Optimising Rule-Based Variable Speed Limit Control: A Data-Driven Approach for Parameter Calibration

Highlights

- A systematic data-driven framework is proposed for calibration of rule-based VSL algorithms.
- Introduces a novel Speed-Based Queue Detection module using Smoothed Difference of Squared Speeds for dynamic queue detection.
- Novel integration of Adaptive Smoothing Method with Edge Detection for accurate back of queue estimation on congested motorways.
- The framework automates VSL parameter calibration, minimizing human intervention and enhancing operational efficiency in traffic management.

Abstract

Variable Speed Limit (VSL) systems have evolved from static limits to dynamic control strategies aimed at mitigating congestion and improving safety. While advanced Reinforcement Learning (RL)-based approaches have been explored, rule-based VSL remains the state-of—the-practice due to its operational simplicity, explainability, and lower computational requirements. These systems typically rely on predefined activation and deactivation thresholds to detect congestion at loop detector locations, followed by estimating queue propagation rates to determine back-of-queue positions and apply upstream speed reductions. However, these parameters are manually calibrated based on expert judgment and historical data, making them time-consuming, error-prone, and less adaptive to fluctuating traffic conditions.

This study addresses this gap by proposing a data-driven framework for automating the calibration of rule-based VSL parameters. A critical aspect of effective VSL implementation is the ability to estimate back-of-queue locations accurately, as improper calibration can lead to operational inefficiencies or safety risks. Conventional loop detectors, typically spaced every 500 meters or so, are spatially sparse and fail to provide realistic queue profiles, limiting VSL systems' in protecting the back of congestion. To overcome this limitation, this study introduces an Adaptive Smoothing Method with Edge Detection (ASM-ED) to generate detailed queue profiles by performing a traffic flow theory informed interpolation of the sparse detector data and identifying congestion propagation patterns. Additionally, a Speed-Based Congestion Detection Module using a Smoothed Difference of Squared Speeds (SDSS) module is proposed to dynamically identify congestion onset and dissipation.

By integrating these modules, the proposed framework systematically improves VSL activation and deactivation thresholds, reducing reliance on manual calibration. The results demonstrate that the approach enhances the adaptability of rule-based VSL systems while preserving their interpretability and ease of implementation. This study provides a practical, scalable solution for agencies seeking to improve congestion management without the complexities associated with RL-based control.

1 Introduction

Effective motorway control is critical for managing congestion and enhancing safety, and Variable Speed Limit (VSL) systems have emerged as a key tool for optimal motorway control. Over the past decades, the evolution of VSL systems—from static limits to dynamic, model-based approaches—has seen some significant advancements.

1.1 Evolution of VSL Control Systems

Early VSL systems were largely built around predefined thresholds, where speed limits were triggered by measured traffic states—such as flow, density, or occupancy, and sometimes by weather conditions (Rämä, 1999; Smulders & Helleman, 1998). These rule-based systems aimed primarily at speed harmonization to reduce crash potential and improve safety (Hegyi et al., 2005). Although early field implementations showed improvements in capacity and reduced speed variability (Hoogen & Smulders, 1994; Zackor, 1991), the inherent limitations of discrete thresholds often led to oscillations in traffic flow. To address these issues, researchers integrated fuzzy logic into rule-based systems, enabling smoother transitions by converting crisp inputs into fuzzy variables—thus accommodating the inherent uncertainty in traffic conditions (D. Li & Ranjitkar, 2015).

As the limitations of rule-based systems became evident, the focus shifted towards analytical and control theory-based methods. These approaches introduced predicted future traffic states and aimed to prevent congestion rather than simply reacting to it. Model Predictive Control (MPC) emerged as a prominent framework, where short-term traffic predictions based on macroscopic models (e.g., the second order model and METANET) informed the optimal VSL settings. Such methods effectively reduced shockwave occurrences and stabilized traffic flow (Hegyi et al., 2005, 2008). Other refinements involved incorporating real-time state estimation—such as extended Kalman filters—to improve the responsiveness and robustness of VSL systems (Alessandri et al., 1999). Despite the complexity and computational demands of these models, they provided a more coordinated and proactive means to manage traffic dynamics.

More recently, the advent of machine learning, and specifically reinforcement learning (RL), has transformed VSL control strategies. In RL-based systems, the VSL controller learns to optimize speed limits through continuous interaction with the traffic environment, balancing the trade-off between travel time, congestion, and safety. Various studies have demonstrated significant reductions in travel time and improved throughput using different RL techniques—from Q-learning and deep RL to multi-agent frameworks (Gregurić et al., 2022; Heyao et al., 2024; Ke et al., 2021; Kušić et al., 2018, 2020; Z. Li et al., 2017, 2021; Sun et al., 2024; Vinitsky et al., 2018; Vrbanić et al., 2021; Walraven et al., 2016; Wang et al., 2019; Zhu & Ukkusuri, 2014). The dynamic learning capability of these systems helps to overcome the unpredictability of real-world traffic conditions and to customise control actions to specific network characteristics. Despite these advancements, such advanced approaches for VSL control remain primarily in the research domain rather than widespread deployment.

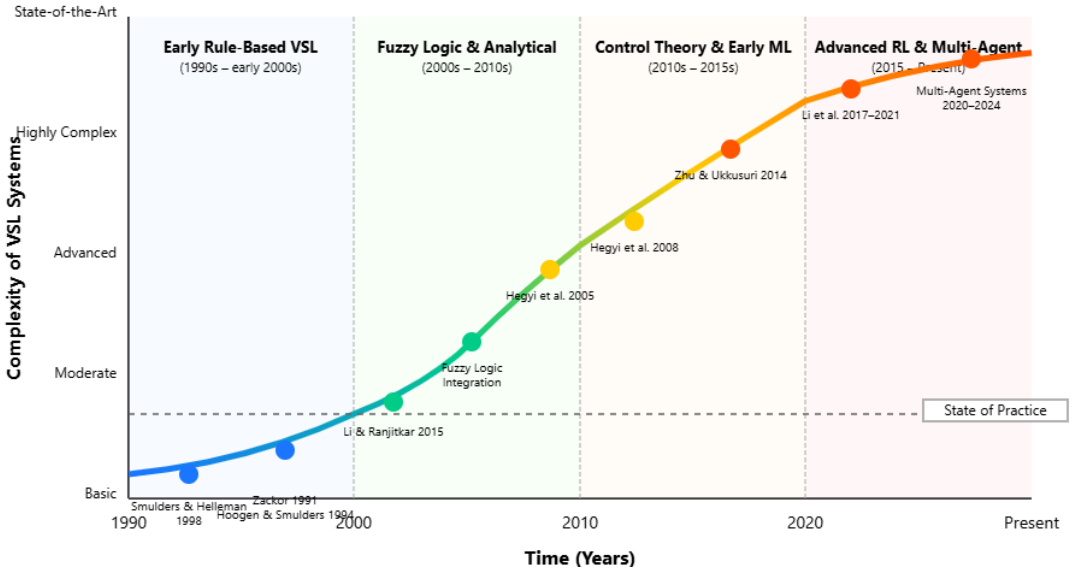


Figure 1 Evolution of VSL control

1.2 VSL Control systems – The-state-of-the-practice

Despite promising advancements in state-of-the-art in VSL control, rule-based systems continue to dominate practical implementations due to several factors. First, they offer greater transparency and explainability, which is critical for systems directly affecting driver behaviour and public safety. Second, they have substantially lower computational requirements, making them feasible for widespread deployment across extensive road networks. Third, they integrate more seamlessly with existing (legacy) traffic management infrastructure, requiring less specialized expertise to maintain and operate.

Typically, rule-based VSL systems activate or deactivate control actions (speed changes) using pre-set thresholds at loop detector locations, and they estimate queue propagation rates to determine back-of-queue positions for upstream speed reductions.

The calibration process is traditionally performed manually by traffic engineers based on historical data and expert judgment. This approach is resource-intensive, time-consuming, and cannot be performed frequently to adapt to dynamic traffic conditions or seasonal variations.

A critical aspect of effective VSL implementation is the accurate detection and localization of congestion, particularly the back-of-queue position. Conventional loop detectors used for congestion detection are typically spaced at intervals of approximately 500 meters, providing spatially sparse data that fails to capture detailed queue profiles. This limitation hampers the ability of VSL systems to protect the back of congestion effectively, potentially leading to safety risks if speed reductions are not applied at appropriate locations (Abdel-Aty et al., 2006).

1.3 Methods for Calibrating Rule-Based VSL Approaches

Various approaches have been proposed in the literature for calibrating rule-based VSL systems. Researchers have examined the optimization of threshold parameters through simulation-based studies (Hoogen & Smulders, 1994; Zackor, 1991) and has explored the integration of fuzzy logic to mitigate the rigidity of crisp thresholds (D. Li & Ranjitkar, 2015). Knoop et al., (2010) explored the calibration of fundamental diagrams for different traffic regimes, finding that accurate representation of congestion onset and dissipation is critical for effective VSL control. Their research highlighted the importance of capturing the spatial and temporal dynamics of congestion for optimal VSL parameter selection. For addressing detector data sparsity, Treiber & Helbing, (2002) proposed advanced interpolation and smoothing

techniques called Adaptive Smoothing Method (ASM) to generate more detailed traffic state estimates between detector locations.

Optimization-based approaches have also been explored for calibrating VSL parameters. Hegyi et al., (2008) formulated the calibration problem as an optimization with the objective of maximizing throughput while minimizing total travel time. However, these approaches often require substantial computational resources and may not be practical for applications in large-scale networks.

1.4 Research Gap and Objective

Despite these advances, the systematic, data-driven calibration of rule-based VSL parameters remains challenging for road agencies. Current practices often rely on trial-and-error approaches or simplified guidelines that may not capture the complex dynamics of traffic flow or adapt to changing conditions. This gap highlights the need for more robust, automated calibration methods that can enhance the performance of rule-based VSL systems while preserving their operational simplicity.

