The International Journal of Robotics Research
000(00):1–19

DOI:doi number
http://mms.sagepub.com

Minimum Sensitivity Control for Planning with Parametric and Hybrid Uncertainty

Alex Ansari* **and Todd Murphey**
Department of Mechanical Engineering, McCormick School of Engineering and Applied Science, Northwestern University, USA

Abstract

This paper introduces a method to minimize norms on nonlinear trajectory sensitivities during open-loop trajectory optimization. Specifically, we derive new *parametric sensitivity* terms that measure the variation in nonlinear (continuous-time) trajectories due to variations in model parameters, and *hybrid sensitivities*, which account for variations in trajectory caused by sudden transitions from nominal dynamics to alternative dynamic modes. We adapt continuous trajectory optimization to minimize these sensitivities while only minimally changing a nominal trajectory. We provide appended states, cost, and linearizations, required so that existing open-loop optimization methods can generate minimally sensitive feedforward trajectories.

Though there are several applications for sensitivity optimization, this paper focuses on robot motion planning, where popular sample-based planners rely on local trajectory generation to expand tree / graph structures. While such planners often use stochastic uncertainty propagation to model and reduce uncertainty, this paper shows that trajectory uncertainty can be reduced by minimizing first-order sensitivities. Simulated vehicle examples show parametric sensitivity optimization generates trajectories optimally insensitive to parametric model uncertainty. Similarly, minimizing hybrid sensitivities reduces uncertainty in crossing mobility hazards (e.g. rough terrain, sand, ice). Examples demonstrate the process yields a planner that uses approximate hazard models to automatically and optimally choose when to avoid hazardous terrain and when controls can be adjusted to traverse hazards with reduced uncertainty. Sensitivity optimization offers a simple alternative to stochastic simulation and complicated uncertainty modeling for nonlinear systems.

Keywords
Optimal control, motion control, motion planning, nonlinear control systems

1. Introduction

Feedforward control strategies can improve tracking of reference trajectories and reduce feedback and sensing requirements, but these strategies perform poorly when real world conditions vary from those modeled. To improve reference tracking

* Corresponding author; email: alexanderansari2011@u.northwestern.edu

under uncertain conditions that vary from those modeled, this paper derives means to adapt standard trajectory optimization to include norms on trajectory sensitivities. First, we contribute new *parametric sensitivity* terms, which measure the variation in trajectory due to a variation in model parameters. By incorporating these parametric sensitivities into optimal control calculations, alternative trajectories can be selected that remain optimally insensitive to fluctuations in model parameters when tracked using feedforward control. Hence, better tracking can be achieved using significantly less feedback or none at all. Next, we define a new *hybrid sensitivity* term that measures the change in trajectory that results due to a sudden transition from nominal dynamics to an alternate and uncertain dynamic mode (uncertain in the sense that only approximate model dynamics for the alternative model are available). While the hybrid sensitivity term is based on a quantity (the mode insertion gradient in Wardi and Egerstedt (2012)) used to schedule dynamic modes in hybrid system optimization, we derive means to generate trajectories for continuous (nonlinear) systems that minimize a norm on this term to reduce sensitivity to uncertain dynamic modes which may be encountered throughout state space.

The sensitivity optimization methods described apply broadly to differentiable nonlinear systems. This paper focuses on applications in the domain of robotic motion planning. In particular, we note that the proposed parametric sensitivity techniques are similar in spirit to ensemble control from Becker and Bretl (2012) and Becker et al. (2014). They can, for example, be used to plan grasps insensitive to local shape and model variations in Kehoe et al. (2014). They also offer a simple alternative to the robust control synthesis techniques used in Marino et al. (2014) to steer microrobots in a manner that remains insensitive to inaccuracies in modeled forces. However, both parametric and hybrid sensitivity optimization techniques are particularly appropriate for sample-based motion planning (LaValle 2006). In this setting, sensitivity optimization offers the means to generate reliable trajectories to extend rapidly exploring random trees (RRTs) (LaValle and Kuffner 2001) or to connect roadmap nodes (Prentice and Roy 2009) while compensating for uncertainty in model dynamics. Sensitivity optimization provides a simple and efficient alternative to direct uncertainty modeling in planning.

While the potential for parametric sensitivity optimization to develop trajectories that are insensitive to uncertainty in model parameters has relatively straightforward application in planning, the utility of hybrid sensitivity optimization may be less obvious. We show that hybrid sensitivity optimization reduces tracking uncertainty generated in crossing poorly understood regions of the state space containing mobility hazards[1]. As an example, consider a driver traveling along a winter road. Most such drivers plan courses to avoid ice patches or slow and align with the direction of travel before passing through them. These intuitive driving styles are applied to keep the vehicle moving in a safe and predictable manner that minimizes trajectory uncertainty induced by the hazard. These driving styles can also be posed as solutions to an optimal control problem that includes trajectory uncertainty in the objective. In this context, the driver attempts to track a desired trajectory, minimize control input, and avoid un-planned changes in trajectory that could be caused by road hazards. Much less obvious, however, these uncertainty / hazard mitigation behaviors are also the natural result of the process of hybrid sensitivity optimization.

By minimizing a norm on hybrid sensitivity terms in trajectory optimization, synthesized trajectories remain optimally insensitive to discrete dynamic changes (e.g. those that might occur when the wheels of a vehicle slip over ice or stick in sand). As global terrain models are generally unavailable, the exact effect of mobility hazards is uncertain. We show that by minimizing hybrid sensitivities to approximated dynamic hazard models a planner can efficiently minimize trajectory uncertainty generated in crossing them. Similar to traditional dynamic planners that assume rigid obstacles, sensitivity optimization can plan trajectories that circumvent hazardous regions. However, the process can also vary controls to pass through these regions in a reliable and insensitive manner when avoidance is not feasible. In the winter driving example, hybrid sensitivity optimization can adjust heading to align the vehicle and avoid steering over the ice, minimizing its uncertain influence.

[1] We define mobility hazards as regions of state space containing uncertain or difficult-to-model dynamics that differ from nominal dynamics in other regions. For example, an icy area on an otherwise smooth road presents a mobility hazard that a robot might need to cross (or avoid).

This paper presents three example vehicle scenarios to demonstrate the performance of the parametric and hybrid sensitivity optimization techniques. First, as an application of the parametric sensitivity optimization approach, a variable inertial vehicle modeling an experimental robot is simulated driving under varying friction. We show that the techniques presented are capable of producing insensitive vehicle driving control policies with improved tracking performance under uncertain model conditions. Example 4.2 depicts a nonlinear, $2D$ kinematic version of the vehicle modeling the geometry of sensitivity avoidance around a sand pit. In Example 4.3, a high-speed skidding car model demonstrates controlled hazard mitigation in crossing a dynamically uncertain region expected to contain ice. We show that even approximate models of ice as a low friction environment and sand as high friction incorporate sufficient information for the planner to produce trajectories that employ intuitive strategies to deal with these obstacles. For the purposes of illustration, Extension 1 includes simulations of robotic vehicles using sensitivity optimized controllers to track desired trajectories in uncertain environments corresponding to the examples just described.

Following this Introduction, Section 2 presents a brief background on existing theory and an overview of the iterative, nonlinear optimal control technique leveraged to provide feasible solutions for trajectories and controllers at each iteration. Sections 3.1 and 3.2 describe how the method can be adapted to minimize norms on parametric and hybrid sensitivities, respectively. Section 4 presents the three vehicle examples used to demonstrate the performance of the parametric (Section 4.1) and hybrid (Sections 4.2 and 4.3) sensitivity optimization techniques discussed. Concluding remarks are presented in Section 5.

2. Background

This paper extends the contributions of the authors' preliminary works in parametric (Ansari and Murphey 2013*a*) and hybrid sensitivity optimization (Ansari and Murphey 2013*b*). Here, we highlight the similarities between and provide parallel derivations for how to compute and optimize these separate sensitivity measures, contributing a unified approach to sensitivity based planning for uncertainty. With regard to parametric sensitivities, the techniques described build upon existing research in sensitivity optimization and optimal control. For example, in Becker and Bretl (2012), ensemble control is utilized to develop approximate control strategies that steer a unicycle subject to bounded model perturbations. The potential for improved feedforward control to enhance disturbance rejection for controllers with both feedback and feedforward terms is demonstrated in Singer and Seering (1990). Methods similar to those presented in this paper but restricted to linear time invariant systems are discussed in Byrne and Burke (1976) and Kreindler (1969). In Chen et al. (2009) and Fernandes de Paula and Henrique Carvalho de Ferreira (2011), H_∞ methods for controller sensitivity minimization are derived for linear and time-discretized systems. While other examples can be cited, the existing sensitivity optimization techniques identified rely on specific topological or structural system properties, linear dynamics, or are in other ways limited in generality.

With applications similar to those of the hybrid sensitivity optimization techniques discussed, several authors have developed methods for path and trajectory planning that account for critical or hazardous areas of the state space. For example, Boyd and Vandenberghe (2004) use barrier functions for general inequality constrained optimization. Hauser and Saccon (2006) use barrier functions in trajectory optimization to plan trajectories that avoid obstacles. Sample-based motion planners in Jaillet and Simeon (2004) and van den Berg et al. (2006) use RRTs (LaValle and Kuffner 2001) and similar local trajectory generation schemes to connect nodes in probabilistic roadmaps and plan paths through environments with moving obstacles. In Petti and Fraichard (2005), RRTs plan deterministic trajectories through collision free space in similar environments. Results from Likhachev and Ferguson (2009) demonstrate Anytime Dynamic A* in computing suboptimal paths at the DARPA grand challenge. These planning techniques all use deterministic local trajectory generation to explore collision free states but do not compensate for the influence of terrain mobility factors or motion uncertainty.

To account for uncertainty, sample-based motion planners from Candido and Hutchinson (2010) and Candido and Hutchinson (2011) apply a belief roadmap approach (Prentice and Roy 2009) and optimize a sequence of local feedback policies to provide a composite partially observable markov decision process (POMDP) policy. These planners use particle filtering to propagate and minimize motion and sensing uncertainty in environments with static obstacles. In the same environments, van den Berg et al. (2011) linearizes systems along deterministic RRT candidate paths and approximates motion and sensing uncertainty using LQG methods to estimate a priori state distributions. While planners from Kewlani et al. (2009) and Melchior and Simmons (2007) directly account for model and even terrain uncertainty, they rely on stochastic response surfaces and Monte Carlo sampling to propagate uncertainty through RRTs. The authors of Kobilarov et al. (2011) perform stochastic optimization of dynamically adaptive random trees to develop trajectories for minimal uncertainty estimation in environments with obstacles. Planners from Banerjee et al. (2010) model systems according to a POMDP and use stochastic dynamic programming to compute optimal trajectories in environments with obstacles. A partially closed-loop Receding Horizon control approach in Du Toit and Burdick (2012) approximately solves a similar stochastic dynamic programming problem. Feedback motion planing from Tedrake et al. (2010) uses stability regions to develop sparse trees of LQR-stabilized trajectories for smooth nonlinear systems. While all these planners are adept at computing paths around rigid obstacles, only a few consider terrain mobility factors, and those that address uncertainty typically do so at great computational expense.

Hybrid sensitivity optimization is unique in that it applies to general differentiable systems and yet allows trajectory generation to accommodate broad classes of static and dynamic mobility hazards without explicit uncertainty modeling. It avoids costly nonlinear uncertainty propagation techniques (e.g particle filtering and stochastic response surfaces) and instead minimizes easily computed, first-order sensitivities to mitigate uncertain dynamic effects in hazardous regions. Though most alternatives allow only rigid obstacles, the approach naturally modifies both path and controls to minimize the influence of mobility hazards. Through the process, hybrid sensitivity based planners can compute reliable trajectories both through and around hazardous regions.

