

**Task Order: 22**

**March 22, 2020**

**Developing Analysis, Modeling, and Simulation (AMS) Tools for Connected and Automated Vehicle (CAV) Applications**

**Algorithm Description Document: Speed Harmonization Model**

SI Conversion Chart. See https://www.fhwa.dot.gov/publications/convtabl.cfm for html version.

Table of Contents

[Executive Summary 6](#_Toc35779368)

[Chapter 1. PURPOSE OF THIS MODEL 7](#_Toc35779369)

[Purpose of this Document 7](#_Toc35779370)

[Purpose of this Model 7](#_Toc35779371)

[Document Overview 8](#_Toc35779372)

[Chapter 2. MODEL DEVELOPMENT AND LOGIC 9](#_Toc35779373)

[Descriptions of Model Logic 9](#_Toc35779374)

[Model Development 22](#_Toc35779375)

[Chapter 3. MODEL CALIBRATION AND VALIATION 25](#_Toc35779376)

[Dataset 25](#_Toc35779377)

[Calibration Approach 27](#_Toc35779378)

[Calibration and Validation Process 28](#_Toc35779379)

[Chapter 4. BASIC GUIDANCE ON MODEL IMPLEMENTATION 32](#_Toc35779380)

[Chapter 5. Use Case 34](#_Toc35779381)

[Implementation of the developed model into a traffic simulation tool 34](#_Toc35779382)

[Design of simulation experiments 37](#_Toc35779383)

[Simulation results for the different scenarios 39](#_Toc35779384)

[Chapter 6. SUMMARY AND RECOMMENDATIONS 42](#_Toc35779385)

[Acknowledgments 44](#_Toc35779386)

[References 45](#_Toc35779387)

[APPENDIX 47](#_Toc35779388)

[A.1. Main Algorithm 47](#_Toc35779389)

[A.2. Traffic Monitoring Module 50](#_Toc35779390)

[A.3. Congestion Prediction Module 54](#_Toc35779391)

[A.4. Speed Control Module 69](#_Toc35779392)

List of Figures

[Figure 1. Diagram. Speed Harmonization Overall Framework. 9](#_Toc35779393)

[Figure 2. Flowchart. Main Algorithm. 11](#_Toc35779394)

[Figure 3. Flowchart. Traffic Monitoring Algorithm. 12](#_Toc35779395)

[Figure 4. Flowchart. Decentralized SPDHRM Strategy Evaluation Algorithm. 14](#_Toc35779396)

[Figure 5. Flowchart. Centralized SPDHRM Strategy Evaluation Algorithm. 15](#_Toc35779397)

[Figure 6. Flowchart. Optimization-based SPDHRM Strategy Evaluation Algorithm. 16](#_Toc35779398)

[Figure 7. Flowchart. Decentralized SPDHRM Strategy Implementation Algorithm. 18](#_Toc35779399)

[Figure 8. Flowchart. Centralized SPDHRM Strategy Implementation Algorithm. 19](#_Toc35779400)

[Figure 9. Flowchart. Optimization-based SPDHRM Strategy Implementation Algorithm. 21](#_Toc35779401)

[Figure 10. Formulas. Edie’s generalized definition of the traffic flow characteristics. 22](#_Toc35779402)

[Figure 11. Formula. Travel time index. 23](#_Toc35779403)

[Figure 12. Formulas. Mathematical Formulation of the Optimization-based SPDHRM Strategy. 24](#_Toc35779404)

[Figure 13. Formulas. Simplified Mathematical Formulation of the Optimization-based SPDHRM Strategy. 24](#_Toc35779405)

[Figure 14. Photo. Data Collection Location on Interstate 35 near Austin, TX. 26](#_Toc35779406)

[Figure 15. Photo. Vehicle Detection and Tracking in Aerial Images. 27](#_Toc35779407)

[Figure 16. Illustration. Sample Trajectory Data Collected on Interstate 35 near Austin, TX. 27](#_Toc35779408)

[Figure 17. Equation. Error in the gap between the lead vehicle and the target vehicle. 27](#_Toc35779409)

[Figure 18. Diagrams. as a function of acceleration (a) Scenario A, and (b) Scenario B. 30](#_Toc35779410)

[Figure 19. Diagrams. Utility and probability density function for human following and AV following. 31](#_Toc35779411)

[Figure 20. Flowchart. Speed decision tree for the SPDHRM system (Mittal, Kim, Mahmassani, & Hong, 2018; Talebpour et al., 2013). 33](#_Toc35779412)

[Figure 21. Formula. Value function for the uncongested regime. 34](#_Toc35779413)

[Figure 22. Formula. Value function for the congested regime. 34](#_Toc35779414)

[Figure 23. Formula. Binary probabilistic regime selection model. 35](#_Toc35779415)

[Figure 24. Formula. Total utility function for the choice of acceleration. 35](#_Toc35779416)

[Figure 25. Formula. Probability density function for the evaluation of drivers’ stochastic response. 35](#_Toc35779417)

[Figure 26. Formula. The intelligent driver acceleration model. 36](#_Toc35779418)

[Figure 27. Formula. Maximum speed of automated vehicles. 36](#_Toc35779419)

[Figure 28. Formula. Acceleration model for automated vehicles. 37](#_Toc35779420)

[Figure 29. Formula. Safe following distance formula. 37](#_Toc35779421)

[Figure 30. Formula. Acceleration of automated vehicles. 37](#_Toc35779422)

[Figure A1. Flowchart. Main algorithm. 48](#_Toc35779423)

[Figure A2. Flowchart. Traffic monitoring algorithm. 51](#_Toc35779424)

[Figure A3. Flowchart. Decentralized SPDHRM strategy evaluation algorithm. 55](#_Toc35779425)

[Figure A4. Flowchart. Centralized SPDHRM strategy evaluation algorithm. 56](#_Toc35779426)

[Figure A5. Flowchart. Optimization-based SPDHRM strategy evaluation algorithm. 57](#_Toc35779427)

[Figure A6. Flowchart. Decentralized SPDHRM strategy implementation algorithm. 70](#_Toc35779428)

[Figure A7. Flowchart. Centralized SPDHRM strategy implementation algorithm. 71](#_Toc35779429)

[Figure A8. Flowchart. Optimization-based SPDHRM strategy implementation algorithm. 73](#_Toc35779430)

List of Tables

[Table 1. Variables Used in the Predictive Model (Elfar, 2019). 23](#_Toc35779431)

[Table 2. Prospect theory model calibration results. 29](#_Toc35779432)

[Table 3. Scenarios for evaluating accuracy of congestion prediction models. 38](#_Toc35779433)

[Table 4. Scenarios for evaluating effectiveness of the speed harmonization strategies. 38](#_Toc35779434)

[Table 5. Accuracy result of the congestion prediction model under different monitoring timestep and section length scenarios. 39](#_Toc35779435)

[Table 6. Performance metrics of different speed control strategies along with various market shares of CVs and AVs. 40](#_Toc35779436)

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 signs at fixed locations to display updated speeds. 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 dynamic 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.

Chapter 1 defines the problem, its objective, and the challenges of traditional system. Chapter 2 provides detailed information of the 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 chapter 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 developed simulation tool are provided. The information essentially can help transportation professionals to incorporate the methodology in other traffic analysis tools.

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. The United States Department of Transportation (USDOT) has led the development, research, and standards making to support these technologies and is currently developing deployment and implementation approaches and guidelines.

In order for CAV applications to be deployed, state and local transportation agencies must first be able to effectively and fully quantify the impacts of such deployments and identify which application best addresses their unique transportation problem. Traffic analysis, modeling, and simulation (AMS) tools provide an efficient means to evaluate transportation improvement projects prior to deployment. Current AMS tools are not well-suited for evaluating CAV applications due to their inability to incorporate vehicle connectivity/communication and automated driving features. To mitigate this gap, Federal Highway Administration (FHWA) has sponsored this project to develop CAV applications/models based on field data to support CAV simulation community. Three CAV applications were developed under this project. They are a lane changing model for light duty CAVs, a combined application model that integrates speed harmonization and coordinated merge, and an improved cooperative adaptive cruise control (CACC) model for light duty CAVs.

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

Purpose of this Model

Speed harmonization, 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 strategy are (Hani S Mahmassani, Elfar, Shladover, & Huang, 2018):

* Improvement in traffic safety by effectively delaying or eliminating traffic breakdown (Talebpour et al., 2013);
* Higher efficiency in fuel consumption and reduction in emissions as a result of effectively suppressing development of oscillation in vehicle speed (Li, Cui, An, & Parsafard, 2014; Wang, Daamen, Hoogendoorn, & Van Arem, 2015; Yang & Jin, 2014); and
* Improvement in traffic efficiency 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 speed harmonization strategy. Traditionally, this strategy was 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: 1) fixed infrastructure sensors provide an incomplete representation of traffic flow dynamics which can significantly diminish the effectiveness of the strategy; 2) relying on fixed road sensors and signs limits the applicability of the speed control under, which significantly affects the strategy performance; 3) it is difficult to develop accurate models that are capable of predicting future traffic state utilizing data from fixed traffic sensors (Elfar, Talebpour, & Mahmassani, 2019, 2020).