This paper focuses on a data-driven framework designed to automate the calibration of rule-based VSL parameters. The proposed framework aims to reduce human intervention by integrating modules that enhance the estimation of back-of-queue locations and dynamically detect congestion. Specifically, the Adaptive Smoothing Method with Edge Detection (ASM-ED) module performs a traffic flow theory-informed interpolation of sparse loop detector data, generating detailed queue profiles and identifying congestion propagation patterns. In parallel, the Speed-Based Congestion Detection Module, utilizing a Smoothed Difference of Squared Speeds (SDSS) approach, dynamically detects the onset and dissipation of congestion. Together, these modules enable more responsive VSL activation and deactivation thresholds, thereby improving the operational efficiency of rule-based VSL systems without sacrificing their inherent simplicity and interpretability.

The remaining of the paper is structured as follows. Section 2 presents a conceptual explanation of a hypothetical rule based VSL system with illustrative examples using sample data. Section 3 presents the overall methodology for the calibration of the rule-based VSL parameters, followed by detailed explanation of the different components in the calibration framework. Section 4 discusses in detail the implementation of the proposed calibration framework and the results. Section 5 highlights the key findings from the study.

2 Conceptual explanation of Rule Based VSL

2.1 Congestion Detection (CD) Module

The Congestion Detection module is used to detect queues on motorway sections equipped with vehicle detectors. By detecting congestion, the algori0thm allows for proactive measures such as adjusting speed limits to minimize conflicts and enhance safety.

2.1.1 Working Principle

The Congestion Detection algorithm operates at specified time intervals and follows a step-by-step process to evaluate each detector site within predefined zones. It compares site measures against defined thresholds and parameters to detect congestions. It also monitors the end of congestions and provides notifications when they have cleared. The different parameters used by the algorithm to ascertain the presence and absence of congestions are:

1. **Speed Activation Threshold**: The threshold value for average speed that, when crossed, indicates the activation of congested conditions at a Detector Site.

2. **Speed Deactivation Threshold**: The threshold value for average speed that, when crossed, indicates the end of congestion conditions at a Detector Site.
3. **Occupancy Activation Threshold**: The threshold value for average occupancy that, when crossed, indicates the activation of congested conditions at a Detector Site.
4. **Occupancy Deactivation Threshold**: The threshold value for average occupancy that, when crossed, indicates the end of congested conditions at a Detector Site.
5. **Number of Consecutive Intervals**: Number of intervals that the algorithm must detect congested conditions before reporting the occurrence of a potential congestion.

The step by step working of the CD module can be summarised as follows:

1) At the specified time interval:
    a) For each Congestion Detection Zone:
        i) Retrieve a list of Detector Sites marked for evaluation.
        ii) For each Detector Site:
            (1) Retrieve the site measures, including Average Speed and Average Occupancy.
            (2) Compare the site measures against Activation thresholds to determine if congestion is being detected.
            (3) If congestion is detected:
                (a) Record the Activation Time, increment the Activation Count, and set the Deactivation Count to zero for the site.
                (b) Determine if congestion is being detected based on the activation count.
            (4) If congestion is not detected at the site but either the Activation Count or the Deactivation Count is greater than zero:
                (a) Check if the site may be experiencing the end of congestion conditions.
                (b) Compare the site measures against Deactivation thresholds to determine if the congestion has cleared.
                (c) If the congestion has cleared, increment the Deactivation Count for the site.
                (d) Process the next Congestion Detector Site within the Detection Zone.

By iteratively evaluating the Detector Sites, the CD algorithm provides real-time information of the status of the congestions on the motorway.

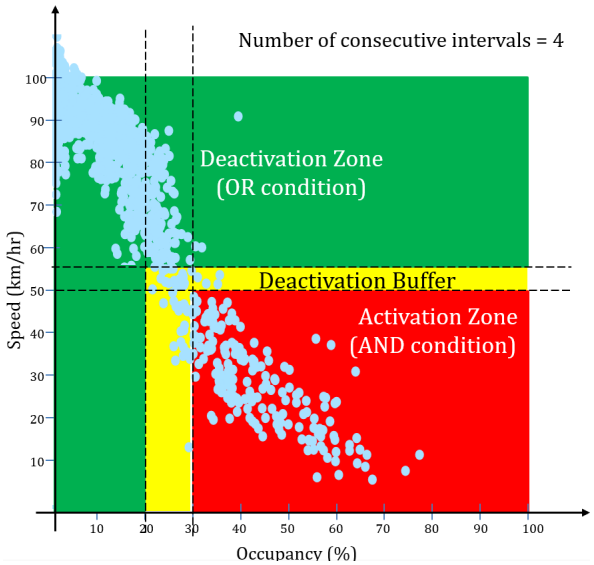
2.1.2 Illustration using sample data

Consider a detector site where the occupancy and speed thresholds are available for multiple time intervals. A sample data set is shown in *Table 1*. The DS is part of a congestion detection zone with CD module enabled.

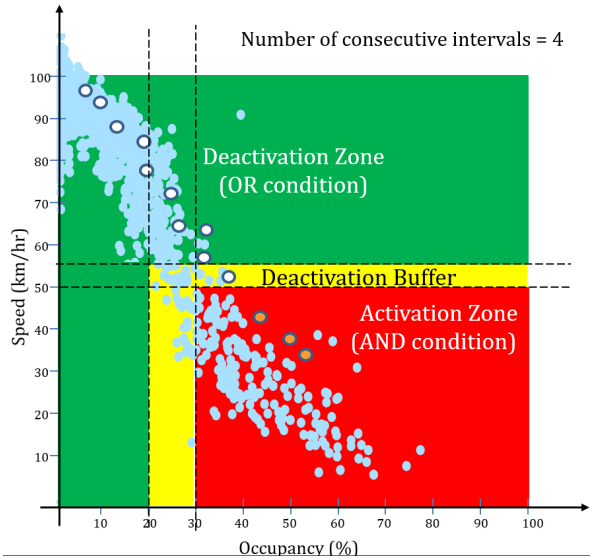
Table 1 Sample speed-occupancy data for CD algorithm demonstration

Time Interval	Occupancy (%)	Speed (km/hr)	Activation Counter	Congestion Status
1	5.6	97.3	0	Uncongested
2	8.8	94.2	0	Uncongested
3	12.6	88.6	0	Uncongested
4	18.1	85.3	0	Uncongested
5	18.5	78.5	0	Uncongested
6	24.3	72.7	0	Uncongested
7	31.5	64.5	0	Uncongested
8	25.5	65.6	0	Uncongested
9	31.1	58.0	0	Uncongested
10	36.7	53.4	0	Uncongested
11	43.3	44.1	1	Building
12	49.7	38.9	2	Building
13	52.9	35.5	3	Building
14	54.5	30.3	4	Congested
15	48.1	33.1	5	Congested
16	42.3	35.6	6	Congested
17	37.1	41.5	7	Congested
18	32.3	47.6	8	Congested
19	26.4	49.1	9	Congested
20	22.1	54.3	0	Congested
21	18.1	60.5	0	Uncongested
22	13.2	66.8	0	Uncongested
23	9.0	74.6	0	Uncongested
24	5.8	81.9	0	Uncongested
25	4.6	89.6	0	Uncongested

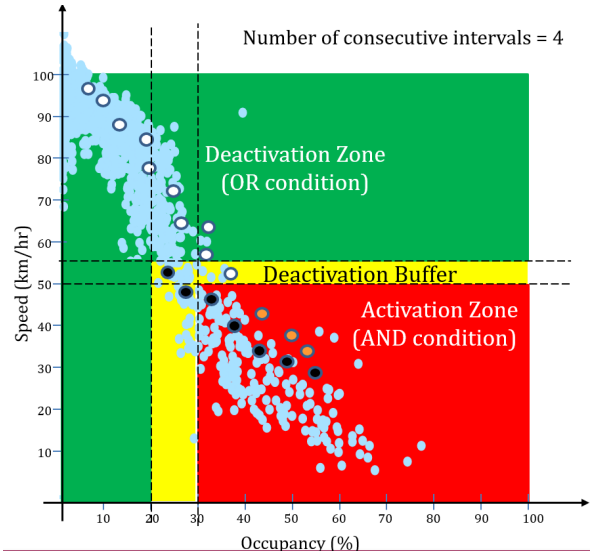
Let the activation and deactivation speed threshold be 50 km/hr and 55 km/hr respectively. Also, let the activation and deactivation occupancy thresholds be 30% and 20% respectively. At time instance 11, the occupancy and speed activation thresholds are met, the Congestion status is set to 'Building'. At time instance 14, the consecutive time interval counter value is met (4 consecutive intervals) and the Congestion status is updated to 'Congested'. At time intervals 19 and 20, though the occupancy value is less than the activation threshold (26.4% < 30% and 22.1% <30%) the Congestion status is still 'Congested' as the deactivation thresholds are not met. This region is depicted in *Figure 2* as buffer zone, in addition to the activation and deactivation zones. The three zones (activation, deactivation, and buffer) are arrived at based on the activation and deactivation thresholds. The black dots in the figure indicate 'Congested' state, white dots indicate 'Uncongested' Congestion status, and the orange dots indicate 'Building' state.



