Toward Autonomous Vehicle Safety Verification from Mobile Cyber-Physical Systems Perspective

Sailesh Kandula, Tridib Mukherjee, and Sandeep K. S. Gupta
The IMPACT Laboratory
School of Computing & Informatics
Arizona State University, Tempe, AZ
http://impact.asu.edu/

ABSTRACT

Safety certification of Autonomous Vehicles (AVs) require guarantees on AVs' safety at design time. To this effect, this paper proposes modeling abstractions that allow architectural representation of AVs and their surroundings, i.e. representation of different components, and enable safety analysis from such representation without requiring any expertise on formal methods. Toward this direction, AVs are considered as Cyber-Physical Systems with Mobile computing nodes (MCPS), where each node (i.e. an AV) can have intentional (as determined by AVs' controller) and unintentional (e.g., in case of skids) motion characteristics depending on the physical environment (e.g. road condition). The modeling abstractions are used to analyze safety of passengers in an AV that collides with guard rail due to skid along a curved segment on the AZ-83 highway.

Categories and Subject Descriptors: I.2.9 [Robotics]: Autonomous vehicles; I.6.5 [Model Development]: Modeling methodologies

General Terms: Safety, Model-based verification.

Keywords: Mobile cyber-physical systems.

1. INTRODUCTION AND MOTIVATION

Autonomous Vehicles (AVs) are getting increasing attention from the research community in recent years [3]. A major concern however is the lack of usable verification and certification techniques [9] of AVs to ensure safety of passengers. Research on AVs' safety have principally focused on: (a) formal methods (requiring rigorous formal modeling and analysis expertise) to verify safe behavior of AVs' control system [11, 20]; and (b) designing motion control and collision avoidance algorithms that ensure safe behavior [10, 18]. This paper seeks to complement such approaches by facilitating intuitive modeling (through abstractions of system components and their interactions) and safety analysis (at design time) from a system architectural perspective so that usable tools can be developed for safety verification (and certification). The modeling abstractions should be modular to allow easy representation of complex scenarios involving heterogeneous vehicle characteristics, control and planning algorithms, and various physical scenarios (involving road characteristics and other static and moving objects in the surroundings of an AV)[1].

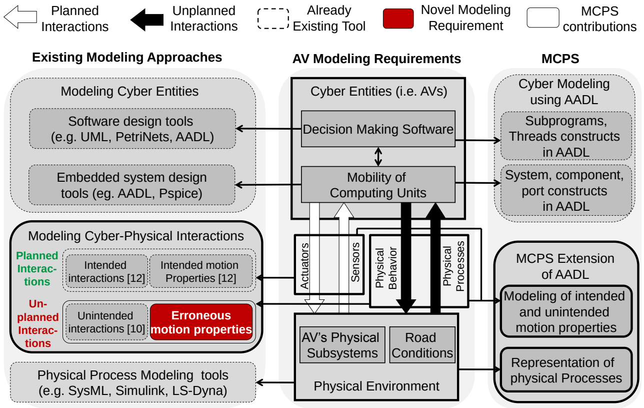
Planned Interactions
Unplanned Interactions
Already Existing Tool
Novel Modeling Requirement
MCPS contributions
Existing Modeling Approaches
AV Modeling Requirements
MCPS
Modeling Cyber Entities
Software design tools (e.g. UML, PetriNets, AADL)
Embedded system design tools (eg. AADL, Pspice)
Modeling Cyber-Physical Interactions
Planned Interac- tions
Intended interactions [12]
Intended motion Properties [12]
Un- planned Interac- tions
Unintended interactions [10]
Erroneous motion properties
Physical Process Modeling tools (e.g. SysML, Simulink, LS-Dyna)
Cyber Entities (i.e. AVs)
Decision Making Software
Mobility of Computing Units
Actuators
Sensors
Physical Behavior
Physical Processes
AV's Physical Subsystems
Road Conditions
Physical Environment
Cyber Modeling using AADL
Subprograms, Threads constructs in AADL
System, component, port constructs in AADL
MCPS Extension of AADL
Modeling of intended and unintended motion properties
Representation of physical Processes

Figure 1: MCPS contributions—modeling abstractions to capture unintended motion properties.

