Reachability Computation of Low-Order Models for the Safety Verification of High-Order Road Vehicle Models

Matthias Althoff and John M. Dolan

***Abstract*— We present an approach to verify the planned maneuvers of an automated car. The main idea is to compute the occupancy of the automated car on the road using reachable sets, which makes it possible to check if one collides with other traffic participants, or leaves the drivable area. The specialty of the presented approach is that all possible uncertainties in the form of sensor noise, uncertain friction coefficient, and uncertain initial states, are considered. Maneuvers are periodically verified on-board to account for the variety of possible traffic situations, requiring an efficient algorithm. Thus, the underlying vehicle model has to be a compromise between accuracy and simplicity. The inexactness of the model is compensated by adding disturbance to the model such that it contains high-order model behavior. This is demonstrated by exploring the state space with rapidly-exploring random trees (RRTs) of a high-order model and check whether it leaves the reachable area of the low-order model used for verification.**

I. Introduction

A major motivation for developing (semi-)automated cars is the vision of accident-free driving, which can also be seen as the precondition for making automated cars a reality. The main challenges in verifying the safety of those vehicles is that (i) every traffic situation is different, (ii) the vehicle behavior has to be safe considering all sources of uncertainty, (iii) the vehicle has to be safe even when certain decision-making components fail. In this work we address all three issues: Our approach is flexible by verifying each traffic situation individually on-board. We compute all possible states which the ego-vehicle and other traffic participants can reach from a set of possible initial states, under a set of possible inputs and parameters. We also describe a fail-safe verification procedure, i.e., the vehicle comes to a safe stop even when decision-making components fail.

Since the safety verification relies on mathematical models of vehicle behavior, the result can only be as good as the model describing the real behavior. Due to the time constraints of the verification procedure, the vehicle dynamics model has to be chosen such that only the main effects are considered. However, in this work, we show that even high-order models are represented by the reachability analysis of low-order models when the set of initial states is enlarged and disturbance is added. From now on, we refer to this property as behavior inclusion.

We relax the problem of behavior inclusion by checking only a finite number of test maneuvers: evasive maneuver, moose test, and cornering. One way to check behavior inclusion is to check if the reachable set of the high-order model is enclosed by the one of the low-order model. Note that this requires to project the states of the high-order model onto the ones of the low-order model. However, the reachable set computation of the high-order model is too challenging for current reachable set algorithms due to the large number of states and the large nonlinearity measure. Instead, we try to falsify behavior inclusion by searching for states of the high-order model, which are not in the reachable set of the low-order model. We use rapidly-exploring random trees (RRTs) to guide the simulation such that interesting simulation traces are further explored, while uninteresting ones are abandoned. If no falsification can be found during intensive offline testing, the reachable set of the low-order model (computed online) is assumed to contain all high-order behaviors.

Matthias Althoff is with Faculty of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA, email: malthoff@ece.cmu.edu

John Dolan is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, email: jmd@cs.cmu.edu

A. Related Work

There is a rich literature on reachability analysis of dynamical systems with continuous or hybrid (mixed discrete/continuous) dynamics. Many of the recent advances are summarized in [1] and the references therein. Since the vehicle model in this paper has nonlinear continuous dynamics, we focus on this class of systems: Most approaches compute reachable sets of nonlinear systems by abstracting to differential inclusions of simpler dynamics, either by simplifying the dynamics within regions of a fixed state space partition [2], [3], resulting in a hybrid system, or by simplification in the vicinity of the reachable set [4]–[6]. The latter approach generally outperforms fixed state space partitions since it does not require the consideration of hybrid dynamics. Approaches which do not use abstraction are mostly based on optimization techniques which are computationally too expensive for an online verification [7]–[9]. The method applied in this work is based on [5], which uses zonotopes as a set representation for nonlinear systems in contrast to the other referenced approaches. As a consequence, the proposed approach, which abstracts to linear systems, is efficient, since zonotopes show great performance for linear systems [10].

There is a rich literature on finding counterexamples for (safety) specifications of dynamic systems. In this work, we use RRTs, which were originally developed for planning problems in robotics [11]. The extension to other control problems, such as discrete and hybrid systems, is described in [12]. Recently, variants of the classical RRT algorithm have been used for falsifying properties of dynamic systems by

optimizing the coverage of the state space [13]–[15]. Other coverage-based methods find trajectories that are representative for neighboring trajectories based on sensitivity analysis [16] or approximate bisimulation metrics [17]. While RRTs are specialized for increasing coverage of the state space, another line of Monte-Carlo-based techniques guides simulations such that temporal logic properties are falsified with high probability [18].

In a previous work, the authors verified maneuvers of automated cars using reachability analysis [19]. This work is an extension in many respects: We use RRTs to check whether the high-order dynamics is enclosed by computing the reachable set of the low-order model when enlarging uncertainties. In addition, we consider uncertain road-tire friction, which is considered the most influential and unpredictable parameter of the vehicle dynamics, and we consider load transfers in the low-order model to improve the reachable set results. Finally, a description of how the trajectory planner interacts with the verification module is presented.

B. Outline

In Sec. II we introduce the fail-safe procedure for safety verification. The models of the low- and high-order vehicle dynamics, including the control law of the vehicle, are shown in Sec. III. The reachable set computation is presented in Sec. IV and the RRT algorithm in Sec. V. Finally, the results of the reachability analysis and the model falsification are presented in Sec. VI.

II. Fail-Safe Verification Procedure

Given a reference trajectory of the vehicle planner, the presented verification procedure decides if this trajectory can be safely followed. This decision is made based on the occupancy of other traffic participants and the ego-vehicle. If the occupancy of the ego-vehicle does not intersect with that of other vehicles for any of the considered consecutive time intervals $[t_k, t_{k+1}]$, and does not leave the drivable area, the maneuver is safe (see Fig. 1).

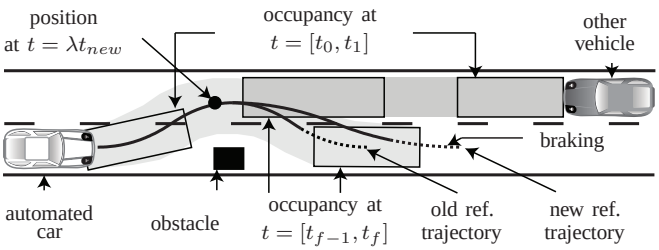


Fig. 1. Verification by checking emptiness of occupancy intersection.

