Safe Traffic Intersections: Metrics, Tubes, and Prototype Simulation for Solving the Dilemma Zone Problem

Leonard Petnga
Department of Civil and Environmental Engineering
University of Maryland, College Park, MD 20742, USA
Email: lpetnga@umd.edu

Mark A. Austin
Department of Civil and Environmental Engineering
and Institute for Systems Research
University of Maryland, College Park, MD 20742, USA
Email: austin@isr.umd.edu

Abstract—**Our research is concerned with the modeling and design of cyber-physical transportation systems (CPTS), a class of applications where the tight integration of software with physical processes allows for the automated management of system functionality, superior levels of performance, and safety assurance. Part of the safety assurance problem is prevention of deadly accidents at traffic intersections and, in particular, finding ways for vehicles to traverse the dilemma zone (DZ), an area at a traffic intersection where drivers are indecisive on whether to stop or cross at the onset of a yellow light. State-of-the-art approaches to the dilemma zone problem treat the cars and stoplights separately, with the problem formulation being expressed exclusively in either spatial or temporal terms. In this paper, we formulate a methodology that accounts for two-way interactions between the cars and stoplights, and propose quantitative metrics and three-dimensional dilemma tubes as a means for compactly describing sets of conditions for which the vehicle-light system will be in an unsafe state. The proposed metrics enable simple and actionable decision capabilities to deal with unsafe configurations of the system. The second purpose of this paper is to describe a pathway toward the integration of dilemma metrics and dilemma tubes with an ontological framework. The associated platform infrastructure supports algorithmic implementations of simulation and reasoning for resolving unsafe configurations of CPTS, such as those created by the DZ problem.**

Keywords-Dilemma Zone; Metrics; Cyber-Physical Transportation Systems; Artificial Intelligence; Safety.

I. Introduction

This paper describes the development and simulation of metrics for safety analysis of cyber-physical transportation systems (CPTS). It builds upon our previous work [1] on tubes and metrics for solving the dilemma zone problem at traffic intersection. During the past three decades, transportation systems have been transformed by remarkable advances in sensing, computing, communications, and material technologies. The depth and breadth of these advances can be found in superior levels of automobile performance and new approaches to automobile design that are becoming increasing reliant on sensing, electronics, and computing to achieve target levels of functionality, performance and cost. By 2016, as much as 40% of an automobile's value will be embedded software and control related components [2][3]. Looking ahead, even greater levels of automation will be needed for self-driving cars [4][5].

While consumers applaud the benefits of these advances and the products they enable, engineers are faced with a multitude of challenges that are hindering the system-level development of cyber-physical transportation systems (CPTS). These challenges include: (1) the integration of cyber-physical systems (CPS) technologies into existing infrastructure, (2) the realization of "zero fatality" transportation systems, and (3) the development of formal models and credible, actionable performance and safety metrics [6]. To this end, metrics for system safety are needed to: (1) evaluate the operation and control of transportation systems in a consistent and systematic way, (2) identify, measure, and predict dynamic interactions among system components, (3) set standards that serve as measure of effectiveness (MoEs) and can guide model-based systems engineering (MBSE) efforts. And yet, despite these advances, accidents at traffic intersections claim around 2,000 lives annually within the US alone [7]. A key component of this safety problem is the dilemma zone (DZ), which is an area at a traffic intersection where drivers are indecisive on whether to stop or cross at the onset of a yellow light.

In this project, we consider the interplay among the key elements of transportation systems at traffic intersections, and the consequences of their interactions on overall traffic system level safety. This paper focuses on one aspect of the dilemma zone problem, namely, development of metrics to capture the essence of these interactions, and support the characterization of the problem and its representation using three-dimensional dilemma tubes. Section II is a review of existing approaches to the dilemma zone problem and their limitations with regard to the current trend toward CPTS. Section III introduces the new dilemma zone metrics and their tubular representation. Sections IV and V describe the system architecture and simulation prototype, respectively. Metrics for the assessment of safety analysis are introduced in Section VI. The paper concludes with discussion, conclusions and future work.

II. Dilemma Zone Problem and Cyber-Physicality of Traffic Systems

Dilemma Zone: Definition and Existing Solution Approaches. Also called the twilight zone, Amber signal or decision zone, the dilemma zone is the area at a traffic intersection where drivers are indecisive on whether to stop or cross at the onset of a yellow light. Research [8] indicates that under such circumstances only 90% of drivers will "play it safe" and decide to stop. Consequently, the behavior of users
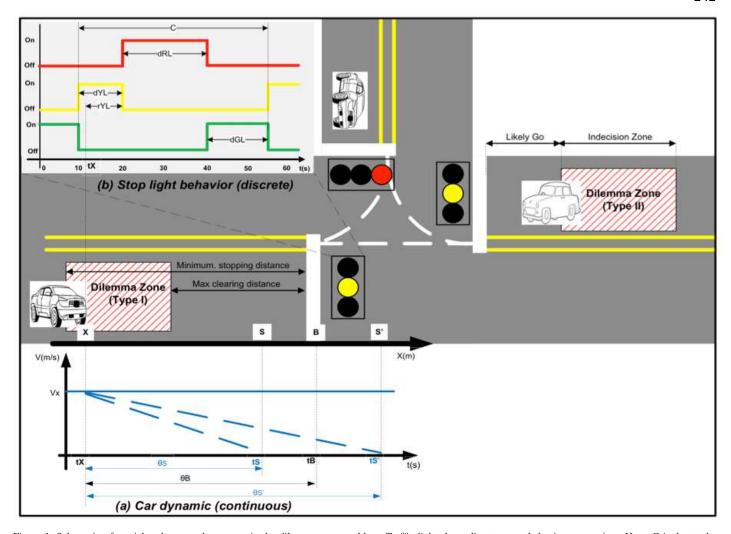
Figure 1. Schematic of spatial and temporal concerns in the dilemma zone problem. Traffic lights have discrete state behavior versus time. Here, C is the total cycle time for the lights. Variables dGL, dYL and dRL represent the duration of the green, yellow, and red lights, respectively. Variables rYL is the time remaining for the yellow light. Vehicles have dynamic behavior that varies continuously with time. Here, θS is the time it takes the vehicle to fully stop before the stopline, θB is the time to reach the intersection while traveling at speed V_x, and $\theta S'$ is the time it takes the vehicle to fully stop after the stopline.

in "twilight zones" is responsible for hundreds of lives lost and billions of dollars in damages at stop light intersections in the United States [7].

From an analysis standpoint (see Figure 1), scholars distinguish two types of dilemma zone that differ by the perspective adopted on the problem. Type I dilemma zone formulations place the "physics of the vehicle" at the center of the problem formulation and are concerned with the difference between the distance from the stop line to the nearest vehicle that can stop safely (i.e., minimum stopping distance) and the distance from the stop line of the farthest vehicle that can cross the intersection at the onset of the yellow light (i.e., maximum clearing distance) [9][10]. Therefore, the physical parameters of the situation (e.g., car speed, road and car conditions, and so forth) are the key determinant of whether the car will be able to safely cross the intersection or stop prior to the stop line. Type II dilemma zone formulations (see the right-hand side of Figure 1) are defined with regard to the driver's behavior and decision-making as the vehicle approaches the intersection and the onset of a yellow light. The boundaries of this type of DZ are also

sometimes measured with a temporal tag (i.e., representing the duration to the stop line) added to the probabilistic estimate [11]. In this work, we will adopt the Type I definition of the dilemma zone.