Architectural modeling and analysis of control systems for avionics and smart-cars has been limited to verification of fault-free (or fault-tolerant) software and hardware components [16, 2]. However, since safety vulnerabilities can be caused by and can also impact conditions in the physical environment, it is important to capture the inter-dependencies of the cyber (i.e. software and hardware) components with the physical environment (similar to the formal methods [11, 20] where the physical dynamics have been modeled as dynamic systems). Architectural modeling of Cyber-Physical Systems (CPSs), i.e. systems with strong inter-dependencies among cyber and physical components, have investigated safety verification mainly for applications where the cyber components are static, i.e. they do not change their relative position with time [13]. Such verification declares the system safe when any impact of intended and un-intended interaction from the static cyber components on the physical environment is within a threshold.

Properties of intended mobility of cyber components have

[1] Note that unlike the conventional automobile simulation software (e.g., LS-Dyna [5]), the analysis of the modeling abstractions need to consider the behavior of the control and planning algorithms in AVs.

been captured in architectural modeling for CPSs by Bhave et al. [17]. Safety vulnerabilities for AVs can however occur due to: (i) sensing errors, leading to inaccurate or even incorrect estimation of the AVs' surroundings (i.e. unintended interaction); (ii) controller errors, that can cause incorrect decision making regarding the actions to perform (e.g., acceleration, deceleration, and steering); (iii) actuation error, that can cause inaccurate execution of the actions (i.e. unintended interaction); and (iv) dynamics in the physical environment, e.g. ice on road or curvature of road segment can cause the vehicle to skid (i.e. un-intentional motion behavior). Safety criteria for AVs have to be holistic in nature that captures the extent of damage to the physical world. In this regard, it is important that the criteria is generic enough to enable system designers to define safety depending on the scenario; e.g., safety can be avoiding vehicle collisions [11, 20], or it can be minimizing human injuries when collisions are inevitable (as in skid situations).

This paper proposes a preliminary modeling framework that captures the intended and un-intended impact of mobile computing units, e.g. AVs, on the physical environment, e.g. passengers. We refer to a CPS with mobile computing nodes as Mobile CPS (MCPS). The main contributions are:

1. **modeling abstractions** for MCPS to capture computing nodes' intended and un-intended mobility; and
2. **safety verification** of AVs from their architectural model using the MCPS constructs.

Fig. 1 depicts modeling requirements for AVs, the shortcoming of related research, and the contributions of MCPS modeling abstractions (presented in the next section).

2. MODELING ABSTRACTIONS

Four basic modeling abstractions (constructs) are identified: (i) *MCPS*, (ii) *LCPS*, (iii) *safety threshold*, and (iv) *analysis parameters.* The constructs are hierarchically shown in Fig. 2. The first construct, *MCPS*, is at the top of the hierarchy, while the remaining three are sub-constructs of *MCPS.* Autonomous vehicles, passengers traveling in them and the road conditions are modeled as a Local CPS (LCPS). MCPS consists multiple LCPSs. The *LCPS* construct has the following sub-constructs: (a) *Computing Node*, (b) *Physical System Parameters*, (c) *Unintended Region of Mobility (UIROm)*, and (d) *Intended Region of Mobility (IROm).*

The *computing node* construct models a computing system inside LCPS. A computing node (e.g., an AV) has computing and physical properties associated with it; both of these are modeled by a *Computing Properties* construct. A list of computing properties that affect safety of physical system are modeled as *Impacting Parameters.* An example of impacting parameter is velocity of AV. Errors in sensor and actuators are modeled in *Error Parameters.* Properties that describe physical system's behavior are modeled using *Physical System Properties.* This construct has a sub-construct, *Impacted Parameters*, which lists subset of physical system parameters that are affected by the impacting parameters. Safety threshold can be defined on one of the impacted parameters. The *UIROm* construct is used to specify the computing node's behavior in case there is an unintended mobility. It has three sub-constructs: (i) *Computing Mobility*, which can be used to specify equations that describe the mobility of computing nodes (e.g., motion equations of AVs); (ii) *Minimum Threshold*, which allows specification of thresholds on impacting parameters beyond which unintended mobility (e.g. skids for AVs) can occur; and (iii) *Physical Process*, which can be used to specify: (a) mathematical equations/table describing the relation between impacting and impacted parameters; and (b) any control logic/fault tolerant mechanism that tries to minimize or avoid any unintended motion. For example, traction control systems that are used to minimize skidding can be modeled by the *physical process* construct.

