increasingly prevalent, the abundance and precision of available data will continue to expand (SMATS, 2021).

The subsequent section offers an overview of some studies that have utilized trajectory data from different types of CVs to assess performance at signalized intersections. Additionally, some studies showcase the application of proposed performance metrics for ranking intersections, and corridors, or conducting before-after evaluations.

2.2.1 ATSPMs using trajectory data

This section presents a collection of studies that highlight the utilization of trajectory data for evaluating and optimizing traffic signal operations.

These studies showcase various applications and methodologies that leverage trajectory data to enhance our understanding of signalized intersections, arterials, and corridors. Zheng & Liu (2017) propose an approach to estimate traffic volume at signalized intersections, aiding in signal design optimization. Fourati and Friedrich (2019) introduce a method to estimate intersection capacity, providing a more accurate assessment of performance factors. Chen et al. (2020) focus on evaluating green time utilization at signal-controlled intersections, enabling a deeper understanding of signal performance and optimization opportunities.

Using vehicle data to improve traffic signals on corridors has been the focus of many recent studies. Waddell et al. (2020) utilize third-party vehicle trajectory data to evaluate the performance of signalized intersections on corridors. Their study provides valuable insights into delay, stop percentage, and travel time reliability, enabling practitioners to identify issues and prioritize improvements. Saldivar-Carranza et al. (2021) leverage high-fidelity vehicle trajectory data to measure split failures, downstream blockage, and other operational performance measures. Their methodology, including the use of the Purdue probe diagram, offers a comprehensive assessment of signalized corridors, aiding in the identification of capacity challenges and retiming opportunities.

Some studies propose using measured performance to rank arterials or individual intersections. Day et al. (2015b) analyzed and rank travel times on arterials, and Wunsch et al. (2015) focus on ranking individual intersections. Dunn et al. (2019) assess arterial signal performance based on speed reduction. Khattak et al. (2020) evaluated the impact of adaptive traffic control on specific arterials as a before-after evaluation study. Saldivar-Carranza et al. (2023) utilize trajectory-based ATSPMs to prioritize and evaluate signal retiming opportunities at intersections. Collectively, these studies demonstrate the value of trajectory data in assessing and improving traffic signal operations, offering insights into traffic volume estimation, capacity assessment, green time utilization, corridor performance, signal ranking, and retiming prioritization.

Zheng et al. (2017) suggested a method to gauge traffic volume using GPS trajectory data from CVs even when the market penetration rates are low. Vehicle arrivals at signalized intersections are modeled as a time-dependent Poisson process, accounting for signal coordination. The estimation problem is formulated as a maximum likelihood problem, and an expectation maximization (EM) process is developed to solve it. The estimation algorithm takes into account both the vehicle trajectories approaching an intersection and the traffic signal status. Two case studies using CV data from the Safety

Pilot Model Deployment project and vehicle trajectory data from a commercial navigation service are conducted to validate the estimation algorithm. The results show reasonable accuracy compared to manually collected and loop detector data. The mean absolute percentage error (MAPE) of the estimation ranges from 9% to 12%, indicating reasonable accuracy. The proposed approach has the potential to assist traffic management agencies in evaluating and operating traffic signals with limited CV data, paving the way for future signal operations without the need for detectors.

Fourati and Friedrich (2019) proposed a method to measure capacity at intersections using trajectory data. Both model-based estimation and direct measurement methods for measuring capacity require significant effort to gather relevant data, and they only provide a snapshot of capacity at a specific moment in time. However, they suggest a different method to determine the capacity of signalized intersections over time using trajectories from probe vehicles. This capacity evaluation can be employed to gauge the degree of saturation based on real demand or to analyze various possible scenarios related to demand or signal timing. By accumulating trajectories within the cyclic operation of intersections, this approach compensates for potentially low penetration rates. The method uses the existing methods to determine the cycle time and approach green time then the Saturation Flow Rates (SFR) directly derived from the saturation time headway using two parameters for calibration. The experiment conducted using a commercial dataset at an intersection in Munich demonstrated precise estimation of signal timing and saturation flow values that were similar to those calculated according to the German guideline.

Chen et al. (2020) focus on the evaluation of green time utilization at signal-controlled intersections, which serves as the basis for signal optimization. Traditionally, the most common input for this evaluation has been traffic volume data from fixed detectors, which may suffer from accuracy and availability issues. In their study, a novel approach is proposed that utilizes trajectory data instead of volume data. Three indicators based on stop-wave and dissipation-wave information are introduced to comprehensively assess the utilization of green time. Chen et al. (2020) conducted a field case study to demonstrate the effectiveness of the proposed method, and a sensitivity analysis to illustrate its applicability. The findings indicate that maintaining a 2-hour database at a penetration rate of 10% ensures a mean absolute percent error (MAPE) below 10% for the overall green time utilization rate, indicating stable performance. Similar results can be achieved with a 5% penetration rate by utilizing a 4-hour data interval.

Existing methods for evaluating green time utilization include volume-to-capacity-based and occupancy-based approaches. The commonly used volume-to-capacity ratio (v/c ratio) method relies on aggregated traffic volume data collected by fixed detectors, while the green occupancy ratio (GOR) method utilizes phase duration and detector occupancy data. Chen et al. (2020) highlights the shortcomings of these methods such as the need for detectors, the maintenance of detectors and also occupancy ratio dependency on the detector geometry, traffic composition and vehicle speed. In contrast, using trajectory data, the green time utilization rate can be measured without relying on volume information. Furthermore, the detailed vehicle information available in trajectory data allows for a more comprehensive understanding of the specific portions of green time that are not fully utilized.

Chen et al. (2020) demonstrated a comprehensive evaluation of pre-timed traffic signals using three quantitative indicators. By relying solely on low penetration rate trajectory data and signal timing, they used a superimposition technique to overlay trajectory data from multiple cycles to estimate the green time utilization. Considering both the overall and specific utilization of green lights, their approach offers a novel and insightful method for evaluating green time utilization at signal-controlled intersections.

Waddell et al. (2020) use third-party vehicle trajectory data to evaluate the performance of 136 intersections on ten signalized corridors in Columbus, Ohio state in the US. High-level corridor summary metrics, including average percent of vehicles stopping, average delay, and level of travel time reliability, were calculated for each corridor direction. Additionally, intersection-level metrics were introduced to help practitioners identify issues, improve signal timings, and prioritize infrastructure investments.

The authors utilized probe trajectory data obtained from the Ohio Department of Transportation (ODOT) which represented approximately 2% of the total traffic volume, with an equal distribution between light-duty and medium-duty vehicles. The trajectory data included latitude and longitude waypoints for trips in the selected corridors, but with varying reporting frequencies (pings). The data filtering process focused on vehicles with ping frequencies of 60 seconds or less, and only vehicles that reported data on both the downstream and upstream sides of an intersection were included in the analysis. Intersection-level metrics, such as average delay (in seconds), average stop percentage, and average level of travel time reliability (LOTTR), were calculated to assess signal performance in 2-hour intervals on all weekdays in 2017 (Waddell et al., 2020).

The study demonstrates the advantages of using probe vehicle trajectory data, such as scalability nationwide with limited effort and non-reliance on permanent infrastructure. It concludes that the low penetration rate necessitates the stacking of data to generate these performance measures which restricts their potential as a complete substitute for active management (Waddell et al., 2020).

In contrast to the focus of other studies on travel time and delay metrics derived from trajectory data, Saldivar-Carranza et al. (2021) took a different approach. They utilized high-fidelity vehicle trajectory data to measure a range of performance indicators, including split failure (SF), downstream blockage, quality of progression, arrival on green (AOG), and level of service (LOS). By leveraging this comprehensive set of metrics, the study aimed to gain deeper insights into the operational aspects of signalized intersections. Their research presents a case study conducted on an eight-intersection corridor in Indianapolis, where the performance of four different timing plans was evaluated. This study utilized a large dataset consisting of over 160,000 trajectories and 1.4 million GPS samples collected on weekdays in July 2019 from 5:00 a.m. to 10:00 p.m. The trajectory data included latitude and longitude waypoints, with a ping frequency of 3 seconds between data points. The research introduces a new visualization tool called the Purdue probe diagram (PPD), which offers a holistic view of a vehicle's experience along the corridor. The PPD allows for the assessment of various operational performance measures, including arrivals on green, split failures, and downstream blockage. By utilizing the PPD, agencies and researchers can quickly assess the proportion of vehicles arriving on green, identify locations with an insufficient allocation of green time, and understand the impact of downstream intersection spillback.

This study aimed to showcase the effectiveness of utilizing probe vehicle trajectory data for assessing traffic signal performance measures, including split failures, downstream blockage, quality of progression, and traditional level of service metrics outlined in the Highway Capacity Manual. The research concludes by discussing the suitability of probe vehicle trajectory data for corridors with varying traffic volumes and proposes a methodology applicable to corridors with average daily traffic of approximately 15,000 vehicles per day on the mainline approaches. The paper also highlights the potential for cloud-based implementation of these techniques. Overall, this study demonstrates the value of high-fidelity vehicle trajectory data in analyzing and visualizing the operational performance of signalized corridors, emphasizing the capabilities of the Purdue probe diagram as a valuable tool in this context (Saldivar-Carranza et al., 2021).

Day et al. (2015b) introduced a methodology to analyze and rank travel times on a series of 28 arterials comprising 341 signalized intersections in Indiana, USA. The data consisted of minute-by-minute speed records, which were aggregated into 15-minute bins. The mean value of the bins represented the measured travel time for each segment during that duration. The study divided a typical day into morning peak, midday, and afternoon peak intervals to align with the commonly observed time-of-day signal timing patterns on arterials with coordinated signal systems. Measured travel times require normalization to account for variations in path lengths and ideal speed characteristics of the corridors. This is achieved through two methods: calculating the travel rate, which is the average travel time divided by the corridor distance, representing the reciprocal of speed, or computing the normalized travel time based on ideal speed involves dividing the average measured travel time by the optimal time corresponding to the speed limit. The authors found that both metrics were proportional but chose ideal-speed normalized travel time as a better metric for ranking different corridors. They also evaluated the normalized reliability of travel time calculated as the ratio of the standard deviation of actual travel time to the speed limit travel time. A ranking index for arterials was developed using a weighted combination of both the average travel time and its unreliability. The study revealed that arterials with more traffic signals exhibited higher average travel times and less reliability.

This pilot study focused on analyzing arterial mobility by examining travel times on 28 arterials and ranking them using a composite index based on trajectory data. Nonetheless, it is crucial to highlight that the study solely relied on data from Wednesdays for evaluation without conducting any subsequent validation. Furthermore, the authors did not explicitly showcase any practical application or use case for their findings (Day et al., 2015b).

While Day et al. (2015b) demonstrated arterial ranking utilizing the trajectory data, Wunsch et al. (2015) ranked individual intersections in a large area in Bavaria, Germany. The authors focused on determining delay times and path-specific travel times on road segments including 2300 traffic signals. By mapping the probe data to intersections, they calculated key performance indexes (KPIs) for ranking purposes. The total delay, average delay, and travel time index were considered as ranking KPIs. Total delay was derived by comparing the cumulative delays experienced by vehicles at an intersection to the free-flow speed. The average delay was calculated by dividing the total delay by the number of vehicles, while the travel time index represented the ratio of observed travel times to free-flow speed travel times.

The study identified limitations related to the calculation of reference speed and travel time for paths, which could result in over or underestimating delay. Additionally, the study did not address the scenario of nearby signals, where parts of paths from one signal could overlap with paths from other signals, leading to potential misallocation of delay. To overcome this, the authors suggested analyzing block signals together by adjusting the circle radius threshold and defining paths across multiple junctions. Despite these limitations, the study confirmed the reliability of using Floating Car Data (FCD) analysis to evaluate intersection performance to optimize traffic signals (Wunsch et al., 2015).

One of the main applications of ranking arterials or individual intersections is to consider them for prioritizing signal retiming schedules conducted by traffic agencies. This utilization ensures that the most crucial areas receive priority during regular maintenance and signal adjustment activities. Dunn et al. (2019) conducted a study using segmented probe vehicle speed data to assess the performance of 1026 traffic signals along 79 corridors in Austin, Texas, for the purpose of signal retiming. The data, covering 87% of the area, was obtained from a third-party vendor and included average vehicle speeds over road segments at minute-by-minute intervals. The data was aggregated into 15-minute bins for three time-of-day periods: morning peak, midday, and evening peak. The ranking process utilized three metrics: the percentage of the corridor experiencing any slowdown, the percentage of the corridor experiencing a slowdown greater than 3 miles per hour (MPH), and the maximum slowdown within each corridor segment. The ranking process involved evaluating all corridors based on the three metrics and three time-of-day periods for the worst-performing direction in each corridor. The final ranking was determined by averaging the rankings across the three time periods.

The study aimed to validate a ranking methodology used for selecting corridors to be retimed by the City of Austin. Out of the 27 corridors scheduled for retiming, only seven were included in the ranking methodology's selection. The remaining 20 corridors, which were not part of the ranking selection, were generally ranked low and not prioritized for retiming. Evaluating the quantitative value of the ranking methodology was challenging due to its hypothetical nature. Nevertheless, the research evaluated the scope for enhancement in both corridor sets by contrasting average segment travel times during various periods of the day. Findings revealed that corridors identified by the ranking method displayed a substantially greater scope for enhancing travel time compared to those determined by the City's timetable. On average, the set chosen via ranking exhibited a potential for travel time improvement that was 96% higher. Such results underscore the proficiency of the ranking approach in gauging corridor performance and pinpointing those with a more pronounced potential for refinement (Dunn et al., 2019).

Performance measurement metrics can also be utilized in before-after evaluation studies, particularly when alterations such as traffic signal retiming occur. These metrics serve as valuable tools for assessing the impact and effectiveness of the changes made to the system. By comparing the performance metrics before and after the alteration, transportation professionals can determine the effectiveness of signal timing and make informed decisions based on the observed outcomes. Khattak et al. (2020) conducted a study on the performance of scalable urban traffic control (SURTRAC) intersections in Pittsburgh, Pennsylvania, using real-world GPS floating car runs and INRIX probe vehicle data. The study focused on evaluating the impact of adaptive traffic control on various performance measures, such as travel time, speed, number of stops, volatility, and planning time index. Results indicated that adaptive control had positive effects on speed, travel time reliability, and the planning time index. The study emphasized the benefits of using multiple data sources and performance measures, demonstrating the potential of SURTRAC for managing congestion in signalized urban arterial networks. The findings highlighted the improved performance of the ASCT system and its decentralized algorithm, providing valuable insights for planning agencies and engineers. Additionally, the study considered the measure of volatility, examining variations in speeds, accelerations, and decelerations to understand how ASCT influences instantaneous driving over time.

Saldivar-Carranza et al. (2023) conducted a recent study where trajectory-based ATSPMs were employed to prioritize and evaluate signal retiming opportunities at over 100 intersections. The study analyzed more than 400,000 trajectories and 6,800,000 GPS data points to identify critical movements with high split failures (SFs) and donor movements that could potentially redistribute split to the critical movement. The methodology involved the use of a Relative Performance Diagram (RPD) to assess split failures and downstream blockage, providing a scalable approach for identifying retiming opportunities without requiring vehicle detection or communication tools.

Based on the findings, three intersections were selected for retiming. After implementing the retiming process, a post-retiming review showed significant improvements in signal performance measures. These improvements included up to a 30% reduction in split failures, a decrease of 53 seconds per vehicle in control delay, and a 21% increase in arrivals on green for the benefited movements. Notably, the techniques used in this study can be applied to any intersection where CV data are available, allowing for prompt identification of capacity challenges and the tactical deployment of retiming resources to improve signal operations (Saldivar-Carranza et al., 2023).

Green Light, an initiative by Google Research, employs artificial intelligence (AI) to optimize traffic light patterns, aiming to reduce vehicle emissions and improve urban mobility. The program identifies existing traffic light parameters and uses Google Maps data to model traffic flow, offering city traffic engineers AI-generated recommendations to enhance timing configurations, potentially reducing stops by up to 30% and greenhouse gas emissions by 10%. Implemented across 70 intersections in 12 cities, it's designed to create "green waves" to minimize stop-and-go traffic without requiring additional hardware. The service is provided at no cost during its early research phase, ensuring user data privacy and using aggregated data to improve traffic flow for all road users (Google Research, n.d.).

The Green Light initiative by Google, which currently resides in the research phase, does not extensively disclose the intricacies of its algorithm or the precise methods it employs. The information provided on their official site indicates that the initiative utilizes data from Google Maps to infer the phases and timing of existing traffic signals. This approach intimates a level of opacity concerning the algorithm's detailed workings, which remains within the proprietary domain of Google's research endeavors. Such a methodology, while not entirely transparent, holds potential efficacy for enhancing traffic management and reducing emissions in urban environments.

2.2.2 Summary

ATSPM is a key component of performance-based management of traffic signals, which emphasizes the use of objective performance metrics to guide decision-making and signal optimization efforts. By analyzing the trajectory data collected from various sources, such as probe vehicles, fleet vehicles, crowd-sourced GPS data, high-resolution connected cars, and true CV, traffic agencies can gain insights into the efficiency and effectiveness of signalized intersections.

The use of trajectory data in ATSPM offers several advantages. It provides a comprehensive view of traffic patterns, enabling a more accurate assessment of signal performance and identification of congestion hotspots. Moreover, trajectory data allows for the evaluation of signal timing strategies and the estimation of travel times and delays experienced by different user groups.

By leveraging a diverse range of trajectory data sources, traffic agencies can enhance their understanding of signal performance and make data-informed decisions to optimize signal timing, improve traffic flow, and mitigate congestion. These advancements in traffic signal performance measurement provide a solid foundation for evidence-based and proactive management of signalized intersections.

2.3 Research gaps

Several studies have proposed performance metrics, either as standalone metrics or combinations of multiple metrics, to assess and rank arterials or individual intersections. Some studies focus solely on introducing new metrics without further application, while others demonstrate practical implementations, such as ranking arterials or intersections based on the proposed metrics. Additionally, performance metrics have been utilized to evaluate the impact of alterations, such as the adoption of adaptive signal control, through before-after studies. One clear objective of performance measurement is to assist agencies in prioritizing their signal retiming schedules, enabling them to allocate their efforts effectively.

The currently used metrics largely assess the performance of signal control in relation to the balance between demand and capacity yet fall short of predicting whether revising signal timing plans could lead to improvements. The identification of poorly performing locations does not guarantee an improvement through signal retiming. For example, if the capacity is insufficient, the retiming process may not bring about substantial benefits. This limitation poses a significant disadvantage in the current performance metrics identified in the literature review. The existing literature offers a plethora of metrics and graphical representations that effectively capture existing performance, thus assisting in identifying bottlenecks and congestion points. But, when there are multiple congested locations, especially in large cities, traffic agencies often grapple with the efficient allocation of their limited resources to areas where retiming would yield the greatest benefits. While some studies propose the use of a single metric or a combination of metrics as an index to prioritize signalized intersections, this approach has its drawbacks. A prevalent challenge is that previously developed indices often mirror overall congestion levels, hence prioritizing intersections in high-traffic or densely populated areas. These, however, might not necessarily be the locations presenting the highest potential for improvement. Consequently, traffic practitioners may find themselves focusing on heavily congested intersections already operating at or near their capacities - where significant improvements are unlikely. Conversely, intersections with less overall traffic might offer a higher potential for efficiency improvement. To bridge this gap, research is needed to develop metrics that can accurately predict the benefits of retiming.

Saldivar-Carranza et al.'s study from 2023 is a notable exception as it seeks to predict potential improvements by considering the signal timing plans. They propose utilizing the CV-based split failure ratios of critical and conflicting movements on a two-dimensional graph. This allows for the identification of intersections characterized by a high split failure ratio on a critical movement and a low split failure ratio on a conflicting movement. In these situations, the conflicting movement could potentially relinquish its green time to the critical movement. The straightforwardness of their method is beneficial, but it does have a certain drawback: split failures only arise in extreme situations, and therefore it would not detect potential improvement opportunities in the absence of high split failures.

Addressing these issues requires a more targeted and precise approach that can accurately identify intersections where signal retiming can have the most significant impact. In the following chapters, a novel methodology is proposed, aiming to fill this gap and provide traffic agencies with a tool to efficiently prioritize signal retiming efforts based on true improvement opportunities.

Chapter 3: Methodology

This chapter describes the proposed methodology for estimating the signal sub-optimality using CV trajectory data. It explains how to calculate the Traffic Signal Sub-Optimality index (TSSO) and discusses the data required for evaluation, including real connected car data and microsimulation. The chapter also includes explanations of the accuracy and sensitivity analysis, examining the impact of various factors on the TSSO index.

Resolving the problems mentioned in the previous chapter requires a precise strategy that correctly pinpoints intersections where signal retiming could yield substantial improvements. This study suggests a novel methodology to tackle this challenge, offering traffic agencies a practical tool to efficiently prioritize their signal retiming efforts based on the greatest improvement opportunities. This methodology entails the development of a framework for estimating the TSSO index. This framework, complete with its assumptions, formulas, and calculations, is designed to measure traffic signal suboptimality by estimating the delay that could potentially be reduced by signal retiming. The trend of the index value can then be used to determine the necessity of signal retiming at a given intersection.

3.1 Traffic Signal Sub-Optimality index (TSSO)

The process of determining the TSSO involves five steps, which are depicted in Figure 3.1. The initial step involves analyzing trajectory data from CVs to establish the average vehicle delays for every movement at the intersection. In Step 2, the degree of saturation of each movement is estimated using the back-calculation method and the Canadian Capacity Guide (CCG) delay model. The third step uses formulas based on HCM signal optimization methodology to calculate an optimal signal plan. Step 4 utilizes the green times from the previous step to estimate the new degree of saturation for each movement, and then employs the delay model to calculate the average vehicle delays for each movement under the hypothetical optimal signal condition. Finally, Step 5 involves calculating the total reduction in average vehicle delay at the intersection level, which is reported as the index value (in seconds).

