Developing Analysis, Modeling, and Simulation Tools for Connected and Automated Vehicle Applications

Algorithm Description Document: Speed Harmonization Model

August 2020

U.S. Department of Transportation
Federal Highway Administration

Research, Development, and Technology
Turner-Fairbank Highway Research Center
6300 Georgetown Pike
McLean, VA 22101-2296

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle
Developing Analysis, Modeling, and Simulation Tools for Connected and Automated Vehicle Applications Algorithm Description Document: Speed Harmonization Model

5. Report Date
August 2020

6. Performing Organization Code:

7. Author(s)
Hani Mahmassani, Moein Hosseini, Alireza Talebpour, Amr Elfar, and Zhitong Huang

8. Performing Organization Report No.

9. Performing Organization Name and Address
Leidos Inc.
11251 Roger Bacon Drive
Reston, VA 20190

10. Work Unit No.

11. Contract or Grant No.
DTFH61-12-D-00030, TO 22

12. Sponsoring Agency Name and Address
Office of Operations Research and Development
Federal Highway Administration
6300 Georgetown Pike
McLean, VA 22101-2296

13. Type of Report and Period Covered

14. Sponsoring Agency Code

15. Supplementary Notes
The government task managers were John Halkias and Gene McHale.

16. Abstract
This study presents a predictive speed harmonization system that utilizes the detailed vehicle trajectories of connected vehicles, the communication capabilities of connected vehicles, and machine learning algorithms. This innovative system collects detailed information from the connected vehicles within a road segment of interest, predicts locations of traffic congestion, and updates the speed limits for the connected vehicles in order to mitigate congestion. To leverage the opportunities created by the V2I and V2V communication systems, we developed a simulation tool that incorporates a variety of speed harmonization strategies: centralized, decentralized, and optimization-based. Case studies of multiple operational scenarios show that the proposed speed harmonization system can reduce the severity and lengths of traffic shockwaves and improve the overall traffic stability. The decentralized and optimization-based strategies are more effective in improving the traffic stability by decreasing the speed and travel time variations. They further enhance the traffic stability by decreasing the maximum density and by providing a smoother transition from the uncongested state to the congested state.
In order to help future users to implement the proposed predictive speed harmonization system, this document also includes pseudocode of this system in the appendix.

17. Key Words
Connected and automated vehicle, traffic simulation, microsimulation, coordinated merging, speed harmonization, and market penetration rate.

18. Distribution Statement
No restrictions.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	79	N/A

Form DOT F 1700.7 (8-72)

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1,000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2,000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2,000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	2.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACC	adaptive cruise control
ADAS	advanced driver assistance systems
AMS	analysis, modeling, and simulation
AV	automated vehicle
CAV	connected and automated vehicle
CACC	cooperative adaptive cruise control
CV	connected vehicle
SPDHRM	speed harmonization
IDM	intelligent driver model
NGSIM	Next Generation SIMulation
TMC	traffic management center
TTI	travel time index
UAV	unmanned aerial vehicle
USDOT	United States department of transportation
V2V	vehicle-to-vehicle
V2I	vehicle-to-infrastructure

EXECUTIVE SUMMARY

Speed harmonization is an active traffic management strategy that is used for delaying traffic flow breakdown and mitigating congestion by changing speed limits based on prevailing traffic, weather, and road conditions. Traditional implementations utilize fixed roadway sensors to collect traffic information and variable speed limit signs at fixed locations to display updated speed limits. Moreover, most implementations use simple models such as a reactive rule-based decision tree to activate the control strategy. Due to the incomplete picture of traffic flow dynamics provided by the fixed infrastructure sensors, the effectiveness of these strategies is impaired. Furthermore, since the road sensors and the advisory speed signs are in fixed locations, the performance is reduced due to the limited set of scenarios that this type of implementation can accommodate. Last but not least, reactive speed harmonization strategies are generally less effective than predictive ones.

To overcome the aforementioned challenges, this study presents a predictive speed harmonization system that utilizes the detailed vehicle trajectories of connected vehicles, the communication capabilities of connected vehicles, and machine learning algorithms. This innovative system collects detailed information from the connected vehicles within a road segment of interest, predicts locations of traffic congestion, and updates the speed limits for the connected vehicles in order to mitigate congestion. To leverage the opportunities created by the V2I and V2V communication systems, we developed a simulation tool that incorporates a variety of speed harmonization strategies: centralized, decentralized, and optimization-based. Case studies of multiple operational scenarios show that the proposed speed harmonization system can reduce the severity and lengths of traffic shockwaves and improve the overall traffic stability. The decentralized and optimization-based strategies are more effective in improving the traffic stability by decreasing the speed and travel time variations. They further enhance the traffic stability by decreasing the maximum density and by providing a smoother transition from the uncongested state to the congested state.Chapter 1 defines the problem, its objective, and the challenges of the traditional system. Chapter 2 provides detailed information of the speed harmonization framework and the logic behind each component of the framework. Furthermore, mathematical models that are used in different modules of the system are elaborated. In chapter 3, it is shown how the data used in the current study was calibrated. The basic guidance on how to implement the proposed framework into a simulation tool and a case study that shows the performance of the proposed model are presented in chapters 4 and 5, respectively. Chapter 6 concludes this study with a summary of findings and provides recommendations for a successful implementation of the speed harmonization framework. In the appendix, the pseudocode of the functions used in the developed simulation tool are provided. The information essentially can help transportation professionals to incorporate the methodology in other traffic analysis tools.

This research has been performed under a FHWA project entitled “Developing Analysis, Modeling, and Simulation Tools for Connected and Automated Vehicle Applications” (contract number: DTFH6116D00030-0022). To get more information of this FHWA project, readers are encouraged to reference the final project report of this project (Lu et al, forthcoming). This report is under FHWA publication process and it will be available soon.

CHAPTER 1. PURPOSE OF THIS MODEL

PURPOSE OF THIS DOCUMENT

Connected and automated vehicle (CAV) technologies offer potentially transformative societal impacts, including significant mobility, safety, and environmental benefits. State and local agencies are interested in harnessing the potential benefits of CAVs. However, for agencies to be able to plan beneficial deployments of infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) technology, it is important to be able to robustly predict the impacts of such deployments and identify which applications best address their unique transportation problems. Traffic analysis, modeling, and simulation (AMS) tools provide an efficient means to evaluate transportation improvement projects before deployment.

However, current AMS tools are not well suited for evaluating CAV applications due to their inability to represent vehicle connectivity and automated driving features. The development of a new generation of tools involves spending a lot of resources and time to develop, calibrate, and validate. Many independent researchers have developed models of CAV systems based on a divergent array of underlying assumptions. As a result, there is little consensus in the literature regarding the most likely impacts of CAV technologies.

Thus, there is a desire for a consistent set of models to produce realistic and believable predictions of CAV impacts. These models can be based on the best available data and include the most accurate possible representations of the behaviors of drivers of conventional vehicles and CAVs. Deployment concepts, strategies, and guidelines are also key for allowing State and local agencies to understand how and where to deploy CAV technologies.

To meet these goals, the Federal Highway Administration (FHWA) sponsored a project entitled 'Developing Analysis, Modeling, and Simulation Tools for Connected and Automated Vehicle Applications' (contract number: DTFH6116D00030-0022). This project aimed to develop AMS models for the most prominent CAV applications and to incorporate these models into existing AMS simulation tools. Three CAV applications were developed under this project: a lane changing (LC) model for light duty CAVs, a joint application model that integrates speed harmonization (SPDHRM) and coordinated merge (CM), and an improved cooperative adaptive cruise control (CACC) model for light duty CAVs. The final project report (Lu et al, forthcoming) is under FHWA publication process and it will be available soon.

This document presents the SPDHRM model of the joint application in detail. The objective of this document is to provide detailed information of this model to benefit the CAV simulation community. This document is expected to help future users easily adopt and customize this model in a traffic simulation tool to meet their simulation needs. To this end, this document describes the algorithms/logic of this model in detail. It also illustrates how this model was developed, calibrated, and validated. Pseudocode of this model is included in the appendix.

PURPOSE OF THIS MODEL

SPDHRM, as a traffic control strategy, adjusts the speed limit of a freeway section based on prevailing traffic conditions (Talebpour, Mahmassani, & Hamdar, 2013). The strategy helps mitigate shockwave formation, damp shockwave propagation, improve traffic homogeneity by minimizing the spatial variance of speed, and accelerate the recovery from a traffic breakdown

(H. Mahmassani, Rakha, Hubbard, & Lukasik, 2012; Hani S Mahmassani, 2016). Benefits of implementing such a strategy are (Hani S Mahmassani, Elfar, Shladover, & Huang, 2018):

- Traffic safety improvement by effectively delaying or eliminating traffic breakdown (Talebpour et al., 2013);
- Reduced fuel consumption and emissions as a result of effectively suppressing the development of vehicle speed oscillation (Li, Cui, An, & Parsafard, 2014; Wang, Daamen, Hoogendoorn, & Van Arem, 2015; Yang & Jin, 2014); and
- Traffic efficiency improvement by reducing the total time spent in the network (Wang et al., 2015).

Shockwave detection and speed limit broadcasting to upstream vehicles constitute the major components of the SPDHRM strategy. Traditionally, this strategy is conducted using sensors embedded in the infrastructure for the shockwave detection component, and variable speed limit signs at prespecified locations for the speed limit broadcasting component. This setup faces three main challenges. First, fixed infrastructure sensors provide an incomplete representation of traffic flow dynamics, which can significantly diminish the effectiveness of the strategy. Second, relying on fixed road sensors and signs significantly affects the strategy performance. Finally, it is difficult to develop accurate models that are capable of predicting the future traffic state utilizing data from fixed traffic sensors (Elfar, Talebpour, & Mahmassani, 2019, 2020).

Connected vehicles (CV) as probe vehicles can monitor their surrounding traffic conditions and communicate that information to the infrastructure and to other connected vehicles. Capabilities of CVs enable more accurate shockwave detection and greater range of effectiveness in speed limit broadcasting compared to conventional methods. Besides the benefits of CVs, AVs can help to dampen the effect of shockwaves in the traffic flow.

DOCUMENT OVERVIEW

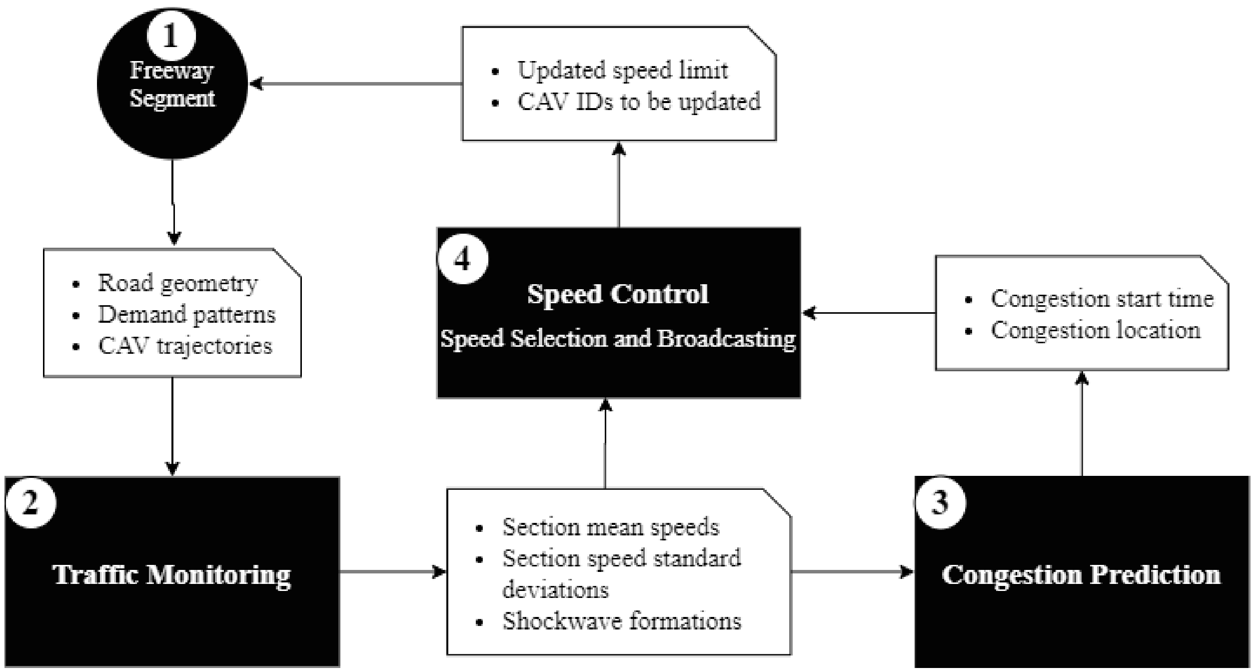
This document will introduce a new SPDHRM model capable of reducing or eliminating shockwaves by utilizing the big stream of data generated by CAVs and the predictive capability of machine learning algorithms. In the following sections, we introduce the model details and the logic behind each element in the model. Model calibration and validation is discussed next. This section is followed by a simulation-based analysis of the impacts of the proposed model on traffic flow dynamics. The document concludes with a summary of findings.

CHAPTER 2. MODEL DEVELOPMENT AND LOGIC

This chapter discusses the model development procedure and provides details about the overall design of the SPDHRM model.

DESCRIPTIONS OF MODEL LOGIC

A set of novel SPDHRM algorithms were developed that utilize machine learning to predict the onset of congestion and to activate the SPDHRM in a highway segment. These algorithms also utilize various methods of communicating the updated speed limits to the CVs (automated or human-driven) and non-CVs (automated or human-driven). The overall framework is shown in Figure 1.

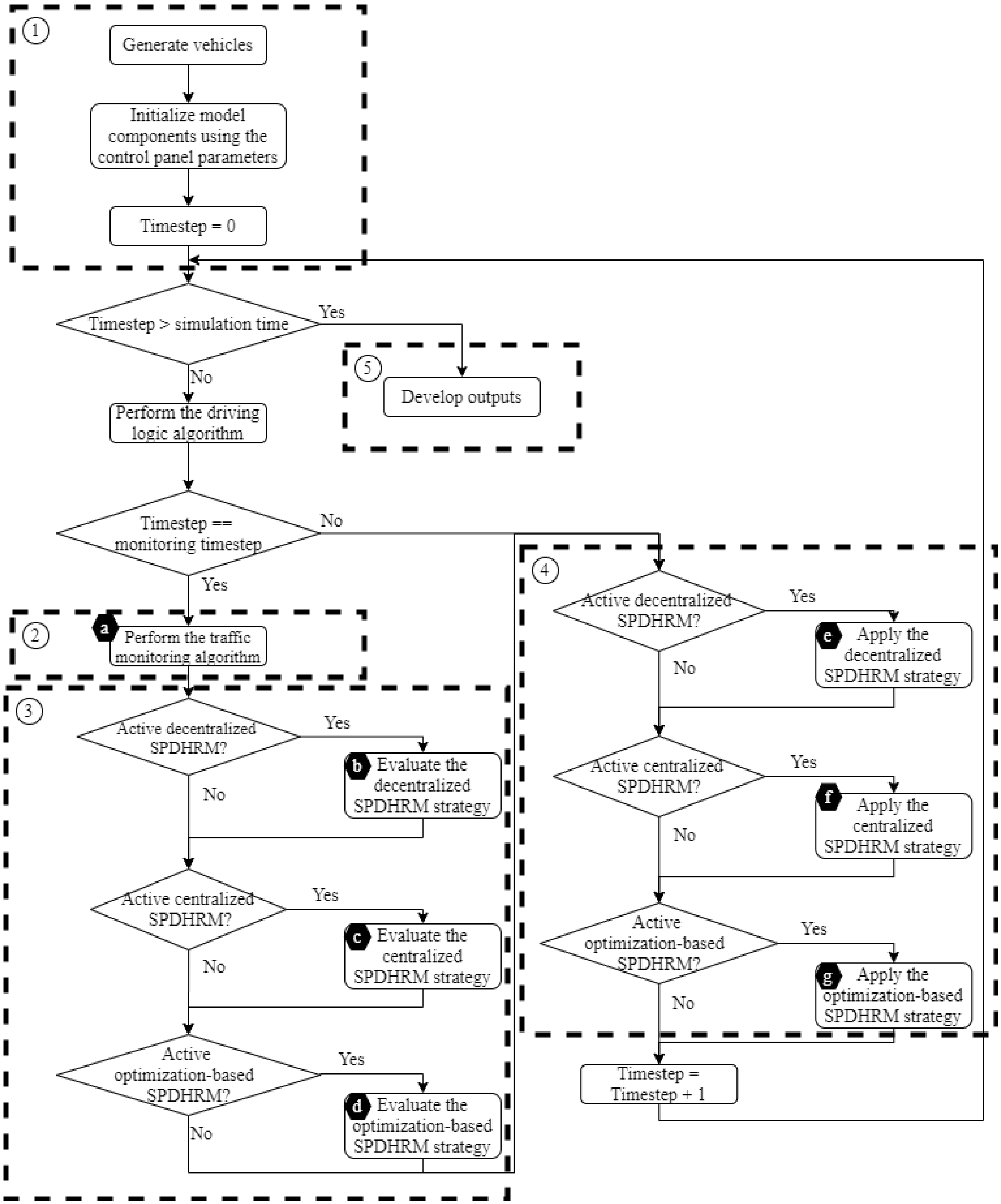


© Amr Elfar

CAV = connected automated vehicle. V2I = vehicle to infrastructure. ID = identification number.

Figure 1. Diagram. SPDHRM overall framework (Elfar 2019).

The main algorithm of the simulation tool is shown in Figure 2. As shown in the figure, the first component corresponds to the Freeway Segment element of the framework. It includes the inputs, outputs, and the driving logic of the tool. The driving logic contains the CF and LC models that specify the interaction among vehicles. The Traffic Monitoring module of the framework relates to the second element of the algorithm. Then, as the Congestion Prediction module (third element of the algorithm), the model predicts the congestion characteristics and evaluates the SPDHRM strategy selected by the user (decentralized, centralized, or optimization-based). The fourth element in the algorithm is related to the Speed Control module of the framework. This part of the algorithm implements the SPDHRM strategy by determining the advisory speed for each vehicle.

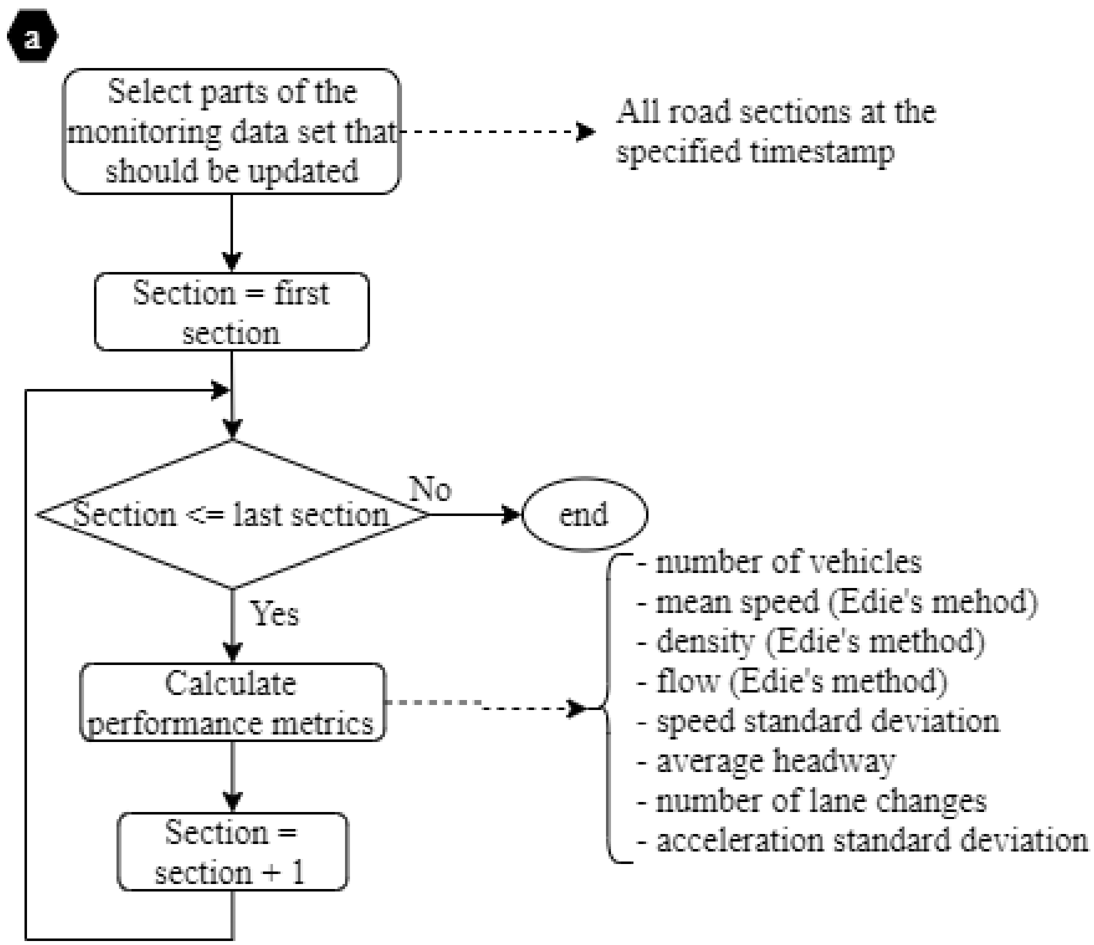


Source: FHWA

SPDHRM = speed harmonization.

Figure 2. Flowchart. Main algorithm.

The traffic monitoring module, as shown in Figure 3, calculates the performance metrics for all sections of the road segment at each monitoring timestep. The performance metrics are calculated using the CVs and CAVs in order to be used in the congestion prediction and speed control modules. Simultaneously, the performance metrics are calculated for all vehicles in the system to evaluate the accuracy of the prediction models. As a result, to evaluate and implement the SPDHRM strategies, only the information of CAVs is analyzed.