Connected vehicles as probe vehicles can monitor their surrounding traffic conditions and communicate that information with the infrastructure and 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, automated vehicles can help to dampen the effect of shockwaves in the traffic flow.

Document Overview

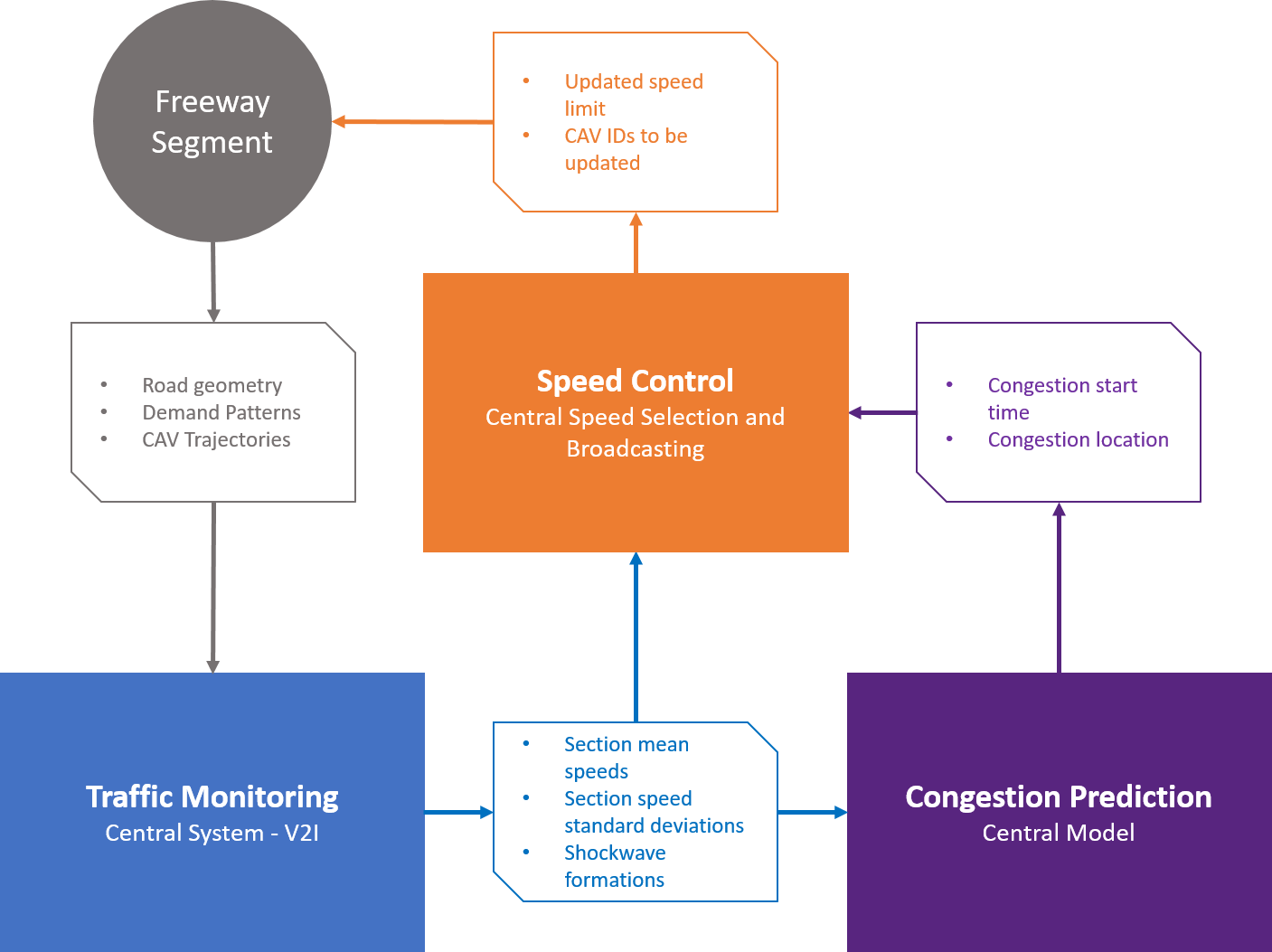
This document will introduce a new speed harmonization model capable of reducing or eliminating shockwaves by utilizing the big stream of data generated by CAVs and the predictive capability of machine learning algorithms. In the following sections, we introduce the model details and the logic behind 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 is concluded with a summary of the findings.

Chapter 2. MODEL DEVELOPMENT AND LOGIC

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

Descriptions of Model Logic

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



Source: Elfar, 2019.

Figure . Diagram. Speed Harmonization Overall Framework.

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 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.

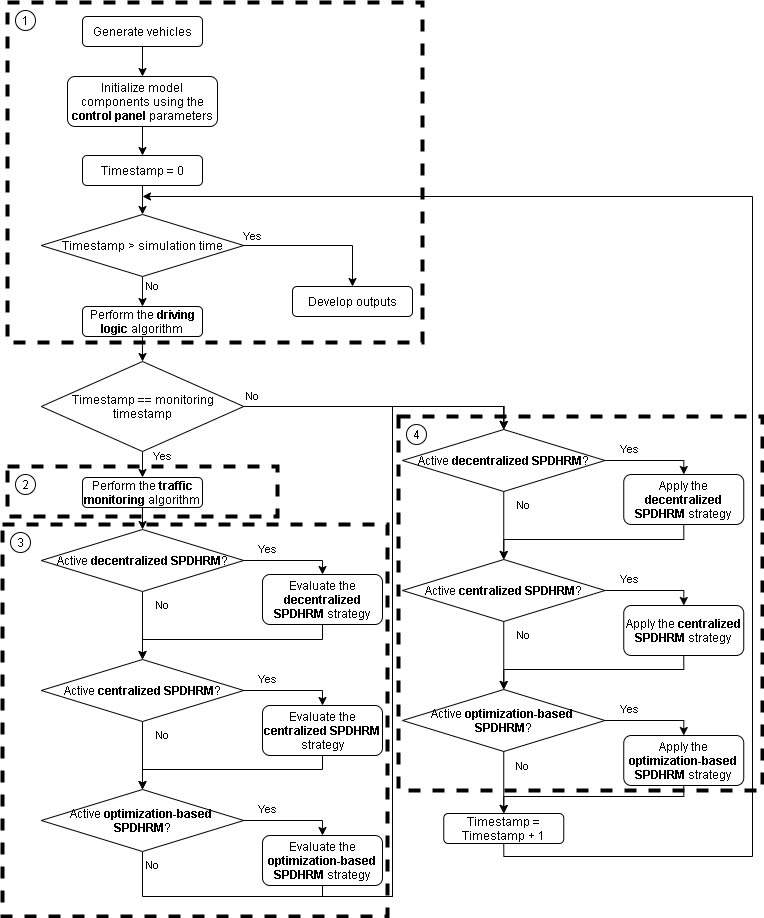
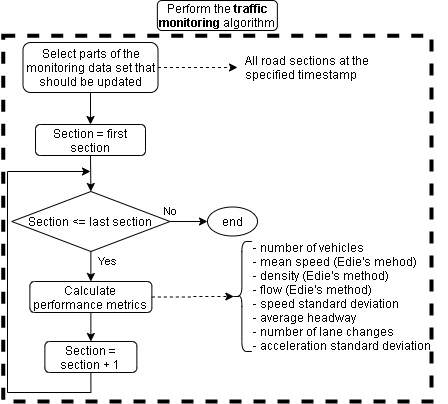
 **Source: FHWA**

Figure . 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 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.

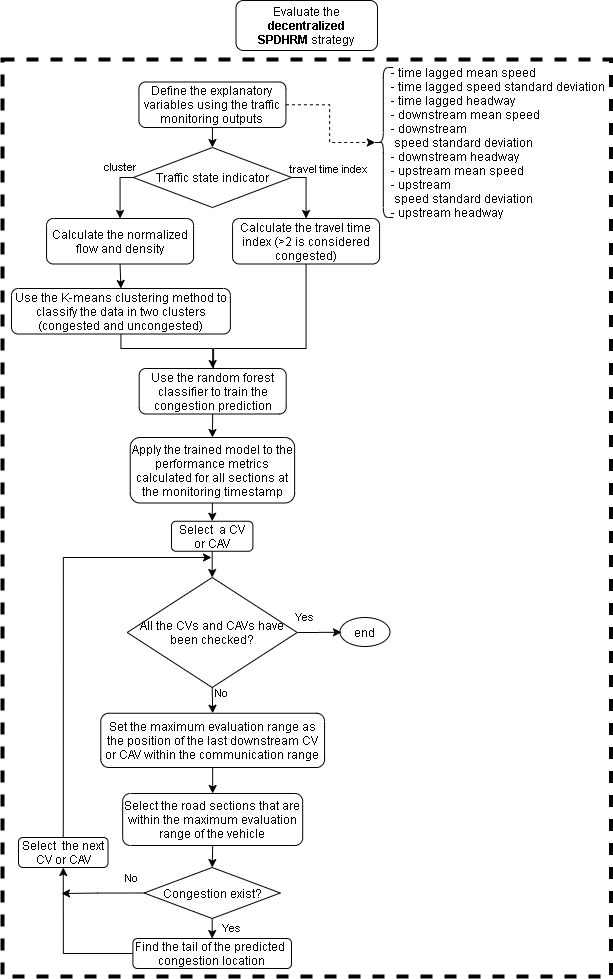


**Source: FHWA**

Figure . Flowchart. Traffic Monitoring Algorithm.

