Managing Driving Modes in Automated Driving Systems

David Ríos Insua, William N. Caballero, Roi Naveiro,

To cite this article: David Ríos Insua, William N. Caballero, Roi Naveiro, (2022) Managing Driving Modes in Automated Driving Systems. Transportation Science 56(5):1259-1278. https://doi.org/10.1287/trsc.2021.1110

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article’s accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2022 The Author(s)

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org
Managing Driving Modes in Automated Driving Systems

David Rios Insua, William N. Caballero, Roi Naveiro

Abstract. Current technology is unable to produce massively deployable, fully automated vehicles that do not require human intervention. Given that such limitations are projected to persist for decades, scenarios requiring a driver to assume control of a semiautomated vehicle, and vice versa, will remain a feature of modern roadways for the foreseeable future. Herein, we adopt a comprehensive perspective of this problem by simultaneously considering operational design domain supervision, driver and environment monitoring, trajectory planning, and driver-intervention performance assessment. More specifically, we develop a modeling framework for each of the aforementioned functions by leveraging decision analysis and Bayesian forecasting. Utilizing this framework, a suite of algorithms is subsequently proposed for driving-mode management and early warning emission, according to a management by exception principle. The efficacy of the developed methods is illustrated and examined via a simulated case study.

1. Introduction

Automated driving systems (ADSs) constitute a major technological innovation that will reshape transportation. Recent breakthroughs in Artificial Intelligence, coupled with advances in computational hardware, enable cutting-edge perception and control algorithms to be executed in real time. Despite these advances, it is widely accepted that, although future roadways will almost certainly be populated with fully automated vehicles, the massive deployment of such ADSs is decades away (Hillier, Wright, and Damen 2014, Mahmassani 2016). In the interim, the proportion of semiautomated vehicles on global roadways is expected to increase.

Given these projections, it is important to recall the six-level Society of Automotive Engineers (SAE) driving-automation taxonomy (Society of Automobile Engineers 2018). Therein, level 0 describes vehicles with no automated capacity, and the remaining levels describe vehicles with increasing automated features that culminate in fully automated, level-5 ADSs. In these vehicles, the ADS retains control under all road conditions; no design restrictions exist. Conversely, when operational design domain (ODD) (Czarnecki 2018) conditions are exceeded in level-3 and -4 vehicles, the ADS may ask the driver to take control via a Request-to-Intervene (RtI) command.

Provided that recent estimates indicate semiautomated level-3 and -4 ADSs will constitute 25% of the global market by 2040, traffic in the coming decades will be a heterogeneous mix of manual and automated vehicles. Therefore, until level-5 vehicles predominantly populate global roadways, RtI decisions and their management will remain a crucial safety-related issue (Caballero, Rios Insua, and Banks 2021). However, relatively few studies focus on the RtI decision.
McCall et al. (2019) provide a taxonomy of driving-mode transitions (e.g., scheduled, driver- or system-initiated emergency) and discuss its relation to the SAE automation taxonomy. Other authors have examined the effect of HMIs on RTIs (e.g., Walch et al. 2015 and Eriksson and Stanton 2017), but the algorithmic specifics related to the management of driving modes are less developed. Thus, to ensure the safe operation of level-3 and -4 ADSs, this paper proposes an integrated model for RTIs within realistic, dynamic environments.

The temporal evolution of the driving environment, coupled with its repeated sampling by the ADS sensors, implies that decision analysis (French and Insua 2000) and Bayesian forecasting (West and Harrison 2006) are ideal methods for our problem: These methodological constructs are foundational to making driving-mode management decisions. Their utilization allows for an explicit representation of state uncertainty, thereby reducing the effect of error propagation in ADS operations and decision support. Although similar tools have been utilized for individual ADS management tasks (e.g., see the driver-intention modeling work of Li et al. 2014), we jointly leverage decision analysis and Bayesian forecasting for several interrelated ADS forecasting and decision functions. In this manner, we build upon the foundational work of the AutoMate Consortium (2019), providing tailored statistical tools that may be implemented within a broader human-machine interface (e.g., see Castellano et al. 2018).

Herein, we formally define a set of probabilistic models for the core decision-support modules required in driving-mode management decisions. These modules are subsequently leveraged to develop several control algorithms based on expected utility maximization. A novel simulation environment is also developed to evaluate this framework. Testing illustrates the efficacy of our approach and also uncovers several tradeoffs associated with control algorithm parameterization. Finally, this empirical evaluation also serves to highlight several decision-support dilemmas that would require further ethical and regulatory analysis of a qualitative nature.

In Section 2, we begin by defining models for the decision-support modules required in our approach to driving-mode management. Namely, we present models for operational design domain supervision, environment- and driver-state monitoring, driving-mode assessment, trajectory planning, and driver-intervention performance assessment. Subsequently, in Section 3, we develop a suite of decision-support algorithms to transition between various driving modes and emit relevant warnings based on these models. These algorithms allow the ADS to reason about the encountered uncertain environment, but output a deterministic best response. In Section 4, we set forth a simulation experiment to examine the efficacy of our approach; the results highlight several decision-support dilemmas arising in this domain. Notably, we observe a fundamental dilemma in driving-mode management, wherein the ADS must decide between transferring control to a distracted driver or retaining control when its ODD is about to be exceeded. Finally, Section 5 provides a discussion of our results and explores avenues of future inquiry.

2. Decision-Support Modules for Driving-Mode Management

Consider a level-3 or -4 ADS programmed with several driving modes that uses a Human Machine Interface (HMI) to communicate with the driver; see the AutoMate Consortium (2019) for related prototypes. Example driving modes include automated, manual, and emergency, wherein the ADS drives itself, a person operates the vehicle, and the ADS performs evasive maneuvers to ensure safety, respectively. Utilizing a variety of sensors, the ADS captures data from the driver and the environment for various tasks (e.g., driving-mode selection). In this section, models for core ADS management functions are presented, including ODD supervision; environment monitoring (EM); driver-state monitoring (DSM); trajectory planning; driving-mode assessment (DMA); and, finally, driver-intervention performance assessment (DIPA). Collectively, these models constitute distinct decision-support modules from which information is derived and utilized in subsequent sections to manage transitions between driving modes, emit warnings, and perform emergency procedures.

To better convey the function of each module, Figure 1 summarizes the interrelationships between them. The environmental and DSM modules observe the state at time $t$ and forecast future states through time $t + k$. Subsequently, the ADS planned trajectory is updated by the corresponding module. With this information, relevant warnings are issued, and, if necessary, an RtI is issued. If an RtI is executed, the driver’s behavior is assessed via the DMA model leveraged within the DIPA module. Relevant information is fed forward, and the process repeats for time $t + 1$. If the ODD module’s forecast exceeds established tolerances, then automated mode is abandoned. Formal algorithms prescribing these interactions are the focus of Section 3.

Based on these interrelationships, it is apparent that each module must account for the dynamic features of the driving environment to ensure safe operations. Predicting departures from the status quo enables a more proactive and cautious approach to ADS management. Therefore, utilizing the models developed in this section, we prescribe thresholds with associated computational definitions for undesirable or impermissible system states. These characterizations allow...
for the definition of relevant early warnings that effectively account for the existing uncertainty. Moreover, the models underlying the system’s dynamics further facilitate DMA and DIPA by enabling the use of the decision-analysis techniques described herein.

To increase predictive accuracy, the modules utilize a Bayesian approach to predict future behavior based upon previous observations, implying that the ADS sensors underpin each model’s performance. More formally, letting $D_t$ designate the observational data collected up to time $t$, the ADS utilizes this information to schedule decisions a few steps ahead—for example, $k = 10$ time intervals of 0.5 seconds. The number and length of intervals typically depend on traffic and driver conditions, as well as on the algorithms’ computational demands; at a minimum, their values should cover the driver’s reaction time with an additional safety buffer. Conceptually, the number of intervals and their length can vary according to the underlying conditions, but, in our discussion, we assume, without loss of generality, that $k$ is fixed.

Finally, the notation utilized within this manuscript is summarized in Table 1. The use of bold font is reserved for multivariate variables. Unless otherwise stated, uppercase font is reserved for random variables, whereas lowercase is used for their realizations. Likewise, we reserve (·) for vectors and [] for matrices. When convenient, sequences of variables over time are denoted with condensed notation—for example, $y_{t:2} = (y_t, y_{t+1}, \ldots, y_2)$. With the exception of the trajectory and RTI decisions (i.e., $z_t$ and $d_t$), the variables are used to reason about uncertain quantities.

### 2.1. Operational Design Domain Supervision

The ODD constrains the conditions under which an ADS can operate in automated mode (Czarnecki 2018). The ODD typically references the road state (e.g., traffic volume, weather, or visibility conditions); the behavior of the ADS (e.g., speed limits); and the vehicle’s state (e.g., tire inflation level). Assume that quantities in each of these categories are checked at time $t$ through three blocks of variables—that is, $g^1_t$, $g^2_t$, and $g^3_t$, respectively. These blocks of variables are concatenated into a single vector such that $g_t = (g^1_t, g^2_t, g^3_t) \in \mathcal{G}$, where $\mathcal{G}$ represents the prespecified ODD.

The evolution of some of the ODD variables can be accommodated by deterministic physical models. For example, given the current position, speed, and acceleration, we can forecast the ADS’s location and, consequently, whether it will stay within its lane over the next $k$ time periods. However, other variables require stochastic models—for example, headway distances are dictated by the uncertain behavior of neighboring drivers. Therefore, to maintain generality, we shall use probabilistic models to forecast all ODD variables. A flexible strategy for this utilizes state space models (West and Harrison 2006) such that

$$G_t = \phi_t(F_t, V_t), \quad F_t = \gamma_t(F_{t-1}, W_t),$$

where $G_t$ are the observable ODD variables (with realizations $g_t$); $F_t$ are unobservable state variables; $V_t$ and $W_t$ are vectors of random noises; and, $\phi_t$ and $\gamma_t$ are observation and state functions, respectively. Given the short time intervals considered herein, an important example is the local level model (i.e., the random walk plus noise): $G_t = F_t + V_t, F_t = F_{t-1} + W_t$. 