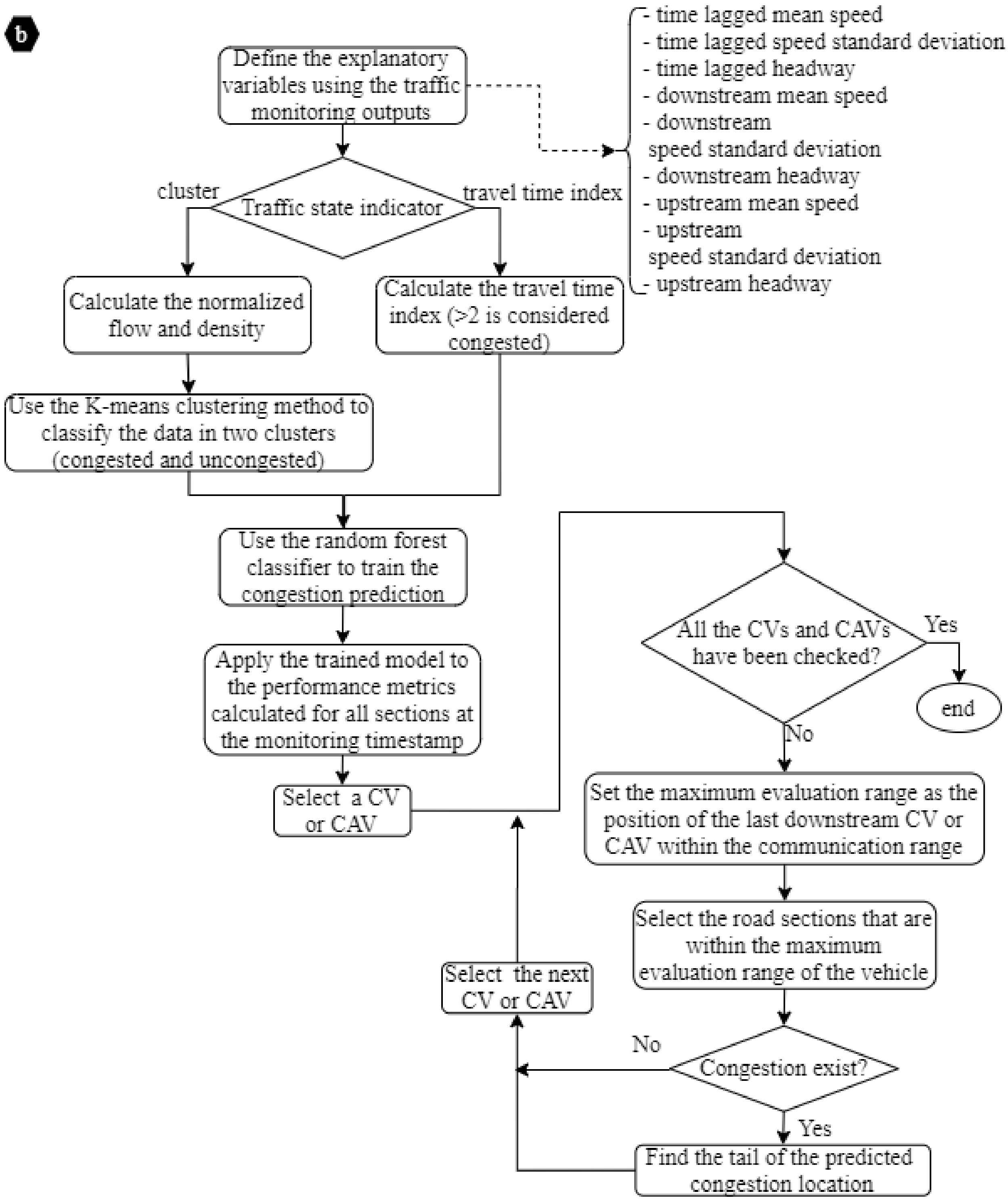


Source: FHWA

Figure 3. Flowchart. Traffic monitoring algorithm.

Figure 4 to Figure 6 show the algorithms in the congestion prediction module. Based on the settings defined in the control panel, one of the algorithms is performed: centralized, decentralized, or optimization-based. This module determines the location and time of the congestion along with the advisory speed limit. The training part of the algorithms could be performed in an online or offline manner. The training part could be skipped if a predefined congestion prediction model is utilized.

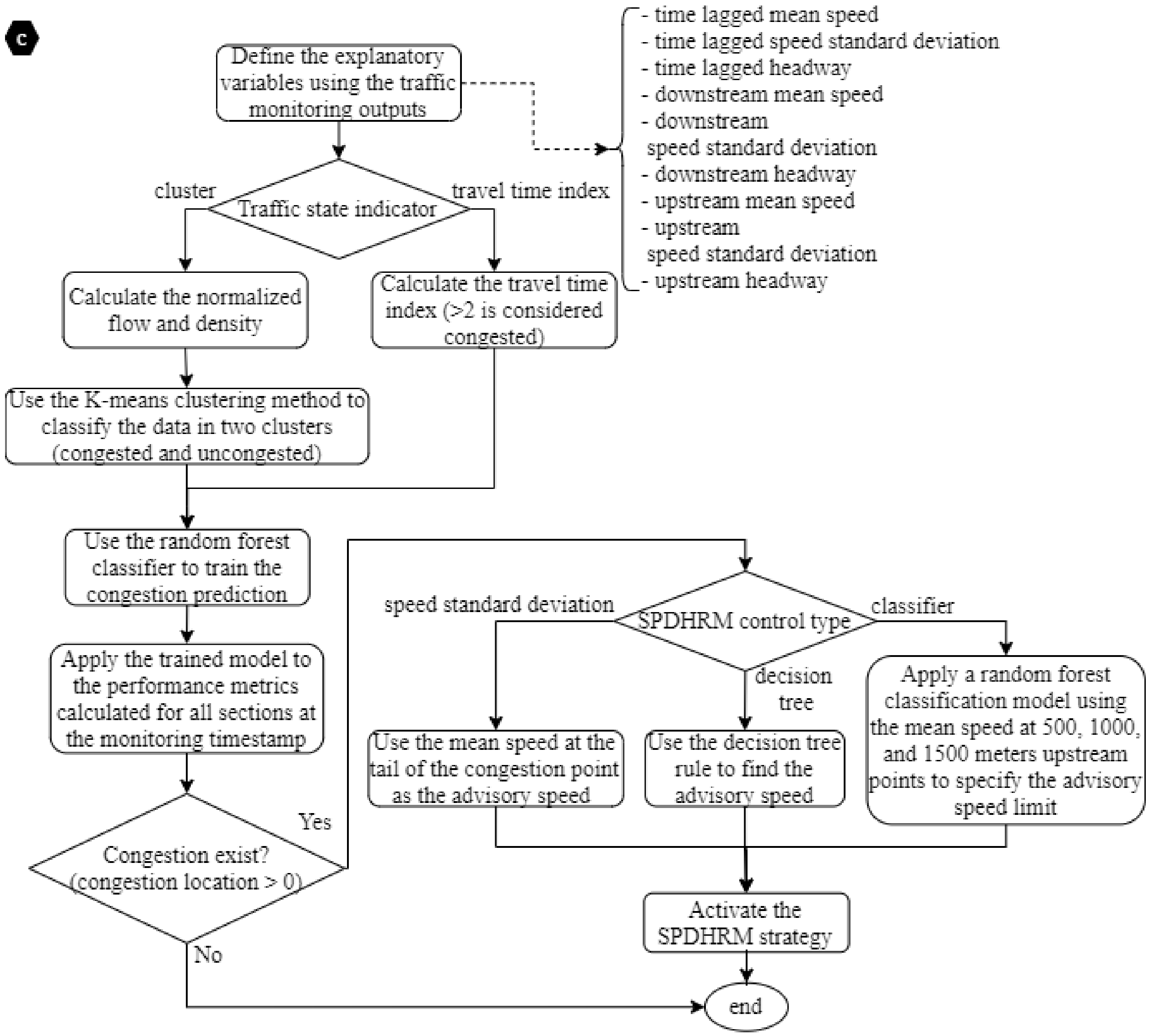
In the centralized strategy, the system evaluates the state of the transport facility through information received from CAVs and detectors. Then, it predicts future states using machine learning algorithms. Finally, the advisory messages are prepared to be broadcasted to CAVs in order to minimize disturbance (speed standard deviation). On the other hand, in the decentralized strategy, each CAV receives information from a cluster or fleet of CAVs within a detection/connection range. Then, each vehicle utilizes individualized or group-based machine learning algorithms to predict the future state of clusters. Finally, the vehicles adjust their longitudinal and lateral driving behavior to minimize disruption in a cluster or fleet of vehicles, i.e., self-homogenize. The advisory speed limit in the optimization-based strategy is determined by solving an optimization problem that seeks to maximize the distance traveled by the vehicles in a specified time period (prediction time horizon). Based on the available computational resources, the complexity of the optimization problem could be adjusted ranging from jointly determining the advisory speed for each vehicle and the broadcasting distance (high complexity) to selecting the advisory speed limit and the broadcasting distance from a limited set (low complexity).



Source: FHWA

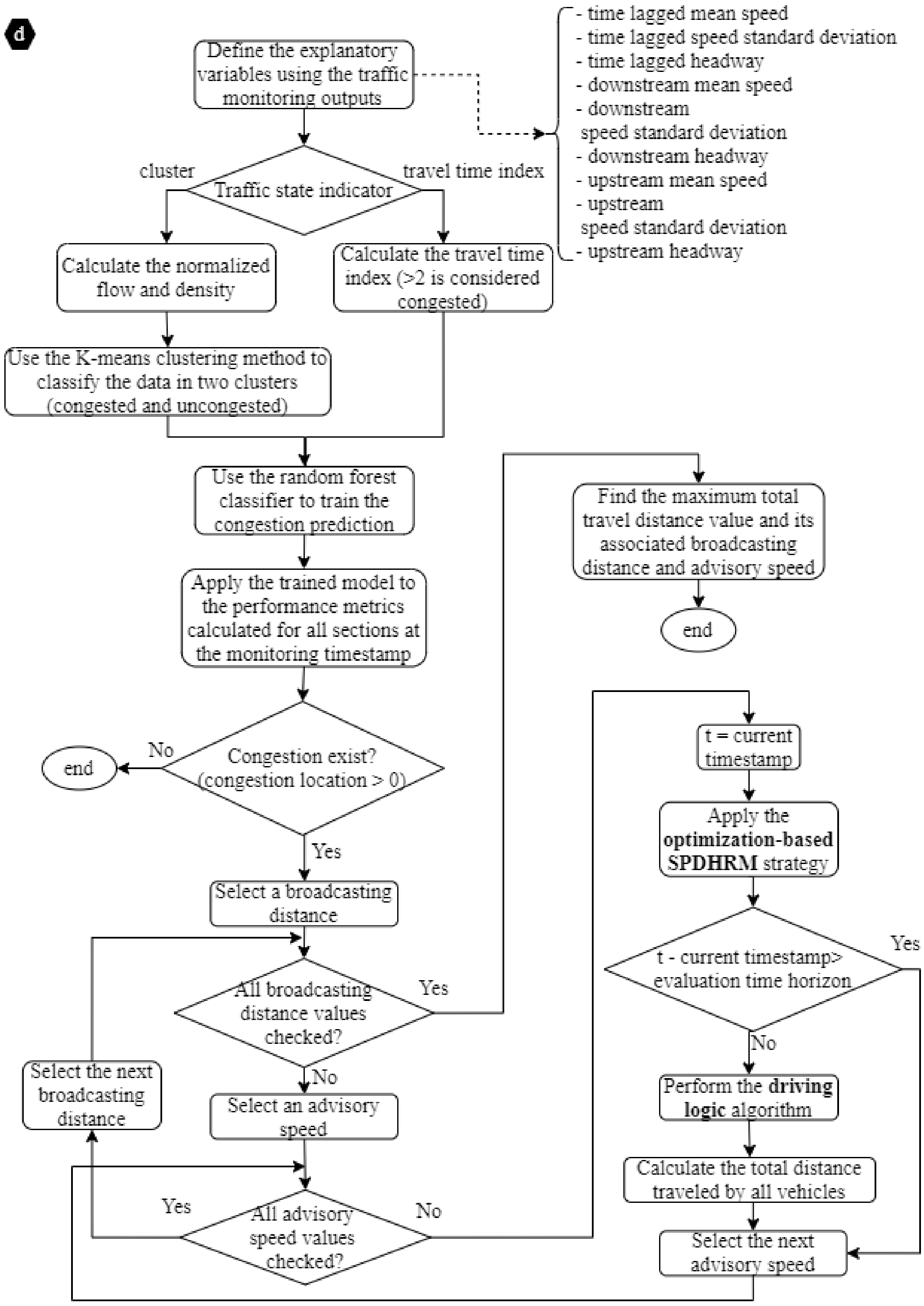
CV = connected vehicle. CAV = connected automated vehicle.

Figure 4. Flowchart. Decentralized SPDHRM strategy evaluation algorithm.



Source: FHWA
SPDHRM = speed harmonization.

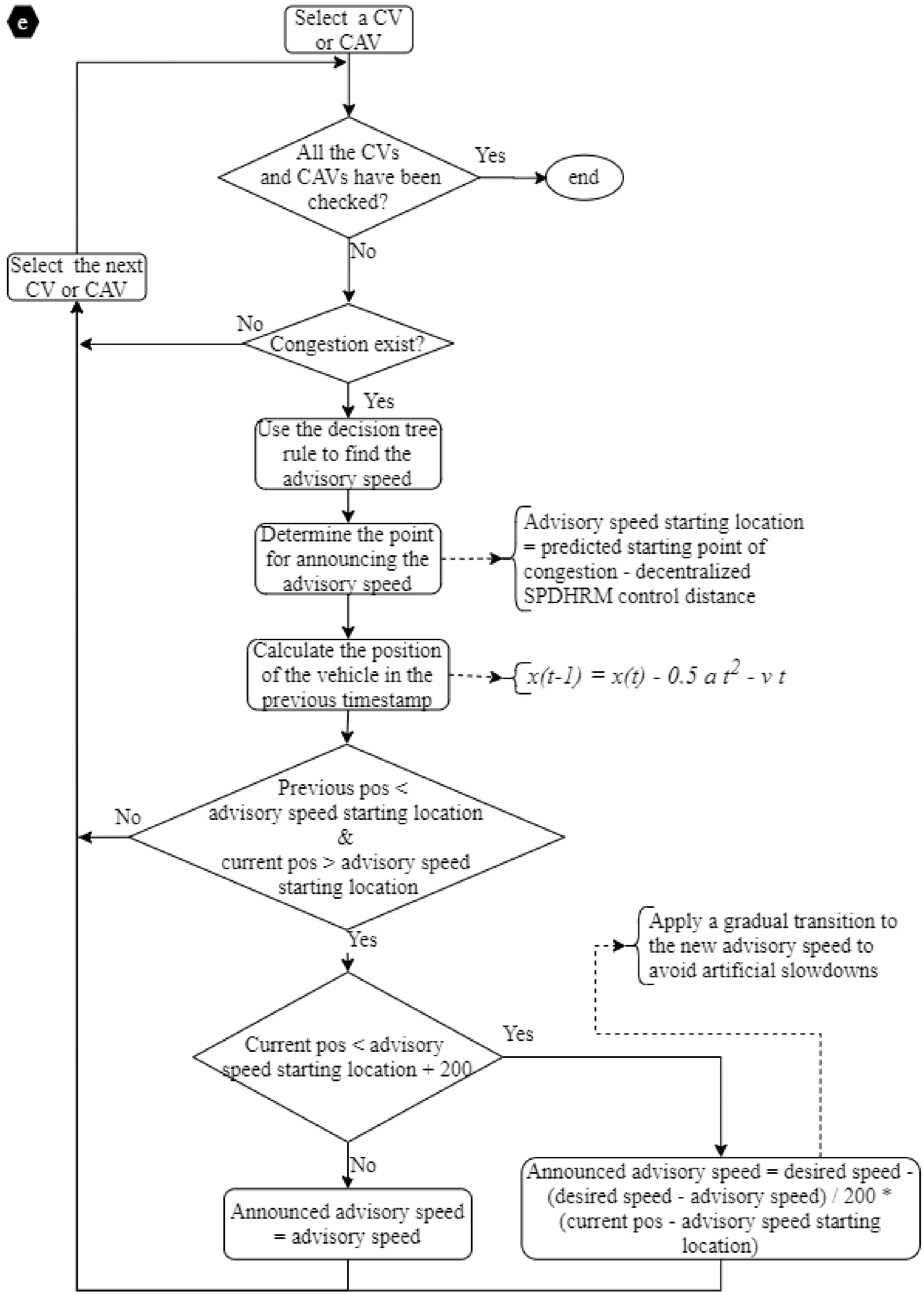
Figure 5. Flowchart. Centralized SPDHRM strategy evaluation algorithm.



Source: FHWA
SPDHRM = speed harmonization.

Figure 6. Flowchart. Optimization-based SPDHRM strategy evaluation algorithm.

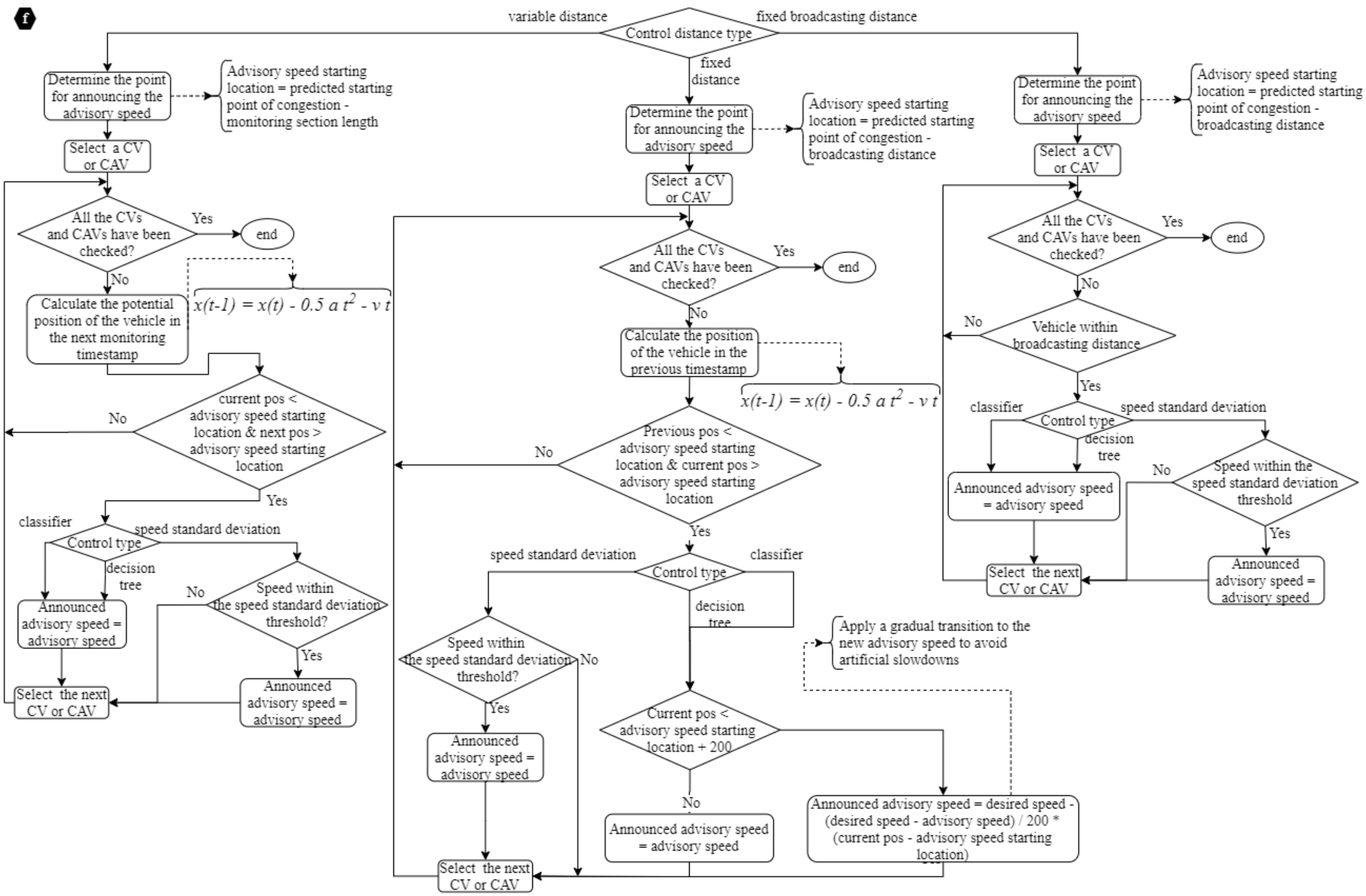
Figure 7 to Figure 9 show the algorithms of the speed control module. A step-by-step procedure is performed to communicate an advisory speed limit to CVs and AVs. Extra precautions are taken to prevent artificial slowdowns in the simulation by applying a gradual transition in the advisory speed limit communicated to the vehicles.



Source: FHWA

CV = connected vehicle. CAV = connected automated vehicle. x = position. a = acceleration. v = speed. t = timestep.