Rather than compete with sample-based motion planners, LaValle and Kuffner (2001) explains that trajectory optimization methods like those in this paper complement planning. In addition to using sensitivity optimization for local trajectory generation (extending RRTs and connecting PRMs), it can refine paths produced by deterministic motion planners. Deterministic planners are efficient at exploring and developing paths through high dimensional space. Though not generally optimal, these paths provide globally feasible seed trajectories that benefit trajectory optimization. By iteratively refining these paths subject to constraints, sensitivity optimization offers a simple alternative to direct uncertainty modeling in planning that can develop reliable trajectories.

The hybrid sensitivity terms utilized in this paper were developed from research in the area of hybrid systems. Specifically, we make use of the mode insertion gradient (see Chudoung and Beck (2001), Egerstedt et al. (2003), Egerstedt et al. (2006), Wardi et al. (2012), and Wardi and Egerstedt (2012)) to provide a measure of the sensitivity of a cost function to a discrete change in dynamics. More formally, the term provides a first-order approximation of how a cost function is impacted by a discrete dynamic change. As the approximation is valid for some neighborhood around the mode transition time, the mode insertion gradient has been used in mode scheduling (Egerstedt et al. (2006), Wardi et al. (2012), Wardi and Egerstedt (2012)) and switching time optimization problems (Egerstedt et al. 2003) to determine the optimal time to insert or remove dynamic modes from hybrid trajectories. By optimizing trajectories with respect to an L_2 norm on the mode insertion gradient, we are able to ensure trajectories remain insensitive to new dynamic modes that may be encountered at any point along a trajectory.

The sensitivity minimization methods formulated in this paper can use a number of optimization techniques that rely on iterative solutions to approximate LQR problems. This paper uses a nonlinear projection-based optimal control algorithm detailed in Hauser and Meyer (1998), Hauser (2002), and Rucco et al. (2010). We emphasize this approach for three reasons— 1) it applies to continuous-time linear and nonlinear systems without a-priori discretization, 2) it provides

feasible trajectories at each iteration, and 3) it provides feedback controllers at each iteration. The following Section 2.1 provides an overview of this algorithm.

2.1. The Optimal Control Algorithm

This section presents a general, iterative means to compute optimal trajectories that minimize norms on state and controls for nonlinear systems. The algorithm performs iterative optimization of a cost functional of the form

$$J = \int_{t_0}^{t_f} l(X(t), U(t))dt + m(X(t_f)) \tag{1}$$

with respect to the state and control pair, $\xi(t) = (X(t), U(t))$, constrained by dynamics

$$\dot{X}(t) = f(X(t), U(t)). \tag{2}$$

To perform this nonlinear constrained optimization, the algorithm iteratively minimizes quadratic models of the cost (1) subject to locally linearized constraints. Local approximations are derived using the quadratic model[2]

$$g(\zeta) = DJ(\xi) \circ \zeta + \frac{1}{2} \int_{t_0}^{t_f} \|\zeta\|^2 dt, \tag{3}$$

where $\zeta(t) = (z(t), v(t))$ represents perturbations to the state, $X(t)$, and controls, $U(t)$, from the current trajectory. The model (3) is subject to the constraint that the perturbations locally obey the linearized dynamics

$$\dot{z}(t) = A(t)z(t) + B(t)v(t). \tag{4}$$

The $A(t)$ and $B(t)$ terms in the constraint above are linearizations of dynamics (2) with respect to state and controls, respectively. Performing constrained minimization of the quadratic model with respect to its argument, ζ, provides the direction of "steepest" descent in the cost functional. This minimization can be calculated as the solution to an LQR problem for which there are well known methods of optimization (see Anderson and Moore (1990)).

Through standard iterative descent techniques, at each iteration the previous best trajectory is perturbed in the descent direction ζ. However, in doing so trajectories diverge from the manifold of feasible trajectories, $\mathcal{T}$, to produce the infeasible state and controls pair, $(\alpha(t), \mu(t))$. To ensure feasible trajectories result at each iteration, a proportional feedback control gain, $K(t)$, is utilized in a feedback projection. With constraint (2) and a quadratic model similar to (3), a second LQR problem can be solved to obtain an optimal feedback control gain, $K(t)$, from the current linearized trajectory (see Hauser and Meyer (1998)).

In the projection, the current feasible control solution, $U(t)$, is computed as a function of the feasible state, $X(t)$, from

$$U(t) = \mu(t) + K(t)[\alpha(t) - X(t)]. \tag{5}$$

The state is then computed from the resulting control law based on dynamics (2). Following this methodology, optimization can be halted at any time to return the current best feasible solution, $(X(t), U(t))$,[3] necessary for open-loop control and the corresponding optimal feedback control law, $K(t)$. In the included examples, open-loop controls, $U(t)$, will be compared to solutions derived when the process above is adapted to minimize norms on proposed sensitivity terms. These examples

[2] $DJ(\xi)$ in equation (3) refers to the slot derivative of $J(\xi)$ with respect to its argument. More generally $D_n F(arg_1, arg_2, \ldots, arg_n)$ refers to the slot derivative of function F with respect to its n^{th} argument.

[3] Assuming a minima exists, this gradient-based method for minimizing (1) subject to (2) is guaranteed to converge within arbitrary tolerance. Upon convergence, $U(t)$ represents the locally optimal feedforward control law associated with locally optimal state trajectory $X(t)$.

Algorithm 1 Optimal Control Algorithm

Specify initial trajectory, $\xi(t)$, with guess of optimal solution.[4]
while convergence tests fail [5] **do**
  Approximate cost functional along the trajectory, $\xi(t)$, using a local quadratic model (3)
  Minimize the local model (3) subject to (4) by solving an LQ problem to obtain $\zeta(t)$
  Minimize a 2^{nd} quadratic model to obtain optimal feedback gain, $K(t)$
  Perturb $\xi(t)$ by $\zeta(t)$
  Use $K(t)$ to project perturbed $\xi(t)$ onto $\mathcal{T}$
  Apply Armijo line search to scale $\zeta(t)$ until the perturbed and projected $\xi(t)$ provides sufficient decrease [6]
  Return the updated $\xi(t)$ as current best solution
end while

Algorithm 1. The projection-based optimal control algorithm from Hauser and Meyer (1998) develops trajectories that minimize norms on state tracking errors and control effort for nonlinear dynamical systems. The algorithm can be halted at any time to provide the current best feedforward trajectory $\xi(t) = (X(t), U(t))$ (and an optimal LQR feedback gain matrix $K(t)$). This paper defines appended versions of the terms above and uses the algorithm as is to develop trajectories that simultaneously minimize norms on proposed sensitivity terms.

will show sensitivity optimized solutions trajectories, $(X(t), U(t))$, are more reliable in that robots implementing these open-loop trajectories are less affected by uncertain model conditions. Hence, sensitivity optimized trajectories offer better tracking with reduced feedback and sensing requirements. For more detail on the trajectory optimization methods described, refer to Hauser and Meyer (1998) and Hauser (2002). An overview of the approach is provided in Algorithm 1.

In comparing optimal control solutions in Section 4.1-4.3, standard (non-sensitivity optimized) results use the methods described to derive controllers that drive simulated robotic vehicles to track a desired path while minimizing a norm on the state tracking error and applied control. In this paper, the norm is a quadratic incremental cost,

$$\begin{aligned} l(X(t), U(t)) \; &= \frac{1}{2}(X(t) - X_d(t))^T \, Q \, (X(t) - X_d(t)) + \\ &+ \frac{1}{2}(U(t) - U_d(t))^T \, R \, (U(t) - U_d(t)) \, . \end{aligned} \tag{6}$$

Additionally, we incorporate a terminal cost into equation (1) to provide a norm on the state tracking error at the terminal time,

$$m(X(t_f)) = \frac{1}{2}(X(t_f) - X_d(t_f))^T \, P_1 \, (X(t_f) - X_d(t_f)) \, . \tag{7}$$

Through the choice of Q, R, and P_1 matrices, this cost functional provides a straightforward means to influence trade-offs made during optimization.

3. Sensitivity Optimization

This section derives new parametric (Section 3.1) and hybrid sensitivity (Section 3.2) terms. By appending to the state, we demonstrate how norms on these sensitivities can be incorporated in a manner that facilitates use of the general approach to nonlinear optimal control presented in Section 2.1. Resulting optimal control formulations select trajectories that optimize the trade-off between state tracking goals, control effort, and the sensitivity of planned motion to variations in model parameters and uncertain dynamic modes.

[4] Often, $\xi(t)$ is initialized as the trajectory resulting from null (zero) control, $U(t) = \mathbf{0}$, and an initial state $x(t_0) = x_0$.
[5] A number of convergence criteria may be applied to halt trajectory optimization (e.g. a norm on the difference between trajectory solutions, the magnitude of the step size from Armijo, or the magnitude of $g(\zeta)$). In practice, we test a norm on the directional derivative, $\|DJ(\xi) \circ \zeta\| > \epsilon$ for $\epsilon \in (0, 1)$.
[6] See Wright and Nocedal (1999) for a description of the Armijo line search algorithm and a discussion of sufficient decrease conditions.

3.1. Parametric Sensitivity

The cost functional (1) and optimization approach described in Section 2.1 can compute feedforward control laws capable of optimally driving a system along a desired trajectory under modeled conditions. However, tracking performance quickly degrades when actual parameters do not match those modeled. This section derives a strategy to minimize a norm on parametric sensitivity in the optimization process that can plan insensitive trajectories for both linear and nonlinear systems. That is, the approach selects optimally insensitive trajectories close to the desired trajectory that can be safely executed in spite of parametric fluctuations using only open-loop control.

This approach to parametric sensitivity optimization requires a measure of sensitivity to minimize. First, since the state depends on dynamics themselves dependent on a parameter, we define the state as a function of a model parameter of interest, $X(t) := X(t,p)$, $p \in \mathbb{R}$. The measure of parametric sensitivity used in the proposed optimization is a norm on $\frac{dX(t,p)}{dp}$, the first-order change in the system's state with respect to a change in the parameter. Due to the property that scalar multiplication and mixed partial derivatives commute, the expression $\frac{d}{dt}\frac{dX(t,p)}{dp}$ is equivalent to $\frac{d\dot{X}(t,p)}{dp}$. Therefore, $\frac{dX(t,p)}{dp}$ can be calculated by taking the derivative of (2) with respect to the parameter and solving the resulting ODE, given by

$$\frac{d}{dt}\left(\frac{dX(t,p)}{dp}\right) = D_1 f(X(t,p),U(t),p)\,\frac{dX(t,p)}{dp} + D_3 f(X(t,p),U(t),p), \tag{8}$$

subject to the initial condition $\frac{dX(t,p)}{dp}\mid_{t=t_0} = \mathbf{0}$.

In the optimal control formulation discussed in Section 2.1, the incremental cost functional (6) only requires solutions to its arguments, $X(t)$ and $U(t)$, to evaluate. As derived above, calculation of the sensitivity term depends on the forward integration of a differential equation of the dynamics and thus cannot be added as an additional term to the objective. Instead, the parametric sensitivity term can be appended to the state vector, $X(t) := X(t,p)$. The appended state vector is denoted $\overline{X}(t) := [X(t,p)^T,\ \frac{dX(t,p)}{dp}^T]^T$.