The occupancy of other vehicles is based on an assumption of maximum acceleration and a simple set of traffic rules, such as respecting speed limits and driving in dedicated lanes (see the example in [19]). If the other traffic participant is an automated car, it might simply broadcast its own predicted occupancy. The occupancy of the ego-vehicle has to be computed in more detail, since it has to be checked whether the reference trajectory can be tracked under all disturbances and if the deviation is small enough to avoid crashes. In this paper, we focus on the computation of the reachable set of the ego-vehicle. The road occupancy is obtained from the reachable set by enlarging the set of center of gravity positions based on the vehicle body dimensions and the set of vehicle orientations [19].

The interaction between the trajectory planner and the verification planner is important, especially since the trajectory planner might not find a safe trajectory on time. Thus, we propose a fail-safe approach, i.e., even if the planning algorithm or the verification algorithm does not terminate on time, the automated vehicle comes to a safe stop. In order to achieve the fail-safe property, we consider reference trajectories that consist of two parts: The first part describes the actual maneuver the vehicle should follow, and the second part describes a braking trajectory, which brings the vehicle to a safe stop (see Fig. 1). *Safe stop* means that the vehicle should not stop in, e.g., lanes with oncoming traffic or in railroad intersections. Note that the braking maneuver is only executed when the vehicle does not find a new trajectory on time.

In order to ensure that the vehicle always follows a verified maneuver, new maneuvers are restricted to branch off previously verified maneuvers at specific positions. These positions are chosen such that the new maneuver is already verified when the vehicle approaches it. This is predictable since the required time of the proposed verification algorithm is proportional to the execution time of the planned maneuver. Given the ratio $\lambda = \frac{t_{ver}}{t_{exec}}$ of verification time t_{ver} to maneuver execution time t_{exec}, the planning algorithm has to plan a reference trajectory which branches off the previous one after λt_{new} time, where t_{new} is the execution time of the new maneuver (see Fig. 1). In case the verification takes unexpectedly longer, one can still use the previously verified one. The vehicle model for checking these plans is presented next.

III. Vehicle Dynamics

We briefly introduce the low-order dynamic model used for the reachability analysis and the high-order model to check the validity of the reachable sets.

A. Low-Order Model

The basis of the low-order model is a bicycle model which describes the basic effects of the lateral dynamics for constant velocity, which is, e.g., used for yaw stabilization of vehicles. The name *bicycle model* refers to the fact that the front and rear wheel pairs are each lumped into one wheel, since the roll dynamics is not considered (see Fig. 2 and [20, Chap. 2.6]). In order to model the controlled vehicle, we add the possibility to accelerate the vehicle and include the equations describing the position on the road. These two enhancements result in the model described in our previous work [19]. In the current work, we additionally consider the load transfer of the vehicle due to longitudinal acceleration a_x (neglecting suspension dynamics), such that the vertical

forces on the front and rear axis $F_{z,f}$ and $F_{z,r}$ become

$$F_{z,f} = \frac{mgl_r - ma_x h}{l_r + l_f}, \quad F_{z,r} = \frac{mgl_f + ma_x h}{l_r + l_f},$$

with parameters from Tab. I. These forces are inserted into the derivation of the equations for the slip angle (at the center of gravity) β and the yaw rate $\dot{\Psi}$ [20, Chap. 2.6]. The dynamics for the inputs δ (steering angle), a_x (longitudinal acceleration), and the parameters in Tab. I are:

$$\begin{aligned}
\dot{\beta} =& \frac{\mu}{v(l_r + l_f)} \Big(C_{S,f}(gl_r - a_x h)\delta \\
& - (C_{S,r}(gl_f + a_x h) + C_{S,f}(gl_r - a_x h))\beta \\
& + (C_{S,r}(gl_f + a_x h)l_r - C_{S,f}(gl_r - a_x h)l_f)\frac{\dot{\Psi}}{v} \Big) - \dot{\Psi} \\
\ddot{\Psi} =& \frac{\mu m}{I_z(l_r + l_f)} \Big(l_f C_{S,f}(gl_r - a_x h)\delta \\
& + (l_r C_{S,r}(gl_f + a_x h) - l_f C_{S,f}(gl_r - a_x h))\,\beta \\
& - \left(l_f^2 C_{S,f}(gl_r - a_x h) + l_r^2 C_{S,r}(gl_f + a_x h) \right) \frac{\dot{\Psi}}{v} \Big) \\
\dot{v} =& a_x \\
\dot{s}_x =& v\cos(\beta + \Psi) \\
\dot{s}_y =& v\sin(\beta + \Psi)
\end{aligned}$$

Rewriting the above equations in state space form yields a 6-dimensional model with the state vector $x = [\beta,\, \Psi,\, \dot{\Psi},\, v,\, s_x,\, s_y]^T$. Note that we do not use the cornering stiffness C, as is typically done for bicycle models, but separate the effect of the friction coefficient μ, the cornering stiffness coefficient C_S, and the vertical force F_z, such that $C_i = \mu C_{S,i} F_{z,i}$ and $i = \{f, r\}$ for the front and rear axle. This separation is done because the friction coefficient is the most dominant parameter, which is investigated later. The uncertainty of the friction is specified by an interval in Tab. I, representing dry conditions.

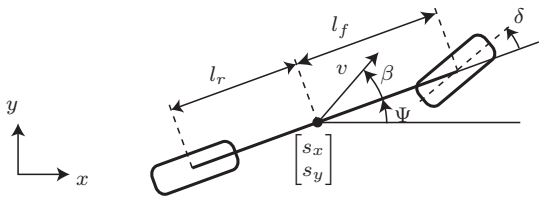


Fig. 2. Bicycle model.

TABLE I

BICYCLE MODEL PARAMETERS.

description	symbol	value	unit
vehicle mass	m	1093.3	kg
moment of inertia (yaw)	I_z	1791.6	kg m^2
distance from c.g. to front axle	l_f	1.1562	m
distance from c.g. to rear axle	l_r	1.4227	m
height of c.g. above ground	h	0.6137	m
cornering stiffness coefficient	$C_{S,i}$	23.196	1/rad
friction coefficient	μ	$[0.85, 1.05]$	—

B. High-Order Model

The high-order model is taken out of [21, Appendix A]. Unlike the bicycle model, this model considers the vertical load of all 4 wheels due to roll, pitch, and yaw, their individual spin and slip, and nonlinear tire dynamics. The multi-body dynamics is described by 3 masses: The unsprung mass and the sprung mass of the front and rear axles. The forces between these masses are described by the dynamics of the suspension and the tire model.