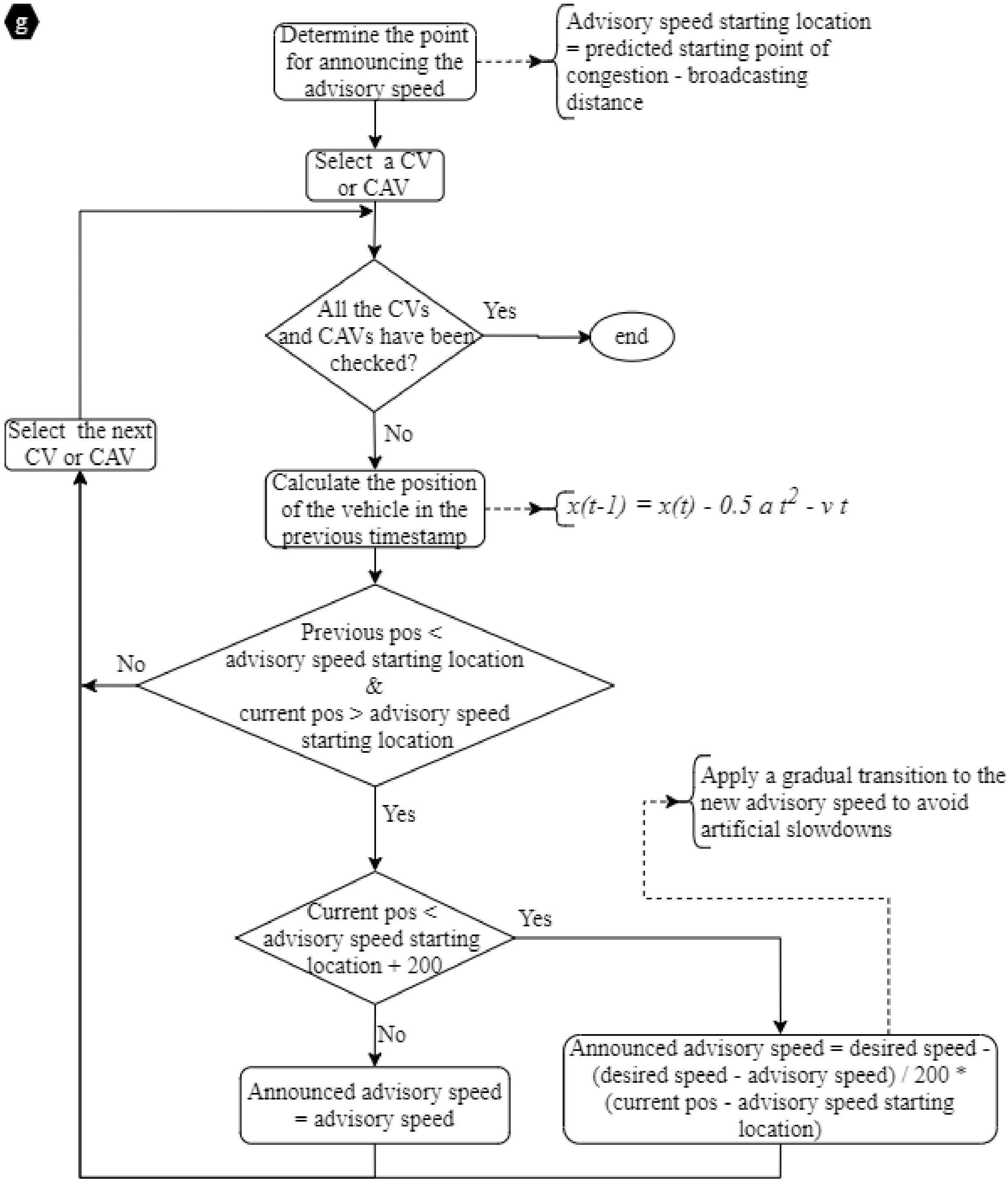
Figure 7. Flowchart. Decentralized SPDHRM strategy implementation algorithm.



Source: FHWA

a = acceleration. CV = connected vehicle. CAV = connected automated vehicle. x = position. a = acceleration. v = speed. t = timestep.

Figure 8. Flowchart. Centralized SPDHRM strategy implementation algorithm.



Source: FHWA

CV = connected vehicle. CAV = connected automated vehicle. x = position. a = acceleration. v = speed. t = timestep.

Figure 9. Flowchart. Optimization-based SPDHRM strategy implementation algorithm.

MODEL DEVELOPMENT

In the traffic monitoring module, the study area is divided into sections with a specified monitoring length determined by the user. Flow, mean speed, and density for each section at each timestep were calculated using Edie's generalized definitions for individual facilities:

$$q(A) = \frac{d(A)}{|A|}$$

(a) Edie's definition of flow

$$k(A) = \frac{t(A)}{|A|}$$

(b) Edie's definition of density

$$v(A) = \frac{d(A)}{t(A)}$$

(c) Edie's definition of speed

Figure 10. Formulas. Edie's generalized definition of the traffic flow characteristics.

where A represents a spatiotemporal area which in a highway setting equals the number of lanes times the section length times the monitoring timestep. $q(A)$, $k(A)$, $v(A)$ denote flow, density, and mean speed for observed vehicles in section A, respectively; $d(A)$ and $t(A)$ represent the total distance traveled and total time spent by all vehicles in section A, respectively. Finally, $|A|$ is the area covered by section A. Other statistics of the traffic such as the standard deviation of the speed could be calculated for each road section at the monitoring timestep.

Different methodologies could be used in order to predict the state of the traffic and the location and time of congestion points. Machine learning is one way to take advantage of a large amount of information that can be generated by connected vehicles. Some studies investigated congestion prediction with machine learning using various data sources. For example, Thianniwet et al. (2009) developed a congestion classifier by applying a decision tree algorithm on the movement patterns of vehicles collected through phone global positioning systems (GPS). In another study by Pattara-Atikom (2007) , neural networks were used as a machine learning algorithm to estimate traffic congestion from cell dwell time. Some studies utilized a hybrid of multiple techniques such as a study by Vlahogianni et al. (2008) that identified traffic patterns using clustering and then forecasted flow using neural networks.

The first step for building predictive models is to identify the traffic state from vehicle trajectories. The traffic state is used as the dependent variable in the machine learning-based classification model. Travel time index (TTI) (Texas Transportation Institute, 2006) and K-means clustering (Hartigan, 1975) are the two methods implemented in the tool that would enable the traffic state to be classified in one of the two classes used in this study: congested or uncongested. The K-means clustering algorithm segments data points into clusters (groups) where the total distances calculated from each point to its respective cluster center is minimized. As density and flow values are significantly different, the values were scaled to be between 0 and 1 before running the algorithm. Then, the density values were multiplied by a factor of 5 (arbitrarily chosen) to give density values a higher weight and force the algorithm to split the clusters based on density and define the critical density at which traffic breaks down. The TTI defined in Figure 11 has been suggested as a way of identifying traffic congestion (Dong et al., 2009). The *TTI* could be calculated for each section of the study area.

$$TTI = \frac{Mean\ Speed^{-1}}{Free\ Flow\ Speed^{-1}}$$

Figure 11. Formula. Travel time index.

The congestion prediction module utilizes the traffic characteristics and performance metrics calculated for each road section to predict the location and time of congestion formation within a short time horizon (10–30 seconds [s]). Besides the traffic flow characteristics calculated using Edie's generalized formulation, Elfar et al. (2018) showed that the speed standard deviation could be used as an indicator of shockwave formation and propagation. They found that a 30 percent connectivity level is necessary to identify the shockwave accurately.

For the prediction model that utilizes machine learning algorithms, temporally lagged and spatially distributed variables are developed for the traffic characteristics and performance metrics. As a result, the developed congestion models have temporally lagged models, which are a type of time-series model trained to predict current and future values of the dependent variable using explanatory variable values observed in previous timesteps. Therefore, by plugging in the current values of independent variables, the model predicts the future values of the dependent variable. Elfar (2019) showed that for an accurate prediction of the congestion state of a road section, the following three statistics calculated in the previous monitoring timestep could be considered as the explanatory variables: the mean speed in the specified section, the mean speed in the downstream section, and the speed standard deviation in the downstream section. Elfar et al. (2018) tested three types of machine learning techniques for the prediction model: binary logistic regression, random forests, and artificial neural networks. They found that the models with the binary logistic regression and random forests algorithms performed slightly better than the model with the neural network algorithm. In this simulation tool, the random forest technique is implemented.

Table 1. Variables used in the predictive model.

Variable Type	Variable Name	Description
Dependent variable	Traffic State	Binary: the state of traffic whether congested or uncongested as identified using the travel time index with a threshold above 2 or the K-means clustering algorithm.
Explanatory variable 1	Lagged mean speed in current section	Continuous: the average speed of individual vehicles in the current section, lagged 10, 20, or 30 seconds.
Explanatory variable 2	Lagged mean speed in downstream section	Continuous: the average speed of individual vehicles in the next downstream section, lagged 10, 20, or 30 seconds.
Explanatory variable 3	Lagged speed standard deviation in downstream section	Continuous: the speed standard deviation of individual vehicles in the next downstream section, lagged 10, 20, or 30 seconds.

In the speed control module, the optimization-based SPDHRM strategy evaluates a wide set of potential speed limits. It selects the limit that maximizes traffic speed and mitigates congestion.

The formulation proposed in Figure 12 enables the possibility of assigning a speed limit for each connected vehicle in the system.

$$\max \sum_{t=t_\circ}^{t_\circ+t_{oh}} \sum_{v\in V} DT_{tv}(u_v^{m5})$$
$$u_{min} \le u_v^{m5} \le u_{max}\,, \qquad \forall v \in V^{us}$$
$$u_v^{m5} = 5 * u_v, \qquad \forall v \in V^{us}$$
$$u_v \text{ integer}, \qquad \forall v \in V^{us}$$

Figure 12. Formulas. Mathematical formulation of the optimization-based SPDHRM strategy.

Where $DT_{tv}(u_v^{m5})$ represents the distance traveled by all vehicles in the vehicle set (V^{us}) over an optimization horizon (t_{oh}). The optimal speed of each vehicle as the decision variable varies between a set of minimum (u_{min}) and maximum (u_{max}) values. The second condition in the formulation limits the speed selected for each vehicle to multiples of 5. This would ensure that the drivers could practically adhere to the new speed limits. Maximizing the distance traveled over a fixed time period is equivalent to maximizing the traffic speed.

A simulation-based optimization approach can be adopted in order to find the advisory speed for each vehicle that would collectively result in maximizing the distance traveled by all vehicles because of the interactions of the vehicles captured by the CF, LC behavior, vehicle classes, and traffic control. The primary limitation of simulation-based optimization problems is the computationally intensive and time-consuming simulations that are associated with finding the optimal solution. The number of decision variables could be reduced significantly to practically solve the optimization problem. A simplified version of the problem is shown in Figure 13 where the speed and the broadcasting distance are selected from sets. Furthermore, instead of solving for a unique speed limit per vehicle, the problem is reformulated to solve for one speed limit for all vehicles upstream of the predicted congestion location.

$$\max \sum_{t=t_\circ}^{t_\circ+t_{oh}} \sum_{v\in V} DT_{tv}(u,d)$$
$$u \in U = \{u_{min}\,, (u_{min} + 5),\ \ldots, u_{max}\}$$
$$d \in D$$

Figure 13. Formulas. Simplified mathematical formulation of the optimization-based SPDHRM strategy.

Where *u* is the candidate speed selected from the set of speed values (*U*), and *d* is the candidate broadcasting distance selected from the set of broadcasting distance values (*D*).

CHAPTER 3. MODEL CALIBRATION AND VALIATION

In order to calibrate and validate the SPDHRM calibration, two key features were developed and integrated into Northwestern University's microscopic simulation platform:

- SPDHRM: A set of novel SPDHRM algorithms were developed that utilize machine learning to predict the onset of congestion and to activate the SPDHRM in a highway segment. These algorithms also utilize various methods of communicating the updated speed limits to the CVs (automated or human-driven) and non-CVs (automated or human-driven).

In addition to these models, the simulation platform utilizes several already calibrated and validated CF and LC models for non-connected HVs, connected HVs, CAVs, and non-connected AVs (Talebpour & Mahmassani, 2016).

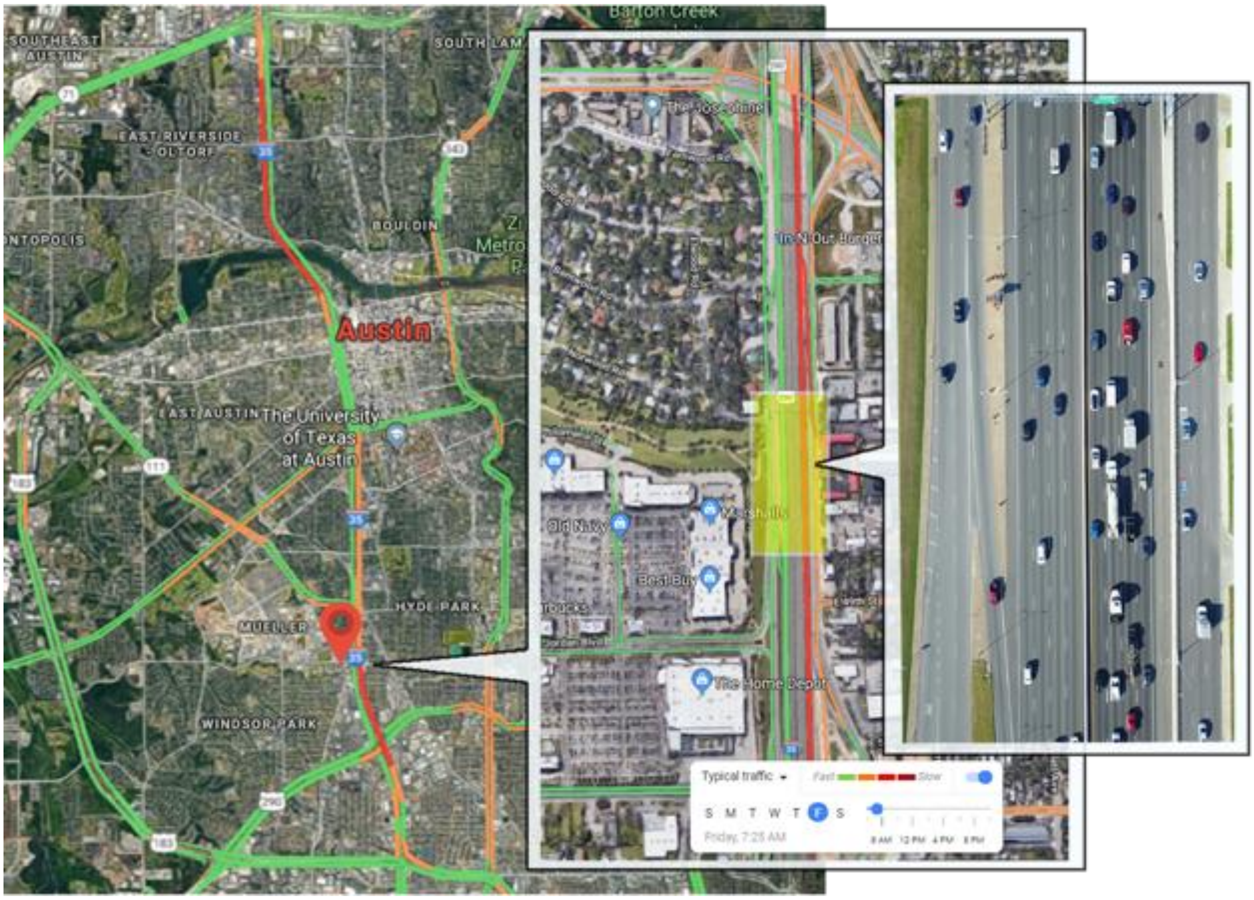
Since most of the models used were already calibrated and validated based on the NGSIM US–101 Dataset (*Next Generation Simulation: US101 Freeway Dataset* 2007), the focus of this calibration and validation effort was on the calibration of CF models of HVs to capture the effects of interacting with AVs on driver behavior. The validation effort ensures the accuracy of the calibration process.

Dataset

Vehicle trajectories are one of the cornerstones of modern traffic flow theory with applications in driver behavior studies and AV research. Unfortunately, the existing vehicle trajectory data sets are limited, mostly due to the high cost of data collection and preparation. Moreover, with the arrival of advanced driver assistance systems (ADAS) and automated vehicles, there is a potential to see changes in human driving behavior when interacting with these technologies. As a result, there is a need for new vehicle trajectory data sets that cover various levels of automation. Aerial imagery using small unmanned aerial vehicles (UAV) is an economical and effective solution to collect trajectory data.

A new trajectory data set was collected on Interstate 35 (I–35) in Austin, Texas (see Figure 14). to address the shortcomings of the existing vehicle trajectory data sets. A platoon of three SAE Level 1 AVs with adaptive cruise control (ACC) technology was circulating in the traffic stream during the data collection. Two UAVs (e.g., drones) were used for the aerial videography of the traffic stream. The trajectory of the vehicles were extracted from the video frames recorded in the bird's-eye view from a segment of the roadway (See Figure 15). In every video frame, the location of the vehicles was estimated for a fixed coordinate system and reference point on the ground. Every video recording was converted to a sequence of images (i.e., frames) separated at a constant rate over time (e.g., 25 frames per second). Tracking the location of any vehicle over the sequence of images enabled extracting the vehicle's trajectory over time.

The vehicle trajectory extraction is performed in four steps: image stabilization, vehicle detection, vehicle tracking, trajectory construction. In the image stabilization step, all the images are transformed to match a reference field of view. Then the vehicles are detected in every frame and tracked over the sequence of images. Finally, the vehicles' location and trajectories are constructed by converting the image coordinates to the adopted reference coordinates on the ground. Figure 16 shows a sample of collected vehicle trajectory data.

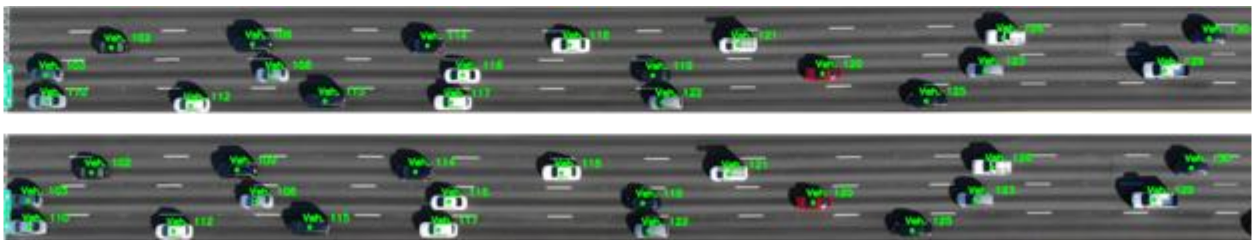


Original Photo: © 2019 Google® (See Acknowledgements).

Figure 14. Photo. Data collection location on Interstate 35 near Austin, Texas.

(a) Bird's Eye View

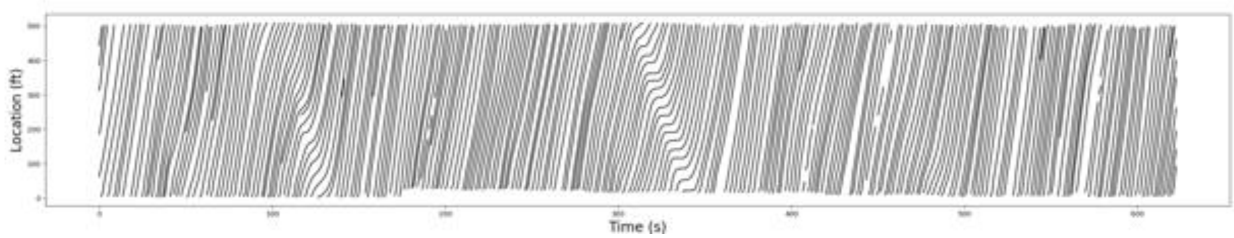
(b) Vehicle Detection Using Convolutional Neural Network



(c) Vehicle Tracking

Source: FHWA

Figure 15. Photo. Vehicle detection and tracking in aerial images.



Source: FHWA
1 ft = 0.3048 m.

Figure 16. Illustration. Sample trajectory data collected on Interstate 35 near Austin, Texas.

Calibration Approach

This study adopts the genetic algorithm calibration approach introduced by Hamdar et al. (Samer Hani Hamdar, Treiber, & Mahmassani, 2009). The approach relies on comparing the driving behavior in the data set with the simulated behavior based on a set of model parameters. For CF models, the error is calculated based on the error in the gap between the lead vehicle and the target vehicle:

$$F_{mix}[s^{sim}] = \sqrt{\frac{1}{|s^{data}|}\langle\frac{(s^{sim} - s^{data})^2}{|s^{data}|}\rangle}$$

Figure 17. Equation. Error in the gap between the lead vehicle and the target vehicle.

Where

$$\langle z \rangle = \frac{1}{\Delta T}\int_0^{\Delta T} z(t)dt$$

Figure 18. Equation. Definition of ⟨ ⟩.

, $s^{sim}(t) = x_l^{data}(t) - x^{sim}(t)$, and $s^{data}(t) = x_l^{data}(t) - x^{data}(t)$. For LC models, the same process is followed with one key difference: the error in the gap between the LC vehicle and both the new leader and new follower is considered.

Once the error function is defined, the genetic algorithm heuristic can be implemented as follows:

- The parameters of a CF/LC model are initialized to random numbers. Each set of these parameters is called a chromosome and the total of N_{GA} chromosomes will be created.
- The "fitness" of each chromosome is determined using the aforementioned error function.
- Except for the chromosome with the lowest error value, every other chromosome will be evolved through cross-over and mutation (see Hamdar et al. (2009) for the definition of cross-over and mutation in genetic algorithm).
- The process is terminated once a minimum error threshold is achieved by the best chromosome. The parameters of that chromosome will form the calibration results.

Following the procedure outlined above, an initial set of 100 parents will be initiated. These parents will produce 900 children at each iteration and the top 99 children will join the best of the parents to move to the next iteration. The calibration process stops once the error is below 5 percent or less than 0.1 percent improvement in error is observed for more than 20 consecutive iterations.

Calibration and Validation Process

The behavioral parameters of drivers in microscopic simulation models are expected to be correlated. Kim and Mahmassani (2011) presented a methodology to capture this correlation across the parameters of each driver. They showed that sampling from the empirical data while accounting for the correlation between the parameters of each sample (individual drivers) is the best method for capturing heterogeneity in microscopic simulation models. The same method was utilized in this study for the calibration of CF and LC models.

In order to calibrate and validate the model, each vehicle trajectory in the data set was divided into calibration and validation sets. The calibration set had about four times more data points than the validation set, with all selected randomly from the data points in the vehicle trajectory data set. The model was first calibrated using the data in the calibration set utilizing the same error function presented in the previous section. The calibrated model parameters were then used to simulate the data in the validation set and the results (gap between vehicles) were compared. The outcome of this calibration and validation process is a set of CF/LC parameters for each individual vehicle trajectory in the data set.

As discussed previously, the models utilized in the adopted simulation platform have gone through a similar calibration and validation process based on NGSIM US–101 dataset (2007). Accordingly, this study only focuses on the cases when a human driver interacts with an AV in the data set. The focus was on a human driver following an AV. Note that Rahmati et al. (2019) showed that there is potential to see significant changes in driver behavior in this case.

A note on calibrating and validating car-following and lane-changing models for AVs

Regarding CF models of AVs, the adopted simulation framework utilizes the ACC/CACC models that were calibrated based on empirical data (2014). Accordingly, the CF behavior of AVs will not be calibrated again in this study.

Regarding LC models of AVs, these models are designed based on the capabilities and characteristics of an automated vehicle. Accordingly, any LC trajectory generated by the models can be followed in real-world.

Calibration and Validation Results

This section presents the calibration results for the CF model of human drivers. The selected CF model is the Prospect theory model developed by Hamdar et al. (2008) and extended by Talebpour et al. (2011).

Table 2 and Table 3 show the calibration results for the Austin data along with the data collected by Rahmati et al. (2019). The details of the model are presented in the next section. The model consists of three core parameters that were calibrated in this study. w_m and γ capture drivers' different preferences when dealing with acceleration and deceleration. In other words, drivers put more negative weight on deceleration. w_c is the crash weighting factor and higher values of this parameter represent more cautious drivers.