Appending to the state in this manner requires new definitions for appended weighting matrices, $\overline{Q} \in \mathbb{R}^{2n\times 2n}$ and $\overline{P}_1 \in \mathbb{R}^{2n\times 2n}$, to replace positive semi-definite weight matrices, $Q \in \mathbb{R}^{n\times n}$ and $P_1 \in \mathbb{R}^{n\times n}$, which define norms on state tracking error in (1). The elements of these weighting matrices should be chosen to induce the desired quadratic norms on state tracking error and sensitivity terms while remaining positive semi-definite. These modifications, result in the parametric sensitivity optimization cost functional[7]

$$\bar{J} = \frac{1}{2}\int_{t_0}^{t_f} \|\overline{X}(t) - \overline{X}_d(t)\|^2_{\overline{Q}} + \|U(t) - U_d(t)\|^2_R\, dt + \frac{1}{2}\|\overline{X}(t_f) - \overline{X}_d(t_f)\|^2_{\overline{P}_1}. \tag{9}$$

To provide the quadratic norm on sensitivities in (9), the second half of the desired trajectory $\overline{X}_d(t)$ should be $\mathbf{0}$.

The appended descent directions, $\overline{\zeta}(t) = (\overline{z}(t), v(t))$, required to iteratively optimize (9) are computed as solutions to the constrained minimization problem

$$\min_{\overline{\zeta}} \overline{g}(\overline{\zeta}) = D\bar{J}(\overline{\xi}) \circ \overline{\zeta} + \frac{1}{2}\int_{t_0}^{t_f} \|\overline{\zeta}\|^2 dt \tag{10}$$

subject to

$$\dot{\overline{z}}(t) = \overline{A}(t)\overline{z}(t) + \overline{B}(t)v(t). \tag{11}$$

[7] The notation, $\|\cdot\|^2_M$, indicates a norm on the argument where matrix, M, provides the metric (i.e. $\|X(t)\|^2_Q = X(t)^T\, Q\, X(t)$).

As formulated, $\overline{g}(\overline{\zeta})$, provides a local quadratic approximation to (9) analogous to the way $g(\zeta)$ in (3) locally approximates (1). Noting that $g(\zeta)$ relies on computation of

$$\begin{aligned} DJ(\xi)\circ\zeta &= \int_{t_0}^{t_f} D_1 l(X(t),U(t))\, z(t) + D_2 l(X(t),U(t))\, v(t)\, dt + Dm(X(t_f))\, z(t_f) \\ &= \int_{t_0}^{t_f} (X(t)-X_d(t))^T\, Q\, z(t) + (U(t)-U_d(t))^T\, R\, v(t)\, dt + (X(t_f)-X_d(t_f))^T\, P_1\, z(t_f), \end{aligned} \tag{12}$$

one need only replace the terms in (12) with appended versions to construct the appended $D\bar{J}(\overline{\xi})\circ\overline{\zeta}$ required for $\overline{g}(\overline{\zeta})$. However, constraint (11) requires new linearizations, $\overline{A}(t)$ and $\overline{B}(t)$, of the appended state dynamics.

The linearization of the appended dynamics,

$$\overline{f}(\overline{X}(t),U(t),p) = \begin{pmatrix} f(X(t,p),U(t),p) \\ \frac{d\dot{X}(t,p)}{dp} \end{pmatrix}, \tag{13}$$

with respect to the appended state vector, $\overline{X}(t)$, and the control vector, $U(t)$, are

$$\overline{A}(t) = \begin{pmatrix} D_1 f(X(t,p),U(t),p) & \mathbf{0} \\ \frac{\partial}{\partial X(t,p)}\frac{d\dot{X}(t,p)}{dp} & D_1 f(X(t,p),U(t),p) \end{pmatrix} \tag{14}$$

and

$$\overline{B}(t) = \begin{pmatrix} D_2 f(X(t,p),U(t),p) \\ \frac{\partial}{\partial U(t)}\frac{d\dot{X}(t,p)}{dp} \end{pmatrix}, \tag{15}$$

respectively. The two new equations that emerge from the linearizations, $\overline{A}(t)$ and $\overline{B}(t)$, are obtained by taking partial derivatives of (8). These required partial derivatives are [8]

$$\left(\frac{\partial}{\partial X(t,p)}\frac{d\dot{X}(t,p)}{dp}\right)_{i,m} = \sum_j \frac{\partial A_{i,j}}{\partial X_m}\left(\frac{dX(t,p)}{dp}\right)_j + \left(\frac{\partial}{\partial X(t,p)} D_3 f(X(t,p),U(t),p)\right)_{i,m} \tag{16}$$

and

$$\left(\frac{\partial}{\partial U(t)}\frac{d\dot{X}(t,p)}{dp}\right)_{i,m} = \sum_j \frac{\partial A_{i,j}}{\partial U_m}\left(\frac{dX(t,p)}{dp}\right)_j + \left(\frac{\partial}{\partial U(t)} D_3 f(X(t,p),U(t),p)\right)_{i,m}. \tag{17}$$

Plugging (16) and (17) into (14) and (15) provides linearizations of the appended dynamical system required for the LQR problems used in each iteration of optimization. By appending the state vector with sensitivity terms in this manner, the existing projection based optimization approach can therefore be applied to parametric sensitivity optimization without modification to Algorithm 1.

3.2. Hybrid Sensitivities

To select for trajectories that remain insensitive to sudden changes in dynamics (such as may be encountered in crossing hazardous terrain), it is necessary to quantify the sensitivity of a trajectory to a discrete change in dynamics. If cost functional

[8] Subscripts in (16) and (17) reference the matrix (vector) components. Hence, $A_{i,j}$ refers to an element of $D_1 f(X(t),U(t),p)$, $(\frac{dX(t,p)}{dp})_i$ references the i^{th} element of the sensitivity vector, and $\frac{\partial}{\partial X_m}$ references the partial derivative w.r.t the m^{th} component of $X(t,p)$.

(1) provides a norm on the state tracking error and applied controls across a given trajectory, the mode insertion gradient,

$$\frac{dJ}{d\lambda^+} = \rho(t)^T(f_2(\xi(t)) - f(\xi(t))), \tag{18}$$

based on the adjoint variable,

$$\dot{\rho}(t) = -D_1 l(X(t), U(t))^T - D_1 f(X(t), U(t))^T \rho(t) \tag{19}$$
$$\rho(t_f) = Dm(X(t_f))^T,$$

provides the first-order sensitivity of the cost functional to a switch to a new dynamic mode, $f_2(\xi(t))$, in a neighborhood where the mode switch lasts for an infinitesimal duration (Egerstedt et al. 2006).

More specifically, in this paper $f(\xi(t))$ represents the nominal dynamics of a robotic system, and the mode insertion gradient models the sensitivity of cost (1) to switching to an alternate dynamic mode, $f_2(\xi(t))$, for a duration, $\lambda \in \mathcal{R}^+$, around the time, t, at which (18) is evaluated. The mode insertion gradient only models the cost sensitivity in a neighborhood of where the duration of the switch in dynamics to $f_2(\xi(t))$ is positive and close to zero, $\lambda^+ := \lambda \to 0^+$. While (18) has been used to determine when to insert new dynamic modes in mode scheduling optimization (see Section 2), here we show how the standard trajectory optimization methods introduced in Section 2.1 can be modified to incorporate and minimize an L_2 norm on the mode insertion gradient. Because changes in cost functional directly relate to changes in trajectory, we define this L_2 norm as the measure of a trajectory's hybrid sensitivity and show that (by minimizing the norm during trajectory optimization) it is possible to develop trajectories with reduced sensitivity to potentially sudden and uncertain dynamic effects experienced in crossing hazardous terrain.

While it is possible to append the state vector with (18) to incorporate a norm on the mode insertion gradient into the cost functional, calculating (18) first requires solving for the adjoint variable, $\rho(t)$ by backwards integration of (19). As an alternative to appending (18), a norm on the mode insertion gradient can be incorporated into (1) by first defining an appended state vector, $\overline{X}(t) = [X(t)^T,\ \rho(t)^T]^T$. This choice results in an appended dynamics vector

$$\bar{f}(\overline{X}(t), U(t)) = \begin{pmatrix} f(X(t), U(t)) \\ -Q\left(X(t) - X_d(t)\right) - A(t)^T \rho(t) \end{pmatrix}, \tag{20}$$

and, for appropriately defined weighting matrices, allows use of the trajectory optimization Algorithm 1.

Though $X(t)$ is calculated from forward integration of dynamics (2), $\overline{X}(t)$ is the solution of a two-point boundary value problem (TPBVP), where the first half of (20) requires forward integration of (2) from initial time t_0, and the second half requires backwards integration of (19) from final time t_f. In fact, the system of equations (20) appears identical to the TPBVP resulting from optimal control based on Pontryagin's Maximum Principle or the Hamilton-Jacobi-Bellman equations (see Anderson and Moore (1990)). Note, however, that the two are not equivalent because the system above is simulated from a feedforward control, $\dot{X}(t) = f(X(t), U(t))$, and $\dot{\rho}(t)$ depends on the current open-loop $X(t)$ at each iteration of trajectory optimization, rather than an optimal feedback solution (i.e. (20) is not based on $\dot{X}(t) = f(X(t), -K(t)X(t))$ where $K(t)$ is the optimal feedback regulator from LQR). Though subtle, the difference is important because it implies the first half of (20) can be solved by forward simulation (independently of the second half), after which one can compute the second half of (20) by backward simulation. Thus the problem is much more tractable than the alternative TPBVP in optimal control which requires numerically sensitive shooting-methods for integration or solutions to systems of Riccati equations.

As in the case of parametric sensitivity optimization, appending the state in this form requires new appended weighting matrices, $\overline{Q} \in \mathbb{R}^{2n \times 2n}$ and $\overline{P}_1 \in \mathbb{R}^{2n \times 2n}$, to replace $Q \in \mathbb{R}^{n \times n}$ and $P_1 \in \mathbb{R}^{n \times n}$ from (1). To incorporate a norm on the

mode insertion gradient we define[9]

$$\overline{Q} = \begin{pmatrix} Q & \mathbf{0} \\ \mathbf{0} & w(f_2 - f)(f_2 - f)^T \end{pmatrix}$$

and

$$\overline{P}_1 = \begin{pmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

The w term in the $\overline{Q}$ weighting matrix above is a scalar multiplier used to weight the norm[10] on the sensitivity terms relative to the state tracking error and applied controls. As defined, $\overline{Q}$ provides a quadratic form on the state tracking error and sensitivities determined by the mode insertion gradient. Because the adjoint variable is $\mathbf{0}$ at the final time, the entries of $\overline{P}_1$, which provide the quadratic form on these terms, can be left as 0. Thus, $\overline{P}_1$ only provides a norm on the state tracking error at the final time, as in the case of matrix P_1.

Similar to parametric sensitivity optimization, these modifications allow a new hybrid sensitivity optimization cost functional to be defined according to (9). With the choice of weighting matrices above, the resulting hybrid sensitivity cost functional includes terms identical to the original trajectory optimization cost functional (1) except for the quadratic norm

$$\frac{1}{2}\|\overline{X}(t) - \overline{X}_d(t)\|^2_{\overline{Q}} \;=\; \frac{1}{2}(X(t) - X_d(t))^T\, Q\, (X(t) - X_d(t)) + \frac{w}{2}[\rho(t)^T(f_2 - f)]^2.$$

This norm on the appended state tracking error introduces a term to the integral of (9) that is equivalent to a weighted square of the mode insertion gradient.[11] With this addition, the cost functional incorporates an L_2 norm on the sensitivity of trajectories to discrete changes in dynamics, which defines their hybrid sensitivity.