We considered all suspension forces in [21, Appendix A] originating from springs, dampers, and anti-roll bars. We do not consider flexibilities in the steering system, bump stops, and squat/lift forces caused by the suspension geometry. The vehicle parameters are taken from vehicle 14 in [21, Appendix E], which is a BMW 320i. Parameters of that vehicle applicable to the bicycle model are shown in Tab. I.

For the tire dynamics we use the PAC2002 Magic-Formula tire model, which is widely used in industry [22]. The combined lateral and longitudinal tire forces are computed from the slip angle, the camber angle, and the vertical tire force described in [21, Appendix A]. The tire parameters for all 4 wheels are taken from the example of a PAC2002 tire property file in [22, pp. 52–59]. The cornering stiffness coefficient $C_{S,i}$ required for the bicycle model is obtained from the nonlinear model by linearizing at zero slip angle.

Rewriting all equations as a state space model yields a model with 28 states. All states, including their initial values, are listed in Tab. II, where the pairs lf, rf, lr, rr indicate left/right and front/rear. Denoting the initial states of the bicycle model by a superscripted *b*, the initial states of the high-order model due to coordinate system transformations and zero initial tire slip assumption are: $\Psi_0 = -\Psi_0^b$, $\dot{\Psi}_0 = -\dot{\Psi}_0^b$, $\omega_0 = v_{x,0}/R$, $v_{x,0} = \cos(-\beta_0^b)v_0^b$, $v_{y,0} = \sin(-\beta_0^b)v_0^b$, $v_{yf,0} = v_{y,0} + l_f\dot{\Psi}_0$, $v_{yr,0} = v_{y,0} - l_r\dot{\Psi}_0$, $z_{i,0} = F_{zi,0}/(2K_{zt})$ ($i \in \{r, f\}$), $s_{x,0} = s_{x,0}^b$, $s_{y,0} = -s_{y,0}^b$, which use the effective tire radius R, the distances from the center of gravity to the front and rear axle l_f, l_r, the tire spring rate K_{zt}, and the vertical forces $F_{zf,0}$, $F_{zr,0}$ of the front and rear due to gravity.

TABLE II

INITIAL VALUES OF THE HIGH-ORDER MODEL.

sprung mass		unsprung mass		other	
name	init. val.	name	init. val.	name	init. val.
yaw ang.	Ψ_0	roll ang. (f)	0	wheel speed (lf)	ω_0
yaw rate	$\dot{\Psi}_0$	roll rate (f)	0	wheel speed (rf)	ω_0
roll angle	0	roll ang. (r)	0	wheel speed (lr)	ω_0
roll rate	0	roll rate (r)	0	wheel speed (rr)	ω_0
pitch ang.	0	y-vel. (f)	$v_{yf,0}$	pin joint diff. (f)	0
pitch rate	0	y-vel. (r)	$v_{yr,0}$	pin joint diff. (r)	0
x-velocity	$v_{x,0}$	z-pos. (f)	$z_{f,0}$	x-position	$s_{x,0}$
y-velocity	$v_{y,0}$	z-vel. (f)	0	y-position	$s_{y,0}$
z-position	0	z-pos. (r)	$z_{r,0}$		
z-velocity	0	z-vel. (r)	0		

C. Vehicle Controller

The vehicle controller is identical to the one proposed in [19], except that the controller parameter vector $k = [0.2,\ 2,\ 0.3,\ 1,\ 10]^T$ is slightly changed to smaller gains so that the control performance is still good for larger sensor noise. We use a positioning system that combines GPS data with inertial measurements to accurately measure the positions s_x, s_y, the yaw angle Ψ, the yaw rate $\dot{\Psi}$, and the velocity v. The corresponding sensor noise is combined in the vector $u = [u_x,\ u_y,\ u_\Psi,\ u_{\dot{\Psi}},\ u_v]^T \in [-1,1]0.08 \times [-1,1]0.08 \times [-1,1]0.2\pi/180 \times [-1,1]0.2\pi/180 \times [-1,1]0.08$. The reference values for the control are denoted by a subscripted *d* and are held constant for time intervals $[t_k, t_{k+1}]$, where $t_k = k\,r$, $k \in \mathbb{N}$ is the time step, and $r \in \mathbb{R}^+$ is the step size. These values are combined in $w = [s_{x,d},\ s_{y,d},\ \Psi_d,\ \dot{\Psi}_d,\ v_d]^T$. With the introduced variables, the control law for the steering angle δ and the vehicle acceleration a_x is

$$\begin{aligned}\delta =& k_1\Big(\cos(\Psi_d)(s_{y,d} - s_y - u_y) - \sin(\Psi_d)(s_{x,d} - s_x - u_x)\Big)\\ &+ k_2(\Psi_d - \Psi - u_\Psi) + k_3(\dot{\Psi}_d - \dot{\Psi} - u_{\dot{\Psi}}),\\ a_x =& k_4\Big(\cos(\Psi_d)(s_{x,d} - s_x - u_x) + \sin(\Psi_d)(s_{y,d} - s_y - u_y)\Big)\\ &+ k_5(v_d - v - u_v).\end{aligned}$$

Combining the vehicle controller with the low- and high-order model yields the corresponding controlled vehicle dynamics, denoted by $\dot{x} = f(x, w, u, p)$, where $p := \mu$ is the uncertain friction coefficient. The uncertain friction coefficient could also be modeled as part of the uncertain input u; however, uncertain road friction is treated differently in the subsequent reachability analysis, which is emphasized by this separate variable.