Conducting ANOVA test (Wilcox, 1996) between values in Table 2 and Table 3 showed that two of the key model parameters show no statistically significant difference (i.e., w_m and γ), indicating that drivers' utility in response to acceleration and deceleration were the same for both cases of following an AV and following another human driver.

Table 2. Car-following model calibration results for human following.

Parameters	Mean	Standard Deviation
w_m	0.268	0.443
w_c	115200.00	21432.52
γ	0.71	0.62

Table 3. Car-following model calibration results for AV following.

Parameters	Mean	Standard Deviation
w_m	0.271	0.365
w_c	81432.83	2870.79
γ	0.69	0.61

AV = autonomous vehicle.

Unlike w_m and γ parameters, the crash weighing factor, w_c showed statistically significant difference between the two cases. Following an AV, human drivers' behavior resulted in much less w_c compared with the case of following another human driver (w_c is 36 percent less for the AV following case than the human following case). Such a significant difference shows that human drivers are more comfortable following the AV compared with another human driver and they feel safer. Such an observation on w_c can also be interpreted from the risk-taking perspective. The smooth behavior of the AV encourages more risk-taking behaviors by the following human driver, resulting in lower values of w_c.

CHAPTER 4. BASIC INFORMATION ON MODEL IMPLEMENTATION

The three modules of the framework can be incorporated in a simulation tool. The information that is transferred to the traffic monitoring at each monitoring timestep is as follows: vehicle position (lane and location in the lane), speed, acceleration, gap and speed difference between the vehicle and the leader, desired speed, relative position of the vehicle in the traffic (leader's ID, follower's ID, potential left-lane leader's ID, potential left-lane follower's ID, potential right-lane leader's ID, potential right-lane follower's ID), maximum deceleration, target exit location, and vehicle class (regular, connected, or automated). Once the input is analyzed by the three modules, the desired speed of each vehicle is updated. The new desired speed would be used in the CF and LC models in the next simulation steps. The modules are called repetitively at each monitoring timestep.

Besides the input from the simulation tool, each module involves a set of parameters. The traffic monitoring module has two major parameters that would be useful in determining the spatiotemporal area used in the Edie's generalized definitions of the traffic flow characteristics.

- Monitoring section length
- Monitoring timesteps

The predictive model settings depend on the type of SPDHRM strategy selected by the user. If the centralized method is used, the user should determine:

- Control type: Two control types are incorporated in the model, namely decision tree and speed standard deviation. The logic of the decision tree is to select suitable speed limits, according to average speed of road section. For example, the decision tree logic could flow such that speed limits are either 1) 56 mph, when average speed of the road section is larger than 46.6 mph; 2) 43.5 mph, when average speed of the road section is larger than 34 mph and is smaller than 46.6 mph; or 3) 34 mph, when average speed of the road section is less than 34 mph (Talebpour et al. 2013; Mittal et al. 2018). The speed standard deviation approach adjusts the advisory speed of a vehicle if the vehicle speed falls within a certain range of the average speed in the road section. The range is defined as the speed standard deviation in the road section multiplied by a threshold defined by the user.
- Control distance type: Three types of control distance could be specified: fixed broadcasting distance, variable distance depending on vehicle speed, or fixed broadcasting point. The fixed broadcasting distance allows the new advisory speed to be communicated to the connected vehicles that are within a specified distance upstream of the predicted congestion location. The variable distance control distance type extends the previous approach. Compared to the fixed broadcasting distance control type, this approach allows dissemination of the new speed limit to vehicles that are farther from the congestion location but possess a higher speed. The third approach sends the new advisory speed limit to the connected vehicles at a pre-specified location on the roadway. This approach is similar to the conventional method with the exception that the advisory speed limit is sent to the connected vehicles through a vehicle-to-infrastructure (V2I) communication platform.

- Compliance error: This parameter controls the level of compliance of CVs with the updated speed limit. It is assumed that the CAVs would fully comply with the advisory speed limit.

Under the decentralized SPDHRM strategy settings, the evaluation distance to be used in the congestion prediction module and the CV communication range can be specified. A control distance similar to the one defined for the centralized SPDHRM strategy is incorporated in the model.

The optimization-based SPDHRM strategy involves the user to determine three parameters: the optimization horizon, a list of potential values for advisory speed, and a list of potential values for broadcasting distance.

CHAPTER 5. USE CASE AND SENSITIVITY STUDY

IMPLEMENTATION OF THE DEVELOPED MODEL INTO A TRAFFIC SIMULATION TOOL

The objective of this case study is to conduct a proof-of-concept test of the proposed framework. The focus is on the implementation of the major components of the SPDHRM framework. The framework was incorporated into a microsimulation tool developed in Python® (Van Rossum and Drake, 1995). The microsimulation tool is a special-purpose platform for simulating mixed traffic conditions on freeways with the possibility of including connected vehicles and AVs in the system. In the current case study, the testbed uses a 3.1 mi (5 km) section of a two-lane highway with an on-ramp that starts at the 1.67 mi (2,700 m) marker of the segment and has a length of 0.19 m (300 m).

In the simulation platform, distinct CF models are defined to specify the behavior of each agent: (1) manually driven vehicles (regular vehicles), (2) CVs, and (3) AVs.

In the microsimulation platform, manually driven vehicles use the acceleration model first developed by Hamdar et al. (2008) and extended by Talebpour et al. (2011). The model was formulated based on Kahneman and Tversky's prospect theory. Two value functions, one for modeling driver behavior in congested regimes and the other for modeling driver behavior in uncongested regimes, were introduced. The following formula shows the value function for the uncongested regime:

$$U_{PT}^{UC}(a_n) = \frac{\left[w_m + (1-w_m)\left(\tanh\left(\frac{a_n}{a_0}\right)+1\right)\right]}{2}\left[\frac{\left(\frac{a_n}{a_0}\right)}{\left(1+\left(\frac{a_n}{a_0}\right)\right)^{\left(\frac{\gamma-1}{2}\right)}}\right]$$

Figure 19. Formula. Value function for the uncongested regime.

Where U_{PT}^{UC} denotes the value function for the uncongested traffic conditions. $\gamma > 0$ and w_m are parameters to be estimated and calibrated and $a_0 = 3.28\ ft/s^2 (1\ m/s^2)$ is used to normalize the acceleration. On the other hand, the following formula shows the value function for the congested regime:

$$U_{PT}^{C}(a_n) = \frac{\left\lfloor w'_m + (1-w'_m)\left(\tanh\left(\frac{a_n}{a_0}\right)+1\right)\right\rfloor}{2}\left(\frac{a_n}{a_0}\right)^{\gamma'}$$

Figure 20. Formula. Value function for the congested regime.

Where U_{PT}^{C} denotes the value function for the congested traffic conditions. $\gamma' > 0$ and w'_m are parameters to be estimated and calibrated. At each evaluation timestep, the driver evaluates the gain from a candidate acceleration selected from a feasible set of values. The surrounding traffic condition is taken into consideration by the driver throughout the acceleration evaluation process. The driver utilizes the following binary probabilistic regime selection model to evaluate each acceleration value:

$$U_{PT}(a_n) = P(C).U_{PT}^{C} + P(UC).U_{PT}^{UC}$$

Figure 21. Formula. Binary probabilistic regime selection model.

Where U_{PT}, $P(C)$, and $P(UC)$ denote the expected value function, the probability of driving in a congested traffic condition, and the probability of driving in uncongested traffic condition, respectively. After calculating the expected value function, the total utility function for acceleration could be written as follows:

$$U(a_n) = (1 - p_{n,i})U_{PT}(a_n) - p_{n,i} w_c k(v, \Delta v)$$

Figure 22. Formula. Total utility function for the choice of acceleration.

Where $p_{n,i}$ is the crash probability. Finally, the following probability density function is used to evaluate the stochastic response of the drivers:

$$f(a_n) = \begin{cases} \dfrac{e^{\beta_{PT} U(a_n)}}{\int_{a_{\min}}^{a_{\max}} e^{\beta_{PT} U(a')}\, da'} & a_{\min} < a_n < a_{\max} \\ 0 & Otherwise \end{cases}$$

Figure 23. Formula. Probability density function for the evaluation of drivers' stochastic response.

Where β_{PT} is the sensitivity of choice to the utility $U(a_n)$.

CVs are capable of exchanging information with other vehicles and infrastructure-based equipment. The information is exchanged through the vehicle-to-vehicle (V2V) and V2I communications networks. As a result, the driver receives information about other connected vehicles as well as updated information containing traffic management center (TMC) decisions (e.g., real-time changes in speed limit). The drivers' behavior may change based on the information conveyed to the driver. The reliability and the frequency of the information received by the driver plays a significant role in the drivers' behavior and on the overall performance of the traffic network.

An active V2V communication network allows the drivers to be aware of other drivers' behavior, the driving environment, road condition, and weather condition. As a result, the driving behavior could be modeled using a deterministic acceleration modeling framework. The simulation tool utilizes the Intelligent Driver Model (IDM) to model this connected environment. Because the IDM is able to capture various congestion dynamics and provides greater realism than most of the deterministic acceleration modeling frameworks.

The acceleration model specified by the IDM entails the vehicle's current speed, the ratio of the current spacing to the desired spacing, the difference between the leading and the following vehicles' velocities, and subjective parameters such as desired acceleration, desired gap size, and comfortable deceleration.

$$a_{IDM}^{n}(s_n, v_n, \Delta v_n) = \bar{a}_n \left[1 - \left(\frac{v_n}{v_0^n} \right)^{\delta_n} - \left(\frac{s^*(v_n, \Delta v_n)}{s_n} \right)^2 \right]$$

(a) The acceleration in the intelligent driver acceleration model.

$$s^*(v_n, \Delta v_n) = s_0^n + T_n v_n + \frac{v_n \Delta v_n}{2\sqrt{\bar{a}_n \bar{b}_n}}$$

(b) The effective safe gap in the intelligent driver acceleration model.

Figure 24. Formula. The intelligent driver acceleration model.

Where δ_n is the free acceleration exponent; T_n is the desired time gap; a_n is the maximum acceleration; b_n is the desired deceleration; $s_0{}^n$ is the jam distance; and $v_0{}^n$ is the desired speed. These parameters could be calibrated to better capture the behavior of connected vehicles.

If the V2V communication network is inactive, the driving behavior of connected vehicles would be similar to that of isolated manually driven vehicles. In the presence of V2I communications, the TMC decisions, such as the speed limits in the case of SPDHRM, could be transferred to the drivers. However, their reaction times would still be like regular drivers.

AVs can continuously monitor other vehicles in their vicinity, which results in a deterministic behavior in interacting with other drivers. Furthermore, they can quickly react to any perturbations in the driving environment. Therefore, the CF behavior of AVs could be specified by a deterministic modeling framework. Talebpour and Mahmassani (2016) developed a CF model for AVs based on the previous simulation studies by Van Arem et al. (2006) and Reece and Shafer (1993). They simulated similar individual sensors installed on the AVs in order to generate the input data for the acceleration model.

Considering the sensor range and limitations in accuracy, the AVs can be ready to react to any situation outside of their sensing range once it is detected (e.g., a vehicle at a complete stop right outside of the sensors detection range). Furthermore, if a leader is spotted, the speed of the AV could be adjusted in a way that allows it to stop if the leader decides to decelerate with its maximum deceleration rate and reach a full stop. Considering different situations that involves immediate reaction of the AV, the maximum safe speed can be calculated using the following equations:

$$\Delta x_n = (x_{n-1} - x_n - l_{n-1}) + v_n \tau_n + \frac{v_{n-1}^2}{2a_{n-1}^{decc}}$$

(a) relative position of vehicle n (with respect to the leader)

$$\Delta x = min(\,SensorDetectionRange, \Delta x)$$

(b) relative position of vehicle n (with respect to the leader and the sensor detection range)

$$v_{max} = \sqrt{-2a_i^{decc}\,\Delta x}$$

(b) Maximum safe speed for autonomous vehicles.

Figure 25. Formula. Maximum speed of autonomous vehicles.

Where *n* and *n-1* represent the AV and its leader, respectively; x_n is the position of vehicle *n*; l_n is the length of vehicle *n*; v_n is the speed of vehicle *n*; τ_n is the reaction time of vehicle *n*; and a_n^{decc} is the maximum deceleration of vehicle *n*.

Besides the safety constraint, the following formula, adopted from the model proposed by Van Arem et al. (2006), updates the acceleration of the AV at every decision point:

$$a_n^d(t) = k_a a_{n-1}(t-\tau) + k_v(v_{n-1}(t-\tau) - v_n(t-\tau)) + k_d(s_n(t-\tau) - s_{ref})$$

Figure 26. Formula. Acceleration model for automated vehicles.

Where a_n^d is the acceleration of vehicle *n*; and k_a, k_v, and k_d are model parameters that could be calibrated. s_n is the spacing and s_{ref} is the maximum between the minimum distance (s_{min}), following distance based on the reaction time (s_{system}), and safe following distance (s_{safe}). In this study, the minimum distance is set at *6.56 ft (2 m)* and s_{safe} is calculated according to the following formula:

$$s_{safe} = \frac{v_{n-1}^2}{2}\left(\frac{1}{a_n^{decc}} - \frac{1}{a_{n-1}^{decc}}\right)$$

Figure 27. Formula. Safe following distance formula.

Finally, the acceleration of the AV can be calculated using the following equation:

$$a_n(t) = min\left(a_n^d(t), k\big(v_{max} - v_n(t)\big)\right)$$

Figure 28. Formula. Acceleration of autonomous vehicles.

Where *k* is a model parameter. Van Arem et al. (2006) suggested using the following values for the model parameters: $k=1$, $k_a=1$, $k_v=0.58$, and $k_a=0.1$.

DESIGN OF SIMULATION EXPERIMENTS

Two sets of scenarios for the sensitivity analysis were defined. Various market penetration rates of CVs and CAVs are considered in the scenarios. It is assumed that all vehicles are equipped with communication features and as a result would be able to interact with the traffic monitoring, congestion prediction, and speed control modules of the simulation tool. The first set of scenarios is used to evaluate the accuracy of the congestion prediction model. The second set of scenarios are used to compare the effectiveness of various SPDHRM strategies with different parameters. The scenarios are shown in Table 4 and

Table 5. Each scenario was simulated 10 times to incorporate the randomness in the input data such as the lane assigned to a vehicle at the entry point, the time at which the vehicle enters the simulated environment, the vehicle initial speed, and the vehicle class (CV or CAV), etc.

Table 4. Scenarios for evaluating accuracy of congestion prediction models.

Scenario ID	MPR of CV (%)	MPR of AV (%)	Monitoring Timestep (s)	Section length (ft)
A1	100	0	10	330
A2	70	30	10	330
A3	30	70	10	330
A4	0	100	10	330
A5	100	0	10	660
A6	70	30	10	660
A7	30	70	10	660
A8	0	100	10	660
A9	100	0	10	1640
A10	70	30	10	1640
A11	30	70	10	1640
A12	0	100	10	1640
A13	100	0	20	330
A14	70	30	20	330
A15	30	70	20	330
A16	0	100	20	330
A17	100	0	20	660
A18	70	30	20	660
A19	30	70	20	660
A20	0	100	20	660
A21	100	0	20	1640
A22	70	30	20	1640
A23	30	70	20	1640
A24	0	100	20	1640
A25	100	0	30	330
A26	70	30	30	330
A27	30	70	30	330
A28	0	100	30	330
A29	100	0	30	660
A30	70	30	30	660
A31	30	70	30	660
A32	0	100	30	660
A33	100	0	30	1640
A34	70	30	30	1640
A35	30	70	30	1640
A36	0	100	30	1640

% = percent. CAV = connected automated vehicle. CV = connected vehicle. ft = feet. MPR = market penetration rate. s = seconds.

Table 5. Scenarios for evaluating effectiveness of the speed harmonization strategies.

Scenario ID	MPR of CV (%)	MPR of AV (%)	Speed Control Strategy	Broadcasting distance (ft)
B0*	0	0	NA	NA
B1	100	0	Centralized	3280
B2	70	30	Centralized	3280
B3	30	70	Centralized	3280
B4	0	100	Centralized	3280
B5	100	0	Centralized	1640
B6	70	30	Centralized	1640
B7	30	70	Centralized	1640
B8	0	100	Centralized	1640
B9	100	0	Decentralized	Sensor Range**
B10	70	30	Decentralized	Sensor Range
B11	30	70	Decentralized	Sensor Range
B12	0	100	Decentralized	Sensor Range
B13	100	0	Optimization based	VA
B14	70	30	Optimization based	VA
B15	30	70	Optimization based	VA
B16	0	100	Optimization based	VA

* This is the base case scenario where the SPDHRM algorithm is deactivated and there are only manually driven vehicles (without connectivity features) in the system.
** The sensor range is assumed to be 500 ft (150 m)
% = percent. CAV = connected automated vehicle. CV = connected vehicle. ft = feet. MPR = market penetration rate. VA = variable. NA = not applicable.

SIMULATION RESULTS FOR THE DIFFERENT SCENARIOS

In order to train and test the congestion prediction model, the status of each section of the highway was determined based on the travel time index. Then, the random forest-based model was developed by training the model with 80 percent of the simulation data. The model performance was evaluated by comparing the status predicted by the model and the status determined by TTI on the remaining 20 percent of the data. Table 6 shows the result of analyzing the first set of scenarios. The average and standard deviation of the prediction model accuracy was calculated for each pair of monitoring timestep and section length. The accuracy is reported for the entire flow, the congested instances, and the uncongested instances. As the goal of the first sensitivity analysis is to determine the best pair of monitoring timestep and section length, the average accuracy was calculated for the four different scenario types created by changing the market penetration rate of CVs and CAVs. It is assumed that all the vehicles are equipped with connectivity features. Therefore, there are minor differences between the accuracy values among various monitoring timestep and section length pairs. The source of difference is the CF model used for each vehicle class. Among the analyzed scenarios, the scenario with a monitoring timestep of 20 s and section length of 660 ft (200 m) has a slightly better performance (higher accuracy and lower standard deviation in accuracy) compared to the other scenarios. Analyzing a more comprehensive set of scenarios with manually driven vehicles that do not possess connectivity features could determine the most suitable parameter combination that accurately predicts the traffic status.

Table 6. Accuracy result of the congestion prediction model under different monitoring timestep and section length scenarios.

Monitoring Timestep (s)	Section length (ft)	Average overall accuracy (%)	Average congested accuracy (%)	Average uncongested accuracy (%)	Standard deviation overall accuracy (%)	Standard deviation congested	Standard deviation uncongested accuracy (%)
10	330	97.14	89.56	98.96	0.11	0.40	0.08
10	660	97.15	89.80	98.93	0.15	0.51	0.10
10	1640	97.14	89.63	98.94	0.09	0.50	0.09
20	330	97.15	89.71	98.93	0.11	0.57	0.08
20	660	97.19	89.94	98.94	0.09	0.38	0.07
20	1640	97.14	89.75	98.93	0.10	0.43	0.10
30	330	97.19	89.85	98.95	0.10	0.49	0.09
30	660	97.12	89.60	98.93	0.11	0.49	0.12
30	1640	97.17	89.88	98.91	0.09	0.42	0.07

% = percent. ft = feet. s = seconds

The selected pair from the previous sensitivity analysis was incorporated into the speed control module to compare the effectiveness of various SPDHRM strategies.

Table 7 shows the result of scenarios involving different speed control strategies and various market penetration rates of CVs and CAVs. The scenarios where no SPDHRM strategy has been applied are used as a reference to assess the effectiveness of the examined speed control strategies. As can be seen, the effect of CAVs in the full automation scenario dominates the effect of the speed control strategies. As a result, applying the speed control module does not change the performance metrics of the vehicles on the road. As expected, the optimization-based control system is more successful than the centralized and decentralized control systems due to the decreased speed variation and the increased reliability in travel time (lower travel time standard deviation). Although these cases show a slight decrease in the average speed and a marginal increase in the average travel time, the strategies were able to prevent or postpone formation of shockwaves in the system.

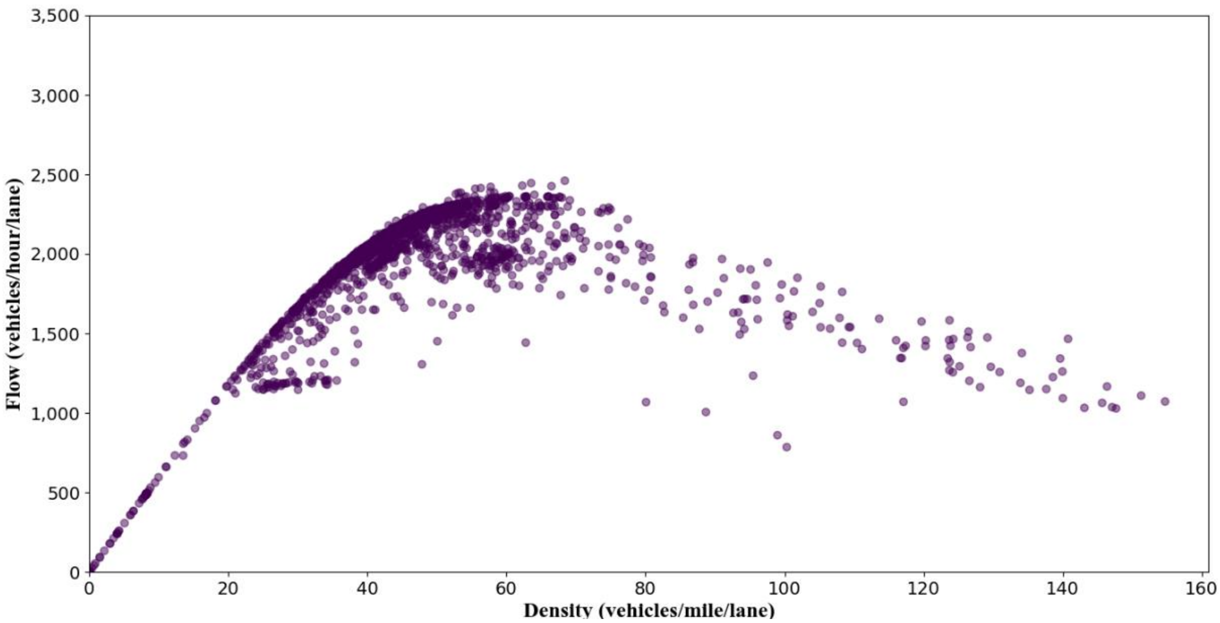
Table 7. Performance metrics of different speed-control strategies along with various market shares of connected vehicles and automated vehicles.