The first step in the process involves analyzing trajectory data from CVs to determine the average vehicle delays for each movement at the intersection. CV data provides GPS coordinates of the vehicles with time stamps for every point and must be processed based on the intersection's geometry and location to extract travel times and delay values for each movement. The acquired delay values are then aggregated over a selected interval period, such as 15 minutes, to calculate the average values. We assume that the trajectories are accurate to the lane level and refreshed every 2 seconds which matches the current available data from High Resolution Connected Cars.

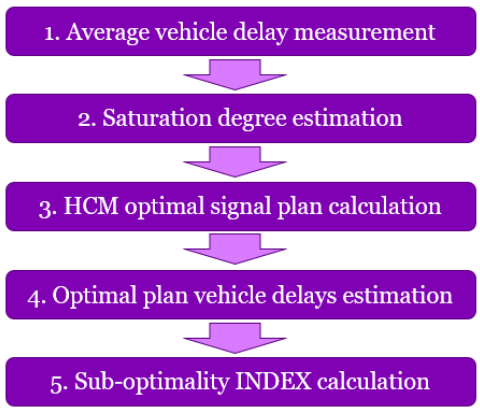


Figure 3-1. The proposed framework for calculating the signal sub-optimality index

In the second step, the CCG delay model were utilized to estimate the degree of saturation of each movement based on the average delay value obtained from CV data. The CCG delay expression, which includes uniform and overload delay, is shown in Equation 3-1.

$$delay = k_f d_1 + d_2 \qquad (3\text{-}1)$$

where

$$d_1 = \frac{c\left(1 - \frac{g}{c}\right)^2}{2\left(1 - x_1 \frac{g}{c}\right)}$$

$$d_2 = 15t_e \left[(x - 1) \sqrt{(x - 1)^2 + \frac{240x}{Ct_e}} \right]$$

where d_1 is the deterministic delay (seconds), c is the cycle time (seconds), g is the effective green time (seconds), x_1 equals *Minimum (x, 1)*, d_2 is the overload delay (seconds), x is the degree of saturation, C is the capacity (vehicles per hour), t_e is the evaluation time (minutes), and k_f is the progression factor.

From Equation 3-1, we can derive the degree of saturation, represented as x. The current signal timing plan defines the effective green time and cycle time. Typically, a 15-minute evaluation period is used. For independent intersections with unpredictable incoming traffic, a progression factor of one is assumed. To streamline the problem, we assume that each movement has just one phase of discharging. With this assumption in place, the capacity can be calculated as a proportion of the saturation flow rate, determined by the effective green time to cycle time ratio. Inserting these assumptions into Equation 3-1 converts it into a delay model equation associated with the degree of saturation. Given the model's non-linear character, techniques like the golden section search can be utilized for its back-calculation.

The process of determining the saturation flow rate is complex, often influenced by a variety of fluctuating road and traffic conditions unique to each location. In practice, a typical approach is to start with a conventional value, such as 1900 vehicles per hour per lane(vphl), which suggests an average saturation headway of approximately 1.9 seconds. This initial rate is then refined to reflect actual conditions using various adjustment factors for considerations such as lane width, pedestrian traffic, and vehicle composition, as advised by design guidelines (NIATT Lab Manual, n.d.).

Beginning with a baseline of 1900 vphl for the saturation flow rates (SFR) of both through and left-turn movements, a heuristic approach involving trial and error is employed to fine-tune these figures. Given the complete set of vehicle trajectory data, it's feasible to ascertain real vehicle counts and average delays for each movement in set intervals, for instance, every quarter of an hour. A visual assessment is conducted using a graph that plots the average delays against vehicle counts as points around the delay model curve, allowing for a qualitative evaluation of the initial SFR assumptions on the model accuracy. This visual method is applied to both left and right turns to incrementally adjust the SFR from the initial estimate. An illustrative example of such a plot used for SFR refinement is depicted in Figure 3.2. This figure provides a graphical representation of the SFR refinement process by comparing the delay model curve against measured delays as points, before and after SFR adjustments. The SFR parameters for TH movements have been adjusted from the initial estimate of 1900 vphl to 2000 vphl, and for LT movements to 1500 vphl.

In this research, the objective is to modestly customize the SFR values to better align with the simulated conditions, enhancing accuracy but not to the extent of striving for precise and definitive SFR determination. This tailored approach acknowledges the hypothetical nature of this method, understanding that exact replication in real-world scenarios is infeasible. Moreover, subsequent sensitivity analysis on the SFR reveals that precise adjustments have minimal impact on the overall outcomes, confirming that a highly detailed SFR estimation is unnecessary at this stage.

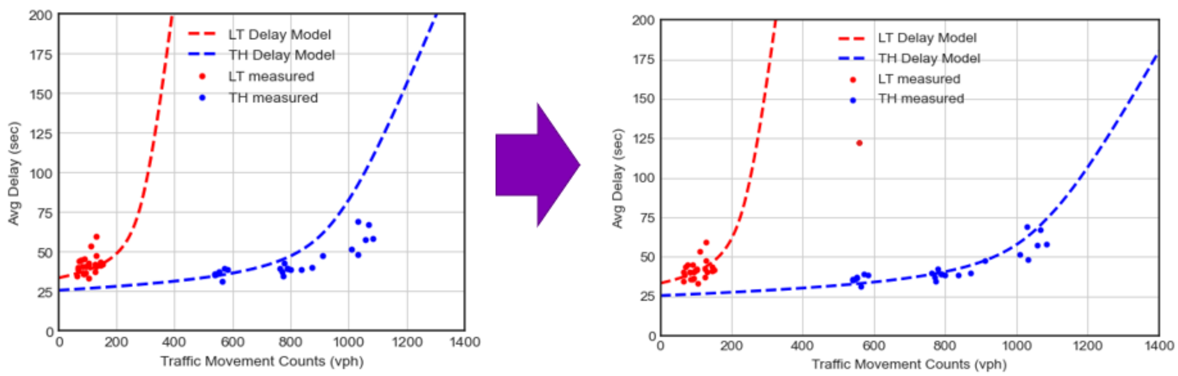


Figure 3-2. Refining the SFR parameters on the vehicle delay model

Step 3 involves determining the optimal signal plan based on the current traffic conditions. The degrees of saturation estimated in Step 2 are used to calculate the flow ratio for each movement, and the optimal signal plan is determined based on the estimated flow ratios in each phase. To simplify the problem, we maintain a consistent signal phase configuration, focusing solely on optimizing the green time for each phase. The allocation of green time is calculated based on the flow ratio of each movement relative to the total sum of flow ratios for critical movements. The optimal green times are calculated using Equations 3-2, 3-3 and 3-4, based on the optimal signal timing approach recommended by HCM.

The flow ratio for movement j, y_j, is:

$$y_j = x_j \frac{g_j}{c} \tag{3-2}$$

where x_j is the saturation degree, and g_j is the effective green time (seconds) for movement j, and c is the cycle time (seconds).

The optimal green time for movement j, g_{j_opt}, is:

$$g_{j_{opt}} = g_t \frac{y_j}{Y} \tag{3-3}$$

where g_t is the total green available for splits (seconds), y_j is the flow ratio calculated from Equation 3-2, and Y is the sum of all critical flow ratios.

The optimal cycle time, c_{opt}, is:

$$c_{opt} = \frac{1.5L+5}{1-Y} \tag{3-4}$$

where L is the total lost time (seconds), and Y is the sum of all critical flow ratios.

In Step 4, the average vehicle delays under the optimal signal plan are estimated. The delay model (Equation 3-1) from Step 2 is used to estimate the vehicle delays for each movement, with the optimal signal plan green and cycle times as input parameters. The new degrees of saturation are estimated from their previous values, considering the effect of green time changes. Equation 3-5 shows the calculation of new saturation degrees under the optimal plan.

The new saturation degree of movement j, x_{j_new}, is:

$$x_{j\_new} = x_j \frac{g_j * c_{opt}}{g_{j\_opt} * c} \tag{3-5}$$

where x_j is the saturation degree, g_j is the green time (seconds), g_{j_opt} is the optimal green time (seconds), c_{opt} is the optimal cycle time (seconds), and c is the cycle time (seconds).

Finally, Step 5 involves calculating the index value based on the measured vehicle delays and estimated delays under the optimal signal plan to determine the sub-optimality of the signal. Equation 3-6 is used to calculate the final index value, which represents the total delay reduction achievable by the optimal signal plan. Therefore, a higher index value suggests that the signal suboptimality caused greater delays, which can be mitigated by signal retiming (i.e., the intersection is more critical to retime). If the right-turn movement lacks a dedicated phase, the delays it causes can be omitted from the index calculations. This five-step framework is repeated for every time interval to calculate a series of index values over the period under consideration.

The sub-optimality index value, TSSO, is:

$$TSSO = \sum_j \left(measured\_delay_j - estimated\_delay_j \right) \quad (3\text{-}6)$$

where $measured\_delay_j$ is the measured average delay for movement j (seconds) and $estimated\_delay_j$ is the estimated delay for movement j under the optimal signal plan (seconds).

It should be noted that other types of indices could potentially be defined, such as one using delay weighted by estimated movement flow or CV volume, which could be explored in future research.

3.2 A simulation-based study - framework

To evaluate the proposed method, it is necessary to obtain a reliable dataset that could serve as a benchmark for comparisons. However, real-world CV datasets often consist of data from a small sample of vehicles on the road, which cannot be used as ground truth for benchmarking. To address this challenge, a microsimulation-based approach was employed to generate a simulated CV dataset along with ground truth. Trajectory data from all the vehicles within the simulation was exported, enabling the performance of the proposed method to be evaluated across varying market penetration rates. The results were compared against a comprehensive ground-truth dataset comprising all vehicles. This approach permitted the effective validation of the accuracy and robustness of the proposed methodology.

The evaluation process is outlined in detail in Figure 3-3, which provides a visual illustration of the key steps involved in assessing the performance of our proposed approach through VISSIM simulation.

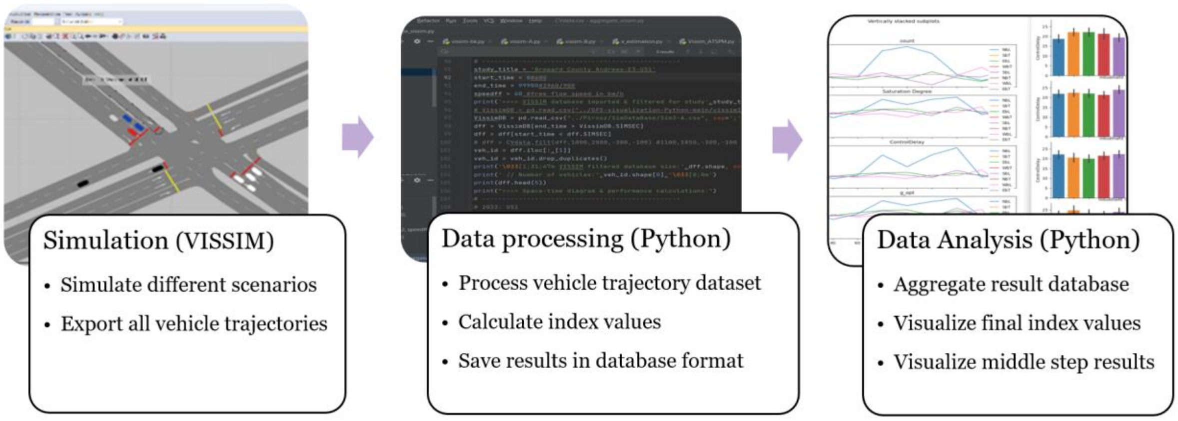


Figure 3-3. Steps for evaluating the sub-optimality index with microsimulation

A 4-leg intersection, featuring two main lanes and one pocket lane for each turning movement, was considered to simulate a typical urban intersection, as illustrated in Figure 3-4. The North-South Road is deemed the major street, whereas the East-West Road is regarded as the minor street. An eight-phase standard ring barrier fixed-time traffic signal plan with protected left turn phases was implemented, as illustrated in Figure 3-5. The trajectory data of all vehicle movements were exported to a CSV database, which included the vehicle IDs and (x, y) coordinates of their trajectories. This permitted the evaluation of results from different penetration rates and facilitated their comparison with the ground truth results using all vehicle data.

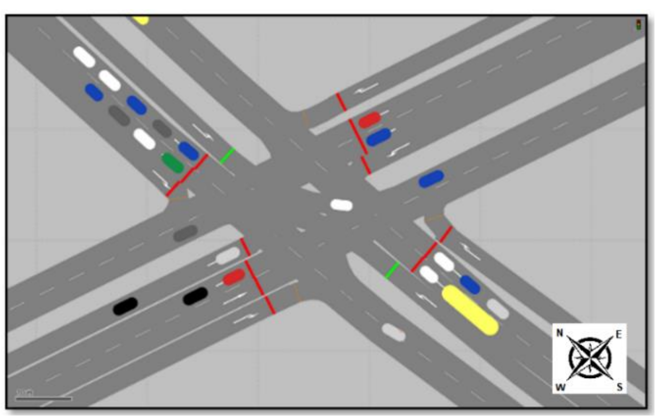


Figure 3-4. The 4-leg signalized intersection simulated in VISSIM

For the second step in the evaluation process, a Python code was developed to implement the proposed method of index estimation for the simulated CV dataset. The code includes the five-step framework described earlier, examining the trajectory data of all vehicles in the simulated dataset. To account for the adjusted penetration rate, a random selection step is included after calculating the delay for each vehicle. The calculated index values using all vehicle trajectory data are considered the ground truth for comparison against different market penetration rates, which only use sampled vehicle data. During each interval, the calculated suboptimality index value and the corresponding optimal signal plan's green times were saved and stored in a designated database.

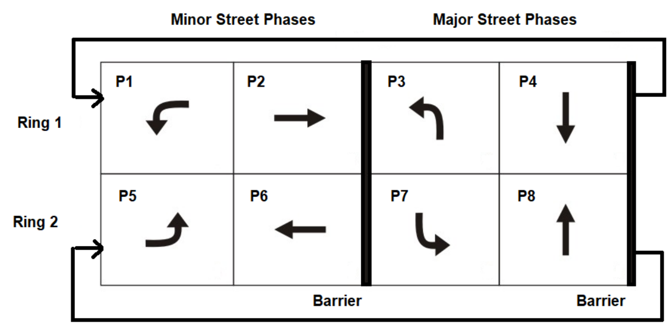


Figure 3-5. The RBC traffic signal plan with protected left turn phases

In the final stage of the evaluation, a Python code was employed as a tool for visualizing and examining the results from the proposed method, which includes the intermediate results of the steps described in the framework. The Python code was used to explore the estimated index values and optimal signal plan green times stored in the database for each interval of the simulation period. Utilizing this tool enabled the gaining of insights into the performance of the proposed method, allowing for the investigation of trends and patterns in the results obtained from different market penetration rates and traffic scenarios.

3.3 Scenarios for VISSIM simulations

The proposed methodology was evaluated through a series of scenarios designed and simulated via the VISSIM microsimulation software. The initial stage involved simulating two distinct scenarios: one dealing with an increase in demand and the other assessing alterations in turning movement ratios without an overall change in approach demand. This was followed by six additional scenarios meant to represent alterations in traffic demand across competing movements. The purpose of these simulations was to gauge the performance of the TSSO index across varying conditions and ascertain its effectiveness in identifying instances where signal retiming could lead to substantial benefits.

Table 3-1 provides a detailed view of the setup parameters universally applied across all scenarios. This includes assumptions like SFRs used in the delay model, as well as selected values for data aggregation interval and simulation parameters. Table 3-2 describes the specific traffic demand parameters employed in each scenario.

Table 3-1. Simulations and index calculations parameters

Parameter	**Definition**	**Value**
Te	Delay model evaluation period	15 min
Sf_TH	Through movement SFR	2000 vph
Sf_LT	Left turn movement SFR	1500 vph
Kf	Progression factor	1
Tint	Data aggregation interval	30 min
Period	Simulation time	4 hours
Time_step	Vehicle’s trajectory sampling rate	2 sec
CL	Cycle time	96 sec
gTH	Through green time	25 sec
gLT	Left turn green time	15 sec
amber	Amber time	3 sec
All_red	Red clearance time	1 sec
Des_speed	Approach posted speed	50 km/h

In Table 3-2, different traffic volumes were used to simulate varying scenarios. The approach traffic volumes of 500, 1000, and 1300 vehicles per hour (vph) were labeled as low, medium, and high traffic, respectively, while 1450 vph is nearing the full capacity of the approach. These benchmarks were crucial for examining the changes in traffic across all the scenarios presented. For the second scenario, a focus was placed on the change in the ratio of left-turning traffic to approach traffic, which increased from 17% to 27%, all while keeping the approach traffic steady at 1000 vph, a medium level. Furthermore, for the examination of variations in traffic demand, shifts in traffic direction were simulated. This involved a consistent increase in Northbound traffic coupled with a corresponding decrease in Southbound traffic.

Table 3-2. Traffic demand variation scenarios

#	Name	Description
1	All approach increased	Overall traffic increased, yet turning movement ratios stayed the same.
2	Major LT-increased, Minor Medium	Left turn traffic on the major street rose while all approach traffic stayed at a medium level.
3	Major Increased, Minor Low	Traffic volume on the major street went up, but it remained low on minor streets.
4	Major Increased, Minor Medium	While the major street saw more traffic, the minor streets sustained a medium volume.
5	Major Increased, Minor High	The major street's traffic volume climbed, with the minor streets experiencing high traffic.
6	Major Shift, Minor Low	Traffic flow changed direction from Southbound to Northbound on the major streets, with minor streets holding a low volume.
7	Major Shift, Minor Medium	A similar directional shift on the major streets, but this time with the minor streets at a medium volume.
8	Major Shift, Minor High	Once again, the major street traffic shifted direction, but with high traffic on the minor streets.

The research presents eight traffic scenarios for evaluation:

1. **Scenario #1:**

Traffic volumes at each intersection approach were increased at four distinct levels: 500 vph, 1000 vph, 1300 vph, and 1450 vph. It's essential to clarify that the examined ratios of left turns, through movements, and right turns (LT/TH/RT) were consistent across varying traffic volumes. Specifically, for a cumulative approach volume of 1450 vph, the distribution was composed of 100 vph for right turns, 1100 vph for straight-through movements, and 250 vph for left turns.

2. **Scenario #2:**

This design mirrored a shift in the turning movement ratio to accommodate more left-turning vehicles on the major street. Each approach has a total traffic flow of 1000 vph. Yet, the turning movement ratio on the North and South bounds increasingly favoured left turns. These ratios evolved from the foundational 250/1100/100 vph (LT/TH/RT) to configurations such as 300/1050/100, 350/1000/100, and 400/950/100. In absolute terms, this transition caused the left-turn demand to grow from an initial 172 vph to 276 vph, spread over four stages.

3. Scenarios #3 to #5:

These scenarios are focused on opposing traffic movements. As depicted in Figure 3-5, the signalling phase ring and barrier diagram create a boundary between major and minor street phases. This means that green light time is divided between these traffic movements. Therefore, increasing the green light duration for major street movements necessitates a decrease for minor street movements, and vice versa. Figure 3-6 illustrates the competing movement scenarios by showing the demand per capacity over simulation time for various scenarios. Traffic volumes on the major streets grew in four consecutive phases.

4. Scenarios 6 to 8:

These scenarios were designed to mimic common situations on primary roads. For instance, morning traffic typically heads toward city centers, reversing in the afternoon. The objective here is to examine how these traffic shifts affect the TSSO index. Minor street traffic remains constant at either low, medium, or high levels. As with earlier scenarios, four traffic volume levels are considered. However, Northbound traffic grows hourly, while Southbound traffic decreases at the same rate. The turning traffic ratios stay fixed at 250/1100/100 vph (LT/TH/RT).

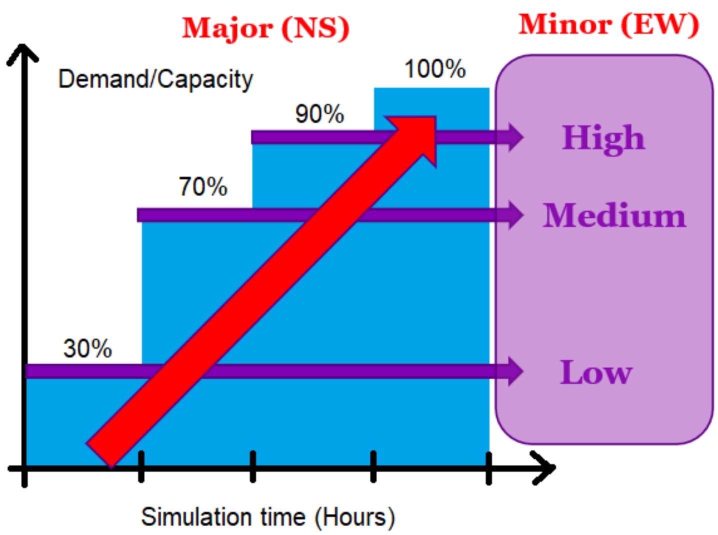


Figure 3-6. Simulation scenarios 3, 4, and 5, based on competing movements rationale

The primary objective of these carefully created scenarios is to assess the performance of the proposed TSSO index under varied traffic conditions. These scenarios further aid in identifying opportunities where adjustment of traffic signal timing could significantly enhance intersection efficiency. The findings from these simulations will deliver pivotal insights into the practical efficacy of our proposed method in real-world traffic situations.

