Trajectory Optimization for Falsification: A Case Study of Vehicle Rollover Test Generation Based on Black-box Models [⋆]

Sunbochen Tang, Nan Li, Ilya Kolmanovsky, and Anouck Girard [*]

[*] *Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48019 USA*
E-mail: {tangsun, nanli, ilya, anouck}@umich.edu

Abstract: In this paper, we consider optimization of trajectories for automotive vehicle rollover testing. In particular, *worst-case* trajectories that are most likely to cause rollover accidents are determined through trajectory optimization. Our approach combines online local-model identification and gradient-based input update, and can be applied to *black-box* type models, e.g., a high-fidelity vehicle dynamics model given as a simulation code and not as an explicit set of equations. With our approach, a library of worst-case trajectories corresponding to different operating conditions (e.g., vehicle mass, road surface conditions, etc.) can be constructed and subsequently used in hardware tests.

Keywords: trajectory optimization, data-driven methods, automotive applications, verification and validation, design of experiments

1. INTRODUCTION

Achieving safer transportation has been a major goal in the automotive industry. Compared to other types of traffic accidents, rollover accidents have higher fatality rates (NHTSA, 2018). Extensive efforts have been pursued in both academia and industry to address vehicle rollovers (Rajamani, 2011). Various rollover prevention mechanisms have been proposed based on, e.g., differential braking in Chen and Peng (2001); Gáspár et al. (2005); Lu et al. (2007); Li et al. (2016) or steering intervention in Carlson and Gerdes (2003); Solmaz et al. (2007); Bencatel et al. (2017); Liu et al. (2019). To evaluate vehicle roll stability and rollover resistance, as well as to assess performance of these rollover prevention mechanisms, standardized tests including static tests, e.g., based on measurement of the static stability factor (NHTSA, 2017), and dynamic tests such as fishhook (NHTSA, 2004) and sine-with-dwell (NHTSA, 2007) maneuvers have been established.

However, the standardized tests may not correspond to the *worst case* for all vehicles or over all operating conditions of the same vehicle. Here, *worst case* designates operation scenarios or maneuver trajectories that are most likely to cause the vehicle to have a rollover accident.

The procedure of designing/identifying a worst-case trajectory for a given system such that a given system specification is violated by that trajectory is generally called *falsification* (since the specification is disproved/falsified), and such a trajectory is usually called a *falsifying trajectory.* As it is typically unknown *a priori* whether such a trajectory exists or not, procedures for designing worst-case trajectories and for identifying (the existence of) falsifying trajectories are sometimes interpreted interchangeably, both called *falsification.*

In the literature, several approaches have been proposed to address the falsification problem of control systems, based on, e.g., optimal control theory in Ma and Peng (1999a,b), Yaghoubi and Fainekos (2019), or sampling-based techniques in Cheng and Kumar (2008), Nghiem et al. (2010) and in the temporal logic falsification tool S-TaLiRo (Annpureddy et al., 2011). In our previous work (Li et al., 2017, 2018), we have also proposed an approach to falsifying control systems by exploiting optimal control theory. In particular, our approach is applicable to systems that do not have explicit models. By integrating online local-model identification and gradient-based input update into an iterative algorithm, our approach handles the system by treating it as a *black-box* generating input-output responses. For continuous-time systems represented by ordinary differential equations, this approach is presented in Li et al. (2017), and an extended version handling systems with input and/or state time-delays has appeared in Li et al. (2018).

The contributions of this paper are as follows: 1) In this paper, we describe a discrete-time variant of the approach in Li et al. (2017, 2018). On the one hand, discrete-time formulation translates more naturally into the computational implementation of the approach. On the other hand, we have found through simulation experiments that the discrete-time version of our approach has greater numerical stability than its continuous-time counterpart. 2) This paper considers a new case study of vehicle rollover test generation. We first illustrate the use of our approach to determine two distinct worst-case trajectories for a given vehicle model operating in two different road conditions. We then discuss the application of our approach to constructing a worst-case trajectory library corresponding to a range of operating conditions, which can inform future hardware tests. 3) To account for the fact that the dynamics during a vehicle rollover event experiences discontinuous changes, which may be modeled as event-triggered mode switches (Yoon et al., 2007), we have modified the iterative algorithm of Li et al. (2017, 2018) so that it is applicable to such mode-switching hybrid systems.

[⋆] This work has been supported by the National Science Foundation Award CNS 1544844.

Furthermore, compared to previous optimal control-based falsification approaches, such as the one in Ma and Peng (1999a,b), our approach has the following differences: 1) Unlike restricting the cost function to be quadratic as in Ma and Peng (1999a,b), our gradient-based input update can handle a broader class of cost functions and thereby a wider range of falsification objectives. Examples of non-quadratic cost functions treated by our approach have been reported in Li et al. (2017, 2018). 2) When applied to system models that do not have explicit equation forms, the approach of Ma and Peng (1999a,b) uses a standard numerical differentiation method to estimate the derivative information, which is not uncommon in the treatment of optimal control problems based on models (Kasac et al., 2010). This requires the algorithm to be able to artificially manipulate the state value of the system at every time instant. In contrast, our approach uses collected input-output trajectory data to estimate needed gradient information, treating the system as a black-box. This makes our approach applicable to industry-level models that are sealed and not inspectable from outside. 3) Although the application to vehicle rollover test is considered both in Ma and Peng (1999a,b) and in this paper, in this paper we explicitly account for the discontinuous changes of vehicle dynamics during rollover accidents through modeling those as event-triggered mode switches, which is not pursued in Ma and Peng (1999a,b).