IV. Reachability Analysis

This section describes the basic principle for computing reachable sets subject to sensor noise and uncertain friction coefficient. We denote the solution of the vehicle dynamics $\dot{x} = f(x, w, u, p)$ for $x(0) = x_0$, $t \in [0, t_f]$, and trajectories $w(\cdot)$, $u(\cdot)$ by $\chi(t, x_0, w(\cdot), u(\cdot))$. Note that $w(\cdot)$ refers to a trajectory, where $w(t)$ refers to the value of the trajectory at time t. The exact reachable set for a given reference trajectory $w^*(\cdot)$ and a set of sensor noise values $\mathcal{U}$ is

$$\begin{aligned}\mathcal{R}^e([0,t_f]) = \Big\{&\chi(t, x_0, w(\cdot), u(\cdot), p(\cdot))\Big| t \in [0,t_f],\\ &x_0 \in \mathcal{R}(0), w(t) = w^*(t), u(t) \in \mathcal{U}, p(t) \in \mathcal{P}\Big\}.\end{aligned}$$

The uncertain input $u(t)$ is a piecewise continuous function, whereas the reference function $w(t)$ is constant within time intervals $\tau_k = [t_k, t_{k+1}]$ and updated at times t_k. Although $p(t)$ may vary continuously over time, we restrict this function to be piecewise constant, as for $w(t)$, in order to apply a more accurate reachability approach for uncertain parameters. Since for nonlinear systems, the reachable set cannot be computed exactly, we compute overapproximations $\mathcal{R}([0,t_f]) \supseteq \mathcal{R}^e([0,t_f])$.

The overapproximations in this work are obtained by linearizing the nonlinear dynamics $\dot{x} = f(x, w, u, p)$ so that techniques for linear systems can be applied as proposed in an earlier work [5]. In order to guarantee an overapproximative result, the linearization error is considered as an additional uncertain input, as presented in the next subsection.

A. Basic Procedure

For a concise notation of the linearization procedure, the state vector x and the input vector u are combined to form a new vector $z = [x^T,\ u^T]^T$. The reference trajectory is not included, since it is certain, and thus a linearization with respect to that vector is not required. The parameter p, however, is uncertain, but its influence is not linearized. Although this linearization is possible, it would result in much larger linearization errors in the vehicle dynamics (this has been tested, but results are not shown due to space limitations). In addition, the parameter influences the system dynamics by a multiplication with the state (unlike more complicated nonlinear operations), which can be elegantly expressed by uncertain state and input matrices for which efficient reachable set approaches exist for constant [23] and time-varying parameters [24].

Using a first-order Taylor expansion around the linearization point $[z^{*T},\ w^{*T}]^T$, the original differential equation of the i^{th} coordinate is enclosed by the differential inclusion

$$\begin{aligned}&\forall t \in \tau_k :\\ &\dot{x}_i \in \underbrace{f_i(z^*,\ w^*, p)}_{=c_i(p)} + \underbrace{\frac{\partial f_i(z, w^*, p)}{\partial z}\Big|_{z=z^*}(z - z^*)}_{=[A(p)(x-x^*)+B(p)(u-u^*)]_i} \oplus \mathcal{L}_i(\tau_k),\end{aligned} \tag{1}$$

where $\oplus$ denotes a Minkowski addition[1] and $\mathcal{L}$ is the set of Lagrange remainders

$$\begin{aligned}\mathcal{L}_i(\tau_k) = \Big\{&\frac{1}{2}(z - z^*)^T \frac{\partial^2 f_i(\xi, w^*, p)}{\partial z^2}(z - z^*)\Big|\\ &\xi \in \mathcal{R}(\tau_k) \times \mathcal{U}, p \in \mathcal{P}\Big\}.\end{aligned}$$

For more detailed information on the computation of $\mathcal{L}$, the interested reader is referred to [19], [25].

The linearization point $z^*(\tau_k) = [x^*(\tau_k), u^*]^T$ is chosen as $u^* = \texttt{center}(\mathcal{U})$ and $x^*(\tau_k) = \tilde{x}(t_k)$, where we linearize along the nominal trajectory $\tilde{x}(\cdot)$ obtained by a simulation starting in the center of $\mathcal{R}(0)$ subject to the input u^*.

For each time interval $[t_k, t_{k+1}]$ ($t_{k+1} = t_k + r$), the system is linearized, making it possible to apply the superposition principle for an input $v(t)$ so that $x(t_{k+1}) = x^h(t_{k+1}) + x^i(r)$, where

$$\begin{aligned}x^h(t_{k+1}) &= e^{Ar}x(t_k) \qquad \text{(initial state solution)}\\ x^i(r) &= \int_0^r e^{A(r-t)}v(t)\,\mathrm{d}t \qquad \text{(input solution).}\end{aligned} \tag{2}$$

This approach is also used when computing with sets, yielding the reachable set of a time interval $\mathcal{R}([t_k, t_{k+1}])$ in 3 steps (see Fig. 3). These steps involve the multiplication of sets ($\mathcal{A} \otimes \mathcal{B} := \{a\,b | a \in \mathcal{A}, b \in \mathcal{B}\}$):

[1] Given are sets in Euclidean space $\mathcal{A}$, $\mathcal{B}$: $\mathcal{A} \oplus \mathcal{B} = \{a+b | a \in \mathcal{A}, b \in \mathcal{B}\}$

1) Initial state solution:
   $\mathcal{R}^h(t_{k+1}) = \{e^{A(p)r} | p \in \mathcal{P}\} \otimes \mathcal{R}(t_k)$
2) Convex hull computation $\mathrm{CH}(\mathcal{R}(t_k), \mathcal{R}^h(t_{k+1}))$ for the approximation within $[t_k, t_{k+1}]$.
3) Addition of the reachable set of the input solution $\mathcal{R}^i(r)$ and an error term $\mathcal{D}$ (see [25]) making the result overapproximative:
   $\mathcal{R}(t_{k+1}) = \mathrm{CH}(\mathcal{R}(t_k), \mathcal{R}^h(t_{k+1})) \oplus \mathcal{R}^i(r) \oplus \mathcal{D}$.

When the set of uncertain inputs does not contain the origin, the above procedure has to be slightly modified [25]. For a tight overapproximation, it is important to account for parametric dependencies, which are addressed in the next subsection.

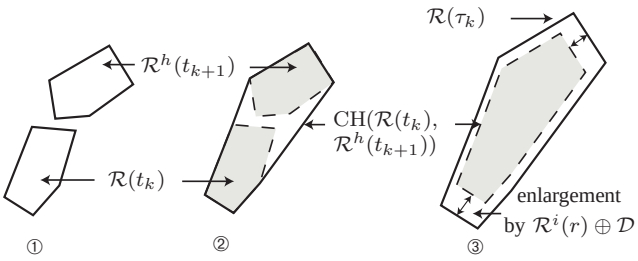


Fig. 3. Computation of the reachable set for a time interval $[t_k, t_{k+1}]$.