To calculate the descent directions, $\overline{\zeta}(t) = (\overline{z}(t), v(t))$, for iterative optimization of the hybrid sensitivity version of (9), the constrained minimization problem given by (10) must again be solved subject to (11). In the case of parametric sensitivities, the form of $\overline{g}(\overline{\zeta})$ remained largely the same as $g(\zeta)$ and required only substitution of select terms by their modified, sensitivity appended versions. However, in the hybrid sensitivity case the linear terms differ more between the two models because $\overline{Q}$ now depends on $X(t)$ and $U(t)$. Following the first line of (12), $D\bar{J}(\overline{\xi}) \circ \overline{\zeta}$ is composed of partial derivatives of the incremental and terminal cost with respect to the (appended) state and controls. These partial derivatives are [12]

$$D_1\bar{l}(\overline{X}(t), U(t)) = (X(t) - X_d(t))^T\, Q + w(\rho(t)^T(f_2 - f))\rho(t)^T(A_2(t) - A(t)), \tag{21}$$

$$D\overline{m}(\overline{X}(t_f)) = (X(t_f) - X_d(t_f))^T\, P_1,$$

and

$$D_2\bar{l}(\overline{X}(t), U(t)) = w(\rho(t)^T(f_2 - f))\rho(t)^T(B_2(t) - B(t)) + (U(t) - U_d(t))^T\, R. \tag{22}$$

[9] Note that $f(\xi(t))$ and $f_2(\xi(t))$ are vector fields defined for all state and control pairs and are evaluated along the current trajectory, $\xi(t), t \in [t_0, t_f]$. The dependence of dynamics $f(\xi(t))$ and $f_2(\xi(t))$ on the current iteration's trajectory solution, $\xi(t)$, has been dropped for brevity.

[10] The outer product $(f_2 - f)(f_2 - f)^T$ in $\overline{Q}$ is only guaranteed to be positive semi-definite. Though informally referred to as a norm, the quadratic form on the mode insertion gradient is technically a seminorm (as is the case for state tracking errors in (6)).

[11] The $\overline{X}_d(t)$ terms associated with the adjoint variables in $\overline{X}(t)$ are set to 0 to obtain an L_2 norm on the magnitude of the mode insertion gradient.

[12] The $A(t)$ and $A_2(t)$ terms in (21) represents linearizations of dynamics $f(\xi(t))$ and $f_2(\xi(t))$, respectively, with respect to the un-appended state, $X(t)$. Similarly, $B(t)$ and $B_2(t)$ from (22) represent linearizations with respect to $U(t)$.

To constrain the solutions to (10), linearizations of dynamics (20) with respect to $\overline{X}(t)$ and $U(t)$ are required. Respectively, these linearizations are represented by the $\overline{A}(t)$ and $\overline{B}(t)$ terms in (11), with

$$\overline{A}(t) = \begin{pmatrix} A(t) & \mathbf{0} \\ -Q & -A(t)^T \end{pmatrix}$$

and

$$\overline{B}(t) = \begin{pmatrix} B(t) \\ \mathbf{0} \end{pmatrix}.$$

As will be shown, these linearizations and $\overline{g}(\overline{\zeta})$ enable the iterative computation of optimal trajectories balancing sensitivity to sudden changes from dynamics $f(\xi(t))$ to $f_2(\xi(t))$, relative to state tracking error and control magnitude.

4. Example Systems

This section includes three examples that compare feedforward trajectories produced by standard trajectory optimization methods (i.e. following Algorithm 1) to feedforward trajectories produced using the same optimization algorithm with appended versions of the state, linearizations, and weighting matrices derived in Sections 3.1 and 3.2. Although the optimal control algorithm generates LQ feedback controllers, these controllers assume full state feedback exists and yield tracking performance and robustness properties that are highly dependent on LQ parameters and which vary as a function of trajectory. Rather than comparing performance under the variety of possible feedback conditions, we emphasize that better feedforward solutions diverge less and reduce reliance on feedback.[13] The following examples are intended to show that sensitivity optimization generates open-loop solutions that are more reliable in the sense that robots following these trajectories stay closer to their planned course when actual conditions do not match nominal conditions from planning.

The three scenarios involve trajectory planning for different nonlinear robotic vehicle models. First, example 4.1 demonstrates feasibility of the proposed parametric sensitivity optimization methods on a model simulating a real robotic vehicle with complicated, variable inertial dynamics. Results show parametric sensitivity optimization plans trajectories that track a goal while mitigating the effect of uncertainty in modeled friction. Examples 4.2 and 4.3 are based on more common and well-understood vehicle models to facilitate analysis and reproduction of results. These sections show hybrid sensitivity optimization adjusting trajectories to safely pass through hazardous regions and around obstacles that impact systems in ways which cannot be avoided. The hybrid sensitivity examples demonstrate that even approximate hazard models (e.g. modeling ice as low friction and sand as reducing control authority) provide sufficient abstractions for planners to develop intuitive hazard mitigation strategies. To better illustrate sensitivity optimized trajectories, Extension 1 includes video simulations of robotic vehicles using sensitivity optimized controllers in uncertain environments with model conditions and mobility hazards corresponding to examples.

4.1. Example I: Minimal Parametric Sensitivity Trajectories

We first compare the performance of standard and parametric sensitivity optimization approaches (see Sections 2.1 and 3.1) in planning trajectories insensitive to parametric variations for a robotic variable inertial vehicle (VIV) modeling a real (physical) robot described in Meirion-Griffith et al. (2014), Nie et al. (2009), and Ruffatto III et al. (2014). The robotic vehicle considered here is parallel steered with four wheels modeled in point contact with the ground subject to viscous friction. The model includes a controllable mass that can be accelerated between its front and back end along the vehicle's

[13] For evidence that improved feedforward trajectories from sensitivity optimization can be tracking under low bandwidth, full state, and partial state LQ feedback conditions see Ansari and Murphey (2013*a*)

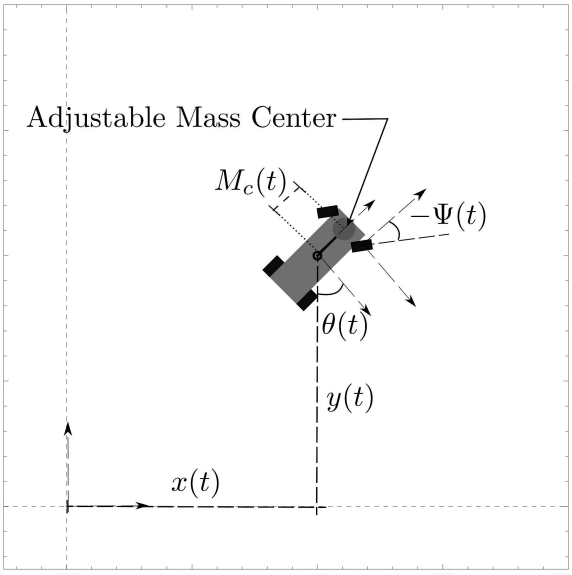

Fig. 1. Dynamic vehicle model with configuration variables

central axis. Adjusting the placement of this mass modifies the vehicle's overall inertia and the normal forces at the tire contacts as it drives.

For this VIV example, the vehicle controls are defined as the forward/backward thrust developed by the tires, $F(t)$, a kinematic control for the steering angle, $\Psi(t)$, and the acceleration of the adjustable mass in one dimension between its front and rear, $\nu(t)$. These are represented by the first three elements of the control vector, $U(t)$, respectively. The state space for the model, reflected in Fig. 1, consists of its $(x(t), y(t))$ location and heading angle, $\theta(t)$, provided relative to a fixed world frame, the displacement of the mass center relative to the vehicle's geometry center, $M_c(t)$, and the derivatives of these states. The state space vector, $X(t)$, and control vector, $U(t)$, are

$$X(t) = [x(t),\ y(t),\ \theta(t),\ M_c(t),\ \dot{x}(t),\ \dot{y}(t),\ \dot{\theta}(t),\ \dot{M}_c(t)]^T$$

and

$$U(t) = [F(t),\ \Psi(t),\ \nu(t)]^T.$$

The dynamic equations of motion are derived using a Lagrangian formulation and solving the forced Euler Lagrange (EL) equations, including the viscous friction forces. The equations are too unwieldy to be included but can be calculated using standard symbolic processing software (in this case *Mathematica*).

In computing optimal trajectories for the vehicle model described, all simulations and figures presented use matrices[14]

$$Q = \mathrm{diag}[15,\ 15,\ 100,\ 45,\ \frac{1}{2},\ \frac{1}{2},\ \frac{1}{10},\ 1], \tag{23}$$

$$R = \mathrm{diag}[\frac{1}{4000},\ \frac{900}{\pi},\ 1], \tag{24}$$

[14] Though the weights seem specific, we selected values for tracking simple trajectories and then re-scaled the values to achieve desired performance on desired trajectories with differing degrees of difficulty. In the numerical simulations described we found a number of different parameter combinations yield similar performance in tracking trajectories of similar difficulty.

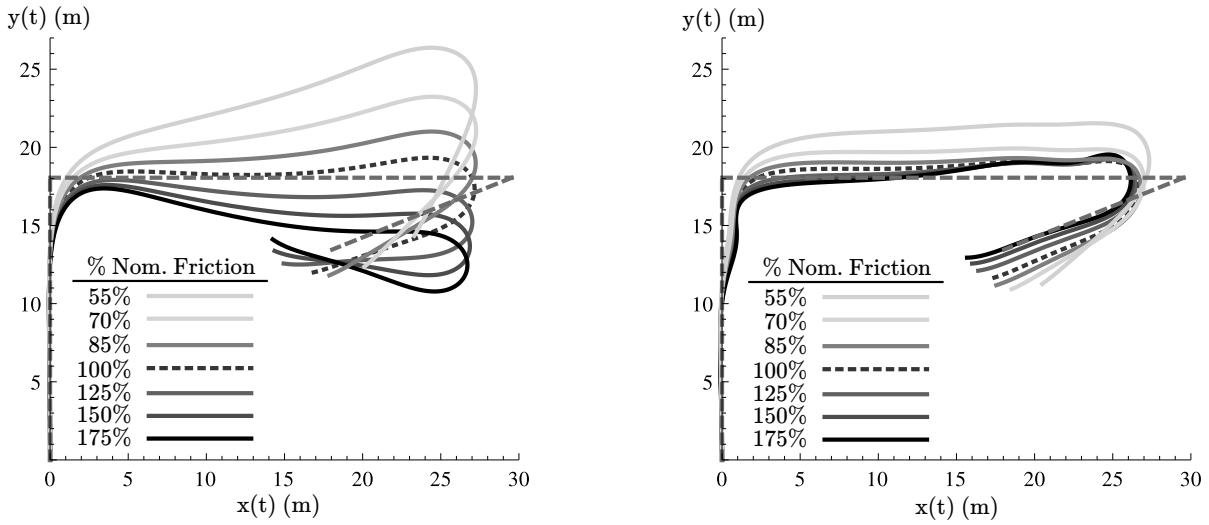

Fig. 2. Standard state based trajectory optimization (2a) and parametric sensitivity (2b) optimization techniques tracking a desired trajectory (long dashing) using feedforward control. The curves reflect trajectories that would result if actual friction were 55%, 70%, 85%, 100%, 125%, 150%, and 175% of the nominal modeled value of 0.7.