Speed Control Strategy	Broadcasting distance (ft)	MPR of CV (%)	MPR of AV (%)	Average speed (mi/h)	Speed standard deviation (mi/h)	Average travel time (s)	Travel time standard deviation (s)
No SPDHRM	NA	0	0	44.36	13.21	251.88	31.52
No SPDHRM	NA	0	100	62.08 (39.93)*	1.19 (-90.98)	179.07 (-28.91)	5.46 (-82.67)
No SPDHRM	NA	30	70	57.99 (30.70)	9.22 (-30.22)	176.05 (-30.11)	50.66 (60.73)
No SPDHRM	NA	70	30	40.60 (-8.48)	16.80 (27.17)	182.72 (-27.46)	136.24 (332.24)
No SPDHRM	NA	100	0	41.13 (-7.30)	15.59 (17.99)	263.20 (4.49)	42.65 (35.32)
Centralized	3280	0	100	62.08 (39.93)	1.19 (-90.98)	179.07 (-28.91)	5.46 (-82.67)
Centralized	3280	30	70	57.59 (29.81)	9.51 (-28.04)	177.46 (-29.55)	50.19 (59.24)
Centralized	3280	70	30	40.66 (-8.35)	16.96 (28.41)	180.89 (-28.19)	135.24 (329.07)
Centralized	3280	100	0	40.46 (-8.80)	16.20 (22.67)	252.48 (.24)	47.90 (51.98)
Centralized	1640	0	100	62.08 (39.93)	1.19 (-90.98)	179.07 (-28.91)	5.46 (-82.67)
Centralized	1640	30	70	58.13 (31.03)	8.84 (-33.08)	176.86 (-29.79)	46.78 (48.42)
Centralized	1640	70	30	41.51 (-6.44)	16.80 (27.14)	176.48 (-29.93)	132.21 (319.45)
Centralized	1640	100	0	43.38 (-2.21)	13.34 (.95)	254.44 (1.02)	31.52 (.01)
Decentralized	Sensor Range**	0	100	62.08 (39.93)	1.19 (-90.98)	179.07 (-28.91)	5.46 (-82.67)
Decentralized	Sensor Range	30	70	56.83 (28.10)	10.65 (-19.41)	177.72 (-29.44)	53.71 (70.40)
Decentralized	Sensor Range	70	30	41.74 (-5.91)	14.71 (11.36)	176.97 (-29.74)	133.16 (322.46)

Speed Control Strategy	Broadcasting distance (ft)	MPR of CV (%)	MPR of AV (%)	Average speed (mi/h)	Speed standard deviation (mi/h)	Average travel time (s)	Travel time standard deviation (s)
Decentralized	Sensor Range	100	0	41.59 (-6.25)	14.81 (12.14)	267.07 (6.03)	33.63 (6.68)
Optimization-based	VA	0	100	62.08 (39.93)	1.19 (-90.98)	179.07 (-28.91)	5.46 (-82.67)
Optimization-based	VA	30	70	57.94 (30.59)	9.14 (-30.80)	172.11 (-31.67)	56.88 (80.46)
Optimization-based	VA	70	30	41.79 (-5.81)	16.35 (23.80)	178.44 (-29.16)	132.72 (321.06)
Optimization-based	VA	100	0	39.99 (-9.86)	16.66 (26.13)	245.80 (-2.41)	69.26 (119.73)

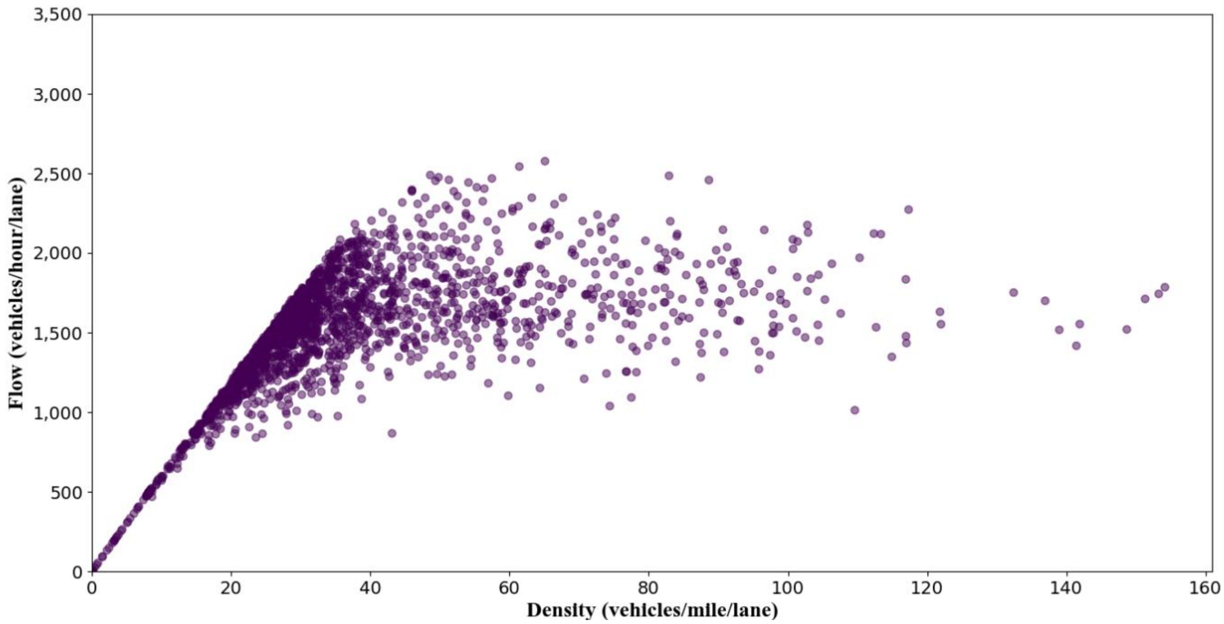
* Numbers in parentheses represent percent changes in the performance metric of a scenario compared to the base case scenario where the SPDHRM algorithm is deactivated and there are only manually driven vehicles (without connectivity features) in the system.
** The sensor range is assumed to be 500 ft.
% = percent. CAV = connected automated vehicle. CV = connected vehicle. ft = feet. s = seconds. SPDHRM = speed harmonization. NA = not applicable. VA = variable. mi = miles. h = hour.

Figure 29-Figure 33 show the fundamental diagrams for the base case (manually driven vehicles without connectivity features) and for different SPDHRM strategies when CVs constitute 70 percent of the traffic and 30 percent of the vehicles are CAVs. Comparing the maximum y-axis values of the diagrams shows that applying any of the SPDHRM strategies would improve the system capacity. Although activating the centralized SPDHRM strategy benefits the system based on the performance metrics represented in Table 7, higher number of points are observed in high density values. Due to the higher number of traffic slowdowns and the higher spread in the data points, the centralized SPDHRM is not as efficient as the other SPDHRM strategies. The decentralized and optimization-based SPDHRM strategies were able to improve the system performance while decreasing the maximum density. Furthermore, the traffic under these two cases experiences higher stability due to the smoother transition from the uncongested state to the congested states.



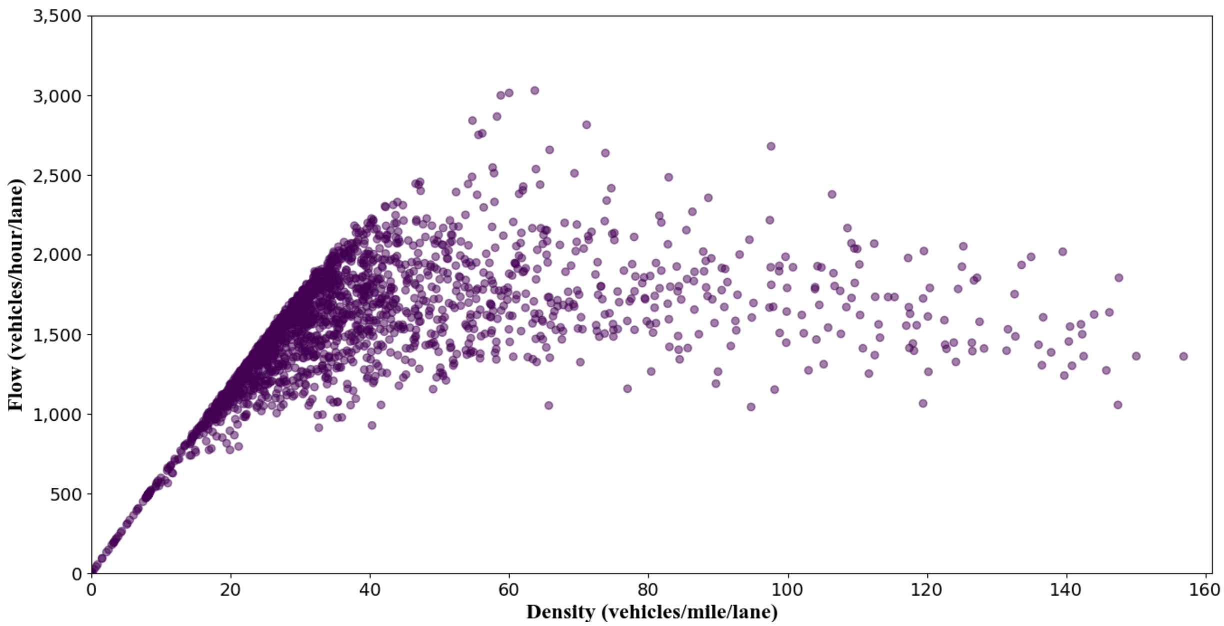
Source: FHWA.

Figure 29. Diagram. Fundamental diagram of the base case [100% market penetration rate of manually driven vehicles].



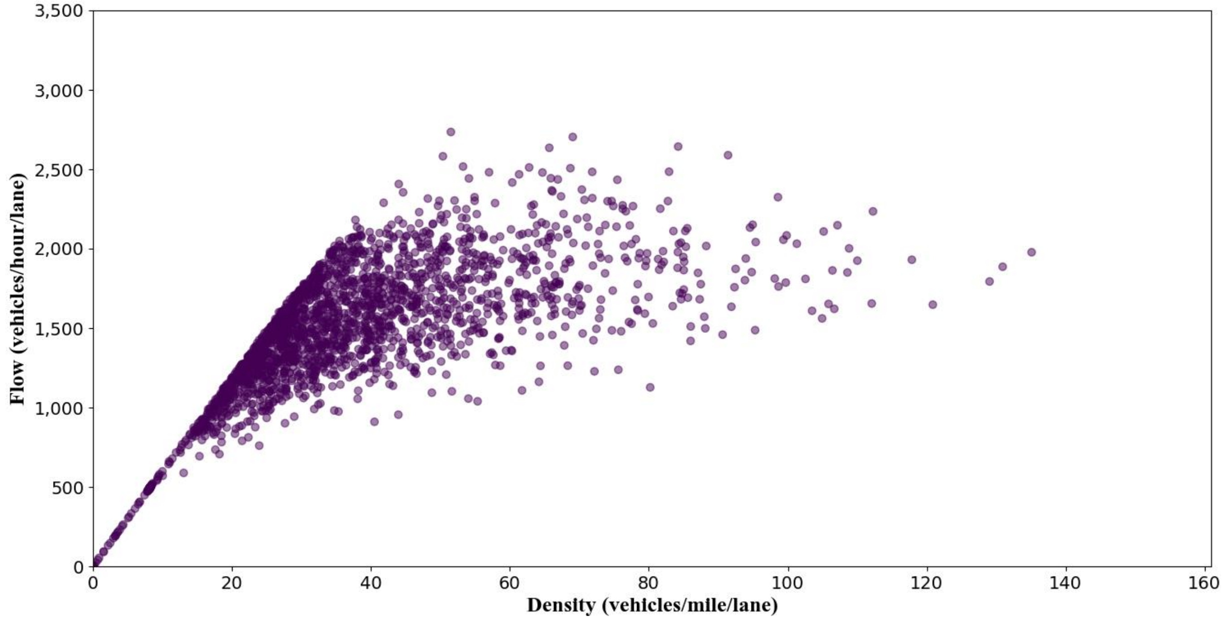
Source: FHWA

Figure 30. Diagram. Fundamental diagram of the use case without any SPDHRM strategy [70% market penetration rate of CVs and 30% market penetration rate of CAVs].



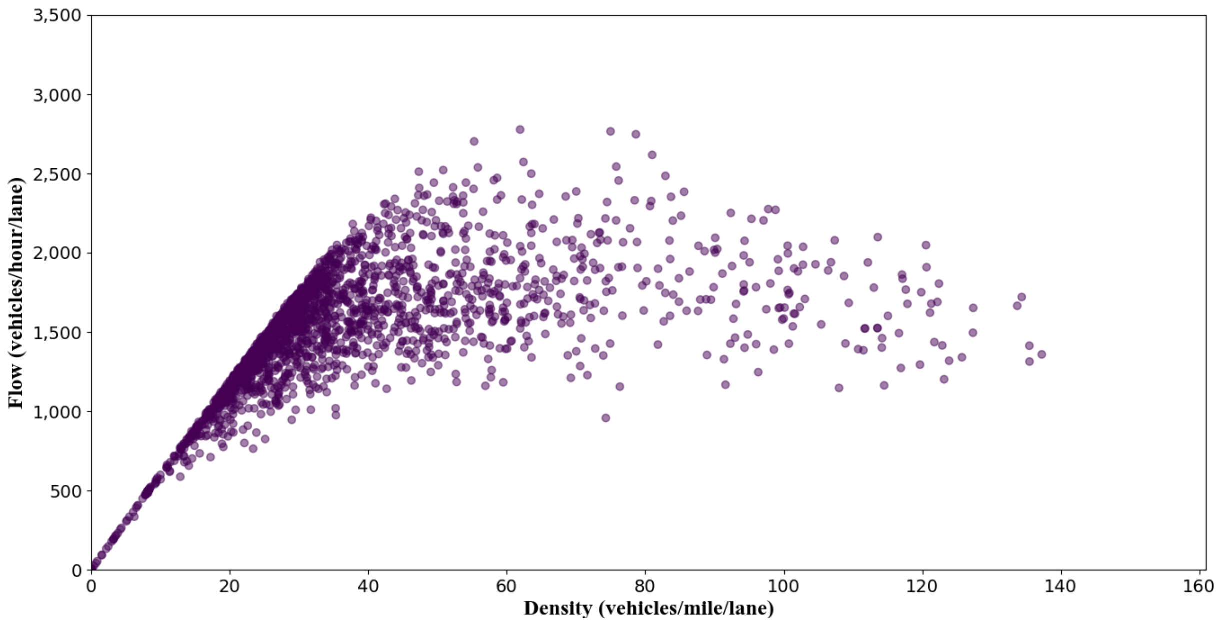
Source: FHWA

Figure 31. Diagram. Fundamental diagram of the use case with the centralized SPDHRM strategy (broadcasting distance = 3280 ft) [70% market penetration rate of CVs and 30% market penetration rate of CAVs].



Source: FHWA

Figure 32. Diagram. Fundamental diagram of the use case with the decentralized SPDHRM strategy [70% market penetration rate of CVs and 30% market penetration rate of CAVs].



Source: FHWA

Figure 33. Diagram. Fundamental diagram of the use case with the optimization-based SPDHRM strategy [70% market penetration rate of CVs and 30% market penetration rate of CAVs].

Since, the congestion prediction model is independent of the speed control module, performing a bi-level sensitivity analysis such as the one described could significantly decrease the number of scenarios that could be analyzed to reach a comprehensive conclusion. As a next step, scenarios with manually driven vehicles will be incorporated vehicle to accommodate different scenarios with partial connectivity.

CHAPTER 6. SUMMARY AND RECOMMENDATIONS

The objective of this study is to develop innovative traffic management strategies that utilize the big stream of data generated by CAV systems and the predictive capability of machine learning algorithms. A methodological framework was proposed for developing predictive traffic management and control strategies utilizing CAV systems. The framework consists of three main components: 1) traffic monitoring, 2) traffic state prediction, and) control strategy. The traffic monitoring component describes how the detailed vehicle trajectories broadcasted by CAVs can be used to estimate traffic properties and track traffic shockwaves without relying on road sensors. The traffic state prediction component describes how the traffic properties estimated through CAVs can be used to predict future traffic states. The congestion prediction models have various safety and traffic performance applications. For instance, the models can be used to warn drivers ahead of traffic slowdowns to prevent potential accidents. In terms of traffic operations, the models can be integrated into traffic control algorithms to enhance their performance. This module uses machine learning algorithms to develop models that consider traffic flow dynamics in order to reliably predict the traffic state under different scenarios. Finally, the control strategy component describes announcement of new speed limits as a control action that can be executed through CAVs.

A case study that involves a mixture of connected and automated vehicles are analyzed by a simulation tool that possesses the SPDHRM framework. The details of how to incorporate the proposed framework into a simulation tool are discussed. The flowcharts of the various modules of the framework as well as the pseudocodes in the appendix provide useful information for a successful implementation of the framework in traffic analysis tools. The results of the case study show that the proposed SPDHRM system can reduce the severity and lengths of traffic shockwaves, and improve the overall traffic stability. In the case study, the traffic stabilization effect of automated vehicles dominates the effect of the speed control strategies. The system performance analysis could be further analyzed in a partially connected environment. Partial connectivity not only influences the accuracy of congestion prediction models but also the effectiveness of the speed control module.

Three main factors acting jointly or separately trigger traffic breakdown: 1) high traffic load, 2) bottlenecks, and 3) disturbances caused by individual vehicles. High traffic loads occur when traffic demand exceeds the sustainable throughput of a road section. Capacity reductions or “bottlenecks” may be permanent, such as on-ramps and off-ramps, or temporary such as traffic accidents or slow-moving vehicles. As for traffic disturbances, those refer to temporary perturbations in the traffic flow. The proposed predictive SPDHRM strategies focus on the first factor of the three.

Another approach to reducing the likelihood of a traffic breakdown that can be developed in future work would focus on minimizing the disturbances caused by individual vehicles. Traffic perturbations can be caused by lane-change maneuvers, abrupt braking, speeding, or long-lasting overtaking maneuvers of trucks. A traffic disturbance minimization strategy could also utilize the information broadcasted by CAVs to estimate the speed standard deviation of traffic in real-time. Instead of predicting traffic congestion, however, the new strategy would predict future traffic disturbances for all road segments by predicting its proxy, the traffic speed standard deviation. Once disturbances are predicted to occur, the strategy would identify the vehicles that are likely to cause those traffic disturbances by, for example, measuring their speeds and acceleration relative to other vehicles (aggressiveness), and send out advisory speed to those vehicles in order to prevent or minimize potential disturbances.

Another area to explore in future work is integrating the developed SPDHRM strategies with traffic disturbance minimization strategies and/or other active traffic control strategies such as queue or collision warning. Studies have shown that implementing a combination of active strategies, such as SPDHRM and ramp metering can outperform the implementation of individual strategies (Laval, Guin, Chilukuri, & Cho, 2019; Chen & Ahn, 2015; Khondaker & Kattan, 2015).

ACKNOWLEDGMENTS

The original map in Figure 14 has been modified and is the copyright property of Google® Maps™ and can be accessed from https://www.google.com/maps/.(#) The map overlays show the data collection location on Interstate 35 near Austin, TX.

REFERENCES

Dong, J., & Mahmassani, H. (2009). Flow breakdown and travel time reliability. *Transportation Research Record: Journal of the Transportation Research Board*(2124), 203-212.

Elfar, A. (2019). *Predictive Traffic Operations and Control of Connected and Automated Vehicle Systems.* Northwestern University,

Elfar, A., Talebpour, A., & Mahmassani, H. S. (2018). Machine learning approach to short-term traffic congestion prediction in a connected environment. *Transportation Research Record, 2672*(45), 185-195.

Elfar, A., Talebpour, A., & Mahmassani, H. S. (2019). *Predictive Speed Harmonization in a Connected Environment: A Machine Learning Approach*. Paper presented at the Transportation Research Board Annual Meeting, Washington D.C.

Elfar, A., Talebpour, A., & Mahmassani, H. S. (2020). *Predictive Speed Harmonization in Traffic Flow with Connected and Automated Vehicles: Machine Learning Application*. Paper presented at the Transportation Research Board Annual Meeting, Washington D.C.

Elfar, A., Xavier, C., Talebpour, A., & Mahmassani, H. S. (2018). Traffic shockwave detection in a connected environment using the speed distribution of individual vehicles. *Transportation Research Record, 2672*(20), 203-214.

Hamdar, S. H., Treiber, M., & Mahmassani, H. S. (2009). *Calibration of a stochastic car-following model using trajectory data: exploration and model properties*. Retrieved from

Hamdar, S. H., Treiber, M., Mahmassani, H. S., & Kesting, A. (2008). Modeling driver behavior as sequential risk-taking task. *Transportation Research Record, 2088*(1), 208-217.

Hartigan, J. A. (1975). *Clustering algorithms.* John Wiley & Sons Inc.

(2007). *Next Generation Simulation: US101 Freeway Dataset Federal Highway Administration.* Washington, D.C.

Kim, J., & Mahmassani, H. S. (2011). Correlated parameters in driving behavior models: Car-following example and implications for traffic microsimulation. *Transportation Research Record, 2249*(1), 62-77.

Li, X., Cui, J., An, S., & Parsafard, M. (2014). Stop-and-go traffic analysis: Theoretical properties, environmental impacts and oscillation mitigation. *Transportation Research Part B: Methodological, 70*, 319-339.

Lu, X., Liu, H., Li, X., Li, Q., Mahmassani, H., Talebpour, A., Hosseini, M., Huang, Z., Hale, D. K., and Shladover, S. E. (Forthcoming). Developing Analysis, Modeling, and Simulation Tools for Connected and Automated Vehicle Applications. FHWA.

Mahmassani, H., Rakha, H., Hubbard, E., & Lukasik, D. (2012). Concept development and needs identification for Intelligent Network Flow Optimization (INFLO): assessment of relevant prior and ongoing research.

Mahmassani, H. S. (2016). 50th anniversary invited article—autonomous vehicles and connected vehicle systems: Flow and operations considerations. *Transportation Science, 50*(4), 1140-1162.

Mahmassani, H. S., Elfar, A., Shladover, S. E., & Huang, Z. (2018). *Development of an Analysis/Modeling/Simulation (AMS) Framework for V2I and Connected/Automated Vehicle Environment*. Retrieved from

Milanés, V., & Shladover, S. E. (2014). Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transportation Research Part C: Emerging Technologies, 48*, 285-300.