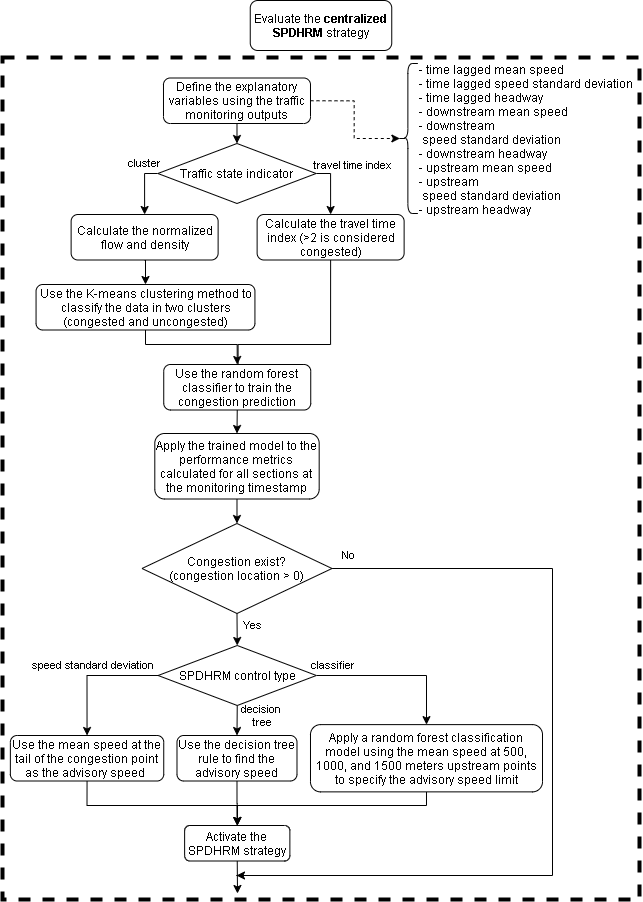
Figure 4 to Figure 6 show the algorithms in the congestion prediction module. Based on the settings defined in the control panel, one of the algorithms is performed: Centralized, decentralized, or optimization-based. The location and time of the congestion along with the advisory speed limit are determined 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).



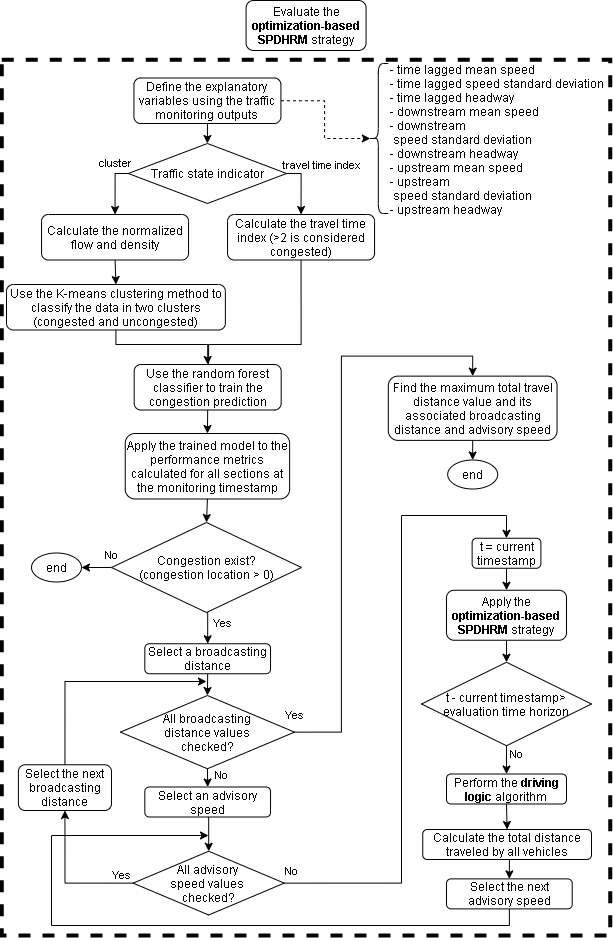
**Source: FHWA**

Figure . Flowchart. Decentralized SPDHRM Strategy Evaluation Algorithm.



**Source: FHWA**

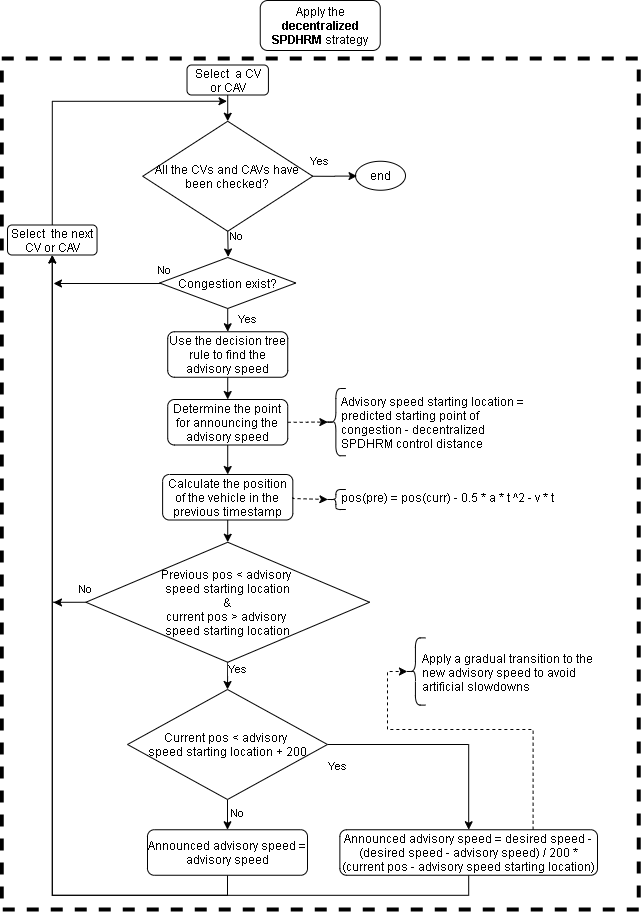
Figure . Flowchart. Centralized SPDHRM Strategy Evaluation Algorithm.