Past research has focused on finding ways to mitigate, or eliminate, DZs using mostly a pure traffic control engineering view of the problem. These efforts have resulted in signal timing adjustment solutions that ignore or cannot properly account for the physics of vehicles or driver's behaviors [12][13][14]. In order to deal with uncertainties, other scholars have used stochastic approaches such as fuzzy set [9] and Markov chains [10]. For all of these traditional techniques, the baseline of the solution can be either reduced (explicitly or not) to a space- or temporal-based dilemma zone, but not both.

Autonomous Cars and Intelligent Traffic Control Systems. Recent work [15][16] illustrates the switch of researchers' interest toward investigating solutions to the DZ problem
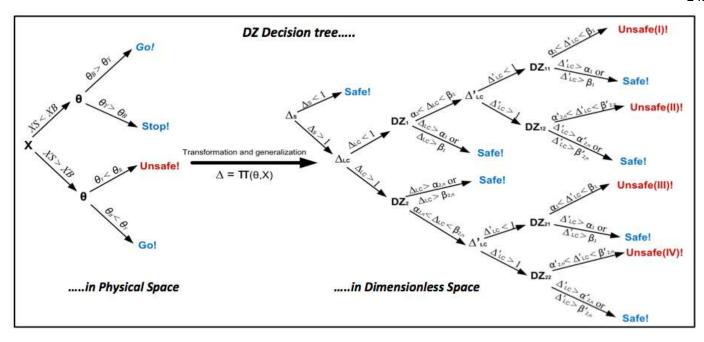
Figure 2. Framework for decision-making. Left: decision-making in the physical space. Right: decision-making in the dimensionless space.

that incorporate both the car physics and light timing, while also providing a pathway forward for vehicle-to-infrastructure (V2I) interactions and integration. These solutions will soon become a reality, in part, because of an increased use of artificial intelligence in automating the command and operation of both cars and traffic signals. For automobiles, many aspects of autonomy – from braking to cruise control and driving functions – are in advanced stages of experimentation. Finding ways to put smartness into vehicles has contributed to reduced fatalities on highways mostly in the developed world. The enhancement of traffic signal controls with artificial intelligence is an idea whose time has arrived – indeed, we now have the capability to determine the position, speed and direction of vehicles, and adjust light cycling times in a coordinated way to make the intersection crossing more efficient. Researchers have been developing and testing various technologies with mixed results [17][18][19]. As a case in point, a pilot study conducted by Carnegie Mellon University, reports a 40% reduction of intersection waiting times, an estimated 26% decrease in travel time, and a projected 21% decrease of CO_2 emissions [19]. Tapping into the full potential of these intelligence capabilities is hindered by practical constraints that include: (1) most vehicles cannot currently communicate with traffic light controllers, and (2) autonomous vehicles still struggle in operating safely in adverse weather conditions (heavy rain, snow covered roads, etc.) and changing environment (temporary traffic signals, potholes, human behaviors, etc.). In this paper, we assume that these problems will be resolved by ongoing research activities.

Toward Cyber-Physical Traffic Management Systems. Real-time situational awareness (e.g., traffic, location, speed) and decision, combined with vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications and control are valid and effective pathways for a solution to both congestion and

safety at intersections. As such, we fully adopt a CPS view of the traffic system with regard to the DZ problem. The value of this perspective has already been demonstrated by Petnga and Austin [20]. Autonomous vehicles (i.e., the physical system) interact with the light (i.e., the cyber system) with the objective of maximizing traffic throughput, while ensuring vehicle crossings are safe at the intersection. Enhanced performance and safety at the intersection have been proven possible, thanks to the critical role of temporal semantics in improving system level decision-making. Also, when bi-directional connections between the vehicle and light are possible, new relationships can be established to characterize their tight coupling – this, in turn, enables the various computers in the CPTS to exchange information, reason, and make informed decisions. These capabilities become safety-critical for situations – hopefully, rare situations – where behavior/physics of a vehicle is such that they can neither stop, nor proceed, without entering and occupying the intersection while the traffic light is red. Therefore, the development of metrics for the DZ problem will greatly benefit from and enrich the CPTS perspective.

III. Metrics for Characterizing the Dilemma Zone Problem

Safety Requirements to Decision Trees and Dilemma Metrics. The core safety requirement for the car-light system that must prevail at all times is as follows: "No vehicle is allowed to cross the intersection when the light is red." This is a hard constraint whose violation is the driving force behind accidents at intersections.

Understanding the mechanisms by which system-level safety is achieved or violated is critical to addressing the DZ challenge. This task is complicated by the need to work with mixtures of continuous (vehicle) and discrete (traffic light) behavior as illustrated in Figure 1 (a) and (b). We propose that decision trees are a suitable framework for representing

the multitude of decision-making pathways. Some of these pathways will correspond to behaviors that are safe. Others will be unsafe and need to be avoided. The tree shown on the left-hand side of Figure 2 shows the decision tree of the autonomous car - in the physical space - when it knows the traffic lights critical parameters at the time the decision is made. Petnga and Austin [20][21] have shown that the probability of the car making the right decision is higher when it knows before hands the following: (1) Duration Θ_Y of the yellow light before it turns red; (2) Vehicle stopping distance XS, and (3) Travel duration, Θ_B, or distance, XB, to the traffic light.

Moving forward requires a deep understanding of the interrelationships between cross-cutting system parameters from the various domains (car, light, time, space) involved at meta level. Also, the ability of the system to efficiently reason about unsafe situations and propose a satisfactory way out is critical. We argue that this complexity can be kept in check by casting the problem in dimensionless terms and setting up a transformation,

$$\Delta = \Pi(\Theta, X), \tag{1}$$

of the initial decision tree from the physical space to a dimensionless space. Expressing the system decision tree in dimensionless space as a result of the transformation Π necessitates the definition of intermediary variables and parameters.

We begin by noting that the car will not always catch the onset of the yellow light; thus, what is really relevant for efficient decision-making here is the time left before the stop light turns red. Using the remaining duration of the yellow light r_{YL}, its full duration d_{YL} and the ones of the green and red lights ie d_{GL} and d_{RL}, we define the duration of a stop light cycle C, reduced cycle C_{YL} and cycle index k as follows:

$$
\begin{align}
C &= d_{YL} + d_{RL} + d_{GL} \tag{2} \\
C_{YL} &= r_{YL} + d_{RL} + d_{GL} \tag{3} \\
k &= \frac{C}{C_{YL}} \tag{4} \\
&\tag{5}
\end{align}
$$

The short (α_1) and full (α_2) yellow light duration as well as the short (β_1) and full (β_2) stop light indexes are defined as follows:

$$
\begin{align}
\alpha_1 &= \frac{r_{YL}}{C_{YL}} \tag{6} \\
\alpha_2 &= \frac{d_{YL}}{C_{YL}} \tag{7} \\
\beta_1 &= \frac{r_{YL} + d_{RL}}{C_{YL}} \tag{8} \\
\beta_2 &= \frac{d_{YL} + d_{RL}}{C_{YL}}. \tag{9}
\end{align}
$$

We add to the aforementioned physical variables the stopping duration Θ_B' of the car – should it decide to stop – and define

the **car stopping distance metric** Δ_S, the **light-car crossing time metric** Δ_{LC} and the **light-car stopping time metric** Δ_{LC}' as follows:

$$
\begin{align}
\Delta_S &= \frac{XS}{XB} \tag{10} \\
\Delta_{LC} &= \frac{\Theta_B}{C_{YL}} \tag{11} \\
\Delta_{LC}' &= \frac{\Theta_B'}{C_{YL}}. \tag{12}
\end{align}
$$

All these metrics are dimensionless and serve as the key decision points of the dimensionless decision tree shown on the right-hand side of Figure 2.