---

*Figure 1. Incorporation of DIPAs into the RTI Decision Process*
Table 1. Description of Notation

| Symbol | Definition |
|--------|------------|
| $g_i$  | Prespecified ODD in which automated driving is viable |
| $G_i$  | Observable ODD variables where $G_i = (G_i^1, G_i^2, G_i^3)$ |
| $F_t$  | Unobservable ODD state variables |
| $V_i, W_i$ | Random noise vectors |
| $Y_i$  | Exogenous, environmental conditions |
| $\theta_i$ | Latent driver state |
| $x_i$  | Sensor outputs from driver-state monitoring system |
| $\gamma_i$ | Sequence of environmental conditions over $k$ time intervals |
| $\delta_i$ | Sequence of driver states over $k$ time intervals |
| $z_i$  | ADS trajectory plan over $k$ time intervals, $z_i = \{z_{i1}, z_{i2}, \ldots, z_{ik}\}$ |
| $d_i$  | Driving mode $i$ (e.g., manual or automated) |
| $q_1$  | Forecasted probability of exceeding ODD |
| $q_2$  | Forecasted probability of abnormal ODD evolution |
| $q_3$  | Forecasted probability of abnormal environmental behavior |
| $q_4$  | Forecasted probability of undesirable driver state |
| $q_5$  | Forecasted probability of abnormal, observed driver behavior |
| $q_6$  | Forecasted probability of poor driver Rt performance |
| $\omega_{j,c}$ | Critical warning threshold for forecasted probability, $j \in \{1, \ldots, 6\}$ |
| $\omega_{j,s}$ | Standard warning threshold for forecasted probability, $j \in \{1, \ldots, 6\}$ |
| $u(\cdot)$ | Utility function for ADS behavior under current conditions |
| $\psi(\cdot)$ | Expected utility operator |
| $\phi(\cdot)$ | Observation function in state space model |
| $\gamma(\cdot)$ | State function in state space model |
| $D_i$  | Observational data received up to time $t$ |

Standard posterior predictive computations (West and Harrison 2006) allow us to compute the posterior $p(F_t | D_t)$ once $g_i$ is observed. From it, we recursively define, for $i = 1$ to $k$,

$$p(F_{t+i} | D_t) = \int p(F_{t+i} | F_{t+i-1}) p(F_{t+i-1} | D_t) dF_{t+i-1}.$$ 

Based on such distributions, we can construct the forecasting models for the observable ODD variables, for $i = 1$ to $k$, via

$$p(G_{t+i} | D_t) = \int p(G_{t+i} | F_{t+i}) p(F_{t+i} | D_t) dF_{t+i}.$$ 

Although the forecasting of some ODD variables may be difficult (e.g., weather), state space models have proved an effective means of doing so (e.g., see Dong et al. 2011 and de Chalendar and Glynn 2021). The fidelity of these forecasts can be increased if $D_t$ includes information from vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) systems as well.

**2.1.1. ODD-Monitoring Warnings.** Based on the forecasting model $p(G_{t+i} | D_t)$, the ADS may determine whether the probability of exceeding the ODD—that is, $q_1 = Pr(G_{t+i} \notin g_i | D_t)$—is too great. In cases wherein the vehicle will likely exceed such limits, the ADS should issue an alert, and the automated mode should be abandoned. For the ODD-monitoring module, and within subsequent modules discussed later, we introduce two levels of alerts (critical and standard warnings) with respective thresholds $\omega_{1,c} > \omega_{1,s}$. This implies that if $q_1 > \omega_{1,c}$, the ADS issues a critical warning; otherwise, if $q_1 > \omega_{1,s}$, a standard warning is issued.

Moreover, a warning may also be issued when an unexpected change in the observable ODD variables is detected. For example, if $q_2 = Pr([G_{t+i}] > \{g_{t+i}\} | D_t)$ is sufficiently small, wherein $| \cdot |$ is an appropriate norm, the ADS may alert the driver of an unexpected change in ODD compliance. Because the observed conditions were assessed unlikely, the driver may need to be alerted to ensure safe operations. Thresholds $\omega_{2,c} > \omega_{2,s}$ can be utilized to emit critical and standard warnings, as previously described.

**2.2. Environment Monitoring**

Environment monitoring refers to highly dynamic conditions that need to be predicted in ADS operations. Whereas some of these conditions coincide with the road-state ODD variables (i.e., $g_i$), more stable conditions (e.g., light) are not typically considered environment conditions. Herein, we assume there are $p$ exogenous, environmental conditions $Y_i = (Y_{i1}, \ldots, Y_{ip})$ of interest in a driving scene. These variables capture highly dynamic properties of the environment relevant to ADS operations. For example, such conditions may reflect the presence of objects, people, animals, or other vehicles in the driving scene. More formally, the environment variables are relevant to the semantic (i.e., the driving scene identification) and prediction
(i.e., the driving scene’s evolution) layers in McAllister et al. (2017).

We consider a model \( p(Y_{t+1} \mid Y_t) \) that describes the predictive evolution of the environmental variables between two consecutive time steps. As in Section 2.1, based on the posterior \( p(Y_t \mid D_t) \), we recursively derive, for \( i = 1, 2, \ldots, k \), the corresponding predictive distributions through

\[
p(Y_{t+i} \mid D_t) = \int p(Y_{t+i} \mid Y_{t+i-1})p(Y_{t+i-1} \mid D_t)\,dY_{t+i-1}.
\]

The performance of this model is critically linked to the reliability of the ADS’s sensors. Although current technology is, at times, limited in this regard (e.g., light detection and ranging sensors in rainstorms), we assume herein that the sensors’ input faithfully capture present conditions. In subsequent sections, we discuss the relaxation of this assumption.

### 2.2.1. EM Warnings.

Based on the forecasting model \( p(Y_{t+k} \mid D_t) \), once we have observed \( Y_{t+k} \), we may compute whether \( q_i = \Pr(Y_{t+k} > y_{t+k} \mid D_t) \) is sufficiently small for a given norm \(| \cdot |\). If this is the case, the ADS should warn the driver of unlikely environmental behavior, suggesting that a sudden change may have occurred. Such alerts are provided to increase driver awareness in potentially hazardous situations. As before, we assume there exist two warning levels triggered by distinct thresholds \((\omega_{3c} > \omega_{3b})\).

### 2.3. Driver-State Monitoring

To effectively predict how a driver will react to an Rtl, the ADS must understand the driver’s ability to undertake it (Koesdwipay et al. 2016, Yi et al. 2019a). This is often accomplished via vehicle-oriented (e.g., acceleration or driving path) or driver-oriented (e.g., eye closure or hand position) approaches (Hecht et al. 2018, Akai et al. 2019). Given the substantial effect of driver behavior on roadway safety (Brookhuis and De Waard 2010, Wang et al. 2020), predictive models have been a focus of recent DSM research (Torres, Ohashi, and Pessin 2019; Yi et al. 2019b), with a few notable examples adopting a Bayesian perspective (Agamenonio, Nieto, and Nebot 2011; Straub, Zheng, and Fisher 2014). We build upon this literature by constructing an alternative Bayesian model for DSM. Whereas we develop our forecasting framework assuming multivariate, quantitative driver states, we also explicitly address warning structures for univariate, qualitative states given their recent emphasis in government and academia (Ranney, Garrott, and Goodman 2001; Stutts et al. 2001; Dong et al. 2011).

Let the true driver state be a latent variable \( \theta_t \in \Theta \). At each time step \( t \), the ADS collects \( n \) monitoring variables \( X_t = (X^{(1)}_t, \ldots, X^{(n)}_t) \) from interior-focused ADS sensors (e.g., driver-facing cameras). Although the driver’s state is of primary interest, a subset of sensors may be utilized to observe passenger behavior, which, in turn, can be used as DSM input as well. The associated variables may be continuous or discrete (e.g., head position or blink count). Whereas it is logical to assume that the driver’s state affects their observed behaviors, the driver’s state itself will also be influenced by environment-monitoring variables (e.g., the sudden appearance of an obstacle). Therefore, these values should be taken into account for driver-state monitoring purposes. Under these assumptions, Figure 2 depicts the structure of our dynamic DSM model.

In addition to \( p(Y_{t+1} \mid Y_t) \) discussed in the previous section, we assume the following quantities are available to develop inferences about our model.

- \( p(Y_1) \): Prior distribution over the environmental conditions,
- \( p(\theta_t \mid Y_1) \): Prior dependence of the driver state given the environmental conditions,
- \( p(X_1 \mid \theta_1) \): Prior dependence of the monitoring variable given the initial driver state,
- \( p(\theta_{t+1} \mid \theta_t, Y_{t+1}) \): Predictive evolution of the driver state given their previous state and the current environmental conditions,
- \( p(X_{t+1} \mid \theta_{t+1}, X_t) \): Predictive evolution of the monitoring variables, given the monitoring variable at time \( t \) and the state at time \( t + 1 \).

Under this configuration, whenever period \( t \) starts, we may determine distributions \( p(\theta_t \mid D_t) \) and \( p(Y_{t+1} \mid D_t) \), wherein \( D_t \) includes the compendium of environment- and interior-focused sensor observations.