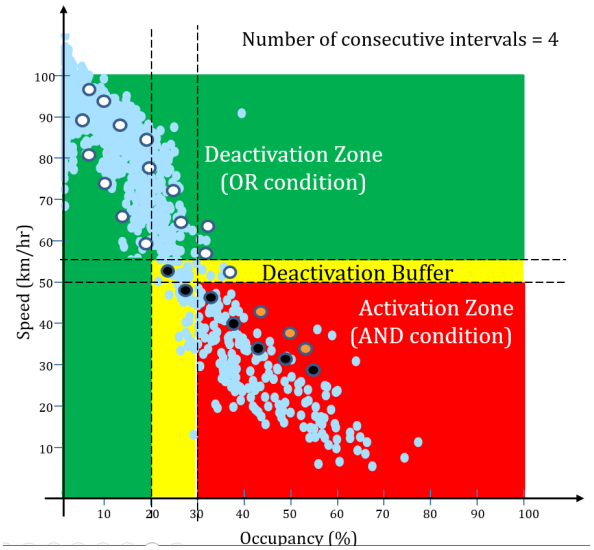
(a) Deactivation, activation and buffer zones with underlying 1-day traffic state data



(b) Sample traffic states evolving from 'Uncongested' (white dots) to 'Building' (Orange dots)



(c) Congestion status remaining 'Congested' in buffer zone



(d) Congestion status transitioning to 'Uncongested' in the deactivation zone

Figure 2 Visualisation of Congestion status transitions

At time interval 21, both deactivation speed and deactivation occupancy thresholds are met, and the Congestion status is updated to 'Uncongested'. It is to be noted that even if one of the deactivation thresholds is met, the Congestion status will be considered as 'Uncongested'.

2.2 Congestion Protection (CP) Module

The Congestion Protection (CP) module is designed to address the management of Congestions by recommending speed changes to prevent secondary rear-end collisions. It works in conjunction with the Congestion Detection (CD) algorithm, which notifies the CP algorithm of the occurrence of a Congestion at a specific time and location.

2.2.1 Working Principle

The CP algorithm utilizes the provided Congestion measures from CD algorithm to calculate conditions at the rear end of the congestion, including the speed and distance from a downstream Variable Speed Limit (VSL) site to the back of the Congestion. Based on these measures, the algorithm recommends speed changes to ensure that approaching vehicles encounter a speed limit of 60 km/hr or less before reaching the Congestion.

To anticipate the location of Congestion within the road network, the CP algorithm monitors the flow rates of traffic using vehicle detector infrastructure. It determines the speed at which an approaching vehicle will encounter a Congestion by observing their respective speeds and predicting the location where the Congestion and incoming vehicles will meet.

The steps involved in the CP algorithm to predict the possibility of an incoming vehicle encountering a Congestion before passing a speed threshold (say 60km/hr) VSL gantry are as follows:

1. Determine the speed of the Congestion propagation:
    1. Use one-minute flow data to predict the change in the Congestion length between two consecutive detectors after the initial detection at the downstream detector.
    2. Observe the time taken for the Congestion detection algorithms to identify queuing characteristics at an upstream detector after the initial detection at a downstream detector.
2. Determine the speed of incoming vehicles.
3. Calculate the speed of Congestion encounter by comparing the speed of incoming vehicles with the speed of Congestion propagation.
4. Use the speed of encounter to find the time and location of the encounter, thus determining the encounter distance.
5. Evaluate the need to change the speed on the upstream gantry sign based on the calculated encounter distance. If the encounter distance indicates that an incoming vehicle would encounter the Congestion before passing another 60 km/hr VSL gantry, a speed change recommendation is made to accommodate the stopped or slow-moving traffic in the Congestion.

2.2.2 Illustration using sample data

Consider a hypothetical study corridor shown in *Figure 3*.

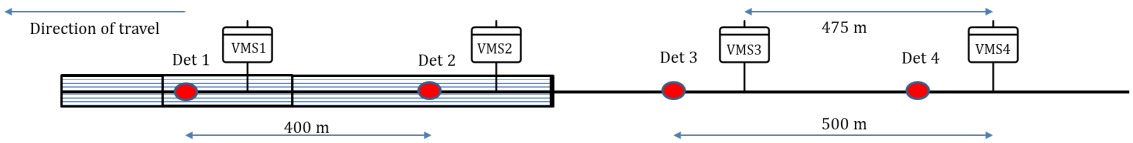


Figure 3 Illustration of CP-study corridor, the shaded represents the extent of the Congestion

The study corridor has four detector sites and corresponding VSL sites associated with each DS. The spacing between two successive detector site is 400 m and the spacing between two consecutive VSL site is 475 m. Let the initial position (stage 1) of the Congestion be as shown in *Figure 3*. The Congestion statuses for each of these detectors are known (reported by CD algorithm). Knowing the Congestion status and the distances between the DS, the Congestion length can be estimated. The Congestion lengths are shown in *Table 2*.

Table 2 Congestion status and Congestion length

Time Interval (min)	Detector 1				Detector 2				Detector 3				Congestion Length (m)
	Occupancy (%)	Speed (km/hr)	Flow (veh/hr/lane)	Congestion Status	Occupancy (%)	Speed (km/hr)	Flow (veh/hr/lane)	Congestion Status	Occupancy (%)	Speed (km/hr)	Flow (veh/hr/lane)	Congestion Status	
1	43.3	44.1	2387	Building	25.5	65.6	2091	Uncongested	18.5	78.5	1815	Uncongested	0
2	49.7	38.9	2417	Building	31.1	58	2255	Uncongested	24.3	72.7	2208	Uncongested	0
3	52.9	35.5	2347	Building	36.7	53.4	2450	Uncongested	31.5	64.5	2540	Uncongested	0
4	54.5	30.3	2064	Congested	43.3	44.1	2387	Building	25.5	65.6	2091	Uncongested	400
5	48.1	33.1	1990	Congested	49.7	38.9	2417	Building	31.1	58	2255	Uncongested	400
6	42.3	35.6	1882	Congested	52.9	35.5	2347	Building	36.7	53.4	2450	Uncongested	400
7	37.1	41.5	1925	Congested	54.5	30.3	2064	Congested	18.5	78.5	1815	Uncongested	800
8	32.3	47.6	1922	Congested	48.1	33.1	1990	Congested	24.3	72.7	2208	Uncongested	800
9	26.4	49.1	1620	Congested	42.3	35.6	1882	Congested	31.5	64.5	2540	Uncongested	800
10	22.1	54.3	1500	Congested	37.1	41.5	1925	Congested	25.5	65.6	2091	Uncongested	800
11	18.1	60.5	1369	Uncongested	32.3	47.6	1922	Congested	31.1	58	2255	Uncongested	400
12	13.2	66.8	1102	Uncongested	26.4	49.1	1620	Congested	36.7	53.4	2450	Uncongested	400
13	9	74.6	839	Uncongested	22.1	54.3	1500	Congested	31.5	64.5	2540	Uncongested	400
14	5.8	81.9	594	Uncongested	18.1	60.5	1369	Uncongested	25.5	65.6	2091	Uncongested	0
15	4.6	89.6	515	Uncongested	13.2	66.8	1102	Uncongested	24.3	72.7	2208	Uncongested	0

Based on the Congestion lengths at each time interval, the Congestion growth (Congestion propagation) speeds can be obtained. The Congestion propagation speed is adopted by the CP algorithm by following the logic depicted in *Figure 4*. Congestion propagation speed is adopted as one of the three values:

- Initial (From CP initial parameter set)
- Estimated (Estimate from CD data)
- Observed (Observed using Congestion length data from CD algorithm)

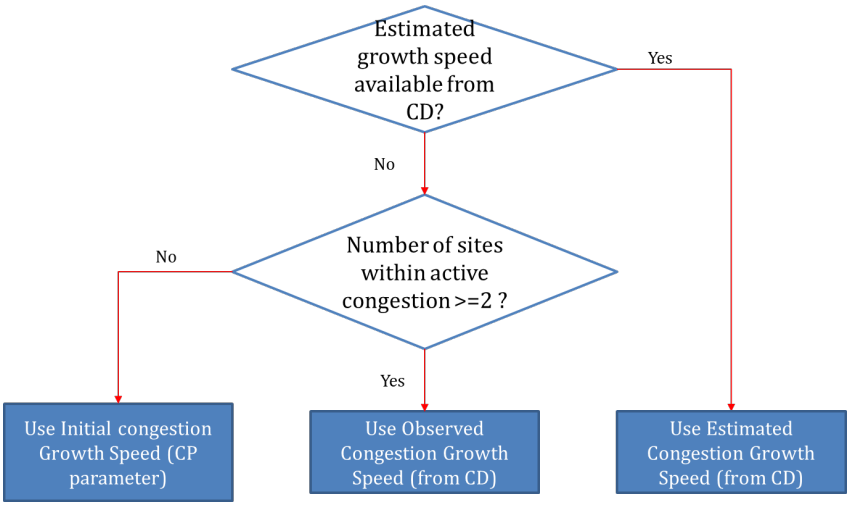


Figure 4 Congestion propagation speed selection logic.

2.2.2.1 Estimated Congestion propagation speed using input-output method

1. Use one minute flow rate μ (veh/hr) and λ (veh/hr) at detector site n - 1 (downstream) and n (upstream), respectively.
2. Determine net vehicle accumulation (N_{net}) between upstream detector site n and downstream detector site n -1 for the last p intervals using trapezoidal rule as shown in Figure 5.