**Source: FHWA**

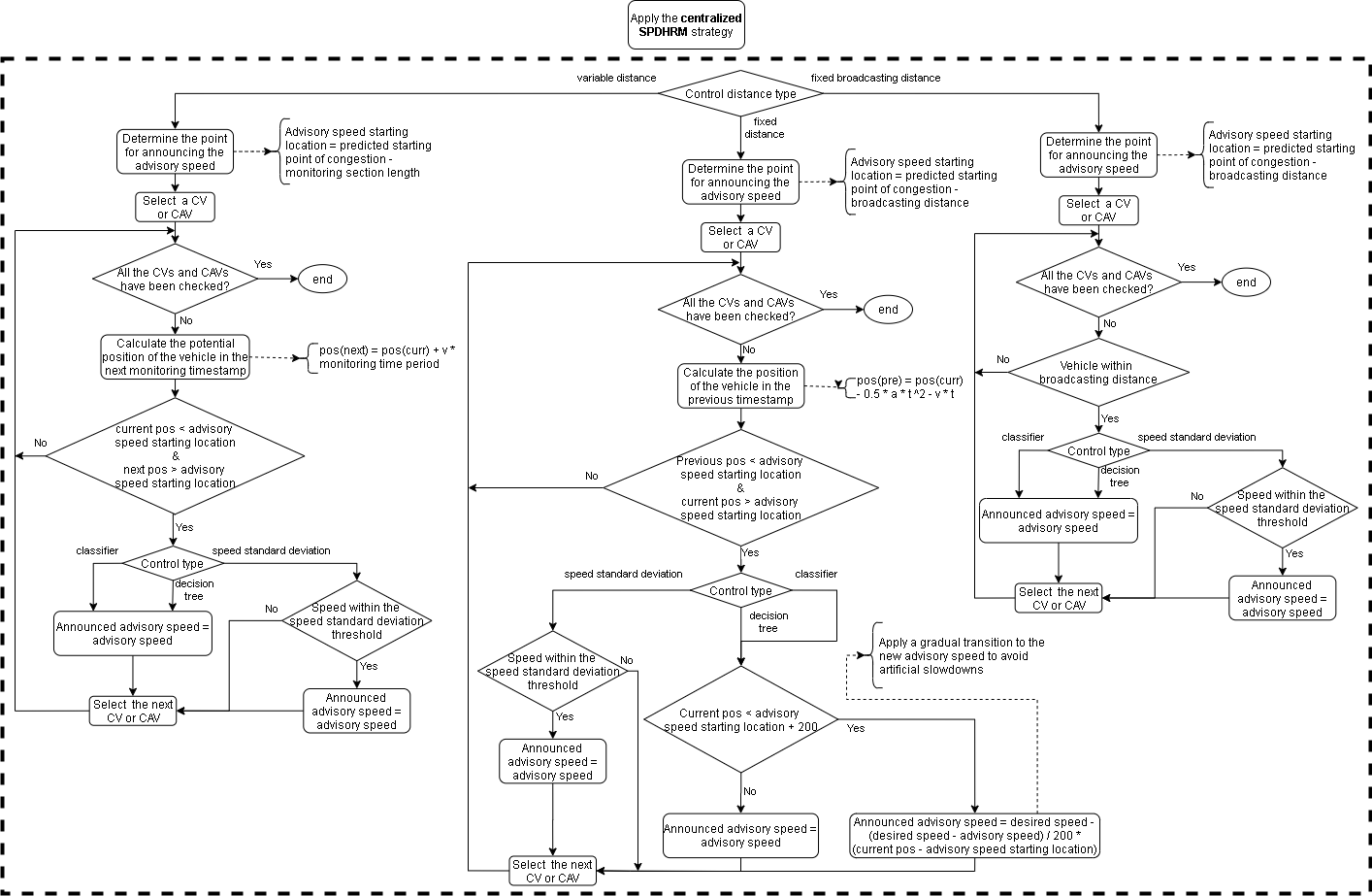
Figure . Flowchart. Optimization-based SPDHRM Strategy Evaluation Algorithm.

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 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.



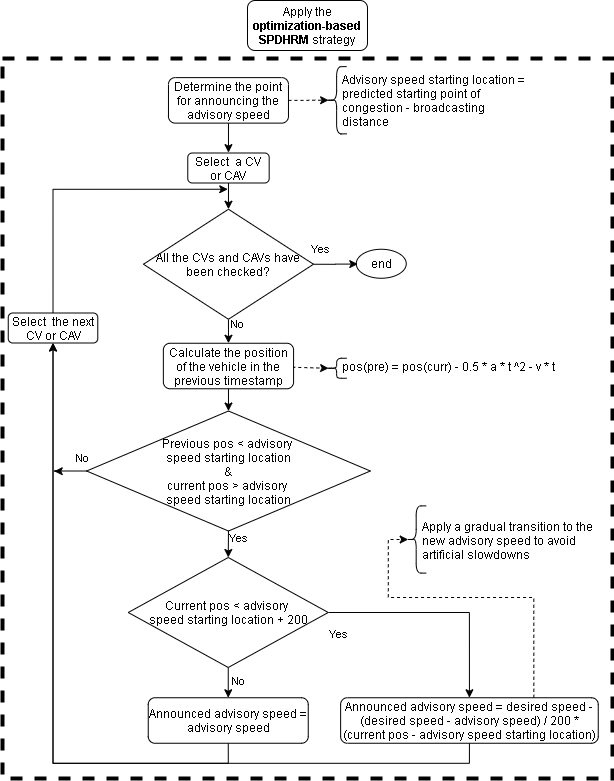
**Source: FHWA**

Figure . Flowchart. Decentralized SPDHRM Strategy Implementation Algorithm.



**Source: FHWA**

Figure . Flowchart. Centralized SPDHRM Strategy Implementation Algorithm.



**Source: FHWA, 2020**

Figure . 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 time step were calculated using Edie’s generalized definitions for individual facilities:

Figure . 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 were could be calculated for each road section at the monitoring timestamp.

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 GPS. In another study by Pattara-Atikom and Peachavanish (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 the predictive models is to identify the traffic state from vehicle trajectories. The traffic state is used as the dependent variable in the machine leaning-based classification model. Travel time index (TTI) and K-means clustering 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 Travel time index 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.

Figure . 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). 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 shockwave formation and propagation. They found that a 30% connectivity level is necessary to accurately identify the shockwave.

For the prediction model that utilizes machine learning algorithms, temporally lagged and spatial distributed variables are developed for the traffic characteristics and performance metrics. As a result, the developed congestion models are temporally lagged models which are a type of time-series models trained to predict current and future values of the dependent variable using explanatory variable values observed in previous time steps. Therefore, by plugging in current values of independent variables, the model would predict 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 timestamp 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 perform slightly better than the model with the neural network algorithm. In this simulation tool, the random forest technique is implement.

Table . Variables Used in the Predictive Model (Elfar, 2019).

|  |  |  |
| --- | --- | --- |
| **Variable** | | **Description** |
|  | **Dependent Variable** |  |
|  | Traffic State | Binary: the state of traffic whether congested or uncongested as identified using the travel time index (TTI) with a threshold above 2 or the k-means clustering algorithm. |
|  | **Explanatory Variables** |  |
|  | Lagged Mean Speed in Current Section | Continuous: the average speed of individual vehicles in the current section, lagged 10, 20, or 30 seconds |
|  | Lagged Mean Speed in Downstream Section | Continuous: the average speed of individual vehicles in the next downstream section, lagged 10, 20, or 30 seconds |
|  | 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 and selects the limit that maximizes traffic speed and by virtue mitigate congestion. The formulation proposed in Figure 12 enables the possibility of assigning a speed limit for each connected vehicle in the system.

Figure . Formulas. Mathematical Formulation of the Optimization-based SPDHRM Strategy.

Where represents the distance traveled by all vehicles in the vehicle set () over an optimization horizon (). The optimal speed of each vehicle as the decision variable varies between a set of minimum () and maximum () 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.

Due to the interactions of the vehicles captured by the car-following, lane-changing behavior, vehicle classes, and traffic control, a simulation-based optimization approach should be adopted in order to find the advisory speed for each vehicle that would collectively result in maximizing the distance traveled by all vehicles. The key limitation of simulation-based optimization problems is the computationally intensive and time-consuming simulations that are associated with finding the optimal solution. To practically solve the optimization problem, the number of decision variables needs to be reduced significantly. 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.

Figure . Formulas. Simplified Mathematical Formulation of the Optimization-based SPDHRM Strategy.

Chapter 3. MODEL CALIBRATION AND VALIATION

In order to calibrate and validate the joint application calibration developed under CAV-AMS Phase II project, two key features have been developed and integrated into Northwestern’s microscopic simulation platform:

* Speed Harmonization: A set of novel speed harmonization algorithms were developed that utilize machine learning to predict the onset of congestion and to activate the speed harmonization in a highway segment. These algorithms also utilize various methods of communicating the updated speed limits to the connected vehicles (automated or human-driven) and non-connected vehicles (automated or human-driven).
* Merge coordination: A couple of algorithms were developed to enable merge coordination in connected and non-connected driving environments. These algorithms aim to prevent shockwave formation in the target lane, even at very small time-headways.

In addition to these models, the simulation platform utilizes several already calibrated and validated car-following and lane-changing models for non-connected human-driven vehicles, connected human-driven vehicles, connected automated vehicles, and non-connected automated vehicles (Talebpour & Mahmassani, 2016).

Since most of the utilized models 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 will be on the calibration of car-following models of human-driven vehicles to capture the effects of interacting with automated vehicles on driver behavior. The validation effort will ensure 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 in automated vehicle research. Unfortunately, the existing vehicle trajectory datasets 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 datasets that cover various levels of automation. Aerial imagery using small unmanned aerial vehicles (UAVs) is an economical and effective solution to collect trajectory data.

To address the shortcomings of the existing vehicle trajectory datasets, a new trajectory dataset was collected on Interstate 35 in Austin, TX (See **Original Photo: © 2019 Google® (See Acknowledgements).**

Figure 14). A platoon of three SAE Level 1 automated vehicles with 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 can be 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 can be estimated for a fixed coordinate system and reference point on the ground. Every video recording is 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 enables 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.

A close up of a map

Description automatically generated

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

Figure . Photo. Data Collection Location on Interstate 35 near Austin, TX.

|  |
| --- |
| A screen shot of a computer  Description automatically generated**(a) Bird’s Eye View** |
| A close up of a computer  Description automatically generated |
| 1. **Vehicle Detection Using Convolutional Neural Network** |
| A picture containing road, car, green, electronics  Description automatically generated |
| 1. **Vehicle Tracking**   **Source: FHWA**  Figure . Photo. Vehicle Detection and Tracking in Aerial Images.  A close up of a curtain  Description automatically generated  **Source: FHWA** |

Figure . Illustration. Sample Trajectory Data Collected on Interstate 35 near Austin, TX.