Navigating the Decision Tree. Navigation of the decision tree is facilitated by the equation pair:

$$
\begin{align}
n &= E\left(\frac{\Delta_{LC} - 1}{k}\right) \tag{13} \\
n' &= E\left(\frac{\Delta_{LC}' - 1}{k}\right) \tag{14}
\end{align}
$$

We employ the integer part function E to define indexes n and n'. Equations (13) and (14) simplify the definition of α and β indexes when $\Delta_{LC} > 1$ or $\Delta_{LC}' > 1$ as follows.

$$
\begin{align}
\alpha_{2,n} &= k * \alpha_2 + k * n + 1 \tag{15} \\
\beta_{2,n} &= k * \beta_2 + k * n + 1 \tag{16} \\
\alpha_{2,n}' &= k * \alpha_2 + k * n' + 1 \tag{17} \\
\beta_{2,n}' &= k * \beta_2 + k * n' + 1 \tag{18}
\end{align}
$$

Along with equations (6) through (9), the values of α and β (see equations (15) through (18)) are necessary and sufficient to constrain the dimensionless metrics Δ_S, Δ_{LC} and Δ_{LC}' and render a complete view of all possible outcomes of the decision tree in a dimensionless space Δ. From the right-hand side of Figure 2, we can see that there are four possible configurations of the system for which it is unsafe.

From Dilemma Metrics to Dilemma Tubes. Each of the decision tree pathways on the right-hand side of Figure 2 that leads to an unsafe system state can be represented as a "dilemma tube" in the Δ space, as shown in Figure 3. For instance, equations (6), (8), and (10) through (12) provide the foundational elements for defining Tube I. The boundaries of each of the four tubes (i.e., I, II, III and IV) correspond to the above-mentioned parameters, with the maximum value of Δ_S i.e., Δ_{Smax} corresponding to the maximum value of all the Δ_S values in the system. Physically, this is determined by the physics of the family of vehicles crossing the intersection and the configuration of the traffic intersection as captured by equation (10). If, at any point in time, the system is projected to enter an unsafe state, this situation will be materialized as a point coordinate $P_\Delta(\Delta_S, \Delta_{LC}, \Delta_{LC}')$ that is located inside a particular tube. The physical interpretation of such
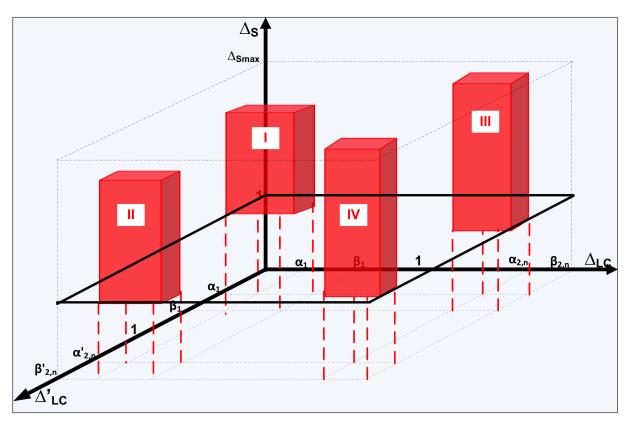
Figure 3. Dilemma tubes in the dimensionless (Δ) space.

phenomenon is that the autonomous car does not have a good decision option, and will need external help to safely cross the intersection.

Scenarios that lead to unsafe system configurations (e.g., see the right-hand side of Figure 2) will follow branches of the decision tree that terminate with an "Unsafe" system state. While the actual behaviors might not evolve along the pathways presented in the decision tree, the end result will invariably be the same (i.e., the system will be projected to enter an unsafe state). In practice, simulation and safety calculations can be done concurrently and the location of the resulting point coordinate relative to any of the four dilemma tube types easily determined. A final important point to note is that since each of the tubes is mutually exclusive, a vehicle can only be in one of the four dilemma tubes at a time, or in any location in the remaining part of the Δ space, i.e., a safe region.

Knowing in which tube the unsafe state has been materialized is critical in determining the appropriate course of action to prevent the occurrence of an accident.

IV. SYSTEM ARCHITECTURE

This section introduces a Java-based software system infrastructure that adheres to the CPTS perspective and supports the tube framework described in Sections II and III. As illustrated in Figure 4, the system architecture contains workspaces for traffic intersection simulation. The main modules of the infrastructure are as follows:

1. Component Modeling. The component modeling module plays a central role in the system simulation. Physical entity models are organized into static and dynamic components, as shown in the mid-section of Figure 4. Examples of the former include the traffic intersection (i.e., the spatial boundary), traffic lights, and their associated sensors. Their key attributes are not expected to change over time such as the stoplight durations d_{YL}, d_{RL} and d_{GL} for the yellow, red and green for each cycle. The remaining duration of the yellow light (r_{YL}) is a key attribute of interest for our study that does decrease with time. As such, the component modeling module needs a clock to account for the elapsed time. In our formulation, sensors play a key role in determining the location (X) and velocity (v) of a vehicle as a function of time. With X and v in place, vehicle accelerations can be computed from the underlying equations of motion. Also, the vehicle braking force (F_b) is subject to change over time; thus, it is a variable of the system.

2. Tube Modeling and Metrics Computation Support. DZ tubes are modeled as software entities because they are not physical entities. In order to properly account for the multiple facets of tubes in this framework, and provide flexibility in the architecture, we propose that tube models serve as a data repository platform and bridge between the computation and the integration modules (see the dashed boxes and connecting arrows in Figure 4).

The interface for the data repository platform distinguishes *base tubes* (not visualized) from *dilemma tubes*. The former
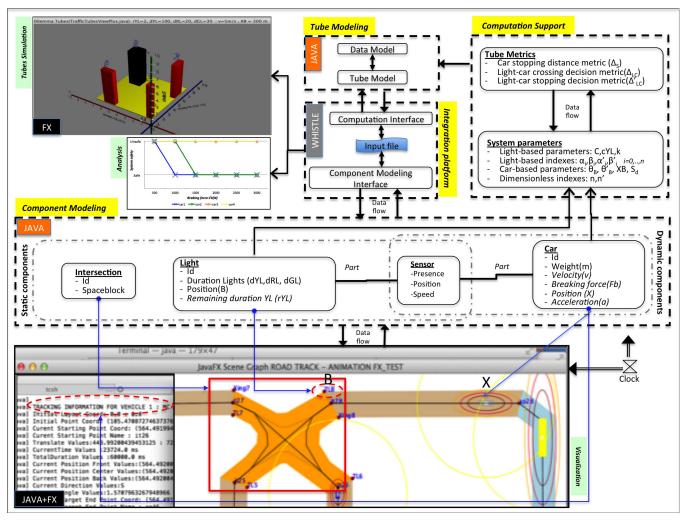
Figure 4. Dilemma tubes simulation system architecture.

store the basic initial configuration of the stop light, and information that will be used to create the latter (i.e., dilemma tubes). Dilemma tubes of various types allow for the representation of unsafe system states as defined by the car stopping distance metric Δ_S, the light-car crossing time metric Δ_{LC}, and the light-car stopping time metric Δ'_{LC} and specifications in equations (4) to (18). This separation of concerns provides modularity and flexibility to the architecture, enabling the support for modeling of complex intersections with multiple stop lights on multi-lanes and/or complex intersection configurations (T,Y,X, etc.).

The visualization system interface (not shown) connects with the integration module, thereby allowing for flows of data to/from the visualization display, and in accordance with the adopted GUI technology. In our software prototype (see the top left-hand corner of Figure 4), the display is controlled from the integration module.