2. PROBLEM FORMULATION

We consider systems that can be modeled by the following discrete-time model,

$$x_{k+1} = f(x_k, u_k, k) + d_k, \tag{1}$$

where k denotes the discrete-time instant, $x_k \in \mathbb{R}^n$ denotes the system state, $u_k \in \mathbb{R}^{n_u}$ denotes a controlled input, and $d_k \in \mathbb{R}^n$ denotes an uncontrolled input. The d_k may represent disturbance inputs to the system and/or unmodeled dynamics, which is assumed to be sufficiently small.

We are interested in optimizing the input signal u_k to minimize a cost function in the form of

$$J = \sum_{k=0}^{N-1} L(x_k, u_k, k) + K(x_N), \tag{2}$$

where $L(\cdot,\cdot,k)$ and $K(\cdot)$ are continuously differentiable functions. Although in general (2) may represent various optimization objectives, in this paper we consider it as a measurement of the system performance in terms of satisfying a given specification (Fainekos and Pappas, 2009) – smaller values of (2) represent worse specification satisfaction. Therefore, minimizing (2) represents the goal of designing/identifying an input trajectory u_k to falsify the specification. In particular, we consider the case where the model (1) is a black-box with measurable input-output responses. For instance, it may represent a simulation code where an explicit form of the model is unknown while simulation trajectories can be generated. For this, we make the following assumptions: 1) the $f(\cdot,\cdot,k)$ are unknown continuously differentiable functions of x and u; and 2) the initial condition x_0 can be reset and the states x_k can be measured. We remark that the approach proposed in what follows does not rely on any particular parametrization of f and the need to *globally* identify f from collected data, which could, in general, be difficult.

3. ITERATIVE INPUT UPDATE BASED ON DATA-DRIVEN GRADIENT ESTIMATION

To begin with, we define the Lagrangian function associated with the system (1) and the cost function (2),

$$\mathcal{J} = K(x_N) + \lambda_0^T x_{\text{init}} - \lambda_N^T x_N + \sum_{k=0}^{N-1} \big[\mathcal{H}_k + \lambda_{k+1}^T d_k - \lambda_k^T x_k\big], \tag{3}$$

where x_{init} denotes the given nominal initial condition, $\lambda_k \in \mathbb{R}^n$ are the Lagrange multipliers, and $\mathcal{H}_k$ are the Hamiltonian functions,

$$\mathcal{H}_k = L(x_k, u_k, k) + \lambda_{k+1}^T f(x_k, u_k, k). \tag{4}$$

Now assume a nominal trajectory (x_k^0, u_k^0, d_k^0) is given. In the vicinity of (x_k^0, u_k^0, d_k^0), the variation of $\mathcal{J}$ with respect to variations in (x_k, u_k, d_k) can be estimated as

$$\delta\mathcal{J} = \Big(\frac{\partial K}{\partial x_N}\Big|_0 - \lambda_N^T\Big)\delta x_N + \sum_{k=0}^{N-1} \Big[\Big(\frac{\partial \mathcal{H}_k}{\partial x_k}\Big|_0 - \lambda_k^T\Big)\delta x_k + \frac{\partial \mathcal{H}_k}{\partial u_k}\Big|_0 \delta u_k + \lambda_{k+1}^T \delta d_k\Big] + \mathcal{O}^{\mathcal{J}}, \tag{5}$$

where $|_0$ represents evaluations at the nominal trajectory, $\mathcal{O}^{\mathcal{J}}$ denotes higher-order-terms in the Taylor expansion (5), and

$$\frac{\partial \mathcal{H}_k}{\partial x_k}\Big|_0 = \frac{\partial L_k}{\partial x_k}\Big|_0 + \lambda_{k+1}^T \frac{\partial f_k}{\partial x_k}\Big|_0, \tag{6}$$

$$\frac{\partial \mathcal{H}_k}{\partial u_k}\Big|_0 = \frac{\partial L_k}{\partial u_k}\Big|_0 + \lambda_{k+1}^T \frac{\partial f_k}{\partial u_k}\Big|_0, \tag{7}$$

in which $L_k(\cdot,\cdot) = L(\cdot,\cdot,k)$ and $f_k(\cdot,\cdot) = f(\cdot,\cdot,k)$.

Consider Lagrange multipliers determined as follows:

$$\lambda_N^T = \frac{\partial K}{\partial x_N}\Big|_0, \qquad \lambda_k^T = \frac{\partial \mathcal{H}_k}{\partial x_k}\Big|_0, \tag{8}$$

resulting in

$$\delta\mathcal{J} = \sum_{k=0}^{N-1} \Big[\frac{\partial \mathcal{H}_k}{\partial u_k}\Big|_0 \delta u_k + \lambda_{k+1}^T \delta d_k\Big] + \mathcal{O}^{\mathcal{J}}. \tag{9}$$

Then, consider

$$\delta u_k = -\eta_k \frac{\partial \mathcal{H}_k}{\partial u_k}\Big|_0^T, \tag{10}$$

with $\eta_k > 0$, and obtain

$$\delta\mathcal{J} = -\sum_{k=0}^{N-1} \eta_k \Big\|\frac{\partial \mathcal{H}_k}{\partial u_k}\Big|_0\Big\|_2^2 + \sum_{k=0}^{N-1} \lambda_{k+1}^T \delta d_k + \mathcal{O}^{\mathcal{J}}, \tag{11}$$

where $\|\cdot\|_2$ represents the 2-norm.

Suppose $\frac{\partial \mathcal{H}_k}{\partial u_k}\big|_0$ are not all zero and δd_k and $\mathcal{O}^{\mathcal{J}}$ are sufficiently small, then $\delta J = \delta\mathcal{J} < 0$, i.e., the cost (2) can be decreased via the update $u_k^0 \to u_k^0 + \delta u_k$.