Mittal, A., Kim, E., Mahmassani, H. S., & Hong, Z. (2018). Predictive dynamic speed limit in a connected environment for a weather affected traffic network: a case study of Chicago. *Transportation Research Record, 2672*(19), 13-24.

Next Generation Simulation: US101 Freeway Dataset (2007).

Pattara-Atikom, W., & Peachavanish, R. (2007). *Estimating road traffic congestion from cell dwell time using neural network.* Paper presented at the 2007 7th International Conference on ITS Telecommunications.

Rahmati, Y., Khajeh Hosseini, M., Talebpour, A., Swain, B., & Nelson, C. (2019). Influence of Autonomous Vehicles on Car-Following Behavior of Human Drivers. *Transportation Research Record*, 0361198119862628.

Reece, D. A., & Shafer, S. A. (1993). A computational model of driving for autonomous vehicles. *Transportation Research Part A: Policy and Practice, 27*(1), 23-50.

Talebpour, A., & Mahmassani, H. S. (2016). Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies, 71*, 143-163.

Talebpour, A., Mahmassani, H. S., & Hamdar, S. H. (2011). Multiregime sequential risk-taking model of car-following behavior: specification, calibration, and sensitivity analysis. *Transportation Research Record, 2260*(1), 60-66.

Talebpour, A., Mahmassani, H. S., & Hamdar, S. H. (2013). Speed harmonization: evaluation of effectiveness under congested conditions. *Transportation Research Record, 2391*(1), 69-79.

Texas Transportation Institute. (2006). *Travel Time Reliability: Making It There On Time, All The Time.* Federal Highway Administration.

Thianniwet, T., Phosaard, S., & Pattara-Atikom, W. (2009). *Classification of road traffic congestion levels from GPS data using a decision tree algorithm and sliding windows.* Paper presented at the Proceedings of the world congress on engineering.

Van Arem, B., Van Driel, C. J., & Visser, R. (2006). The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems, 7*(4), 429-436.

Van Rossum, G. and Jr, F.L. Drake, 1995. "Python tutorial (Vol. 620)". *Amsterdam: Centrum voor Wiskunde en Informatica.*

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2008). Temporal evolution of short-term urban traffic flow: a nonlinear dynamics approach. *Computer-Aided Civil and Infrastructure Engineering*

23(7), 536-548.

Wang, M., Daamen, W., Hoogendoorn, S. P., & Van Arem, B. (2015). *Connected variable speed limits control and vehicle acceleration control to resolve moving jams.* Paper presented at the 94th Annual Meeting Transportation Research Board, Washington, USA, 11-15 January 2015; Authors version.

Chen, D., & Ahn, S. (2015). Variable speed limit control for severe non-recurrent freeway bottlenecks. *Transportation Research Part C: Emerging Technologies, 51*, 210-230.

Hartigan, J. A. (1975). *Clustering algorithms.* John Wiley & Sons Inc.

Khondaker, B., & Kattan, L. (2015). Variable speed limit: A microscopic analysis in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies, 58*, 146-159.

Laval, J. A., Guin, A., Chilukuri, B. R., & Cho, H. W. (2019). *Travel-time Optimization on I-285 with Improved Variable Speed Limit Algorithms and Coordination with Ramp Metering Operations (No. FHWA-GA-19-1414).* Georgia Institute of Technology. School of Civil and Environmental Engineering.

(2007). *Next Generation Simulation: US101 Freeway Dataset Federal Highway Administration.* Washington, D.C.

Texas Transportation Institute. (2006). *Travel Time Reliability: Making It There On Time, All The Time.* Federal Highway Administration.

Wilcox, R. R. (1996). *Statistics for the social sciences.* Academic Press.

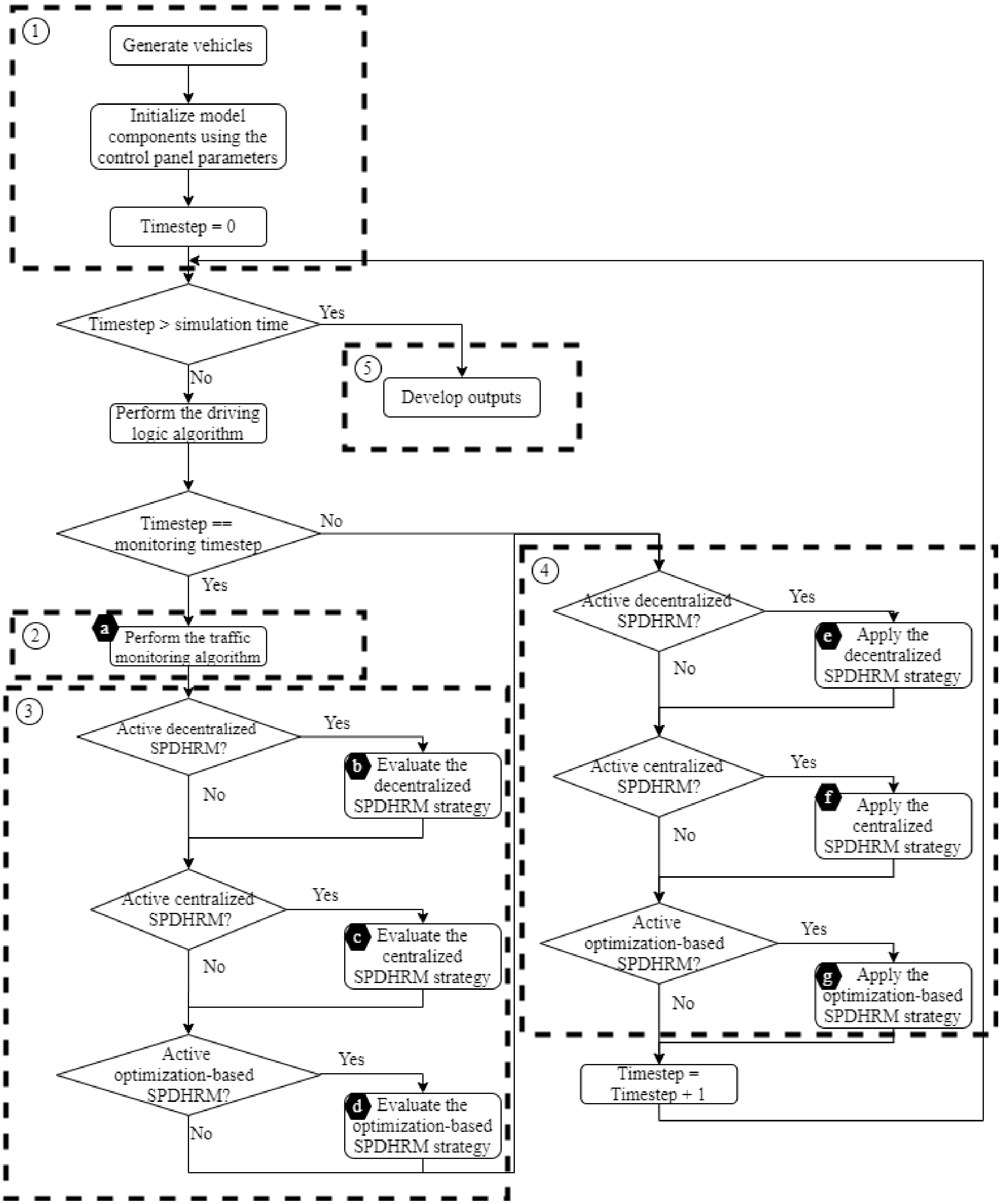
Yang, H., & Jin, W.-L. (2014). A control theoretic formulation of green driving strategies based on inter-vehicle communications. *Transportation Research Part C: Emerging Technologies, 41*, 48-60.

APPENDIX

This section gives detailed descriptions and simulation functions used by the three modules of the speed harmonization framework. The functions are arranged in alphabetical order so that the users can easily locate the function description with the function name. Each function item provides the function syntax, functionality description, input and output definitions, sub-functions, and pseudo code. The information can help users understand how the functions are implemented in the speed harmonization framework. This is essential for developing similar algorithms in different simulation platforms. In the pseudo code section, this document describes critical decision and computation processes involved in each function, while omitting many auxiliary steps.

A.1. Main Algorithm

This function controls the implementation process of the simulation logic. The logic flow of this function is shown in Figure A1.



Source: FHWA

Figure A1. Flowchart. Main algorithm.

main

Syntax

main

Description

This function contains four components. The first component corresponds to the "Freeway Segment" element of the framework. It includes the inputs, outputs, and the driving logic of the tool. The driving logic contains the car-following and lane-changing models that specify the interaction among vehicles. The "Traffic Monitoring" module of the framework relates to the second element of the algorithm. Then, as the "Congestion Prediction" module (third element of the algorithm), the model predicts the congestion characteristics and evaluates the speed harmonization strategy selected by the user (decentralized, centralized, or optimization-based). The fourth element in the algorithm is related to the "Speed Control" module of the framework. This part of the algorithm implements the speed harmonization strategy by determining the advisory speed for each vehicle.

Input Arguments

Importing the generated vehicles array and the control panel. The generated vehicles array contains the following information: vehicle initial position (lane and location in the lane), initial speed, initial acceleration, gap and speed difference between the vehicle and the leader, desired speed, relative position of the vehicle in the traffic (leader's ID, follower's ID, potential left lane leader's ID, potential left lane follower's ID, potential right lane leader's ID, potential right lane follower's ID), maximum deceleration, target exit location, and vehicle class (regular, connected, or automated). The following lists the module inputs that are specified in the control panel: monitoring section length, monitoring timesteps, control type, control distance type, compliance error, optimization horizon, a list of potential values for advisory speed, and a list of potential values for broadcasting distance.

Output Arguments

The vehicle trajectories are the outputs that are written into a csv file.

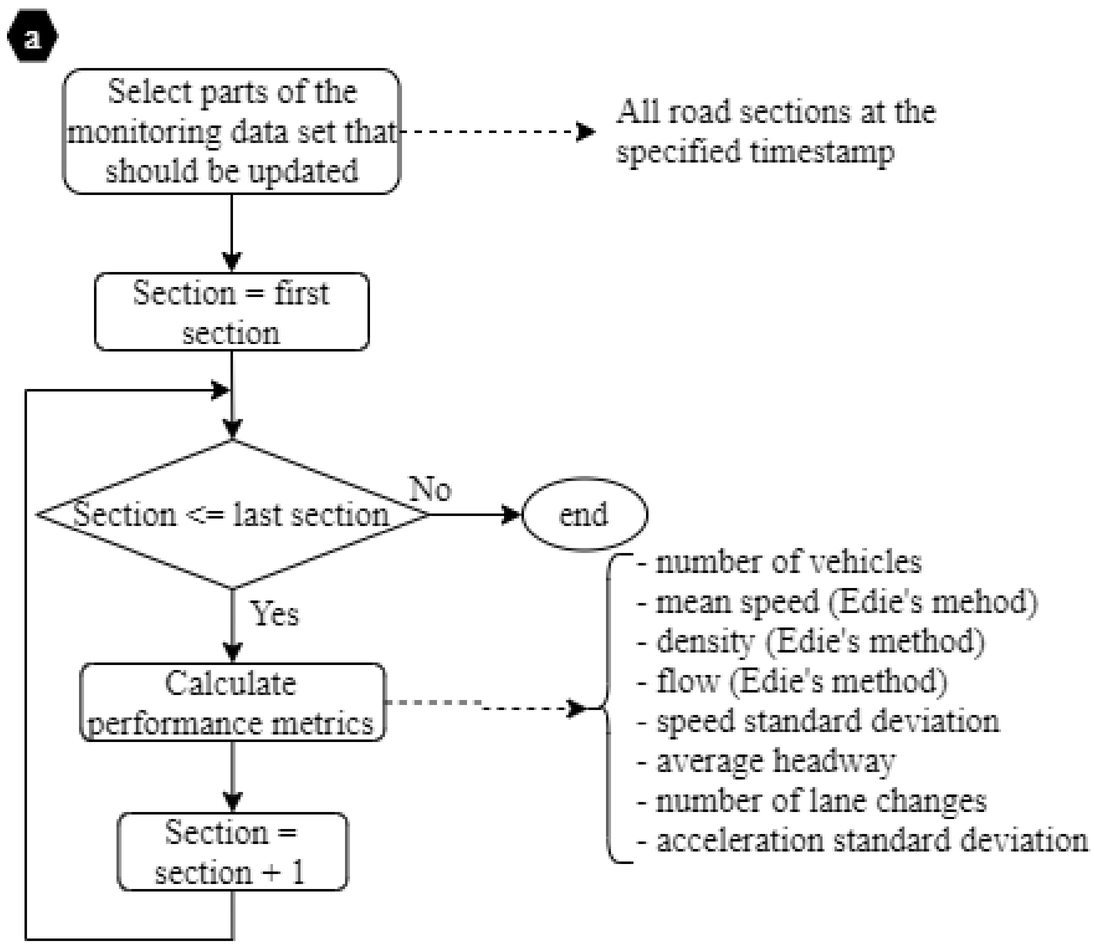
Pseudo-code

- Load generated vehicles
- Load vehicle trajectories
- Load car-following models
- Load initial traffic monitoring array for CVs and CAVs using init_tm_array function
- Load independent traffic monitoring array for all vehicles (this would be used for evaluating the system performance) using init_tm_array function
- If SPDHRM_control is centralized:
    - Initialized centralized SPDHRM vectors
    - Train base model using historical base data and the base_model_simple function
    - Train model to select optimal speed limit using the optimal_spd_pred

- Else if SPDHRM_control is decentralized:
  - Initialized decentralized SPDHRM vectors
  - Train base model using historical base data and the base_model_simple function
- Else if SPDHRM_control is optimization based:
  - Initialized decentralized SPDHRM vectors
  - Train base model using historical base data and the base_model_simple function
- For timestep from the beginning of the simulation to the end of the simulation:
  - Perform the car-following and lane-changing logics
  - Update vehicle trajectories
  - If timestep = monitoring timestep:
    - Update the traffic monitoring array of CVs and CAVs using the update_tm_array_edie
    - Update the traffic monitoring array of all vehicles using the update_tm_array_edie_all
    - If SPDHRM_control is active in the decentralized mode:
      - Reset desired speed to deactivate the previous SPDHRM strategy
      - Evaluate the decentralized SPDHRM strategy using the function dec_eval_SPDHRM
    - If SPDHRM_control is active in the centralized mode:
      - Reset desired speed to deactivate the previous SPDHRM strategy
      - Evaluate the decentralized SPDHRM strategy using the function eval_SPDHRM
    - If SPDHRM_control is active in the optimization-based mode:
      - Reset desired speed to deactivate the previous SPDHRM strategy
      - Evaluate the optimization-based SPDHRM strategy using the function opt_evaluate_SPDHRM
  - If SPDHRM_control is active in the decentralized mode:
    - Update speed using the dec_update_speed function
  - If SPDHRM_control is active in the centralized mode:
    - Update speed using the update_speed function
  - If SPDHRM_control is active in the optimization-based mode:
    - Update speed using the opt_update_speed function
- Generate the visualized results of the simulation

A.2. Traffic Monitoring Module

The traffic monitoring module, as shown in Figure A2, calculates the performance metrics for all sections of the road segment at each monitoring timestep. The performance metrics are calculated using the connected vehicles and connected automated vehicles in order to be used in the congestion prediction and speed control modules. Simultaneously, the performance metrics are calculated for all the vehicles in the system in order to evaluate the accuracy of the prediction models. As a result, for the purpose of evaluating and implementing the speed harmonization strategies, only the information of connected and automated vehicles is analyzed.



Source: FHWA

Figure A2. Flowchart. Traffic monitoring algorithm.

<u>init_tm_array</u>

Syntax

init_tm_array()

Description

This function creates the traffic monitoring array.

Input Arguments

None

Output Arguments

tm_array: the initial traffic monitoring array

Pseudo-code

- Calculate the number of sections
- Calculate the number of monitoring timesteps
- Calculate the number of rows for the array = sections * monitoring timestep
- Return tm_array

update_tm_array_edie

Syntax

update_tm_array_edie(in_tm_array, in_veh_traj, t)

Description

This function updates the traffic monitoring array at the timestep t using CVs and CAVs. The output of the function is used in the congestion prediction module.

Input Arguments

in_tm_array: the input traffic monitoring array with the following attributes: monitoring timestep, section number, number of vehicles in the section, mean speed, density, flow, speed standard deviation, mean headway, number of lane changes, acceleration standard deviation, speed standard deviation in the downstream section, mean speed in the downstream section, mean headway in the downstream section, congestion status, main line volume, ramp volume, SPDHRM type choice.

in_veh_traj: the input vehicle trajectory array. This array possesses the following information: vehicle initial position (lane and location in the lane), initial speed, initial acceleration, gap and speed difference between the vehicle and the leader, desired speed, relative position of the vehicle in the traffic (leader's ID, follower's ID, potential left lane leader's ID, potential left lane follower's ID, potential right lane leader's ID, potential right lane follower's ID), maximum deceleration, target exit location, and vehicle class (regular, connected, or automated).

t: timestep

Output Arguments

in_tm_array : The updated traffic monitoring array.

Pseudo-code

- Find the current monitoring timestep = t / monitoring timestep
- Find the section indices that require to be updated.
- For i in the range of section indices:

- Filter the CVs and CAVs that are in the current monitoring timestep, section i, and on the main lanes.
- If number of filtered vehicles in the section I at the current monitoring timestep is more than zero:
  - Update number of vehicles
  - Calculate Edie's mean speed
  - Calculate Edie's density
  - Calculate flow
  - Calculate speed standard deviation
  - Calculate mean headway
  - Calculate number of lane changes
  - Calculate acceleration standard deviation
  - Calculate speed standard deviation in the downstream section
  - Calculate mean speed in the downstream section
  - Calculate mean headway in the downstream section
- If it is the last section:
  - Set the attributes of the last section equal to the attributes of the second to last section

update_tm_array_edie_all

Syntax

update_tm_array_edie_all(in_tm_array, in_veh_traj, t)

Description

This function updates the traffic monitoring array at the timestep t using all vehicles in the system. The output of the function is used to evaluate the accuracy of the congestion prediction models and the effectiveness of the speed control module.

Input Arguments

in_tm_array: the input traffic monitoring array with the following attributes: monitoring timestep, section number, number of vehicles in the section, mean speed, density, flow, speed standard deviation, mean headway, number of lane changes, acceleration standard deviation, speed standard deviation in the downstream section, mean speed in the downstream section, mean headway in the downstream section, congestion status, main line volume, ramp volume, SPDHRM type choice.

in_veh_traj: the input vehicle trajectory array. This array possesses the following information: vehicle initial position (lane and location in the lane), initial speed, initial acceleration, gap and speed difference between the vehicle and the leader, desired speed, relative position of the vehicle in the traffic (leader's ID, follower's ID, potential left lane leader's ID, potential left lane follower's ID, potential right lane leader's ID, potential right lane follower's ID), maximum deceleration, target exit location, and vehicle class (regular, connected, or automated).

t: timestep

Output Arguments

in_tm_array : The updated traffic monitoring array.

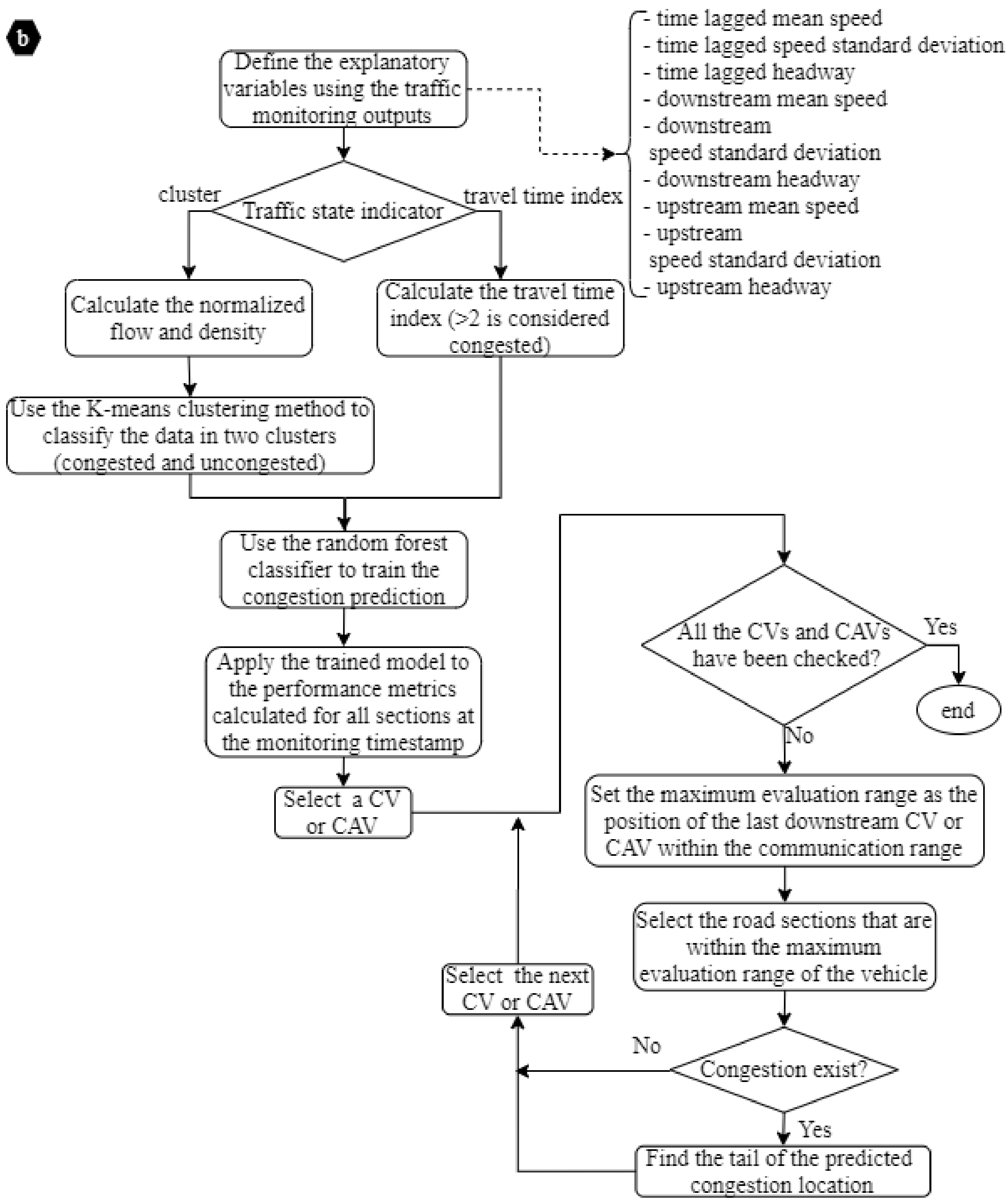
Pseudo-code

- Find the current monitoring timestep = t / monitoring timestep
- Find the section indices that require to be updated.
- For i in the range of section indices:
    - Filter the vehicles that are in the current monitoring timestep, section i, and on the main lanes.
    - If number of filtered vehicles in the section I at the current monitoring timestep is more than zero:
        - Update number of vehicles
        - Calculate Edie's mean speed
        - Calculate Edie's density
        - Calculate flow
        - Calculate speed standard deviation
        - Calculate mean headway
        - Calculate number of lane changes
        - Calculate acceleration standard deviation
        - Calculate speed standard deviation in the downstream section
        - Calculate mean speed in the downstream section
        - Calculate mean headway in the downstream section
    - If it is the last section:
        - Set the attributes of the last section equal to the attributes of the second to last section