On the interface with the computation support module, a *traffic tube* model is created as an extension of a more basic tube model. It is the ultimate data structure of the tube as it links predefined and computed tubes variables. The initial traffic tube is linked to the base tube, and dilemma tubes are

created from updates of corresponding traffic tubes for various values of r_{YL}. The number of dilemma tubes to be visualized is computed by the system based on values of n and n' as defined by equations (13) and (14).

The computation support module enables the correct calculation of the various metrics and variables needed to efficiently characterize the dilemma zone using the tube framework. It receives input data from both the component and the tube modules, processes computation request following formulae in equations (2) thru (18). We distinguish *system parameters* from the three *tube metrics* Δ_S, Δ_{LC}, Δ'_{LC} introduced above. The former are computed car, light or dimension parameters and indexes that will contribute in the computation of the latter. Dimensionless indexes are parameters as they are, by definition, dependent on Δ_{LC} and Δ'_{LC}. Most of these parameters are defined as attributes of the traffic tube model thus, the results are stored as per the specification of that data structure.

3. System Integration. Reaping the benefits of the system architecture requires bringing together its various modules and pieces in an organized but systematic way. Thus, we need a way to assemble system models for the purpose of the various

TABLE I. Summary of simulation parameters.

Element	Variable	Unit	Min	Max	Set value	Predefined parameters
Car	XB	m	10	60	30	m_1=1,500 kg, m_2=2,800 kg,
	F_b	N	3000	8000	5000	m_3=16,500 kg,
	v	m/s	5	30	10	m_4=24,000 kg
Light	r_{YL}	s	0	5	2	d_{RL}=20s
	d_{YL}	s	3	17	5	d_{GL} = 30s

analysis needs. We solve this problem with Whistle [22][23], a tiny scripting language where physical units are deeply embedded within the basic data types, matrices, branching and looping constructs, and method interfaces to external object-oriented software packages. Whistle is designed for rapid, high-level solutions to software problems, ease of use, and flexibility in gluing application components together. Currently, computational support is added enabling Whistle to handle input and output of model data from/to files in various formats (XML, Open Street Map (OSM), Java, etc.). Therefore, an input file (containing any Whistle-compliant program) is an integral and central part of this module. It provides access to other system modules and needed functionality via interfaces encoded as scripts. Also, the sequencing and timing in the execution of the commands is encoded in the program, giving the analyst/modeler the control of the execution of the simulation.

V. SIMULATION PROTOTYPE

We describe in this section an implementation of the framework for a scenario where the system configuration leads to a system state inside Tube I, as shown in Figure 3. The implementation consists of step-by-step assembly of a (typical) dilemma zone scenario, simulation, and analysis of the results. It is subject to three simplifying assumptions: (A1) the air resistance is negligible, (A2) there is a two-way, delay-free communication between the light and the autonomous car, and (A3) computation and reaction times are negligible.

1. Step-by-Step Assembly of a Real-World Scenario. The step-by-step details are as follows:

(i) A traffic system controller of a smart traffic system computes and stores in real-time each stoplight indexes (C, C_{YL}, k, α_i, β_i, i=1,2) based on its corresponding parameters (r_{YL}, d_{GL}, d_{YL}, d_{RL}) using equations (2) through (12).

(ii) An autonomous car approaching the intersection at speed s is given its distance XB to the stop line in real-time. This information is provided either by its on-board radar coupled with its computer or by the intersection controller. The car itself (autonomous vehicle equipped with camera) notices the onset (or the presence) of the yellow light.

(iii) Based on its current acceleration, speed, road conditions, and maximum applicable braking force, the on-board computer of the car estimates the vehicles stopping distance XS, and computes Δ_S (see equation (10)).

(iv) The computer finds that $\Delta_S > 1$, meaning the car cannot be safely immobilized before the stop line. It then determines the normal travel time θ_B to go through the intersection, i.e., to cover the distance XB, should it decides to go at speed s.

(v) The car requests and obtains from the traffic controller the values of α_i, β_i, i=1,2 and the length of the reduced cycle C_{YL}. It then computes the light-car crossing metric Δ_{LC} using equation (11).

(vi) The on-board computer finds that $\alpha_1 < \Delta_{LC} < \beta_1$. At this point, the only way for the car to avoid violating the safety requirement (i.e., never cross the stop line when the light is red) is to hope that while braking, it will cross the stop line when the line is still yellow.

(vii) Using equation (12), the car determines the travel time θ'_B to cover the distance XB while stopping. Then, it computes the light-car stopping time metric Δ'_{LC}.

(viii) The on-board computer finds that $\alpha_1 < \Delta'_{LC} < \beta_1$, which translates as the light will be already red when the car crosses the stop line while stopping.

Individual values of the metrics Δ_S, Δ_{LC} and Δ'_{LC} generate a point coordinate somewhere within the dilemma Tube I, as pictured in Figure 3. The physical interpretation of this system state is that the vehicle does not have a good decision option, and will need a change of course of action or help from the light to safely cross the intersection.

2. Simulation Setup and Coverage. The simulation setup relies extensively on Java and its advanced graphics and media packages JavaFX as supportive technologies to create, test, debug, and deploy a client application. Simulation coverage consists of four cars c_i, $i \in \{1, 2, 3, 4\}$ of different size (sedan, SUV, bus, cargo truck) and a stop light. Vehicles will be distinguished by their weight (m). Vehicle velocity (v), braking force (F_b) and distance to stop light line (XB) are discrete parameters that can be selected within a predefined range by the modeler/analyst. As for the stop light, the duration of the red light (d_{RL}) and green light (d_{GL}) are treated as constants; the duration of the yellow light (d_{YL}) and the corresponding remaining duration (r_{YL}) are discrete variables within predefined range. The range of each parameter is generally distributed around an average value that is used when a fixed value for a specific parameter is needed. Table I summarizes the case vehicles and parameter values employed in this simulation.

3. Simulation Execution and Dilemma Tubes Visualization. Visualization of the dilemma tubes occurs through a processing pipeline that involves the acquisition, storage, processing, flow and restitution of data between the input file and the visualization platform. For the execution of a scenario involving one car and one stop light, the following steps will be completed:

(1) A user creates an input file containing an execution/simulation program in a Whistle-compliant format. In this application

(a) Whistle input file

(b) Variables and Metrics computation

(c) Tubes visualization (dYL = 100s)

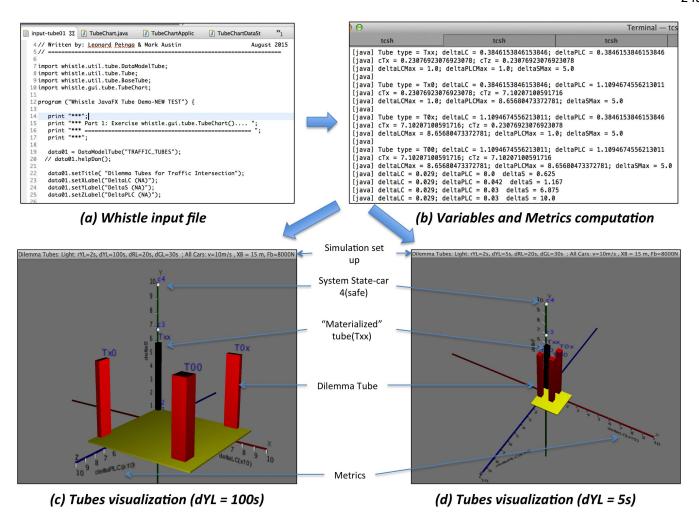*(d) Tubes visualization (dYL = 5s)*

Figure 5. Schematic of system inputs and outputs. The sub-figures are: (a) Whistle input file, (b) variables and metrics computation, (c) tubes visualization for dYL = 100 seconds, and (d) tubes visualization for dYL = 5 seconds.

we use a text file, such as the one shown in Figure 5(a).