The above input update procedure is not yet applicable, because the derivatives $\frac{\partial f_k}{\partial x_k}\big|_0$ and $\frac{\partial f_k}{\partial u_k}\big|_0$ involved in steps (8) and (10) are not known, since f is an unknown function. Note also that it may not be easy to globally estimate the nonlinear function f. Therefore, we pursue local estimation of f, i.e., estimating $\frac{\partial f_k}{\partial x_k}\big|_0$ and $\frac{\partial f_k}{\partial u_k}\big|_0$ instead. For this, linearizing the dynamics (1) in the vicinity of the nominal trajectory (x_k^0, u_k^0, d_k^0) yields

$$\delta x_{k+1} = \frac{\partial f_k}{\partial x_k}\Big|_0 \delta x_k + \frac{\partial f_k}{\partial u_k}\Big|_0 \delta u_k + \delta d_k + \mathcal{O}, \tag{12}$$

where $\delta x_k = x_k - x_k^0$, $\delta u_k = u_k - u_k^0$, and $\delta d_k = d_k - d_k^0$.

Maintaining $u_k = u_k^0$ and perturbing x_k through a perturbation to the initial condition x_0, we obtain

$$\delta\hat{x}_{k+1} = \frac{\partial f_k}{\partial x_k}\Big|_0 \delta\hat{x}_k + \delta\hat{d}_k + \mathcal{O}^x. \tag{13}$$

We now collect m_x such perturbed trajectories,

$$\underbrace{\left[\delta\hat{x}_{k+1}^{1} \cdots \delta\hat{x}_{k+1}^{m_x}\right]}_{\hat{X}_{k+1}^{x}} = \frac{\partial f_k}{\partial x_k}\bigg|_0 \underbrace{\left[\delta\hat{x}_{k}^{1} \cdots \delta\hat{x}_{k}^{m_x}\right]}_{\hat{X}_{k}^{x}} + \underbrace{\left[\delta\hat{d}_{k}^{1} \cdots \delta\hat{d}_{k}^{m_x}\right]}_{\hat{D}_{k}^{x}} + \mathcal{O}^x, \qquad (14)$$

with $m_x \geq n$ such that the matrix $\hat{X}_k^x$ is full rank. Then, we estimate $\frac{\partial f_k}{\partial x_k}\big|_0$ as

$$\widetilde{\frac{\partial f_k}{\partial x_k}}\bigg|_0 = \hat{X}_{k+1}^{x}(\hat{X}_k^x)^+, \qquad (15)$$

where $(\hat{X}_k^x)^+$ represents the Moore-Penrose inverse of $\hat{X}_k^x$,

$$(\hat{X}_k^x)^+ = (\hat{X}_k^x)^T\big(\hat{X}_k^x(\hat{X}_k^x)^T\big)^{-1}. \qquad (16)$$

After the estimate $\widetilde{\frac{\partial f_k}{\partial x_k}}\big|_0$ has been obtained, we pursue estimation of $\frac{\partial f_k}{\partial u_k}\big|_0$. For this, perturb u_k and collect m_u perturbed trajectories

$$\underbrace{\left[\delta\hat{x}_{k+1}^{1} \cdots \delta\hat{x}_{k+1}^{m_u}\right]}_{\hat{X}_{k+1}^{u}} = \frac{\partial f_k}{\partial x_k}\bigg|_0 \underbrace{\left[\delta\hat{x}_{k}^{1} \cdots \delta\hat{x}_{k}^{m_u}\right]}_{\hat{X}_{k}^{u}} \qquad (17)$$
$$+ \frac{\partial f_k}{\partial u_k}\bigg|_0 \underbrace{\left[\delta\hat{u}_{k}^{1} \cdots \delta\hat{u}_{k}^{m_u}\right]}_{\hat{U}_{k}^{u}} + \underbrace{\left[\delta\hat{d}_{k}^{1} \cdots \delta\hat{d}_{k}^{m_u}\right]}_{\hat{D}_{k}^{u}} + \mathcal{O}^u,$$

with $m_u \geq n_u$ such that the matrix $\hat{U}_k^u$ is full rank. Then, we estimate $\frac{\partial f_k}{\partial u_k}\big|_0$ as

$$\begin{aligned}\widetilde{\frac{\partial f_k}{\partial u_k}}\bigg|_0 &= \Big(\hat{X}_{k+1}^u - \widetilde{\frac{\partial f_k}{\partial x_k}}\bigg|_0 \hat{X}_k^u\Big)(\hat{U}_k^u)^+ \\ &= \big(\hat{X}_{k+1}^u - \hat{X}_{k+1}^x(\hat{X}_k^x)^+\hat{X}_k^u\big)(\hat{U}_k^u)^+.\end{aligned} \qquad (18)$$