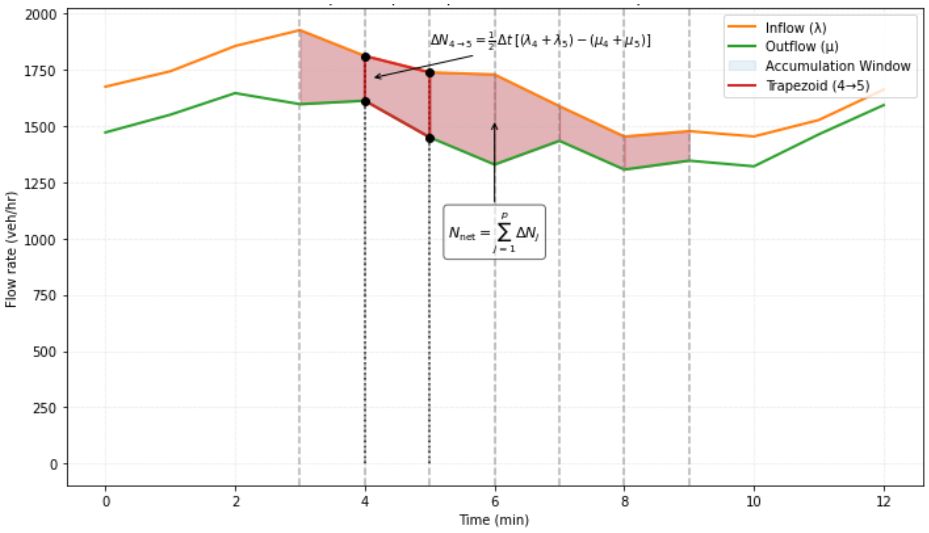


Figure 5 Illustrative synthetic example of input-output based net vehicle change estimation with p=6 successive intervals

3. To convert the net change in terms of number of vehicles to net change in terms of distance (average space occupied by vehicle), X_V (m), the average time headway (T_H) and average vehicle length (L_v) are used as shown in Eq. 1.

$$X_V = \frac{ST_H}{3.6} + L_V \quad \textit{Eq. 1}$$

4. Determine the number of lanes N_L (lanes). Then, convert the net number of vehicle changes to length (m) (Eq. 2):

$$L_Q = \frac{N_{net} X_V}{N_L} \quad \textit{Eq. 2}$$

5. For successive time intervals, estimate the change in Congestion length, ΔL_Q (m/min) (Eq. 3).

$$\Delta L_Q = L_{Q_i} - L_{Q_{i-1}} \quad \textit{Eq. 3}$$

6. Average the ΔL_C for all the time intervals during which the tail of the Congestion is in the tail area. This will be the estimated Congestion propagation speed (S_{Flow}) (Eq. 4).

$$S_{Flow} = \frac{1}{k}\sum_{j=1}^{k} \Delta L_{Q_j} \quad \textit{Eq. 4}$$

2.2.2.2 *Observed Congestion propagation speed*

1) Note down t_{n-1} and t_n as the times when detector *n - 1* and *n* first experienced Congestion.
2) Calculate the time taken for Congestion to reach from detector n -1 to detector n as $T_{n,n-1} = t_n - t_{n-1}$.
3) Knowing the distance between the two detectors *n-1* and *n* ($D_{n,n-1}$), the observed Congestion propagation speed can be calculated as in Eq. 5.

$$S_{obs.} = \frac{D_{n,n-1}}{T_{n,n-1}} \qquad \text{Eq. 5}$$

2.2.2.3 *Recommending speed drop*

Figure 6 shows the scenario where a Congestion is growing along the study corridor (dotted lines) and a vehicle is approaching the Congestion (stage 2). The CP algorithm must decide if or not to recommend a speed drop.

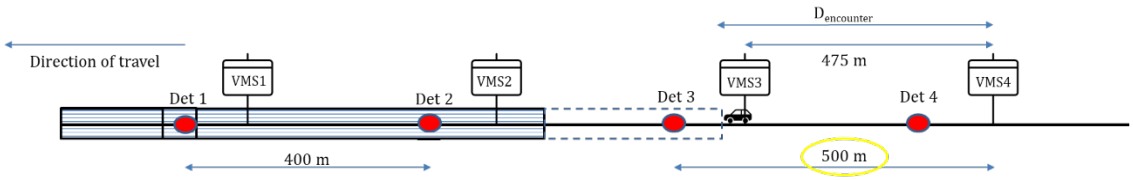


Figure 6 Illustration of CP-study corridor, the dotted region representing Congestion growth

To achieve this, the CP algorithm uses the following steps:

1) Determine the incoming vehicle speed (m/min). This is taken as the speed limit on the VSL site immediately downstream of the arriving vehicle.
2) The speed with which the encounter between the approaching vehicle and the back of the Congestion will occur (encounter speed) is calculated as $S_{enc} = S_{tail} + S_{incoming}$.
3) Using the encounter speed and the distance between the n^{th} detector site and the $(m+1)^{th}$ VSL site (500 m in the example study site), the time to encounter is determined as $T_{enc} = \frac{D_{n,m+1}}{S_{enc}}$.
4) The encounter distance is calculated as $D_{encounter} = S_{incoming} * T_{enc}$.
5) If the encounter distance is less than or equal to the distance between the m^{th} and the $(m+1)^{th}$ VSL sites $(D_{m,m+1})$, the speed at VSL site m is dropped to 60 km/hr.

By predicting the possibility of an incoming vehicle encountering a Congestion, the CP algorithm enables timely speed limit adjustments on upstream gantry signs, enhancing safety and reducing the risk of rear-end collisions.

With the understanding of the working principles of a basic rule based VSL system, the proposed methodology for the data-driven calibration of the rule-based VSL is explained next.

3 Methodology

The reactive VSL framework is designed to detect and respond to traffic Congestions dynamically. The core components include a Congestion Detection (CD) and a Congestion Propagation (CD-CP) algorithm, a Speed-Based Congestion Detection Module, and an Adaptive Smoothing Method (ASM) with Edge Detection module. These components work together to identify Congestions, estimate Congestion growth rates, and refine rule-based thresholds.

3.1 Overall Framework Description

The overall methodology begins by monitoring speed data from loop detectors to assess traffic conditions. The Difference of Squared Speeds (SDSS) is computed and smoothed to detect significant changes in speed, which indicate the onset or dissipation of traffic Congestions.

These smoothed values are used to determine the Congestion activation and deactivation thresholds for VSL implementation.

Once the congestion detection thresholds are identified, they are applied in the CD_CP algorithm, which serves as a rule-based VSL example in this study. The congestion detection component activates the VSL when speeds drop below the threshold, while the congestion propagation component estimates the back of the Congestion, using parameters like average vehicle length and average headway.

However, these parameters often require calibration to accurately reflect real-world traffic conditions. For this, we employ the ASM-Edge Detection module to derive a realistic Congestion profile, which is then compared against the Congestion growth rate estimated by the CP algorithm. The comparison allows us to refine and auto-calibrate key parameters through an optimization process. The entire system iterates until the Congestion growth rates from both methods converge, ensuring the optimized values for use in real-time reactive VSL.

The flowchart provided illustrates the sequence of processes involved, from Congestion detection to Congestion protection and optimization, ensuring the VSL reacts appropriately to fluctuating traffic conditions.

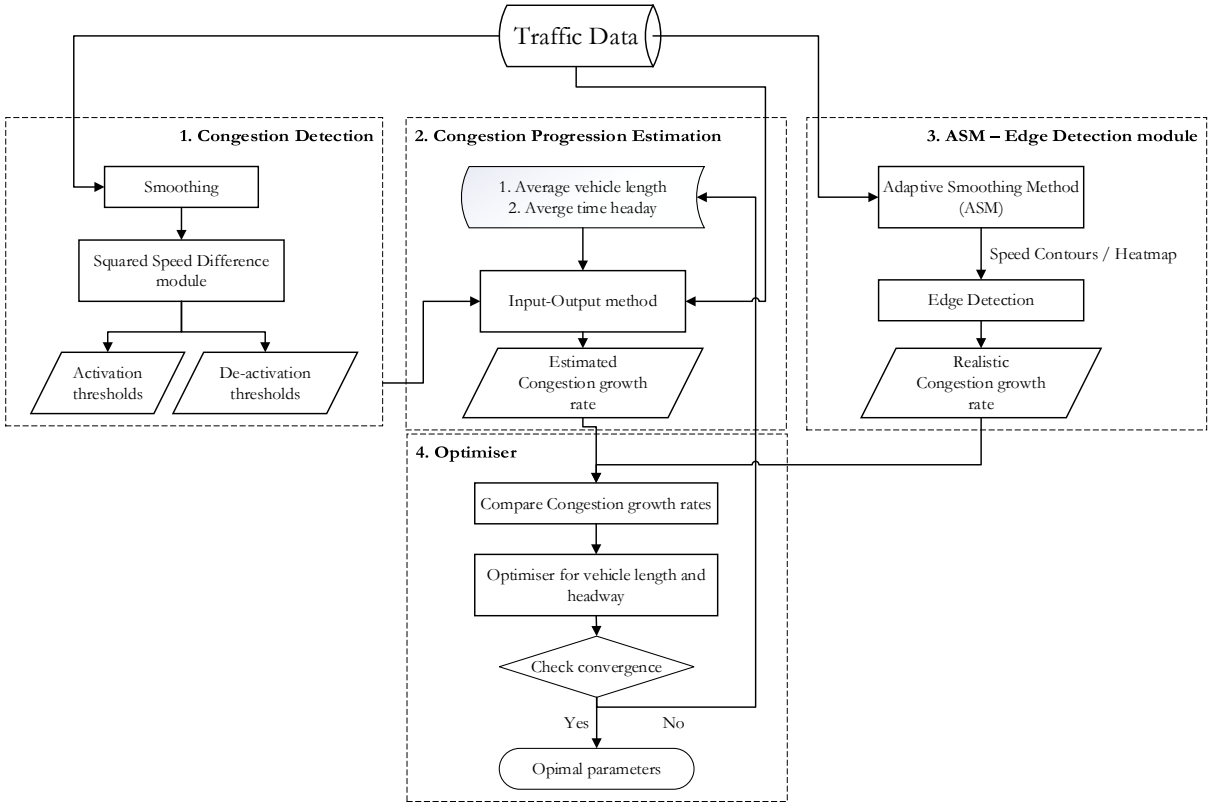


Figure 7 Methodology for data-drive auto-calibration of rule-based VSL

3.2 Speed-Based Congestion Detection Module

The Speed-Based Congestion Detection Module leverages real-time speed data to dynamically identify the onset and dissipation of congestion using the Smoothed Difference of Squared Speeds (SDSS) approach. The first step involves calculating the squared speed difference, represented as in *Eq.* 6:

$$\Delta_2 \mathrm{V} = \mathrm{V}^2_{(\mathrm{x},t+\Delta t)} - \mathrm{V}^2_{(\mathrm{x},t)} \qquad \textit{Eq. 6}$$

This method enhances the detection of sudden speed drops by considering both the magnitude and severity of the drop. Rather than simply using the difference between speed values, the squared speed difference accounts for the impact of the initial speed from which the drop occurs. For instance, a 10 km/h speed drop from 90 km/h is not the same as a 10 km/h drop from 60 km/h. The squared difference is expressed as in *Eq.* 7:

$$\Delta_2 \mathrm{V} = (\mathrm{V}_{(\mathrm{x},t+\Delta t)} - \mathrm{V}_{(\mathrm{x},t)})(\mathrm{V}_{(\mathrm{x},t+\Delta t)} + \mathrm{V}_{(\mathrm{x},t)}) \qquad \textit{Eq. 7}$$

This formulation not only quantifies the speed drop but also reflects the context of the drop, where higher initial speeds contribute to more critical conditions when a drop occurs. Once the squared speed difference is calculated, noise and fluctuations in the data are reduced using exponential smoothing (Brown, 1956), with a smoothing constant of 0.2.