Whereas \( p(Y_{t+1} \mid D_t) \) can be determined using the methods described in Section 2.2, \( p(\theta_{t+1} \mid D_t) \) can be identified by reasoning about the dependencies in Figure 2. Indeed, forecasts for the driver-state variables \( \theta_{t+1} \) can be constructed using standard predictive computations (West and Harrison 2006).

Figure 2. Forecasting Environment and State
\[ p(\theta_{t+1} \mid Y_{t+1}, D_t) = \int p(\theta_{t+1} \mid \theta_t, Y_{t+1})p(\theta_t \mid D_t) d\theta_t, \]
\[ p(\theta_t \mid D_t) = \int p(\theta_{t+1} \mid Y_{t+1}, D_t)p(Y_{t+1} \mid \theta_t) dY_{t+1}. \]

Similarly, it can be observed that the forecast on the monitoring variables \( X_{t+1} \) are given by

\[ p(X_{t+1} \mid D_t) = \int p(X_{t+1} \mid \theta_{t+1}, X_t)p(\theta_{t+1} \mid D_t) d\theta_{t+1}. \]

At time \( t + 1 \), additional sensor input is available, implying that \( X_{t+1} = x_{t+1} \) and \( Y_{t+1} = y_{t+1} \) are observed. The corresponding information update is thus \( D_{t+1} = D_t \cup \{x_{t+1}, y_{t+1}\} \). Utilizing this information, we can proceed in a Bayesian manner to update our estimate of the driver’s true state, according to

\[ p(\theta_{t+1} \mid D_{t+1}) \propto p(\theta_{t+1} \mid D_t, y_{t+1})p(x_{t+1} \mid \theta_{t+1}, x_t). \]

This one-step recursion may continue over time as new observations are received.

However, as previously mentioned, it may be insufficient to only forecast the driver’s state one time period into the future. Instead, the ADS may require a length-\( k \) horizon forecast. Such forecasts can be constructed based on standard predictive computations via

\[ p(\theta_{t+i} \mid Y_{t+i}, D_t) = \int p(\theta_{t+i} \mid \theta_{t+i-1})p(\theta_{t+i-1} \mid D_t) d\theta_{t+i-1}, \quad i = 1, 2, \ldots, k, \]

\[ p(\theta_{t+i} \mid D_t) = \int p(\theta_{t+i} \mid Y_{t+i}, D_t)p(Y_{t+i} \mid \theta_{t+i-1}) dY_{t+i}. \]

Analogously, the \( i \)-length forecast of the monitoring variables can be recursively evaluated through

\[ p(X_{t+i} \mid D_t) = \int p(X_{t+i} \mid \theta_{t+i}, X_{t+i-1})p(X_{t+i-1} \mid D_t) dX_{t+i-1}, \quad i = 1, \ldots, k. \]

These DSM predictions are of utmost importance in both level-3 and -4 ADSs to determine whether a driver is able to take control of the vehicle when ODD conditions are exceeded (i.e., RTIs are only not required in level-5 ADSs).

2.3.1. DSM Warnings. The DSM module can emit warnings based on the inferred driver state and observed driver behaviors. If \( \Theta \) is a finite state space with singleton elements \( \theta^i \), we write \( \Theta = \{\theta^1, \ldots, \theta^k, \theta^{l+1}, \ldots, \theta^m\} \) with values \( \{\theta^1, \ldots, \theta^k\} \) corresponding to favorable states (e.g., aware) and \( \{\theta^{l+1}, \ldots, \theta^m\} \) referring to unfavorable ones (e.g., sleepy or distracted). When the ADS is in manual mode, or when the ODD limits are being approached in automated mode, the vehicle should issue a warning if the probability of encountering an unfavorable driver state is too great. We can determine whether the probability of the driver being in an unfavorable state is too great by considering a threshold \( \omega_d \) and checking whether \( q_d = \sum_{i=1}^m p(\theta^i_{t+1} \mid D_t) \geq \omega_d \). A natural value for this threshold is 1/2, equivalent to checking whether \( \sum_{i=1}^m p(\theta^i_{t+1} \mid D_t) \geq \sum_{i=1}^m p(\theta^i_{t+1} \mid D_t) \). In this manner, it can be assessed whether it is more likely for the driver to be in a favorable state than an unfavorable state; conversely, this threshold could be more demanding. Similar to other modules, we could alternatively implement two alert levels \( (\omega_{d1}, \omega_{d2}) \) for issuing critical and standard alarms, respectively. If the driver does not respond to these alarms and the conditions become too dangerous, the ADS should enter emergency mode.

Based on the driver-behavior-monitoring variables, a warning can also be issued when an unexpected change is detected (e.g., the driver falls asleep). Doing so may be based on whether the probabilities associated with the monitoring variables exceed established thresholds. That is, define \( q_5 = p(\|X_{t+i}\| > \|x_{t+i}\| \mid D_t) \), and let \( \omega_{d1} \) and \( \omega_{d2} \) be the associated warning levels. Utilizing these values, the critical and standard warnings can be issued akin to the previous modules.

2.4. Trajectory Planning

An ADS will have a trajectory-planning system available; we refer an interested reader to Gonzalez et al. (2016), Claussmann et al. (2019), and Katrakazas et al. (2015) for reviews. At a given time \( t \), and for a length \( k \) horizon forecast, it provides the ADS with a trajectory plan \( \bar{\pi} = \{z_{t+i}, z_{t+i+1}, \ldots, z_{t+k}\} \) within the ODD boundaries. This trajectory will be computed mainly to ensure that the ADS stays within road boundaries, abides by local traffic laws, and avoids collisions. For RTI purposes, this module is assumed exogenous.

2.5. Driving-Mode Assessment

At a given time \( t \), assume we have available a trajectory plan \( \bar{\pi} \) and desire to assess the value of driving mode \( d_t \) over a length \( k \) horizon. If a forecast on the environmental states \( \bar{Y} = \{Y_{t+1}, \ldots, Y_{t+k}\} \) and a forecast of the driver state \( \bar{\theta} = \{\theta_{t+1}, \ldots, \theta_{t+k}\} \) have been computed according to Sections 2.2 and 2.3, respectively, we may define

\[ p(\bar{Y}, \bar{\theta} \mid D_t) = p(Y_{t+1} \mid \bar{Y}_t)p(\theta_{t+1} \mid Y_{t+1}, D_t) \prod_{i=2}^k p(\theta_{t+i} \mid \theta_{t+i-1}, Y_{t+i}, D_t)p(Y_{t+i} \mid Y_{t+i-1}). \]

A utility function \( u(d_t, \bar{z}, \bar{Y}, \bar{\theta}, g) \) is used to assess the efficacy of the driving mode over the next \( k \) steps, incorporating the objectives and information for times \( t + 1 \) to \( t + k \). This function is assumed to consider the
driving mode, trajectory plan, environment forecast, driver-state forecast, and last observed ODD conditions. Typical consequences assessed would include comfort, internal and external safety, fuel consumption, travel time, and reaction time. The importance of various objectives would usually change depending on the driving environment. For example, higher speeds typically provide more utility on a highway than in a residential area.

Given the uncertainty, we compute the expected utility of the driving mode through

$$
\psi(d_i) = \int \int u(d_i, z, \bar{Y}, \bar{g}_i) p(\bar{Y}, \bar{g}_i \mid D_i) d\bar{Y} d\bar{g}_i
$$

such that a driving mode yielding the greatest expected utility is preferable. Note that, given the gradual evolution of ODD conditions and the compressed nature of the planning horizon, it is not necessary to forecast the ODD variables for the assessment decision. Because computing the expected utility is the most computationally expensive calculation to be performed, we may also introduce a model to forecast driving-mode assessments. Such forecasts, call them \(\psi(d_i)\), may consequently accelerate the process of selecting the preferred driving mode.

### 2.6. Driver-Intervention Performance Assessment

To wit, after an RtI is issued, ADS operations can be improved if the driver’s performance can be evaluated. Conceptually, driver-intervention performance-assessment systems have been proposed as a means to record and retain historical data about driver interventions (Bianchi 2018, Kosmides et al. 2020, Trustonomy 2020). This information can be used to inform future RtI decisions, thereby providing additional information beyond myopic sensor data. Herein, we formally define an expected-utility-based DIPA methodology.

For a given interval \([t + 1, t + k]\), assume the driver’s actual performance is monitored and its associated utility is computed. Call this value the driving score and represent it as \(u(d_1)\), wherein \(d_1\) designates the manual driving mode. After the ADS transfers control to the driver, \(u(d_1)\) is compared with their expected performance, \(\psi(d_1)\), to assess the intervention. That is, our approach to DIPA is characterized by identifying whether a driver underperformed or overperformed with respect to their expected performance. This is accomplished by examining the difference between \(u(d_1)\) and \(\psi(d_1)\), as follows:

- \(\psi(d_1) - u(d_1) > 0\), the driver under performed,
- \(\psi(d_1) - u(d_1) \leq 0\), the driver over performed.

We consider both discrete and continuous methods of DIPA modeling under this framework.

In the discrete approach, we draw inferences about the probability \(p\) of the driver underperforming. A simple version uses the beta-binomial model (French and Insua 2000): If there were \(h\) underperformances out of \(n\) RtIs, such that \(D_i = \{h, n\}\), the posterior is \(p \mid D_i \sim \text{Beta}(\lambda + h, \kappa + n - h)\), wherein \(\lambda\) and \(\kappa\) are prior parameters.

We may also include \(p\) in the utility function, or an estimate \(\bar{p}\) (e.g., its mean), to make utility more risk-averse as \(p\) increases. For example, if \(p \neq 0\), we could use \(u^{p/p}\), wherein \(u\) is the original utility function, assuming, without loss of generality, that \(u \in [0, 1]\).

