Minimal Sensitivity Control for Hybrid Environments

Alex Ansari and Todd Murphey

Abstract— **This paper presents a method to develop trajectories which remain optimally insensitive to sudden changes in dynamics. The approach is applied to two example systems that model a vehicle's attempt to navigate through potentially hazardous areas of the state space. Through these simplified examples, we show how to automatically plan trajectories which either avoid or adjust controls to safely pass through critical regions of the state space.**

I. INTRODUCTION

When traveling along winter roads drivers attempt to avoid ice patches or slow and align with the direction of travel before passing through. These intuitive driving styles are applied to keep the vehicle driving in a safe and predictable manner. They can also be posed as solutions to an optimal control problem. In this context, the driver attempts to track a desired trajectory, minimize control input, and avoid changes in the planned trajectory which could be caused by road hazards. As ice and other hazards emerge in their path, sacrifices in tracking and/or control adjustments must be made in order to pass through or around obstacles safely. It is important to note that in these circumstances the safety of a trajectory is tied to its predictability. A predictable trajectory is less sensitive to the sudden changes in dynamics which can be encountered near ice or other dynamically disparate regions of the state space.

This paper considers the sensitivity of trajectory optimization to discrete changes in system dynamics that might occur when the wheels of a vehicle slip over ice, stick in sand, or when parts of the system or carried load break or are removed. This sensitivity metric, measuring the change in system trajectory due to a transition from nominal dynamic conditions to alternative dynamic modes associated with potentially hazardous obstacles / regions, will be referred to as the *hybrid sensitivity* of a trajectory. By incorporating and minimizing a norm on these sensitivity terms during trajectory optimization, we show that it is possible to automatically synthesize trajectories which remain optimally[1]insensitive to discrete dynamic changes.

Several authors have developed methods for path and trajectory planning that account for critical or hazardous areas of the state space (e.g. [2], [6], [9], [10], [11], and [12]). Most of these approaches use barrier functions or similar methods in order to calculate trajectories that effectively avoid regions to mitigate their impacts on system dynamics. However, avoidance is not always feasible. Instead of strictly avoiding regions of the state space, controllers produced using the methods developed here can vary both path and applied control in order to pass through dynamically sensitive regions in an optimally insensitive manner. To our knowledge this approach to optimizing hybrid sensitivities in order to deal with dynamically critical regions of the state space is unique.

The techniques in this paper rely on a significant body of work in the area of hybrid systems. Specifically, we make use of the mode insertion gradient (see [3], [4], [5], [13], and [14]) to provide a measure of the sensitivity of a cost function to a switch 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 [5], [13], [14] and switching time optimization problems [4] to determine the optimal time to insert or remove dynamic modes from hybrid trajectories. Optimizing trajectories with respect to an L_2 norm on the mode insertion gradient, we ensure trajectories remain insensitive to new dynamic modes that may be encountered at any point along a trajectory.

The derivations and approach presented for incorporating and optimizing a norm on hybrid sensitivities in optimal control calculations are intended to be general. The methods can be directly applied to any number of iterative optimization techniques which utilize solutions to approximating LQR problems [1]. However, implementation results discussed make use of optimal control methods described in [7] and [8] as they can be applied to continuous time nonlinear systems without a-priori discretization. Additionally, the methodology returns feasible trajectories, $\xi(t) = (X(t), U(t))$, with feedback controllers at each iteration.

Following this introduction, Section II provides a derivation of the methods used to incorporate hybrid sensitivity optimization into iterative optimal control calculations. Sections III and IV describe results obtained by applying the techniques to two example systems – 1) a mass and damper system that models a vehicle passing through a dynamically sensitive region, and 2) a kinematic car that models the geometry of sensitivity avoidance.