3.4 Implementation

This section explains the methodology employed to process raw vehicle trajectory data with the aim of determining the TSSO index. This computational process has been executed using the Python programming language. Notably, for evaluative purposes, the study employs a VISSIM simulation as an alternative to the direct usage of GPS coordinates from CVs. For the sake of clarity and simplicity in the methodology, vehicle trajectories are presented in x-y coordinates rather than genuine GPS coordinates. Nonetheless, this choice does not compromise the integrity of the results due to the availability of formulas that facilitate conversions between these two coordinate systems. The comprehensive processing of vehicle trajectory data can be divided into three primary phases:

1. **Vehicle Delays Analysis:** This phase involves processing the individual data points collected from each vehicle to ascertain a performance metric. Specifically, the study aims to gauge the delays experienced by vehicles at intersections. Therefore, all the trajectory data points are examined to identify movements, such as a Northbound left turn (denoted as the NbL movement). After this identification, the travel time and associated delays for each vehicle are calculated.

2. **TSSO Index Calculation:** After quantifying vehicle delays, this phase focuses on the compilation of these delays—specifically those corresponding to 'through' and 'left turn' movements at intersections—and subsequently determines the TSSO index.

3. **Evaluation and Visualization:** The third step, focused on evaluation, and visualization of the results. This phase integrates the necessary code to either interface with datasets produced from the preceding steps or to reiterate the earlier steps in a defined manner. The ultimate objective is to generate and represent results that are pivotal for comprehensive evaluations.

Figure 3-7 shows a picture of this process, and the following sections provide further detail about each step.

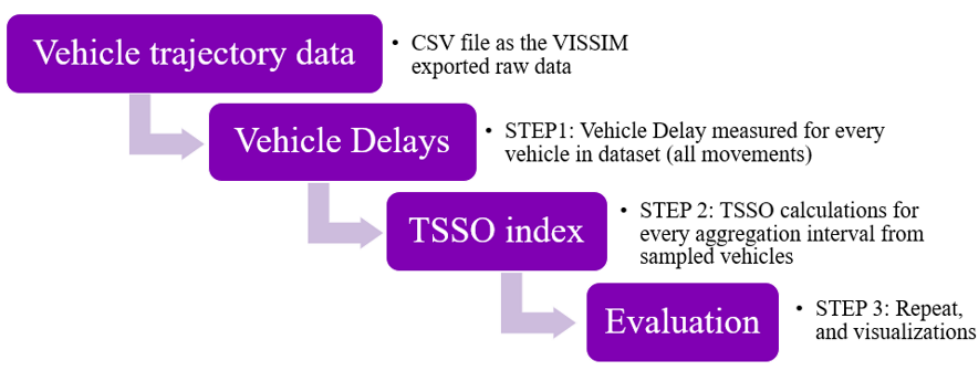


Figure 3-7. The TSSO index implementation procedure

3.4.1 STEP 1: Assessing vehicle delays for individual vehicle movements

This step demands substantial computational capabilities due to the extensive dataset that encompasses the vehicles' trajectory data. The objective is to quantify the delay for each vehicle. During this phase, neither interval times for data aggregation nor vehicle sampling are considered. Instead, the focus is on determining the specific movement of each vehicle and calculating its respective delay. Figure 3-8 provides a schematic representation of this procedure, which is developed in Python.

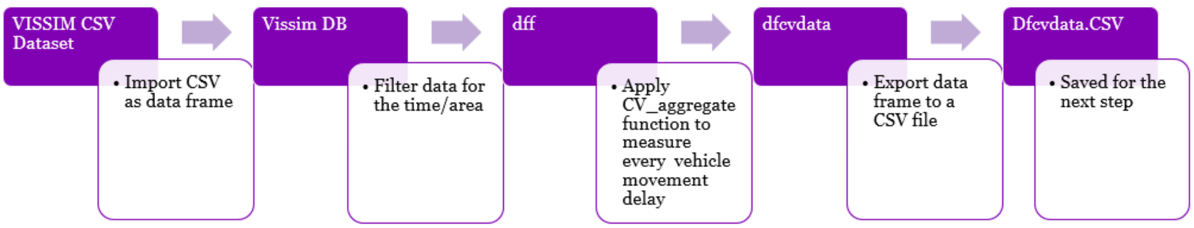


Figure 3-8. Calculating vehicle delays using trajectory data procedure

For each simulation, the trajectory dataset is processed to recognize the movement of every vehicle and evaluate its corresponding delay. The primary component of this process, as depicted in Figure 3-8, employs the *CV_aggregate* function. This function utilizes x-y coordinates based on four reference points positioned around the intersection; these points typically located at the midpoint of each intersection approach, situated approximately 200 meters upstream (as illustrated in Figure 3-9). By leveraging these x-y coordinates, vehicles traversing within a 20-meter radius around the point are detected. This facilitates the identification of the vehicle's movement and the computation of its travel time. Delay is calculated by subtracting the ideal travel time, determined by the speed limit, from the actual travel time. To correct for the minimum non-zero delays that are observed, the smallest recorded delay for each movement is subtracted from all the measured delays for that movement, ensuring a more accurate delay calculation. The function outputs a data frame that includes the vehicle's ID, the recognized movement, the adjusted vehicle delay, and the timestamp marking the completion of the vehicle's movement (T_end). This data frame is then saved as a CSV file for further analysis.

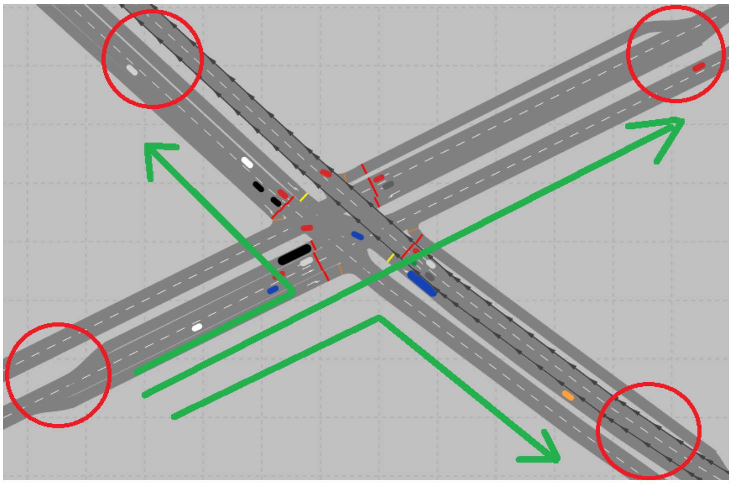


Figure 3-9. Movement identification using circles on intersection approaches

3.4.2 STEP 2: TSSO Index calculation

The primary focus of STEP 2 is on the TSSO calculations, elaborated upon in Section 3.1.1. The process begins by importing the CSV file from STEP 1 into a data frame. This frame contains information regarding each vehicle's identified movement, measured control delay, and ‘Tend’, the timestamp marking the end of the movement. The data is then refined to consider only the Left Turn (LT) and Through (TH) movements. To ascertain the average vehicle delay for each specified time interval, such as every 15 minutes, the 'Tend' parameter is utilized. This parameter aids in computing the interval number for each vehicle. For instance, if a vehicle completes a movement 27 minutes into the simulation, the interval variable would be designated as '2' when considering a 15-minute interval. The findings are stored in the 'Df_interval' data frame which is then used to group vehicle delays into each interval bin.

To calculate the average vehicle delay for each TH and LT movement within specific time intervals, a subset of vehicles is randomly selected as CVs. The measured delays of these vehicles are then grouped by movement and corresponding interval period. The mean delay for the vehicles in each group, associated with a particular movement and time interval, is determined. From this, the degree of saturation (X) is derived. Subsequently, the flow ratio (Y) is computed and integrated into the 'DF_mean' data frame. In this step the average vehicle delays are determined at the movement level.

When categorizing vehicle delays by time interval and movement, there might be instances where no vehicles execute a specific movement within a given interval. This situation is especially likely when a movement experiences low traffic demand, and the observed CV penetration rate is minimal. Under such circumstances, back-calculating the degree of saturation for that particular movement during that interval becomes unfeasible. As this can hinder the determination of the optimal signal plan and the computation of the TSSO index, a data cleaning function was designed. This function identifies instances of zero average vehicle delays resulting from the aforementioned situation and replaces them with the most recent measured delay value. Such a technique ensures the continuity of data, facilitating a more accurate estimation of the TSSO index.

Following the cleansing of the 'Df_mean' dataset, it was subsequently categorized into bins based on the interval variable. The average vehicle delays, corresponding to the TH and LT movements for each bin, were then input into the HCM function. This function determined the optimal signal timing for every bin. Consequently, the delay under this optimal plan was computed and stored as the 'delay_opt' variable within the 'Df_intersection' data frame. This frame encompasses the measured average vehicle delays, signal green times, optimal green times, and anticipated delay times under the optimal signal plan.

Finally, the TSSO index is computed for each bin using both measured and optimal delays. This value is stored in the 'Df_index' data frame. An overview of this procedure is presented in Figure 3-10.

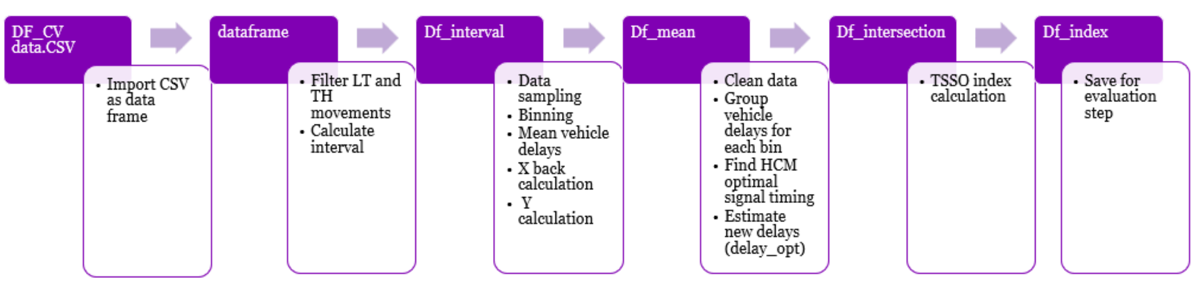


Figure 3-10. Calculating the TSSO Index procedure

3.4.3 STEP 3: Evaluation and visualizations

During this step, visualizations, such as plotting the TSSO index trend over the duration of the simulation, are utilized to showcase the results. The specific approach and objectives of the evaluation play a significant role in shaping this step.

Given the inherent randomness of the vehicle sampling process for a CV, rather than presenting the TSSO trend based on a one-time sampling from a single simulation scenario, it is advised to repetitively sample and perform calculations multiple times on the same processed data from STEP 1. By doing so, diverse sets of sampled vehicles are ensured, and subsequently varied calculated outcomes are obtained, even when derived from the same initial simulation data. Following this methodology, average results are then derived and presented with a 95% confidence interval.

Consequently, STEP 2, which deals with the sampling and subsequent calculations, will be executed multiple times on the outcome of STEP 1 to yield the necessary varied data for comprehensive evaluations. Moreover, executing a sensitivity analysis may necessitate further data manipulations, such as adjusting certain parameters and reinitiating STEP 2 to obtain new comparative results, all of which are addressed in the third step. Figure 3.11 demonstrates the simplified framework.

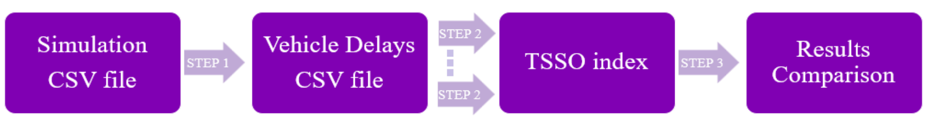


Figure 3-11. The simplified framework of TSSO analysis

3.5 Accuracy and sensitivity analysis

The accuracy and sensitivity analysis of the TSSO index evaluates its performance under varying key factors that could influence its accuracy and reliability. This section does not address external traffic conditions or other real-world elements that might affect the index. Instead, the focus remains on post-simulation variables, including the CV penetration rate and parameters employed in the methodology calculations.

3.5.1 Critical factors influencing the TSSO index

The focus of this analysis is on identifying critical factors that have the potential to impact the accuracy and reliability of the TSSO index:

1. **CV Penetration Rate (PR):** As the CV penetration rate decreases, the availability of real-time data for measuring vehicle delays is reduced, impacting the accuracy of the TSSO index. This poses a challenge in capturing traffic signal sub-optimality under low CV penetration rates.

2. **Aggregation interval time (Tint):** To mitigate the negative impact of a low CV penetration rate on the TSSO index accuracy, increasing the interval time used for data aggregation can be explored. This allows gathering more data from the available CVs, enhancing the index's accuracy.

3. **SFRs (Sf_TH and Sf_LT):** These rates significantly influence the delay calculations and, consequently, can affect the TSSO index. Understanding their impact is crucial for comprehending the index's robustness under different traffic conditions.

4. **Progression Factor (PF or Kf):** The quality of traffic progression is a critical characteristic that can influence average vehicle delay. This is modelled in the Highway Capacity Manual (HCM) delay model as the "Progression Adjustment Factor". This factor is multiplied by the uniform delay to account for the effects of signal progression on traffic flow. In scenarios where arrivals are randomly distributed, the Kf factor is set to unity (1). However, in cases with coordinated signals, the PF factor applies to all coordinated phases. This means that effective signal coordination, where a high proportion of vehicles arrive during green phases, can significantly reduce the overall delay. Specifically, a PF value of less than one indicates more efficient traffic flow due to successful signal coordination.

Table 3-3 presents the variations in parameters considered for the TSSO index sensitivity analysis. Detailed results are provided in the next chapter.

Table 3-3. Variations in parameters for TSSO sensitivity analysis

Parameter	**Definition**	**Variations**
PR	CV penetration rate	5, 10, 20%
Tint	Data aggregation interval	15, 30, 45, 60 minutes
Sf_TH	Through movement SFR	1800, 2000, 2200 vph
Sf_LT	Left turn movement SFR	1350, 1500, 1650 vph

3.6 Limitations

In the previous sections, the TSSO index was formulated based on certain simplifying assumptions. While these assumptions facilitated the resolution of the problem, they also delineate the scope of the methodology's application. This section consolidates and explicates those assumptions for clarity.

The analysis of the TSSO index was confined to individual intersections operating under fixed-time signal control, with the intersection design including designated left and right turn pockets, and excluding any shared lanes. Furthermore, the traffic signal phases were assumed to accommodate a single phase for each direction of traffic flow. These conditions were presumed to streamline the complexity of the problem, enabling the practical application of the proposed concept. Nevertheless, the potential for extending the methodology beyond these constraints exists and warrants further exploration to enhance its applicability to a broader spectrum of scenarios.

The methodology was specifically developed for intersections with fixed-time signalization, and its suitability for adaptive or actuated control signals remains unexplored within the scope of this study.

Additionally, the approach is configured for isolated intersections. In the context of corridors, one might opt to preserve a uniform cycle time, which simplifies the process but overlooks potential optimizations. Alternatively, one could attempt to optimize cycle times across a network of coordinated intersections prior to determining the optimal splits for each. The latter method, while potentially offering a more refined optimization, introduces a greater level of complexity. When dealing with widely spaced intersections, the propagation factor can be presumed to be one; however, closer intersections may require an estimation of this factor using trajectory data, which can then be integrated into the delay model to refine the accuracy of the TSSO index.

Chapter 4: Results and Discussion

This chapter presents the findings generated through the application of the TSSO index methodology. Utilizing Python for programming and the VISSIM microsimulation software for data simulation, the chapter achieves a comprehensive understanding of the performance of the proposed TSSO for a wide range of intersection traffic conditions. The outcomes shed light on the effectiveness and implications of the TSSO index, setting the groundwork for further discussions and recommendations.

The chapter provides a detailed analysis of the TSSO index values obtained from VISSIM simulations of a sample intersection. It evaluates the TSSO index under a variety of traffic scenarios, examining the merits of the methodology under study. Sensitivity analyses are also included, assessing the impact of variables such as CV penetration rates and data aggregation intervals on the accuracy and reliability of the TSSO index.

4.1 TSSO index results

In this section, the behaviour of the TSSO index across distinct traffic scenarios will be examined. This exploration aims to discern the efficacy of the TSSO index in identifying locations most conducive to signal retiming benefits. For a comprehensive understanding of the methodology, it is useful to observe the optimal signal timings determined during the intermediate stages of TSSO estimation.

4.1.1 Scenario 1 and 2: Traffic demand increases on all approaches versus major street left turns

Figure 4-1 illustrates the variations in the optimal signal plan over the simulation time for the first two scenarios described in Table 3-2. Each hatched band in the diagram represents the green time for one phase in the optimal plan. The green times of phases in each one of the barriers add up to the total green time of the optimal cycle time. The results shown in these diagrams are calculated using data from all the vehicles(i.e., 100% CV penetration rate). Hence, Figure 4-1 provides a visual representation of the optimal signal plan variations over the simulation period.

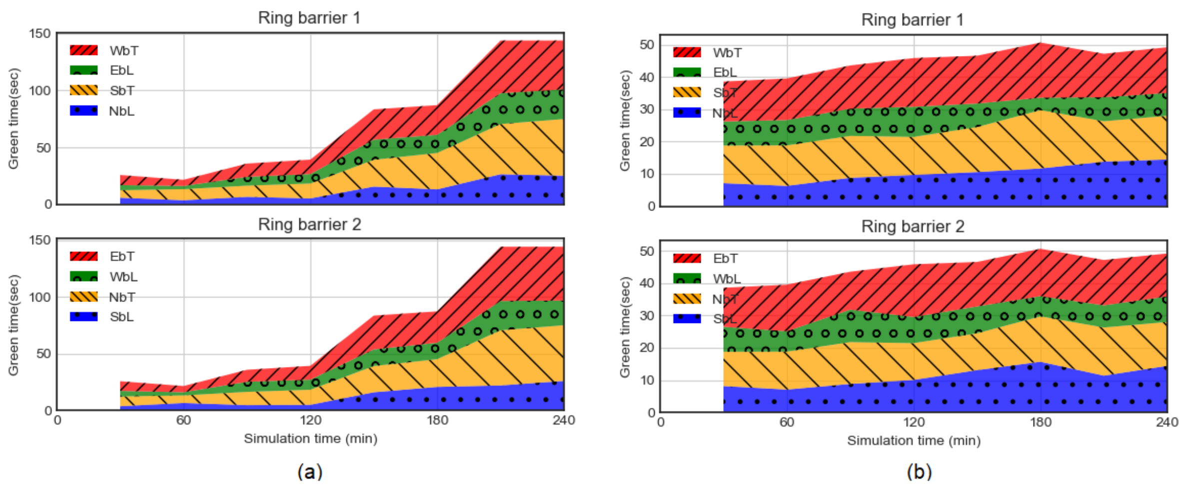


Figure 4-1. The optimal signal plan variation over simulation time: a) All approach traffic increased (scenario 1); and b) LT traffic increased on major streets (scenario 2)

Figure 4-2 illustrates the variations in the index over the course of the simulation for the first two scenarios, with a penetration rate of 10%. Due to inherent variability in the sampling process for determining the TSSO index, the random selection was executed 30 times. For each run, the TSSO index was calculated. Subsequently, the mean value was derived, and a 95% confidence interval was computed, represented as error bars in the figure. The shaded regions surrounding the dotted line serve to highlight the range of plausible outcomes for the TSSO index estimation under the given CV penetration rate. This unpredictability mirrors real-life situations where only a limited number of CVs can be observed.

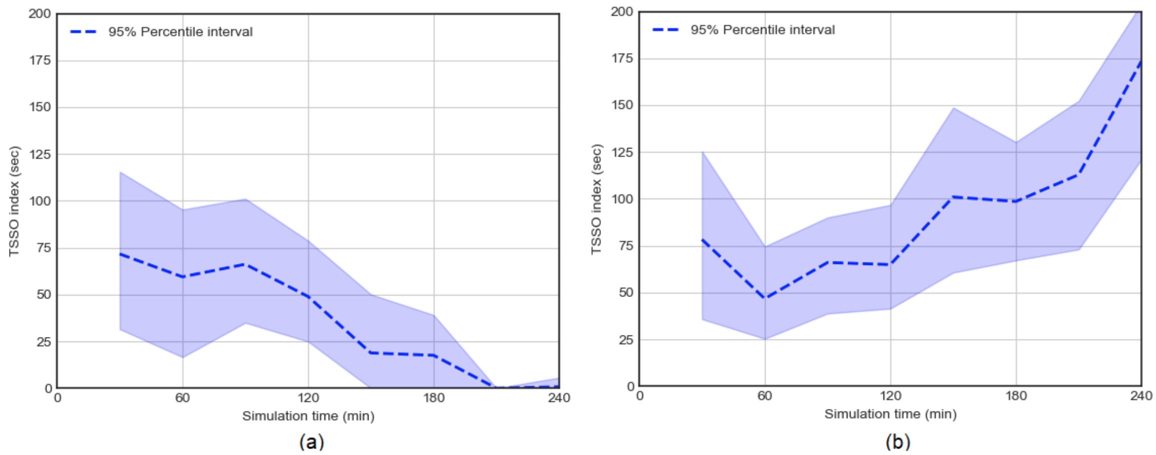


Figure 4-2. Variation and accuracy of the estimated index with 10% penetration rate a) All approach traffic increased (scenario 1), and b) LT traffic increased on major streets (scenario 2)

In Figure 4-1, a comparison of optimal signal plan variations between Scenario 1 and Scenario 2 reveals distinct trends. In Scenario 1, the green time for all phases expands uniformly and substantially to accommodate a universal increase in traffic. As a result, the Highway Capacity Manual (HCM) optimal cycle time grows, ultimately hitting its upper limit of 160 seconds. This leads to a diminishing TSSO index, approaching nearly zero (Figure 4-2). This drop indicates that despite the extension in cycle time, the potential for delay improvement is limited, and no further gains can be made, largely because the intersection is nearing full capacity for all traffic phases.