In the continuous case, the focus is on the utility differences, \(\xi = \psi(d_1) - u(d_1)\). We can draw inferences on the mean utility difference \(\mu\) based on a normal-normal model (French and Insua 2000). Indeed, if there were \(n\) RtI cases with observed performances \(\xi_1, \xi_2, \ldots, \xi_n\), we can assume a prior \(\mu \sim N(\mu_0, \sigma_0^2)\) and a likelihood function \(\xi \mid \mu, \sigma^2\sim N(\mu, \sigma^2)\), resulting in a posterior \(\mu \mid D_i \sim N\left(\mu_1 = \frac{\xi_0 + n \bar{\sigma}^2}{\sigma_0^2 + n \bar{\sigma}^2}, \frac{\bar{\sigma}^2}{\sigma_0^2 + n \bar{\sigma}^2}\right)\), wherein \(\bar{\xi}\) is the performance sample mean. Subsequently, some factor could be included in the utility function to increase risk aversion when the driver is expected to underperform. For example, assuming, again, that utility is scaled within \([0, 1]\), we could retain the original utility function \(u\) if the posterior mean \((\mu_\lambda)\) is less than zero (i.e., an overperformance is expected) and use \(\frac{u}{u_\lambda}\) otherwise. Such a definition ensures that the utility function becomes increasingly risk-averse as the expected underperformance worsens.

#### 2.6.1. DIPA Warnings

Consider now warnings related to DIPA. In the discrete approach, if the probability of \(\rho\) being large—that is, \(q_\rho = \text{Pr}(\rho \geq \beta \mid D_i)\) for \(0 \leq \beta \leq 1\)—is too great, the ADS may advise the driver that their proficiency is insufficient for the present scenario and that additional training is advisable. This information could also be used in driving-mode management decisions.

In the continuous approach, if the posterior indicates a higher underperformance probability, computed through \(q_\beta = \text{Pr}(\mu > 0 \mid D_i) > \beta\), the ADS may advise the person that their driving skills are insufficient for the current scenario and that additional training is advisable. In either scenario, relevant warning thresholds, \(\omega_{0\epsilon}\) and \(\omega_{0\beta}\), can be established and utilized as before.

### 3. Managing Driving-Mode Transitions

We detail herein a decision-analytic framework that tackles the RtI issue and the management of driving modes presented in Figure 1. The developed approach emphasizes a management by exception principle (West and Harrison 2006)—that is, a group of models is used for inference, prediction, and decision support.
under standard driving circumstances until an exception arises that triggers an RTI.

In what follows, we explicitly consider warnings and transitions between three driving modes (i.e., automated, manual, and emergency) designated as $d_0$, $d_1$, and $d_2$, respectively. Algorithm 1 provides an overarching approach for doing so via the utilization of Algorithms 2–6 as subroutines. We note that the algorithms’ outputs depend upon a set of thresholds (i.e., $\omega$ and $\psi$, $\forall j \in \{1, \ldots, 6\}$). Although there exist multiple means for optimizing these thresholds—for example, direct search algorithms (Audet and Dennis 2006) or response surface methodology (Myers, Montgomery, and Anderson-Cook 2016)—herein, such thresholds are determined heuristically via empirical means. Likewise, we assume for the remainder of this section that driving states are univariate and qualitative, in accordance with Stutts et al. (2001) and Ranney, Garrott, and Goodman (2001). Generalizations in both these regards are promising areas of future inquiry.

3.1. Transitions from Automated

We begin with transitions from the automated mode summarized in Algorithm 3. Inputs are available from the trajectory, DSM, environment, and ODD modules. The system periodically issues predictive risk assessments based on compliance with the ODD. If $q_1 = P_r(G_{1 \in k} \notin \mathcal{O} \mid D_i)$ is sufficiently great, the ADS determines it very likely that ODD conditions will be exceeded. Therefore, the driver is alerted, and the ADS assesses the utility of the automated and manual driving modes. If $\psi(d_0) > \psi(d_1)$, the automated mode is preferable; however, if the ADS is critically close to approaching its design limits, the system should enter the emergency mode and issue the appropriate alert. Otherwise, an RTI is issued to the driver through the HMI, and a DIPA is subsequently performed. If the driver performs too poorly, as assessed via the continuous DIPA model, the ADS determines that the driver is not in a favorable state and triggers the emergency mode. If performance is adequate, the driver maintains control until further notice.

3.2. Transitions from Manual

Transitions from manual mode are summarized in Algorithm 4. In manual mode, the driver controls the vehicle, implying that the ADS’s planned trajectory operates in the background. However, to guarantee proper driving-mode management, the DSM and environment modules continue to operate in the foreground. At a certain point, the driver requests the automated mode. To ensure safety, noncompliance with operational domain conditions is assessed by checking whether $q_1 = P_r(G_{1 \in k} \notin \mathcal{O} \mid D_i)$ is too great. If so, the ADS issues a warning; if ignored by the driver, it may enter into the emergency mode. That is, the system checks whether $\psi(d_0) > \psi(d_1)$; if this condition is false, the system assesses that the manual mode is preferable and alerts the driver. If the warning is neglected, the ADS enters into emergency mode. Conversely, if the condition is true, the system enters into automated mode.

While a driver could theoretically request a transition to emergency mode, we do not consider such interactions herein. Should a designer wish to do so, only minor modifications would be required. Finally, although we assume that the ADS is able to override dangerous human decisions, doing so sets aside related ethical quandaries regarding a human’s right to err.

3.3. Transitions from Emergency

The emergency mode can be considered as an alternative automated mode having just one criteria (i.e., safety); related transitions are summarized in Algorithm 5. The ADS is assumed to be equipped with a computationally efficient trajectory-planning system that enables it to safely stop. Once stopped, and, assuming conditions allow for the resumption of ADS operations, the driver may request to activate either the automated or manual modes. The ADS assesses whether the operational conditions are met and whether the driver’s state is adequate. If driving is feasible, it evaluates which of the two modes is preferable and proceeds accordingly. If driving is not feasible, the ADS will emit the corresponding warning and remain idle.

Finally, although not discussed in detail herein, we note that the incorporation of ethical decision making within the emergency mode is a nuanced issue. Lin (2016) presents several vignettes illustrating moral dilemmas associated with ADS, which point designers to important problems. For further information in this regard, we refer the interested reader to the works of Awad et al. (2018); Basl and Behrends (2020); Caballero, Rios Insua, and Banks (2021); and Caballero, Naveiro, and Rios Insua (2021).

3.4. Summary

The (mostly qualitative) scheme in Algorithm 1 summarizes the complete ADS management procedure. The three driving modes are denoted AUTO, MANUAL, and EMERG. Commands on the same line are processed in parallel. The variables $\psi_k$ and $\psi_1$ designate the assessment (k steps ahead) of the AUTO and MANUAL modes, respectively. Algorithm 1 utilizes Algorithms 2–6 as subroutines.
Algorithm 1 (ADS Controller)
Input: Priors for ODD, environment, driver state. Utility function
Output: Trajectory from ORIGIN to DESTINATION; Implementation of AUTO/EMERG commands.
while DESTINATION not reached do
Read internal and external sensors. Compute ethical trajectory. (k steps ahead)
Assess driving modes ($\psi_0$, AUTO; $\psi_1$, MANUAL). Issue WARNINGS.
Manage from DRIVING MODE. If any DIPA pending, resolve it
end while

Algorithm 2 (Issuing WARNINGS)
Input: Thresholds for warnings ($\omega_{1c}$, $\omega_{1s}$)
Output: Warnings to be issued.
Compute $q_j$ $\forall$ j $\in$ {1, ..., 6}
if $q_1 > \omega_{1c}$ then
   Warning: About to exceed ODD limits
else if $q_1 > \omega_{1s}$ then
   Warning: Reaching ODD limits
end if
if $q_2 < \omega_{2s}$ then
   Warning: Exceptionally abnormal ODD observations
else if $q_2 < \omega_{2s}$ then
   Warning: Abnormal ODD observations
end if
if $q_3 < \omega_{3s}$ then
   Warning: Exceptionally abnormal environment behaviour
else if $q_3 < \omega_{3s}$ then
   Warning: Abnormal environment behaviour
end if
if $q_4 < \omega_{4c}$ then
   Warning: Potentially dangerous driver state
else if $q_4 < \omega_{4s}$ then
   Warning: Unfavorable driver state
end if
if $q_5 < \omega_{5s}$ then
   Warning: Potentially dangerous driver behaviour observations
else if $q_5 < \omega_{5s}$ then
   Warning: Odd driver behaviour observations
end if

Algorithm 3 (Managing from AUTO)
Input: $q_1$, $\psi_0$, $\psi_1$.
Output: Mode to be followed
if $q_1 > \omega_{1c}$ then
   if $\psi_0 > \psi_1$ then
      Undertake EMERG
   else
      Issue RtI, Start DIPA
   end if
else
   Remain AUTO
end if

Algorithm 4 (Managing from MANUAL)
Input: $q_1$, $\psi_0$, $\psi_1$.
Output: Mode to be followed
if Driver requests AUTO then
   if $q_1 > \omega_{1c}$ then
      if $\psi_0 > \psi_1$ then
         Start EMERG
      else
         Warning MANUAL mode preferable
      end if
   else
      Start AUTO
   end if
else
   Remain MANUAL
end if

Algorithm 5 (Managing from EMERG)
Output: Mode to be followed
if $\psi_1 > u(d_1)$ then
   $\lambda = \lambda + 1$.
else
   $\kappa = \kappa + 1$.
end if
Compute $q_6 = Pr(\beta(\lambda, \kappa) \geq \beta)$
if $q_6 > \omega_{6c}$ then
   Warning Exceptionally poor driving performance
else if $q_6 > \omega_{6s}$ then
   Warning Poor driving performance
end if

Algorithm 6 (Resolving a DIPA)
Input: $\psi_1$, $\lambda$, $\kappa$.
Output: Driver-intervention performance assessed.
if $\psi_1 > u(d_1)$ then
   $\lambda = \lambda + 1$.
else
   $\kappa = \kappa + 1$.
end if

4. Simulation and Empirical Analysis

This section sets forth a simulated environment, within which the proposed driving-mode management framework is tested. Utilizing this simulation, empirical analysis is provided to characterize the performance of our framework on a myriad of measures. We begin in Sections 4.1 and 4.2 by detailing the simulator itself; that is, we discuss the driving environment’s configuration and the ADS controller’s implementation. Subsequently, empirical results are provided in Section 4.3. Beyond illustrating the efficacy of our driving-mode management framework, these results underscore a fundamental dilemma in level-3 and -4 ADSs—management framework, these results underscore a fundamental dilemma in level-3 and -4 ADSs—that is, the transfer (or not) of ADS control to a distracted driver when ODD limits are very likely to be exceeded. Although it is initially unclear whether the ADS should transfer control to the driver, thereby allowing the human to assume the risks of their distraction, or if a conservative approach should be adopted via the continuation of the AUTO mode, we illustrate herein how this dilemma can be addressed via expected utility maximization.