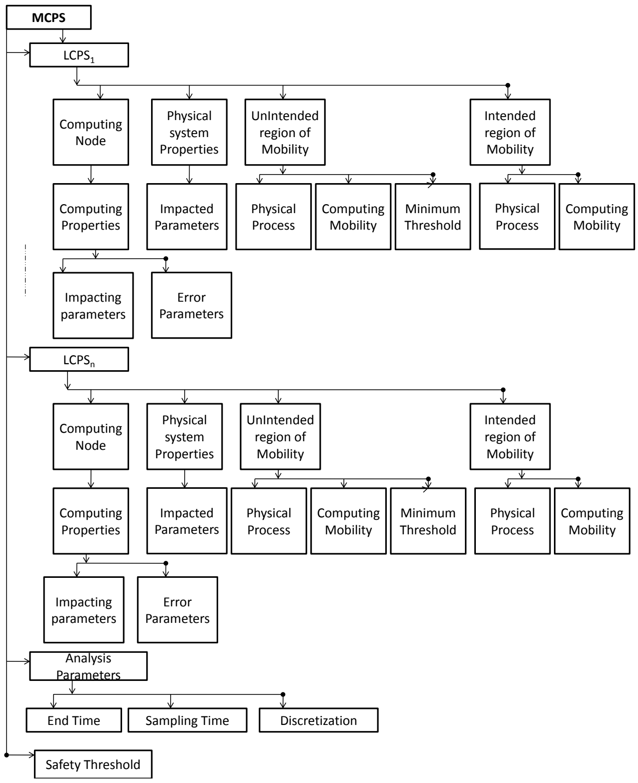

Figure 2: MCPS constructs

The *IROm* construct is used to model the mobility and physical behavior of a computing node during planned motion. The planned trajectory can be based on the navigation and path planning algorithm of the AV. *Computing Mobility* and *Physical Process* sub-constructs (similar to *UIROm* construct) are used for this purpose. The *safety threshold* construct allows specification of the threshold value. The *analysis parameters* construct allows system designers to specify parameters that control the accuracy and complexity of safety verification. There are three sub-constructs of this construct: *time duration*, which allows specification of the duration for which the system will be analyzed; *sampling time*, which is used to specify the frequency at which the overlapping of motion is checked[2]; and *discretization*, which is an optional sub-construct that allows system designers to provide differential equation solver parameters (e.g., initial condition, boundary conditions, and discretization).

3. CASE STUDY

The applicability of MCPS modeling constructs is demonstrated through a case study which involves determining the

[2] The sampling time affects complexity of safety verification algorithm as well as accuracy of analysis results.

Table 1: Model Parameters

Parameter	Value
Coefficient of Friction	0.65
Mass of AV	1700kg
Type of AV	pickup truck
Sampling Time	0.005s
Radius of lane	72m
Distance between Guard Rail and AV	3ft
Type of Guard Rail	W Bean
Total Time Duration	5s

safety of passengers traveling in a autonomous pickup truck (AV) on Mile Post 44 (a horizontal curve) on Arizona 83 highway [19]. The AV can potentially skid along the horizontal curve and collide with a guard rail. A recent report [8] published by National Cooperative Highway Research Program (NCHRP) identifies that the average crash rate along horizontal curves is three times more than any other horizontal road segment. Fig. 3 shows direction of AV's motion (the dashed arrow in the figure) along the horizontal curve. The trajectory is assumed to be along the center of the lane.

Table 1 lists: (i) the parameters of the curve such as curve radius [19], distance between the AV and the guard rail [7], the coefficient of friction of road [4], and the type of guard rail along the curve [6]; (ii) the parameters of the AV such as type of AV (i.e. pickup truck), AV's mass (i.e. the mass of a pickup truck) [6] and (iii) the parameters related to the safety verification algorithm such as sampling time (i.e. the time between two consecutive computations of AV's speed) and the total time duration for which safety analysis is performed. The passenger traveling in an AV is considered unsafe (in case of skid along the horizontal curve) if the probability of a serious injury is non-zero. This probability depends on the change in AV velocity, x, in a collision (i.e. reduction in velocity after impact), and is computed as [12]:

$$P = \frac{1}{1 + exp(4.0139 - 0.1252x)}. \qquad (1)$$

It is important to determine the speed and the angle at which the AV collides with the guard rail in order to compute the velocity after collision [5]. These parameters can be obtained by analyzing the behavior of AV along the curve.