B. Considering Parametric Dependencies

In order to consider parametric dependencies, we take advantage of the special structure whereby the parameter p influences the system matrix $A(p)$, the input matrix $B(p)$, and the constant input $c(p)$. After normalizing the parameter $p \in \mathcal{P} = [\underline{p}, \overline{p}]$, such that it is mapped to values $\beta(p) = (2p - \overline{p} - \underline{p})/(\overline{p} - \underline{p}) \in [-1, 1]$, we can express the aforementioned variables of the vehicle model as $A(\beta) = C_A + \beta G_A$, $B(\beta) = C_B + \beta G_B$, $c(\beta) = c_c + \beta g_c$, where $C_A, G_A \in \mathbb{R}^{n \times n}$, $C_B, G_B \in \mathbb{R}^{n \times m}$, $c_c, g_c \in \mathbb{R}^{n \times 1}$, and $\beta \in [-1, 1]$. The matrices $A(\beta)$, $B(\beta)$ can be bounded by a matrix zonotope

$$\mathcal{H} = \left\{ C_H + \sum_{i=1}^{\kappa} \beta_i G_H^{(i)} \middle| \beta_i \in [-1, 1], C_H, G_H^{(i)} \in \mathbb{R}^{n \times n} \right\},$$

which is used later to bound the matrix exponential. The matrix C_H is referred to as the matrix center and $G_H^{(i)}$ is referred to as a matrix generator. The vector $c(\beta)$ can be analogously bounded by a zonotope which equals a matrix zonotope, except that the center matrix and the generator matrices are replaced by a center vector and generator vectors. Matrix zonotopes are a generalization of interval matrices and thus provide tighter bounds for matrix sets [23].

We overapproximately compute the exponential matrix $e^{A(\beta)r}$ using a finite Taylor series up to order η with a remainder term $\mathcal{E}(r)$, where $\mathcal{E}(r)$ is computed as in [23]:

$$e^{A(\beta)r} \in \sum_{i=0}^{\eta} \frac{(A(\beta)r)^i}{i!} \oplus \mathcal{E}(r). \tag{3}$$

When the system matrix is bounded by a general matrix zonotope, it is difficult to tightly bound the set of possible exponential matrices for large values of η. However, in the event of only one uncertain parameter, we propose a new and compact overapproximation:

Theorem 1 (Matrix Exponential Set (Single Parameter)): The set of matrix exponentials $\{e^{Ar} | A \in \mathcal{A}\}$, where $\mathcal{A} = \{C + \beta G | \beta \in [-1, 1], C_H, G_H^{(i)} \in \mathbb{R}^{n \times n}\}$ is a matrix zonotope with one generator matrix, can be tightly overapproximated by

$$\{e^{Ar} | A \in \mathcal{A}\} \subseteq \mathcal{K}(r) \oplus \mathcal{E}(r).$$

The matrix exponential remainder $\mathcal{E}(r)$ is computed as in [23] and $\mathcal{K}(r)$ is a matrix zonotope with the center

$$C_K = \sum_{l=0}^{\eta} D_l \frac{t^l}{l!} + \sum_{l=1}^{\lfloor \eta/2 \rfloor} G_K^{(2l)},$$

where $\lfloor \cdot \rfloor$ is the floor function, and the generators

$$G_K^{(l)} = \begin{cases} \sum_{i=l}^{\eta} F_i^{(l)} \frac{t^i}{i!}, & \text{for uneven } l \\ 0.5 \sum_{i=l}^{\eta} F_i^{(l)} \frac{t^i}{i!}, & \text{for even } l. \end{cases}$$

The matrix center and generators use auxiliary matrices, which are iteratively obtained for a given $i \in \mathbb{N}^+$:

$$\begin{aligned} & & D_i &= D_{i-1} C \\ & & F_i^{(1)} &= D_{i-1} G + F_{i-1}^{(1)} C \\ l = 2 \ldots (i-1): \quad & & F_i^{(l)} &= F_{i-1}^{(l-1)} G + F_{i-1}^{(l)} C \\ & & F_i^{(i)} &= F_{i-1}^{(i-1)} G, \end{aligned}$$

where $D_0 = I$ (I is the identity matrix), $F_0^{(1)} = 0$. □

Proof: We first show that $(C + \beta G)^i = D_i + \sum_{l=1}^{i} \beta^l F_i^{(l)}$ by induction:

$$\begin{aligned} (C + \beta G)^i &= \underbrace{(D_{i-1} + \sum_{l=1}^{i-1} \beta^l F_{i-1}^{(l)})}_{=(C+\beta G)^{i-1}} (C + \beta G) \\ &= \underbrace{D_{i-1} C}_{=D_i} + \underbrace{\beta (D_{i-1} G + F_{i-1}^{(1)} C)}_{=\beta F_i^{(1)}} \\ &\quad + \underbrace{\sum_{l=2}^{i-1} \beta^l (F_{i-1}^{(l-1)} G + F_{i-1}^{(l)} C)}_{\sum_{l=2}^{i-1} \beta^l F_i^{(l)}} + \underbrace{\beta^i F_{i-1}^{(i-1)} G}_{\beta^i F_i^{(i)}}. \end{aligned}$$

Using the Taylor terms of the matrix exponential in (3), we obtain

$$\begin{aligned} \sum_{i=0}^{\eta} \frac{(C + \beta G)^i r^i}{i!} &= \sum_{i=0}^{\eta} \frac{(D_i + \sum_{l=1}^{i} \beta^l F_i^{(l)}) r^i}{i!} \\ &= \underbrace{\sum_{i=0}^{\eta} D_i \frac{r^i}{i!}}_{=:\tilde{C}_K} + \sum_{l=1}^{\eta} \beta^l \underbrace{\sum_{i=l}^{\eta} F_i^{(l)} \frac{r^i}{i!}}_{=:\tilde{G}_K^{(l)}} \end{aligned}$$

When computing the set of matrices for $\beta \in [-1, 1]$, the even powers have the range $\beta^{2l} \in [0, 1]$, while the uneven powers are in the range $\beta^{2l+1} \in [-1, 1]$. Since matrix zonotopes

have generators with ranges $[-1, 1]$, the matrix generator values representing even powers can be multiplied by 0.5 and the center part can be added to the center of the matrix zonotope, tightening the result. Thus, $\tilde{C}_K$ becomes C_K and $\tilde{G}_K^{(l)}$ becomes $G_K^{(l)}$, as stated in the theorem. □