4.1. Driving-Environment Configuration

The driving-environment configuration describes the underlying decision context within which the ADS operates. It considers (1) road attributes, (2) vehicle attributes, (3) driver features, (4) ODD assumptions, (5) environmental dynamics, (6) driver-behavior-and-state evolution, (7) trajectory-planning assumptions, and (8) preference-model assumptions. The simulation collectively utilizes each component to evaluate the developed driving-mode management framework.

4.1.1. Road Attributes. The ADS moves along a single-lane, straight road decomposed into cells of equal length.

4.1.2. Vehicle Attributes. The ADS is level 3 with two modes, AUTO and MANUAL. In AUTO mode, it is able to select among four speeds representing the number of cells advanced per unit time interval: 0 (i.e., stopped), 1, 2, or 3. Speed choices are transmitted immediately at the beginning of the time interval. The EMERG mode is implemented within the AUTO mode due to the simplified environmental dynamics. MANUAL mode operations comport with the driver features.

4.1.3. Driver Features. The driver (she) may be in two states, distracted ($\theta^1$) or aware ($\theta^2$)—that is, $\Theta = \{\theta^1, \theta^2\}$, wherein state $\theta^1$ is unfavorable and state $\theta^2$ is favorable. In MANUAL mode, the driver is also able to select among five possible speeds (i.e., number of cells advanced per unit time): 0, 1, 2, 3, or 4. If the driver is aware, speed choices are transmitted immediately at the beginning of the time interval; otherwise, there is a one-time-interval delay. Likewise, the driver requires one time interval to respond to an RtI when aware, but three time intervals when distracted.

4.1.4. ODD Assumptions (Section 2.1). The ODD requires a minimum three-cell separation between the ADS and any obstacle appearing in the road. The ODD mandates that the ADS does not utilize its maximum speed when encountering a puddle. Further detail on the types of obstacle that may appear is provided subsequently.

4.1.5. Environmental Dynamics (Section 2.2). Two types of objects may appear in the driving scene; each is associated with a particular color for illustrative purposes.

- A rock (dark gray object) may appear and cover one cell of the road. The driver must stop to avoid crashing with it (i.e., select speed 0). Once stopped, the rock disappears, and the ADS may resume its trip.
- A puddle (light gray object) may appear and cover one cell of the road. While in AUTO mode, if the ADS crosses a cell occupied by a puddle at maximum speed (i.e., speed 3), it will skid with probability 0.95. Alternatively, if the driver is distracted and crosses a puddle-occupied cell at speeds 2, 3, or 4, it will skid with probability 0.5, 0.8, or 0.85, respectively.

Table 2 reflects the environmental dynamics of the road state, $p(Y_{t+1} \mid y_t)$. For example, after encountering a rock, the next cell of the road will necessarily be clean—that is, $p(Y_{t+1} = \text{Clean} \mid Y_t = \text{Rock}) = 1$.

Collectively, this implies that the set of possible environmental conditions a cell may take is $\mathcal{Y} = \{\text{Rock, Puddle, Clean}\}$. As reflected in Table 3, we assume that objects are perceived without ambiguity some time in advance. For example, in AUTO mode, the ADS sensors detect the presence of a rock three cells in advance. Conversely, perceptions in MANUAL mode depend on whether the driver is distracted or aware.

4.1.6. Driver-Behavior-and-State Evolution (Section 2.3). The ADS is equipped with a DSM system that monitors the number of driver blinks per time interval ($X_t$). This discrete monitoring variable may assume

| $Y_{t+1}$ | Rock | Puddle | Clean |
|-----------|------|--------|-------|
| Rock      | 0    | 0      | 1     |
| Puddle    | 0.4  | 0.6    |       |
| Clean     | 0.05 | 0.05   | 0.90  |
one of three possible values: 1, 2, or 3 blinks. The probability, \( p(X_t \mid \theta_t) \), of each behavior given the driver’s true state is shown in Table 4. Unlike Section 2.3, \( X_t \) does not depend on \( X_{t-1} \) in this simulation.

The driver-state evolution, \( p(\theta_{t+1} \mid y_{t+1}, \theta_t) \), is presented in Table 5. The driver state does exhibit temporal associations and depends on the previous driver state as well as the current environmental conditions. For example, if the driver is aware (i.e., in state \( \theta^1 \)) and the next cell is clean, the driver will be distracted (i.e., in state \( \theta^2 \)) with probability 0.15 in the next time period.

### 4.1.7. Trajectory-Planning Assumptions (Section 2.4)

Trajectory planning is performed \( k = 5 \) time intervals in advance and accordingly conjoined with speed choices over the same horizon.

### 4.1.8. Preference-Model Assumptions (Section 2.5)

The utility function is defined such that its value increases with ADS speed. However, to maximize safety, the ADS should avoid collisions, and, to maximize comfort, the ADS should maximize time spent in AUTO mode. Table 6 reflects the utilities attained at various speeds per cell. For example, a cell crossed at speed 1 provides utility 0.1.

Moreover, each collision with a rock is associated with a utility of \(-10\), each skidding incident incurs a utility of \(-10\), and each cell driven in automated mode provides a utility of 0.1. The assessment of a trip’s total utility is computed by summing over the individual cell utilities.

### 4.1.9. Simulating the Driving Environment

To simulate the environment, a road length is first selected. The first cell is initialized as clean. Thereafter, we deduce the presence of puddles and rocks based on Table 2. Once the driving environment is established, we simulate driver awareness from Table 5; the driver is assumed to be initially aware at the start of the trip. Subsequently, we simulate the number of driver blinks, in accordance with Table 4. For this manuscript, a realization of the described driving environment was coded in Python. Further information regarding the coding of our empirical trials is provided in subsequent sections.

### 4.2. ADS Controller Implementation

Herein, we describe how ODD supervision, environment monitoring, driver-state monitoring, trajectory planning, and manual driving planning are conducted by the ADS within the simulation environment. Subsequently, we describe how each of these elements are leveraged within the overarching ADS controller.

#### 4.2.1. ODD and Environment Monitoring (Sections 2.1 and 2.2)

The ADS is limited by its sensing capabilities when performing ODD and environment monitoring. If a rock is present within one of the next three cells, the ADS will sense it with certainty and issue a warning regarding its presence. Conversely, puddles are sensed only one cell in advance. Therefore, the ADS must build forecasting models regarding the presence of puddles in cells \( t + 2 \) and \( t + 3 \), given the content of cells \( t + 1 \). Because the ADS sensors provide no input about the contents of cells \( t + 4 \) and \( t + 5 \), forecasting models are also required to reason about these quantities.

More formally, because the elements of Table 2 are not known by the ADS, it must learn the environmental dynamics to produce forecasts regarding the contents of the next five cells—that is, the ADS must learn the environmental dynamics to perform ODD and environment monitoring. This learning is accomplished.
by using Dirichlet-multinomial models (French and Insua 2000).

Define \( p_i = p(y_{t+1} \mid y_t = i) \) to be the ADS’s beliefs about the environmental dynamics, represented as a multinomial distribution, wherein \( y_{t+1} \) denotes the contents of cell \( t + 1 \), and \( i \) corresponds to the values rock, puddle, or clean. Uncertainty about \( p_i \) is modeled with a Dirichlet distribution, such that \( \alpha_i \sim \text{Dir}(\alpha_i) \). A priori, we set \( \alpha_i = 1 \), but update this estimate as follows. When the ADS enters into cell \( t \), it observes the cell’s environmental conditions (i.e., \( y_t = i \)) and immediately identifies the contents of cell \( t + 1 \) (i.e., \( y_{t+1} \)). With this information, \( p_i \) is updated to \( p_i \mid y_{t+1} \sim \text{Dir}(\hat{\alpha}_i) \), such that \( \hat{\alpha}_{i,y_{t+1}} = \alpha_{i,y_{t+1}} + 1 \) is the value for the \( y_{t+1} \) (e.g., rock) component of \( \hat{\alpha}_i \). After this update is performed, predictions for cells \( t + 2, t + 3, t + 4, \) and \( t + 5 \) are made in a sequential fashion. This is accomplished by (1) using the current estimate of \( p_i \), (2) identifying whether cells \( t + 2 \) or \( t + 3 \) contain a rock, and (3) making subsequent observations. To simplify calculations, we utilize a point estimate of \( p_i \) that approximates \( p(y_{t+1} = j \mid y_t = i) \) by \( \mathbb{E}[p_{ij}] = \frac{\alpha_{ij}}{\sum_k \alpha_{ik}} \).