3.1 Behavior Modeling using MCPS constructs

AV's properties such as its mass, velocity and steering angle (generated by AV's control algorithm [15]), wheel base, and its type (i.e. pickup truck) are modeled as **computing properties**. Velocity (a.k.a. impact velocity) and angle (a.k.a. impact angle) at which the AV collides with the guard rails are **impacting parameters**. Impact angle, i.e. the angle at which the AV collides the guard rail, is shown in Fig. 3 as β. Impact velocity is the tangential velocity of the AV along the curve. The properties of the curve, i.e. radius, coefficient of friction, type of guard rail, distance between the AV and guard rail are modeled as **Physical System Parameters**. Human injury is an **impacted parameter**. The behavior of an AV during skid is modeled as **UIROm**. The condition for a vehicle to skid, specified using the **Minimum Threshold** construct, depends on the frictional and centrifugal forces on the vehicle. The tangential direction of AV's skid is modeled using **Computing Mobility**. The constructs are implemented as an annex to Abstract Architecture Description Language (AADL) standard [2][3].

3.2 Safety Analysis

Based on the MCPS model, the safety verification is performed by computing the probability of serious injury to passengers. The computation is based on speed of AV (an impacting parameter) using control algorithm specified as part of *Physical Process* construct. The speed is checked to determine if there is a skid by using equations specified in *Minimum Threshold* construct. The final velocity after collision is computed using a pre-computed table that maps the impact velocity and angle to final velocity. The table can be obtained by offline simulation of vehicles. For this case-study, we used LS-Dyna [5], a finite element simulation software generally used to determine vehicle deacceleration and deformation during crash. The final velocity is then used to compute probability of serious human injury (Eq. 1). If the probability is non-zero the system is declared as unsafe. Fig. 4 shows the general steps of safety analysis.

3.3 Validation

The correctness of safety analysis is evaluated by comparing the probability of severe injury to passengers with the probability of having serious accident by a pick up truck along MP44 based on the data provided in AZ-83 report [19]. We assume the speed of AV follows a Normal distribution with mean of 55 MPH and standard deviation of 6.8. These values are based on data published by Arizona Department of Transportation report [1]. The validation has two steps:

Probability of serious injury as per safety analysis: The probabilities of serious injury at speeds between 45-60MPH are computed (Eq. 1) and multiplied with the probability distribution of vehicle speed. The sum of the resulting values determines the probability of serious injury.

Probability of serious injury (as per [19]): Authors in [19] report that the total number of accidents during 2002-2008 along MP 44 were 240. Out of these 10% of accidents were due to pick up trucks, 14% of these accidents were serious in nature (incapacitated injury, fatal injury etc), 74% of these accidents were due to speeding. Multiplying all these values gives the total number of accidents due to speeding of pickup truck leading to serious passenger injury as 3.12, dividing this value by 240 gives the probability which is 0.013. The probability of serious injury from safety analysis is 0.014; whereas the probability when computed using data from AZ-83 assessment [19] is 0.013.

4. DISCUSSION

The MCPS modeling constructs can be applied by the system engineers to perform safety verification without requiring any specific analytical expertise. The constructs need to be applied in complex systems with multiple AVs and complicated road segments and intersections. The MCPS and LCPS constructs need to incorporate the behavior of AV when it reacts to uncertain events. Further, the MCPS modeling constructs are generic in nature to specify different types of control behavior or even different domains such as first responder applications, e.g., a building under fire. In such case, first responders can use mobile networks to communicate among themselves and use location information from fire sensors to identify severe damages. Extreme temperatures can cause localization errors [14] and fire fighters

[3] The implementation can be accessed from `http://impact.asu.edu/MCPS/AV_MCPS_AADLModel.aadl`.

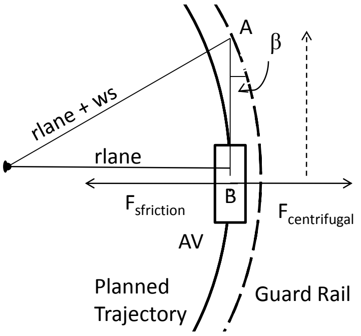

Figure 3: Autonomous Vehicle driving along a Horizontal Curve. Here, $rlane$ is the radius of planned trajectory, $F_{centrifugal}$ and $F_{sfriction}$ represent centrifugal force (i.e. an outward acting force that pushes the vehicle away from the center) and frictional force (that provides necessary traction towards the center), respectively.