(2) The program instantiates a tube DataModel matched to the needs of the simulation. This will later serve as a place holder for the various versions of tubes as they are constructed and displayed.

(3) The system is initialized. This is done by configuring the stop light with predefined values to d_{YL}, d_{RL} and d_{GL}. As for the car, if the engineering simulation module (e.g., racetrack) is hooked to the integration platform, then a car type is selected based upon its weight and its physical parameters (initial velocity, trajectory and position). The corresponding component models are interfaced with the integration module.

Computational requirements during the simulation can be reduced through pre-computation and storage of the dilemma tube parameters, as described in the following steps (4)-(7). This is done for various values of r_{YL} and dimensionless indexes n and n' (see equations (13) and (14)).

(4) The number of dilemma tubes N that need to be visualized at each iteration of r_{YL} is determined as follows:

$$N = \begin{cases} 1 & \text{if } n \text{ and } n' \text{ are undefined} \\ n+2 & \text{if } n \geq 0 \text{ and } n' \text{ undefined} \\ n'+2 & \text{if } n \text{ undefined and } n' \geq 0 \\ (n+2)(n'+2) & \text{if } n' \geq 0 \text{ and } n \geq 0 \end{cases} \quad (19)$$

In equation (19), n is undefined when $\Delta_{LC} < 1$ and n' is undefined when $\Delta'_{LC} < 1$. In this configuration, the only tubes that can be viewed are of Type I, as per Figure 3.

(5) From the input file, a method of the tube DataModel file is called to generate a baseline empty tube as per the initial configuration of the traffic light. This results in the creation and storage of a new BaseTube that acts as a placeholder for the set durations of the three lights. For simulations involving multiple stoplights, the same method can be called repeatedly for each set of stoplights. Each call of this method will result in a TrafficTube model being created and instantiated.

(6) Next, a new method is called to create and update dilemma tubes for the given input baseline tube. This leads to: (a) the calling of the traffic tube instance, the extraction and storage

of the set value for d_{YL}, then, (b) the creation of the dilemma tubes via an update of the traffic tube for the decreasing values of r_{YL} from d_{YL} to 0. Besides the value of r_{YL}, the values of n and n' as well as the input baseline tube are needed. The foundational variables needed to display each dilemma tube are computed, i.e., the tube type, dimensions on axis and coordinates of their location in the dimensionless (delta) space, as shown in Figure 5(b). The total number of dilemma tubes created is determined, as per equation (19). In this case, we have $n = n' = 0$, which leads to four dilemma tubes, Txx, Txo, Tox and Too which are of types I, II, III and IV, respectively.

(7) The dilemma tubes are sorted and grouped by r_{YL}. This information will allow control of the display of tubes in a way that is consistent with the unfolding of r_{YL}

(8) With the computation and storage of dilemma tubes completed, we can now make the move toward their visualization. The first step consists of enabling Whistle access to the visualization tube model in order to create an instance of a JavaFX 3D chart. For those cases where the engineering simulation module is hooked to Whistle, the racetrack and its contents will be uploaded and displayed as per the set up in (3). Otherwise, the simulation can be done with the system state in the dimensionless space computed separately based on the initial set up and targeted configurations.

(9) The 3D scene for the tube charts is created then, the data stream system is configured and the data (flow) channel tube between the input file and the 3D GUI is created and initialized.

(10) The simulation of the engineering module is started. As the car follows the path toward the intersection stop line located at B, its position X is sensed. The remaining duration on the yellow light r_{YL} is measured from the clock. Both quantities are sent back to the computation module for processing. For each pair (XB, r_{YL}), the values of Δ_{LC}, Δ_S and Δ'_{LC} are computed as per equations (10), (11) and (12). As a group, these values define the state of the system in the Δ space.

(11) The set of dilemma tubes corresponding to the value of r_{YL} is pulled from storage (see step 7) and "pushed" through the channel (see step 9) to the display GUI. We can now visualize an output similar to the ones shown in Figures 5(c) and (d). The yellow plate is the *Plan Tube* for the system in the $(\Delta_{LC}, \Delta'_{LC})$ space. It is built from the maximum values of both parameters for the set of dilemma tubes available for display and defines the system boundary at $\Delta_S = 1$ for which the dilemma tubes take shape.

(12) Identification mechanisms are encoded into the channel system to single out *materialized tube(s)* – that is, tubes for which the safety of the system has to be checked. Materialized tubes are within the immediate vicinity of a system state and, as such, depending on how compact the tube system is, there could be many of them. There is always at least one materialized tube at any moment (in black in Figure 5(c) and (d)). When a materialized tube contains a system state, it means that the system is unsafe. Such cases are quantified as "active tubes." We note here that the physical interpretation of an active tube is not that of an actual violation of the system

safety constraint (see Section III), but that it will happen in the immediate future, and certainly within the time left on the yellow light (if any).

(13) Configuration of the tube system. The way the tubes appear on the visualization GUI depends on the values of dimensionless indexes n and n'. To identify the formation of the tubes, we look at the tubes from the top view in the plan $(\Delta'_{LC}, \Delta_{LC})$ in the computer screen reference system, i.e., with Δ'_{LC} pointing downward and Δ_{LC} pointing right. As for the value of N in equation (19), four types of formation are possible:

$$TubeFormation = \begin{cases} point & \text{if } n \text{ and } n' \text{ are undefined} \\ line & \text{if } n \geq 0 \text{ and } n' \text{ undefined} \\ I & \text{if } n \text{ undefined and } n' \geq 0 \\ rectangle & \text{if } n' \geq 0 \text{ and } n \geq 0 \end{cases}$$
(20)

In the *point formation* the only tube that can be displayed is of Type I. In the *line formation*, realized tubes appear aligned horizontally on an axis parallel to the Δ_{LC} axis. A similar formation is observed in the *I formation* with the tubes being aligned vertically following the Δ'_{LC} in the dimensionless space. The boundary of the last type of formation has the shape of a rectangle. When $n = n'$, it becomes a square as for the four-tube formation in Figure 5 (c).

VI. SAFETY ANALYSES

The purposes of this section are two-fold. First, we employ the simulation platform described in Section V to identify and analyze the key factors that affect the system level safety of the traffic system. In the second part of this section, single and set-pair factor safety analyses are performed to investigate how system safety depends on systematic adjustments to single factors (e.g., vehicle braking force) and combined sets of parameters.

1. Safety Factors Identification. Under the set of assumptions (A1) to (A3), and from Table I, the following six factors are single out for further consideration: weigh of the car(m), car velocity(v), car braking force(Fb), distance to stoplight (XB), remaining duration of the yellow light (r_{YL}), and configured duration (d_{YL}). For these studies we pick $n = n' = 0$ which leads to a four-tube square formation.

2. Single Factor Safety Analysis.

a/ Effect of Car Weight and Velocity. For this analysis, we use the set of four cars and assign for each simulation run a velocity within the range in Table I with a step of $5m/s$. The remaining four parameters are fixed to their set value. For each run, we observe and record the presence and name of any active tube (synonym of unsafe system) as well as the identity of the car whose state has been materialized in the active tube. The absence of any active tube means the system is safe for all vehicles. The results are summarized in a *parameter-based safety profile* as shown in Figure 6(a).