After both the estimates $\widetilde{\frac{\partial f_k}{\partial x_k}}\big|_0$ and $\widetilde{\frac{\partial f_k}{\partial u_k}}\big|_0$ have been obtained, we update the input u_k according to (10) with the $\frac{\partial f_k}{\partial x_k}\big|_0$ and $\frac{\partial f_k}{\partial u_k}\big|_0$ in (6) and (7) replaced with $\widetilde{\frac{\partial f_k}{\partial x_k}}\big|_0$ and $\widetilde{\frac{\partial f_k}{\partial u_k}}\big|_0$, respectively. Then, the variation of $\mathcal{J}$ can be written as

$$\begin{aligned}\delta\mathcal{J} = &\sum_{k=0}^{N-1}\Bigg(\lambda_{k+1}^T\Big[\Big(\frac{\partial f_k}{\partial x_k}\bigg|_0 - \widetilde{\frac{\partial f_k}{\partial x_k}}\bigg|_0\Big)\delta x_k + \delta d_k\Big] \\ &- \eta_k\Big(\frac{\partial L_k}{\partial u_k}\bigg|_0 + \lambda_{k+1}^T\frac{\partial f_k}{\partial u_k}\bigg|_0\Big)\Big(\frac{\partial L_k}{\partial u_k}\bigg|_0 + \lambda_{k+1}^T\widetilde{\frac{\partial f_k}{\partial u_k}}\bigg|_0\Big)^T\Bigg) + \mathcal{O}^{\mathcal{J}} \\ = &\sum_{k=0}^{N-1}\big(-P_k + \lambda_{k+1}^T Q_k\big) + \mathcal{O}^{\mathcal{J}},\end{aligned} \qquad (19)$$

where

$$\begin{aligned}P_k &= \eta_k\Big(\frac{\partial L_k}{\partial u_k}\bigg|_0 + \lambda_{k+1}^T\widetilde{\frac{\partial f_k}{\partial u_k}}\bigg|_0\Big)\Big(\frac{\partial L_k}{\partial u_k}\bigg|_0 + \lambda_{k+1}^T\widetilde{\frac{\partial f_k}{\partial u_k}}\bigg|_0\Big)^T \\ Q_k &= \delta d_k + \Big(\frac{\partial f_k}{\partial x_k}\bigg|_0 - \widetilde{\frac{\partial f_k}{\partial x_k}}\bigg|_0\Big)\delta x_k \\ &\quad - \eta_k\Big(\frac{\partial f_k}{\partial u_k}\bigg|_0 - \widetilde{\frac{\partial f_k}{\partial u_k}}\bigg|_0\Big)\Big(\frac{\partial L_k}{\partial u_k}\bigg|_0 + \lambda_{k+1}^T\widetilde{\frac{\partial f_k}{\partial u_k}}\bigg|_0\Big)^T.\end{aligned}$$

Suppose $\frac{\partial L_k}{\partial u_k}\big|_0 + \lambda_{k+1}^T\widetilde{\frac{\partial f_k}{\partial u_k}}\big|_0$ are not all zero. Then it can be seen from the above expressions that for sufficiently small δd_k, $\mathcal{O}^x$, $\mathcal{O}^u$ and $\mathcal{O}^{\mathcal{J}}$, $\delta\mathcal{J} < 0$ can be achieved, since $\delta\mathcal{J} \to -\sum_{k=0}^{N-1} P_k$ as $\delta d_k, \mathcal{O}^x, \mathcal{O}^u, \mathcal{O}^{\mathcal{J}} \to 0$.

4. FALSIFICATION ALGORITHM

In some systems, dynamics may change discontinuously depending on the location of system state in the state space. Such phenomena can often be modeled as event-triggered mode switches as follows,

$$x_{k+1} = \sum_{i=1}^{m} f^i(x_k, u_k, k)\,\mathbb{I}_{X^i}(x_k) + d_k, \qquad (20)$$

where the sets $\{X^1, \cdots, X^m\}$ form a partition of the state space $\mathbb{R}^n$ satisfying $\bigcup_{i=1}^m X^i = \mathbb{R}^n$ and $X^i \cap X^j = \emptyset$ for $i \neq j$, $\mathbb{I}_{X^i}(x)$ are set-membership indicator functions taking value 1 if $x \in X^i$ and 0 otherwise, and the functions f^i represent the system dynamics in each *mode* $i = 1, \cdots, m$. Switches between modes are triggered as states x_k enter different sections X^i of the state space. For a given trajectory x_k, we denote by $i_k(x_k)$, or simply i_k when there is no ambiguity, the mode at time k along that trajectory, which is assumed to be measurable. As a matter of fact, the dynamics during a vehicle rollover accident, which are addressed in this paper, experience such discontinuous changes. Specifically, the roll dynamics before and after one-side wheel lift-off can be modeled as two modes of the vehicle system (Yoon et al., 2007).

To treat such systems, necessary steps are introduced into our algorithm to promote convergence of the algorithm iterates. Firstly, when estimating $\frac{\partial f_k}{\partial x_k}\big|_0$ and $\frac{\partial f_k}{\partial u_k}\big|_0$, the perturbed trajectories $\hat{x}_k$ should have the same mode trajectory i_k as the nominal trajectory x_k^0. This can be promoted by maintaining perturbations $\delta\hat{x}_k^i$ in (14) and (17) to be small. In addition, the update step sizes η_k for the inputs u_k are adjusted in each iteration so that the cost values associated with the algorithm iterates are monotone non-increasing.

Our algorithm of iterative gradient-based input update with data-driven gradient estimation for identifying falsifying trajectories is formally presented as Algorithm 1. In Algorithm 1, the termination conditions *TC-A* to *-C* are as follows:

- *TC-A*: The iteration count θ has reached $\theta_{\max}$.
- *TC-B*: The absolute change in the cost J_θ over a moving window of the θ_{win} most recent iterations is less than a threshold value ε.
- *TC-C*: The update step size $\boldsymbol{\eta}_\theta$ has been below a threshold value $\boldsymbol{\eta}_{\min}$.

Remark 1: It can be shown that the cost sequence J_θ generated by Algorithm 1 is guaranteed to be monotone non-increasing, and under the assumption that the cost function (2) is lower bounded, J_θ must converge as θ increases. Then, based on the above termination conditions *TC-A* to *-C*, Algorithm 1 must terminate after a finite number of iterations.