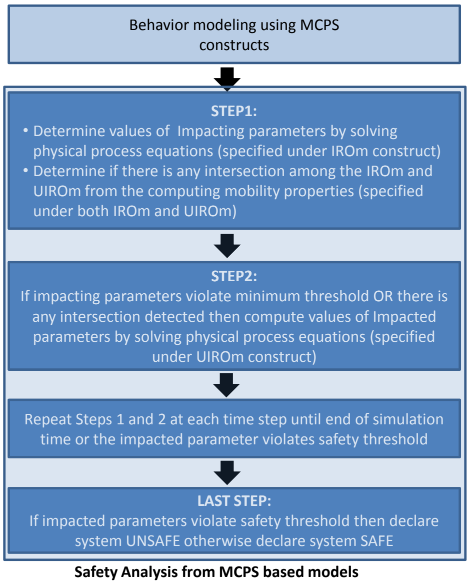
Behavior modeling using MCPS constructs
STEP1:
• Determine values of Impacting parameters by solving physical process equations (specified under IROm construct)
• Determine if there is any intersection among the IROm and UIROm from the computing mobility properties (specified under both IROm and UIROm)
STEP2:
If impacting parameters violate minimum threshold OR there is any intersection detected then compute values of Impacted parameters by solving physical process equations (specified under UIROm construct)
Repeat Steps 1 and 2 at each time step until end of simulation time or the impacted parameter violates safety threshold
LAST STEP:
If impacted parameters violate safety threshold then declare system UNSAFE otherwise declare system SAFE
Safety Analysis from MCPS based models

Figure 4: Steps in safety analysis.

can often be redirected to incorrect locations thereby delaying evacuation. We can model incorrect locations and behavior of sensor using *UIRom* and *Physical Process* constructs. The safety criteria can be a threshold on victim's physiological state under smoke asphyxiation.

Acknowledgments

The research is funded in part by NSF CNS grant #0831544. The authors are thankful to Ayan Banerjee for technical insights on model-based analysis, Prof. S. D. Rajan, director of the Computational Mechanics Lab, for allowing access to LS-Dyna, and Aditya Vaidya for assistance in LS-Dyna.

5. REFERENCES

[1] Actual Speeds on the road compared to posted speed limits, Final Report 551, October 2004.

[2] Architecture Analysis and Design Language. `http://www.aadl.info/aadl/currentsite/`.

[3] Darpa Grand Challenge. `http://www.darpa.mil/grandchallenge/index.asp`.

[4] Engineers Handbook. `http://www.engineershandbook.com/Tables/frictioncoefficients.htm`.

[5] LSDyna. `http://www.lstc.com/lsdyna.htm`.

[6] National Crash Analysis Center at George Washington University. `http://www.ncac.gwu.edu/vml/models.html`.

[7] Shoulder Width published by US department of Federal Highway Administration. `http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3%/3_shoulderwidth.htm`.

[8] Volume 7:NCHRP Report 500, A Guide for reducing collisions on horizontal curves, 2007.

[9] M. Althoff et. al. Safety assessment of autonomous cars using verification techniques. In *Proc. of the American Control Conference, 2007*, pages 4154–4159, 2007.

[10] M. Althoff et. al. Model-based probabilistic collision detection in autonomous driving. *Trans. Intell. Transport. Sys.*, 10:299–310, June 2009.

[11] M. Althoff et. al. Safety verification of autonomous vehicles for coordinated evasive maneuvers. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 1078 –1083, june 2010.

[12] G. Bahouth et. al. Development of URGENCY 2.1 for the Prediction of Crash Injury Severity.

[13] A. Banerjee et. al. BAND-AiDe: A tool for cyber-physical oriented analysis and design of body area networks and devices. *ACM Transactions on Embedded Computing Systems (TECS) (to appear)*, 2011.

[14] K. Bannister et. al. Wireless Sensor Networking for Hot Applications: Effects of Temperature on Signal Strength, Data Collection and Localization. In *Hotmnets '08*, 2008.

[15] A. M. Bedford et. al. *Engineering Mechanics: Statics and Dynamics(5th Edition)*. Prentice Hall, 2008.

[16] B. Berthomieu et. al. Formal Verification of AADL models with Fiacre and Tina. In *ERTSS 2010 – 5th International Congress and Exhibition on Embedded Real-Time Software and Systems*, may 2010.

[17] A. Bhave et. al. Augmenting software architectures with physical components. In *Proceedings of the Embedded Real Time Software and Systems Conference (ERTS)*, 19-21 May 2010.

[18] J. Bruce et. al. Real-Time Multi-Robot Motion Planning with Safe Dynamics. In *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pages 159–170. Springer Netherlands, 2005.

[19] T. Tech. AZ-83 roadway assessment report,Rosemont copper project, 2009.

[20] T. Wongpiromsarn et. al. Verification of Periodically Controlled Hybrid Systems: Application to An Autonomous Vehicle. *Special Issue of the ACM Transactions on Embedded Computing Systems (TECS)*, 2010. To appear.