The SDSS is then analysed to identify the Congestion activation and deactivation conditions. The minimum value of $\Delta_2 V$ represents the Congestion or congestion onset condition, while the maximum value of $\Delta_2 V$, identified after the onset, indicates the Congestion deactivation condition.

For each day, the corresponding traffic state variables—speed and occupancy—at the onset and deactivation points are extracted from historical traffic data over an extended period (e.g., 6 months). These onset and deactivation points are then plotted on speed-occupancy fundamental diagrams.

To robustly determine the thresholds, Kernel Density Estimation (KDE) (Parzen, 1962) is applied to identify the densest region of these points on both the activation and deactivation diagrams. The speed and occupancy values corresponding to the densest points are chosen as the activation and deactivation thresholds for the Congestion onset and termination. These thresholds provide a data-driven, systematic approach to triggering and deactivating variable speed limits based on actual congestion patterns.

3.3 Adaptive Smoothing Method with Edge Detection Module

One of the primary challenges with conventional loop detector data is their spatial sparsity—typically, detectors are spaced approximately every 500 meters, providing Congestion condition information only at those specific locations. This restricts the ability to accurately estimate detailed Congestion profiles, particularly when capturing the rate at which Congestions grow or dissipate.

To compensate for this limitation, traditional approaches often rely on the input-output method, which estimates the Congestion growth rate by calculating the net change in vehicles within the section between two detectors (refer to Figure 5). This net change is then converted into a Congestion length using assumptions such as average vehicle length and average headway. The resulting change in Congestion length is used to estimate the Congestion tail propagation speed. However, the accuracy of this method is dependent on the calibration of these parameters. Moreover, as traffic characteristics evolve, these parameters need to be dynamically adjusted for reliability.

This is where the Adaptive Smoothing Method (ASM) with Edge Detection is useful. The ASM module addresses the spatial sparsity of loop detectors by using a traffic theory-informed interpolation of the traffic state variables (e.g., speed) between detectors, producing a continuous traffic state estimation across space and time. This interpolation creates a detailed speed contour, revealing the variation of speed along the roadway.

From this speed contour, sudden speed drops, indicative of Congestion formation, can be visually identified. To systematically capture these Congestion profiles, an edge detection

algorithm is employed, which automates the process of identifying the location and extent of the Congestion from the speed contour plot. The result is a more realistic Congestion profile that reflects actual traffic conditions with greater spatial resolution.

These realistic Congestion profiles serve as benchmarks for calibrating the input-output method parameters—specifically, average vehicle length and average headway. By aligning the estimated Congestion growth rate with these observed profiles, the reliability of the input-output method is significantly improved. Once these parameters are optimally calibrated, the reactive VSL system can better respond to dynamic traffic patterns, ensuring timely speed reductions based on accurate Congestion location and propagation rates.

4 Implementation and Results

4.1 Study Site

The study site used for implementing the data-driven VSL calibration framework is the Centenary Highway inbound corridor in Brisbane, between Bullockhead Street to Centenary bridge. Figure 8 illustrate the Map view of the study site. The study stretch is about 7 km long with 17 Detector Sites (DS) and 9 VSL gantries.

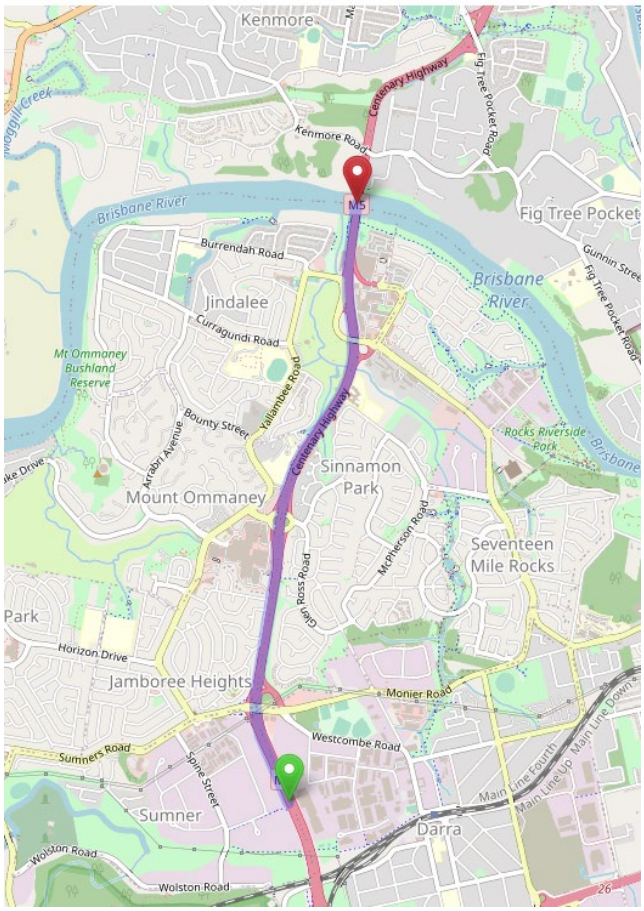


Figure 8 Study corridor (Google map view)

4.2 Data Used

The data used for the implementation of the optimisation framework is the 20 second aggregated individual vehicle detector data (per lane) from all the 17 detector sites in the study corridor. The 20 second data was obtained f for the year 2023. A representative sample of the 20 second dataset is shown in Table 4.

Table 3 Sample 20 second detector data

START_DATETIME_AEST	END_DATETIME_AEST	VOLUME	OCCUPANCY	FLOW	TIME_MEAN_SPEED	SPACE_MEAN_SPEED	SPEED_OBSERVATIONS	VEHICLE_DETECTOR_NAME
2023-01-27T00:46:20.113Z	2023-01-27T00:46:40.114Z	0	0				0	316-13
2023-01-27T06:31:40.055Z	2023-01-27T06:32:00.056Z	9	8.7		94.5555556	94.42714836	9	317-01
2023-01-27T08:41:40.061Z	2023-01-27T08:42:00.09Z	13	18.4		76.5384615	75.89137734	13	315-05
2023-01-27T10:35:40.088Z	2023-01-27T10:36:00.089Z	12	16.4		85.5833333	85.27538022	12	MCH-320VD07
2023-01-27T10:32:40.048Z	2023-01-27T10:33:00.049Z	7	7		89	88	7	318-03
2023-01-27T10:15:40.122Z	2023-01-27T10:16:00.112Z	4	7.7		93.5	93.40762639	4	319-13
2023-01-27T04:06:40.059Z	2023-01-27T04:07:00.07Z	0	0				0	315-05
2023-01-26T23:50:40.056Z	2023-01-26T23:51:00.067Z	0	0				0	315-05
2023-01-13T08:03:20.107Z	2023-01-13T08:03:40.108Z	5	6		104	103	5	MCH-323VD11
2023-01-14T12:38:40.086Z	2023-01-14T12:39:00.097Z	11	15		83	83	11	315-05

The following data processing steps are applied on the raw data to make it suitable to be used for implementing the auto-calibration framework. A

- Convert date columns to datetime: Ensures the date and time columns are in the correct format for analysis.
- Sort the data: Sorts the data by START_DATETIME_AEST in ascending order.
- Filter the data:
    - Select data within the specified date range.
    - Include only working weekdays (Monday to Friday).
- Calculate flow: Creates a new column FLOW by multiplying the VOLUME column by a factor to convert from vehicles per 20 seconds to vehicles per hour.
- Replace missing speed values and extreme speed values with the following conditions:
    - If both speed and occupancy are 0, replace speed with NULL value.
    - If flow is less than or equal to 180 veh/hour (1 veh/20second) and speed is greater than 110, replace speed with 100. Replace extreme values of speed with the free flow speed.
- Apply exponential smoothing to speed, flow and occupancy columns with a smoothing constant of 0.3 (in line with current practice).
- Find worst lane: For each detector site, find the worst lane as the lane (vehicle detector) with the lowest smoothed speed (worst performing) at that time.

This provides us with the 20 second dataset in the required format that can be given as input to the congestion detection and protection (reactive VSL) algorithm.

4.3 Speed Profile based Activation and Deactivation Threshold Calibration

The methodology described in Section 3, is implemented on dataset described in Section 4.2. As a first step, the $\Delta_2 V$ profile for each detector for a given time period is generated using *Eq.* 6. The $\Delta_2 V$ profile for detector site 315-NB for 26th September 2023 is presented in Figure 9.