Remark 2: Although the perturbed trajectories $\hat{x}_k$ for estimating $\frac{\partial f_k^{i_k}}{\partial x_k}\big|_0$ and $\frac{\partial f_k^{i_k}}{\partial u_k}\big|_0$ are required to have the same mode trajectory i_k as the nominal trajectory x_k^0, Algorithm 1 allows the updated trajectory $\mathbf{x}_{\theta+1}$ to have different mode trajectories from the previous nominal trajectory $\mathbf{x}_\theta$ as long as $\mathbf{x}_{\theta+1}$ corresponds to a lower cost value than $\mathbf{x}_\theta$.

5. VEHICLE ROLLOVER TEST GENERATION

We apply the proposed falsification algorithm to vehicle rollover test trajectory generation. Specifically, we pursue

Algorithm 1 Iterative Falsification Algorithm

```
Set an initial guess for input $\mathbf{u}_0 = \{u_{k,0}\}_{k=0}^{N-1}$ and an initial update step size $\boldsymbol{\eta}_0 > 0$, set the maximum number of iterations $\theta_{\max}$, and initialize the iteration count $\theta = 0$.
Simulate the black-box model (20) with the nominal initial condition $x_0$ and input $\mathbf{u}_0$, and obtain a nominal state trajectory $\mathbf{x}_0 = \{x_{k,0}\}_{k=0}^{N}$.
Evaluate the cost associated with $(\mathbf{x}_0, \mathbf{u}_0)$ according to (2), denoted as $J_0$.
while TC-A and TC-B not satisfied do
    Initialize $\hat{X}_k^x = [\,]$ for all $k = 0, \cdots, N$.
    while rank$(\hat{X}_k^x) < n$ for some $k = 0, \cdots, N-1$ do
        Add a random perturbation $\delta\hat{x}_0$ to the initial condition $x_0$, and simulate (20) with $x_0 + \delta\hat{x}_0$ and $\mathbf{u}_\theta$ to obtain a state trajectory $\hat{\mathbf{x}}_\theta = \{\hat{x}_{k,\theta}\}_{k=0}^{N}$.
        if $i_k(x_{k,\theta}) = i_k(\hat{x}_{k,\theta})$ for all $k$ then
            Set $\hat{X}_k^x \leftarrow [\hat{X}_k^x \;\; \delta\hat{x}_{k,\theta}]$, $\delta\hat{x}_{k,\theta} = \hat{x}_{k,\theta} - x_{k,\theta}$.
        end if
    end while
    Estimate $\frac{\partial f_k^{i_k}}{\partial x_k}\big|_\theta$ using (15) for all $k$.
    Initialize $\hat{U}_k^u = [\,]$ for all $k = 0, \cdots, N-1$ and $\hat{X}_k^u = [\,]$ for all $k = 0, \cdots, N$.
    while rank$(\hat{U}_k^u) < n_u$ for some $k = 0, \cdots, N-1$ do
        Add a random perturbation signal $\delta\hat{\mathbf{u}}_\theta = \{\delta\hat{u}_{k,\theta}\}_{k=0}^{N-1}$ to $\mathbf{u}_\theta$, and simulate (20) with $x_0$ and $\mathbf{u}_\theta + \delta\hat{\mathbf{u}}_\theta$ to obtain a state trajectory $\hat{\mathbf{x}}_\theta = \{\hat{x}_{k,\theta}\}_{k=0}^{N}$.
        if $i_k(x_{k,\theta}) = i_k(\hat{x}_{k,\theta})$ for all $k$ then
            Set $\hat{U}_k^u \leftarrow [\hat{U}_k^u \;\; \delta\hat{u}_{k,\theta}]$ and $\hat{X}_k^u \leftarrow [\hat{X}_k^u \;\; \delta\hat{x}_{k,\theta}]$, $\delta\hat{x}_{k,\theta} = \hat{x}_{k,\theta} - x_{k,\theta}$.
        end if
    end while
    Estimate $\frac{\partial f_k^{i_k}}{\partial u_k}\big|_\theta$ using (18) for all $k$.
    Initialize $\boldsymbol{\eta}_\theta = \boldsymbol{\eta}_0$ and $\tilde{J} = +\infty$.
    while $\tilde{J} > J_\theta$ and TC-C not satisfied do
        Compute $\delta\tilde{\mathbf{u}}_\theta = \{\delta\tilde{u}_{k,\theta}\}_{k=0}^{N-1}$ using (8) and (10) with $\eta_k = \boldsymbol{\eta}_\theta$.
        Simulate (20) with $x_0$ and $\mathbf{u}_\theta + \delta\tilde{\mathbf{u}}_\theta$, obtain a state trajectory $\tilde{\mathbf{x}}_\theta = \{\tilde{x}_{k,\theta}\}_{k=0}^{N}$, and evaluate the associated cost (2) and assign it to $\tilde{J}$.
        Set $\boldsymbol{\eta}_\theta \leftarrow \frac{\boldsymbol{\eta}_\theta}{2}$.
    end while
    Set $(\mathbf{x}_{\theta+1}, \mathbf{u}_{\theta+1}) = (\tilde{\mathbf{x}}_\theta, \mathbf{u}_\theta + \delta\tilde{\mathbf{u}}_\theta)$ and $J_{\theta+1} = \tilde{J}$.
    Set $\theta \leftarrow \theta + 1$;
end while
```

identification of, if they exist, driver input trajectories that cause the vehicle model under test to have rollover accidents. Here, we consider a sport utility vehicle model provided by the high-fidelity vehicle dynamics simulation software *CarSim*, and focus on the case where the vehicle drives with constant longitudinal speeds and the driver input is the steering wheel profile. In particular, the *CarSim* model is treated as a black-box, i.e., only provides input-output responses to our falsification algorithm.