For this particular configuration of the traffic system, the active tube for all runs is the tube Txx, which is of Type I. The
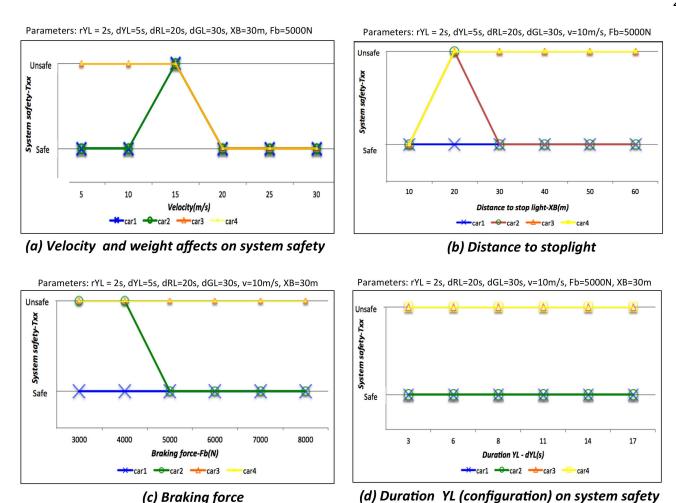
Figure 6. Parameters-based single factor safety profiles.

heavier cars (#3 and #4) violate the safety constraint at lower speed ($v \leq 15m/s$), while small and mid-size vehicles (#1 and #2) would not violate the safety constraint if they operate on both sides of velocity $v = 15m/s$. The combined effects of inertia and velocity play against safety (i.e., heavier cars lack agility – at velocity $v \leq 15m/s$, they can neither stop before nor clear the intersection within the 2s time interval). We note the troubling "unsafe" state for all cars at $v = 15m/s$. To summarize, operating heavier vehicles within higher velocity range and, small and average size vehicle at lower or higher velocities are the only way to keep the traffic system safe.

A quick evaluation of the sensitivity of the safety profile to changes in any of the fixed parameters shows that the only one for which it doesn't change significantly is d_{YL}. For instance, if we consider changes in r_{YL}, smaller and mid-size vehicles become safer as long as r_{YL} grows beyond $2s$ ($3s$ for heavier vehicles). At lower r_{YL} ($\leq 1s$), all vehicles tend to be unsafe except for smaller ones at low velocity ($v \leq 10m/s$). Given the relatively far distance ($XB = 30$ m) at which this evaluation is performed, there might still be room for improvement as the car gets closer to the intersection stop line, especially at low velocities.

b/ Effects of the Car Distance to the Intersection. For this study, we use the same set of four cars and keep track of the distance to the stop line, this time with a step of $10m$ which is used to define the location of sensing points for the system. And as with the previous analysis, the remaining four parameters are fixed to their set value. System safety is tracked by observing and recording the presence and name of active tubes along with the identity of the car whose state has been materialized in the active tube. Finally, the distance-to-stop-line safety profile (see Figure 6(b)) is generated.

We observe that as heavier vehicles (#3 and #4) approach the intersection, they are mostly unsafe until the last checkpoint, where their dynamic capabilities allow them to either stop safely before or clear the intersection within the remaining $2s$ on the yellow light. The small vehicle (#1) is safe all the time; with the exception of checkpoint $XB = 20m$ (which corresponds to the last location where heavier vehicles transition to a safe state), the mid-size vehicle (#2) are safe. An examination of the sensitivity of this profile to perturbations in r_{YL} reveals that heavier cars are more sensitive than mid-size and small cars. Away from the light ($XB \geq 50m$), heavier cars are unsafe and they will require $5s$, $4s$ and $3s$ on r_{YL}, respectively at $40m$, $30m$ and $20m$ to avoid violating the

intersection safety requirement. Mid-size vehicles, in contrast, only require $3s$ at $20m$ to stop.

c/ Effects of the Car Braking Force. The same protocol is followed to study how car braking force affects system safety. To that end, we systematically vary the parameter F_b within the defined range in Table I using a $1000N$ step. This results in the braking force safety profile shown in Figure 6(c).

For this configuration of the system, the effect of the braking force is well perceived for the mid-size car (#2) as it leaves the unsafe state when F_b increases and passes the $5,000N$ threshold. Under the same circumstances, heavier cars (#3 and #4) certainly need a braking force outside the current simulation range – in fact, our set value for the maximum force of $8,000N$ does not help switch the system back into safety. In other words, even a $8,000N$ braking force is insufficient to counter the kinetic energy of the vehicles and immobilize them within $XB = 30m$ and $r_{YL} = 2s$. Small cars are much more agile, and the minimum braking force of $3,000N$ is good enough to keep the smallest car (#1) safe.

As the value of r_{YL} decreases, the safety profile for car #1 is not affected as all for all values of F_b. However, below $5,000N$, the mid-size and heavier cars would require $r_{YL} \leq 4s$ to remain safe. Above that threshold force, only heavier car will need the same amount of time to stay safe. Thus, we can conclude that the higher the inertia of the vehicle, the higher breaking force and time on yellow light are needed for the system to remain safe.

d/ Effects of the Initial Configuration of the Yellow Light. As a final step in this experiment, we would like to understand how the configuration of the stoplight by the traffic engineer and, in particular, the duration of the yellow light d_{YL}, affects the system safety. To that end, we consider a fixed stoplight cycle duration $C = 55s$ and assign a progressively increasingly high percentage of that duration to the yellow light from 5% to 30% with a step of 5%; thus, the data range shown in Table I. The simulation is run for the various values of d_{YL} and results of the safety profile are shown in Figure 6(d).

We see from the safety profile that, for a given value of $r_{YL} = 2s$, increasing the actual configuration of the yellow light does not affect the outcome of system safety. However, a look at the corresponding tube formation shows that, as the value of d_{YL} increases, so is the spacing between the tubes. This translates into more room for safety, should the system manage to get out of unsafe situations, i.e., the volume occupied by the tubes. The contrast between the tube formations in Figures 5(c) and (d) illustrates this phenomenon. When $d_{YL} = 5s$, a low value, the rectangle formation is compact, and the tubes are closed to each other (see Figure 5 (d)). Should they realize all, there will be little to no room to avoid a violation of the safety constraint. Conversely, at higher $d_{YL} = 100s$ (for illustration only) there is plenty of room between the tubes. This means that, should there exist a mechanism to take advantage of the availability of this safety space to adjust r_{YL} to higher values, the safety of the system will be improved. These observations make the case for reconfigurable traffic lights that are capable of adjusting the remaining duration of the yellow light to resolve safety issues. Also, we note the variation in tube sizes in Figures 5(d) and 5(c), with Txx being the smaller and Too the bigger.

This observation can be traced back to index k, as per equation (4), and its further propagation into the parameters that define the tubes as shown in Figure 3, especially those defined by equations (15) to (18). Finally, we note that $0 \leq r_{YL} \leq d_{YL}$ thus, the two variables are dependent. Setting d_{YL} from an initial position d_{YL1} to $d_{YL2} \geq d_{YL1}$ allows r_{YL} to add $d_{YL2} - d_{YL1}$ to its range which, as we have seen so far, adds more safe room for the overall system.
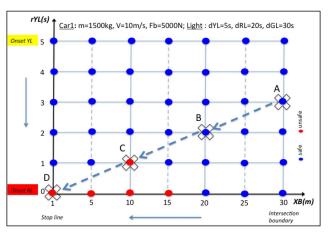
3. Set (pair) Factor Safety Analysis. Despite the valuable insight provided by single factor analyses in understanding system level safety, they provide just a "snapshot" view of the system through the perspective of the parameter considered for the analysis. The sensitivity of most safety profiles to changes in the values of r_{YL} clearly shows that even though most factors are set or controlled independently, their interaction is the key driver behind system level safety. Thus, there is a need to look at changes to system safety caused by adjustments to combined sets of parameters.
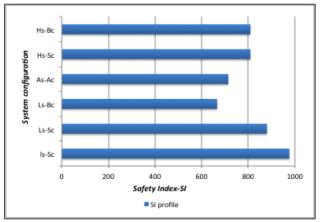
a/ Parameter-based Safety Template for Pair (r_{YL}, XB). Pairing the six parameters leads to fifteen possible sets. However, given that parameters such as r_{YL} and d_{YL} are dependent and others such as m and XB are constrained by the vehicle physics, not two sets of parameters are equally important or relevant for this study. Thus, we won't be analyzing the system safety for all pairs, but we will be looking at the pair (r_{YL}, XB), which illustrates the cyber-physicality of the traffic system as introduced in Section II. The protocol of the study described here can be repeated and applied to other pairs as well.