II. Optimizing Hybrid Dynamical Sensitivities

The derivations presented in this section can be applied to any optimal control algorithm. For the implementation examples presented in Section III, we utilize an optimal

Alex Ansari and Todd Murphey are with the Department of Mechanical Engineering, McCormick School of Engineering and Applied Science, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, USA, alexanderansari2011@u.northwestern.edu, t-murphey@northwestern.edu

[1] The trajectory optimization and optimal control techniques discussed are local methods. As such, optimal trajectories are locally optimal with respect to a cost functional.

control based approach to trajectory optimization described in [7] and [8]. Below, Section II-A provides a brief overview of the general, iterative approach to trajectory optimization for which the sensitivity optimization techniques discussed can be applied. With this as a framework, Section II-B derives the method for optimizing hybrid sensitivities in trajectory optimization.

A. Overview of Trajectory Optimization

The trajectory optimization approach used here is traditionally applied to iteratively optimize quadratic cost functionals of the form[2]

$$\begin{aligned} J =& \frac{1}{2}\int_{t_0}^{t_f} \|X(t) - X_d(t)\|_Q^2 + \|U(t) - U_d(t)\|_R^2 \, dt \\ &+ \frac{1}{2}\|X(t_f) - X_d(t_f)\|_{P_1}^2 \ , \end{aligned} \tag{1}$$

where matrices

$$Q \geq 0, \ R > 0, \text{ and } P_1 \geq 0.$$

Trajectory optimization is performed by minimizing cost functional (1) with respect to trajectory, $\xi(t) = (X(t), U(t))$, constrained by dynamics

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

For a first order method, optimization is achieved by iteratively approximating (1) with a model, $g(\zeta)$, that is quadratic with respect to perturbations to the trajectory, $\zeta(t) = (z(t), v(t))$. Curves $z(t)$ and $v(t)$ represent perturbations to the current trajectory's state, $X(t)$, and controls, $U(t)$, respectively. With this approximation, LQR methods as described in [1] can be directly applied to solve the constrained minimization problem[3]

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

subject to constraint

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

Where $A(t)$ and $B(t)$ represent linearizations of the dynamics (2) with respect to the state and controls, constraint (4) requires that perturbations locally obey the system dynamics.

Solutions to minimization (3) provide first order descent directions at each iteration. Following from traditional steepest descent, the trajectory optimization approach perturbs each iteration's trajectory in these directions, using a line search to determine scaling for sufficient decrease. Through a feedback projection, each perturbed trajectory is projected to a nearby solution obeying system dynamics. As such, the process ensures feasible trajectories, $\xi(t)$, at each iteration. The techniques guarantee incremental progress and convergence to an optimizer if one exists.

Following this trajectory optimization proceedure, controllers can be calculated that, under model conditions, minimize norms on state tracking error and applied controls while steering systems along a specified course. For further details the reader is referred to the works of [7] and [8].

B. Incorporating Hybrid Sensitivities

In order to select trajectories that remain insensitive to sudden changes in dynamics, it is necessary to quantify the sensitivity of a trajectory to a discrete dynamic switch. Where cost functional (1) provides a norm on the state tracking error and applied controls across a given trajectory, the mode insertion gradient, denoted

$$\frac{dJ}{d\lambda^+} = \rho(t)^T (f_2(\xi(t)) - f(\xi(t))), \tag{5}$$

provides a measure of the sensitivity of the cost functional to a dynamic mode insertion of infinitesimal duration [5].[4]Because changes in the cost functional relate to changes in trajectory, (5) also provides a measure of the sensitivity of a trajectory to the insertion of a new dynamic mode.

Using the tools of trajectory optimization previously described, it is difficult to directly incorporate (5) into cost functional (1) because (1) should be a function of only the current trajectory. While it is possible to append the state vector with (5) to incorporate the mode insertion gradient into the cost functional, calculating (5) first requires solving for the adjoint variable, $\rho(t)$. This necessitates backwards integration of the relation

$$\dot{\rho}(t) = -D_1 l(X(t), U(t))^T - D_1 f(X(t), U(t))^T \rho(t), \tag{6}$$

where $l(X(t), U(t))$ is the incremental cost (portion under the integral) from (1) and $\rho(t_f) = \mathbf{0}$. Because it relies on integration of the dynamics, (5) is not a simple function of the current trajectory and cannot be appended to the state without further modifications to the underlying trajectory optimization algorithm.