A rollover event typically exhibits two phases: At first, due to roll dynamics, the vehicle's vertical load starts concentrating on one side of the wheels, the extent to which can be measured using the Load Transfer Ratio (LTR), defined as

$$LTR = \frac{F_{z,L1} + F_{z,L2} - F_{z,R1} - F_{z,R2}}{F_{z,L1} + F_{z,L2} + F_{z,R1} + F_{z,R2}}, \tag{21}$$

where $F_{z,i}$ denotes the vertical force on each of the four wheels. Then, after LTR reaches 1 or -1, one side of the wheels start lifting off the ground, and as the vehicle's roll angle continues increasing, rollover happens.

Note, firstly, that by definition, $-1 \leq LTR \leq 1$, and as a result, the dynamics after one-side wheel lift-off (i.e., after LTR reaches 1 or -1) cannot be reflected by LTR. However, one-side wheel lift-off does not immediately lead to a rollover accident, as rollover requires additional work to lift the vehicle's center of gravity (CG) up. Thus, although LTR has been used as a conservative indicator for designing rollover avoidance devices (Bencatel et al., 2017; Liu et al., 2019), it may not be suitable for identifying trajectories that indeed cause rollover accidents. Secondly, the dynamics before and after one-side wheel lift-off changes in a discontinuous manner, which can be modeled as two modes of the vehicle system with mode switches triggered when LTR reaches 1 or -1 (Yoon et al., 2007).

According to the above observations, we consider a roll dynamics model in the following form,

$$x_{k+1} = \sum_{i=1}^{3} f^i(x_k, u_k, k)\, \mathbb{I}_{X^i}(x_k) + d_k, \tag{22}$$

$$\mathbb{I}_{X^i}(x_k) = 1, \text{ with } i = \begin{cases} 1, & \text{if } -1 < LTR_k < 1, \\ 2, & \text{if } LTR_k \geq 1, \\ 3, & \text{if } LTR_k \leq -1, \end{cases} \tag{23}$$

where the states are $x_k = [\phi_k, p_k, r_k, v_k]^T$ with ϕ_k, p_k, r_k and v_k denoting, respectively, the roll angle, roll rate, yaw rate and lateral velocity of the vehicle at time k, and $u_k = SW_k$ is the steering wheel angle at k. Note that although a high-fidelity vehicle model, such as a *CarSim* model, may involve hundreds of states, these four states have been shown to be most representative of the roll dynamics (Yoon et al., 2007). The effects of the other states are assumed to be embedded into the disturbance term d_k. We will show that the reduced-order model (22) can represent the roll dynamics of the full *CarSim* model with acceptable accuracy.

We pursue trajectories of u that push the vehicle to its rollover limit by minimizing the following cost function:

$$J = -\sum_{k=0}^{N} p_k^2. \tag{24}$$

Such a cost function choice is motivated by the fact that a rollover event corresponds almost always to the absolute value of roll angle, or, the accumulation of roll rate, exceeding a threshold. By comparing several alternatives for the cost function (24), we have found through simulation experiments that our algorithm is more likely to converge to a rollover trajectory by maximizing the accumulation of roll rate squares.

5.1 Reduced-order model validation

Although our algorithm does not rely on the exact form and parameter values of the model (22) to operate, we do expect that the reduced-order model can approximate the full model dynamics with satisfactory accuracy, i.e., with sufficiently small d_k, to guarantee algorithm performance. Fig. 1 illustrates the accuracy of the online estimated local model by our algorithm in approximating the *CarSim* model dynamics. The black dashed lines represent the nominal trajectory (x_k^0, u_k^0) around which a linear time-varying model (12) is identified. The red lines represent the response predicted by the identified linear model when the initial condition x_0^0 and input signal $\{u_k^0\}_{k=0}^{N-1}$ are perturbed, and the green lines represent the *CarSim* model response to the same perturbations. The perturbation to the initial condition is achieved through a small steering wheel input applied for a period of time before the initial time, i.e., $k = 0$ corresponding to 1[s] in the plots, which is also how Step 7 of Algorithm 1 is realized.

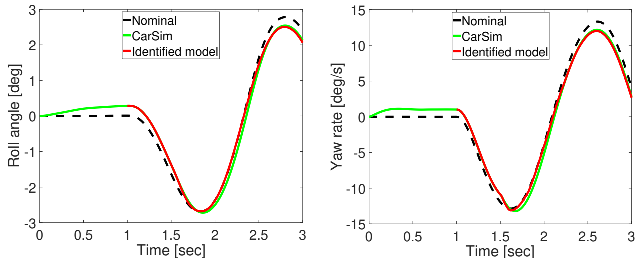

Fig. 1. Validation of identified reduced-order model.

5.2 *Algorithm application to CarSim model*

We now apply our algorithm to identify *worst-case* trajectories for the *CarSim* vehicle model under test, i.e., driver steering wheel profiles that most likely cause the vehicle to have rollover accidents. In particular, we consider two cases: In the first case, the vehicle drives at a speed of 100[km/h] on a flat road with a friction coefficient 1.0. In the second case, the vehicle drives also at a speed of 100[km/h] on a road also with a friction coefficient 1.0 but with a bank angle of arctan(1/10)[rad]. We consider steering wheel angles bounded by $|u| \leq 120°$ to represent physical limitations of driver maneuver. When initializing the algorithm, we consider a sine wave as the initial guess for steering wheel profile (black dashed lines in Fig. 2(a)), which represents a typical obstacle avoidance maneuver. The blue lines in Fig. 2(a)-(c) represent the steering wheel, LTR, and roll angle trajectories after algorithm convergence, where some intermediate trajectories during algorithm iterations are also plotted by green lines. Fig. 2(d) plots the cost values during algorithm iterations, from which it can be observed that the cost monotonically non-increases at most iterations until it converges. At a few instances the trend is slightly reversed. This is due to the fact that in our implementation of Algorithm 1, the condition $\tilde{J} > J_\theta$ in Step 22 was checked up to a tolerance.