Conversely, Scenario 2 displays a different pattern. As depicted in Figure 4-1b, the optimal cycle time only marginally increases by about 10 seconds. Most of the additional capacity required for increased left-turn traffic on the major street is managed through adjustments to green splits, rather than cycle time increases. Such fine-tuning effectively reduces vehicle delays as only specific phases see an uptick in traffic. The TSSO index in this scenario shows a steadily rising trend, as illustrated in Figure 4-2b, underscoring the potential for improvement demonstrated by the optimal signal plan.

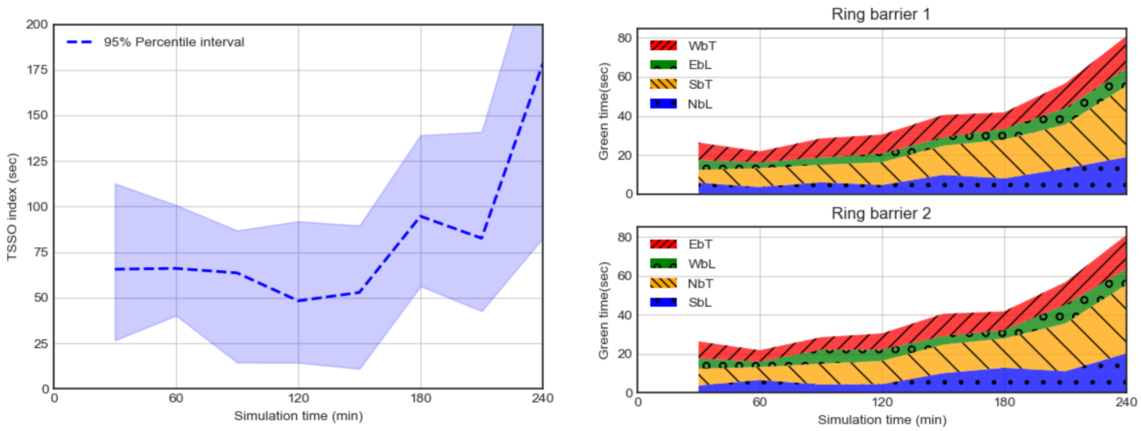
It is worth noting that both the optimal cycle time and splits are determined while the TSSO index is being calculated. Although the cycle time change in Scenario 2 is minimal, it is not constant. However, it's possible to configure the TSSO index calculations to hold the cycle time constant and only optimize green splits. This approach could be advantageous in situations where the cycle time is pre-set due to coordination requirements and is not to be altered.

4.1.2 Scenarios 3, 4 and 5: Major street traffic demand increases while minor street traffic remains at low, medium, and high

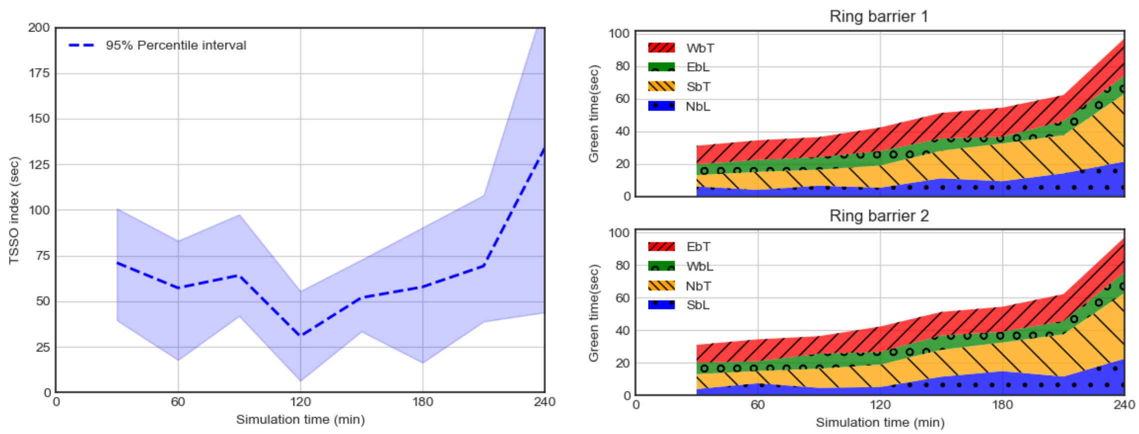
The scenarios are designed to examine competing traffic movements, where the signal phases for both major and minor streets collectively make up the entire cycle time. In all three scenarios, the traffic demand on the major street approach is incrementally increased in four steps, ranging from 500 vehicles per hour (vph) to 1450 vph. On the other hand, the traffic demand on the minor street is kept constant at preset low, medium, and high levels in scenarios 3, 4, and 5, respectively.

Figure 4-3 illustrates the TSSO index values and variations in the optimal signal plan for scenarios 3, 4, and 5. Consistent with previous figures, the TSSO index is displayed for a 10% CV penetration rate with a 95% confidence interval represented as error bands around the mean value. The optimal signal plans are shown assuming 100% CV penetration.

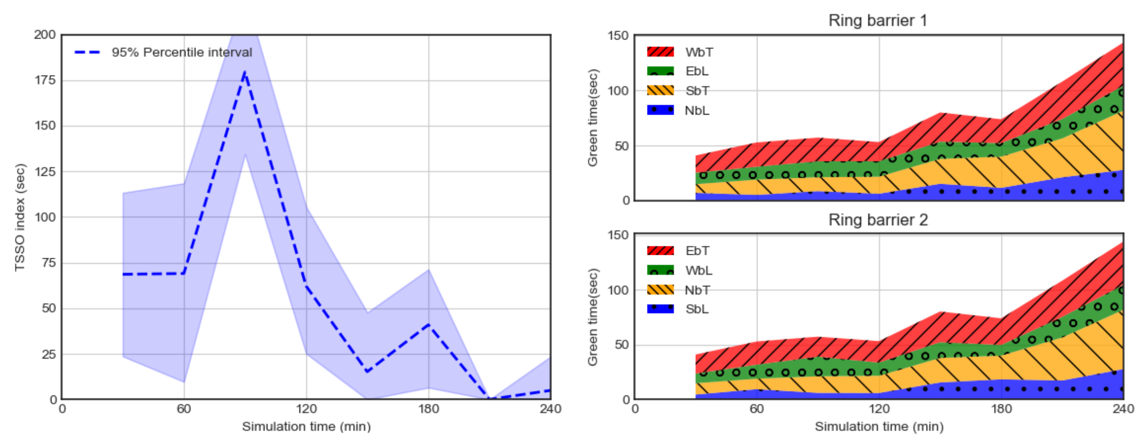
In Scenario 3 (see Figure 4-3a), there is an obvious rise in the TSSO index, indicating substantial room for reducing delays via signal retiming. However, the increase in the TSSO index is less pronounced in Scenario 5 (see Figure 4-3b), while in Scenario 6 (see Figure 4-3c), the TSSO index even shows a decline. The plots on the right side of Figures 4-3a to 4-3c reveal increases in the optimal cycle time that correspond to the elevated traffic demand on the major street. It is evident that the optimal cycle time is greater in scenarios with higher minor street traffic demand, which is both logical and intuitive.



a) Minor street traffic is low



b) Minor street traffic is medium



c) Minor street traffic is high

Figure 4-3. Evaluation of the TSSO index (left), and optimal signal plan variations (right) amid rising major street traffic, contrasted with minor street traffic at low (scenario 3), medium (scenario 4), and high (scenario 5) levels

Figure 4-3 confirms that the TSSO index, which reflects changes in traffic load on the major and minor streets, follows a logical pattern. In Scenario 3, with low traffic on the minor street, signal retiming can enhance performance as the major street traffic increases, taking advantage of the minor street's unused capacity. In contrast, in Scenario 4, with medium traffic on the minor street, the potential for improvement diminishes, as reflected in the lesser rise of the TSSO index over time. Scenario 5 reveals a declining TSSO index, suggesting that the intersection is nearing full capacity, making further delay reduction via retiming unfeasible. This typically occurs when both streets have high traffic demand, and a near-zero TSSO index indicates that there is no more room for delay reduction through retiming.

4.1.3 Scenarios 6, 7 and 8: Major street traffic demands direction change while minor street traffic remains at low, medium, and high

The scenarios are designed to examine the TSSO index in a typical real-world situation where the traffic demand on a major street is not only highly imbalanced but also reverses direction. For example, a corridor leading toward an urban center such as a downtown area usually experiences directional traffic flow, which switches direction between morning and evening peak hours. Scenarios 6 to 8 aim to mimic this shifting, directional traffic demand on the major street, while maintaining a constant level of traffic on the minor street, categorized as low, medium, or high, similar to the previous set of scenarios.

Figure 4-4 presents both the TSSO index and the optimal signal plan, maintaining the same underlying assumptions as in the earlier scenarios. As illustrated in this figure, significant reductions in delay are generally possible at the beginning or end of the simulation when the majority of the traffic on the major street is moving predominantly in one direction. To illustrate, the optimal signal plan variations in Figure 4-4a reveals that during the initial simulation hour, the Northbound approach has the least traffic (500 vph), while the Southbound approach has the highest volume at 1450 vph. In contrast, this situation is reversed in the final simulation hour. The common thread in both cases is the notable imbalance of traffic flowing primarily in one direction.

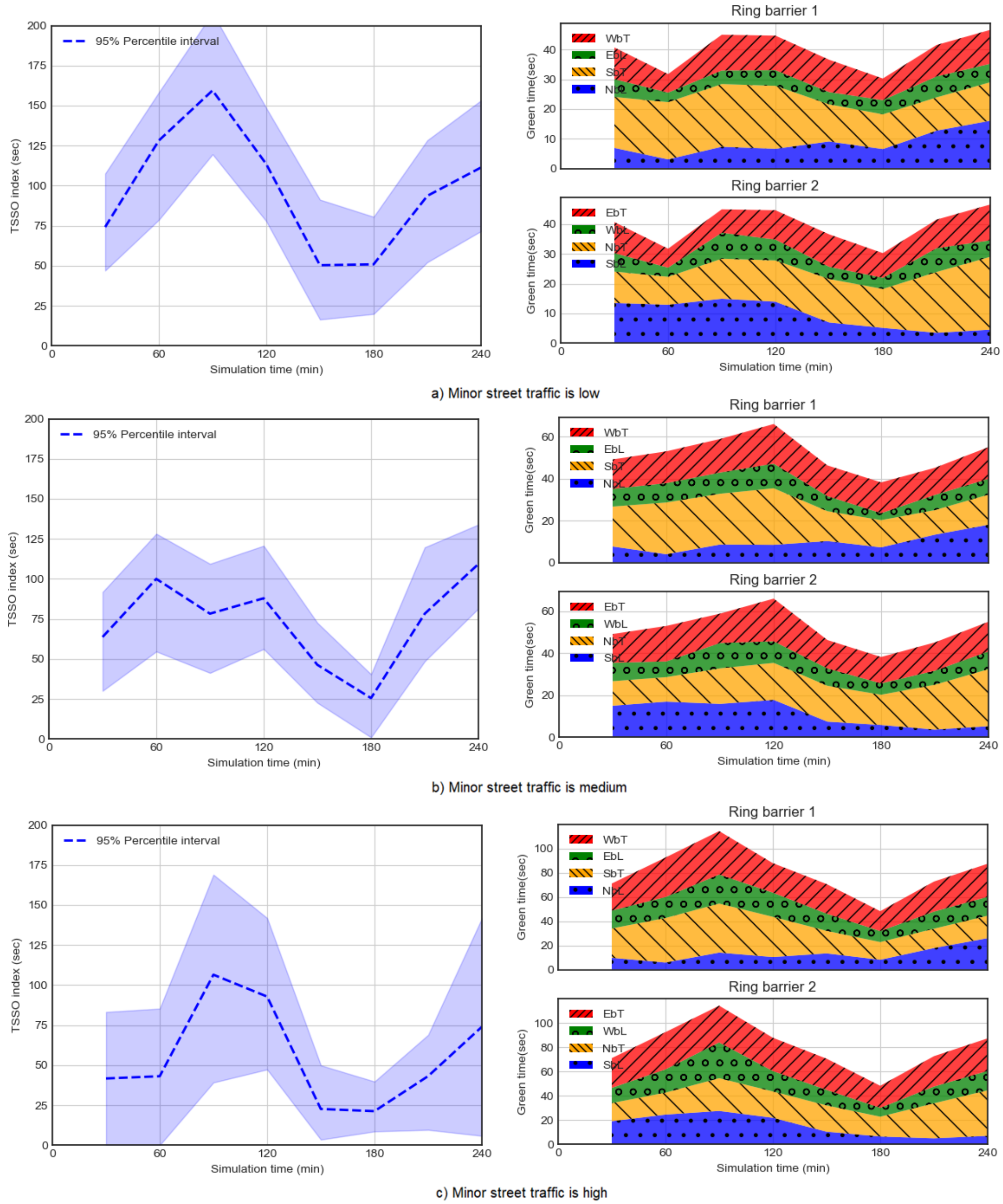


Figure 4-4. Evaluation of the TSSO index (left), and optimal signal plan variations (right) amid traffic direction change on major street, contrasted with minor street traffic at low (scenario 6), medium (scenario 7), and high (scenario 8) levels

The unique challenges posed by imbalanced and directionally reversing traffic flows on major streets typically necessitate specialized signal timing configurations, commonly employed by traffic agencies. They set distinct signal timings for both the morning and evening peak hours to better manage these considerable shifts in traffic patterns. These nuances are well-captured by the TSSO index, whose elevated values during these directional peaks highlight the need for tailored signal retiming to minimize overall intersection delay.

Additionally, the peaks of the TSSO index are influenced by the volume of traffic on the minor street. Consistent with the findings from previous scenarios, a lower volume of traffic on minor streets allows for the reallocation of extra capacity to the green phases of the major street, which in turn leads to a reduction in overall intersection delays. On the other hand, high traffic demand on minor streets can limit the extent to which signal retiming could contribute to delay reduction. However, this dampening of TSSO peaks is less pronounced here compared to the earlier scenarios. This is likely because when traffic demand increases in one direction, it decreases in the opposite direction, thereby always leaving some amount of spare capacity in the less busy direction that can be harnessed to mitigate delays. Consequently, even in the last scenario (Scenario 8), where the minor street has high traffic, the TSSO index does not fall to zero, unlike in the earlier set of scenarios.

The right-side plots in Figure 4-4 vividly illustrate the subtleties in optimal signal timing under these conditions. Unlike the straightforward increases in green time across all phases observed in some scenarios, here the optimal cycle time remains fairly consistent between the first and last simulation hours. This steadiness, however, belies the considerable changes in the allocation of green phase splits between the Northbound and Southbound directions on the major street, which nearly reverse as the traffic flow changes direction.

These findings align with our initial expectations, particularly considering the complexities of the signal plan ring barrier and the interdependencies among different traffic movements. A higher traffic demand on the minor street necessitates longer minimum green times for East-West movements, constraining the amount of green time that can be allocated to the Northbound or Southbound directions on the major street. However, fine-tuning the phase splits can efficiently manage the unidirectional heavy flows, primarily because the left-turn and through-movement phases for each direction (Northbound or Southbound) are not part of the same ring barrier. This allows for parallel adjustments, dependent on the available green phase time from the opposite direction. This is particularly beneficial when traffic demand is high in only one direction, as reflected by the similar TSSO trends observed across all three scenarios.

4.2 Sensitivity analysis results

This section reports the results of the sensitivity analysis conveyed to evaluate the accuracy and reliability of the TSSO index for variations of the CV penetration rate, aggregation interval time, and SFRs.

4.2.1 Sensitivity to CV penetration rate and aggregation interval time

The penetration rate of CVs stands as an initial and crucial limiting factor when utilizing CV trajectory data to estimate any type of traffic metric. A lower penetration rate inherently limits the pool of sample vehicles available for data collection, thereby leading to decreased accuracy and increased uncertainty. It is important to note that the CV penetration rate is a variable beyond control, as it is determined by the actual composition of vehicles on the road.

In contrast, the aggregation interval time is a parameter that can be adjusted according to the specific needs of the application. Longer aggregation interval times tend to counteract the limitations imposed by low CV penetration rates. This is because a longer interval allows for the inclusion of more sample vehicle data in each point estimation, thereby improving the overall accuracy.

As detailed in Table 3-3 of the previous chapter, a range of CV penetration rates were considered, specifically from 5% to 20%. Additionally, four different aggregation interval times were examined. This resulted in a total of 12 unique combinations to examine the sensitivity of the TSSO index to these variables. The outcomes of this sensitivity analysis are presented in Figures 4-5 to 4-13, which correspond to all eight scenarios outlined in Table 3-2.

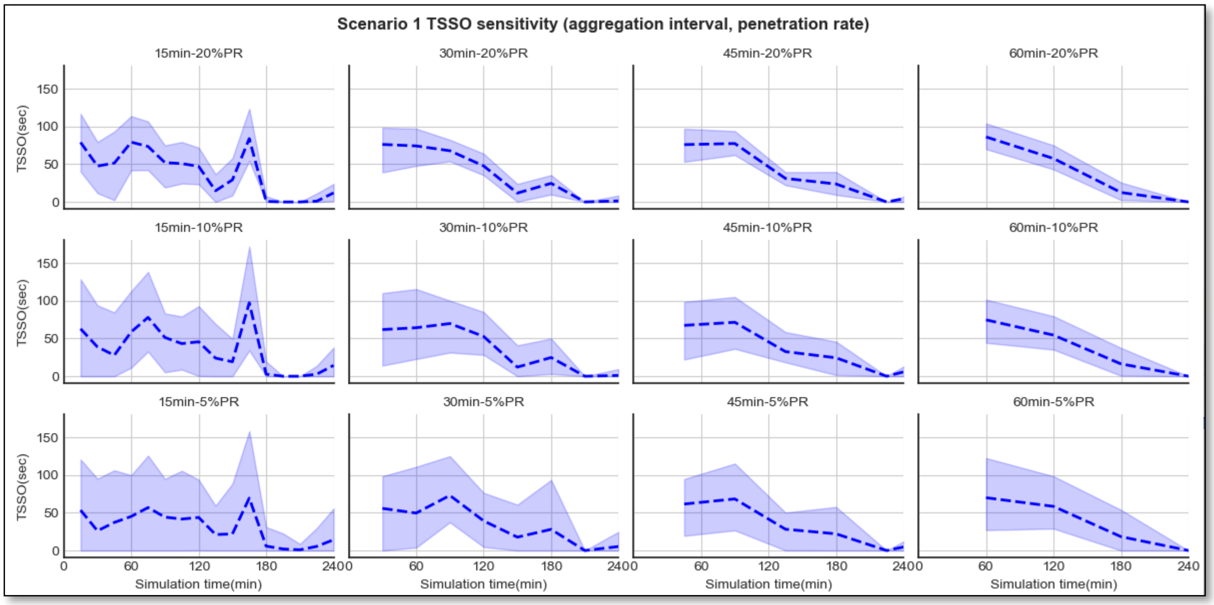


Figure 4-5. TSSO sensitivity to CV penetration rate and aggregation time (scenario 1)

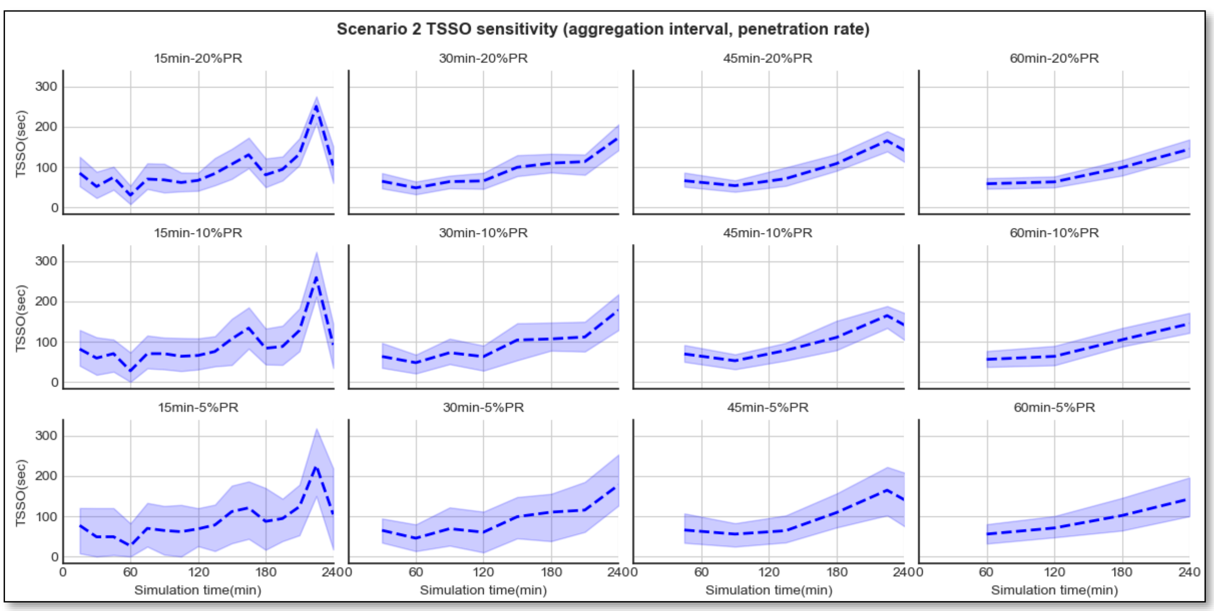


Figure 4-6. TSSO sensitivity to CV penetration rate and aggregation time (scenario 2)