and

$$P_1 = \text{diag}[\frac{5}{2},\ \frac{5}{2},\ 50,\ 25,\ \frac{5}{2},\ \frac{5}{2},\ \frac{1}{4},\ 2] \tag{25}$$

to provide quadratic norms on the state error and controls (according to (6) and (7)) in the cost functional (1) for the standard optimization. The parametric sensitivity optimization approach uses the same R matrix but appends the lower right diagonal of the Q matrix with an 8×8 identity matrix to provide the norm on the 8 additional sensitivity terms in the state vector. For this example, the lower right diagonal of matrix P_1 above is also appended with an 8×8 zero matrix to ignore the parametric sensitivity cost at the final time. For the purposes of analysis, both standard and parametric sensitivity optimization results run for 30 iterations (though solutions converged to those depicted in Fig. 2 within ≈ 10 iterations in both cases). Standard optimization is initialized with state, x_0, specified as zero except for the heading angle of $\frac{\pi}{2}$rad, and an initial control applying $F(0) = 240N$. Sensitivity optimization uses the trajectory from standard optimization as a seed.[15]

The short dashed curves in Fig. 2 show the optimal trajectories resulting when the feedforward controllers form standard state based trajectory optimization (Fig. 2a) and parametric sensitivity optimization (Fig. 2b) are applied to track the desired trajectory in long dashing. Under nominal (simulation) conditions where friction matches the model value of 0.7, the controllers drive the VIV so that its center of geometry perfectly tracks these short dashed curves. With respect to objective performance (1), the optimal trajectory from standard state based optimization yields $J = 1,198$, divided into a control cost of 216, state tracking cost of 387, and parametric sensitivity cost of 595. In comparison, the optimal trajectory from sensitivity optimization achieves a lower cost of $J = 713$, with a control cost of 250, state tracking cost of 414, and parametric trajectory sensitivity cost of 49. Hence, by allowing minor sacrifices in control effort (16% increase) and tracking of the desired trajectory (7% cost increase), parametric sensitivity optimization produces an optimal trajectory

[15] In comparison to the later examples, which require seconds / minutes to compute, both standard and sensitivity optimization for the VIV can take more than an hour on a laptop using interpreted *Mathematica* code. In our experience this is common for systems of similar state dimension (16 appended states). For physical implementation, we typically switch to the (publicly available) compiled trajectory optimization software, *trep* (see Johnson and Murphey (2009)), when the state dimension is greater than 10 to 15. The software uses the methods in Section 2.1 to optimize trajectories for systems with much larger state in real time.

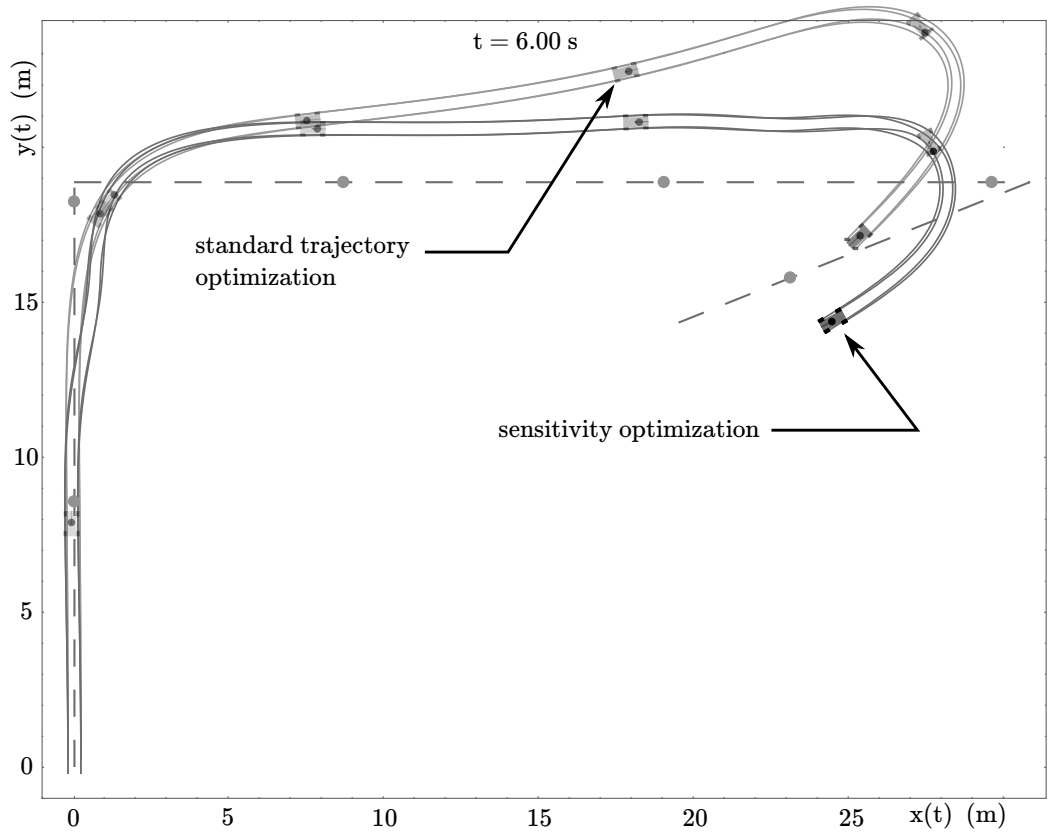

Fig. 3. Time-lapsed trajectories resulting when the feedforward controllers from standard and parametric sensitivity optimization track the desired trajectory (indicated by long dashing) under conditions where friction is 70% of the model value (see Fig. 2). The figure shows each vehicle's state along with the desired position center of geometry (circles along the dashed curve) at 1 s intervals starting at $t = 1$ s and ending at $t = 6$ s. The sensitivity optimized car stays closer to the desired trajectory. Trajectory optimization provides solutions that push the center of mass (black circle) forward to make the sharp turns along the desired trajectory (easiest to see in the case of the sensitivity optimized car at location $(x(5), y(5)) \approx (25m, 20m)$). However, the sensitivity optimization solution uses the center of mass to a greater extent in making the first turn (see Fig. 4). Extension 1 includes a video of the robotic vehicle driving in the situation described.

that is less sensitive (92% sensitivity reduction) and can be followed with reduced L_2 error in varied frictional conditions. As evidence, Fig. 2 shows when friction is varied from the nominal model value of 0.7, sensitivity optimized controllers result in more reliable trajectories that are tightly grouped and diverge less from the nominal than their more sensitive counterparts. Unlike adaptive control strategies, the improved tracking performance is truly open loop, achieved without the added sensing requirements needed for parameter estimation.

Figure 3 provides a time lapse (see also Extension 1 for a corresponding video) of the robotic vehicle's trajectory based on the sensitivity optimization controller and the standard sensitivity optimization controller. The vehicles are depicted driving under conditions that do not match model conditions (driving with 70% of nominal expected friction). Figure 4 shows the sensitivity optimized controller pushes the vehicle center of mass farther forward and steers more to help make the first turn. The strategy results in significantly less divergence from the planned path in spite of the unexpected driving conditions.

4.2. Example II: Minimal Hybrid Sensitivity Trajectories

The first hybrid sensitivity example simulates a robotic vehicle following a desired path that passes in the vicinity of a sand pit. Assume, through sensor measurements, the robot has sufficient information to generate a $2D$ Gaussian modeling

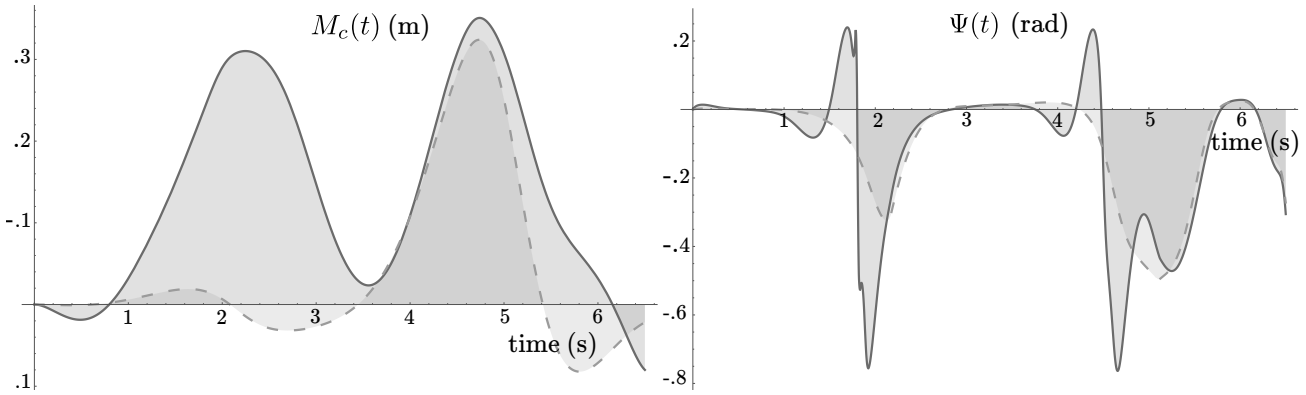

Fig. 4. Comparison of the (controlled) position of the VIV center of mass and steering angle based on the optimal feedforward controller from parametric sensitivity optimization (solid) and the standard trajectory optimization (dashed). The sensitivity optimized controller steers more and moves the center of mass much farther forward to help make the first turn (see Fig. 3).

the expectation of encountering sand as a function of (x, y) location in state space. Additionally, assume the robot has an approximate model of its dynamics when driving over sand, f_{sand}. The following section combines these two models to develop a dynamic model, f_2, for hybrid sensitivity optimization that approximates the vehicle's dynamics through the state space. Using the Gaussian expectation, f_2 transitions between nominal dynamics, f, in areas where the robot does not expect to encounter sand, and f_{sand} in areas where sand is expected (see Fig. 5). Hence, the Gaussian acts as a scaling factor that increases the difference $f_2 - f$ in (18), magnifying the weight on hybrid sensitivities when the robot's trajectory nears the expected location of sand. We now show how sensitivity-enhanced trajectory optimization automatically provides trajectories that avoid uncertain, hazardous areas (the sand) when there is no way to modify controls to mitigate the potential changes in dynamics that may be generated in crossing these regions.