To keep the derivation general and to allow use of the trajectory optimization techniques described in Section II-A, a norm on the mode insertion gradient can be incorporated into (1) by first defining a new appended state vector, $\overline{X}(t) = [X(t), \ \rho(t)]^T$. This choice results in an appended dynamics vector

$$\bar{f}(\overline{X}(t), U(t)) = \begin{pmatrix} f(X(t), U(t)) \\ -Q \cdot (X(t) - X_d(t)) - A(t)^T \rho(t) \end{pmatrix}. \tag{7}$$

Additionally, it should be noted that while $X(t)$ is calculated from forward integration of dynamics (2), $\overline{X}(t)$ is calculated as the solution of a two point boundary value problem. Where the first half of vector (7) requires forward integration of (2) from initial time t_0, the second half requires backwards integration of (6) from final time t_f. Because the first half of this dynamics vector can be solved independently of the second however, the problem remains tractable.

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

[3] $DJ(\xi)$ refers to the slot derivative of $J(\xi)$ with respect to its argument. 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.

[4] As used in this paper, $f(\xi(t))$ represents the system dynamics without the mode insertion and $f_2(\xi(t))$ represents the system dynamics that corresponding to a new dynamic mode inserted for an infinitesimal duration.

Appending the state vector in this form 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). To incorporate a norm on the mode insertion gradient we define[5]

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

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

The w term in the $\overline{Q}$ weighting matrix above is a scalar multiplier used to weight the norm 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. As such, $\overline{P}_1$ only provides a norm on the state tracking error at the final time, as in the case of matrix P_1.

With these modifications, a new cost functional for sensitivity optimization can be defined as

$$\begin{aligned}\bar{J} =& \frac{1}{2}\int_{t_0}^{t_f} \|\overline{X}(t) - \overline{X}_d(t)\|^2_{\overline{Q}} + \|U(t) - U_d(t)\|^2_R \, dt \\ &+ \frac{1}{2}\|\overline{X}(t_f) - \overline{X}_d(t_f)\|^2_{\overline{P}_1}. \end{aligned} \tag{8}$$

Equation (8) is identical to the original trajectory optimization cost functional (1) except for the term

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

The norm on the appended state tracking error introduces a term to the integral of (8) which is equivalent to a weighted square of the mode insertion gradient.[6]With this addition, the cost functional incorporates an L_2 norm on the sensitivity of trajectories to discrete changes in dynamics. As will be shown, optimal trajectories computed using this cost functional can balance sensitivities to sudden changes from dynamics $f(\xi(t))$ to dynamics $f_2(\xi(t))$, relative to state tracking error and the magnitude of applied controls.

To calculate the descent directions, $\overline{\zeta}(t) = (\overline{z}(t), v(t))$, required to iteratively optimize (8), the constrained minimization problem

$$\min_{\overline{\zeta}} \overline{g}(\overline{\zeta}) = D\bar{J}(\overline{\xi}) \cdot \overline{\zeta} + \frac{1}{2}\int_{t_0}^{t_f} \|\overline{\zeta}\|^2 dt \tag{9}$$

must be solved subject to the constraint

$$\dot{\overline{z}}(t) = \overline{A}(t)\overline{z}(t) + \overline{B}(t)v(t). \tag{10}$$