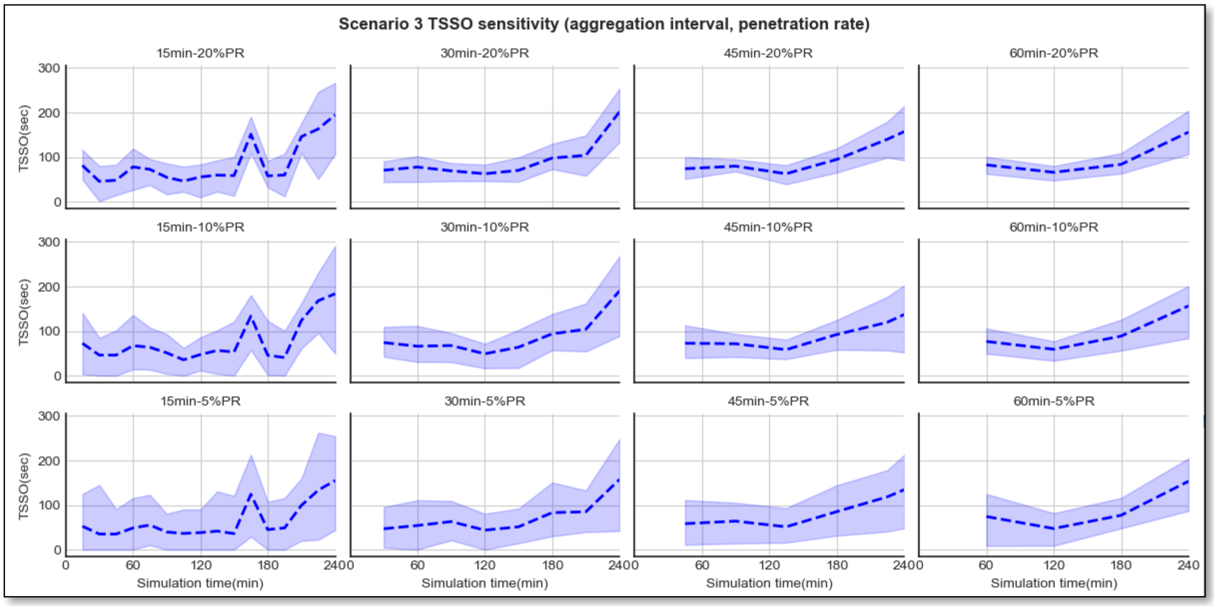


Figure 4-7. TSSO sensitivity to CV penetration rate and aggregation time (scenario 3)

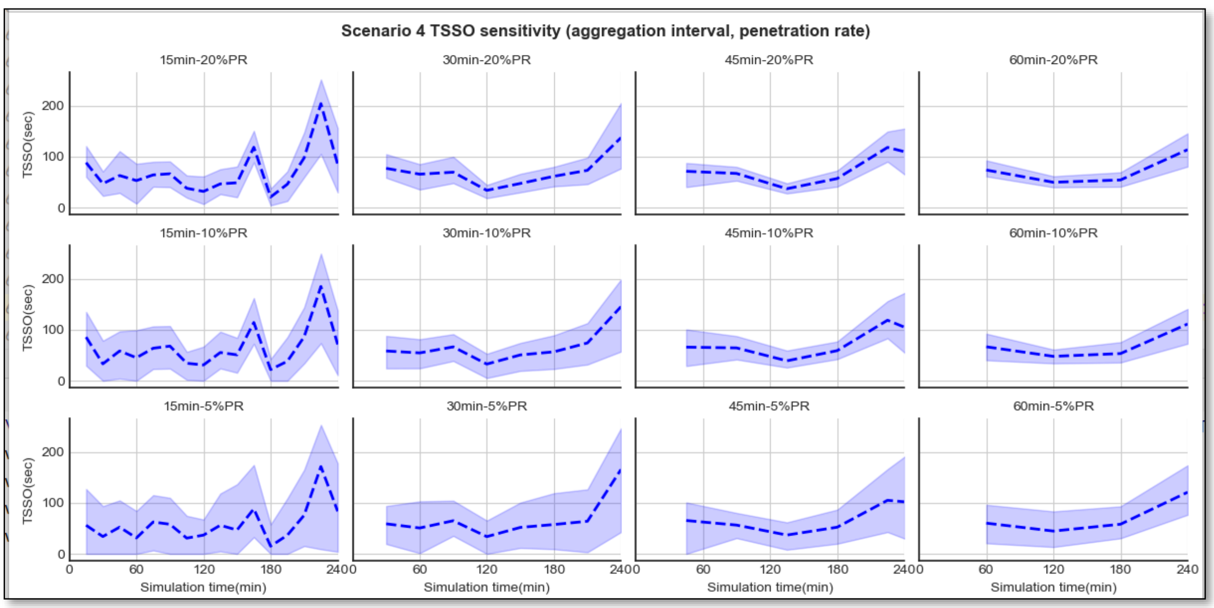


Figure 4-8. TSSO sensitivity to CV penetration rate and aggregation time (scenario 4)

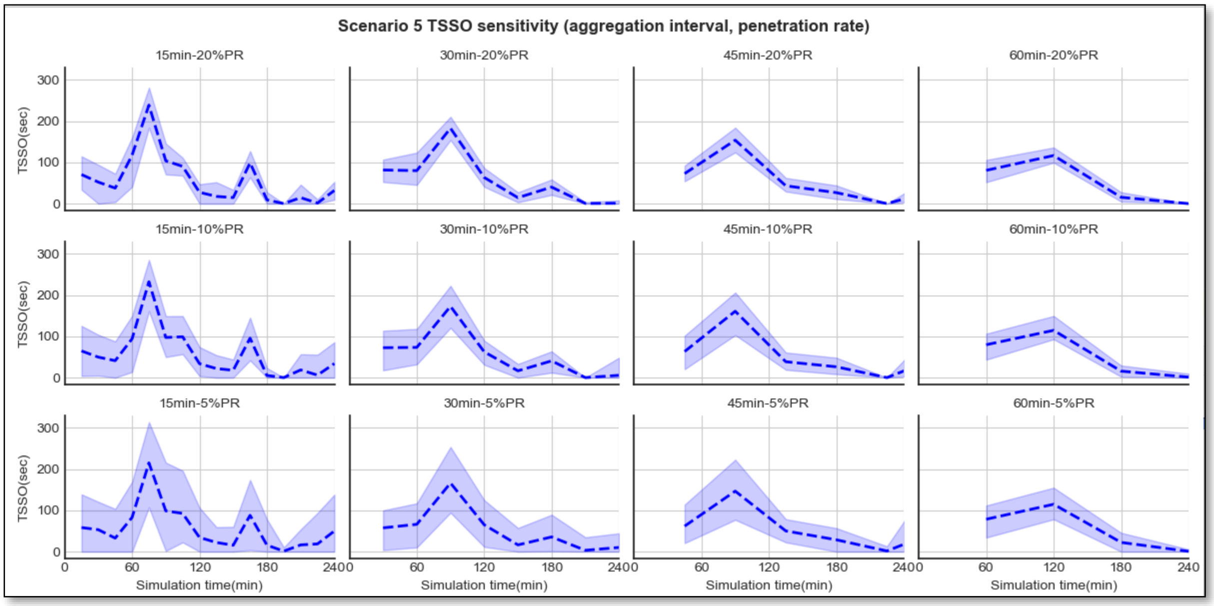


Figure 4-9. TSSO sensitivity to CV penetration rate and aggregation time (scenario 5)

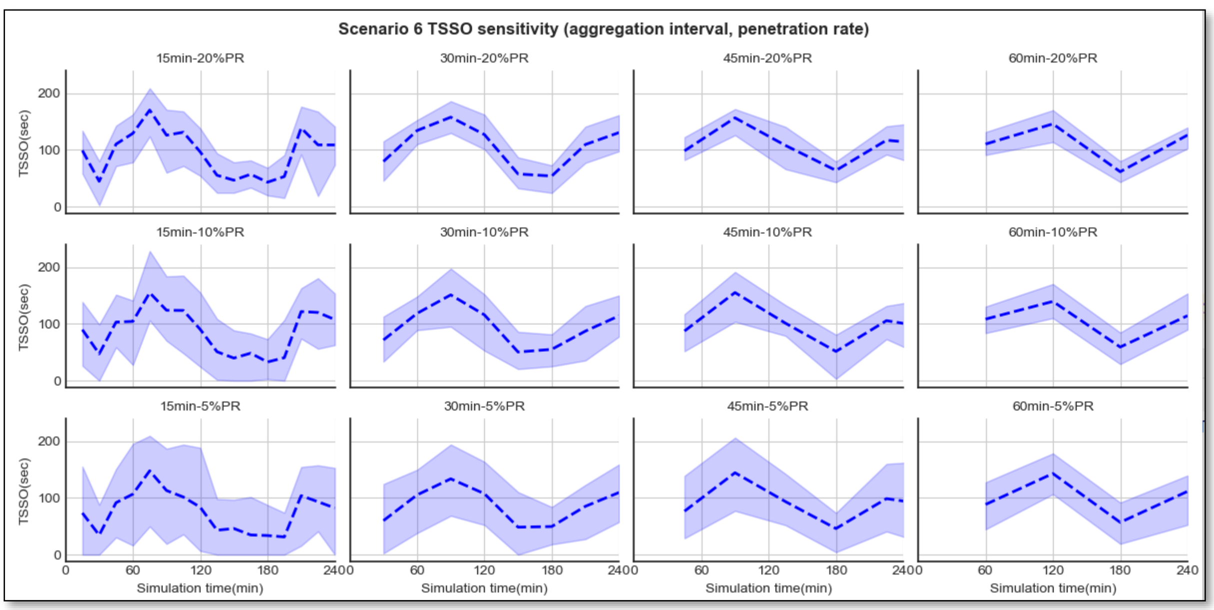


Figure 4-10. TSSO sensitivity to CV penetration rate and aggregation time (scenario 6)

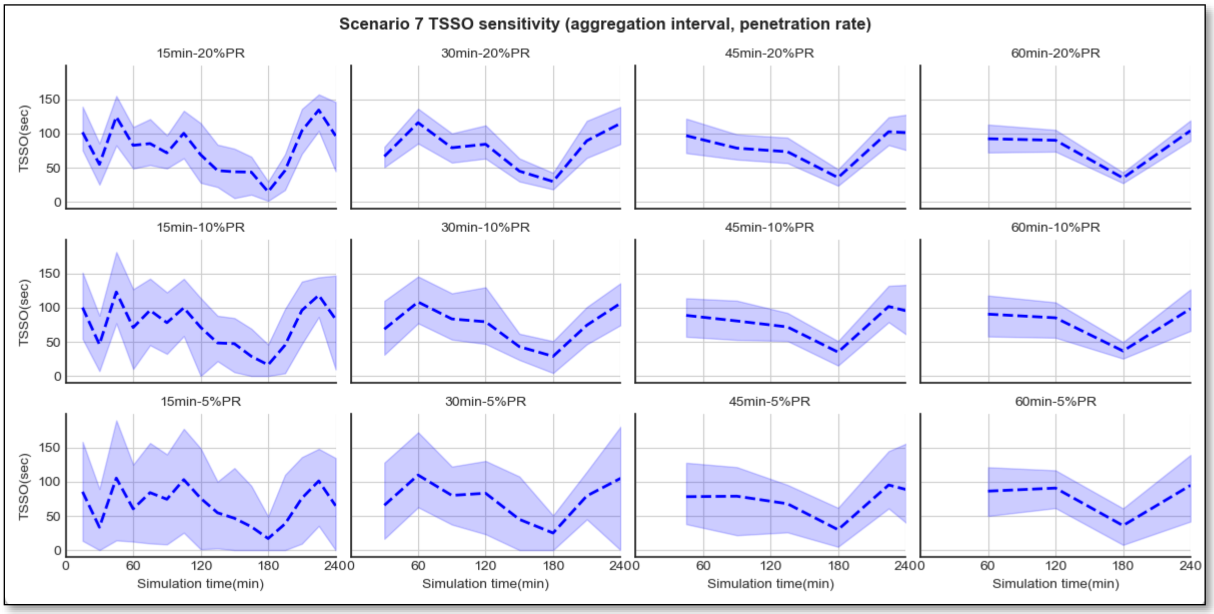


Figure 4-11. TSSO sensitivity to CV penetration rate and aggregation time (scenario 7)

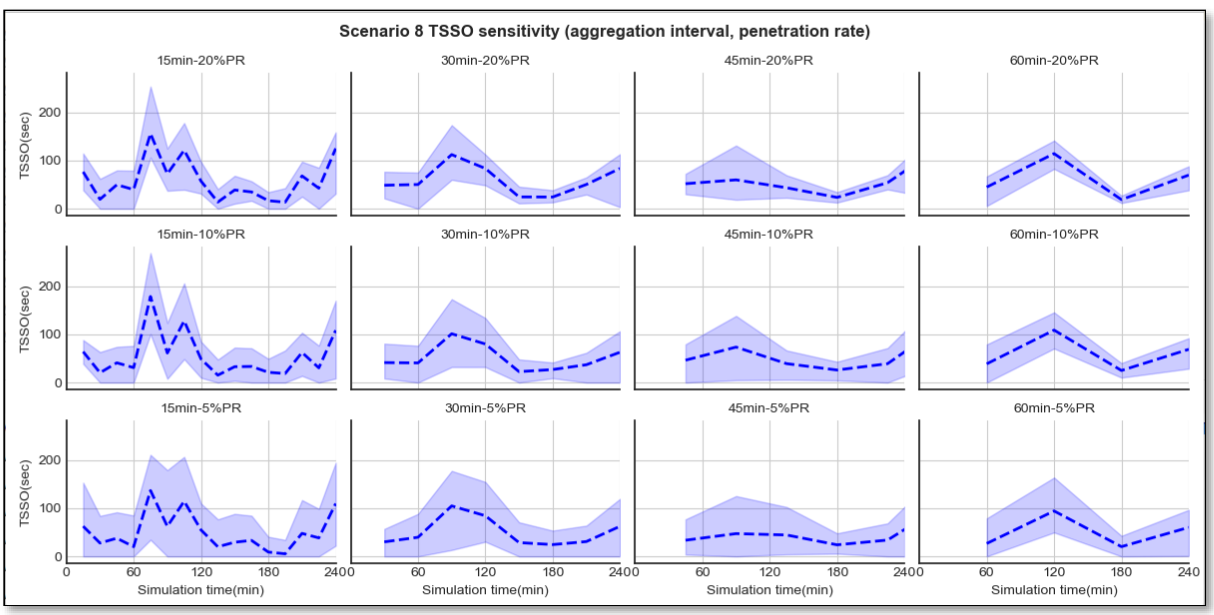


Figure 4-12. TSSO sensitivity to CV penetration rate and aggregation time (scenario 8)

Observations from the figures illustrate that as the aggregation interval extends (from left to right), the representation of the TSSO index exhibits a smoother curve. Correspondingly, as the penetration rate decreases (from top to bottom), the uncertainty, represented by the height of the error bars, intensifies. These outcomes affirm the anticipated behavior of the TSSO index. When data are aggregated over larger time periods, the TSSO is effectively smoothed. Conversely, the precision of the TSSO dwindles as the sample size reduces.

The sensitivity analysis on CV Penetration Rate revealed that while even with a 5% CV penetration rate and 15-minute aggregation time interval, it is possible to detect the vigorous changes in TSSO but there is a high amount of uncertainty. Hence, it is recommended that at least a 30-minute interval is considered for such a low CV penetration rate.

4.2.2 Sensitivity to delay model parameters

This section provides an overview of the results for variations in the parameters used in the CCG delay model, which serves as the basis for calculating the TSSO index. These calculations enable the back-estimation of the degree of saturation for each movement at the intersection. As detailed in Table 3.2, variables considered for sensitivity analysis include the SFR for both left-turn and through movements.

Figure 4-13 illustrates the delay model, based on the CCG, that is used to reverse-calculate the degree of saturation for both left-turn (LT) and through (TH) traffic movements. The model is plotted for the baseline SFR and for values that deviate by plus or minus 10% from the baseline. This is done to highlight the model's sensitivity to fluctuations in the SFR. The proximity of the variant model lines to the baseline suggests that the sensitivity of the TSSO index to alterations in the SFR parameter is likely to be minimal or negligible.

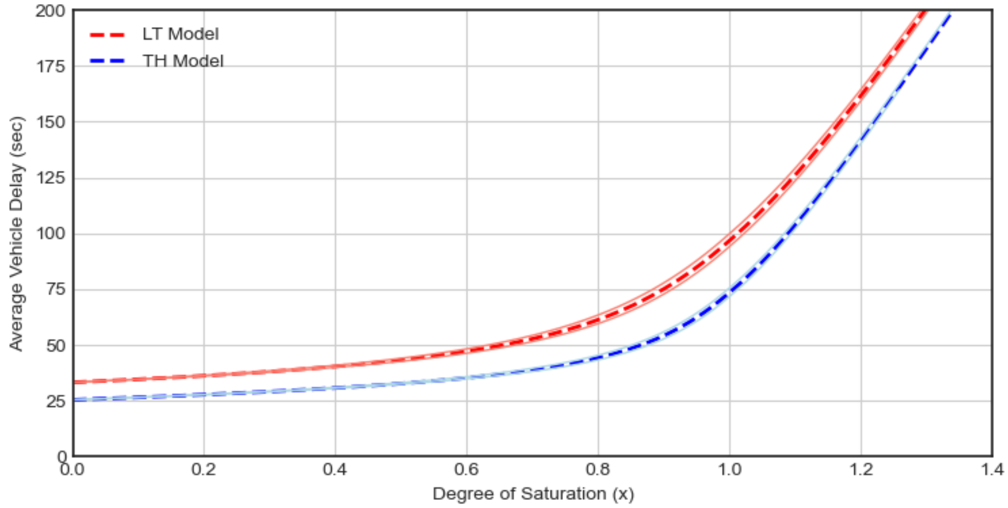


Figure 4-13. Average vehicle delay model sensitivity to ±10% SFR variations

Figure 4-14 illustrates the impact of varying the SFR parameter on the TSSO index outcomes for scenario 1. To clarify, we choose a 100% CV penetration rate to highlight the effects of SFR variations of ±10%. This choice helps remove variations from random sampling, focusing attention on variations mainly from changes in SFR. Since the TSSO index is fundamentally based on the statistical relationship between average vehicle delay and the degree of saturation, using a longer aggregation time, like 30 minutes, aids in minimizing random fluctuations in the results. This approach offers a more distinct view of the impacts due to changes in SFR and enhances the trustworthiness and precision of our findings.

As indicated in the figure, the fluctuation in the estimated TSSO is more pronounced for the left-turn (LT) SFR parameter than for the right-turn (RT) SFR. While the TSSO variation hovers around 10% at approximately 80 seconds, the percentage of variation appears to be higher for lower TSSO values. The results indicate that TSSO index estimations are relatively robust against minor variations in SFR parameters. The model's low sensitivity to these parameters underscores its applicability in various scenarios without necessitating frequent recalibrations.

Taken together, the results from the sensitivity analysis on CV Penetration Rate and Aggregation Interval Time suggest that the TSSO index's reliability and accuracy can be optimized by calibrating these two variables in conjunction. For example, in scenarios where the CV Penetration Rate is low, extending the Aggregation Interval Time can mitigate the adverse impacts on the TSSO index's accuracy. On the other hand, the TSSO index's robustness against small variations in SFR parameters increases its generalizability and makes it a dependable tool for traffic management systems.

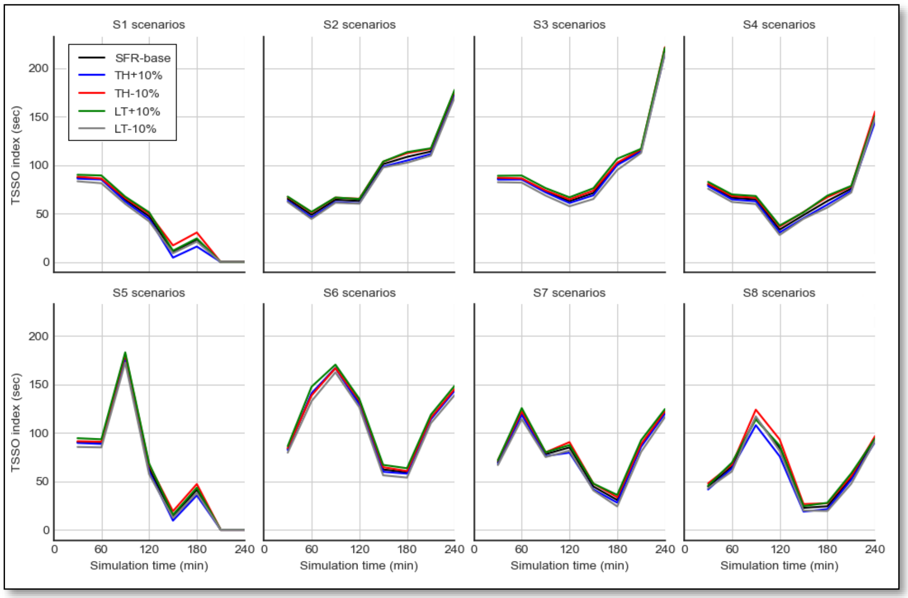


Figure 4-14. TSSO sensitivity to ±10% SFR variations in delay model (scenario 1 to 8)

Chapter 5: Conclusions

5.1 Significance and implications of the TSSO index

The TSSO (Traffic Signal Sub-Optimality) index improves how agencies can prioritize their signal retiming initiatives. Traditional metrics like average vehicle delay provide an initial indicator of problem intersections or time periods but fall short in assessing actionable pathways for improvement. The TSSO index addresses this gap by synthesizing both traffic signal phase and timing settings along with average vehicle delays across multiple traffic movements. This enriched information is then subject to a sophisticated computational process that distills it all into a single, easy-to-understand metric: the TSSO index, which quantifies the potential reduction in average vehicle delay in seconds. Armed with this actionable index, practitioners are enabled to evolve from merely identifying where delays are most severe to actively pinpointing where and when these delays can be mitigated through strategic signal retiming. Utilizing elements from the HCM and CCG, including simplified signal optimization techniques and delay models, the TSSO index could become as a effective tool that guides agencies in directing their retiming efforts to the most impactful locations and times.