To simplify interpretation of results, hybrid sensitivity examples utilize specialized versions of the robotic vehicle model presented earlier. For the scenario just described, the vehicle dynamics are simplified to that of a kinematic model. The state-space vector of the robot, $X(t) = [x(t),\ y(t),\ \theta(t)]^T$, represents its $2D$ location and orientation. The control vector, $U(t) = [v(t),\ \omega(t)]^T$, encompasses both forward and angular velocity controls. The robot's dynamics are

$$f(X(t), U(t)) = \begin{pmatrix} v(t)cos(\theta(t)) \\ v(t)sin(\theta(t)) \\ \omega(t) \end{pmatrix}. \tag{26}$$

Assuming that getting caught in sand will have a near uniform slowing effect on the robot's movements, the dynamics of the robot when in sand are approximately modeled as being scaled down by

$$f_{sand}(X(t), U(t)) = \beta \cdot f(X(t), U(t)). \tag{27}$$

Note that the sensitivity optimization methods described are insensitive to specific choice of scalar parameter, $\beta \in (0, 1)$, in the dynamics used to model these hazardous regions as long as the relative dynamic effects are captured. In this case, the applied factor $\beta = \frac{1}{10}$ produces similar trajectories compared to other fractional multiples and affects the optimization in the same manner as choosing an alternative weight, w, in matrix $\overline{Q}$. Combining models (26) and (27), the expected influence of the sand throughout the state space is provided by dynamics,

$$f_2(X(t), U(t)) = f\,(1 - 0.9\, e^{(\frac{-(x(t)-\mu_x)^2}{2\sigma_x^2} - \frac{(y(t)-\mu_y)^2}{2\sigma_y^2})}), \tag{28}$$

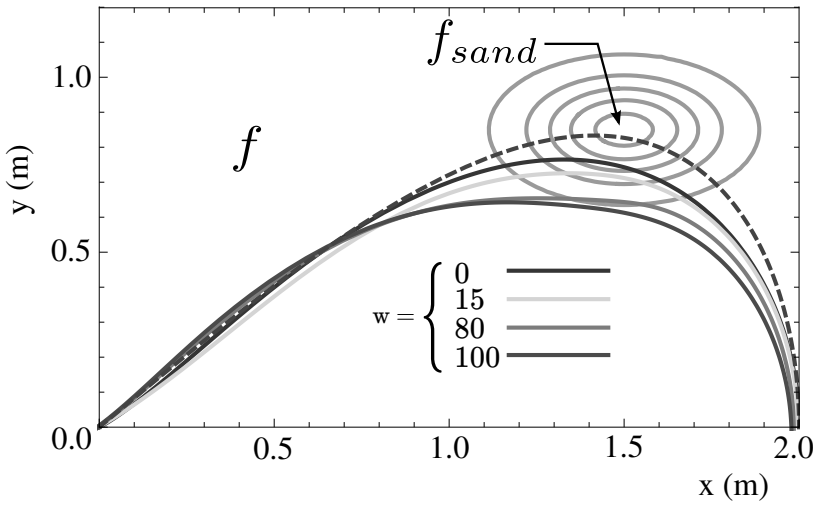

Fig. 5. State trajectories resulting from feedforward controllers produced by standard (the curve with weight, $w = 0$, in the sensitivity norm) and sensitivity optimization techniques. The dashed curve reflects the desired trajectory, and the contours depict level sets of the Gaussian representing the expected influence of the sand pit dynamics in f_2. Dynamics f_2 transition between f_{sand} at the center of the Gaussian and f elsewhere. As the weight on sensitivity optimization is increased, trajectories apply greater degrees of obstacle avoidance around the sand.

based on the Gaussian distribution (centered at $(\mu_x, \mu_y) = (1.5m, 0.85m)$ with standard deviations in x and y directions of $(\sigma_x, \sigma_y) = (0.18m, 0.1m)$), which represents the expectation of encountering sand. As mentioned, the Gaussian effectively scales the hybrid sensitivity norm in transitioning between nominal dynamics (26) and sand dynamics (27) as the system approaches the hazard. [16]

While standard trajectory optimization methods can plan feedforward trajectories using dynamic models (similar to f_2), which represent the dynamics in hazardous regions, such trajectories are sensitive to model inaccuracies. In practice, accurate terrain models are difficult to obtain (especially from on-line measurement), so the sensitivity optimization approach applied here requires only models (i.e. (27)) that approximate which components of the robot's dynamic vector fields are influenced by hazards. Hybrid sensitivity optimization provides ways to adjust path and control to minimize sensitivity to these components. At times this means avoiding these poorly understood (due to model uncertainty) hazard regions; and, as the subsequent example will show, at times there are ways to pass through these regions in ways that leave the robot unaffected by their potential for influence.

Results in this section (see Fig. 5) are based on weight matrices $Q = \mathrm{diag}[1,\ 1,\ 0.0001]$, $R = \mathrm{diag}[0.01,\ 0.01]$, and $P_1 = \mathrm{diag}[100,\ 100,\ 100]$. The standard trajectory optimization solution (the curve of $w = 0$ in Fig. 5) is initialized with $x_0 = [0,\ 0,\ \frac{\pi}{7}\ \mathrm{rad}]$ and requires 3 iterations to converge ($\approx 2s$ on a laptop running *Mathematica*). Sensitivity optimized trajectories are calculated for different choices of $w = 15, 80$, and 100 in $\overline{Q}$. In each case, the standard optimization results are used as a seed and sensitivity optimization solutions require an additional $1 - 2$ iterations ($\approx 2s$) to converge. As in the previous case, all trajectories in Fig. 5 correspond to feedforward optimal solutions from trajectory optimization.

Based on the kinematic vehicle model (28), the robot in this example cannot take advantage of inertial effects to build momentum and slide over the hazard. Because entering the sand will definitely affect trajectory, avoidance is the only means to reduce the impact of and trajectory uncertainty generated by the hazard. Results included in Fig. 5, demonstrate trajectories applying varying degrees of obstacle avoidance to minimize hybrid sensitivities in regions around the sand.

[16] As the subsequent example shows, the choice of a Gaussian to represent the robot's expectation is unimportant and a wide range of (even non-smooth) functions can be used to represent the expected influence of the hazard.

Using dynamics of vehicle and hazard, the approach automatically determines that avoidance is optimal and produces a natural response to the obstacle as discussed in Section 1.

To measure performance of the trajectories in Fig. 5, we use a quadratic cost functional with the weight matrices described and $w = 100$ in $\overline{Q}$. Standard state based optimization yields a trajectory cost of 6.4E^{-2} while hybrid sensitivity optimization achieves costs of 6.3E^{-2}, 5.2E^{-2}, and 5.4E^{-2} for solutions derived from sensitivity weights $w = 15, 80,$ and 100, respectively. In each case the control effort remains approximately the same (sensitivity optimization reduces the control cost around 4%). However, as the figure illustrates, increasing the weight on the hybrid sensitivity norm causes the vehicle to make sacrifices in trajectory tracking (tracking cost increases from 1.99E^{-3} for standard optimization to 1.05E^{-2} for sensitivity optimization with $w = 100$) in order to give the sand pit a wider birth (reducing sensitivity cost from 1.66E^{-2} for standard optimization to 1.02^{-4} for $w = 100$). A similar effect can be achieved by increasing the standard deviation or shape of the Gaussian distribution, which encodes the expected influence of the hazard through (28). Based on the shape and dynamics of these regions, hybrid sensitivity optimization optimally adjusts controls and applies varying levels of obstacle avoidance to compute reliable and often intuitive trajectories through or around obstacles.

4.3. Example III: Minimal Hybrid Sensitivity Trajectories

The second hybrid sensitivity example demonstrates a (dynamically) more complicated version of the robotic vehicle than in the previous case tracking an infeasible desired trajectory at high speed. The desired trajectory brings the vehicle (a nonlinear dynamic skidding car model) in the vicinity of an icy region, after which it takes a 60° turn while maintaining a constant forward velocity of $s = 27\frac{m}{s}$ ($\approx 60mph$). Again, we assume the robot has collected sufficient measurements to roughly localize the icy region. However, in this example the expected location of the hazard is modeled using a non-smooth (piecewise) box function, $box(x, y)$, rather than a Gaussian (see Fig. 8). The box function evaluates to $box(x, y) = 1$ in areas where the robot expects to encounter ice and $box(x, y) = 0$ otherwise. We use hybrid sensitivity optimization to develop vehicle trajectories that tracks the desired trajectory with reduced sensitivity to slipping on ice.

The robot's dynamics in this example correspond to a standard front-steered vehicle model discussed in LaValle (2006) and used for motion planning in Pepy et al. (2006). The state vector includes the vehicle's $2D$ position $(x(t), y(t))$, orientation $\theta(t)$, its yaw rate, $\dot{\theta}(t)$, and its lateral skidding velocity, $v_y(t)$, so that

$$X(t) = [x(t),\ y(t),\ \theta(t),\ \dot{\theta}(t),\ v_y(t)]^T.$$

The robot only has control over the front steering angle, $U(t) = [\Psi(t)]$, and its forward speed is assumed to be constant. The vehicle's configuration matches that of the VIV in Fig. 1 except that it does not include a movable center of mass (no $M_c(t)$ variable) and $(x(t),\ y(t),\ \theta(t))$ provide the location and orientation of a (right-handed) body frame at the center of mass rather than the center of geometry. Figure 8 shows the vehicle with its center of mass frame. The axis orthogonal the vehicle's heading indicates the direction of lateral skidding velocity $v_y(t)$.

The vehicle model parameters, which match those in LaValle (2006), are designed for automotive applications. These parameters include the constant forward velocity of $s = 27\frac{m}{s}$, mass of $m = 1,460kg$, distances from center of mass to front and rear axles of $a = 1.2m$ and $b = 1.5m$, front and rear cornering stiffness of $c_f = 17,000\frac{N}{rad}$ and $c_r = 20,000\frac{N}{rad}$, and inertia of $I_z = 2,170\frac{kg}{m^2}$. The vehicle dynamics,

$$f(X(t), U(t)) = \begin{pmatrix} s\,cos(\theta(t)) - v_y(t)\,sin(\theta(t)) \\ s\,sin(\theta(t)) + v_y(t)\,cos(\theta(t)) \\ \dot{\theta}(t) \\ \big(af_f\,cos(\Psi(t)) - bf_r\big)/I_z \\ \big(f_f\,cos(\Psi(t)) + f_r\big)/m - s\,\dot{\theta}(t) \end{pmatrix}, \tag{29}$$

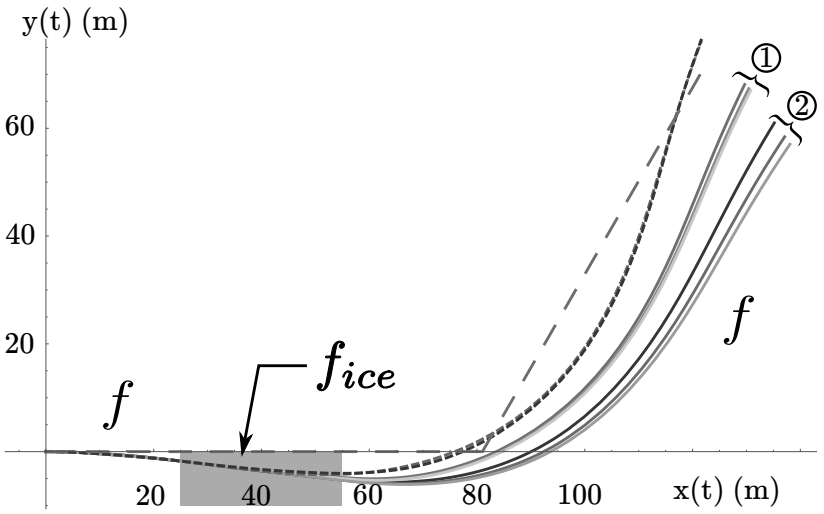

Fig. 6. Standard and hybrid sensitivity optimization ($w = 1$) produce the same $2D$ optimal feedforward trajectory for the robot's center of mass (short dashed curve) under nominal conditions (without ice). The two sets (marked as (1) and (2)) of successively lighter solid curves are simulations of tracking results that would occur if the ice patch encountered exactly matched the anticipated dynamics, f_2, (the darkest curve in each set); and if dynamics, f_2, were incorrect and the actual ice produced $\frac{1}{10}$ or $\frac{1}{20}$ the expected friction (the two successively lighter curves in each set). The three hybrid sensitivity results in set (1) (the curves are tightly grouped and so may look like two curves), stay closer to the planned (nominal short dashed) trajectory while the standard optimization results in set (2) are both farther away and diverge (relative to each other) more as friction reduces. The long dashed curve indicates the desired trajectory.

depend on front and rear lateral tire forces,

$$f_f = -c_f \left(tan^{-1}\big([v_y(t) + a\,\dot{\theta}]/s\big) - \Psi(t) \right)$$

and

$$f_r = -c_r\, tan^{-1}\big([v_y(t) - b\,\dot{\theta}]/s\big).$$

Based on the assumption that ice will reduce tire friction forces, the ice dynamics, f_{ice}, are equivalent to (29) except that forces f_f and f_r are scaled by a factor, $\beta \in (0, 1)$. We (arbitrarily) choose $\beta = \frac{1}{5}$. For the reasons described in the previous example, the algorithm is insensitive to the particular choice of β. Instead, hybrid sensitivity optimization searches for trajectories that are not affected by differences in tire forces that can be generated by ice.