For set factor studies, all the parameters considered vary within their individual, predefined range. The other parameters are configured to their set values as presented in Table I. Running the simulation and recording the safety state of the system results in the creation of a parameter-based *safety template*, such as the one seen on Figure 7(a). This particular template is created with the configuration: $K \equiv (m = 1,500kg, v = 10m/s, Fb = 5,000N, d_{YL} = 5s, d_{RL} = 20s, d_{GL} = 30s)$. The template shows the safety state of each system operational point. A red dot signifies that under K, the system state is in an active tube (i.e., the system is unsafe). A blue dot means the system is safe. In practical terms, the template is an indicator of safety – for instance, under configuration K, if car #1 crosses the intersection boundary (XB=30m) when there is only $3s$ left on the yellow light, the system will be safe as it will be located at $A(30m, 3s)$, which is a safe operational point on the template. If, however, the configuration K remains unchanged, the system will be unsafe $2s$ later at location $C(10m, 1s)$. Therefore, for the system to remain safe under K, the car has to enter the intersection when there is at least $4s$ left on the yellow light. These examples illustrate the greater insight, we can gain using safety templates, in the interplay between system parameters and their effects on system level safety.

b/ Parameter-based Safety Indexes for Pair (r_{YL}, XB). A subspace Us that contains all unsafe states of the system for the configuration K can be defined as follows:

$$Us^K_{(r_{YL}, XB)} = \begin{cases} 0s \leq r_{YL} \leq 1s \\ 1m \leq XB \leq 15m. \end{cases} \quad (21)$$

(a) Safety template for the set (XB,rYL)   *(b) Safety Index chart for set (XB, rYL)*

Figure 7. Parameters-based safety templates and indexes.

Intuitively, one might think that a smaller subspace Us translates to a safer system, but this is only part of the story. Considering that an unsafe subspace might also contain safe states, as observed in this case, we ought to be able to quantitatively assess the safety of a configuration in a clear and simple way. To this end, we introduce the parameter-based *configuration safety index* SI as follows:

$$SI^K_{(r_{YL},XB)} = \left(1 - \frac{n_{U_K}}{n_K}\right) * 1000. \tag{22}$$

Here, n_{U_K} is the number of unsafe states (red dots) in Us and n_K the total number of states in the template for configuration K. For the safety template shown in Figure 7(a), we count $n_{U_K} = 5$ unsafe states and $n_K = 6 * 7 = 42$ total states. This leads to a configuration safety index of $SI^K_{(r_{YL},XB)} = 880$.

By systematically adjusting the vehicle weight (m) and velocity (v) we can generate an ensemble of safety templates, and then for each, compute the safety index. This leads to the safety index chart shown in Figure 7(b). The chart shows that for high speeds, both the smallest vehicle (S_c) and heaviest vehicle (B_c) have similar levels of safety. The smallest vehicle does a better job at lower velocities. In-between, the mid-size vehicle (A_c) cannot do better at average velocity (A_s). These results are consistent with the findings in 1.a/.

We note that this safety index does not capture the topology of unsafe and safe points in the Us subspace for (r_{YL}, XB). As seen in a/ above, that distribution is critical in predicting the future state of the system. Therefore, we cannot use the safety index SI to that same end. However, it can be used for a high level estimate of the parameter-based safety appreciation of the system safety before diving into topological considerations of Us for further investigation. To that extent, the two approaches serve complementary purposes.

4. Beyond Predefined Configurations and Pair Factors. Any change in the value of a parameter in the configuration K in Section 3.a/ automatically forces the switch and use of a different safety template (with the new value for that parameter) to predict the state of the system when the car reaches the stop line. This limits the ability of the Systems Engineer to navigate the design space of the traffic system. A possible solution is to flatten all independent variables in a pentagon-like diagram which will give a partial view of the whole design space. The actual full design space is much more complex (i.e., a five-dimensional shape) and almost impossible to visualize. Any combination of values of the five parameters (m, v, F_b, r_{YL}, XB), each within its respective range, is theoretically a valid point.

VII. DISCUSSION

Our preliminary results are contingent upon assumptions (A1) through (A3) listed in Section V. Neglecting air resistance (A1) certainly simplifies the account of the dynamics of the cars but it comes at a price. With the acceleration null, the velocity is assumed constant on XB which leads to a constant value of Θ_B in equation (10) for all vehicles at the same velocity for the the same value of XB. This propagates all the way to the tubes visualization where, under such circumstances, points for the various cars will be stuck in the plan (Δ_S, Δ'_{LC}) at a single Δ_{LC} value. One opportunity for future work is to account for the air resistance in the dynamics of the car, through a drag force $f = k_1 * v^2$ for instance. This will lead to a more accurate model of the vehicle dynamic that will ultimately improve the quality of the results. The immediate effect on this tube framework will be the distribution of system states along the axis Δ_{LC} as well.

Task execution of the scenario introduced in Section V requires intensive computations and communication at multiple steps; this makes it hard for assumptions (A2) and (A3) to survive any physical prototype testing of the system. In fact, as many researchers have pointed out, not only do real-world computations and communication require finite amounts of time to complete [24][25], but delays of unacceptable duration can trigger accidents in traffic scenarios that are safety critical. Given that such considerations are platform-dependent, there should be in a future iteration of this work a mechanism to

account for delay information in the execution model, perhaps along the lines of what has been accomplished with Ptolemy [26].

VIII.   Conclusion and Future Work

The purpose of this paper has been to introduce and describe a new and innovative tubular (3D) characterization of the dilemma zone problem. We have discussed the modeling, design and prototype simulation of a tubular framework that supports the study and analysis of the dilemma zone problem using a set of dimensionless metrics.

State-of-the-art approaches to the dilemma zone problem treat the cars and stoplights separately, with the problem formulation being expressed exclusively in either spatial or temporal terms. By taking on a systems perspective that allows for two-way interactions between the cars and stoplights, the proposed method leads to a dilemma tubes formulation that compactly describe sets of conditions for which the vehicle-light system will be in an unsafe state.

The essential elements of the two-way interaction are formally captured by three metrics: (1) the car stopping distance metric Δ_S, (2) the light-car crossing time metric Δ_{LC}, and (3) the light-car stopping time metric Δ'_{LC} in dimensionless space (Δ). These three metrics work together to define a simple and precise way safety of the system in a manner that is consistent with the system decision tree. To support this formulation we have developed a flexible software architecture for the computation of metrics and implementation of the tubes. Simulations were performed and tubes were visualized under sets of physical and cyber parameters for the car and the light extracted from the system design space.