The results from our simulation experiments have shown that the proposed TSSO index could be applied to identifying opportunities for signal retiming. The simulated scenarios highlight that retiming signals can substantially impact delays when there is a considerable imbalance in the degree of saturation, primarily when allocated green times are not properly matched with traffic demands. Essentially, readjusting green times could balance the degree of saturation across all movements, thereby reducing overall vehicle delays.

The quality of the TSSO index was found to be dependent on the delay data obtained from CVs. Simulations indicate that with penetration rates as low as 5%, it is possible to correctly detect extremely unbalanced conditions, while an increased penetration rate of 10% improves accuracy further. For scenarios with lower penetration rates or traffic volumes, compensatory measures could be employed such as using longer aggregation periods or accumulating data over multiple days.

5.2 Applicability and sensitivity analysis

The TSSO index was evaluated using specific scenarios designed to assess its ability to identify potential reductions in signal timing delays. While the simulation results demonstrated that the TSSO index can successfully estimate the potential for signal retiming, a sensitivity analysis was also conducted to determine how the accuracy of the TSSO index is impacted by various factors. These factors include the CV penetration rate, aggregation interval time, and SFR parameters.

Our sensitivity analysis revealed that lower CV penetration rates can impact the accuracy of the TSSO index. However, extending the aggregation interval time can compensate for this, particularly when the penetration rate is low. As for the SFR parameters, our analysis indicated that they are crucial for the delay model's back-calculations and can affect the TSSO index results, but they are not as sensitive as other factors. Notably, even ±10% variations in SFR parameters were found to have a negligible impact on the final TSSO index outcomes.

5.3 Limitations and future directions

This research evaluated the proposed method with an 8-phase signal plan with only protected left turn phases, implying that with some modifications, it could be adapted for different types of intersections. Factors like delay model parameters, which could be calibrated based on specific conditions like weather or seasons, also might affect the accuracy of the TSSO index. Furthermore, a sensitivity analysis could help identify the impact of various factors on the accuracy of the TSSO index.

While this study offers valuable insights into the applicability of the TSSO index in traffic management, it does have some limitations. For example, the impact of permitted left turn, shared lanes and low-traffic movements has not been investigated. In addition, the study observed that lower penetration rates and shorter aggregation times could compromise the accuracy of delay estimations. Moreover, our simulations did not account for the influence of platoon arrivals, a common occurrence at closely spaced intersections due to upstream filtering effects.

These limitations underscore the need for additional targeted simulations to evaluate their impact and formulate mitigation strategies. For instance, one potential approach could involve utilizing raw trajectory data to estimate the effect of vehicle platoons. By calculating the arrival-on-green percentage, it would be possible to estimate the progression factor (K_f) in the delay model. Incorporating this progression factor into optimal signal timing and TSSO index calculations could further refine the model's accuracy and applicability.

Compared to Carrenza et al.'s method in (24), TSSO index estimation needs more calculations and assumptions. Yet, it offers advantages in identifying retiming opportunities beyond just the most extreme conditions when the split failure rate increases, hence paving the way for studies that seek to pinpoint when and where signal retiming is most needed.

5.4 Conclusions

In this study, a method was developed to assess the effectiveness of traffic signal plans by utilizing CV data. The TSSO index, which was proposed, was evaluated through simulations conducted under specific conditions. The results of these simulations have highlighted the capability of the TSSO index to pinpoint instances where modifications to traffic signal timings could augment intersection efficiency by minimizing overall vehicle delay.

Simulations were conducted to replicate imbalanced traffic demand and signal timing, and it was affirmed through these simulations that the TSSO index possesses the capability to discern opportunities for signal retiming when non-critical movements have surplus capacity beneficial to critical movements. Furthermore, scenarios involving alterations in the dominant traffic flow direction on the major street were considered in the study. In these instances, the potential of the TSSO index in pinpointing the conceivable benefits of signal retiming was demonstrated. The results suggest that even a 5 to 10% penetration rate of CVs is sufficient to detect these opportunities.

In light of the limitations and opportunities revealed by this research, there are several crucial areas that warrant future investigation. First, additional research should examine the influence of various signal phase and timing configurations, including the incorporation of permitted left turns. Such an exploration would contribute to the generalization of the TSSO index methodology. Second, the impact of shared lanes and low-traffic movements on the TSSO index deserves scrutiny. Lastly, the phenomenon of platoon arrivals, commonly observed at closely spaced intersections, should be explicitly accounted for in future studies. This might entail utilizing raw trajectory data to estimate the percentage of vehicles arriving during green phases, which would subsequently allow for the calculation of a progression factor, known as Kf, in the delay model. Undertaking these research initiatives would serve to enhance the accuracy and applicability of both optimal signal timing and the TSSO index, rendering the TSSO index a more effective tool for traffic management.

References

Administration, F. H., & Transportation, U. S. D. O. (2008). Traffic Signal Timing Manual.

Canadian Automobile Association. 2021a. *Breaking Bottlenecks: Congestion Solutions for Canada*. Retrieved from Available online: https://www.caa.ca/app/uploads/2021/01/Congestion-solutions-Summary-ENG-V2.pdf

Chen, B., Fu, T., Wang, L., Yu, C., & Ma, W. (2020). Efficiency Evaluation of Signal Control Intersections Based on Trajectory Data. CICTP 2020. Retrieved from https://doi.org/10.1061/9780784483053.026

Day, C. M., D. M. Bullock, H. Li, S. M. Lavrenz, W. B. Smith, and & Sturdevant J. R. (2015). *Integrating Traffic Signal Performance Measures into Agency Business Processes.* Purdue University, West Lafayette, Indiana. Retrieved from http://dx.doi.org/10.5703/1288284316063

Day, C. M., Remias, S. M., Li, H., Mekker, M. M., McNamara, M. L., Cox, E. D., & Bullock, D. M. (2015). Performance Ranking of Arterial Corridors Using Travel Time and Travel Time Reliability Metrics. Transportation Research Record: Journal of the Transportation Research Board, 2487(1), 44–54. Retrieved from https://doi.org/10.3141/2487-04

Day, C.M., J.R. Sturdevant, & Bullock D.M. (2010). Outcome Oriented Performance Measures for *Management of Signalized Arterial Capacity*, TRB Paper 10-0008, Presented at the 89th Annual Meeting of the Transportation Research Board, Washington, D.C., Jan. 10–14.

Dunn, M. R., Ross, H. W., Baumanis, C., Wall, J., Lammert, J., Duthie, J., Ruiz Juri, N., & Machemehl, R. B. (2019). Data-Driven Methodology for Prioritizing Traffic Signal Retiming Operations. Transportation Research Record: Journal of the Transportation Research Board, 2673(6), 104–113. Retrieved from https://doi.org/10.1177/0361198119843236

Eckert, J., & Al-Habashna, A. (2022). *Traffic volume estimation from traffic camera imagery: Toward real-time traffic data streams*. Reports on Special Business Projects. Retrieved from https://www150.statcan.gc.ca/n1/pub/18-001-x/18-001-x2022001-eng.htm

Federal Highway Administration. (2022, July 14). EDC-4 Innovations. Retrieved from https://www.fhwa.dot.gov/innovation/everydaycounts/edc-4.cfm.

FHWA. Everyday Counts: An Innovation Partnership with States. FHWA-19-CAI-013. Federal Highway Administration, Washington, D.C., 2019. Retrieved from: https://ops.fhwa.dot.gov/arterial_mgmt/pdfs/EDC-4-Factsheet_ATSPMs.pdf

Fourati, W., & Friedrich, B. (2019, May 18). Trajectory-Based Measurement of Signalized Intersection Capacity. Transportation Research Record: Journal of the Transportation Research Board, 2673(10), 370–380. Retrieved from https://doi.org/10.1177/0361198119848412

Google Research. (n.d.). Green Light. Retrieved from https://sites.research.google/greenlight/

Gordon, R. (2010). *Traffic Signal Retiming Practices in the United States*. NCHRP Synthesis 409, Transportation Research Board, Washington, D.C.

Hristova, B. (2023). *Data shows how much time Canadians spent in rush hour traffic in 2022*. CBC News. Retrieved from https://www.cbc.ca/news/canada/hamilton/traffic-time-2022-1.6755102

Jenelius, E., & Koutsopoulos H. N. (2013). Travel Time Estimation for Urban Road Networks using Low-Frequency Probe Vehicle Data. Transportation Research Part B: Methodological, Vol. 53, pp. 64–81.

Khattak, Z. H., Magalotti, M. J., & Fontaine, M. D. (2020). Operational performance evaluation of adaptive traffic control systems: A Bayesian modeling approach using real-world GPS and private sector PROBE data. Journal of Intelligent Transportation Systems, 24(2), 156–170. Retrieved from https://doi.org/10.1080/15472450.2019.1614445

Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., et al. (2008). *Traffic Signal Timing Manual*. Report No. FHWA-HOP-08-024, Federal Highway Administration.

Leitner, D., Meleby, P., & Miao, L. (2022). Recent advances in traffic signal performance evaluation. Journal of Traffic and Transportation Engineering (English Edition), 9(4), 507–531. Retrieved from https://doi.org/10.1016/j.jtte.2022.06.002

Livey. (2015). *2015 Capital Budget Briefing Note: Funding for Corridor Signal Coordination Studies of Additional 500 Traffic Signals*. Retrieved from in www.toronto.ca.

Lonzer, B. (2019). A Primer on Automated Traffic Signal Performance Measures. Kittelson & Associates. Retrieved from https://www.kittelson.com/ideas/a-primer-on-automated-traffic-signal-performance-measures/.

National Academies of Sciences, Engineering, and Medicine. (2020). Performance-Based Management of Traffic Signals. Washington, DC: The National Academies Press. https://doi.org/10.17226/25875

Nevers, B., Urbanik, T., Lee, K., et al. (2020). *Performance-Based Management of Traffic Signals*. NCHRP Report 954, Transportation Research Board, Washington, D.C.

NIATT Lab Manual. (n.d.). Capacity and saturation flow rate. University of Idaho. Retrieved from https://www.webpages.uidaho.edu/niatt_labmanual/Chapters/signaltimingdesign/theoryandconcepts/CapacityAndSaturationFlowRate.htm

Saldivar-Carranza, E. D., Li, H., Platte, T., & Bullock, D. M. (2023). Systemwide Identification of Signal Retiming Opportunities with Connected Vehicle Data to Reduce Split Failures. Transportation Research Record: Journal of the Transportation Research Board, 036119812311688. Retrieved from https://doi.org/10.1177/03611981231168844

Saldivar-Carranza, E., Li, H., Mathew, J., Fisher, C., & Bullock, D. M. (2022). Signalized Corridor Timing Plan Change Assessment Using Connected Vehicle Data. *Journal of Transportation Technologies*, 12(03), 310–322. Retrieved from https://doi.org/10.4236/jtts.2022.123019

Saldivar-Carranza, E., Li, H., Mathew, J., Hunter, M., Sturdevant, J., & Bullock, D. M. (2021, April 30). Deriving Operational Traffic Signal Performance Measures from Vehicle Trajectory Data. *Transportation Research Record: Journal of the Transportation Research Board*, *2675*(9), 1250–1264. Retrieved from https://doi.org/10.1177/03611981211006725

Saldivar-Carranza, E., Li, H., Mathew, J., Hunter, M., Sturdevant, J., & Bullock, D. M (2021). Deriving Operational Traffic Signal Performance Measures from Vehicle Trajectory Data. Transportation Research Record: Journal of the Transportation Research Board, 2675(9), 1250–1264. Retrieved from https://doi.org/10.1177/03611981211006725

SMATS Traffic Solutions. (2021). *Detector vs. Trajectory Generated Automated Traffic Signal Performance Measures (ATSPMs)*, SMATS. Retrieved from https://www.smatstraffic.com/project/detector-vs-trajectory-atspm/

State of North Carolina. (2019). *NCDOT ATSPM Implementation Plan*. https://connect.ncdot.gov/resources/safety/Teppl/TEPPL%20All%20Documents%20Library/NCDOT%20Guide%20on%20ATSPM.pdf

Sunkari, S. (2004). *The Benefits of Retiming Traffic Signals. Institute of Transportation Engineers ITE Journal*, Vol. 74, No. 4, pp. 26–29

Traffic Index ranking, TomTom Traffic Index. (2022). Traffic Index Ranking. Retrieved from https://www.tomtom.com/traffic-index/ranking/

Urbanik, T., A. Tanaka, B. Lozner, E. Lindstrom, K. Lee, S. Quayle, S Beaird, et al. (2015). *Signal Timing Manual – Second Edition.* NCHRP Report 812, Transportation Research Board, Washington, D.C.

Waddell, J. M., Remias, S. M., Kirsch, J. N., & Young, S. E. (2020, August 27). *Scalable and Actionable Performance Measures for Traffic Signal Systems using Probe Vehicle Trajectory Data*. Transportation Research Record: Journal of the Transportation Research Board, 2674(11), 304–316. Retrieved from https://doi.org/10.1177/0361198120941847

Wunsch, G., Bolling, F., von Dobschutz, A., & Mieth, P. (2015, January). Bavarian Road Administration's Use of Probe Data for Large-Scale Traffic Signal Evaluation Support. Transportation Research Record: Journal of the Transportation Research Board, 2487(1), 88–95. Retrieved from https://doi.org/10.3141/2487-08

Zhao, Y., Zheng, J., Wong, W., Wang, X., Meng, Y., & Liu, H. X. (2019, June 19). Estimation of Queue Lengths, Probe Vehicle Penetration Rates, and Traffic Volumes at Signalized Intersections using Probe Vehicle Trajectories. *Transportation Research Record: Journal of the Transportation Research Board*, *2673*(11), 660–670. Retrieved from https://doi.org/10.1177/0361198119856340.

Zheng, J., & Liu, H. X. (2017, June). Estimating traffic volumes for signalized intersections using connected vehicle data. Transportation Research Part C: Emerging Technologies, 79, 347–362. Retrieved from https://doi.org/10.1016/j.trc.2017.03.007

Appendix A: Python Code

Part a: Python code for implementing STEP #1

```
# STEP1: Vehicle delay measurements from vehicle trajectory dataset
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import sys
import winsound

# ------------------ FUNCTION DEFINITIONS ------------------


# Maps indices to unique RGB colors using a specified colormap.
def get_cmap (n, name='hsv'):
    return plt.cm.get_cmap(name, n)


# filter database for specific area and period
def CVdata_filt(df, x1, x2, y1, y2):
    dff = df[x2 > df.COORDFRONTX]
    dff = dff[dff.COORDFRONTX > x1]
    dff = dff[y2 > dff.COORDFRONTY]
    dff = dff[dff.COORDFRONTY > y1]
    return dff


# extract start/end time of a trip from P1 to P2
def CVdata_id(trip, p1, p2, d=40):
    startp = CVdata_filt(trip, p1[0]-d, p1[0]+d, p1[1]-d, p1[1]+d)
    endp = CVdata_filt(trip, p2[0]-d, p2[0]+d, p2[1]-d, p2[1]+d)
    tstart = startp['SIMSEC'].min()
    tend = endp['SIMSEC'].max()
    flag = (tend > (tstart+5)) and (startp.shape[0] > 0) \
           and (endp.shape[0] > 0)
    flag = int(flag == True)
    return np.array([flag, tstart, tend])


# return NOS, stop delay & travelled distance for a trip
def tripPM(trip):
    td = 0  # travel distance
    nos = 0  # number of stop counter!!!
    stopdelay = 0
    nos_cntrl = True  # vehicle is not stopped!
    for i in range(trip.shape[0]-1):
        deltaD = math.sqrt((trip.iat[i+1, 5] - trip.iat[i, 5])**2 +
                           (trip.iat[i+1, 6] - trip.iat[i, 6])**2)
        # deltaD = trip.iat[i + 1, 4] - trip.iat[i, 4] #POS version!
```

```python
        deltaT = trip.iat[i + 1, 0] - trip.iat[i, 0]
        td = td + deltaD
        speed = deltaD/(deltaT+0.00001)
        # speed less than 3.6km/h detected as stopped vehicle (1m/s)
        if abs(speed) < 1:
            stopdelay = stopdelay + deltaT
            if nos_cntrl:
                nos = nos + 1
                nos_cntrl = False
        else:
            nos_cntrl = True
    return np.array([nos, stopdelay, td])


# aggregate TrajDB for the movement from P1 to P2
def CV_aggregate(trajDB, p1, p2, speedff):
    # [id, TT, NOS, StopDelay, ControlDelay, Tstart, Tend]
    cvdata = np.zeros((0, 7))
    vehid = trajDB.iloc[:, [1]]
    # list of all vehicle's ID
    vehid = vehid.drop_duplicates()
    for index in range(vehid.shape[0]):
        # read the vehicle ID from the list
        id = vehid.iat[index, 0]
        # filter database for one vehicle-trip
        trip = trajDB[trajDB.NO == id]
        [flag, tstart, tend] = CVdata_id(trip, p1, p2)
        # The vehicle actually made the movement during the interval
        if flag > 0:
            # trim the trip to the exact movement
            trip = trip[trip.SIMSEC > tstart]
            trip = trip[trip.SIMSEC < tend]
            # sort data points
            trip = trip.sort_values('SIMSEC')
            [nos, stopdelay, td] = tripPM(trip)
            # travel time
            tt = tend - tstart
            controldelay = tt - 3.6 * td / speedff
            cvdata = np.append(cvdata, [[id, tt, nos, stopdelay,
                                          controldelay, tstart, tend]],
                               axis=0)
            # print(index, end=(' '))
    dfCVdata = pd.DataFrame(cvdata, columns=['id', 'TT', 'NOS',
                                             'StopDelay','ControlDelay',
                                             'Tstart', 'Tend'])
    return dfCVdata


# return 16 movements start/end points
def MovementDf(pn, pe, ps, pw):
    movement_df = pd.DataFrame([
    ['NbL', ps, pw], ['NbT', ps, pn], ['NbR', ps, pe], ['NbU', ps, ps],
```

```
    ['EbL', pw, pn], ['EbT', pw, pe], ['EbR', pw, ps], ['EbU', pw, pw],
    ['SbL', pn, pe], ['SbT', pn, ps], ['SbR', pn, pw], ['SbU', pn, pn],
    ['WbL', pe, ps], ['WbT', pe, pw], ['WbR', pe, pn], ['WbU', pe, pe]])
    return movement_df


# Graphically show the calculation progress
def loading_bar(progress):
    """Display a loading bar.

    Args:
    - progress (float): Progress percentage between 0 and 1. For example,
0.5 for 50%.
    """
    bar_length = 50  # adjust this value if you'd like a longer or shorter
loading bar

    arrow = '-' * int(round(progress * bar_length) - 1) + '>'
    spaces = ' ' * (bar_length - len(arrow))

    sys.stdout.write('\r[{}] {:.2f}%'.format(arrow + spaces, progress *
100))
    sys.stdout.flush()
# --------------------------------------------------

study_title = 'Sample intersection for TSSO analysis'
start_time = 0  # 600
end_time = 99999  # 3960/900
speedff = 60  # free flow speed in km/h
print('==>> VISSIM database imported & filtered for study',
      study_title, 'from', start_time, 'to', end_time, 'sec')

PN = np.array([125, -106])
PE = np.array([498, -150])
PS = np.array([495, -392])
PW = np.array([-5, -400])


def process(VissimDB):
    dff = VissimDB[end_time > VissimDB.SIMSEC]
    dff = dff[start_time < dff.SIMSEC]
    veh_id = dff.iloc[:,[1]]
    veh_id = veh_id.drop_duplicates()
    MyDF = MovementDf(PN,PE,PS,PW)
    cvdata = np.zeros((0, 8))
    dfCVdata = pd.DataFrame(cvdata, columns=['id', 'TT', 'NOS',
'StopDelay', 'ControlDelay', 'Tstart', 'Tend',
'movement'])
    for i in range(16):
        # print(MyDF.iat[i, 0])
        loading_bar(i / 15)
        dfCVdata_mov = CV_aggregate(dff, MyDF.iat[i, 1], MyDF.iat[i, 2],
```

```
speedff)
        dfCVdata_mov['movement'] = MyDF.iat[i, 0]
        dfCVdata = pd.concat([dfCVdata, dfCVdata_mov])
    return dfCVdata


import pandas as pd
import os

# Specify the folder where your text files are located
folder_path = '../Research_Simulation/SimDataNew/'

# List all files in the folder
all_files = os.listdir(folder_path)

# Filter to only get .txt files
txt_files = [file for file in all_files if file.endswith('.txt')]

# Iterate over each txt file
for file in txt_files:
    # Print file name
    print(file)

    # Construct the full file path
    file_path = os.path.join(folder_path, file)

    # Read the txt file as CSV
    df = pd.read_csv(file_path, sep=';')

    # Process trajectory data
    dfp = process(df)

    # Construct the output path
    output_path = os.path.join(folder_path, file.replace('.txt', '.csv'))

    # Save the dataframe as CSV
    dfp.to_csv(output_path, index=False)

    # Beep sound
    winsound.Beep(777, 1000)
    print()

print("Conversion completed!")
```