As formulated, $\overline{g}(\overline{\zeta})$, provides a local quadratic approximation to (8) analogous to the way $g(\zeta)$ locally approximates (1). However, the linear terms differ between the two models. Where $DJ(\xi) = [\frac{\partial J}{\partial X(t)}, \frac{\partial J}{\partial U(t)}]$ is made up of partial derivatives

$$\frac{\partial J}{\partial X(t)} = \int_{t_0}^{t_f} (X(t) - X_d(t))^T \cdot Q dt + (X(t_f) - X_d(t_f))^T \cdot P_1 \tag{11}$$

and

$$\frac{\partial J}{\partial U(t)} = \int_{t_0}^{t_f} (U(t) - U_d(t))^T \cdot R dt, \tag{12}$$

$D\bar{J}(\overline{\xi}) = [\frac{\partial \bar{J}}{\partial X(t)}, \frac{\partial \bar{J}}{\partial \rho(t)}, \frac{\partial \bar{J}}{\partial U(t)}]$ is composed of partial derivatives

$$\begin{aligned}\frac{\partial \bar{J}}{\partial X(t)} = \int_{t_0}^{t_f} &(X(t) - X_d(t))^T \cdot Q \\ &+ w(\rho(t)^T(f_2 - f))\rho(t)^T(A_2 - A)dt \\ &+ (X(t_f) - X_d(t_f))^T \cdot P_1,\end{aligned} \tag{13}$$

$$\frac{\partial \bar{J}}{\partial \rho(t)} = \int_{t_0}^{t_f} w(\rho(t)^T(f_2 - f))(f_2 - f)^T dt, \tag{14}$$

and

$$\begin{aligned}\frac{\partial \bar{J}}{\partial U(t)} = \int_{t_0}^{t_f} &w(\rho(t)^T(f_2 - f))\rho(t)^T(B_2 - B) \\ &+ (U(t) - U_d(t))^T \cdot R dt.\end{aligned} \tag{15}$$

To constrain the solutions to (9), linearizations of dynamics (7) 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 (10), where

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

and

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

III. Simulations

In the following sections the sensitivity optimization techniques discussed are applied to two simple example systems. These systems provide abstractions of vehicle trajectory optimization scenarios that demonstrate the performance characteristics of the algorithm and its application to robotic vehicles.

A. Example System I

In the first example scenario a simulated robotic vehicle is provided a desired trajectory from its starting position to a desired position 2 meters away in the least time possible. In this scenario, the robot has collected sufficient measurements to localize an ice patch at a point in the state space in between the robot's initial configuration and the final desired position. The approximate location of the ice patch is represented

[5] 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.

[6] 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.

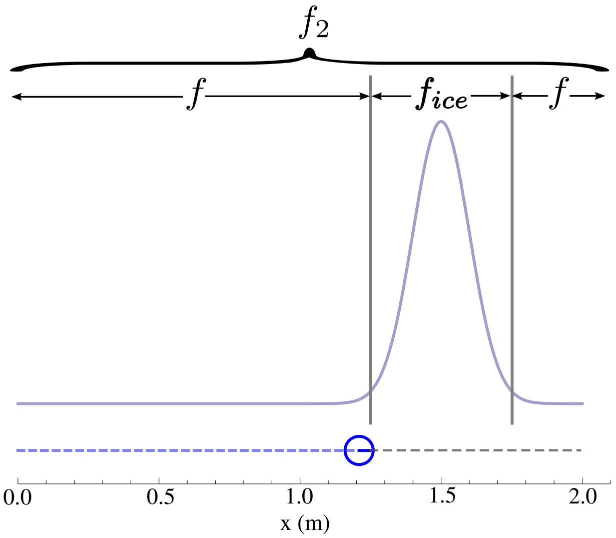

Fig. 1: Dynamics f_2 are designed to transition between dynamics f_{ice} near the expected location of the ice patch and the normal dynamics f in other areas of the state space according to the pdf (light blue). The robot (blue circle) is depicted traveling along its desired trajectory (gray dashed line). The expected location of the ice patch is at the center of the pdf at $x = 1.5m$.