Besides the initial state solution, the constant input $c(\beta)$ is subject to the same parameter (see (1)). Neglecting this dependence, the straightforward computation of the set of input solutions (see (2)) would be performed as in [25] by $\int_0^r e^{\mathcal{A}t}\,\mathrm{d}t \otimes \mathcal{C}$, using $A(\beta) \in \mathcal{A}$, $c(\beta) \in \mathcal{C}$. However, this dependence is important, since the values of $c(\beta)$ might have a dominant effect depending on the linearization point. Using Theorem 1, this dependence can be taken care of by inserting (3) into (2), such that one obtains the partial input solution

$$x^{i,c}(r) = \sum_{i=0}^{\eta} A^i(\beta)c(\beta)\frac{r^{i+1}}{(i+1)!} \oplus \underbrace{\int_0^r \mathcal{E}(r)\,\mathrm{d}t}_{\subseteq \mathcal{E}(r)\,r,\ \text{see [25]}}\ c(\beta).$$

The terms $A^i(\beta)c(\beta) = (C + \beta G)^i(c_c + \beta g_c)$ can be computed similarly, as shown in the proof of Theorem 1, yielding a similar result enclosed by a zonotope. The second input $\mathcal{L}$ (see (1)) contributing to the set of input solutions $\mathcal{R}^i$ does not have these dependencies and is computed as in previous work [23].

We use zonotopes as a representation for reachable sets $\mathcal{R}([t_k, t_{k+1}])$, since they are efficient (complexity with respect to the system dimension n is $\mathcal{O}(n^3)$), numerically stable, and are the only known representation that can efficiently compute the set multiplication with matrix zonotopes.

V. Rapidly-Exploring Random Trees

In this section we describe the RRT algorithm designed to falsify the reachable set of the low-order model. In case of a violation, the reachable set computation of the low-order model can be adapted by enlarging the set of initial states and increasing an additive disturbance $\mathcal{V}$, such that $\dot{x} \in f^b(x, w, u, p) \oplus \mathcal{V}$.

RRTs have been used in [13] to underapproximate reachable sets. In contrast to [13], which tries to generally cover the state space, we consider the problem of covering the area around a reference trajectory uniformly over time. Although we use the same basic technique, we make a modification to generate the same number of samples for each time interval, see Fig. 4:

1) Initialize the discrete set of states for the next time interval as $\mathcal{X}(\tau_{k+1}) = \emptyset$.
2) Generate a sample x_s from the state space.
3) Find the nearest state x_n according to a distance measure ρ so that $x_n = \arg\min(\rho(x_s, x^{(i)}))$, where $x^{(i)} \in \mathcal{X}(\tau_k)$.
4) Obtain the input u which drives x_n to the new state x_{add} closest to x_s.
5) Add x_{add} to the set of states for the next time interval $\mathcal{X}(\tau_{k+1})$.
6) Repeat steps 2-5 for a predefined number of samples, then go to the next time interval and start with step 1.

When initializing $\mathcal{X}(\tau_{k+1}) = \mathcal{X}(\tau_k)$, one obtains the approach in [13].

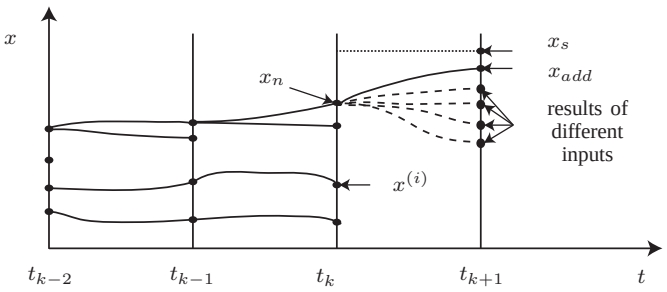


Fig. 4. RRT concept for trajectory tracking.

Another difference is that we sample the space $\mathcal{X}_{rel}$ relative to the nominal trajectory $x^*(t)$ used for linearization, which is chosen as a multidimensional rectangle centered at the origin with edge lengths $l_\beta = 0.4$, $l_\Psi = 0.4$, $l_{\dot{\Psi}} = 4$, $l_v = 2$, $l_{s_x} = 4$, $l_{s_y} = 4$. The edge lengths are chosen such that the reachable sets for all time intervals are enclosed. Best results have been obtained by first choosing x_s as the vertices of $\mathcal{X}_{rel}$ and then uniformly sampling $\mathcal{X}_{rel}$. Combined deterministic and stochastic sampling has been reported as advantageous for sampling-based planning, too [26].

The distance measure is simply chosen as $\rho(x_s, x^{(i)}) = \|N(x_s - x^{(i)})\|_2$ with the normalization matrix $N = \mathrm{diag}(\frac{1}{l_\beta}, \frac{1}{l_\Psi}, \frac{1}{l_{\dot{\Psi}}}, \frac{1}{l_v}, \frac{1}{l_{s_x}}, \frac{1}{l_{s_y}})$. The normalization is important since otherwise the coordinates of high numerical value would be preferred.

The optimal u that drives x_n to x_s, i.e., minimizes $\rho(x_{add}, x_s)$, is simply chosen by testing all vertices of the set of possible inputs $\mathcal{U}$. For the RRT computation, we add the uncertain friction μ as an additional uncertain input to the other inputs u (see Sec. III-C) in order to simplify the notation. Thus, we obtain a manageable set of 64 inputs. In [13], the system dynamics is linearized and linear programming is used to obtain the optimal input. However, for the vehicle dynamics, the input matrix does not have full rank, so that this technique cannot be used. Other optimization techniques are too time consuming due to the high order of the model (28 states). In addition, sampling over all corner cases of u is always numerically stable.

VI. Numerical Experiments

We demonstrate the proposed techniques for reachable sets and RRTs on three standard maneuvers: Evasive maneuver (lane change and braking), moose test (double lane change), cornering (braking towards the apex, followed by accelerating). We first give a detailed description of the maneuvers and then present numerical results.