A.3. Congestion Prediction Module

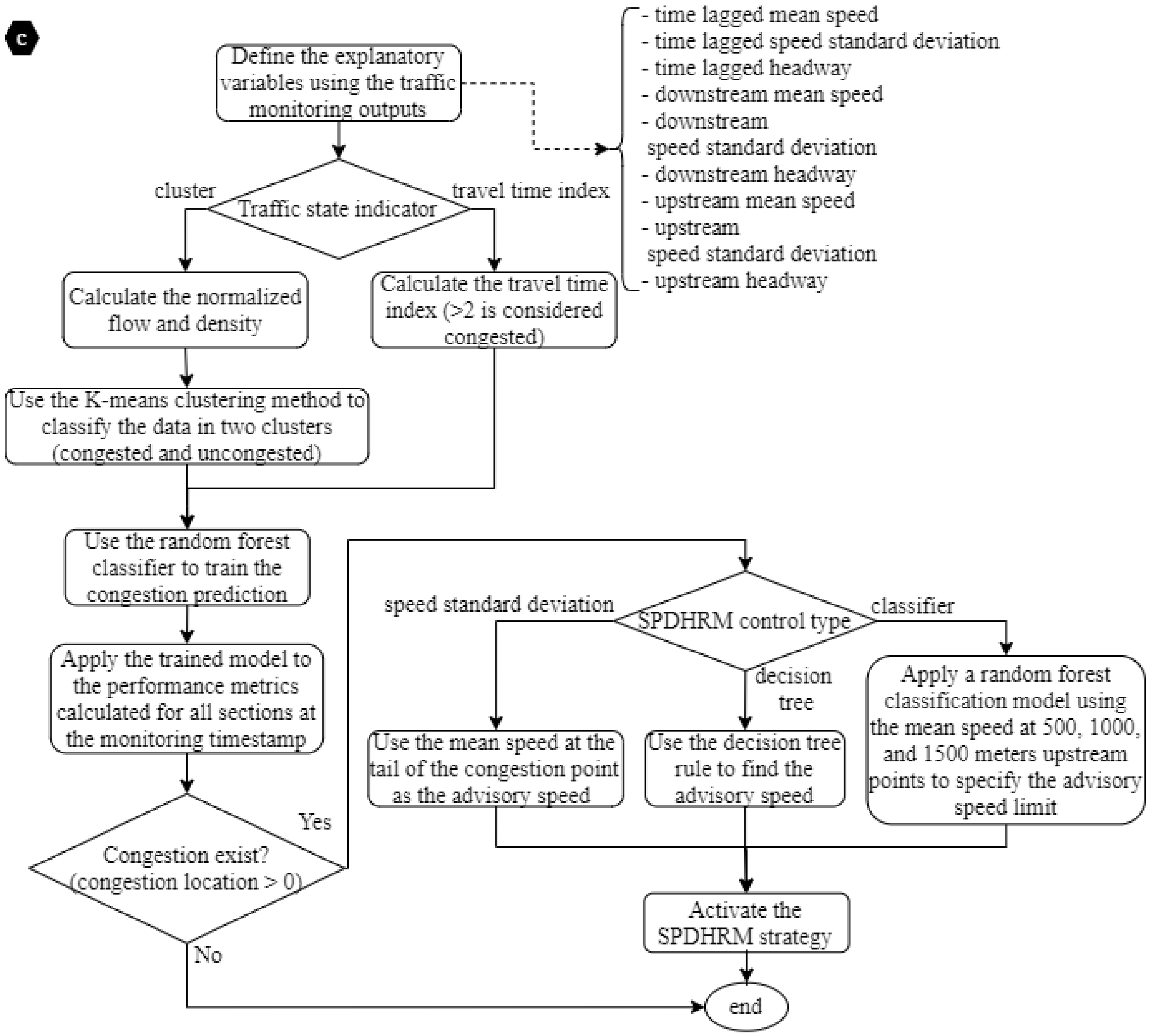
Figure A3 to Figure A5 show the algorithms in the congestion prediction module. Based on the settings defined in the control panel, one of the algorithms is performed: Centralized, decentralized, or optimization-based. The location and time of the congestion along with the advisory speed limit are determined by this module. The training part of the algorithms could be performed in an online or offline manner. The training part could be skipped if a predefined congestion prediction model is utilized.

In the centralized strategy, system evaluates the state of the transport facility through information received from CAVs and detectors. Then, it predicts future states using machine learning algorithms. Finally, the advisory messages are prepared to be broadcasted to CAVs in order to minimize disturbance (speed standard deviation). On the other hand, in the decentralized strategy, each CAV receives information from a cluster or fleet of CAVs within a detection/connection range. Then, each vehicle utilizes individualized or group-based machine learning algorithms to predict the future state of clusters. Finally, the vehicles adjust their longitudinal and lateral driving behavior to minimize disruption in a cluster or fleet of vehicles, i.e. self-homogenize. The advisory speed limit in the optimization-based strategy is determined by solving an optimization problem that seeks to maximize the distance traveled by the vehicles in a specified time period (prediction time horizon). Based on the available computational resources, the complexity of the optimization problem could be adjusted ranging from jointly determining the advisory speed for each vehicle and the broadcasting distance (high complexity) to selecting the advisory speed limit and the broadcasting distance from a limited set (low complexity).



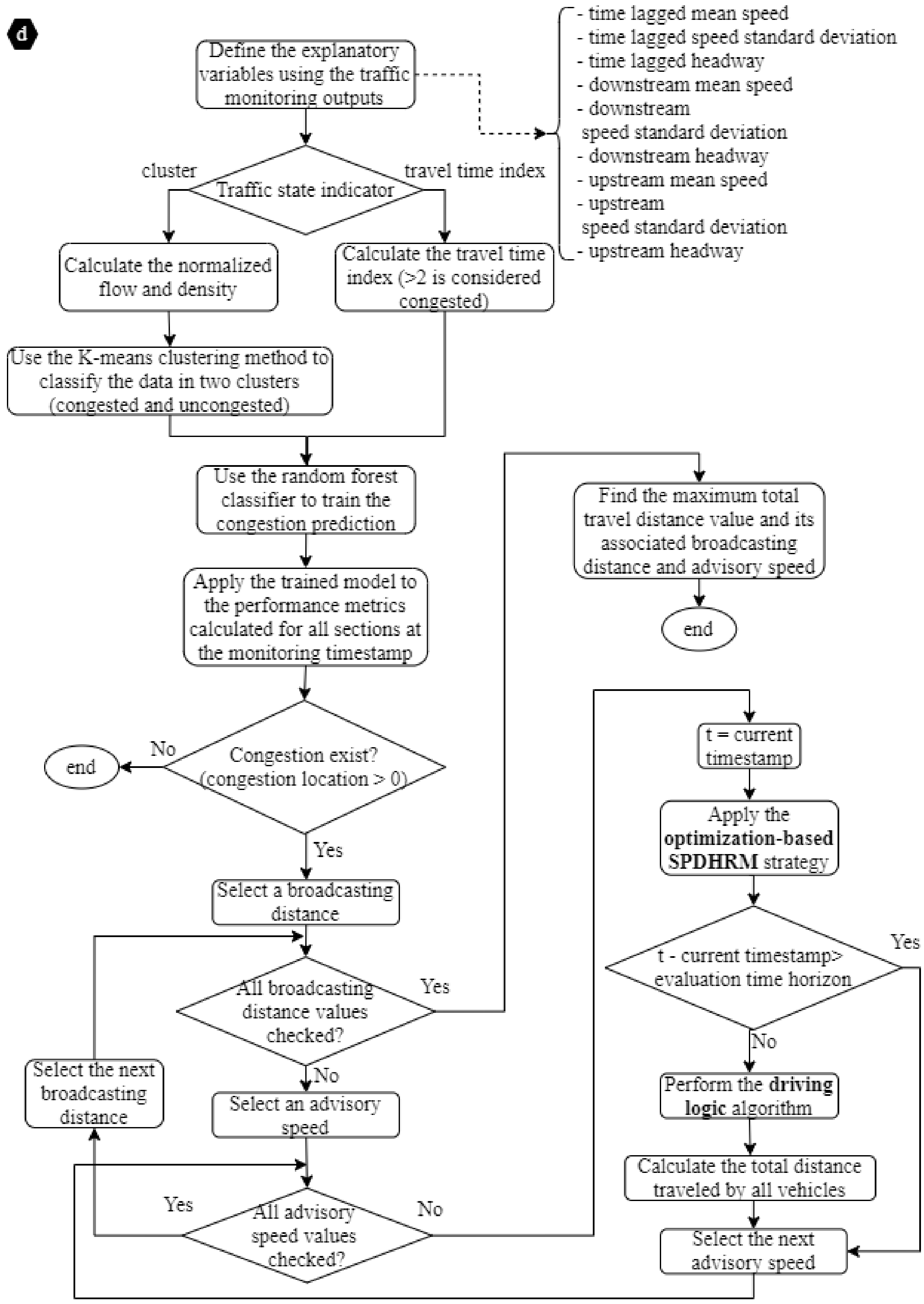
Source: FHWA

Figure A3. Flowchart. Decentralized SPDHRM strategy evaluation algorithm.



Source: FHWA

Figure A4. Flowchart. Centralized SPDHRM strategy evaluation algorithm.



Source: FHWA

Figure A5. Flowchart. Optimization-based SPDHRM strategy evaluation algorithm.

base_model_simple

Syntax

base_model_simple (self, cong_state_type, pred_horizon)

Description

This function trains the congestion prediction model that utilizes the random forest algorithm. The following temporally lagged variables are used as the explanatory variables of the model to predict the congestion status a specified section of the network: mean speed in the specified section, mean headway in the specified section, mean speed in the downstream section, speed standard deviation in the downstream section. This function is useful to build an offline congestion prediction model.

Input Arguments

cong_state_type**:** equal 'cluster' if the method of determining the congestion status is based on clustering the data. Equals 'tti' if the congestion status is determined based on the travel time index.

pred_horizon: the time period over which the congestion status of the network is predicted.

Output Arguments

rf_model : The trained model to be used for congestion prediction.

Pseudo-code

- Open the csv file that contains the four column attributes described above.
- Training data explanatory variables = selected columns of the dataframe that contains the four attributes. Use the input pred_horzon in the column filtering process.
- If cong_state_type = 'cluster':
    - Training data dependent variable = the column of the dataframe that shows the traffic status calculated based on the clustering method
- Else:
    - Training data dependent variable = the column of the dataframe that shows the traffic status calculated based on the travel time index
- rf_model = Train the model by applying the RandomeForestClassifier in the sklearn package with the following settings: n_estimators = 500, max_depth = 2, random_state = 0
- Return rf_model

congestion_loc_fun

Syntax

congestion_loc_fun(in_congestion_pred)

Description

This function specifies the location of congestion in the centralized control setting.

Input Arguments

in_congestion_pred: predicted congestion status for each section of the network

Output Arguments

cong_loc_tail: the tail of the first congestion found (searching from the most upstream section to the most downstream section)

Pseudo-code

- For i in range of indices for the predicted congestion status array:
    - If the congestion location has not been found yet:
        - If the i^{th} road section is congested:
            - cong_loc_tail = i
            - break
- Return cong_loc_tail

congestion_pred

Syntax

congestion_pred(in_tm_array, in_pred_model, t)

Description

This function predicts the starting location of the congestion in the centralized control setting.

Input Arguments

in_tm_array: the input traffic monitoring array.

in_pred_model: the congestion prediction model developed based on random forest algorithm.

t: timestep

Output Arguments

cong_section_id: the ID of the section where the congestion starts

cong_x: the location where the congestion starts

Pseudo-code

- Calculate the monitoring timestep that needs to be updated as t / monitoring timestep
- Find the indices for the section of in_tm_array that needs to be updated using the result of the previous line
- Select the columns of the specified section of in_tm_array that correspond to the four independent variables of the prediction model (discussed in the base_model_simple function)
- Predict the congestion status using in_pred_model
- cong_section_id = apply congestion_loc_fun to the predicted congestion status
- cong_x = cong_section_id * section length – section length

- Return cong_section_id & cong_x

dec_eval_SPDHRM

Syntax

dec_eval_SPDHRM(in_veh_trajs_init, in_veh_traj, in_tm_array, in_pred_model, t)

Description

This function determines new updated speed limit in the decentralized control setting.

Input Arguments

in_veh_trajs_init: the initial vehicle trajectory array.

in_veh_traj: the vehicle trajectory array.

in_tm_array: the input traffic monitoring array.

in_pred_model: the congestion prediction model developed based on random forest algorithm.

t: timestep

Output Arguments

cong_dwnstrm_loc_id: an array storing the location ID of downstream congestion for each vehicle.

cong_dwnstrm_loc_speed: an array storing the downstream congestion speed for each vehicle.

Pseudo-code

- Calculate the monitoring timestep that needs to be updated as t / monitoring timestep
- congestion_pred, section_speed = Predicted congestion attributes using the dec_pred_cong function
- cong_dwnstrm_loc_id, cong_dwnstrm_loc_speed = check downstream congestion for all vehicles using the dec_cong_info_downstream function
- Return cong_dwnstrm_loc_id & cong_dwnstrm_loc_speed

dec_cong_info_downstream

Syntax

dec_cong_info_downstream(in_veh_trajs_init, in_congestion_pred, in_section_speeds)

Description

This function checks for each vehicle whether there is congestion within the maximum of its detection range and control range in the decentralized control setting.

Input Arguments

in_veh_trajs_init: the initial vehicle trajectory array.

in_congestion_pred: predicted congestion status for each section of the network

in_section_speeds: speed values in the road sections.

Output Arguments

cong_dwnstrm_loc_id: an array storing the location ID of downstream congestion for each vehicle.

cong_dwnstrm_loc_speed: an array storing the downstream congestion speed for each vehicle.

Pseudo-code

- Initialize the cong_dwnstrm_loc_id and cong_dwnstrm_loc_speed arrays with zerp values
- For vehicle in list of vehicles:
    - If the vehicle is on the main lanes and is a CV or CAV:
        - Calculate the maximum evaluation distance using the function dec_max_eval_distance
        - If the maximum evaluation distance > 0:
            - Define the number of sections to check as the maximum evaluation distance divided by the section length
            - Find the current section of the vehicle
            - For i in indices of the sections that are located downstream of the vehicle and needs to be checked:
                - If in_congestion_pred[i] = 1:
                    - cong_dwnstrm_loc_id = i + 1
                    - cong_dwnstrm_loc_speed = in_section_speeds[i]
                    - break
- Return cong_dwnstrm_loc_id & cong_dwnstrm_loc_speed

dec_max_eval_distance

Syntax

dec_max_eval_distance(in_veh_position, in_veh_trajs_init)

Description

This function determines the maximum evaluation distance for a vehicle in the decentralized control setting. The evaluation distance is constrained by 1) CV communication range, 2) CV market penetration rate.

Input Arguments

in_veh_position: position of the vehicle in the network.

in_veh_trajs_init: the initial vehicle trajectory array.

Output Arguments

eval_dist_max: maximum evaluation distance.

Pseudo-code

- Filter moving CVs and CAVs that are ahead of target vehicle in the main lanes.
- Calculate the relative distance between the filtered vehicles and the target vehicle.
- Sort vehicles based on their distance to the target vehicle.
- For i in indices of the sorted list of vehicles ahead:
    - Consider the relative distance to the target vehicle for vehicle i and vehicle i + 1
    - Calculate the relative distance between the consecutive vehicle.
    - If the relative distance is less than or equal to the communication range initialized in the control panel:
        - eval_dist_max = relative distance between the target vehicle and vehicle i + 1
    - Else:
        - break
- Return eval_dist_max

dec_pred_cong

Syntax

dec_pred_cong(in_mon_timestep_to_update, in_tm_array, in_pred_model)

Description

This function predicts the congestion location in the decentralized control setting.

Input Arguments

in_mon_timestep_to_update: the monitoring timestep to update.

in_tm_array: the input traffic monitoring array.

in_pred_model: the congestion prediction model developed based on random forest algorithm.

Output Arguments

congestion_pred_out:

section_speeds:

Pseudo-code

- Find the indices for the section of in_tm_array that needs to be updated using the result of the previous line. in_mon_timestep_to_update is used in the filtering process.
- Select the columns of the specified section of in_tm_array that correspond to the four independent variables of the prediction model (discussed in the base_model_simple function)
- congestion_pred_out = Predict the congestion status using in_pred_model
- section_speeds = extract the speed values of the section

- Return congestion_pred_out & section_speeds

det_new_speed

Syntax

det_new_speed(in_tm_array, in_cong_location, in_opt_spd_limit, t)

Description

This function determines new updated speed limit in the centralized control setting.

Input Arguments

in_tm_array: the input traffic monitoring array.

in_cong_location: the congestion location.

in_opt_spd_limit: the optimal variable speed limit function developed based on random forest algorithm.

t: timestep

Output Arguments

new_speed: the new advisory speed

current_ssd: the speed standard deviation for the congestion section

Pseudo-code

- Calculate the monitoring timestep that needs to be updated as t / monitoring timestep
- current_speed = the mean speed for the congestion section at the calculated monitoring timestep
- current_ssd = the speed standard deviation for the congestion section at the calculated monitoring timestep
- Set congestion speed equal to current_speed
- Find the mean speed for the section 500 meters upstream of the congestion location at the calculated monitoring timestep
- Find the mean speed for the section 1000 meters upstream of the congestion location at the calculated monitoring timestep
- Find the mean speed for the section 1500 meters upstream of the congestion location at the calculated monitoring timestep
- If control type is ‘decision tree’:
    - If congestion speed < 55 kph:
        - new_speed = 55 kph
    - Else if congestion speed < 75 kph:
        - new_speed = 70 kph
    - Else:

        - new_speed = 90 kph
- Else if control type is ‘speed standard deviation’:
    - new_speed = current_speed
- Else if control type is ‘classifier’:
    - Create a dataframe by the following column: the congestion speed, the mean speed for the section 500 meters upstream of the congestion location, the mean speed for the section 1000 meters upstream of the congestion location, and the mean speed for the section 1500 meters upstream of the congestion location
    - Predict the optimal variable speed limit using the function in_opt_spd_limit
- Return new_speed & current_ssd

evaluate_SPDHRM

Syntax

evaluate_SPDHRM (in_tm_array, in_veh_trajs_init, in_pred_model, in_opt_spd_limit, t)

Description

This function evaluates whether the SPDHRM should be activated or not in the centralized control setting.

Input Arguments

in_tm_array: the input traffic monitoring array.

in_veh_trajs_init: the input vehicle trajectory array.

in_pred_model: the congestion prediction model developed based on random forest algorithm.

in_opt_spd_limit: the optimal variable speed limit function developed based on random forest algorithm.

t: timestep

Output Arguments

congestion_location: location that the congestion has occurred.

in_VSL_SPDHRM: advisory speed limit.

in_active_SPDHRM: SPDHRM active/inactive status.

in_ssd_SPDHRM: speed standard deviation.

Pseudo-code

- in_active_SPDHRM = 0 (Deactivate the previous SPDHRM indicator)
- in_VSL_SPDHRM = desire speed (Deactivate the previous SPDHRM indicator)
- in_ssd_SPDHRM = 0 (Deactivate the previous SPDHRM indicator)
- Predict congestion_location using the congestion_pred function
- If congestion_location > 0:

- in_VSL_SPDHRM = the new advisory speed limit determined by the function det_new_speed
- in_ssd_SPDHRM = the new speed standard deviation determined by the function det_new_speed
- in_active_SPDHRM = 1 (activate SPDHRM)

- Return congestion_location, in_active_SPDHRM, in_VSL_SPDHRM, & in_ssd_SPDHRM

opt_distance_traveled

Syntax

opt_distance_traveled(in_timestep, in_eval_duration_sec, in_veh_trajs_init, in_cfm, in_lcm, in_brdcst_dist, in_updt_speed, in_congestion_location)

Description

This function calculates the total distance traveled by the vehicles during the optimization horizon time period.

Input Arguments

in_timestep: the current timestep.

in_eval_duration_sec: the optimization horizon in seconds.

in_veh_trajs_init: the input vehicle trajectory array.

in_cfm: car-following model.

in_lcm_m: lane-changing model.

in_brdcst_dist: broadcasting distance.

in_updt_speed: the variable speed limit array.

in_congestion_location: the congestion location.

Output Arguments

total_veh_distance: total distance traveled by the vehicles.

Pseudo-code

- Take a copy of the vehicle trajectory array in order to update the copy version during the in_eval_duration_sec and avoid changing the original vehicle trajectory array
- For t in a list of timesteps that starts from in_timestep and ends at in_eval_duration_sec after the in_timestep
  - If t < in_timestep + the monitoring timestep:
    - Update the copied vehicle trajectories using the opt_update_speed function (This would essentially apply the speed harmonization strategy for the first monitoring timestep only)
  - Else if t = in_timestep + the monitoring timestep:

        - Reset the speed limit
    - Update vehicle trajectories using the car-following and lane-changing models
- Calculate the total distance traveled by each vehicle
- Calculate the total distance traveled by all vehicles
- Return total_veh_distance

opt_evaluate_SPDHRM

Syntax

opt_evaluate_SPDHRM(in_tm_array, in_pred_model, in_timestep, in_eval_duration_sec, in_veh_trajs_init, in_cfm, in_lcm)

Description

This function derives the value of the decision variable in the speed harmonization optimization problem.