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 dataset with the simulated behavior based on a set of model parameters. For car-following models, the error will be calculated based on the error in the gap between the lead vehicle and the target vehicle:

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

Where , , and . For lane-changing models, the same process is followed with one key difference; that is the error in gap between the lane-changing vehicle and both 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 car-following/lane-changing model are initialized to random numbers. Each set of these parameters is called a “chromosome” and the total of 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% or less than 0.1% 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. This study will utilize the same method for calibration of car-following and lane-changing models.

In order to calibrate and validate the model, each vehicle trajectory in the dataset will be divided into calibration and validation sets. The calibration set will have about four times more datapoints than the validation set, all selected randomly from the datapoints in the vehicle trajectory dataset. Utilizing the same error function presented in the previous section, the model will be first calibrated using the data in the calibration set. The calibrated model parameters will be then used to simulate the data in the validation set and the results (gap between vehicles) will be compared. The outcome of this calibration and validation process is a set of car-following/lane-changing parameters for each individual vehicle trajectory in the dataset.

As discussed previously, the models utilized in Northwestern’s simulation platform have gone through a similar calibration and validation process based on NGSIM US-101 dataset. Accordingly, this study will only focus on the cases when a human driver interacts with an automated vehicle in the dataset. The focus will be on a human driver following an automated vehicle. Note that Rahmati et al. (2019) showed that there is potential to see significant changes in driver behavior in this case and real-world data will be utilized to quantify these changes.

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

Regarding car-following models of automated vehicles, Northwestern’s simulation framework utilizes the ACC/CACC models developed by Milanés and Shladover (2014). These models were calibrated based on empirical data. Accordingly, the car-following behavior of automated vehicles will not be calibrated again in this study.

Regarding lane-changing models of automated vehicles, these models are designed based on the capabilities and characteristics of Texas A&M’s automated Chevy Bolt EV. Accordingly, any lane-changing trajectory generated by the models can be followed in real-world.

**Calibration and Validation Results**

Table 2 shows the calibration results along with the ANOVA test outcome for the Austin data along with the data collected by Rahmati et al. (2019). Two of the key model parameters show no statistically significant difference (i.e., and ), 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. Figure 18 shows the function for the average values in scenarios A and B. Note that since the average and are almost the same value, functions are almost identical. This figure indicates that drivers were less sensitive to deceleration and they did not feel significant disutility from braking regardless of the deceleration value. In a real-world context, however, drivers seek to travel at their desired speed and minimize their travel time, while avoiding crashes. Therefore, they try to minimize hard braking and favor higher acceleration rates (considering the comfort level). However, in the context of the experiments conducted in this study, drivers did not feel any urgency to reach the end of the test (especially since each test took about 2-3 minutes to finish). Their main focus was on following their leader in a safe manner assuming they are driving in a highway environment. Accordingly, the result (as presented in Figure 18) was expected.

Table . Prospect theory model calibration results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameters** | **Human Following** | | **AV Following** | | **ANOVA Test**  **(*p*-value)** |
| **Mean** | **Standard Deviation** | **Mean** | **Standard Deviation** |
|  | 0.26 | 0.44 | 0.27 | 0.37 | 0.3597 |
|  | 112200.00 | 20564.31 | 82377.78 | 20360.71 | 0.0001 |
|  | 0.68 | 0.66 | 0.68 | 0.64 | 0.9402 |
|  | 4.98 | 2.42 | 5.33 | 2.61 | 0.4707 |

Unlike function parameters, the crash weighing factor, , showed statistically significant difference between the two sets of experiments. Following an AV, human drivers’ behavior resulted in much less compared with the case of following another human driver ( is 36% 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 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 . Finally, is slightly higher for the AV following scenario, indicating more stable driving behavior by the follower. Note that the difference in between Scenario A and B is not statistically significant.

Source: FHWA

Figure 19 translates the subjective utility functions, , into acceleration probability density functions, for various values of relative speed. Accordingly, three cases are illustrated, 1) follower is approaching the leader with a relative velocity of +10 mph (green plots), 2) keeping a constant space headway with a relative velocity of 0 mph (orange plots), and 3) increasing the space headway with the leader with a relative velocity of -10 mph (blue plots). Comparing the two scenarios, the model calibrated based on the AV following results in more stable behavior, while the model calibrated based on the human following results in higher deceleration rates. Overall, the findings of model calibration indicated a clear difference between humans’ AV following and human following behavior.

|  |  |
| --- | --- |
|  |  |
| **(a)**  **Source: FHWA** | **(b)** |

Figure . Diagrams. as a function of acceleration (a) Scenario A, and (b) Scenario B.

|  |  |
| --- | --- |
|  |  |
| **(a-1)** | **(a-2)** |
|  |  |
| **(b-1)** | **(b-2)** |
|  |  |

Source: FHWA

Figure . Diagrams. Utility and probability density function for human following and AV following.

Note: Utility (a-1 and a-2) and probability density function (b-1 and b-2) for human following (a-1 and b-1) and AV following (a-2 and b-2) for V = 10mph, s = 10m, and various values of ΔV (ΔV = -10mph is represented by blue line, V = 0mph is represented by orange line, and V = 10mph is represented by green line). All model parameters are selected based on the calibration results.

Chapter 4. BASIC GUIDANCE ON MODEL IMPLEMENTATION

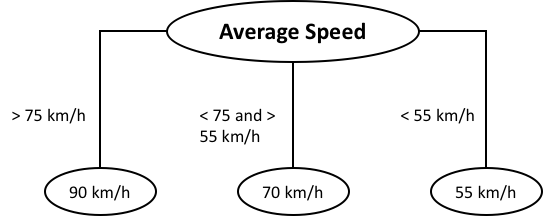
The three modules of the framework could be incorporated in a simulation tool. The information that needs to be transferred to the traffic monitoring at each monitoring timestep is as follow: vehicle position (lane and location in the lane), speed, acceleration, gap and speed difference between the vehicle 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 car-following and lane-changing models in the next simulation steps. The modules are called repetitively at each monitoring timestep.

Besides the input from the simulation tool, each module requires 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 speed harmonization strategy selected by the user. If the centralized method is used, the user should determine:

* Control type: Two types of control type are incorporated in the model, namely decision tree and speed standard deviation. The logic of the decision tree is shown in Figure 20. 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 to disseminate 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 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 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 advised speed limit.



**Source: FHWA**

Figure . Flowchart. Speed decision tree for the SPDHRM system (Mittal, Kim, Mahmassani, & Hong, 2018; Talebpour et al., 2013).

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

The optimization-based speed harmonization strategy requires 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

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 speed harmonization framework. The framework was incorporated in a microsimulation tool developed in Python. The microsimulation tool is a special-purpose platform for simulating mixed traffic conditions on freeways with the possibility of including connected vehicles and autonomous vehicles in the system. In the current case study, the testbed uses a 5-km section of a two-lane highway with an on-ramp which starts at the 2,700th meter of the segment and has a length of 300 meters.

In the simulation platform, distinct car-following models are defined to specify the behavior of each agent: 1) Manually driven vehicles (regular vehicles); 2) Connected vehicles; and 3) Automated vehicles.

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:



Figure . Formula. Value function for the uncongested regime.

Where denotes the value function for the uncongested traffic conditions. and are parameters to be estimated and calibrated and is used to normalize the acceleration. On the other hand, the following formula shows the value function for the congested regime:



Figure . Formula. Value function for the congested regime.

Where denotes the value function for the congested traffic conditions. and are parameters to be estimated and calibrated. At each evaluation time step, 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:



Figure . Formula. Binary probabilistic regime selection model.

Where , , and denote the expected value function, the probabilities 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:



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

Where is the crash probability. Finally, the following probability density function is used to evaluate the stochastic response of the drivers:

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

Where is the sensitivity of choice to the utility .

Connected vehicles are capable of exchanging information with other vehicles and infrastructure-based equipment. The information is exchanges through the vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications networks. As a result, the driver receives information about other connected vehicles as well as updated information containing 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.

Figure . Formula. The intelligent driver acceleration model.

Where is the free acceleration exponent; is the desired time gap; is the maximum acceleration; is the desired deceleration; is the jam distance; and is the desired speed. These parameters need to 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 speed harmonization, could be transferred to the drivers. However, their reaction times would still be like regular drivers.

Automated vehicles 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 car-following behavior of automated vehicles could be specified by a deterministic modeling framework. Talebpour and Mahmassani (2016) developed a car-following model for automated vehicles 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 automated vehicles in order to generate the input data for the acceleration model.