The single safety factor analysis indicates that system-level safety is strongly influenced by the combined effect of car weight m and velocity v, its distance to the stop light XB, and the configuration of the yellow light d_{YL}. Parameter-based safety templates, which are effective in predicting the future state of the system at the stop line, were created by pairing the remaining duration of the yellow light r_{YL} and XB. We have defined a parameter-based safety index SI as a first-order estimate of system level safety. This new metric enables the characterization and comparison of safety templates. All of these analyses work together to provide a deeper understanding of the dilemma zone problem and strategies for resolving unsafe scenarios. The proposed approach and preliminary results are consistent with research that has investigated the critical role of component interactions on the safety of complex systems [27].

Future versions of this work need to fully embrace the cyber-physicality of next generation traffic systems as described in Section II. Key characteristics of these developments would include semantically-enabled and efficient platform structures that can support the modeling, emulation and simulation of the behavior of real-world autonomous cars and intelligent traffic control systems as agent of cyber-physical transportation systems. To that end, an ontological architecture supporting the formal description of the relevant sub-domains involved is needed. Spatio-temporal reasoning supported by appropriately implemented semantic extensions (such as Jscience or Joda time) will enhance traffic agents

decision-making capabilities. For the traffic system, the architectural framework will support reasoning in the dimensionless space and enable light reconfiguration, should a car be heading into a dilemma tube. The dilemma metrics introduced in this paper will be implemented in the Integrator rules engine. This entity (physically a smart traffic controller) will be the ultimate responsible of system-level decisions. Further details on the underlying semantic platform infrastructure supporting this architecture can be found in Petnga and Austin [28].

References

[1]   L. Petnga and M. A. Austin, "Tubes and Metrics for Solving the Dilemma-Zone Problem," The Tenth International Conference on Systems(ICONS 2015), Barcelona, Spain, April 19 - 24, 2015, pp. 119–124.

[2]   J. Sztipanovits, J. A. Stankovic, and D. E. Cornan, "Industry-Academy Collaboration in Cyber-Physical Systems(CPS) Research," *White Paper August 31*, White Paper, August 31, 2009.

[3]   D. Winter, "Cyber-Physical Systems in Aerospace - Challenges and Opportunities," *The Boeing Company*, Safe & Secure Systems & Software Symposium (S5), Beavercreek, Ohio USA, June 14-16, 2011.

[4]   Google inc., "The Latest Chapter for the Self-Driving Car: Mastering City Street Driving," Official Google Blog. N.p., n.d. Web. April, 28 2014, http://googleblog.blogspot.com/2014/04/the-latest-chapter-for-self-driving-car.html; Accessed : November 15, 2015.

[5]   General Motors Corporation (GMC), "GM Unveils EN-V Concept: A Vision for Future Urban Mobility," http://media.gm.com; Accessed : November 15, 2015.

[6]   Energetics Incorporated, "Cyber-physical Systems Situation Analysis of Current Trends, Technologies, and Challenges," Energetics Incorporated for the National Institute of Standards and Technology (NIST), 2012.

[7]   D. Hurwitz, "The "Twilight Zone" of Traffic costs lives at Stoplight Intersections," Oregon State University, Corvallis, Oregon, USA, March 03, 2012.

[8]   C. V. Zeeger and R. C. Deen, "Green-Extension Systems at High-Speed Intersections," Division of Research, Bureau of Highways, Department of Transportation, Commonwealth of Kentucky, April, 1978.

[9]   D. S. Hurwitz, B. H. Wang, M. A. Knodler, D. Ni, and D. Moore, "Fuzzy Sets to Describe Driver Behavior in the Dilemma Zone of High-Speed Signalized Intersections," School of Civil and Construction Engineering, Oregon State University, USA and Department of Civil and Environmental Engineering, University of Massachusetts Amherst, USA, March, 2012.

[10] P. Li, "Stochastic Methods for Dilemma Zone Protection at Signalized Intersections," Doctor of Philosophy Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University, VA, USA, August, 2009.

[11] M.S. Chang, C.J. Messer, and A. J. Santiago "Timing Traffic Signal Change Intervals based on Driver Behavior," Transportation Research Record, 1027, pp. 20-30, 1985.

[12] P.D. Pant and Y. Cheng, "Dilemma Zone Protection and Signal Coordination at Closely-Spaced High-Speed Intersections," Report FHWA/OH-2001/12, Ohio Department of Transportation, Columbus, OH, USA, November, 2001.

[13] D. Maas, "Dilemma Zone Elimination," Sacramento Department of Transportation, Sacramento, CA, USA, 2008.

[14] K. Zimmerman and J.A. Bonneson, "Number of vehicles in the dilemma zone as a potential measure of intersection safety at high-speed signalized intersections," 83rd Annual Meeting of the Transportation Research Board Washington, D.C., USA, January, 2004.

[15] N.G. Wassim, J. Koopmann, J.D. Smith, and J. Brewer, "Frequency of Target Crashes for IntelliDrive Safety Systems," US Department of Transportation – National Highway Transportation Safety Administration, DOT HS 811 381, October, 2010.

[16] J. Chu, "At a Crossroads: New Research Predicts which cars are likeliest to Run Lights at Intersections," http://newsoffice.mit.edu/2011/driving-algorithm-1130; Accessed: November 15, 2015.

[17] Siemens Corporation, "Intelligent traffic solutions," http://www.siemens.com; Accessed : November 15, 2015.

[18] International Business Machine (IBM) Corporation, "IBM Intelligent Transportation Solution for Active Traffic Management," Systems and Technology Group, IBM Corporation, November, 2013.

[19] Carnegie Mellon University (CMU), "Smart Traffic Signals," http://www.cmu.edu/homepage/computing/2012/fall/smart-traffic-signals.shtml; Accessed : November 15, 2015.

[20] L. Petnga and M. A. Austin, "Ontologies of Time and Time-based Reasoning for MBSE of Cyber-Physical Systems," 11th Annual Conference on Systems Engineering Research (CSER 2013), Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

[21] L. Petnga and M. A. Austin, "Cyber-Physical Architecture for Modeling and Enhanced Operations of Connected-Vehicle Systems", 2nd International Conference on Connected Vehicles (ICCVE 2013). Las Vegas, NV, December 2-6, 2013.

[22] P. Delgoshaei, M.A. Austin, and D. A. Veronica, "A Semantic Platform Infrastructure for Requirements Traceability and System Assessment," The Ninth International Conference on Systems (ICONS2014), Nice, France, February 23 - 27, 2014.

[23] P. Delgoshaei, M.A. Austin, and A. Pertzborn, "A Semantic Framework for Modeling and Simulation of Cyber-Physical Systems," International Journal On Advances in Systems and Measurements, Vol. 7, No. 3-4, December, 2014, pp. 223–238.

[24] E. Lee, "Computing Needs Time," Technical Report No. UCB/EECS-2009-30, Electrical Engineering and Computer Sciences University of California at Berkeley, February 18, 2009.

[25] R. Wilhelm and D. Grund, "Computing takes time, but how much?," Communications of the ACM, Vol. 57, Issue. 2, February, 2014, pp. 94–103.

[26] C. Ptolemaeus, Editor, "System Design, Modeling, and Simulation using Ptolemy II", Ptolemy.org, 2014.

[27] N. C. Leveson, "A Systems-Theoretic Approach to Safety in Software-Intensive Systems," IEEE Transactions on Dependable and Secure Computing, Vol. 1, Issue. 1, January-March 2004, pp. 66-86.

[28] L. Petnga and M. A. Austin, "Semantic Platforms for Cyber-Physical Systems," 24th Annual International Council on Systems Engineering International Symposium (INCOSE 2014). Las Vegas, USA, June 30 July 3, 2014.