As a tangible example of this construct, consider making a prediction for the environmental conditions of cell \( t + 4 \). Based on the ADS capabilities, we distinguish two cases:

- Suppose cell \( t + 3 \) contains a rock. Because this can be perceived with certainty via the ADS sensors, the prediction for cell \( t + 4 \) is
  \[
  p(y_{t+4} = j \mid y_{t+3} = \text{Rock}) = \frac{\hat{\alpha}_{ij}}{\sum_k \hat{\alpha}_{ik}}.
  \]

- Assume cell \( t + 3 \) is clean or contains a puddle. Because the ADS sensors cannot perceive the cell’s contents with certainty, we have
  \[
  p(y_{t+4} = j \mid y_{t+3}) = \sum_{y_{t+3}} p(y_{t+4} = j \mid y_{t+3}) p(y_{t+3} \mid y_{t+2}),
  \]
  where \( p(y_{t+3} \mid y_{t+1}) \) can be computed again considering two cases, depending on the content of cell \( t + 2 \). This computation entails calculating \( p(y_{t+2} \mid y_{t+1}) \), thereby reaching the end of the recursion. The quantity \( p(y_{t+2} \mid y_{t+1}) \) again has two cases. If a rock is observed in cell \( t + 2 \), then \( p(y_{t+2} = \text{Rock} \mid y_{t+1}) = 1 \), and 0 otherwise. If not,
  \[
  p(y_{t+2} = j \mid y_{t+1}) = p(y_{t+2} = j \mid y_{t+1} = i) = \frac{\hat{\alpha}_{ij}}{\sum_k \hat{\alpha}_{ik}},
  \]
  where \( i \) is the observed content of cell \( t + 1 \), and \( j \) is either a puddle or clean.

4.2.2. Driver-State Monitoring (Section 2.3). No learning is possible for DSM because our simulation does not include a human in the loop. Therefore, we adopt an optimistic perspective, assuming that the models in Tables 3 and 4 are known by the system. When the ADS moves one cell forward, it produces an estimate \( p(\theta_t \mid D_t) \) of the driver’s state given all the information available up to that point (i.e., \( x_{t+1} \) and \( y_{t+2} \)). This can be computed recursively by using Equations (1), (2), and (3). Additionally, a forecast of the driver’s state in cells \( t + 1 \) through \( t + 5 \) is performed by using Recursion (4).

4.2.3. Trajectory Planning (Section 2.4). As previously described, the ADS produces forecasts regarding the obstacles present in the next five cells using the aforementioned Dirichlet-multinomial models. In particular, the ADS knows with certainty whether the next cell has a puddle and if any of the next three cells has a rock. Therefore, to determine its trajectory, the ADS identifies the most probable environmental conditions and selects a speed to maximize its cell-wise expected utility. The trajectory plan is updated at each cell as the ADS moves forward.

4.2.4. Manual Driving Planning (Section 2.5). This is performed similarly, except that it is based on the driver’s perception and does not include a forecasting component. The driver examines the scene, and, if aware, rocks and puddles are perceived up to five cells ahead. Conversely, if distracted, rocks are perceived up to two cells ahead, and puddles are not perceived. The driver selects the maximum speed by taking into account skid probabilities and costs. If a rock is perceived, the driver will stop. If she encounters a puddle while driving at speeds 0, 1, 2, 3, or 4, a skid will occur, with probabilities 0.0, 0.0, 0.5, 0.8, and 0.85, respectively. These provide expected utilities of 0, 0.1, −4.8, −7.7, and −8.1. Consequently, to maximize their expected utility, the driver will choose speed 1 when a puddle is present. In juxtaposition, if the road is clean, her expected-utility-maximizing choice is to select speed 4. Recall that when the driver is distracted, decisions transmission is delayed by one time interval.

4.2.5. Driver-Intervention Performance Assessment (Section 2.6). DIPA is performed by using the discrete approach in Section 2.6. For each cell the ADS operates in MANUAL mode, the actual utility earned by the driver is compared with the expected utility of MANUAL mode. If the latter is greater than the former, the ADS determines that the driver underperformed in the corresponding cell; otherwise, the driver overperformed. Given that the driver’s state is unknown to the ADS, MANUAL mode’s expected utility is computed by marginalizing over the driver state. Finally, inferences about \( p \), the probability of the driver underperforming, are performed by using a beta-binomial model. This model is updated at each cell wherein the ADS starts in MANUAL mode. At
the end of the trip, the quantity \( \Pr(\rho \geq \beta | D_t) \) informs the ADS about the driver’s proficiency.

4.2.6. ADS Control (Section 3). Utilizing the previously defined elements and based on the framework presented in Algorithm 1, the ADS controller is implemented as follows. This overarching framework clearly illustrates the flexibility of our approach—for example, via tailored warning thresholds for the considered environment.

1. The ADS enters the first cell. It has knowledge about the elements of Tables 4 and 5. The utility function of Table 6 is also known. The Dirichlet distributions for ODD and environment forecasting are initialized by using prior values, and an update is performed after the second cell is observed. The ADS computes \( p(\theta_1 | D_1) \).
2. For cells \( t = 2 \) until \( t = N \), the following steps are utilized:
   a. The ADS enters cell \( t \) in the prescribed mode.
   b. The Dirichlet distributions are updated as cell \( t + 1 \) is observed.
   c. \( p(\theta_1 | D_t) \) is updated.
   d. Forecasts of obstacles and driver states for the next five cells are produced.
   e. Trajectory planning is updated, deciding the speed for the current cell and the five subsequent cells. Decisions are transmitted immediately. The utility gained in the current cell is computed.
   f. If the probability of encountering a puddle in cells \( t + 2 \) or \( t + 3 \) is greater than \( \omega_p^{t+2} = 0.25 \), an alarm indicating a dangerous environmental state is raised. Similarly, if the probability of finding a puddle or a rock in cells \( t + 4 \) or \( t + 5 \) is greater than \( \omega_p^{t+4} = 0.25 \) and \( \omega_p^{t+5} = 0.15 \), respectively, another alarm is raised. Note that, in this case, differing alarm thresholds are utilized for distinct aspects of the environment, and alarm levels are also affected by the distance to the obstacle.
   g. If \( p(\theta_{1+k} = \theta_1^1 | D_t) \) for \( k = 1, 2, \ldots, 5 \) is above \( \omega_k = 0.9 \), an unfavorable driver-state alarm is issued.
   h. When a dangerous road-state alarm is raised, driving modes are assessed as in Algorithm 3. The expected utility of each mode is computed within the next five cells. If MANUAL mode is preferred over AUTO, an RtI is issued.
   i. When issuing the RtI, if the probability of the driver being distracted is above \( \omega_d = 0.9 \), an emergency alarm is raised, and AUTO mode continues. If this probability is above \( \omega_d = 0.5 \), but less than \( \omega_d = 0.9 \), a standard warning is raised; however, the mode is changed to MANUAL. If this probability is below 0.5, the driving mode is changed to MANUAL without an alarm.
   j. The driving mode remains in MANUAL until the probability of the driver being distracted surpasses 0.75. If this does not happen within 10 cells, AUTO mode is then restored.
   k. While in MANUAL mode, driver performance is measured and used to update the DIPA beta-binomial model.
3. Upon arriving at cell \( t = N \), the trip concludes, and the globally attained utility is computed.

The described ADS controller was implemented in Python via five main methods. The predict method provides relevant forecasts at each cell for the subsequent five cells. The update method updates current estimates of the driver and environment states, given the newly observed information. The decide method outputs the decision for the current cell and the trajectory plan for the next five cells. The issue warnings method evaluates the driver- and environment-state forecasts, emitting warnings based on their values. Lastly, the evaluate driving modes method evaluates the AUTO and MANUAL modes and decides whether to issue an RtI. To complete the simulator, an additional move method coordinates amongst the aforementioned: When invoked, the ADS advances one cell forward, update is used to process new information, and predict produces relevant forecasts that are used by decide and issue warnings. Finally, if road-state warnings are produced, evaluate driving modes decides whether to issue a RtI and change the driving mode.

4.3. Results and Discussion

Herein, testing and discussion is provided in different blocks, each having a distinct emphasis. We begin by empirically validating the simulation’s output. This is accomplished by first reviewing a single, small-scale example to verify the validity of both the simulation environment and the developed algorithms. Subsequently, we perform 1,000 runs on a 1,000-cell road to accomplish a more rigorous and in-depth exploration. Based on these results, inferences are also drawn regarding the performance of the ODD and environment-monitoring strategies. Finally, two additional experiments are conducted to isolate and draw conclusions about distinct behavioral facets—that is, one experiment is performed to explore the accuracy of the DSM methodology, and the other is performed to evaluate how the ADS control framework behaves under varied environmental conditions. Code to reproduce the experiments is available at https://github.com/roinaveiro/ads_trusto.

4.3.1. Empirical Validation. Consider an ADS’s trip along the 90-cell road depicted in Figure 3. To optimize space, we plot the one-dimensional road in two dimensions. The ADS begins its journey in cell (1,1) and moves within a given row toward the right; once
it reaches a row’s rightmost cell, it continues in the leftmost cell of the row above. Within each cell, we display the ADS’s driving mode and speed. When operating in MANUAL mode, we also depict whether the driver was distracted or aware. The presence of an obstacle in a cell is denoted by its respective color: dark gray for rocks and light gray for puddles.