Considering the sensor range and limitations in accuracy, the automated vehicles must 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 automated vehicle should 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 requires immediate reaction of the automated vehicle, the maximum safe speed can be calculated using the following equations:

Figure . Formula. Maximum speed of automated vehicles.

Where n and n-1 represent the automated vehicle and its leader, respectively; is the position of vehicle n; is the length of vehicle n; is the speed of vehicle n; is the reaction time of vehicle n; and 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 automated vehicle at every decision point:

Figure . Formula. Acceleration model for automated vehicles.

Where is the acceleration of vehicle n; and , , and are model parameters that need to be calibrated. is the spacing and is the maximum between the minimum distance (), following distance based on the reaction time (), and safe following distance (). In this study, the minimum distance is set at 2.0 m and is calculated according to the following formula:

Figure . Formula. Safe following distance formula.

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

Figure . Formula. Acceleration of automated vehicles.

Where is a model parameter. Van Arem et al. (2006) suggested using the following values for the model parameters:1, , , and

Design of simulation experiments

We defined two sets of scenarios for the sensitivity analysis. Various market penetration rates of connected and automated vehicles are considered in the scenarios. It is assumed that all vehicles are equipped with communication features 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 3 and Table 4. We simulated each scenario 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 AV), etc.

Table . Scenarios for evaluating accuracy of congestion prediction models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario ID** | **MPR of CV (%)** | **MPR of AV (%)** | **Monitoring Timestep (s)** | **Section length (m)** |
| A1 | 100 | 0 | 10 | 100 |
| A2 | 70 | 30 | 10 | 100 |
| A3 | 30 | 70 | 10 | 100 |
| A4 | 0 | 100 | 10 | 100 |
| A5 | 100 | 0 | 10 | 200 |
| A6 | 70 | 30 | 10 | 200 |
| A7 | 30 | 70 | 10 | 200 |
| A8 | 0 | 100 | 10 | 200 |
| A9 | 100 | 0 | 10 | 500 |
| A10 | 70 | 30 | 10 | 500 |
| A11 | 30 | 70 | 10 | 500 |
| A12 | 0 | 100 | 10 | 500 |
| A13 | 100 | 0 | 20 | 100 |
| A14 | 70 | 30 | 20 | 100 |
| A15 | 30 | 70 | 20 | 100 |
| A16 | 0 | 100 | 20 | 100 |
| A17 | 100 | 0 | 20 | 200 |
| A18 | 70 | 30 | 20 | 200 |
| A19 | 30 | 70 | 20 | 200 |
| A20 | 0 | 100 | 20 | 200 |
| A21 | 100 | 0 | 20 | 500 |
| A22 | 70 | 30 | 20 | 500 |
| A23 | 30 | 70 | 20 | 500 |
| A24 | 0 | 100 | 20 | 500 |
| A25 | 100 | 0 | 30 | 100 |
| A26 | 70 | 30 | 30 | 100 |
| A27 | 30 | 70 | 30 | 100 |
| A28 | 0 | 100 | 30 | 100 |
| A29 | 100 | 0 | 30 | 200 |
| A30 | 70 | 30 | 30 | 200 |
| A31 | 30 | 70 | 30 | 200 |
| A32 | 0 | 100 | 30 | 200 |
| A33 | 100 | 0 | 30 | 500 |
| A34 | 70 | 30 | 30 | 500 |
| A35 | 30 | 70 | 30 | 500 |
| A36 | 0 | 100 | 30 | 500 |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario ID** | **MPR of CV (%)** | **MPR of AV (%)** | **Speed Control Strategy** | **Broadcasting distance (m)** |
| B1 | 100 | 0 | Centralized | 1000 |
| B2 | 70 | 30 | Centralized | 1000 |
| B3 | 30 | 70 | Centralized | 1000 |
| B4 | 0 | 100 | Centralized | 1000 |
| B5 | 100 | 0 | Centralized | 500 |
| B6 | 70 | 30 | Centralized | 500 |
| B7 | 30 | 70 | Centralized | 500 |
| B8 | 0 | 100 | Centralized | 500 |
| B9 | 100 | 0 | Optimization-based | --- |
| B10 | 70 | 30 | Optimization-based | --- |
| B11 | 30 | 70 | Optimization-based | --- |
| B12 | 0 | 100 | Optimization-based | --- |

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 by 80% of the simulation data. We evaluated the model performance by comparing the status predicted by the model and the status determined by TTI on the remaining 20% of the data. Table 5 shows the result of analyzing the first set of scenarios. We calculated the average and standard deviation of the prediction model accuracy for each pair of monitoring timestamp and section length. The accuracy is reported for 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 timestamp and section length, we calculated the average accuracy for the four different scenario types created by changing the market penetration rate of CVs and AVs. It is assumed that all the vehicles are equipped with connectivity feature. Therefore, there are minor differences between the accuracy values among various monitoring timestamp and section length pairs. The source of difference is the car-following model used for each vehicle class. Among the analyzed scenarios, the scenario with a monitoring timestep of 20 seconds and section length of 200 meters 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 . Accuracy result of the congestion prediction model under different monitoring timestep and section length scenarios.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Monitoring Timestep (s)** | **Section length (m)** | **Average overall accuracy (%)** | **Average congested accuracy (%)** | **Average uncongested accuracy (%)** | **Standard deviation overall accuracy (%)** | **Standard deviation congested accuracy (%)** | **Standard deviation uncongested accuracy (%)** |
| 10 | 100 | 97.14 | 89.56 | 98.96 | 0.11 | 0.40 | 0.08 |
| 10 | 200 | 97.15 | 89.80 | 98.93 | 0.15 | 0.51 | 0.10 |
| 10 | 500 | 97.14 | 89.63 | 98.94 | 0.09 | 0.50 | 0.09 |
| 20 | 100 | 97.15 | 89.71 | 98.93 | 0.11 | 0.57 | 0.08 |
| 20 | 200 | 97.19 | 89.94 | 98.94 | 0.09 | 0.38 | 0.07 |
| 20 | 500 | 97.14 | 89.75 | 98.93 | 0.10 | 0.43 | 0.10 |
| 30 | 100 | 97.19 | 89.85 | 98.95 | 0.10 | 0.49 | 0.09 |
| 30 | 200 | 97.12 | 89.60 | 98.93 | 0.11 | 0.49 | 0.12 |
| 30 | 500 | 97.17 | 89.88 | 98.91 | 0.09 | 0.42 | 0.07 |

We incorporated the selected pair from the previous sensitivity analysis into the speed control module to compare the effectiveness of various speed harmonization strategies. Table 6 shows the result of scenarios involving different speed control strategies and various market penetration rates of CVs and AVs. The scenarios where no speed harmonization strategy have been applied are used as a reference to assess the effectiveness of the examined speed control strategies. As could be seen, the effect of automated vehicles 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 control system 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 . Performance metrics of different speed control strategies along with various market shares of CVs and AVs.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed Control Strategy** | **Broadcasting distance (m)** | **MPR of CV (%)** | **MPR of AV (%)** | **Average speed (kph)** | **Speed standard deviation (kph)** | **Average travel time (s)** | **Travel time standard deviation (s)** |
| No SPDHRM | --- | 0 | 100 | 99.97 | 1.01 | 180.00 | 0.01 |
| No SPDHRM | --- | 30 | 70 | 92.15 | 17.00 | 174.46 | 55.21 |
| No SPDHRM | --- | 70 | 30 | 67.58 | 25.99 | 178.32 | 133.34 |
| No SPDHRM | --- | 100 | 0 | 69.93 | 24.34 | 262.56 | 40.46 |
| Centralized | 1000 | 0 | 100 | 99.97  (.00)\* | 1.01  (.00) | 180.00  (.00) | .01  (.00) |
| Centralized | 1000 | 30 | 70 | 90.60  (-1.68) | 17.59  (3.46) | 178.80  (2.49) | 53.68  (-2.77) |
| Centralized | 1000 | 70 | 30 | 67.06  (-.78) | 26.32  (1.28) | 179.70  (.77) | 135.78  (1.83) |
| Centralized | 1000 | 100 | 0 | 66.98  (-4.21) | 26.33  (8.20) | 267.96  (2.06) | 42.29  (4.50) |
| Centralized | 500 | 0 | 100 | 99.97  (.00) | 1.01  (.00) | 180.00  (.00) | .01  (.00) |
| Centralized | 500 | 30 | 70 | 91.20  (-1.03) | 17.16  (.94) | 172.66  (-1.03) | 59.16  (7.16) |
| Centralized | 500 | 70 | 30 | 67.03  (-.83) | 26.31  (1.23) | 178.94  (.34) | 134.56  (.91) |
| Centralized | 500 | 100 | 0 | 65.70  (-6.04) | 27.21  (11.81) | 273.25  (4.07) | 40.39  (-.19) |
| Optimization-based | --- | 0 | 100 | 99.97  (.00) | 1.01  (.00) | 180.00  (.00) | .01  (.00) |
| Optimization-based | --- | 30 | 70 | 92.90  (.81) | 15.62  (-8.12) | 176.16  (.97) | 50.12  (-9.22) |
| Optimization-based | --- | 70 | 30 | 67.33  (-.38) | 26.16  (.67) | 182.34  (2.26) | 136.53  (2.39) |
| Optimization-based | --- | 100 | 0 | 69.91  (-.02) | 24.34  (.01) | 267.12  (1.74) | 32.71  (-19.17) |

\*Numbers in parenthesis represent % changes in the performance metric of a scenario compared to the equivalent scenario where the speed harmonization algorithm is deactivated.