With the dynamics f and f_{ice} defined and the box function, $box(x, y)$, to indicate the region where ice is expected, we define the model,

$$f_2(X(t), U(t)) = box(x(t), y(t))\,(f_{ice} - f) + f,$$

to approximate the influence of the ice across the state space. This choice of f_2 is equal to the nominal dynamics (29) when the robot is away from the ice and equal to f_{ice} when the robot is over the region expected to contain the ice. Hence, sensitivity optimization will only include a norm on hybrid sensitivity terms in (9) in areas where trajectories pass through the region expected to contain ice.

As a note, the optimization methods in Sec. 2.1 are derived under the assumption that the dynamics and incremental cost are smooth; however, in practice they work well even when this is not the case. Here, the use of a non-smooth box function highlights that the proposed methods work in spite of piecewise discontinuities in the incremental cost. As one drawback, non-smooth systems may not be able to take advantage of variable time-step integration because discontinuous changes in

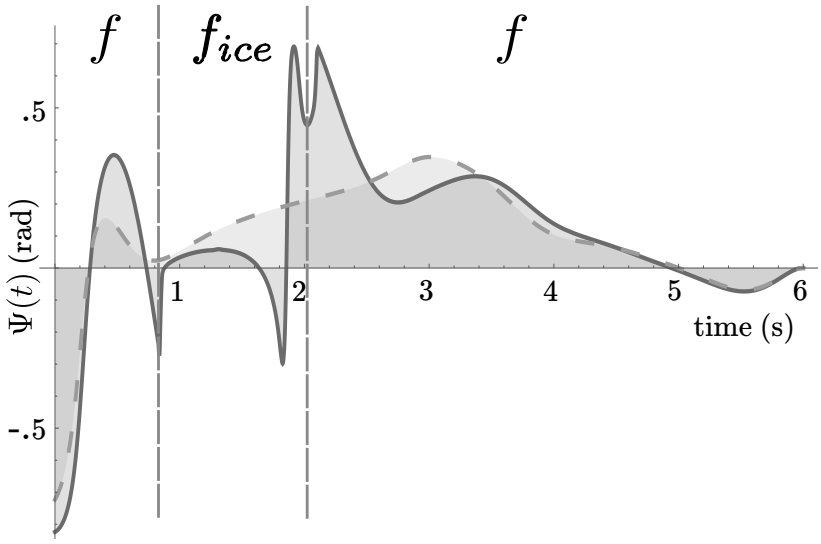

Fig. 7. The feedforward control (front steering angle) for standard (dashed) and hybrid sensitivity optimization (solid). The dashed lines indicate when the robot enters and exits the icy region. Sensitivity optimized trajectories ($w = 1$) steer to a greater degree immediately before and after crossing the ice in order to avoid steering in the (dynamically) uncertain area over the ice.

integrated vector fields can cause controlled error stepping methods to fail (see Hairer et al. (2009)). In implementation, we address the issue by switching to fixed time-step integration when variable time stepping methods fail. Alternatively, piecewise functions can be interpolated using cubic splines or any other method that provides smooth interpolation.

The results in Figs. 6-8 compare feedforward optimal trajectories produced by standard trajectory optimization and hybrid sensitivity optimization. In each case, we restrict convergence criteria to assess solutions after a fixed number of iterations. We perform 30 iterations of standard trajectory optimization with $Q = \text{diag}[10,\ 10,\ 10,\ 1,\ 0.001]$, $R = [100]$, and $P_1 = \mathbf{0}$. With the standard optimization solution as a seed, hybrid sensitivity results are based on 30 iterations of sensitivity optimization using (29) and f_2 as described, with $w = 1$ in $\overline{Q}$.[17] In both cases (standard and hybrid sensitivity optimization) trajectories converged to the those depicted within ≈ 6 iterations.

Figure 6 compares feedforward optimal trajectories produced by standard trajectory optimization and hybrid sensitivity optimization under varying simulated ice conditions. Under nominal conditions, standard optimization produces the short dashed trajectory with a cost of $J = 15,867$, based on a quadratic cost functional with the weight matrices above and $w = 1$. Using the same cost functional, sensitivity optimization provides a solution with $J = 1,017$, which corresponds to the same short dashed trajectory as standard optimization. That is, by adjusting controls (increasing control cost from 12 for standard optimization to 21), sensitivity optimization modifies orientation, yaw rate, and lateral skidding velocity (slightly increasing the overall tracking cost from 787 to 855) to reduce sensitivity from $15,068$ to 141, while driving the vehicle along a $2D$ trajectory that is indistinguishable from that of standard optimization under modeled conditions. Hence, sensitivity optimization tracks the $2D$ desired trajectory (long dashing), equally well under nominal conditions. However, the nominal trajectory brings the robot into a region (shaded area) expected to contain ice. The figure shows how the $2D$ trajectory of the robot's center of mass would be affected if it applied the optimal feedforward control from standard (the three curves in set (2)) and hybrid sensitivity optimization (the three curves in set (1)) and experienced the ice under three different model conditions. The darkest curve in each set represents the trajectory which would result if the robot experienced ice matching the approximated ice model, f_2. The lighter two curves show how trajectories would be

[17] In *Mathematica* each iteration requires $\approx 30s$ and runs at $\approx 60\times$ real time. In complied implementations we typically achieve $2-3$ orders of magnitude speed improvement, therefore we expect these results can be implemented in real time.

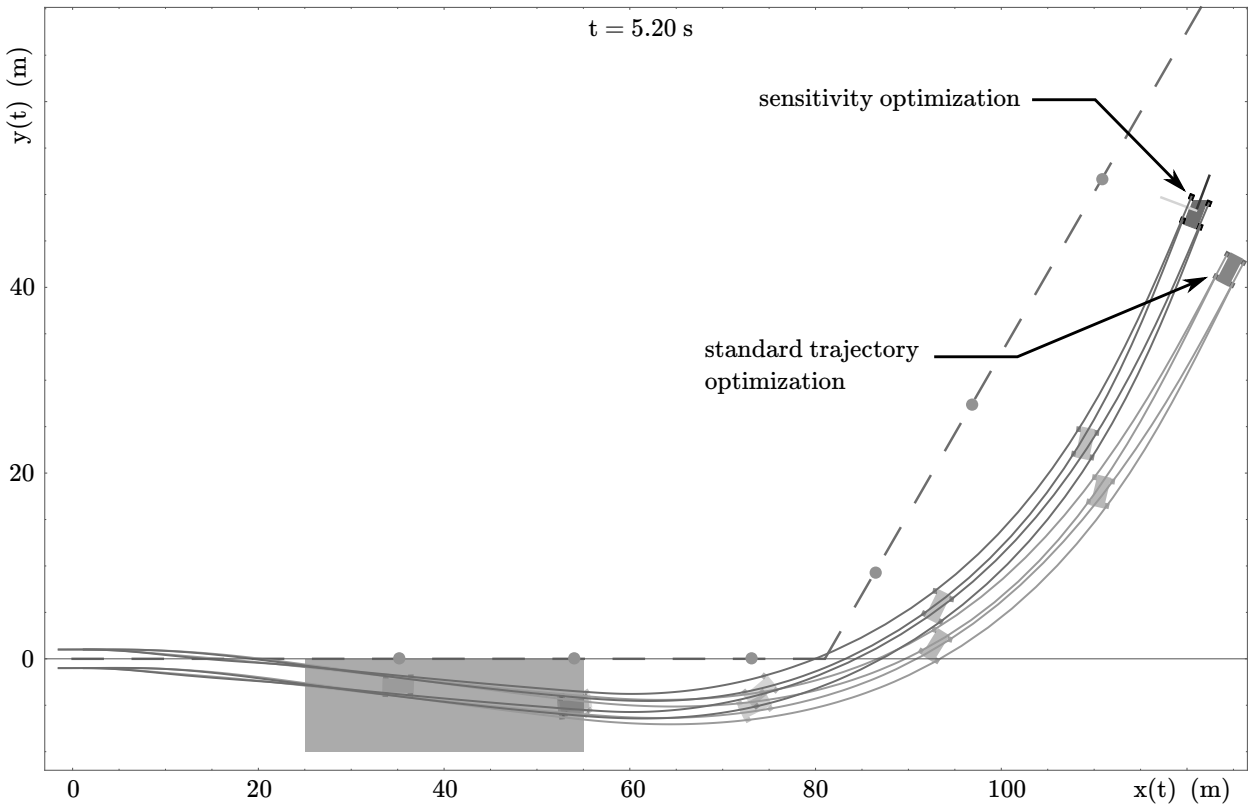

Fig. 8. Standard and hybrid sensitivity optimization tracking a desired trajectory (dashed curve). The cars pass through an icy region, which is simulated according to the model, f_2. The figure depicts the state of both vehicles at $t = 1.3, 2.0, 2.7, 3.4, 4.2$, and 5.2 seconds along their respective trajectories and indicates the desired vehicle position at each time (circles along the dashed curve). Under nominal conditions (without ice) standard and hybrid sensitivity optimization result in the same $2D$ feedforward vehicle trajectory corresponding to the short dashed curve in Fig. 6. However, the sensitivity optimized car avoids steering over the ice and so its trajectory is less affected by the simulated icy conditions in the scenario above. Extension 1 includes a video of the robot's motion corresponding to the time-lapsed trajectory shown here and several others derived for the same vehicle model but for different mobility hazards (in different locations and with different model dynamics corresponding to both sand and ice hybrid sensitivity driving scenarios).

affected if the ice experienced were $2\times$ and $4\times$ more slippery than expected (the robot experiences $\frac{1}{10}$ and $\frac{1}{20}$ the nominal tire forces, f_f and f_r, when crossing the ice).

In all cases in Fig. 6, sensitivity optimized trajectories stay closer to the nominal planned trajectory (short dashed curve). Not only do the standard optimization trajectories diverge more from the planned trajectory due to the ice, but they also diverge more relative to one another.[18]. Without feedback, and even in these conditions where the ice model is inaccurate, sensitivity optimized trajectories do better job tracking the desired trajectory. As Fig. 7 shows, these trajectories manage to avoid sensitivity to the ice (without veering around it) using one of the strategies discussed in Section 1. The sensitivity optimized steering controls (solid) are greater in magnitude both before and after the ice in order to avoid steering when over the ice (region between the dashed vertical lines). There are spikes in steering control effort at the borders where the vehicle enters and, in particular, where it leaves the icy region. This is due to the fact that the desired trajectory is highly aggressive and the relatively large ice region requires that the robot apply some control over the ice or it significantly overshoots the turn. By shrinking the ice or by increasing the norm on hybrid sensitivities relative to state tracking errors, it is possible to further reduce sensitivity to ice and develop trajectories that diverge less (or not at all) under its influence. Figure 8 (see

[18] The three sensitivity optimized trajectories stay close enough together that it is hard to distinguish them in the figure.

also the corresponding video in Extension 1) shows how the trajectory of a robotic vehicle applying a control based on standard trajectory optimization would diverge from a robot applying controls (Fig. 7) from sensitivity optimization when the ice matches the conditions modeled in f_2.