As can be observed in Figure 3, the AUTO mode is clearly preferred over much of the trip. However, in cell (3,2), the presence of two consecutive puddles, coupled with the fact that the driver is aware, triggers the MANUAL mode. After four additional cells, the driver becomes distracted in cell (3,9). A puddle appears in cell (4,4), and the car skids as it crosses the cell at maximum speed. Note that the decision made in cell (4,4) was determined in the cell (4,3) when the driver was distracted. Subsequently, due to the presence of another puddle, the MANUAL mode is invoked once more in cell (4,8). However, as in the remaining cells, the DSM system detects that the driver is distracted, and AUTO mode is quickly restored.

4.3.2. Empirical Exploration. We perform 1,000 experimental runs across a 1,000-cell road. The box plots in Figure 4, respectively, depict (1) the utilities attained, (2) the fraction of time in automated mode, (3) the number of RtIs issued, and (4) the number of emergencies encountered. It can be observed (Figure 4(b)) that, for the majority of simulated time, the ADS operated in AUTO mode. Furthermore, based on the frequencies of RtIs and emergencies (Figure 4, (c) and (d)), it can be inferred that the simulated environment is relatively dangerous (i.e., emergency alarms and RtIs are not uncommon). It is important to test ADS algorithms under such conditions, as they represent worst-case scenarios. However, as a consequence, it can be observed that the utilities attained are relatively low (Figure 4(a)). Given the perilous nature of the operating environment, the performance of the ODD and environment-monitoring systems is of utmost importance; further analysis of these modules is provided later in this section.

DIPA results are summarized in Figure 5. The vertical axis represents the percentage of simulations for which $q_6 = Pr(\rho > 0.5)$ is greater than the corresponding thresholds on the horizontal axis. It can be observed that, even in this dangerous environment, the driver rarely underperformed. For example, $q_6$ is greater than 0.025 in approximately 1% of the simulations. This is a consequence of the fact that control is only transferred when the estimated probability of the driver being aware is sufficiently large.

4.3.3. Efficacy of ODD and Environment Monitoring. A more in-depth analysis of the safety-related incidents and the RtI behavior further characterizes the performance of the ODD and environment-monitoring strategies. Histograms representing the number of skids, the number of crashes, the proportion of aborted RtIs, and the average time in MANUAL mode per RtI are provided in Figure 6. In this setting, an aborted RtI occurs when the ADS predicts with high probability that ODD limits will be exceeded, but determines that AUTO yields greater utility than MANUAL. Because EMERG is embedded within AUTO, the ADS just stops under such conditions. The data in Figure 6 further illustrate the perilous nature of the simulation environment. More often than not, the ADS encounters some adverse safety situation (i.e., a skid or a crash). However, whenever a crash occurs, the vehicle is
being operated in MANUAL mode by a distracted driver. Therefore, the crashes are induced by human error. Collectively, this implies that, in our simulations, the ODD and environment-monitoring modules generally performed as intended.

4.3.4. Accuracy and Effects of DSM. We can verify the efficacy of our DSM framework by (1) altering the conditions necessitating the issue of an RtI, and (2) requiring that the driver maintains control once a MANUAL mode transition occurs. Figure 7 shows the number of crashes and skids when replicating our large-scale experiment with greater RtI-alarm thresholds and a lesser unfavorable driver-state-alarm threshold (i.e., 0.25, 0.3, and 0.5, respectively). It can be observed that substantially less skids and crashes occur under these settings. This behavior occurs due to the decreased time in MANUAL mode, as depicted in Figure 8(b). The corresponding gain in utility can be seen in Figure 8(a). Such results emphasize the value in a manufacturer optimizing these thresholds in accordance with predefined behavioral objectives.

Further analysis of Figure 9 validates the accuracy of the inferences produced by the ADS. Figure 9(a) represents the estimate $p(\theta_t, \theta^1 | D_t)$ over time. The presence of a vertical bar at a given time indicates that the driver is distracted. From this figure, it can be observed that the estimated probability of the driver being distracted tends to be higher when she actually is distracted. The sensitivity of the forecast appears to concord with a conservative approach; the risk of Type II errors is relatively modest, whereas the risk of
Type I errors is relatively greater. Furthermore, Figure 9(b) represents the estimated probability of cell \( t \) containing a rock, a puddle, or being clean, given that cell \( t - 1 \) contained a puddle. We plot the estimates associated in five different simulation runs. The horizontal lines represent the true values of these probabilities from Table 2. It can be observed that the estimates of these quantities converge to the actual probabilities as the ADS proceeds through time. The accuracy and convergence of these quantities add further credence to our environmental monitoring strategy.

4.3.5. ADS Control in Varied Environments. Given these results, we now consider how the simulation outputs change under varied environmental dynamics. Additional experimentation is provided to

**Figure 5.** (Color online) Percentage of Simulations (y-Axis) for Which \( Pr(\rho > 0.5) \) Is Greater than Different Thresholds (x-Axis)

**Figure 6.** (Color online) Histogram of Relevant Outputs I

(a) Number of skids. (b) Number of crashes. (c) Proportion of aborted RtIs. (d) Average length in MANUAL mode per RtI.

Notes. (a) Number of skids. (b) Number of crashes. (c) Proportion of aborted RtIs. (d) Average length in MANUAL mode per RtI.
provide clarity in this regard. More specifically, we alter the puddle probability—that is, \( p(y_{t+1} = \text{Puddle} \mid y_t = \text{Clean}) \) from Table 2—and observe how the simulated ADS behavior changes. The influence of the rock probability on the simulation outputs was also measured, but is not presented; results were qualitatively similar.

We simulated six different values of the aforementioned puddle probability: 0.1, 0.2, …, 0.6. For each puddle probability, we performed 100 simulations on our 1,000-cell road. Average utility attained is presented in Figure 10(a). Error bars correspond to a single standard deviation. An interesting phenomenon can be observed by inspecting this figure. As might be expected, increasing the puddle probability tends to produce a decrease in average utility. Likewise, when more puddles are expected, the number of RIs increases.

However, when the puddle frequency is sufficiently great, the probability of the driver being aware is also much higher—that is, see Table 5. Therefore, for large values of \( p(y_{t+1} = \text{Puddle} \mid y_t = \text{Clean}) \), the ADS is less likely to encounter a distracted driver. This implies that, when evaluating driving modes, the ADS is more likely to determine that MANUAL is preferable to AUTO, thereby increasing the probability of MANUAL transitions. Nonetheless, because the driver is asked to intervene more often, the collective risk of a distracted driver controlling the vehicle increases. Average utility decreases as a result.

Conversely, when increasing the puddle probability beyond a certain limit (in this example, beyond 0.3), we observe that the utility starts to increase. The reason for this phenomenon is that, when many puddles are expected, the ADS detects driver underperformance...
more often. In turn, the ADS does not allow them to take control over the vehicle as frequently. This behavior can be observed in Figure 10(b), wherein we see a decrease in the risk of a distracted driver in control. Consequently, average utility tends to increase as well.

From this discussion, it is clear that the behavior in Figure 10 is driven by the aforementioned fundamental dilemma in level-3 and -4 ADS. Although the resolution of this ethical dilemma does not have a unitary answer, we have illustrated how ADS behavior in this regard can be effectively guided by expected-utility maximization.

5. Conclusion

We have provided a global Bayesian decision-making model to support the management of driving modes in level-3 and -4 ADS. Our integrated approach simultaneously considers operational design-domain supervision, driver and environment monitoring, trajectory planning, and driver-intervention performance assessment. By jointly leveraging decision analysis and Bayesian forecasting, we created a framework to support both driving-mode management and early warning emission—that is, we developed models and algorithms to control driving modes and issue relevant warnings according to a management-by-exception principle. The efficacy of these methods was illustrated and examined within a generalizable simulated case study.

Nonetheless, there exist numerous avenues of future inquiry related to this research. For example, given that our framework relies on accurate data, sensor performance is of utmost importance. However, such performance may be degraded due to numerous factors (e.g., cyber attacks or temporal degradation). This implies that the monitoring and optimization of sensors is a necessary sequel to this research; the ADS must be able to determine when a sensor should and should not be trusted. Moreover, with regard to driver monitoring, individual differences in aptitudes may affect the degree to which environmental factors affect the driver’s alertness, implying that alternative Bayesian forecasting models may prove more accurate than others. Future research may thus utilize competing forecast models that are evaluated for predictive accuracy using the model-monitoring methods of West and Harrison (2006). Competing forecast models of the environmental variables may similarly improve
the performance of the methods presented herein, especially for ADS, that operate in multiple, distinct environments (e.g., semitrailer trucks crossing international borders). Finally, because the algorithms developed are defined utilizing thresholds, these values are highly influential in driver-mode management and warning emission. In this manuscript, thresholds are treated as exogenously determined parameters; however, future research may seek to optimize them. The extensional application of both reinforcement learning (Sutton and Barto 2018) and response surface methodology (Myers, Montgomery, and Anderson-Cook 2016) hold great promise in this regard.

Disclaimer
The views expressed in this article are those of the authors and do not reflect the official policy or position of the U.S. Air Force, U.S. Department of Defense, or U.S. Government.

References
Agamennoni G, Nieto JI, Nebot EM (2011) A Bayesian approach for driving behavior inference. 2011 IEEE Intelligent Vehicles Sympos. (IV) (IEEE, Piscataway, NJ), 595–600.

Akai N, Hirayama T, Morales LY, Akagi Y, Liu H, Murase H (2019) Driving behavior modeling based on hidden Markov models with driver’s eye-gaze measurement and ego-vehicle localization. 2011 IEEE Intelligent Vehicles Sympos. (IV) (IEEE, Piscataway, NJ), 949–956.

Audet C, Dennis JE Jr (2006) Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. 17(1):188–217.