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 should be analyzed to reach a comprehensive conclusion. As the next step, we will incorporate scenarios with manually driven 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 speed harmonization 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 speed harmonization 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 speed harmonization 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 speed harmonization 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 speed harmonization and ramp metering can outperform the implementation of individual strategies.

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.

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.

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.

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.

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.

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.

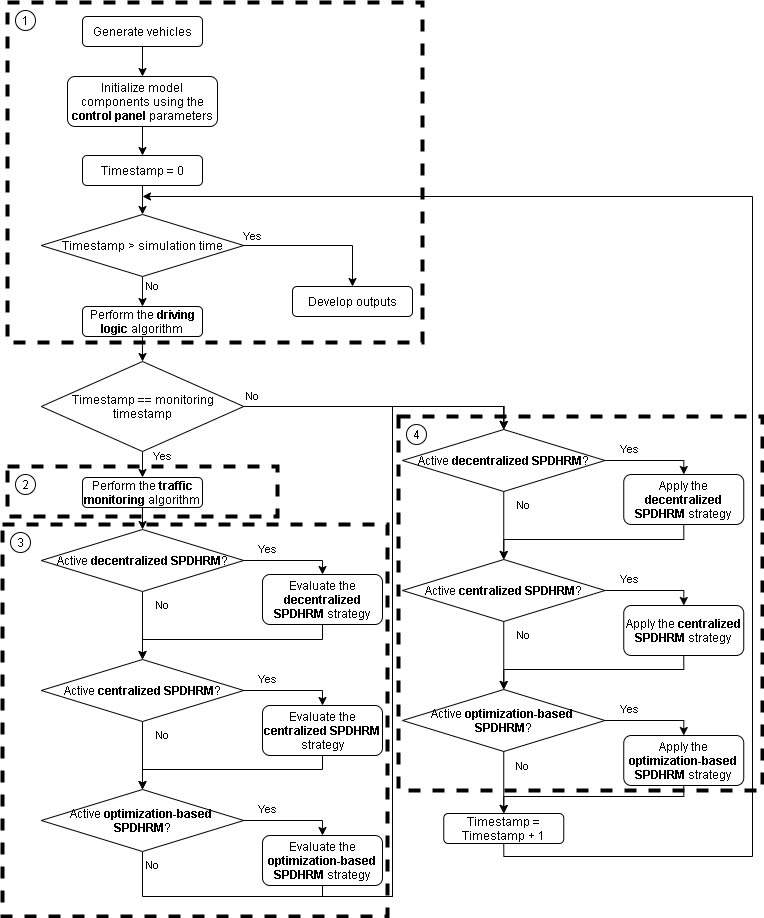
**Source: FHWA**

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**

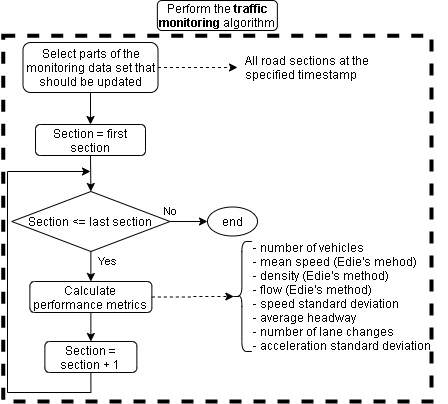
The vehicle trajectories are the outputs that are written into a csv file.

**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 A. Flowchart. Traffic monitoring algorithm.

*init\_tm\_array*

**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.

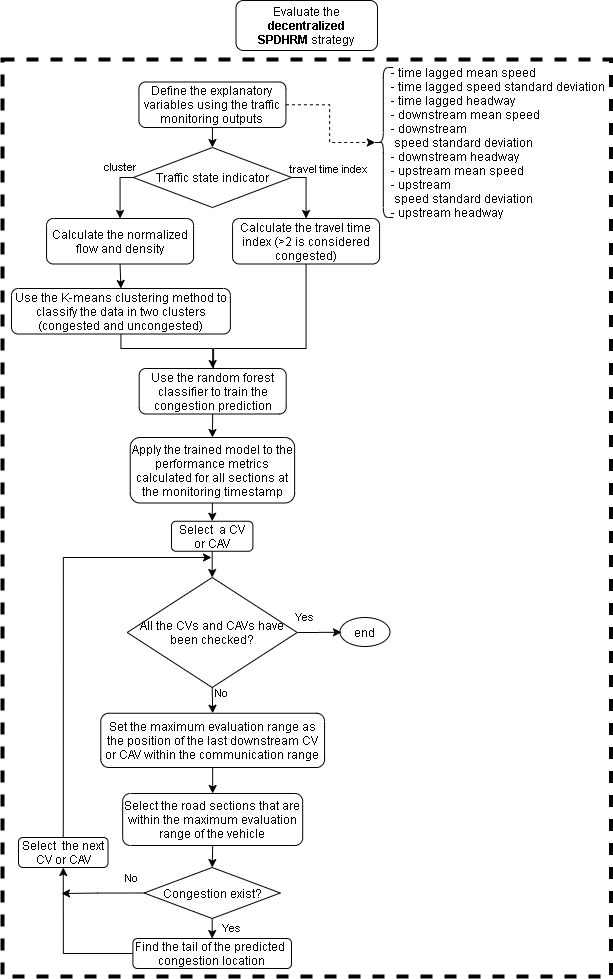
**Pseudo-code**

* 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

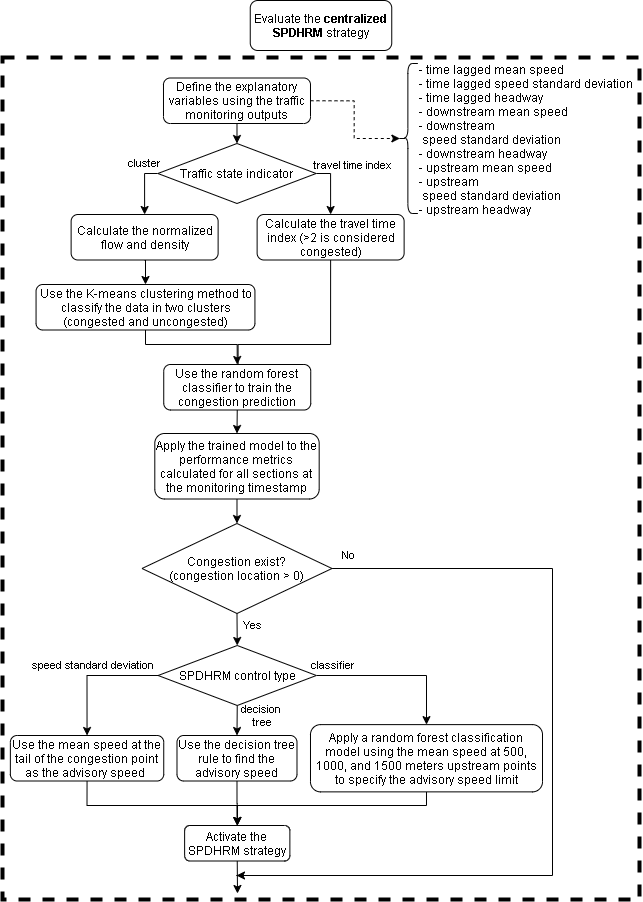
Figure A3 to Figure A5 show the algorithms in the congestion prediction module. Based on the settings defined in the control panel, one of the algorithms is performed: Centralized, decentralized, or optimization-based. The location and time of the congestion along with the advisory speed limit are determined 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).