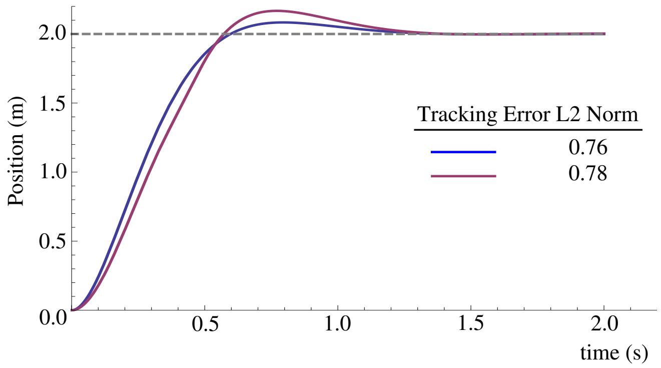

Fig. 2: State trajectories resulting from standard trajectory optimization techniques (blue curve) and sensitivity optimization (purple curve). The dashed gray curve reflects the desired trajectory.

by a Gaussian belief distribution centered at the expected location of the patch. Assuming the robot has a rough model of how the ice influences its dynamics, standard and sensitivity-enhanced trajectory optimization are applied and results compared in Section IV.

For the system described, the vehicle controls vector, $U(t) = F(t)$, consists only of the thrust developed by tires. The vehicle's state vector, $X(t) = [x(t),\, \dot{x}(t)]^T$, includes its position and velocity as it moves in one dimension between initial and final locations. These simplified vehicle dynamics are given by

$$f(X(t), U(t)) = \begin{pmatrix} \dot{x}(t) \\ -b\dot{x}(t) + F(t) \end{pmatrix}, \qquad (16)$$

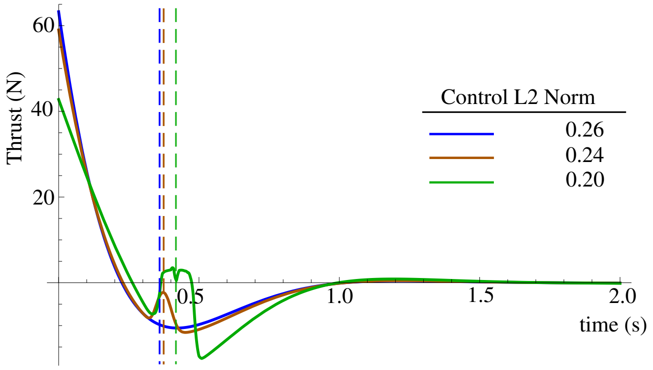

Fig. 3: The magnitude of thrust applied during standard trajectory optimization techniques (blue curve) and sensitivity optimization. Increasing weight, w, in the sensitivity norm produces the brown and green curves. The colored dashed lines reflect the time at which each correspondingly colored trajectory crosses the expected location of the ice patch. Sensitivity optimized trajectories apply less initial acceleration and reduce control authority to minimize braking when crossing the ice. After crossing, braking is reapplied with greater magnitude than nominal trajectories.

where $b = 0.9$ is the viscous frictional damping coefficient assumed under normal driving conditions. For the scenario discussed, it is assumed the dynamics of the vehicle on ice can be roughly modeled by

$$f_{ice}(X(t), U(t)) = \begin{pmatrix} \dot{x}(t) \\ \frac{1}{2}(-b\dot{x}(t) + F(t)) \end{pmatrix}, \qquad (17)$$

reflecting the fact that when on ice, the robot's control authority and viscous damping are reduced drastically.