Part b: Python code for implementing STEP #2 & #3

```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import MultipleLocator
import math
import numpy as np
import matplotlib.patches as mpatches
import winsound


# Global configuration for visualizations
plt.style.use('seaborn-white')
plt.rcParams['axes.grid'] = True


# ------------------ FUNCTION DEFINITIONS ------------------

def load_data_from_folder(folder_path):
    """Load CSV files from the given folder."""
    all_files = os.listdir(folder_path)
    csv_files = [file for file in all_files if file.endswith('.csv')]
    return csv_files


def preprocess_data(df):
    """Clean and preprocess the dataframe for analysis."""

    # Filter df
    df = df[df.Tend < 4 * 3600]
    df = df.query('movement.str.contains("L") or
movement.str.contains("T")', engine="python").copy()

    # Remove the minimum delay observed from each movement delay value
    mov_list = ['NbL', 'SbT', 'EbL', 'WbT', 'SbL', 'NbT', 'WbL', 'EbT']
    for i in range(8):
        mov = mov_list[i]
        value = df.ControlDelay[(df.movement == mov)]
        print('Movement ', mov, 'minimum delay(sec) =', value.min())
        df.loc[df.movement == mov, 'ControlDelay'] =
df.ControlDelay[(df.movement == mov)] - value.min()
    return df


# =================< methodology core functions >=====================

def hcm_delay(x, ge, cl, sfr, te, kf):
    """Calculating delay based on the HCM model."""

    x1 = min(x, 1)
```

```
    uniform_delay = (cl * (1 - ge / cl) ** 2) / (2 - 2 * ge * x1 / cl)
    overload_delay = 15 * te * ((x - 1) + math.sqrt((x - 1) ** 2 + 240 * x
* cl / sfr / ge / te))
    total_delay = kf * uniform_delay + overload_delay
    return [uniform_delay, overload_delay, total_delay]


def gold(effective_green, cycle_length, delay, sfr, te, kf):
    """Degree of saturation back-calculation using Golden search
method."""

    # Golden section search inner function
    def gssrec(a, b, green, cl, delay, sfr, te, kf, tol=1e-5, h=None,
c=None, d=None, fc=None, fd=None):
        """Golden section search for recursion."""

        invphi = (math.sqrt(5) - 1) / 2  # 1 / phi
        invphi2 = (3 - math.sqrt(5)) / 2  # 1 / phi^2
        (a, b) = (min(a, b), max(a, b))
        if h is None: h = b - a
        if h <= tol: return (a, b)
        if c is None: c = a + invphi2 * h
        if d is None: d = a + invphi * h
        if fc is None: fc = (hcm_delay(c, green, cl, sfr, te, kf)[2] -
delay) ** 2
        if fd is None: fd = (hcm_delay(d, green, cl, sfr, te, kf)[2] -
delay) ** 2
        if fc < fd:  # to find the minimum
            return gssrec(a, d, green, cl, delay, sfr, te, kf, tol, h *
invphi, c=None, fc=None, d=c, fd=fc)
        else:
            return gssrec(c, b, green, cl, delay, sfr, te, kf, tol, h *
invphi, c=d, fc=fd, d=None, fd=None)

    xl = 0.3
    xu = 1.4
    er = 1e-5
    (c, d) = gssrec(xl, xu, effective_green, cycle_length, delay, sfr, te,
kf, er)
    return 0.5 * (c + d)


def hcm_opt(y, x, sg_timing, sfr, te, kf, CL_max=160):
    """Estimate optimal signal timing plan and movement delays based on
HCM method"""

    b1 = max((y[1] + y[2]), (y[5] + y[6]))
    b2 = max((y[3] + y[4]), (y[7] + y[8]))
    yt = b1 + b2
    if yt >= 1:
        cl_opt = CL_max
    else:
```

```
        cl_opt = min((1.5 * 12 + 5) / (1 - yt), CL_max)
    # cl_opt = min((1.5 * 12 + 5) / (1 - yt), CL_max)
    g_opt = y * ((cl_opt - 4 * 4) / yt)
    g_factor = [1, b1 / (y[1] + y[2]), b1 / (y[1] + y[2]), b2 / (y[3] +
y[4]), b2 / (y[3] + y[4]),
                b1 / (y[5] + y[6]), b1 / (y[5] + y[6]), b2 / (y[7] +
y[8]), b2 / (y[7] + y[8])]
    g_opt = g_opt * g_factor
    g_opt[0] = cl_opt
    x_opt = sg_timing.to_numpy()[0, :] * x * cl_opt / (g_opt *
sg_timing.to_numpy()[0, 0])
    delay_opt = np.zeros(9)
    for i in range(1, 9):
        x = x_opt[i]
        green_time = g_opt[i]
        cycle_time = cl_opt
        sat_flow_rate = sfr.to_numpy()[0, i]
        delay_opt[i] = hcm_delay(x, green_time + 1, cycle_time,
sat_flow_rate, te, kf)[2]
    delay_opt[0] = delay_opt.sum()
    return [g_opt, delay_opt, yt]


def model_points(signal_timing, sfr_values, lane_values, te, kf):
    """Produce delay model data points for any input values."""

    x_vph = np.arange(0., 1500, 1)
    x_LT = np.zeros(x_vph.shape[0])
    x_TH = np.zeros(x_vph.shape[0])
    LT_delay = np.zeros((x_vph.shape[0], 3))
    TH_delay = np.zeros((x_vph.shape[0], 3))
    for idx, val in enumerate(x_vph):
        x_LT[idx] = val * signal_timing[0] / sfr_values[1] /
lane_values[1] / (signal_timing[1] + 1)
        x_TH[idx] = val * signal_timing[0] / sfr_values[2] /
lane_values[2] / (signal_timing[2] + 1)
        LT_delay[idx] = hcm_delay(x_LT[idx], signal_timing[1] + 1,
signal_timing[0], sfr_values[1], te, kf)
        TH_delay[idx] = hcm_delay(x_TH[idx], signal_timing[2] + 1,
signal_timing[0], sfr_values[2], te, kf)
    return x_vph, x_LT, x_TH, LT_delay, TH_delay


def compute_traffic_delay(green_duration, cycle_duration, sat_flow_rate,
lane_quantity, te, kf):  # NOT USED
    # Generate vehicular traffic volume per hour values
    traffic_vph = np.arange(0., 1500, 1)

    # Calculate saturation degree using vectorized operations
    saturation_degree = traffic_vph * cycle_duration / (sat_flow_rate *
lane_quantity * (green_duration + 1))
```

```
    # Create an empty array for delay_values
    delay_values = np.empty_like(saturation_degree)

    # Apply hcm_delay function element-wise
    for idx, value in enumerate(saturation_degree):
        _, _, total_delay = hcm_delay(value, green_duration + 1,
cycle_duration, sat_flow_rate, te, kf)
        delay_values[idx] = total_delay

    return traffic_vph, saturation_degree, delay_values


# ===========< Vehicle delays to TSSO index functions >===================

def f_interval(df, minute):
    """Segregates the data into time intervals."""

    sec = 60 * minute  # interval time for calculations in seconds
    df['interval'] = df.iloc[:, [6]] / sec
    df['interval'] = df['interval'].astype(int)
    df.interval = df.interval * minute + minute
    return df


def f_mean(df_interval, minute, frac, sg_timing, lane, sfr, te, kf):
    """Calculates TMCs, delay mean values and back-calculate degree of
saturation."""

    df = df_interval.sample(frac=frac)  # data sampling for CV pen. rate
    mean = pd.DataFrame()
    for index, val in enumerate(df['movement'].drop_duplicates()):  # need
reset index???
        mov = df[df.movement == val].groupby(['interval']).mean()
        mov['Mov'] = val
        mov['vph'] = df[df.movement ==
val].groupby(['interval']).id.count() * 60 / (minute * frac)
        mean = pd.concat([mean, mov])
    mean = mean.reset_index()
    # mean = mean.reset_index(drop=True)
    for index in range(mean.shape[0]):
        delay_time = mean.at[index, 'ControlDelay']
        count = mean.at[index, 'vph']  # veh/hour
        green_time = sg_timing[mean.at[index, 'Mov']].values
        cycle_time = sg_timing.at[1, 'CL']
        sat_flow_rate = sfr.at[1, mean.at[index, 'Mov']]
        lanes = lane.at[1, mean.at[index, 'Mov']]
        # te = sfr.at[1, 'CL']  # this is also 'te'
        mean.at[index, 'X'] = gold(green_time + 1, cycle_time, delay_time,
sat_flow_rate, te, kf)  # back calculation
        mean.at[index, 'Y'] = mean.at[index, 'X'] * (green_time + 1) /
cycle_time
        mean.at[index, 'X_real'] = count * cycle_time / sat_flow_rate /
```

```
lanes / (green_time + 1)
    return mean


def f_clean(df_mean):
    """Cleans the mean data by appending missing intervals."""

    append = pd.DataFrame()
    for index, val in enumerate(df_mean['Mov'].drop_duplicates()):
        base = df_mean[df_mean.Mov == val].head(1)
        for key, intrvl in
enumerate(df_mean['interval'].drop_duplicates().values):
            df_row = df_mean[(df_mean.Mov == val) & (df_mean.interval ==
intrvl)]
            count = df_row.id.count()
            if count == 0:
                base.iat[0, 0] = intrvl
                append = pd.concat([append, base])
            else:
                base = df_row
    df_clean = pd.concat([df_mean, append])
    return append


def f_intersection(df_clean, sg_timing, sfr, te, kf):
    """Adds intersection-level statistics, including optimal signal
timings and estimated movement delays."""

    intersection = pd.DataFrame()
    for key, data in df_clean.groupby('interval'):
        mov_list = ['NbL', 'SbT', 'EbL', 'WbT', 'SbL', 'NbT', 'WbL',
'EbT']
        data.Mov = pd.Categorical(data.Mov, categories=mov_list,
ordered=True)
        data.sort_values('Mov', inplace=True)
        y = data['Y'].values[0:8]
        y = np.insert(y, 0, sg_timing.at[1, 'CL'])
        x = data['X'].values[0:8]
        x = np.insert(x, 0, 1000)  # x.max
        x_real = data['X_real'].values[0:8]
        x_real = np.insert(x_real, 0, 1000)  # x.max
        y_real = x_real * y / x
        [g_opt, delay_opt, yt] = hcm_opt(y, x, sg_timing, sfr, te, kf)
        [g_opt_real, delay_opt_real, yt_real] = hcm_opt(y_real, x_real,
sg_timing, sfr, te, kf)
        delay_calc = np.zeros(9)
        for i in range(1, 9):
            delay_calc[i] = hcm_delay(x_real[i], sg_timing.to_numpy()[0,
i] + 1, sg_timing.to_numpy()[0, 0],
                                      sfr.to_numpy()[0, i], te, kf)[2]
        delay_calc[0] = delay_calc.sum()
        data['X_real'] = x_real[1:9]
```

```
        data['delay_opt'] = delay_opt[1:9]
        data['delay_calc'] = delay_calc[1:9]
        data['g_opt'] = g_opt[1:9]
        data['delay_opt_real'] = delay_opt_real[1:9]
        data['g_opt_real'] = g_opt_real[1:9]
        intersection = pd.concat([intersection, data])
    return intersection


def f_index(df_intersection):
    """
    Calculate the index based on the given intersection dataframe.

    Parameters:
    - df_intersection: DataFrame containing intersection data.
    - label: Label to be added to the results.

    Returns:
    - df_intersection: Modified DataFrame with added label.
    - df_index: DataFrame with calculated indices.
    """

    Q = df_intersection.groupby(['interval']).sum()
    Q['index'] = np.maximum(Q['ControlDelay'] - Q['delay_opt'], 0)
    Q['index_real'] = np.maximum(Q['ControlDelay'] - Q['delay_opt_real'],
0)
    index = [df_intersection['interval'].drop_duplicates().values,
Q['index'].values, Q['index_real'].values]
    df_index = pd.DataFrame(np.transpose(index), columns=['interval',
'index', 'index_real'])
    return df_index


# ===============< TSSO index computation functions >=====================

# All calculations from vehicle delays to TSSO index
def compute_tsso_index(df, sg_timing, minute, frac, label, lane, sfr, te,
kf): #
    """Compute the Traffic Signal Sub-Optimality (TSSO) index:

    A wrapper function to process the dataset for a given time interval
and fraction.
    """

    # Consider X minutes intervals for time clustering
    df_interval = f_interval(df, minute)

    # Compute the mean values with the provided parameters
    df_mean = f_mean(df_interval, minute, frac, sg_timing, lane, sfr, te,
kf)

    # Clean the dataset and concatenate
```

```
    append = f_clean(df_mean)
    df_clean = pd.concat([df_mean, append])

    # Compute the intersection values
    df_intersection = f_intersection(df_clean, sg_timing, sfr, te, kf)

    # Calculate the index
    df_index = f_index(df_intersection)

    # Add label variable
    df_intersection['label'] = label
    df_index['label'] = label
    return (df_intersection, df_index)


# Repeat sampling & TSSO calculations N times (95 percentile error bands)
def compute_tsso_accuracy(df, sg_timing, minute, frac, N, lane, sfr, te,
kf):
    """Calculate TSSO 95 percentile accuracy (error bands)
    Repeat sampling and TSSO index calculations (N times with frac as
penetration rate)."""

    df_index_stat = pd.DataFrame()
    for i in range(N):
        [df_intersection, df_index] = compute_tsso_index(df, sg_timing,
minute, frac, i, lane, sfr, te, kf)
        df_index_stat = pd.concat([df_index_stat, df_index])
    df_index_stat = df_index_stat.reset_index()
    return df_index_stat


# Sensitivity analysis for PR & interval variations
def compute_sensitivity_pr_int(df, sg_timing, N, prs, intervals, lane,
sfr, te, kf):
    """Sensitivity analysis for penetration rate and interval variations
."""

    df_index_stat_add = pd.DataFrame()
    # Repeat TSSO calculations for all PR/interval variations
    for j, frac in enumerate(prs):
        for i, minute in enumerate(intervals):
            # Create label from the penetration rate and interval value
            label = "{}min-{}%PR".format(minute, int(frac*100))
            # Calculate TSSO for the selected pr & interval
            index_stat = compute_tsso_accuracy(df, sg_timing, minute,
frac, N, lane, sfr, te, kf)
            index_stat['label'] = label
            df_index_stat_add = pd.concat([df_index_stat_add, index_stat])
    return df_index_stat_add


# Sensitivity analysis for sfr variations
```

```
def compute_sensitivity_sfr(df, sg_timing, lane, sfr, te, kf):
    """Sensitivity analysis for sfr variations ."""

    minute = 30
    frac = 1
    sample = np.array(['SFR-base', 'TH+10%', 'TH-10%', 'LT+10%', 'LT-
10%'])
    mov_columns = ['CL', 'NbL', 'SbT', 'EbL', 'WbT', 'SbL', 'NbT', 'WbL',
'EbT']
    color_map = {sample[0]: 'black', sample[1]: 'blue', sample[2]: 'red',
sample[3]: 'green', sample[4]: 'gray'}

    # base calculation
    [df_intersection, df_index_add] = compute_tsso_index(df, sg_timing,
minute, frac, sample[0], lane, sfr, te, kf)
    # Calculate values for altered parameter
    for i in range(1, len(sample)):
        if i == 1:
            sfr_mod = pd.DataFrame([[15, 1500, 2200, 1500, 2200, 1500,
2200, 1500, 2200]], index=[1], columns=mov_columns)
        elif i == 2:
            sfr_mod = pd.DataFrame([[15, 1500, 1800, 1500, 1800, 1500,
1800, 1500, 1800]], index=[1], columns=mov_columns)
        elif i == 3:
            sfr_mod = pd.DataFrame([[15, 1650, 2000, 1650, 2000, 1650,
2000, 1650, 2000]], index=[1], columns=mov_columns)
        elif i == 4:
            sfr_mod = pd.DataFrame([[15, 1350, 2000, 1350, 2000, 1350,
2000, 1350, 2000]], index=[1], columns=mov_columns)
        [auto1, auto2] = compute_tsso_index(df, sg_timing, minute, frac,
sample[i], lane, sfr_mod, te, kf)
        df_index_add = pd.concat([df_index_add, auto2])
    return df_index_add, color_map


# ===============< visualization functions >========================

# Plot delay model sensitivity to sfr values
def plot_delay_model(signal_timing, lane_values, te, kf):  # HCM delay
model sensitivity to SFR & g/c


    # inner function to plot delay model lane
    def plot_graph(subplot_index, x_LT, x_TH, LT_delays, TH_delays,
sfr_values, xlim, ylim, xlabel, title):
        plt.subplot(2, 1, subplot_index)
        # plt.title(title)
        plt.xlabel(xlabel)
        plt.ylabel("Average Vehicle Delay (sec)")

        # Line styles and labels for the base SFR values
        if sfr_values == [15, 1500, 2000, 1500, 2000, 1500, 2000, 1500,
```

```
2000]:
            plt.plot(x_LT, LT_delays, 'red', linewidth=2, linestyle='--',
label='LT Model')
            plt.plot(x_TH, TH_delays, 'blue', linewidth=2, linestyle='--',
label='TH Model')
        else:
            plt.plot(x_LT, LT_delays, 'salmon', linewidth=1)
            plt.plot(x_TH, TH_delays, 'lightblue', linewidth=1)

        plt.xlim(xlim)
        plt.ylim(ylim)
        plt.legend()
        return

    fig, ax = plt.subplots(2, 1)
    plt.suptitle("Sensitivity of the Delay Model to a 20% Variation in
SFR")

    sfr_values_list = [
        [15, 1500, 2000, 1500, 2000, 1500, 2000, 1500, 2000],
        [15, 1650, 2200, 1650, 2200, 1650, 2200, 1650, 2200],
        [15, 1350, 1800, 1350, 1800, 1350, 1800, 1350, 1800]
    ]

    for sfr_values in sfr_values_list:
        x_vph, x_LT, x_TH, LT_delay, TH_delay =
model_points(signal_timing, sfr_values, lane_values, te, kf)

        # Plot for Traffic Movement Counts
        plot_graph(1, x_vph, x_vph, LT_delay[:, 2], TH_delay[:, 2],
sfr_values, (0, 1400), (0, 200),
                   "Traffic Movement Counts (vph)", "Average Vehicle Delay
over Traffic Movement Counts")

        # Plot for Degree of Saturation
        plot_graph(2, x_LT, x_TH, LT_delay[:, 2], TH_delay[:, 2],
sfr_values, (0, 1.4), (0, 200),
                   "Degree of Saturation (x)", "Average Vehicle Delay over
Degree of Saturation (X)")

    # Adding grid to the plots
    ax[0].grid(True)
    ax[1].grid(True)
    return


# Visualize vehicle delays vs degree of saturation, and TSSO index for all
scenarios
def plot_delay_x(df_intersection, df_index, df_index_stat, signal_timing,
sfr_values, lane_values, te, kf):
    """Plot HCM delay model with dots for actual vehicle delays-degree of
saturation pairs."""
```

```python
    # Plot HCM delay model and X'-X diagrams to show X back-calculation
accuracy
    fig, ax = plt.subplots(2, 2, figsize=(12, 10))
    plt.suptitle("HCM Delay Model")

    # Automatically create the scenario mapper for color coding
    unique_scenarios = sorted(df_intersection['scenario'].unique())
    scenario_mapper = {scenario: i + 1 for i, scenario in
enumerate(unique_scenarios)}

    # Apply the mapper to the 'scenario' column
    df_intersection['scenario_num'] =
df_intersection['scenario'].map(scenario_mapper)

    # ---------------- LT X_real vs X ----------------
    plt.subplot(2, 2, 1)
    plt.xlabel("Real degree of saturation (x)")
    plt.ylabel("Back-calculated Degree of saturation (x')")
    scatter =
plt.scatter(df_intersection.X_real[df_intersection.Mov.str.contains("L")],
                df_intersection.X[df_intersection.Mov.str.contains("L")],

c=df_intersection['scenario_num'][df_intersection.Mov.str.contains("L")],
                label='LT data', s=10, cmap='viridis')

    # Customize colorbar ticks to represent scenario values
    cbar = plt.colorbar(scatter, ticks=list(scenario_mapper.values()))
    cbar.set_label('Scenario Value')
    cbar.set_ticklabels(list(scenario_mapper.keys()))

    plt.plot([0, 2], [0, 2], 'red')
    plt.xlim(0, 1.4)
    plt.ylim(0, 1.4)
    plt.legend()

    # ---------------- TH X_real vs X ----------------
    plt.subplot(2, 2, 2)
    plt.xlabel("Real degree of saturation (x)")
    plt.ylabel("Back-calculated Degree of saturation (x')")
    scatter =
plt.scatter(df_intersection.X_real[df_intersection.Mov.str.contains("T")],
                df_intersection.X[df_intersection.Mov.str.contains("T")],

c=df_intersection['scenario_num'][df_intersection.Mov.str.contains("T")],
                label='TH data', s=10, cmap='viridis')

    # Customize color bar ticks to represent scenario values
    cbar = plt.colorbar(scatter, ticks=list(scenario_mapper.values()))
    cbar.set_label('Scenario Value')
    cbar.set_ticklabels(list(scenario_mapper.keys()))
```

```
    plt.plot([0, 2], [0, 2], 'blue')
    plt.xlim(0, 1.4)
    plt.ylim(0, 1.4)
    plt.legend()

    # ---------------- LT Delay vs X ----------------
    plt.subplot(2, 2, 3)
    plt.xlabel("Degree of saturation (x)")
    plt.ylabel("Average vehicle delay (sec)")
    scatter =
plt.scatter(df_intersection.X_real[df_intersection.Mov.str.contains("L")],

df_intersection.ControlDelay[df_intersection.Mov.str.contains("L")],

c=df_intersection['scenario_num'][df_intersection.Mov.str.contains("L")],
                label='LT data', s=10, cmap='viridis')

    # Customize colorbar ticks to represent scenario values