AutoMate Consortium (2019) Autonome: Revolution in vehicle automation. Accessed October 24, 2019, http://www.automate-project.eu/.

Awad E, Dsouza S, Kim R, Schulz J, Henrich J, Shariﬀ A, Bonnefon J-F, McAllister R, Gal Y, Kendall A, van der Wilk M, Shah A, Cipolla R, Eriksson A, Stanton NA (2017) Takeover time in highly automated vehicles. Noncritical transitions to and from manual control. Human Factors 59(4):689–705.

French S, Insua DR (2000) Statistical Decision Theory (Edward Arnold, London).

Gonzalez D, Perez J, Milanes V, Nashashibi F (2016) A review of motion planning techniques for automated vehicles. IEEE Trans. Intelligent Transportation Systems 17(4):1135–1145.

Hecht T, Feldhutter A, Radlmayr J, Nakano Y, Miki Y, Henle C, Bengler K (2018) A review of driver state monitoring systems in the context of automated driving. Bagnara S, Tartaglia R, Albolino S, Alexander T, Fujita Y, eds. IEA 2018 Proc. 20th Congress Internat. Ergonomics Assoc., Advances in Intelligent Systems and Computing, vol. 823 (Springer International Publishing, Cham, Switzerland), 398–408.

Hillier P, Wright B, Damen P (2014) Readiness for self-driving vehicles in Australia. Workshop Report, 26th Australian Road Research Board Conference, October 19–22, ANZ Stadium, Sydney.

Katrakazas C, Qudous M, Chen WH, Deka L (2015) Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Transportation Res. Part C Emerging Tech. 60:416–442.

Koedsviday A, Soua R, Karray F, Kamel MS (2016) Recent trends in driver safety monitoring systems: State of the art and challenges. IEEE Trans. Vehicular Tech. 66(6):4550–4563.

Kosmides P, Demestichas K, Avgerinakis K, Trouva E, Bianchi S, Barisone A, Rasvas K, et al (2020) Bringing trust to autonomous mobility. 2020 AEIT Internat. Conf. Electrical Electronic Tech. Automotivé AEIT AUTOMOTIVE (Associazione Italiana di Elettrotecnica, Elettronica, Automazione, Informatica e Telecomunicazioni, Milan), TS12.p02.

Li F, Wang W, Feng G, Guo W (2014) Driving intention inference based on dynamic Bayesian networks. Wen Z, Li T, eds. Practical Applications of Intelligent Systems, Advances in Intelligent Systems and Computing, vol. 279 (Springer, Berlin), 1109–1119.

Lin P (2016) Why ethics matters for autonomous cars. Maurer M, Gerdes JC, Lenz B, Winner H, eds. Autonomous Driving: Techni
cal, Legal and Social Aspects (Springer, Berlin), 69–85.

Mahnassani HS (2016) 50th anniversary invited article-autonomous vehicles and connected vehicle systems: Flow and operations considerations. Transportation Sci. 50(4):1140–1162.

McAllister R, Gal Y, Kendall A, van der Wilk M, Shah A, Cipolla R, Weller A (2017) Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning. Sierra C, ed. Proc. 26th Internat. Joint Conf. Artificial Intelligence (AAAI Press, Carles Sierra), 4745–4753.

McCall R, McGee F, Mirnig A, Mesebhcherjakov A, Louveton N, Engel T, Tscheligi M (2019) A taxonomy of autonomous vehicle handover situations. Transportation Res. Part A Policy Pract. 124: 507–522.

Myers RH, Montgomery DC, Anderson-Cook CM (2016) Response Surface Methodology: Process and Product Optimization Using Designed Experiments, Wiley Series in Probability and Statistics (John Wiley & Sons, Hoboken, NJ).

Rios Insua, Caballero, and Naveiro: Managing Driving Modes in ADS

Transportation Science, 2022, vol. 56, no. 5, pp. 1259–1278, © 2022 The Author(s)

de Chalendar JA, Glynn PW (2021) On incorporating forecasts into linear state space model Markov decision processes. Philos. Trans. Roy. Soc. A 379(2120):20190430.

Dong Y, Hu Z, Uchimura K, Murayama N (2011) Driver inattention monitoring system for intelligent vehicles: A review. IEEE Trans. Intelligent Transportation Systems 12(2):596–614.

Eriksson A, Stanton NA (2017) Takeover time in highly automated vehicles: Noncritical transitions to and from manual control. Human Factors 59(4):689–705.

French S, Insua DR (2000) Statistical Decision Theory (Edward Arnold, London).

Gonzalez D, Perez J, Milanes V, Nashashibi F (2016) A review of motion planning techniques for automated vehicles. IEEE Trans. Intelligent Transportation Systems 17(4):1135–1145.

Hecht T, Feldhutter A, Radlmayr J, Nakano Y, Miki Y, Henle C, Bengler K (2018) A review of driver state monitoring systems in the context of automated driving. Bagnara S, Tartaglia R, Albolino S, Alexander T, Fujita Y, eds. IEA 2018 Proc. 20th Congress Internat. Ergonomics Assoc., Advances in Intelligent Systems and Computing, vol. 823 (Springer International Publishing, Cham, Switzerland), 398–408.

Hillier P, Wright B, Damen P (2014) Readiness for self-driving vehicles in Australia. Workshop Report, 26th Australian Road Research Board Conference, October 19–22, ANZ Stadium, Sydney.

Katrakazas C, Qudous M, Chen WH, Deka L (2015) Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Transportation Res. Part C Emerging Tech. 60:416–442.

Koedsviday A, Soua R, Karray F, Kamel MS (2016) Recent trends in driver safety monitoring systems: State of the art and challenges. IEEETrans. Vehicular Tech. 66(6):4550–4563.

Kosmides P, Demestichas K, Avgerinakis K, Trouva E, Bianchi S, Barisone A, Rasvas K, et al (2020) Bringing trust to autonomous mobility. 2020 AEIT Internat. Conf. Electrical Electronic Tech. Automotivé AEIT AUTOMOTIVE (Associazione Italiana di Elettrotecnica, Elettronica, Automazione, Informatica e Telecomunicazioni, Milan), TS12.p02.

Li F, Wang W, Feng G, Guo W (2014) Driving intention inference based on dynamic Bayesian networks. Wen Z, Li T, eds. Practical Applications of Intelligent Systems, Advances in Intelligent Systems and Computing, vol. 279 (Springer, Berlin), 1109–1119.

Lin P (2016) Why ethics matters for autonomous cars. Maurer M, Gerdes JC, Lenz B, Winner H, eds. Autonomous Driving: Technical, Legal and Social Aspects (Springer, Berlin), 69–85.

Mahnassani HS (2016) 50th anniversary invited article-autonomous vehicles and connected vehicle systems: Flow and operations considerations. Transportation Sci. 50(4):1140–1162.

McAllister R, Gal Y, Kendall A, van der Wilk M, Shah A, Cipolla R, Weller A (2017) Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning. Sierra C, ed. Proc. 26th Internat. Joint Conf. Artificial Intelligence (AAAI Press, Carles Sierra), 4745–4753.

McCall R, McGee F, Mirnig A, Mesebhcherjakov A, Louveton N, Engel T, Tscheligi M (2019) A taxonomy of autonomous vehicle handover situations. Transportation Res. Part A Policy Pract. 124: 507–522.

Myers RH, Montgomery DC, Anderson-Cook CM (2016) Response Surface Methodology: Process and Product Optimization Using Designed Experiments, Wiley Series in Probability and Statistics (John Wiley & Sons, Hoboken, NJ).

Rinne TA, Garrott WR, Goodman MJ (2001) NHTSA driver distraction research: Past, present, and future. SAE Technical Paper, Society of Automobile Engineers International, Warrendale, PA.

Society of Automobile Engineers (2018) Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Technical Report, Society of Automobile Engineers International, Warrendale, PA.
Straub J, Zheng S, Fisher JW (2014) Bayesian nonparametric modeling of driver behavior. 2014 IEEE Intelligent Vehicles Symp. Proc. (IEEE, Piscataway, NJ), 932–938.

Stutts JC, Reinfurt DW, Staplin L, Rodgman EA (2001) The role of driver distraction in traffic crashes. Technical Report, AAA Foundation for Traffic Safety, Washington, DC.

Sutton RS, Barto AG (2018) Reinforcement Learning: An Introduction (MIT Press, Cambridge, MA).

Torres R, Ohashi O, Pessin G (2019) A machine-learning approach to distinguish passengers and drivers reading while driving. Sensors 19(14):3174.

Trustonomy (2020) Deliverable 2.2. methodological guidelines. Technical Report, Trustonomy. Accessed February 29, 2020, https://h2020-trustonomy.eu/download/d2-2-deliverable-trustonomy-methodological-guidelines-version-1-0/.

Walch M, Lange K, Baumann M, Weber M (2015) Autonomous driving: Investigating the feasibility of car-driver handover assistance. AutomotiveUI’15 Proc. 7th Internat. Conf. Automotive User Interfaces Interactive Vehicular Appl. (Association for Computing Machinery, New York), 11–18.

Wang Y, Li X, Tian J, Jiang R (2020) Stability analysis of stochastic linear car-following models. Transportation Sci. 54(1):274–297.

West M, Harrison J (2006) Bayesian Forecasting and Dynamic Models, Springer Series in Statistics (Springer, New York).

Yi D, Su J, Liu C, Quddus M, Chen WH (2019b) A machine learning based personalized system for driving state recognition. Transportation Res. Part C Emerging Tech. 105:241–261.

Yi D, Su J, Hu L, Liu C, Quddus M, Dianati M, Chen WH (2019a) Implicit personalization in driving assistance: State-of-the-art and open issues. IEEE Trans. Intelligent Vehicles 5(3):397–413.