If the goal of sensitivity optimization were to develop a trajectory with minimal sensitivity to ice patches that could be encountered anywhere in the state space then f_2 could be directly equated to f_{ice}. However, the expected location of the ice patch can be incorporated into the definition of f_2 to develop trajectories that have nominal state evolution except in the vicinity of ice. In this case,

$$f_2(X(t), U(t)) = \begin{pmatrix} \dot{x}(t) \\ \dfrac{f_{ice_{2,1}}}{(1 + e^{-(x(t)-\mu)^2}/(2\sigma^2))} \end{pmatrix} \qquad (18)$$

incorporates the expected location and influence of the ice on the dynamics. Where $f_{ice_{2,1}}$ is the second component of vector f_{ice}, (18) ensures that dynamics (17) are applied at this location and transitions to dynamics (16) in other areas of the state space according to the Gaussian describing the expected location of the ice. The dynamics imposed by f_2 as the robot passes through different areas of the state space are graphically depicted in Figure 1.

In implementation, we used $\mu = 1.5m$ to center the distribution representing the expected location of the ice, with $\sigma = 0.1m$ as its standard deviation. Optimization results are included in Figures 2 and 3. These figures show sensitivity optimized trajectories develop less initial acceleration to avoid braking and minimize control authority over the ice. By reapplying braking after the ice, these trajectories successfully reach the desired position with similar state tracking performance and slightly less control authority.

B. Example System II

The second example scenario simulates a robotic vehicle following a desired path that passes in the vicinity of a sand pit. Again, the location of the sand pit is only approximately known and represented as the center of a Gaussian belief distribution. It is assumed the robot has a rough model of how the sand will impact its dynamics. Traditional and sensitivity-enhanced trajectory optimization are applied and results compared in Section IV.

For this scenario the kinematic car model is used. In this model, the state space vector of the robot, given by $X(t) = [x(t),\ y(t),\ \theta(t)]^T$, reflects its $2D$ location and orientation. The control vector, $U(t) = [v(t),\ \omega(t)]^T$, encompasses both forward and angular velocity controls. The dynamics associated with this model are

$$f(X(t), U(t)) = \begin{pmatrix} v(t)cos(\theta(t)) \\ v(t)sin(\theta(t)) \\ \omega(t) \end{pmatrix}. \tag{19}$$

Assuming that getting caught in sand will have a near uniform slowing effect on the robot's movements, the dynamics of the robot in sand can scaled down by

$$f_{sand}(X(t), U(t)) = \frac{1}{10} f(X(t), U(t)). \tag{20}$$

The optimization methods discussed remain relatively insensitive to specific parameters in the dynamics used to model these critical regions[7]as long as the relative dynamic effects are captured. In this case, using a factor of $\frac{1}{10}$ compared to other fractional multiples produces similar trajectories and affects the optimization in the same manner as choosing an alternative weight, w.[8]These approximate dynamics and the expected location of the sand are incorporated into dynamics

$$f_2(X(t), U(t)) = f \cdot (1 - 0.9\, e^{(\frac{-(x(t)-\mu_x)^2}{2\sigma_x^2} - \frac{(y(t)-\mu_y)^2}{2\sigma_y^2})}). \tag{21}$$

As opposed to the first example scenario, (21) uses a $2D$ Gaussian distribution centered at (μ_x, μ_y) with standard deviations in x and y directions of (σ_x, σ_y) to transition between (20) at the believed location of the sand and (19) in other areas of the state space (see Figure 4). Simulation parameters $(\mu_x, \mu_y) = (1.5m, 0.85m)$ were applied with $(\sigma_x, \sigma_y) = (0.18m, 0.1m)$. Trajectory optimization results included in Figure 4, demonstrate trajectories applying varying degrees of obstacle avoidance as the vehicle attempts to minimize hybrid sensitivities in regions around the sand.[9]