In both cases, the LTR reaches the wheel lift-off limit (1 or -1), and the roll angle reaches a large value at the end of the maneuver. Due to the positive bank angle, the roll angle reaches an even larger value in the second case, which is consistent with the common sense that vehicles are easier to have rollover accidents when driving on banked roads.

Note that although the initial trajectories (black dashed) do not experience mode switches (mode $i \equiv 1$), mode switches occur once LTR reaches 1 or -1 at around 2.5[s] as the trajectories are evolved over algorithm iterations. Afterwards, the estimated gradients (15) and (18) after about 2.5[s] correspond to mode 2 or 3 of the system. This illustrates the feasibility of using our algorithm to treat switching systems (see Remark 2). Note also that the identified steering wheel profiles are different for the two cases (e.g., at around 1.5[s]). This illustrates the fact that worst-case trajectories for the same vehicle model under different operating conditions can be different.

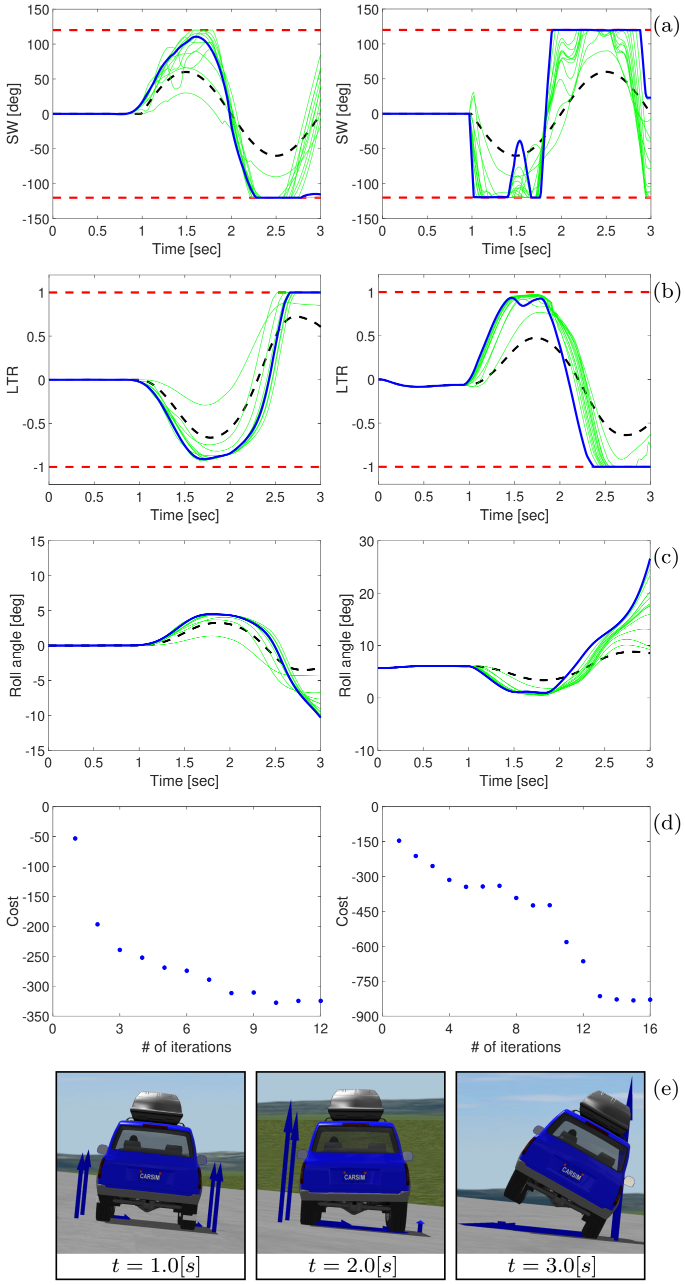

Fig. 2. Identification of worst-case trajectories by Algorithm 1. Left column (a-d): On flat road. Right column (a-d): On road with a positive bank angle. (e) Snapshots of *CarSim* simulation with the converged steering wheel profile (blue) on road with a positive bank angle.

5.3 *Test trajectory library*

A library of worst-case trajectories for a specified vehicle model and for a range of operating conditions can be constructed and used to inform future hardware tests. As an illustrative example, we use our algorithm to construct steering wheel profiles for the *CarSim* vehicle model considered above when the vehicle is driven on flat road with different constant longitudinal speeds, plotted in Fig. 3. We envision that test cases designed based on such a test trajectory library can provide more reliable assessment results on vehicle roll stability and rollover resistance, as well as can reduce overall testing time (Li et al., 2017).

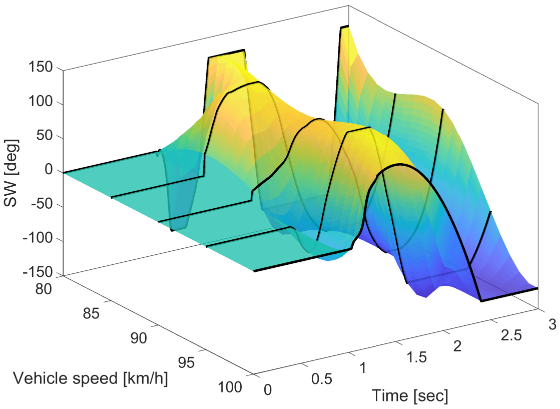

Fig. 3. Steering wheel profiles for rollover testing when driving on flat road with different longitudinal speeds.