    cbar = plt.colorbar(scatter, ticks=list(scenario_mapper.values()))
    cbar.set_label('Scenario Value')
    cbar.set_ticklabels(list(scenario_mapper.keys()))

    [x_vph, x_LT, x_TH, LT_delay, TH_delay] = model_points(signal_timing,
sfr_values, lane_values, te, kf)
    plt.plot(x_LT, LT_delay[:, 2], 'red', linewidth=2, linestyle='--',
label='LT Model')
    plt.xlim(0, 1.4)
    plt.ylim(0, 200)
    plt.legend()

    # ---------------- TH Delay vs X ----------------
    plt.subplot(2, 2, 4)
    plt.xlabel("Degree of saturation (x)")
    plt.ylabel("Average vehicle delay (sec)")
    scatter =
plt.scatter(df_intersection.X_real[df_intersection.Mov.str.contains("T")],

df_intersection.ControlDelay[df_intersection.Mov.str.contains("T")],

c=df_intersection['scenario_num'][df_intersection.Mov.str.contains("T")],
                label='TH data', s=10, cmap='viridis')

    # Customize colorbar ticks to represent scenario values
    cbar = plt.colorbar(scatter, ticks=list(scenario_mapper.values()))
    cbar.set_label('Scenario Value')
    cbar.set_ticklabels(list(scenario_mapper.keys()))

    plt.plot(x_TH, TH_delay[:, 2], 'blue', linewidth=2, linestyle='--',
label='TH Model')
    plt.xlim(0, 1.4)
    plt.ylim(0, 200)
    plt.legend()
```

```python
    plt.tight_layout()

    # --------------- Plot TSSO index -----------------------

    # Filter the dataframe for the desired label
    df_filtered = df_index_stat[df_index_stat['label'] == '30min-20%PR']
    # df_filtered = df_index


    # Update the FacetGrid to use scenario as the facet column
    g = sns.FacetGrid(df_filtered, col='scenario', col_wrap=4)
    g.map_dataframe(sns.lineplot, x="interval", y="index", color='blue',
                    label='Estimated index value with 95% pi error bands',
                    linewidth=2, linestyle='--', errorbar="pi")
    g.set_axis_labels("Simulation time(min)", "TSSO(sec)")
    plt.suptitle("TSSO Index: 30-min Interval, 20% Penetration")
    tick_interval = 60
    for ax in g.axes.flat:
        ax.xaxis.set_major_locator(MultipleLocator(base=tick_interval))
        ax.set_xlim(0, 240)

    g.tight_layout()
    sns.despine()

    return


# Visualize the TSSO index: HCM delay model, x & delay, optimal green, &
TSSO trends
def plot_tsso_index(df_intersection, df_index_stat, signal_timing,
sfr_values, lane_values, te, kf):
    """
        Plot HCM delay model
            x real(counts/sfr)
            vehicle delay(all vehicles)
            optimal signal timing trend
            TSSO index trend"""

    mov_list = ['NbL', 'SbT', 'EbL', 'WbT', 'SbL', 'NbT', 'WbL', 'EbT']

    # Plot HCM delay model and X'-X diagrams to show X back-calculation
accuracy
    fig, ax = plt.subplots(2, 1)  # , figsize=(8, 10))
    plt.suptitle("HCM Delay Model")
    # ------------------------------------
    plt.subplot(2, 1, 1)
    # plt.title("Degree of saturation: Back-calculation(x') vs Real (x)")
    plt.xlabel("Real degree of saturation (x)")
    plt.ylabel("Back-calculated Degree of saturation (x')")

plt.scatter(df_intersection.X_real[(df_intersection.Mov.str.contains("L"))
```

```
],

df_intersection.X[(df_intersection.Mov.str.contains("L"))],
                c='red', label='LT data', s=10)

plt.scatter(df_intersection.X_real[(df_intersection.Mov.str.contains("T"))
],

df_intersection.X[(df_intersection.Mov.str.contains("T"))],
                c='b', label='TH data', s=10)
    plt.plot([0, 2], [0, 2], 'black')
    plt.xlim(0, 1.4)
    plt.ylim(0, 1.4)
    plt.legend()
    # ------------------------------------
    plt.subplot(2, 1, 2)
    # plt.title("Average vehicle delay vs degree of saturation(X)")
    plt.xlabel("Degree of saturation (x)")
    plt.ylabel("Average vehicle delay (sec)")

plt.scatter(df_intersection.X_real[(df_intersection.Mov.str.contains("L"))
],

df_intersection.ControlDelay[(df_intersection.Mov.str.contains("L"))],
                c='red', label='LT data', s=10)

plt.scatter(df_intersection.X_real[(df_intersection.Mov.str.contains("T"))
],

df_intersection.ControlDelay[(df_intersection.Mov.str.contains("T"))],
                c='blue', label='TH data', s=10)
    [x_vph, x_LT, x_TH, LT_delay, TH_delay] = model_points(signal_timing,
sfr_values, lane_values, te, kf)
    plt.plot(x_LT, LT_delay[:, 2], 'red', linewidth=2, linestyle='--',
label='LT Model')
    plt.plot(x_TH, TH_delay[:, 2], 'blue', linewidth=2, linestyle='--',
label='TH Model')
    plt.xlim(0, 1.4)
    plt.ylim(0, 200)
    plt.legend()

    # Plot X & AVG delays of LT & TH movements
    plt.subplots(2, 2, sharex='col', sharey='row')
    plt.suptitle("Degree of saturation(X) & average vehicle delays for LT
& TH movements")
    for i in range(8):
        mov = mov_list[i]
        ii = (i + 1) % 2
        plt.subplot(2, 2, 2-ii)
        plt.title("Degree of saturation(X) from TMC & SFR")
        plt.plot(df_intersection.interval[(df_intersection.Mov == mov)],
                 df_intersection.X_real[(df_intersection.Mov == mov)],
```

```
label=mov, linewidth=2)
        tick_interval = 60

plt.gca().xaxis.set_major_locator(MultipleLocator(base=tick_interval))
        plt.xlim(0, 240)
        # plt.xlabel("Simulation time (min)")
        plt.ylabel("Degree of saturation (X)")
        plt.legend()#loc='upper left')
        plt.subplot(2, 2, 4-ii)
        plt.title("Average vehicle delays")
        plt.plot(df_intersection.interval[(df_intersection.Mov == mov)],
                 df_intersection.ControlDelay[(df_intersection.Mov ==
mov)], label=mov, linewidth=2)
        tick_interval = 60

plt.gca().xaxis.set_major_locator(MultipleLocator(base=tick_interval))
        plt.xlim(0, 240)
        plt.ylim(bottom=0)
        plt.xlabel("Simulation time (min)")
        plt.ylabel("AVG delay (sec)")
        plt.legend() #loc='upper left')

    # Plot optimal signal timing as stacked diagram
    # inner function for signal timing visualization
    def stack_plot(x_val, y_val):
        hatches = [".", "\\", "o", "//"]
        pal = ['blue', 'orange', 'green', 'red']
        L1 = mpatches.Patch(facecolor=pal[0], alpha=1, hatch='.',
label=list(y_val)[0])
        L2 = mpatches.Patch(facecolor=pal[1], alpha=1, hatch=r'\\\\',
label=list(y_val)[1])
        L3 = mpatches.Patch(facecolor=pal[2], alpha=1, hatch='o',
label=list(y_val)[2])
        L4 = mpatches.Patch(facecolor=pal[3], alpha=1, hatch=r'////',
label=list(y_val)[3])
        stacks = plt.stackplot(x_val, y_val.values(), labels=y_val.keys(),
colors=pal, alpha=0.75, baseline="zero")
        for stack, hatch in zip(stacks, hatches):
            stack.set_hatch(hatch)
        plt.legend(handles=[L4, L3, L2, L1], loc=2)
        # plt.ylim(0, 150)
        return

    # Plot optimal signal green times
    x_values = df_intersection.interval[(df_intersection.Mov == 'NbL')]
    g_opt_R1 = {
        'NbL': df_intersection.g_opt[(df_intersection.Mov == 'NbL')],
        'SbT': df_intersection.g_opt[(df_intersection.Mov == 'SbT')],
        'EbL': df_intersection.g_opt[(df_intersection.Mov == 'EbL')],
        'WbT': df_intersection.g_opt[(df_intersection.Mov == 'WbT')],
    }
    g_opt_R2 = {
```

```
        'SbL': df_intersection.g_opt[(df_intersection.Mov == 'SbL')],
        'NbT': df_intersection.g_opt[(df_intersection.Mov == 'NbT')],
        'WbL': df_intersection.g_opt[(df_intersection.Mov == 'WbL')],
        'EbT': df_intersection.g_opt[(df_intersection.Mov == 'EbT')],
    }
    plt.subplots(2, 1, sharex='col')
    plt.suptitle("Optimal signal green times")

    plt.subplot(2, 1, 1)
    stack_plot(x_values, g_opt_R1)
    plt.title('Ring barrier 1')
    plt.ylabel("Green time(sec)")
    # plt.legend(loc='upper left')

    plt.subplot(2, 1, 2)
    stack_plot(x_values, g_opt_R2)
    plt.title('Ring barrier 2')
    plt.xlabel("Simulation time (min)")
    plt.ylabel("Green time(sec)")
    # plt.legend(loc='upper left')
    tick_interval = 60
    plt.gca().xaxis.set_major_locator(MultipleLocator(base=tick_interval))
    plt.xlim(0, 240)

    # Plot TSSO index
    plt.figure()

    # Create the title using f-string
    title = f"Traffic Signal Sub-Optimality index: PR = {int(frac * 100)}%
N={N}"
    plt.title(title)
    plt.xlabel("Simulation time (min)")
    plt.ylabel("TSSO index (sec)")
    sns.lineplot(data=df_index_stat, x="interval", y="index",
color='blue',
                 label='95% Percentile interval', linewidth=2,
                 linestyle='--', errorbar="pi")
    plt.legend(loc=2)
    plt.ylim(0, 200)
    tick_interval = 60
    plt.gca().xaxis.set_major_locator(MultipleLocator(base=tick_interval))
    plt.xlim(0, 240)
    return


# Plot PR-interval sensitivity analysis results
def plot_sensitivity_pr_int(df_index_stat_add):

    g = sns.FacetGrid(df_index_stat_add, col='label', col_wrap=4,
height=2, aspect=1.5)
    g.map_dataframe(sns.lineplot, x="interval", y="index", color='blue',
                    # label='Estimated index value with 95% pi error
```

```
bands',
                    linewidth=2, linestyle='--', errorbar="pi")
    g.set_axis_labels("Simulation time(min)", "TSSO(sec)")
    tick_interval = 60
    for ax in g.axes.flat:
        ax.xaxis.set_major_locator(MultipleLocator(base=tick_interval))
        ax.set_xlim(0, 240)

    return



# Function to plot sfr sensitivity analysis results (simulation TSSO
trend)
def plot_sensitivity_sfr(df_index_add, color_map):
    # Create a FacetGrid, facetting by the 'scenario' variable
    g = sns.FacetGrid(df_index_add, col='scenario', col_wrap=4, height=2,
aspect=1.5)

    # Apply the plotting function to each subplot
    g = g.map_dataframe(sns.lineplot, x='interval', y='index',
hue='label', palette=color_map)

    # Add titles, labels, and legend
    g.set_titles(col_template="{col_name} scenarios")
    g.set_axis_labels("Simulation time (min)", "TSSO index (sec)")
    g.add_legend()

    # Customize x-axis ticks
    tick_interval = 60
    for ax in g.axes.flat:
        ax.xaxis.set_major_locator(MultipleLocator(base=tick_interval))
        ax.set_xlim(0, 240)

    # Save the plots to disk as a PNG file
    # Adjust the size to 9cm x 12cm (converted to inches)
    g.savefig("facet_plot_TSSO_sensitivity.png", dpi=300,
bbox_inches='tight')
    # , figsize=(4.72, 3.54))

    return


# --------------------- MAIN ANALYSIS FUNCTION --------------------

def main_analysis(csv_files, sg_timing, lane, sfr, minute, frac, N, te,
kf, prs, intervals):
    """Main function for analyzing the traffic simulation data."""

    # Lists to store the results for each dataframe in the tuple
    all_df_intersection = []
```

```
    all_df_index_stat = []
    all_df_index_add = []
    # all_color_map = []
    all_df_index_stat_add = []

    # Iterate through each CSV file and conduct the analysis
    for file in csv_files:
        print(f"Analyzing file: {file}")
        file_path = os.path.join(folder_path, file)
        dataframe = pd.read_csv(file_path, sep=',')

        # Complete TSSO analysis for the loaded scenario:
        # Preprocess data
        df = preprocess_data(dataframe)

        # Compute TSSO index
        [df_intersection, df_index] = compute_tsso_index(df, sg_timing,
minute, 1, 'base', lane, sfr, te, kf)

        # Compute TSSO accuracy
        df_index_stat = compute_tsso_accuracy(df, sg_timing, minute, frac,
N, lane, sfr, te, kf)

        # SFR Sensitivity Analysis
        [df_index_add, color_map] = compute_sensitivity_sfr(df, sg_timing,
lane, sfr, te, kf)

        # PR & interval Sensitivity Analysis
        df_index_stat_add = compute_sensitivity_pr_int(df, sg_timing, N,
prs, intervals, lane, sfr, te, kf)

        # Extracting the scenario name from the filename
        scenario_name = os.path.splitext(file)[0]

        # Add the scenario name to each result dataframe
        df_intersection['scenario'] = scenario_name
        df_index_stat['scenario'] = scenario_name
        df_index_add['scenario'] = scenario_name
        df_index_stat_add['scenario'] = scenario_name

        # Append dataframes to the corresponding lists
        all_df_intersection.append(df_intersection)
        all_df_index_stat.append(df_index_stat)
        all_df_index_add.append(df_index_add)
        all_df_index_stat_add.append(df_index_stat_add)

        print(f"Analysis for {file} completed!")

    # Concatenate results
    final_df_intersection = pd.concat(all_df_intersection,
ignore_index=True)
    final_df_index_stat = pd.concat(all_df_index_stat, ignore_index=True)
```

```
    final_df_index_add = pd.concat(all_df_index_add, ignore_index=True)
    final_df_index_stat_add = pd.concat(all_df_index_stat_add,
ignore_index=True)

    # Save the results to EXCEL sheets
    with pd.ExcelWriter('final_output.xlsx', engine='xlsxwriter') as
writer:
        final_df_intersection.to_excel(writer, sheet_name='Intersection',
index=False)
        final_df_index_stat.to_excel(writer, sheet_name='Index_Stat',
index=False)
        final_df_index_add.to_excel(writer, sheet_name='Index_Add',
index=False)
        final_df_index_stat_add.to_excel(writer,
sheet_name='Index_Stat_Add', index=False)

    return


# ------------------ MAIN VISUALIZATIONS FUNCTION ------------------

def main_visualization(color_map, sfr_values, lane_values):
    """Main function for visualizing TSSO results."""

    # Read saved Excel sheets back into dataframes
    final_df_intersection = pd.read_excel('final_output.xlsx',
sheet_name='Intersection')
    final_df_index_stat = pd.read_excel('final_output.xlsx',
sheet_name='Index_Stat')
    final_df_index_add = pd.read_excel('final_output.xlsx',
sheet_name='Index_Add')
    final_df_index_stat_add = pd.read_excel('final_output.xlsx',
sheet_name='Index_Stat_Add')

    # Get all unique scenarios from any one of the dataframes
    all_scenarios = final_df_intersection['scenario'].unique()

    # Plot results for each requested scenario
    while True:  # Infinite loop to keep asking the user for input until
they choose to exit

        # Ask user for scenario input
        scenario = input("Please enter a scenario value (S1 to S8) or type
'end' to exit: ")

        # Check if user wants to exit
        if scenario == 'end':
            break
        # Check if entered scenario is valid
        elif scenario not in all_scenarios:
            print("The answer should be S1 to S8 or 'end' to exit the
code.")
```

```
            continue

        # If scenario is valid:
        # Filter dataframes for the entered scenario
        current_df_intersection =
final_df_intersection[final_df_intersection['scenario'] == scenario]
        current_df_index_stat =
final_df_index_stat[final_df_index_stat['scenario'] == scenario]
        current_df_index_add =
final_df_index_add[final_df_index_add['scenario'] == scenario]
        current_df_index_stat_add =
final_df_index_stat_add[final_df_index_stat_add['scenario'] == scenario]

        # Visualization for the entered scenario:

        # Plot actual x, delay, optimal signal plan, TSSO, HCM delay model
        plot_tsso_index(current_df_intersection, current_df_index_stat,
signal_timing, sfr_values, lane_values, te, kf)

        # Sensitivity Analysis and Visualization (PR & interval)
        plot_sensitivity_pr_int(current_df_index_stat_add)

        # Show the plots
        plt.show()

    # Plot delay model sensitivity to sfr values
    plot_delay_model(signal_timing, lane_values, te, kf)

    # SFR Sensitivity Visualizations
    plot_sensitivity_sfr(final_df_index_add, color_map)

    # Plot HCM delay model with all dots & TSSO accuracy for 30min-5%
facet by scenarios
    plot_delay_x(final_df_intersection, final_df_index_add,
final_df_index_stat_add, signal_timing, sfr_values, lane_values, te, kf)

    return
# *********************************************
# import pandas as pd
# import matplotlib.pyplot as plt

def main_plot(color_map, sfr_values, lane_values):
    """Main function for visualizing TSSO results."""

    # Read saved Excel sheets back into dataframes
    final_df_intersection = pd.read_excel('final_output.xlsx',
sheet_name='Intersection')
    final_df_index_stat = pd.read_excel('final_output.xlsx',
sheet_name='Index_Stat')
    final_df_index_add = pd.read_excel('final_output.xlsx',
sheet_name='Index_Add')
    final_df_index_stat_add = pd.read_excel('final_output.xlsx',
```

```
sheet_name='Index_Stat_Add')

    # Get all unique scenarios from any one of the dataframes
    all_scenarios = final_df_intersection['scenario'].unique()

    # Automatically iterate over all scenarios
    for scenario in all_scenarios:



        # Filter dataframes for the current scenario
        current_df_intersection =
final_df_intersection[final_df_intersection['scenario'] == scenario]
        current_df_index_stat =
final_df_index_stat[final_df_index_stat['scenario'] == scenario]
        current_df_index_add =
final_df_index_add[final_df_index_add['scenario'] == scenario]
        current_df_index_stat_add =
final_df_index_stat_add[final_df_index_stat_add['scenario'] == scenario]

        # Visualization for the current scenario:

        # Plot actual x, delay, optimal signal plan, TSSO, HCM delay model
        # Assuming the function is defined, or replace with your own
function
        # plot_tsso_index(current_df_intersection, current_df_index_stat,
signal_timing, sfr_values, lane_values, te, kf)

        # Sensitivity Analysis and Visualization (PR & interval)
        plot_sensitivity_pr_int(current_df_index_stat_add)

        # Save the plots
        file_name = f"plot_for_{scenario}.png"
        plt.savefig(file_name, dpi=300, bbox_inches='tight')

    # Other plots that don't need scenario filtering can go here
    # For example:

    # Plot delay model sensitivity to sfr values
    # plot_delay_model(signal_timing, lane_values, te, kf)

    # Plot HCM delay model with all dots & TSSO accuracy for 30min-5%
facet by scenarios
    # plot_delay_x(final_df_intersection, final_df_index_add,
final_df_index_stat_add, signal_timing, sfr_values, lane_values, te, kf)

    return

# ********************************************

# ------------------ INITIALIZATION & DATA LOADING ------------------
```

```python
# Configuration parameters
signal_timing = [96, 15, 25, 15, 25, 15, 25, 15, 25]
sfr_values = [15, 1500, 2000, 1500, 2000, 1500, 2000, 1500, 2000]
lane_values = [0, 1, 2, 1, 2, 1, 2, 1, 2]
minute = 30
frac = 0.1
N = 30
te = 15
kf = 1
prs = [0.2, 0.1, 0.05]
intervals = [15, 30, 45, 60]
mov_columns = ['CL', 'NbL', 'SbT', 'EbL', 'WbT', 'SbL', 'NbT', 'WbL',
'EbT']
sg_timing = pd.DataFrame([signal_timing], index=[1], columns=mov_columns)
sfr = pd.DataFrame([sfr_values], index=[1], columns=mov_columns)
lane = pd.DataFrame([lane_values], index=[1], columns=mov_columns)
# -----------
sample = np.array(['SFR-base', 'TH+10%', 'TH-10%', 'LT+10%', 'LT-10%'])
mov_columns = ['CL', 'NbL', 'SbT', 'EbL', 'WbT', 'SbL', 'NbT', 'WbL',
'EbT']
color_map = {sample[0]: 'black', sample[1]: 'blue', sample[2]: 'red',
sample[3]: 'green', sample[4]: 'gray'}

# ------------------ TSSO ANALYSIS FOR ALL SCENARIOS DATABASE ------------

# data folder path
folder_path = '../SimDataNew'

csv_files = [f for f in os.listdir(folder_path) if f.endswith('.csv')]

# Analyzing the vehicle delays csv files and save results into final csv
files
# main_analysis(csv_files, sg_timing, lane, sfr, minute, frac, N, te, kf,
prs, intervals)

# Visualize TSSO results from final csv files
main_visualization(color_map, sfr_values, lane_values)
# main_plot(color_map, sfr_values, lane_values)
print("All analyses completed!")

# Show the plots
plt.show()
```