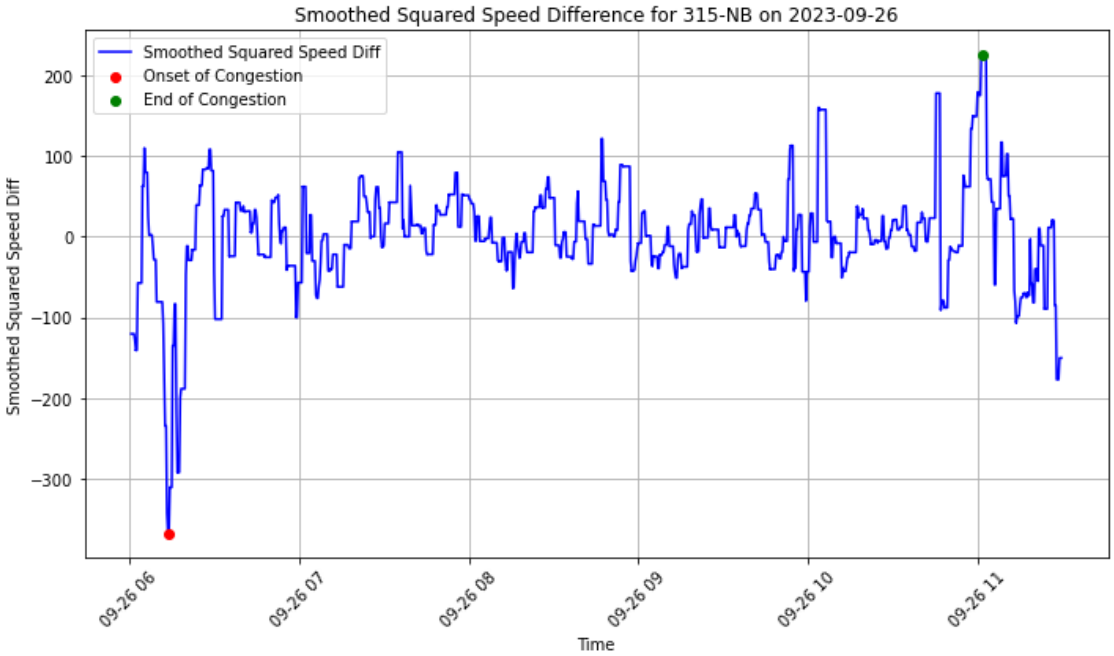


Figure 9 Smoothed $\Delta_2 V$ profile for DS 315-NB on 26th September 2023.

The time corresponding to the minimum value of the $\Delta_2 V$ time series is identified as the onset point. The occupancy and speed corresponding to this time is noted. This process is repeated for all the working days during 2023. The candidate onset points for each day are plotted on a Speed-Occupancy fundamental diagram and Kernel Density Estimation process is used to find the densest point as shown in Figure 10. The densest point corresponds to an activation speed of 51 km/hr and an activation occupancy of 25%.

Similarly, the time corresponding to the maximum value of the $\Delta_2 V$ time series is identified as the deactivation point. The occupancy and speed corresponding to this time is noted. This process is repeated for all the working days during 2023. The candidate deactivation points for each day are plotted on a Speed-Occupancy fundamental diagram and Kernel Density Estimation process is used to find the densest point as shown in Figure 11. It can be seen that the densest point corresponds to a deactivation speed of 57 km/hr and a deactivation occupancy of 15%.

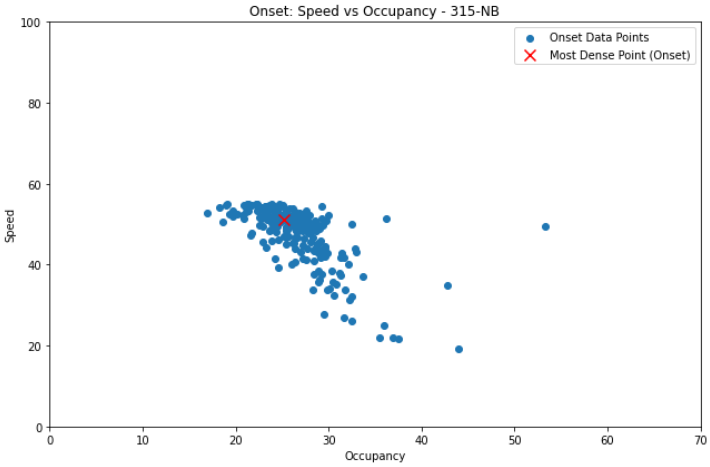


Figure 10 Speed-Occupancy fundamental diagram with candidate onset points for DS 315NB

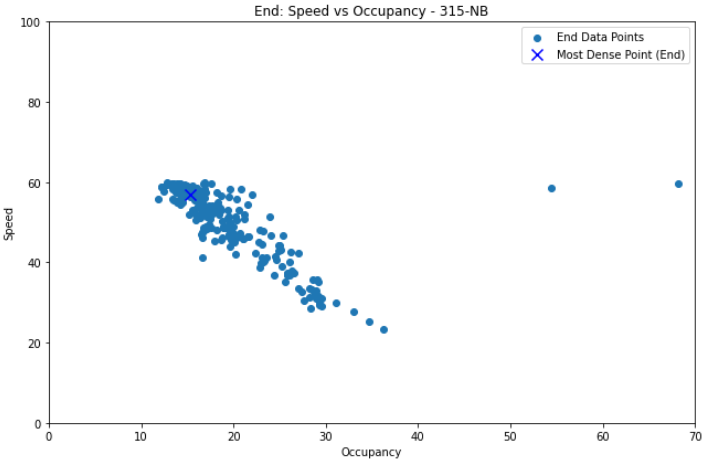


Figure 11 Speed-Occupancy fundamental diagram with candidate deactivation points for DS 315NB

This process is repeated for all the detector sites in the study corridor and the activation and deactivation thresholds are systematically identified. Table 5 presents a summary of the activation and deactivation thresholds proposed for each detector site.

Table 4 Proposed activation and deactivation thresholds for all detector sites

Detector	Activation		Deactivation	
	Speed	Occupancy	Speed	Occupancy
315-NB	51	25	57	15
315-NB1	46	22	56	16
316-NB	51	25	57	16
316-NB1	53	24	57	19
317-NB	51	22	57	15
318-NB	48	23	58	15
319-NB	46	27	56	15
319-NB1	47	28	57	17
MCH320-NB	52	25	57	15
MCH322-NB	50	27	57	16
MCH323-NB	51	27	57	19
MCH325-NB	47	21	57	14

This framework for data driven-calibration of the key queue detection parameters can be extended for other queue detection zone by providing historic vehicle detector data for all detectors in that zone as the input.

Similarly, the methodology can be applied during different weather events to study the impact of weather on VSL algorithm.

4.4 Optimiser Module for the Calibration of Queue Growth Rate Parameters

4.4.1 Formulation of the Optimisation Problem

As discussed in Section 3.1.1, the key idea is to consider the realistic queue profiles obtained from the ASM-ED module as the ground truth queue and find the optimal congestion growth

rate parameters of the congestion protection algorithm (T_H and L_V) that will provide congestion growth rates that match the realistic congestion profiles. To achieve this, the objective function shown in *Eq.* 8 is used.

$$z = minimise \left[\frac{1}{T} \sum_{t=0}^{T} \left| q_{ASM-ED,t} - q_{CD-CP,t} \right| \right] \quad \text{Eq. 8}$$

Where:

- $q_{ASM-ED,t}$ is the location of the realistic queue tail at time t obtained from the ASM-ED module
- $q_{CD-CP,t}$ is the location of the queue tail estimated by the CD-CP algorithm at time t. This value is dependent on the queue growth rate and hence is in turn dependent on L_V and T_H.

An optimisation strategy focusing on optimising the parameters for each individual queue is discussed next.

4.4.2 Individual Queue Based Optimisation

For the implementation of the optimiser module, a set of 26 candidate queues are selected randomly. These 26 queues span across both AM and PM peak as well as weekdays and weekends. The details of these queues are presented in Table 6.

Table 5 Candidate queues for implementing the optimiser module

Index	Initial Detector Site	Duration	Initial Time
14	315-NB	4:08:20	2/09/2024 6:03
120	315-NB1	2:03:40	7/09/2024 14:26
116	315-NB1	3:41:40	7/09/2024 9:32
100	315-NB1	4:07:20	6/09/2024 13:47
87	315-NB1	3:13:00	6/09/2024 6:09
78	315-NB1	4:03:00	5/09/2024 14:46
74	315-NB1	1:59:40	5/09/2024 10:23
70	315-NB1	3:22:20	5/09/2024 6:09
62	315-NB1	3:48:00	4/09/2024 14:14
56	315-NB1	3:44:40	4/09/2024 6:10
50	315-NB1	3:17:00	3/09/2024 14:40
32	315-NB1	4:30:00	3/09/2024 6:07
5	315-NB1	1:43:20	1/09/2024 11:13
1015	315-NB1	4:44:20	26/09/2023 6:14
1026	315-NB1	3:02:00	26/09/2023 14:38
1014	315-NB1	4:05:20	27/09/2023 6:01
1020	315-NB1	2:15:40	27/09/2023 14:23
1013	315-NB1	3:32:40	30/09/2024 9:39
1087	315-NB1	4:01:40	29/09/2024 13:47

1078	315-NB1	3:24:20	28/09/2024 6:09
1062	315-NB1	3:56:00	28/09/2024 14:46
1056	315-NB1	1:52:20	25/09/2024 10:23
1032	315-NB1	3:18:00	25/09/2024 6:09
1050	315-NB1	3:29:40	25/09/2024 6:10
1037	315-NB1	4:24:20	29/09/2024 6:07
1039	315-NB1	1:49:40	1/10/2024 11:13

For each of these queues, the ASM-ED module is implemented to get realistic back of queues. The ASM parameters used for the current study are presented in Table 7.