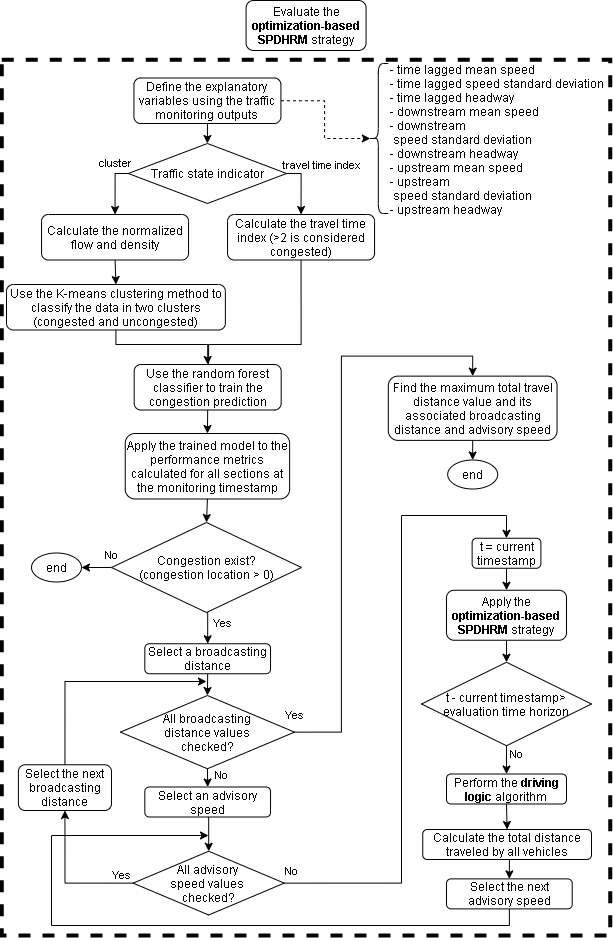
**Source: FHWA**

Figure A. Flowchart. Decentralized SPDHRM strategy evaluation algorithm.



**Source: FHWA**

Figure A. Flowchart. Centralized SPDHRM strategy evaluation algorithm.



**Source: FHWA**

Figure A. 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 ith 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.

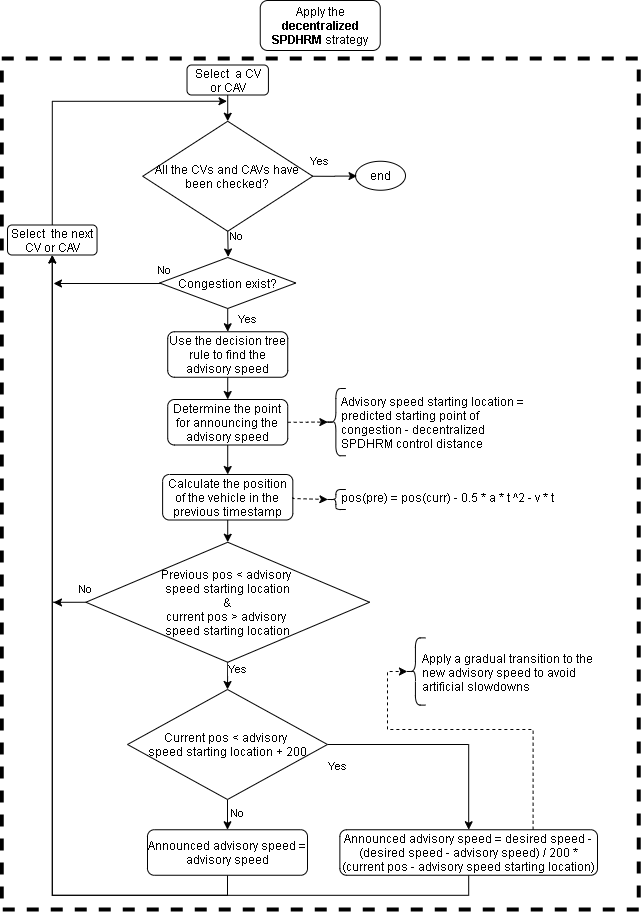
optimal\_speed: optimal advisory speed limit.

**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

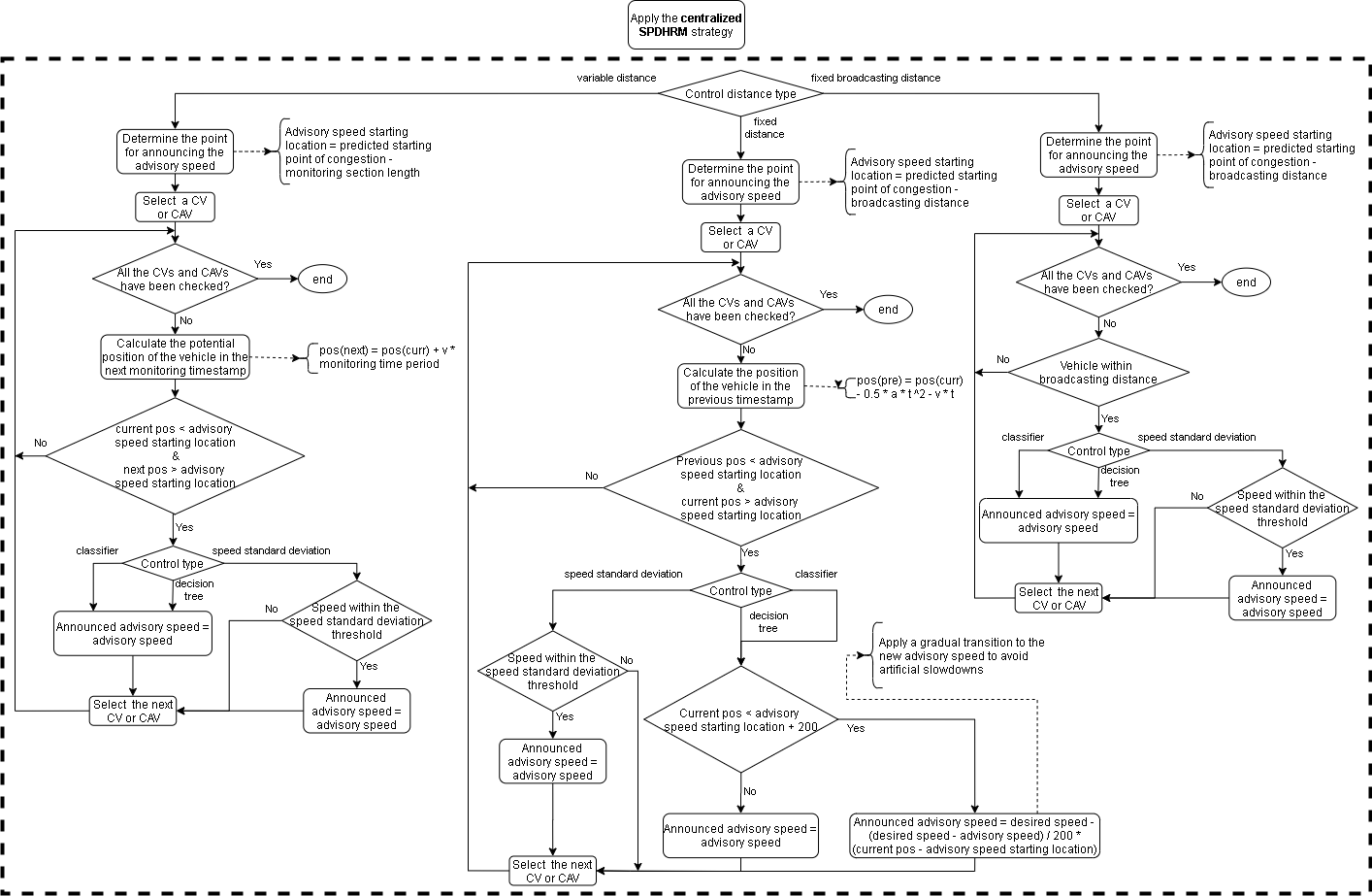
A.4. Speed Control Module

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.



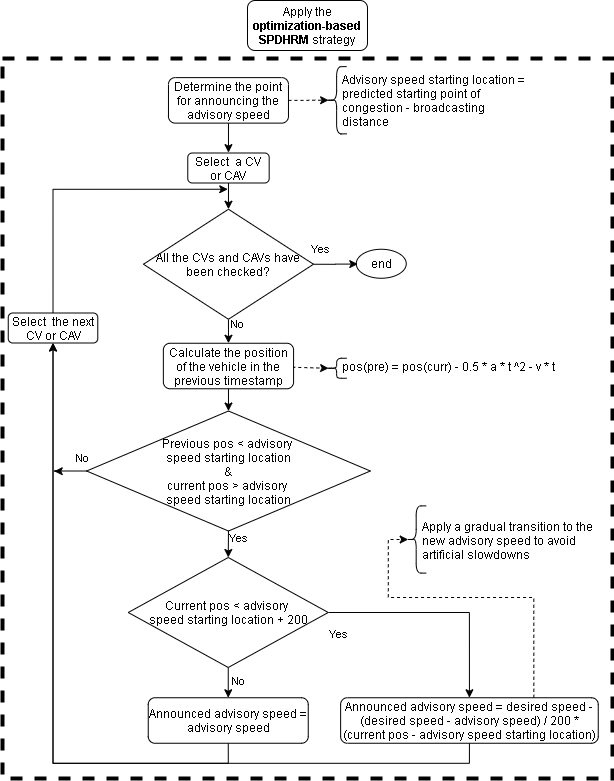
**Source: FHWA**

Figure A. Flowchart. Decentralized SPDHRM strategy implementation algorithm.



**Source: FHWA**

Figure A. 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