A. Tested Maneuvers

The capabilities of a vehicle can be roughly described by Kamm's circle, which shows the border of possible combined lateral acceleration a_x and longitudinal acceleration a_y, combined in $a = [a_x, a_y]^T$. We design the reference

trajectories by providing the direction Φ and the absolute value $a_{abs} = \|a\|_2$ of the acceleration in vehicle-fixed coordinates, ensuring that the accelerations are within Kamm's circle. In addition, we restrict the jerk $\sigma = \|\dot{a}\|_2$ of the vehicle, since the steering, acceleration and braking cannot be changed instantaneously.

We encode the maneuvers by the direction Φ, the absolute value a_{abs}, and their duration. When the acceleration changes, the acceleration rate $\sigma_{max} = 50$ [m/s^3] towards the new acceleration a is applied, resulting in a trajectory for a_x and a_y. The reference values of the maneuvers are obtained from the acceleration as:

$$v_d(t) = \int_0^t a_x(\tau)\mathrm{d}\tau,\ \dot{\Psi}_d(t) = \frac{a_y(t)}{v_d(t)},\ \Psi_d(t) = \int_0^t \dot{\Psi}_d(\tau)\mathrm{d}\tau,$$
$$s_{x,d}(t) = \int_0^t \cos(\Psi_d(\tau))v_d(\tau)\mathrm{d}\tau,$$
$$s_{y,d}(t) = \int_0^t \sin(\Psi_d(\tau))v_d(\tau)\mathrm{d}\tau.$$

In Tab. III we summarize the tested maneuvers. Note that all maneuvers are highly dynamic, i.e., relatively close to the maximum possible tire forces.

TABLE III
SPECIFICATION OF TESTED MANEUVERS.

a_{abs} [m/s^2]	Φ [rad]	duration [s]
evasive maneuver		
$[0, 6, 6, 0]$	$[0, 0.75, -0.75, -1]\Pi$	$[0.4, 0.75, 0.63, 0.65]$
moose test		
$[0, 8, 8, 0, 8, 8, 0]$	$0.5[0, 1, -1, 0, -1, 1, 0]\Pi$	$[0.4, 0.84, 1, 1, 0.84, 1, 0.4]$
cornering		
$[0, 6, 4.8, 0]$	$[0, 0.7, 0.3, 0]\Pi$	$[0.4, 1, 1, 0.4]$

B. Results

For each of the presented maneuvers we compute a RRT and compare the results with reachability analysis. We compute two different reachable sets: The first one considers the uncertain friction as presented in Sec. IV; the second one is computed identically, except that the specific friction $\mu = 0.9$ is considered instead. For both cases, we present the added disturbance for which all RRT states of all maneuvers and time intervals are contained in the corresponding reachable sets $\mathcal{R}([t_k, t_{k+1}])$. In order to efficiently check enclosure of the RRT states, we modify the reachable sets in a post-processing procedure by replacing them by their enclosing boxes for each time interval.

When considering uncertain friction, one only has to add disturbance to the longitudinal acceleration ($\mathcal{V} = 0 \times 0 \times 0 \times [0, -1] \times 0 \times 0$) in order to enclose all RRT states. The reason for the required disturbance is that in the high-order model, the vehicle slows down for large slip angles due to tire friction – an effect that is not modeled by the bicycle model. In the other case, when uncertain friction is not directly modeled, the additive disturbance has to be enlarged by $\mathcal{V} = [-0.15, 0.15] \times 0 \times 0 \times [0, -1] \times 0 \times 0$ in order to account for the uncertainty in the slip angle β due to uncertain friction. These numbers have been obtained by successively enlarging the uncertain input of state x_i when the reachable interval of that state has been violated. This can be easily automated given an increment of the enlargement for each coordinate.

In addition to the additive disturbance, we enlarged the initial set of states, which is chosen as $\mathcal{R}(0) = [-0.02, 0.02] \times [-0.05, 0.05] \times [-0.05, 0.05] \times [14.8, 15.2] \times [-0.2, 0.2] \times [-0.2, 0.2]$, by 5% in each direction for the reachable set computations. The time step for updating the reference trajectory is $r = 0.01$ s.

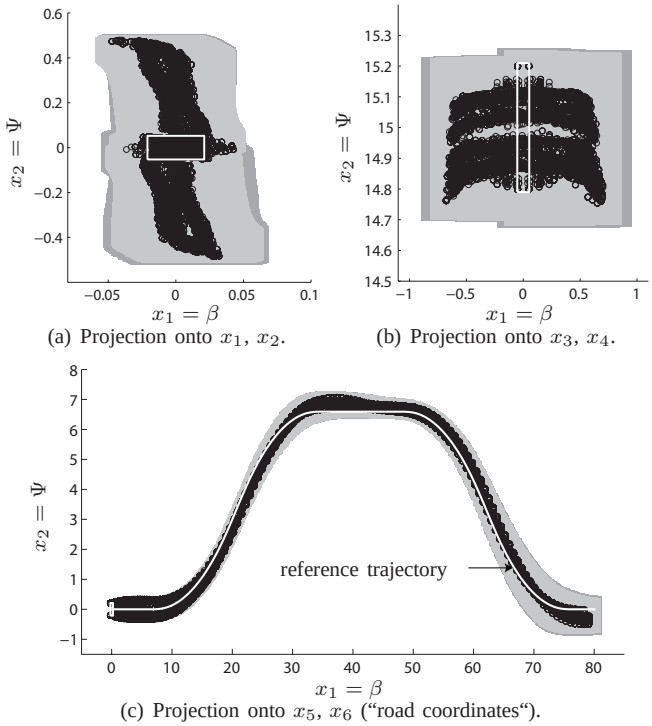
The reachable set for the indirect modeling of uncertain friction is slightly tighter than for the direct modeling, since in the latter case uncertain friction is considered in an overapproximative way. This is illustrated for different projections in Fig. 5 for the moose test; other plots are neglected due to space restrictions, which show similar results. However, the reachable set computations without uncertain parameters are much more efficient. The computation times for a prototype implementation in MATLAB on an Intel i7 Processor with 1.6 GHz and 6 GB memory are shown in Tab. IV. For comparison we also added the computation time of a single simulation run of the high-order model using the standard Runge-Kutta solver (ode45) in MATLAB. The computation times are obtained under the assumption that the linearization is done in a parallel process together with the reachability computation of the linearized system. One can observe that even for the MATLAB implementation, the certain-friction case is faster than the execution time of the maneuvers. Future implementations in C++ should improve these numbers. Note that the reachable set computations of uncertain friction would have become numerically unstable (reachable set grows excessively fast) when the parameter dependence discussed in Sec. IV-B had not been considered.