Table 6 ASM Parameters used

ASM Parameter	**Value Used**	**Justification**
Characteristic Speed of traffic in free flow conditions (c_{free})	80 km/hr	Free flow speed of the corridor
Characteristic Speed of traffic in congested conditions (C_{free})	18 km/hr	Characteristic wave speed parameter from fundamental diagram of study corridor
Temporal Smoothing parameter (τ)	20 seconds	Data aggregation interval
Spatial smoothing parameter (σ)	50 meters	Spatial interpolation interval
Threshold speed between free flow and congestion (V_{thr})	50 km/hr	Critical speed value from fundamental diagram of study corridor

The ASM output for one of the queues (Queue 78) is presented in Figure 12.

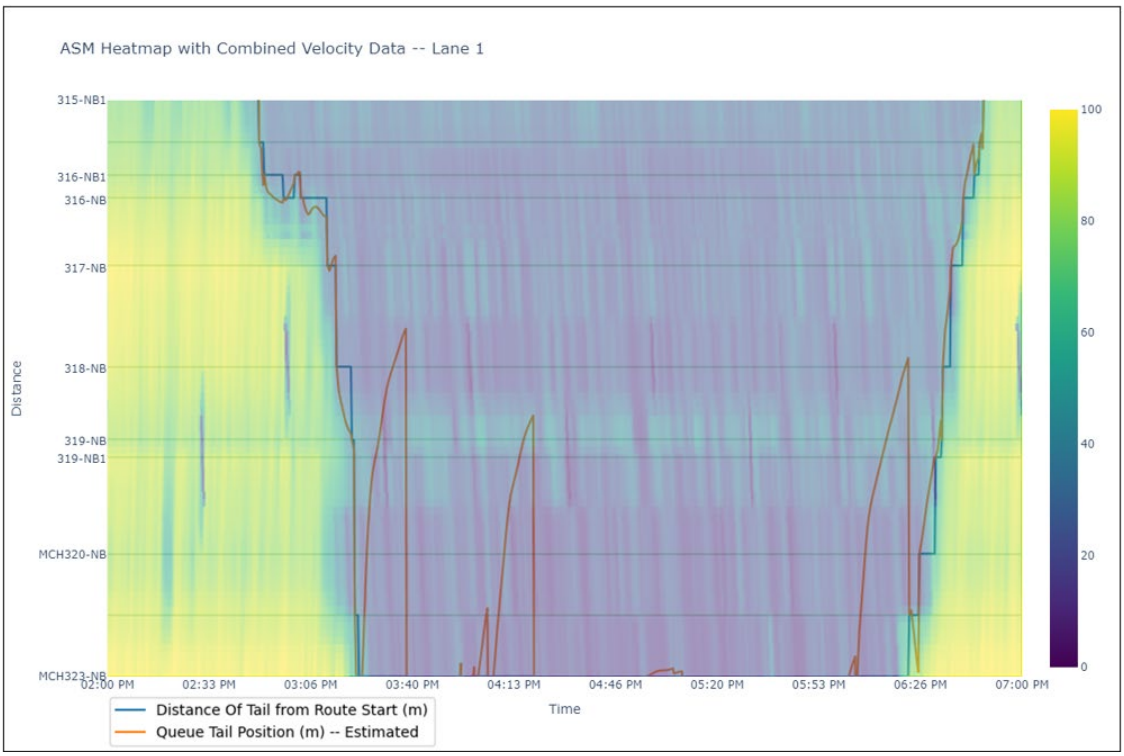


Figure 12 ASM output for queue 78 along with the Queue tail location from the CD-CP algorithm (with default parameters)

As can be seen from Figure 12, the default queue growth rate parameters from the CD-CP algorithm needs to be optimised to match the real queues from the ASM-ED output.

The optimisation formulation presented in Section 4.4.1 is implemented for each of the 26 candidate queues (congestion profiles) with the constraints in Eq. 9 and Eq. 10.

$$1 \leq T_H \leq 2 \quad \textit{Eq. 9}$$

$$3 \leq L_V \leq 4 \quad \textit{Eq. 10}$$

The upper and lower limit constraints are chosen based on engineering judgement and based. A brute force optimiser is run for each of the queues where both T_H and L_V are incremented in increments of 0.1. The optimal parameters for each of these queues are shown in Table 4.8

Table 4.7 Optimal queue growth rate parameters based on individual queue-based optimisation

Index	Initial Time	T_H	L_avg
14	2/09/2024 6:03	1.4	4.1
120	7/09/2024 14:26	1.3	3.9
116	7/09/2024 9:32	1.6	4.6
100	6/09/2024 13:47	1.5	4
87	6/09/2024 6:09	1.1	3.5
78	5/09/2024 14:46	1.4	4.1
74	5/09/2024 10:23	1.5	3.8
70	5/09/2024 6:09	1.1	3.7
62	4/09/2024 14:14	1.3	3.4
56	4/09/2024 6:10	1.5	3.8
50	3/09/2024 14:40	1.6	3.5
32	3/09/2024 6:07	1.6	3.7
5	1/09/2024 11:13	1.4	3.9
1015	26/09/2023 6:14	1.6	4.1
1026	26/09/2023 14:38	1.5	3.3
1014	27/09/2023 6:01	1.1	3.6
1020	27/09/2023 14:23	1.1	3.7
1013	30/09/2024 9:39	1.7	4.5
1087	29/09/2024 13:47	1.4	3.3
1078	28/09/2024 6:09	1.5	3.5
1062	28/09/2024 14:46	1.3	4.3
1056	25/09/2024 10:23	1.6	4.6
1032	25/09/2024 6:09	1.3	4.3
1050	25/09/2024 6:10	1.6	3.7
1037	29/09/2024 6:07	1	3.6
1039	1/10/2024 11:13	1.2	4.3

The distributions of the optimal queue growth rate parameters are shown in Figure 13.

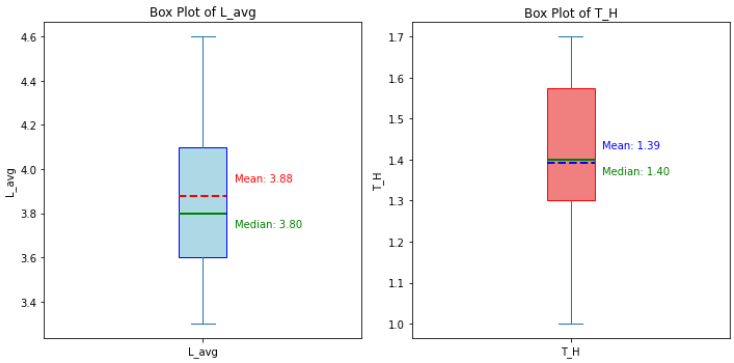


Figure 13 Distribution of optimal queue growth rate parameters

From Figure 13, the optimal vehicle length parameter (L_V) is centred around 3.8 to 3.9 m and the optimal average time headway parameter is centred around 1.4 seconds.

To get a single pair of value for the optimal parameters, an L_V versus T_H scatter plot is generated, and Kernel Density Estimation (KDE) process is applied to find the most dense point as shown in Figure 14.

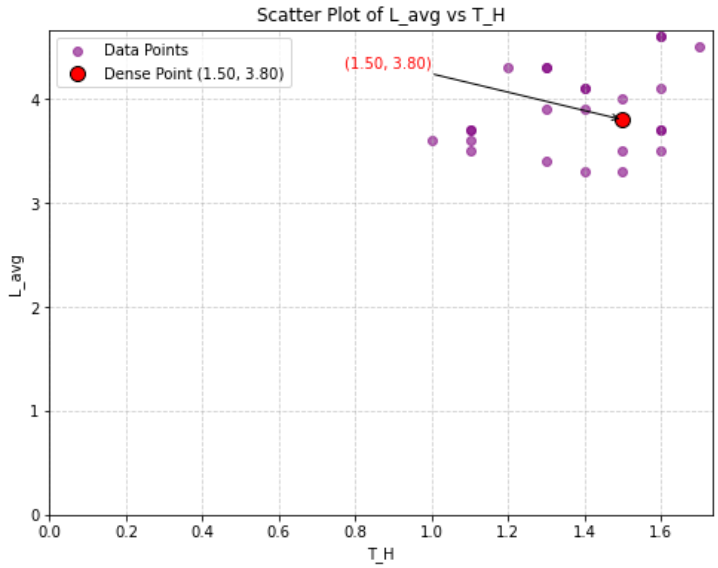


Figure 14 L_V versus T_H scatter plot with KDE for finding the densest point.

From Figure 14, the optimal queue growth parameters are identified as 1.5 seconds for average time headway (T_H) and 3.8 meters for the average vehicle length (L_V).

4.4.3 Optimisation Based on Multiple Queues

As an alternative to the optimisation proposed above (4.4.2), an optimisation by considering all queue profiles at the same time is explored. The objective function in *Eq.* 8 is modified as shown in Eq. 11.

$$z = minimise \left[\left(\frac{1}{N}\right) \sum_{Q=1}^{N} \frac{1}{T} \sum_{t=0}^{T} \left| q_{Q,ASM-ED,t} - q_{Q,QD-QP,t} \right| \right] \qquad \textit{Eq. 11}$$

Where:

- $q_{Q,ASM-ED,t}$ is the location of the realistic queue tail of queue index Q at time t obtained from the ASM-ED module

- $q_{CD-CP,t}$ is the location of the queue tail of queue index Q estimated by the CD-CP algorithm at time t. This value is dependent on the queue growth rate and hence is in turn dependent on L_V and T_H.

To improve the efficiency of the optimiser, instead of a brute force optimiser, a variation of Newton Raphson method called the TNC (Truncated Newton Constrained) optimiser available within the `scipy.optimise` python package is used with the constraints presented in Eq. 9 and Eq. 10.

The optimal parameters obtained after the implementation is as follows:

- Average vehicle length (L_V) is 3.9 meters.
- Average time headway (T_H) is 1.5 seconds.