6. CONCLUSION

In this paper, we described a trajectory optimization-based approach for control system falsification. Our approach integrates online local-model identification and gradient-based input update into an iterative algorithm, and can be applied to black-box type models. In particular, we considered the case study of vehicle rollover test generation. We illustrated the functionality and effectiveness of our falsification algorithm for identifying worst-case trajectories, in terms of most likely causing rollover accidents, of a given vehicle model under various operating conditions. We hope that the proposed falsification approach can be a useful tool that supports the verification and validation of automotive control systems.

REFERENCES

Annpureddy, Y., Liu, C., Fainekos, G., and Sankaranarayanan, S. (2011). S-TaLiRo: A tool for temporal logic falsification for hybrid systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 254–257. Springer.

Bencatel, R., Tian, R., Girard, A.R., and Kolmanovsky, I. (2017). Reference governor strategies for vehicle rollover avoidance. *IEEE Transactions on Control Systems Technology*, 26(6), 1954–1969.

Carlson, C.R. and Gerdes, J.C. (2003). Optimal rollover prevention with steer by wire and differential braking. In *Proceedings of IMECE*, volume 3, 16–21.

Chen, B.C. and Peng, H. (2001). Differential-braking-based rollover prevention for sport utility vehicles with human-in-the-loop evaluations. *Vehicle System Dynamics*, 36(4-5), 359–389.

Cheng, P. and Kumar, V. (2008). Sampling-based falsification and verification of controllers for continuous dynamic systems. *The International Journal of Robotics Research*, 27(11-12), 1232–1245.

Fainekos, G.E. and Pappas, G.J. (2009). Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42), 4262–4291.

Gáspár, P., Szászi, I., and Bokor, J. (2005). Two strategies for reducing the rollover risk of heavy vehicles. *Periodica Polytechnica Transportation Engineering*, 33(1-2), 139–147.

Kasac, J., Deur, J., Novakovic, B., Kolmanovsky, I.V., and Assadian, F. (2010). A conjugate gradient-based BPTT-like optimal control algorithm with vehicle dynamics control application. *IEEE Transactions on Control Systems Technology*, 19(6), 1587–1595.

Li, L., Lu, Y., Wang, R., and Chen, J. (2016). A three-dimensional dynamics control framework of vehicle lateral stability and rollover prevention via active braking with MPC. *IEEE Transactions on Industrial Electronics*, 64(4), 3389–3401.

Li, N., Girard, A., and Kolmanovsky, I. (2018). Optimal control based falsification of unknown systems with time delays: A gasoline engine A/F ratio control case study. *IFAC-PapersOnLine*, 51(31), 252–257.

Li, N., Kolmanovsky, I., and Girard, A. (2017). Model-free optimal control based automotive control system falsification. In *2017 American Control Conference (ACC)*, 636–641. IEEE.

Liu, K., Li, N., Rizzo, D., Garone, E., Kolmanovsky, I., and Girard, A. (2019). Model-free learning to avoid constraint violations: An explicit reference governor approach. In *2019 American Control Conference (ACC)*, 934–940. IEEE.

Lu, J., Messih, D., Salib, A., and Harmison, D. (2007). An enhancement to an electronic stability control system to include a rollover control function. *SAE Transactions*, 303–313.

Ma, W.H. and Peng, H. (1999a). A worst-case evaluation method for dynamic systems. *Journal of Dynamic Systems, Measurement, and Control*, 121(2), 191–199.

Ma, W.H. and Peng, H. (1999b). Worst-case vehicle evaluation methodology–examples on truck rollover/jackknifing and active yaw control systems. *Vehicle System Dynamics*, 32(4-5), 389–408.

Nghiem, T., Sankaranarayanan, S., Fainekos, G., Ivancić, F., Gupta, A., and Pappas, G.J. (2010). Monte-carlo techniques for falsification of temporal properties of nonlinear hybrid systems. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, 211–220.

NHTSA, N.H.T.S.A. (2004). A demonstration of the dynamic tests developed for NHTSA's NCAP rollover rating system. Technical report, U.S. Department of Transportation. Available at `https://www.nhtsa.gov/DOT/NHTSA/NRD/Multimedia/PDFs/VRTC/ca/capubs/RolloverPhaseVIIIReport_NCAPdemo081104.pdf`.

NHTSA, N.H.T.S.A. (2007). FMVSS No. 126: Electronic stability control systems. Technical report, U.S. Department of Transportation. Available at `https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/fmvss/ESC_FRIA_%252003_2007.pdf`.

NHTSA, N.H.T.S.A. (2017). Trends and rollover-reduction effectiveness of static stability factor in passenger vehicles. Technical report, U.S. Department of Transportation. Available at `https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812444`.

NHTSA, N.H.T.S.A. (2018). Traffic safety facts 2016. Technical report, U.S. Department of Transportation. Available at `https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812554`.

Rajamani, R. (2011). *Vehicle dynamics and control*. Springer Science & Business Media.

Solmaz, S., Corless, M., and Shorten, R. (2007). A methodology for the design of robust rollover prevention controllers for automotive vehicles with active steering. *International Journal of Control*, 80(11), 1763–1779.

Yaghoubi, S. and Fainekos, G. (2019). Gray-box adversarial testing for control systems with machine learning components. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 179–184. ACM.

Yoon, J., Kim, D., and Yi, K. (2007). Design of a rollover index-based vehicle stability control scheme. *Vehicle system dynamics*, 45(5), 459–475.