TABLE IV
COMPUTATIONAL TIMES IN SECONDS.

maneuver	maneuver time [s]	comp. time (no unc. par.)	comp. time (unc. par.)	sim. time (high-or.)
ev. maneu.	2.43	1.30	3.97	3.40
moose test	5.48	2.97	8.93	7.14
cornering	2.8	1.53	4.86	4.18

VII. CONCLUSIONS

We have presented two versions of reachable set computations: One which directly considers uncertain friction, another one which considers it indirectly by adding disturbance. It is shown that in both cases, the sets contain all states of a high-order vehicle model generated by a RRT when adding disturbance and slightly enlarging the set of initial states. The computation with a fixed friction parameter is considerably faster and thus preferred, unless the friction coefficient has high uncertainty. When using a fixed friction parameter, the MATLAB prototype was already faster than the execution time of the maneuver. The approach also shows that the simulation of the high-order model takes an amount



(a) Projection onto x_1, x_2. (b) Projection onto x_3, x_4.

(c) Projection onto x_5, x_6 ("road coordinates").

Fig. 5. Reachable set for the moose test (double lane change maneuver). The white box shows the set of initial states, black circles show states from the RRT generation every $5r = 0.05$ [s], the gray regions show the reachable sets: with uncertain parameters (dark gray), without uncertain parameters (light gray).

of time similar to that of the reachable set computation, which encloses the high-order behavior.

In the future, we plan to implement the presented reachability algorithm on a real vehicle and try to falsify the reachable sets by real world driving experiments. The RRT approach is a required step for tuning additive disturbance and gaining enough insight before conducting time-consuming experiments.

ACKNOWLEDGMENTS

The authors gratefully acknowledge partial financial support by the NSF Cyberphysical Systems Program, Award CNS1035813, and by the NSF Expeditions in Computing Program, Award CCF0926181.

REFERENCES

[1] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler, "Recent progress in continuous and hybrid reachability analysis," in *Proc. of the 2006 IEEE Conference on Computer Aided Control Systems Design*, 2006, pp. 1582–1587.

[2] A. Puri, P. Varaiya, and V. Borkar, "ϵ-approximation of differential inclusions," in *Proc. of the 34th IEEE Conference on Decision and Control*, 1995, pp. 2892 – 2897.

[3] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *Hybrid Systems: Control and Computation*, 2003, pp. 20–35.

[4] Z. Han and B. H. Krogh, "Reachability analysis of nonlinear systems using trajectory piecewise linearized models," in *Proc. of the American Control Conference*, 2006, pp. 1505–1510.

[5] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proc. of the 47th IEEE Conference on Decision and Control*, 2008, pp. 4042–4048.

[6] T. Dang, O. Maler, and R. Testylier, "Accurate hybridization of nonlinear systems," in *Hybrid Systems: Computation and Control*, 2010, pp. 11–19.

[7] A. Chutinan and B. H. Krogh, "Computational techniques for hybrid system verification," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 64–75, 2003.

[8] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, "Computational techniques for the verification and control of hybrid systems," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 986–1001, 2003.

[9] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, pp. 947–957, 2005.

[10] A. Girard, C. Le Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid Systems: Computation and Control*, ser. LNCS 3927. Springer, 2006, pp. 257–271.

[11] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *Proc. of the IEEE International Conference on Robotics and Automation*, 1999, pp. 473 – 479.

[12] M. S. Branicky, M. M. Curtiss, J. A. Levine, and S. B. Morgan, "RRTs for nonlinear, discrete, and hybrid planning and control," in *Proc. of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 657–663.

[13] A. Bhatia and E. Frazzoli, "Incremental search methods for reachability analysis of continuous and hybrid systems," in *Hybrid Systems: Computation and Control*, ser. LNCS 2993. Springer, 2004, pp. 142–156.

[14] M. S. Branicky, M. M. Curtiss, J. Levine, and S. Morgan, "Sampling-based planning, control, and verification of hybrid systems," *IEE Proceedings – Control Theory and Applications*, vol. 153, no. 5, pp. 575 – 590, 2006.

[15] T. Dang and T. Nahhal, "Coverage-guided test generation for continuous and hybrid systems," *Formal Methods in System Design*, vol. 34, no. 2, pp. 183 – 213, 2009.

[16] A. Donzé and O. Maler, "Systematic simulations using sensitivity analysis." in *Hybrid Systems: Computation and Control*, ser. LNCS 4416. Springer, 2007, pp. 174–189.

[17] A. Girard and G. J. Pappas, "Verification using simulation," in *Hybrid Systems: Computation and Control*, ser. LNCS 3927. Springer, 2006, pp. 272–286.

[18] T. Nghiem, S. Sankaranarayanan, G. Fainekos, F. Ivančić, A.Gupta, and G. J. Pappas, "Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems," in *Hybrid Systems: Computation and Control*, 2010, pp. 211–220.

[19] M. Althoff and J.-M. Dolan, "Set-based computation of vehicle behaviors for the online verification of autonomous vehicles," in *Proc. of the 14th IEEE Conference on Intelligent Transportation Systems*, 2011.

[20] R. Rajamani, *Vehicle Dynamics and Control*, F. F. Ling and E. F. Gloyna, Eds. Springer, 2006.

[21] R. W. Allen, H. T. Szostak, D. H. Klyde, T. J. Rosenthal, and K. J. Owens, "Vehicle dynamic stability and rollover," U.S. Department of Transportation, Final Report DOT HS 807 956, 1992.

[22] *Adams/Tire help*, MSC Software, 2 MacArthur Place, Santa Ana, CA 92707, April 2011, documentation ID: DOC9805. [Online]. Available: http://simcompanion.mscsoftware.com/infocenter

[23] M. Althoff, B. H. Krogh, and O. Stursberg, *Modeling, Design, and Simulation of Systems with Uncertainties*. Springer, 2011, ch. Analyzing Reachability of Linear Dynamic Systems with Parametric Uncertainties, pp. 69–94.

[24] M. Althoff, C. Le Guernic, and B. H. Krogh, "Reachable set computation for uncertain time-varying linear systems," in *Hybrid Systems: Computation and Control*, 2011.

[25] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Dissertation, Technische Universität München, 2010, http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4.

[26] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Intl. Journal of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.