These values are comparable to the values obtained by applying KDE on the optimal values obtained from individual queue-based optimisation. Hence, these values are recommended as the optimal values for the study corridor based on the dataset used for this optimisation.

5 Discussions

This paper developed a data-driven framework for the calibration of a rule based VSL system. By leveraging historical traffic data, the framework systematically adjusts thresholds and control parameters, reducing reliance on manual calibration.

For calibrating the Congestion Detection parameters – activation and deactivation thresholds, historic speed time series data is used to dynamically identify the onset and dissipation of queues using the Smoothed Difference of Squared Speeds (SDSS) ($\Delta_2 V$) approach. The minimum value of $\Delta_2 V$ represents the queue onset condition, while the maximum value of $\Delta_2 V$ identified after the onset, indicates the queue deactivation condition.

For each day, the corresponding traffic state variables - speed and occupancy - at the onset and deactivation points are extracted from historical traffic data over an extended period (e.g., 12 months). These onset and deactivation points are then plotted on speed-occupancy fundamental diagrams. To robustly determine the thresholds, KDE is applied to identify the densest region of these points on both the activation and deactivation diagrams. The speed and occupancy values corresponding to the densest points are chosen as the activation and deactivation thresholds for the queue onset and termination. These thresholds provide a data-driven, systematic approach to triggering and deactivating queue protection based on actual congestion patterns.

For calibrating the congestion growth rate parameters – average vehicle length and average time headway, the key idea is to consider the realistic queue profiles obtained from the ASM-ED module as the ground truth queue and find the optimal queue growth rate parameters of the CD-CP algorithm (T_H and L_v) that will provide queue growth rates that match the realistic queue profiles.

An optimisation strategy focusing on optimising the parameters for each individual queue was presented. The optimal queue growth rate parameters for different candidate queues and their distributions are presented in the report. From these distributions, it was observed that the optimal vehicle length parameter (L_v) was centered around 3.8 to 3.9 meters and the optimal average time headway parameter (T_H) was centred around 1.4 seconds. To get a single pair of value for the optimal parameters, an L_v versus T_H scatter plot was generated, and KDE process was applied to find the densest point. From the analysis, the optimal queue growth parameters

are identified as 1.5 seconds for average time headway and 3.8 meters for the average vehicle length.

As an alternative to the individual queue-based optimisation, an optimisation by considering all queue profiles at the same time was also presented. The optimal parameters obtained after the implementation were 3.9 meters for average vehicle length (L_v) and 1.5 seconds for average time headway (T_H). These values are comparable to the values obtained by applying KDE on the optimal values obtained from individual queue-based optimisation. Hence, these values are recommended as the optimal values for the study corridor based on the dataset used for this optimisation.

The significance of this research lies in the development of a robust, data-driven auto-calibration framework for rule based VSL algorithm parameters, which leverages historical traffic data to systematically identify activation, deactivation, and queue growth rate thresholds. By making the calibration process data-driven, the framework significantly reduces reliance on manual intervention and associated human efforts, offering a scalable and efficient solution for traffic management systems.

References

Abdel-Aty, M., Dilmore, J., & Dhindsa, A. (2006). Evaluation of variable speed limits for real-time freeway safety improvement. *Accident Analysis & Prevention*, *38*(2), 335–345. https://doi.org/https://doi.org/10.1016/j.aap.2005.10.010

Alessandri, A., Febbraro, A. di, Ferrara, A., & Punta, E. (1999). Nonlinear optimization for freeway control using variable-speed signaling. *IEEE Transactions on Vehicular Technology*, *48*(6), 2042–2052. https://doi.org/10.1109/25.806796

Brown, R. G. (1956). Exponential Smoothing for Predicting Demand. *The Journal of the Operational Research Society*, *7*, 35–42.

Gregurić, M., Kušić, K., & Ivanjko, E. (2022). Impact of Deep Reinforcement Learning on Variable Speed Limit strategies in connected vehicles environments. *Engineering Applications of Artificial Intelligence*, *112*, 104850. https://doi.org/https://doi.org/10.1016/j.engappai.2022.104850

Hegyi, A., De Schutter, B., & Hellendoorn, H. (2005). Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies*, *13*(3), 185–209. https://doi.org/https://doi.org/10.1016/j.trc.2004.08.001

Hegyi, A., Hoogendoorn, S. P., Schreuder, M., Stoelhorst, H., & Viti, F. (2008). SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory. *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 827–832. https://doi.org/10.1109/ITSC.2008.4732611

Heyao, G., Hongfei, J., Ruiyi, W., Qiuyang, H., Jingjing, T., Chao, L., & Xiaochao, W. (2024). Variable Speed Limit Control for Mixed Traffic Flow on Highways Based on Deep-Reinforcement Learning. *Journal of Transportation Engineering, Part A: Systems*, *150*(3), 04023147. https://doi.org/10.1061/JTEPBS.TEENG-8116

Hoogen, E. van den, & Smulders, S. (1994). Control by variable speed signs: results of the Dutch experiment. *Seventh International Conference on Road Traffic Monitoring and Control, 1994.*, 145–149. https://doi.org/10.1049/cp:19940444

Ke, Z., Li, Z., Cao, Z., & Liu, P. (2021). Enhancing Transferability of Deep Reinforcement Learning-Based Variable Speed Limit Control Using Transfer Learning. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 4684–4695. https://doi.org/10.1109/TITS.2020.2990598

Knoop, V. L., Duret, A., Buisson, C., & Arem, B. van. (2010). Lane distribution of traffic near merging zones influence of variable speed limits. *13th International IEEE Conference on Intelligent Transportation Systems*, 485–490. https://doi.org/10.1109/ITSC.2010.5625034

Kušić, K., Dusparic, I., Guériau, M., Gregurić, M., & Ivanjko, E. (2020). Extended Variable Speed Limit control using Multi-agent Reinforcement Learning. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–8. https://doi.org/10.1109/ITSC45102.2020.9294639

Kušić, K., Ivanjko, E., & Gregurić, M. (2018). A Comparison of Different State Representations for Reinforcement Learning Based Variable Speed Limit Control. *2018 26th Mediterranean Conference on Control and Automation (MED)*, 1–6. https://doi.org/10.1109/MED.2018.8442986

Li, D., & Ranjitkar, P. (2015). A fuzzy logic-based variable speed limit controller. *Journal of Advanced Transportation*, *49*(8), 913–927. https://doi.org/https://doi.org/10.1002/atr.1320

Li, Z., Liu, P., Xu, C., Duan, H., & Wang, W. (2017). Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks. *IEEE Transactions on Intelligent Transportation Systems*, *18*(11), 3204–3217. https://doi.org/10.1109/TITS.2017.2687620

Li, Z., Xu, C., Guo, Y., Liu, P., & Pu, Z. (2021). Reinforcement Learning-Based Variable Speed Limits Control to Reduce Crash Risks Near Traffic Oscillations on Freeways. *IEEE Intelligent Transportation Systems Magazine*, *13*(4), 64–70. https://doi.org/10.1109/MITS.2019.2907631

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, *33*(3), 1065–1076.

Rämä, Pirkko. (1999). Effects of Weather-Controlled Variable Speed Limits and Warning Signs on Driver Behavior. *Transportation Research Record*, *1689*(1), 53–59. https://doi.org/10.3141/1689-08

Smulders, S. A., & Helleman, D. E. (1998). Variable speed control: state-of-the-art and synthesis. *9th International Conference on Road Transport Information and Control, 1998. (Conf. Publ. No. 454)*, 155–159. https://doi.org/10.1049/cp:19980174

Sun, D., Jamshidnejad, A., & Schutter, B. De. (2024). A Novel Framework Combining MPC and Deep Reinforcement Learning With Application to Freeway Traffic Control. *IEEE Transactions on Intelligent Transportation Systems*, *25*(7), 6756–6769. https://doi.org/10.1109/TITS.2023.3342651

Treiber, M., & Helbing, D. (2002). Reconstructing the Spatio-Temporal Traffic Dynamics from Stationary Detector Data. *Cooper@tive Tr@nsport@tion Dyn@mics*, *1*. https://mtreiber.de/publications/Reconstructing_the_spatio-temporal_traffic_dynamic.pdf

Vinitsky, E., Parvate, K., Kreidieh, A., Wu, C., & Bayen, A. (2018). Lagrangian Control through Deep-RL: Applications to Bottleneck Decongestion. *2018 21st International*

Conference on Intelligent Transportation Systems (ITSC), 759–765. https://doi.org/10.1109/ITSC.2018.8569615

Vrbanić, F., Ivanjko, E., Mandžuka, S., & Miletić, M. (2021). Reinforcement Learning Based Variable Speed Limit Control for Mixed Traffic Flows. *2021 29th Mediterranean Conference on Control and Automation (MED)*, 560–565. https://doi.org/10.1109/MED51440.2021.9480215

Walraven, E., Spaan, M. T. J., & Bakker, B. (2016). Traffic flow optimization: A reinforcement learning approach. *Engineering Applications of Artificial Intelligence*, *52*, 203–212. https://doi.org/https://doi.org/10.1016/j.engappai.2016.01.001

Wang, C., Zhang, J., Xu, L., Li, L., & Ran, B. (2019). A New Solution for Freeway Congestion: Cooperative Speed Limit Control Using Distributed Reinforcement Learning. *IEEE Access*, *7*, 41947–41957. https://doi.org/10.1109/ACCESS.2019.2904619

Zackor, H. (1991). Speed Limitation on Freeways: Traffic-Responsive Strategies. In M. PAPAGEORGIOU (Ed.), *Concise Encyclopedia of Traffic & Transportation Systems* (pp. 507–511). Pergamon. https://doi.org/https://doi.org/10.1016/B978-0-08-036203-8.50106-1

Zhu, F., & Ukkusuri, S. V. (2014). Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, *41*, 30–47. https://doi.org/https://doi.org/10.1016/j.trc.2014.01.014