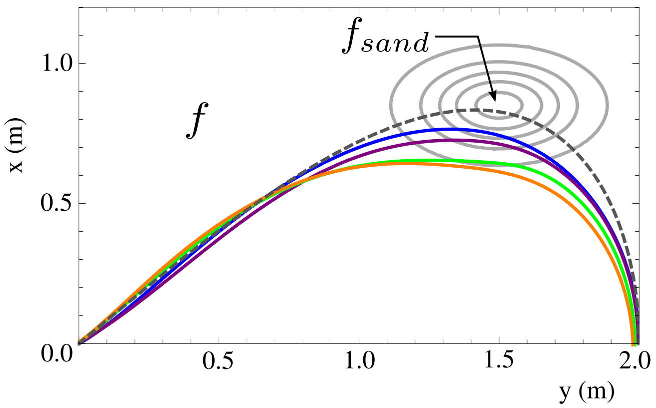

Fig. 4: State trajectories resulting from standard trajectory optimization techniques (blue curve) and sensitivity optimization. Increasing weight, w, in the sensitivity norm produces the purple, green, and orange curves. The dashed gray curve reflects the desired trajectory, and the gray contours depict level sets of the Gaussian representing the location of the sand pit and the influence of the sand dynamics in f_2. Dynamics f_2 transitions 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 in the vicinity of the sand.

IV. Discussion

Simulated trajectory Figures 2 and 3 show the state evolution and applied controls for the robotic vehicle model in the example scenario described in Section III-A as it tracks a desired trajectory. The results included in these figures ignore the effects of the ice patch on the vehicle's state and dynamics. Instead, the purpose of the figures is to demonstrate the different *driving styles* the vehicle develops when the expected location of the ice patch is incorporated in sensitivity optimization calculations.

Through a weighted L_2 norm on the mode insertion gradient, sensitivity optimized trajectories use control authority and adjust state tracking to minimize the sensitivity of resulting trajectories to discrete changes in dynamics. Using these methods, even a rough model of how ice impacts the dynamics of a robotic vehicle recovers one of the natural control strategies discussed in Section I. Demonstrated in Figure 3, the vehicle slows earlier on, reducing the magnitude of applied controls before entering the vicinity of the ice patch and minimizing applied controls. Only after passing through the believed location of the patch does the vehicle begin aggressively braking to slow itself and come to a stop at the desired position. Through this control strategy, the robot minimizes the influence the ice patch could have on its trajectory.

[7] Critical regions refer to areas of the state space where the expected dynamics significantly differ from nominal modeled dynamics. In these examples critical regions refer to the areas of ice and sand.

[8] As a rough measure of this influence, consider that weights used to develop the sensitivity trajectories in Figure 4 vary by a factor of 10 between purple and orange curves.

[9] In all cases the norm on the directional derivative of the cost functional was used to test for convergence using a criterion of 0.1. This value was approximately three orders of magnitude smaller than the initial norm on the directional derivative.

As evinced by the L_2 errors included in Figures 2 and 3, in applying the sensitivity optimized driving style, the vehicle manages to reach the desired final position with only modest sacrifice in tracking error and an overall reduction in the norm on the applied controls. This performance is notable considering that more traditional obstacle avoidance techniques (i.e. barrier functions) may be unable to reach the final position and would perform poorly when the vehicle does not have the option to "change lanes" to avoid the ice. In conditions where ice were actually present, the methods discussed provide plausible trajectories to mitigate its effects and thus could reduce tracking error at relatively little cost.

The example scenario described in Section III-B demonstrates the performance of the algorithm in a $2D$ environment where the optimal trajectories can be interpreted as obstacle avoidance. In this case, the desired trajectory tracked by the robotic vehicle brings it toward areas of the state space where a sand pit is located with high expectation. In this scenario, the robotic vehicle is modeled as a kinematic car and limited so that it cannot take advantage of sliding or inertial dynamic effects. Additionally, because the sand pit dynamics uniformly damp the robot's motions, if the robot enters the sand its trajectory will be significantly affected regardless of applied controls.