The results in this section are notable considering traditional obstacle avoidance techniques (e.g. incorporating state weights or using barrier functions with dynamic planners assuming rigid obstacles) would result in poor trajectory tracking performance in circumventing the icy region altogether. In this case, hybrid sensitivity optimization generates a plausible strategy (reducing steering over the ice) to minimize the impact of the ice on trajectory. As mentioned, even in the hypothetical "worst-case" scenario where the vehicle performed sensitivity optimization unnecessarily and never actually experienced the ice, its trajectory (nominal short dashed curve in Fig. 6) tracks the desired trajectory approximately as well as the standard optimization results. While similar trajectories may be achieved by manually adding weights in (1) that increase the cost of applying steering controls over ice, we highlight that hybrid sensitivity optimization automatically generates this strategy. The approach determines when it is best to avoid hazards and when (and how) controls can be adapted to pass through hazardous areas in a manner that mitigates their dynamically uncertain potential to influence trajectory.

As a final note, there are a number of existing methods to incorporate unilateral state constraints in trajectory optimization (discussed in Section 2). The methods presented here can easily be combined with these techniques to restrict optimization to only consider solutions that pass through valid regions of state space. In these examples, for instance, such methods can be used to bound vehicle trajectories (from both standard and sensitivity optimization) in order to ensure the robot stays within the confines of the road. Additionally, using Sums of Squares techniques as described in Tedrake et al. (2010), one can numerically generate conservative approximations of the regions of attraction surrounding sensitivity optimized trajectories and thus guarantee stability and convergence of closed-loop (LQR) tracking.

5. Conclusions

This paper contributes new continuous parametric and hybrid sensitivity measures and derives an approach to nonlinear sensitivity optimization that enables existing trajectory optimization methods to plan trajectories in uncertain environments. Minimizing parametric model sensitivities, we provide examples that show reliable dynamic trajectories can be planned that are insensitive to uncertainty in model parameters. Further, by minimizing hybrid sensitivities we prove that the planner can reduce trajectory uncertainty generated in crossing uncertain terrain.

In contrast to most existing planning techniques, these methods are derived in a general form. They work directly with continuous state and control spaces and do not require explicit discretization of any kind. The approach accounts for nonlinear, time-varying system dynamics and dynamics associated with hazards and obstacles in the environment. Also, where most planners only consider rigid obstacles that must be avoided, these methods apply to broader classes of obstacles and mobility factors. The derived approach to trajectory planning is unique in that it automatically and optimally determines when to avoid obstacles and when control should be used to traverse hazards with reduced uncertainty. As examples demonstrate, intuitive control policies can be developed that drive systems around/through hazardous regions using only first-order sensitivities and coarse approximations of the uncertain dynamics imposed by hazards. The approach therefore provides an efficient alternative to the sampling based techniques used to propagate stochastic uncertainties in other nonlinear, uncertainty-mitigating planners.

For future work we intend to investigate whether conditions exist under which first-order sensitivities provide insufficient information for reduction of uncertainty. We will assess higher order sensitivities for uncertainty minimization. Finally, we intend to explore stochastic and Gaussian process modeling to determine if first-order sensitivity minimization can formally be considered a special case of uncertainty minimization.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant IIS 1018167. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Anderson, B. D. O. and Moore, J. B. (1990). *Optimal control: linear quadratic methods*. Prentice-Hall, Inc.. Upper Saddle River, NJ, USA.

Ansari, A. and Murphey, T. (2013*a*). Minimal parametric sensitivity trajectories for nonlinear systems. In *American Control Conference (ACC)*. pp. 5011–5016.

Ansari, A. and Murphey, T. (2013*b*). Minimal sensitivity control for hybrid environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 3023–3028.

Banerjee, A., Pomerance, A., Losert, W. and Gupta, S. (2010). Developing a stochastic dynamic programming framework for optical tweezer-based automated particle transport operations. *IEEE Transactions on Automation Science and Engineering* 7(2): 218–227.

Becker, A. and Bretl, T. (2012). Approximate steering of a unicycle under bounded model perturbation using ensemble control. *IEEE Transactions on Robotics* 28(3): 580 –591.

Becker, A., Onyuksel, C., Bretl, T. and McLurkin, J. (2014). Controlling many differential-drive robots with uniform control inputs. *The International Journal of Robotics Research* 33(13): 1626–1644.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Byrne, P. and Burke, M. (1976). Optimization with trajectory sensitivity considerations. *IEEE Transactions on Automatic Control* 21(2): 282 – 283.

Candido, S. and Hutchinson, S. (2010). Minimum uncertainty robot path planning using a POMDP approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 1408–1413.

Candido, S. and Hutchinson, S. (2011). Minimum uncertainty robot navigation using information-guided POMDP planning. In *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 6102–6108.

Chen, H., Ma, M.-M., Wang, H., Liu, Z.-Y. and Cai, Z.-X. (2009). Moving horizon H_∞ tracking control of wheeled mobile robots with actuator saturation. *IEEE Transactions On Control Systems Technology* 17(2): 449 –457.

Chudoung, J. and Beck, C. (2001). The minimum principle for deterministic impulsive control systems. In *IEEE Conference on Decision and Control*. Vol. 4. pp. 3569–3574.

Du Toit, N. and Burdick, J. (2012). Robot motion planning in dynamic, uncertain environments. *IEEE Transactions on Robotics* 28(1): 101–115.

Egerstedt, M., Wardi, Y. and Axelsson, H. (2003). Optimal control of switching times in hybrid systems. In *IEEE International Conference on Methods and Models in Automation and Robotics*.

Egerstedt, M., Wardi, Y. and Axelsson, H. (2006). Transition-time optimization for switched-mode dynamical systems. *IEEE Transactions on Automatic Control* 51(1): 110–115.

Fernandes de Paula, C. and Henrique Carvalho de Ferreira, L. (2011). An easy-to-use H_∞/LTR control solution with mixed-sensitivity properties. *IEEE Transactions on Automatic Control* 56(7): 1709 –1713.

Hairer, E., Nørsett, S. P. and Wanner, G. (2009). Solving ordinary differential equations I: Nonstiff problems.

Hauser, J. (2002). A projection operator approach to the optimization of trajectory functionals. In *IFAC World Congress*. Barcelona, Spain.

Hauser, J. and Meyer, D. (1998). The trajectory manifold of a nonlinear control system. In *IEEE Conference on Decision and Control*. Vol. 1. pp. 1034 –1039 vol.1.

Hauser, J. and Saccon, A. (2006). A barrier function method for the optimization of trajectory functionals with constraints. In *IEEE*

Conference on Decision and Control. pp. 864–869.

Jaillet, L. and Simeon, T. (2004). A PRM-based motion planner for dynamically changing environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 2. pp. 1606–1611 vol.2.

Johnson, E. R. and Murphey, T. D. (2009). Scalable Variational Integrators for Constrained Mechanical Systems in Generalized Coordinates. *IEEE Transactions on Robotics* 25(6): 1249–1261.

Kehoe, B., Warrier, D., Patil, S. and Goldberg, K. (2014). Cloud-based grasp analysis and planning for toleranced parts using parallelized Monte Carlo sampling. *IEEE Transactions on Automation Science and Engineering* PP(99): 1–16.

Kelley, C. T. (1999). *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics (SIAM). Philadelphia, PA.

Kewlani, G., Ishigami, G. and Iagnemma, K. (2009). Stochastic mobility-based path planning in uncertain environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 1183–1189.

Kobilarov, M., Marsden, J. E. and Sukhatme, G. S. (2011). Global estimation in constrained environments. *The International Journal of Robotics Research* pp. 24–41.

Kreindler, E. (1969). Formulation of the minimum trajectory sensitivity problem. *IEEE Transactions on Automatic Control* 14(2): 206 – 207.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.

LaValle, S. M. and Kuffner, J. J. (2001). Randomized kinodynamic planning. *The International Journal of Robotics Research* 20(5): 378–400.

Likhachev, M. and Ferguson, D. (2009). Planning long dynamically feasible maneuvers for autonomous vehicles. *The International Journal of Robotics Research* 28(8): 933–945.

Marino, H., Bergeles, C. and Nelson, B. (2014). Robust electromagnetic control of microrobots under force and localization uncertainties. *IEEE Transactions on Automation Science and Engineering* 11(1): 310–316.

Meirion-Griffith, G., Nie, C. and Spenko, M. (2014). Development and experimental validation of an improved pressure-sinkage model for small-wheeled vehicles on dilative, deformable terrain. *Journal of Terramechanics* 51: 19–29.

Melchior, N. and Simmons, R. (2007). Particle RRT for path planning with uncertainty. In *IEEE International Conference on Robotics and Automation*. pp. 1617–1624.

Murray, R. M., Li, Z. and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press. Boca Raton, FL.

Nie, C., Van Dooren, S., Shah, J. and Spenko, M. (2009). Execution of dynamic maneuvers for unmanned ground vehicles using variable internal inertial properties. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 4226 –4231.

Pepy, R., Lambert, A. and Mounier, H. (2006). Path planning using a dynamic vehicle model. In *Information and Communication Technologies (ICTTA)*. Vol. 1. IEEE. pp. 781–786.

Petti, S. and Fraichard, T. (2005). Safe motion planning in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 2210–2215.

Prentice, S. and Roy, N. (2009). The belief roadmap: efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research* 28: 1448–1465.

Rucco, A., Notarstefano, G. and Hauser, J. (2010). Dynamics exploration of a single-track rigid car model with load transfer. In *IEEE Conference on Decision and Control (CDC)*. pp. 4934 –4939.

Ruffatto III, D., Nie, C. and Spenko, M. (2014). The VIV: A mobile ground robot with variable inertial properties. *Journal of the Robotics Society of Japan* 32(4): 329–332.

Singer, N. C. and Seering, W. P. (1990). Preshaping command inputs to reduce system vibration. *Journal of Dynamic Systems, Measurement, and Control* 112(2): 76 –82.

Tedrake, R., Manchester, I. R., Tobenkin, M. and Roberts, J. W. (2010). LQR-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research* .

van den Berg, J., Abbeel, P. and Goldberg, K. (2011). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research* 30(7): 895–913.

van den Berg, J., Ferguson, D. and Kuffner, J. (2006). Anytime path planning and replanning in dynamic environments. In *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 2366–2371.

Wardi, Y. and Egerstedt, M. (2012). Algorithm for optimal mode scheduling in switched systems. In *American Control Conference*. pp. 4546–4551.

Wardi, Y., Egerstedt, M. and Twu, P. (2012). A controlled-precision algorithm for mode-switching optimization. In *IEEE Conference on Decision and Control*. pp. 713–718.

Wright, S. and Nocedal, J. (1999). *Numerical optimization*. Vol. 2. Springer New York.

Appendix A: Index to Multimedia Extensions

The multimedia extensions to this article can be found online by following the hyperlinks from `www.ijrr.org`.

Extension	Type	Description
1	Video	Minimal sensitivity robotic vehicle trajectories through uncertain environments