Input Arguments

in_tm_array: the input traffic monitoring array.

in_pred_model: the congestion prediction model developed based on random forest algorithm.

in_timestep: the current timestep.

in_eval_duration_sec: the optimization horizon in seconds.

in_veh_trajs_init: the input vehicle trajectory array.

in_cfm: car-following model.

in_lcm: lane-changing model.

Output Arguments

congestion_location: location that the congestion has occurred.

in_brdcst_dist: the broadcasting distance.

in_VSL_SPDHRM: the variable speed limit.

in_active_SPDHRM: SPDHRM active/inactive status.

speed_congestion: the average traffic speed at the congestion section.

speed_500: the mean traffic speed at the section which is 500 upstream of the congestion point.

speed_1000: the mean traffic speed at the section which is 1000 upstream of the congestion point.

speed_1500: the mean traffic speed at the section which is 1500 upstream of the congestion point.

speed_cong_slead: the mean speed in the downstream section of the congestion point.

ssd_cong_slead: the speed standard deviation in the downstream section of the congestion point.

flow_cong: the traffic flow at the congested section.

density_cong: the density at the congested section.

Pseudo-code

- Find the congestion location using the congestion_pred function
- If a congestion point exists:
    - Find the optimal set of broadcasting distance and the advisory speed limit using the function opt_find_optimal_params
    - in_active_SPDHRM = 1 (activate SPDHRM)
    - Calculate the current monitoring timestep as in_timestep divided by the monitoring timestep
    - Select values of the in_tm_array that represent speed_congestion, speed_500, speed_1000, speed_1500, speed_cong_slead, ssd_cong_slead, flow_cong, and density_cong
- Return congestion_location, in_brdcst_dist, in_VSL_SPDHRM, in_active_SPDHRM, speed_congestion, speed_500, speed_1000, speed_1500, speed_cong_slead, ssd_cong_slead, flow_cong, density_cong

opt_find_optimal_params

Syntax

opt_find_optimal_params(in_timestep, in_eval_duration_sec, in_veh_trajs_init, in_cfm_TLPR, in_cfm_IDM, in_cfm_AV_AREM, in_lcm_m, in_lcm_r, in_congestion_location)

Description

This function finds the optimal parameters for the optimization-based speed harmonization strategy.

Input Arguments

in_timestep: the current timestep.

in_eval_duration_sec: the optimization horizon in seconds.

in_veh_trajs_init: the input vehicle trajectory array.

in_cfm: car-following model.

in_lcm: lane-changing model.

in_congestion_location: location that the congestion has occurred.

Output Arguments

optimal_brdcst_dist: optimal broadcasting distance.

optimal_speed: optimal advisory speed limit.

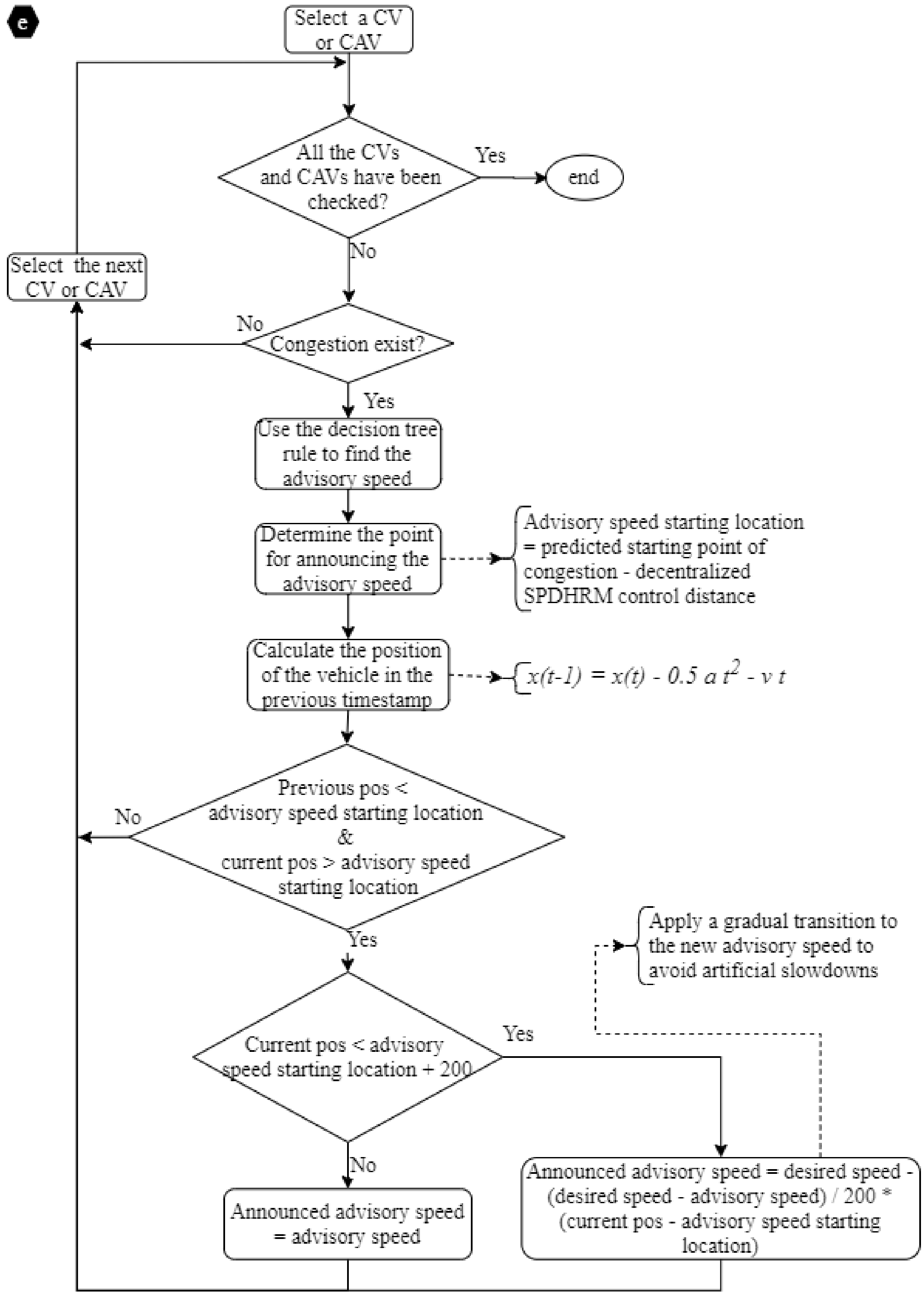
Pseudo-code

- Create an array to store values of total distance travelled by all vehicles when different pairs of broadcasting distance and advisory speed limit are used.

- For broadcasting distance in the list of potential broadcasting distance values:
    - For speed in the list of potential advisory speed values:
        - Calculate the total travelled distance by all vehicles using the opt_distance_traveled function
        - Store the total travelled distance in the array created at the beginning of this function
    - Find the element of the storage array that has the highest total travelled distance
    - Return optimal_brdcst_dist & optimal_speed

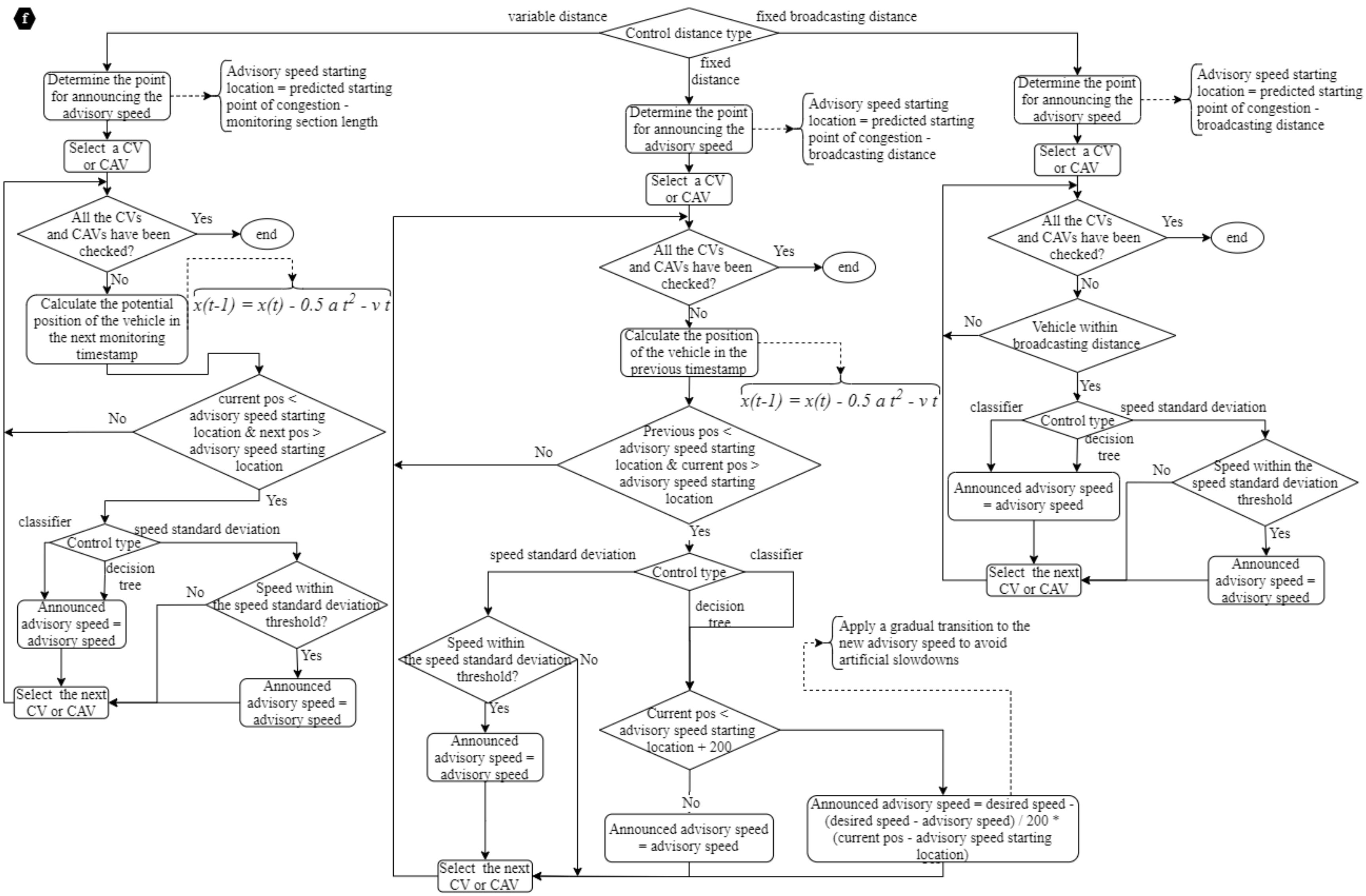
A.4. Speed Control Module

Figure A6 to Figure A8 show the algorithms of the speed control module. A step-by-step procedure is performed to communicate an advisory speed limit to connected vehicles and automated vehicles. Extra precautions are taken to prevent artificial slowdowns in the simulation by applying a gradual transition in the advisory speed limit communicated to the vehicles.



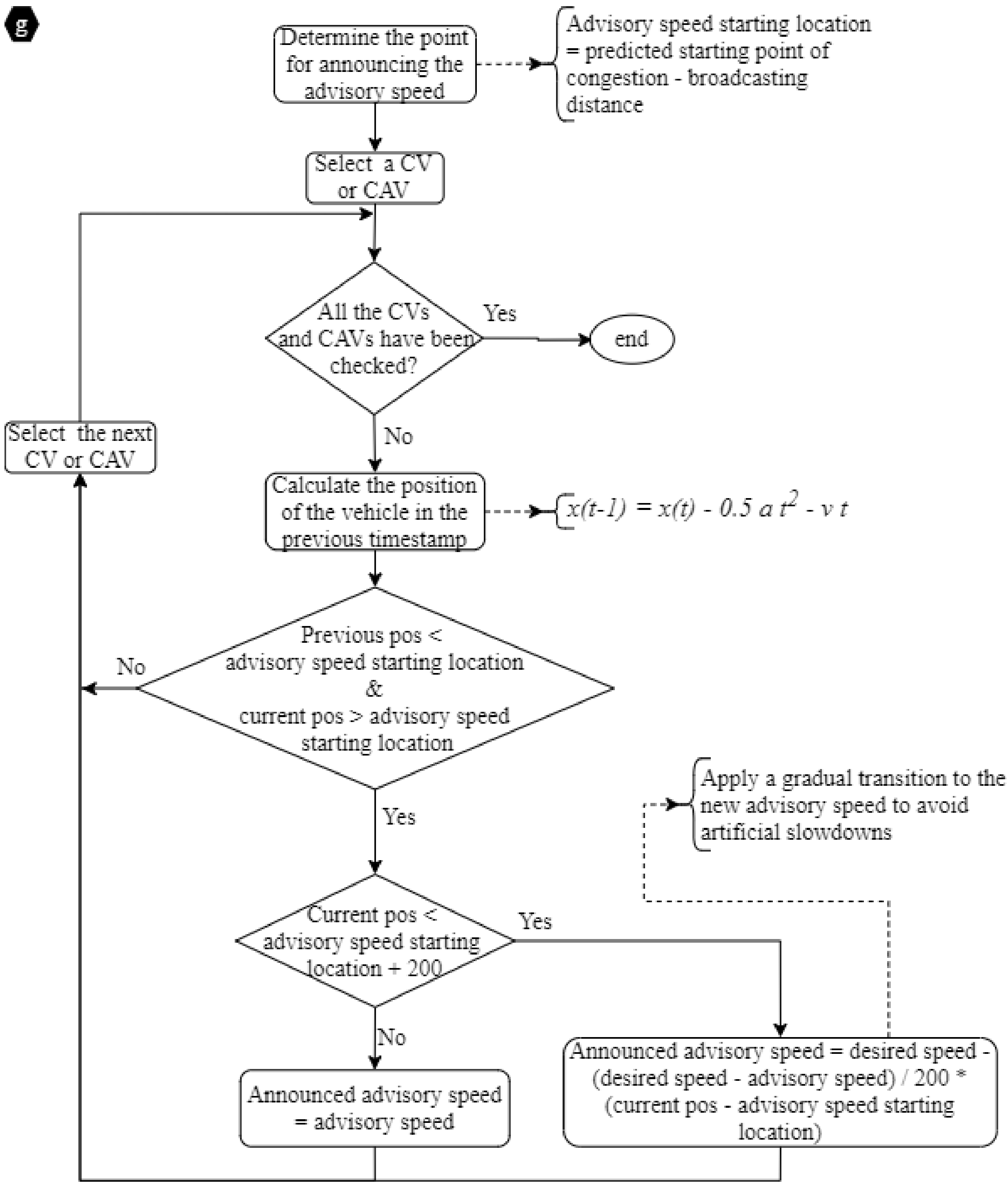
Source: FHWA

Figure A6. Flowchart. Decentralized SPDHRM strategy implementation algorithm.



Source: FHWA

Figure A7. Flowchart. Centralized SPDHRM strategy implementation algorithm.



Source: FHWA

Figure A8. Flowchart. Optimization-based SPDHRM strategy implementation algorithm.

update_speed

Syntax

update_speed(in_control_distance_type, in_congestion_location, in_veh_trajs_init, in_veh_trajs, in_VSL_SPDHRM, in_ssd_SPDHRM, in_veh_active_SPDHRM)

Description

This function applies the speed control in the centralized control setting.

Input Arguments

in_control_distance_type: the control distance type variable which is a number representing one of the following choices: fixed control distance (1), variable distance (2), fixed point (3).

in_congestion_location: the congestion location.

in_veh_trajs_init: the initial vehicle trajectory array.

in_veh_trajs: the vehicle trajectory array.

in_VSL_SPDHRM: the variable speed limit array.

in_ssd_SPDHRM: the speed standard deviation array.

in_veh_active_SPDHRM: SPDHRM active/inactive status.

Output Arguments

in_veh_trajs_init: Updated vehicle trajectories array.

Pseudo-code

- If fixed control distance is selected (in_control_distance_type = 1):
    - Estimate the location where the speed control implementation ends as in_congestion_location * section length
    - Estimate the location where the speed control implementation starts as the broadcasting distance subtracted from the estimated end of the speed control
    - For i in indices of the list of vehicles:
        - If vehicle position is within the speed control range:
            - If control type is decision tree:
                - Update the speed values for CVs and CAVs using in_VSL_SPDHRM
            - Else if control type is speed standard deviation:
                - If the speed is within a standard deviation threshold of the variable speed limit (the threshold is defined by the user in the control panel. The boundary values of the range are in_VSL_SPDHRM ±threshold * in_ssd_SPDHRM):
                    - Update the speed values for CVs and CAVs using in_VSL_SPDHRM

- Else if variable distance is selected (in_control_distance_type = 2):
    - Estimate the location where the congestion starts as in_congestion_location * section length
    - Estimate the location where the speed control implementation starts as the broadcasting distance subtracted from the estimated end of the speed control
    - For i in indices of the list of vehicles:
        - Calculate the previous position of the vehicle
        - If (previous position of the vehicle < location where the speed control implementation starts) & (current position of the vehicle >= location where the speed control implementation starts):
            - in_veh_active_SPDHRM = 1(Activate the SPDHRM strategy for the vehicle)
        - If in_veh_active_SPDHRM = 1:
            - If control type is decision tree:
                - Update the speed values for CVs and CAVs using in_VSL_SPDHRM
            - Else if control type is speed standard deviation:
                - If the speed is within a standard deviation threshold of the variable speed limit (the threshold is defined by the user in the control panel. The boundary values of the range are in_VSL_SPDHRM ±threshold * in_ssd_SPDHRM):
                    - Update the speed values for CVs and CAVs using in_VSL_SPDHRM
- Else if fixed point is selected (in_control_distance_type = 3):
    - Estimate the location where the congestion starts as in_congestion_location * section length
    - Estimate the location where the speed control implementation starts as the broadcasting distance subtracted from the estimated end of the speed control
    - For i in indices of the list of vehicles:
        - Calculate the previous position of the vehicle
        - If (previous position of the vehicle < location where the speed control implementation starts) & (current position of the vehicle >= location where the speed control implementation starts):
            - in_veh_active_SPDHRM = 1(Activate the SPDHRM strategy for the vehicle)
        - If in_veh_active_SPDHRM = 1:
            - If control type is decision tree:

- Update speed gradually over 200 meters to avoid artificial slowdowns based on the following steps
    - Delta_v_des_ratio = (desired speed - in_VSL_SPDHRM) / 200
    - If current position of the vehicle < location where the speed control implementation starts + 200
        - Update the speed values for CVs and CAVs using the following formula: desired speed - Delta_v_des_ratio * (vehicle position - location where the speed control implementation starts)
    - Else:
        - Update the speed values for CVs and CAVs using in_VSL_SPDHRM
- Else if control type is speed standard deviation:
    - If the speed is within a standard deviation threshold of the variable speed limit (the threshold is defined by the user in the control panel. The boundary values of the range are in_VSL_SPDHRM ±threshold * in_ssd_SPDHRM):
        - Update the speed values for CVs and CAVs using in_VSL_SPDHRM

- Return in_veh_trajs_init

dec_update_speed

Syntax

dec_update_speed(in_veh_trajs_init, in_cong_dwnstrm_loc_id, in_cong_dwnstrm_loc_speed, in_veh_active_SPDHRM)

Description

This function applies the speed control in the decentralized control setting.

Input Arguments

in_veh_trajs_init: the initial vehicle trajectory array.

in_cong_dwnstrm_loc_id: an array storing the location ID of downstream congestion for each vehicle.

in_cong_dwnstrm_loc_speed: an array storing the downstream congestion speed for each vehicle.

in_veh_active_SPDHRM: SPDHRM active/inactive status.

Output Arguments

in_veh_trajs_init: Updated vehicle trajectories array.

Pseudo-code

- For i in indices of the list of vehicles:
    - if congestion is detected for vehicle i (in_cong_dwnstrm_loc_id [i] > 0):
        - Determine the variable speed limit based on the decision tree
        - Estimate the location where the congestion starts as in_congestion_location * section length
        - Estimate the location where the speed control implementation starts as the broadcasting distance subtracted from the estimated end of the speed control
        - Calculate the previous position of the vehicle
        - If (previous position of the vehicle < location where the speed control implementation starts) & (current position of the vehicle >= location where the speed control implementation starts):
            - in_veh_active_SPDHRM = 1(Activate the SPDHRM strategy for the vehicle)
        - If in_veh_active_SPDHRM = 1:
            - Update speed gradually over 200 meters to avoid artificial slowdowns based on the following steps
            - Delta_v_des_ratio = (desired speed - variable speed limit) / 200
            - If current position of the vehicle < location where the speed control implementation starts + 200
                - Update the speed values for CVs and CAVs using the following formula: desired speed - Delta_v_des_ratio * (vehicle position - location where the speed control implementation starts)
            - Else:
                - Update the speed values for CVs and CAVs using variable speed limit
- Return in_veh_trajs_init

opt_update_speed

Syntax

opt_update_speed(in_veh_trajs_init, in_congestion_location, in_brdcst_dist, in_VSL_SPDHRM, in_veh_active_SPDHRM)

Description

This function applies the speed control in the optimization-based control setting.

Input Arguments

in_veh_trajs_init: the initial vehicle trajectory array.

in_congestion_location: the congestion location.

in_brdcst_dist: the broadcasting distance.

in_VSL_SPDHRM: the variable speed limit array.

in_veh_active_SPDHRM: SPDHRM active/inactive status.

Output Arguments

in_veh_trajs_init: Updated vehicle trajectories array.

Pseudo-code

- Estimate the location where the congestion starts as in_congestion_location * section length
- Estimate the location where the speed control implementation starts as the broadcasting distance subtracted from the estimated end of the speed control
- For i in indices of the list of vehicles:
    - Calculate the previous position of the vehicle
    - If (previous position of the vehicle < location where the speed control implementation starts) & (current position of the vehicle >= location where the speed control implementation starts):
        - in_veh_active_SPDHRM = 1(Activate the SPDHRM strategy for the vehicle)
    - If in_veh_active_SPDHRM = 1:
        - Update speed gradually over 200 meters to avoid artificial slowdowns based on the following steps
        - Delta_v_des_ratio = (desired speed - variable speed limit) / 200
        - If current position of the vehicle < location where the speed control implementation starts + 200
            - Update the speed values for CVs and CAVs using the following formula: desired speed - Delta_v_des_ratio * (vehicle position - location where the speed control implementation starts)
        - Else:
            - Update the speed values for CVs and CAVs using variable speed limit
- Return in_veh_trajs_init