Under the described conditions, sensitivity optimization returns the results presented in Figure 4, where the robot minimizes its sensitivity to a switch to the sand dynamics by avoiding areas of the state space around the sand. Where the blue curve represents trajectory optimization results that do not account for the sand, the purple, green, and orange curves represent trajectories which result from sensitivity optimizations with increasing weight, w, on the norm of the mode insertion gradient. As the figure makes apparent, increasing the weight on the mode insertion gradient causes the vehicle to make sacrifices in trajectory tracking in order to give the sand pit a wider birth.

In both scenarios dynamically disparate areas of the state space are represented using a Gaussian distribution to approximate their location and extent of influence. By increasing the standard deviation or shape of the belief distributions, it is a straightforward process to adjust the shape and influence of these regions. By increasing the standard deviations of the Gaussian representing the location of the sand in the scenario in Section III-B, one can achieve the same effect as increasing weight, w, to select for trajectories with avoid a greater area of the state space. As shown in Figure 4, when the vehicle cannot minimize sensitivity through controls alone, trajectories results can be influenced to apply varying levels of obstacle avoidance to mitigate the impacts of sand.

V. CONCLUSIONS

The methods in this paper provide a means to incorporate an L_2 norm on the mode insertion gradient in trajectory optimization calculations. The approach is used to select trajectories which avoid dynamic transitions, and thus can be tracked more predictably in the presence of hybrid uncertainty.

Performance is demonstrated under conditions where only approximate locations and models for dynamically disparate regions of the state space are available. Examples illustrate how the approach can be used to synthesize robotic vehicle trajectories which employ intuitive means to avoid or pass through these critical regions in an optimally insensitive manner. As the paths produced can be tracked more predictably in dynamically varied environments, these techniques can provide for more reliable and safer trajectories without necessitating the strict avoidance of hazardous regions.

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

[1] Brian D. O. Anderson and John B. Moore. *Optimal control: linear quadratic methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] Jerawan Chudoung and Carolyn Beck. The minimum principle for deterministic impulsive control systems. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 4, pages 3569–3574. IEEE, 2001.

[4] Magnus Egerstedt, Yorai Wardi, and Henrik Axelsson. Optimal control of switching times in hybrid systems. In *9th IEEE International Conference on Methods and Models in Automation and Robotics*, 2003.

[5] Magnus Egerstedt, Yorai Wardi, and Henrik Axelsson. Transition-time optimization for switched-mode dynamical systems. *Automatic Control, IEEE Transactions on*, 51(1):110–115, 2006.

[6] Philip E Gill, Walter Murray, Dulce B Ponceleon, and Michael A Saunders. Solving reduced KKT systems in barrier methods for linear and quadratic programming. Technical report, DTIC Document, 1991.

[7] J. Hauser and D.G. Meyer. "The trajectory manifold of a nonlinear control system". In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 1, pages 1034 –1039 vol.1, 1998.

[8] John Hauser. "A projection operator approach to the optimization of trajectory functionals". In *IFAC World Congress*, Barcelona, Spain, Jul. 2002.

[9] John Hauser and Alessandro Saccon. A barrier function method for the optimization of trajectory functionals with constraints. In *Decision and Control, 2006 45th IEEE Conference on*, pages 864–869. IEEE, 2006.

[10] Yoram Koren and Johann Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1398–1404. IEEE, 1991.

[11] Sezimária FP Saramago and Valder Steffen Junior. Optimal trajectory planning of robot manipulators in the presence of moving obstacles. *Mechanism and Machine Theory*, 35(8):1079–1094, 2000.

[12] Marc C Steinbach et al. A structured interior point SQP method for nonlinear optimal control problems. *Computational Optimal Control*, 115:213–222, 1994.

[13] Yorai Wardi, M Egerstedt, and P Twu. A controlled-precision algorithm for mode-switching optimization. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 713–718. IEEE, 2012.

[14] Yorai Wardi and Magnus Egerstedt. Algorithm for optimal mode scheduling in switched systems. In *American Control Conference*, pages 4546–4551, 2012.