Robustness of Classifier-in-the-Loop Control Systems: A Hybrid-Systems Approach

Hasan A. Poonawala and Ufuk Topcu

Abstract— **This paper studies continuous-state dynamical systems that use a classifier to determine the control input. Since the classifier output belongs to a finite set, the feedback control is a piece-wise constant function of the state. We therefore model the closed-loop system as a switched system. The decision surfaces of the classifier in the feature space dictate determine the switching surfaces in the state space. Therefore, the classifier affects the stability of the closed-loop system. Any analysis of the nominal closed-loop system may not be valid in states or environments that the training data for the classifier do not represent. We propose techniques to determine when the stability of the nominal system can be extended to unseen states and/or environments.**

I. Introduction

Machine learning techniques are capable of producing classifiers that process complex high-dimensional input data into semantic information. These classifiers are becoming increasingly effective at classifying data such as images and audio, or identifying objects present within data [3], [9]. In classification tasks involving independent features, achieving high classification accuracy is the primary goal of the machine learning algorithm. Recent machine learning algorithms can generate classifiers with high enough accuracy that researchers are attempting to apply them to real-time data collected from sensors in order to control robots [6]. Classifiers used in this context are classifying a sequence of features that may not be independent, and so the performance of the classifier should be evaluated in terms of the closed-loop behavior that it generates. The performance of classifiers used in a feedback loop have not been formally studied. We refer to such closed-loop systems as classifier-in-the-loop systems.

The training data usually consists of a set of known states, the feature measured in those states, and the correct labels associated with the measured features. In practice, machine learning algorithms applied to finite-amount training data produce classifiers that distinguish between a finite set of labels only. This constraint implies that a feedback control strategy based on the a classifier trained from limited data can only command a finite set of control inputs. The control applied to the system will be piece-wise constant, so we model the closed-loop dynamics as a switched system.

The switched system we obtain poses two challenges. First, the switching surfaces in the state space depend on the classifier, which in turn depends on the training data and learning algorithm used. The switching surfaces may lead to unstable behavior or undesired phenomena such as limit cycles. Second, the dynamics and switching surfaces may depend on the environment in which data was collected, and the system may find itself in an environment that the training data do not represent. To address the first challenge, we must analyze the closed-loop behavior of the nominal switched dynamical system that we construct from the classifier and data. To address the second challenge, we must analyze the robustness of this system to perturbations of the switching surface. A classifier that assigns correct labels to features that were not present in the training data is said to have the ability to *generalize*. We focus on the second challenge. We propose a notion of generalization for feedback control using classifiers, and develop tools to decide when such closed-loop systems do indeed generalize.

A significant body of recent research focuses on solving complex control problems as end-to-end machine learning problems [8], [10]. These methods combine reinforcement learning with the representation power of deep neural networks. The method for computing a policy in [8] involves simultaneously implementing reinforcement learning and model predictive control. Reinforcement learning requires that the control must be implemented and its effect experienced in order to drive the learning. The complexity of this approach makes implementation on a real system difficult. In contrast, the work in [6] uses supervised deep learning to derive a mapping from captured image to control command. The control is never implemented during training, and a simple and ingenius trick is sufficient to obtain enough data to train a classifier that results in stable closed-loop behavior even in unseen environments. However, the stability is evaluated only empirically. The main advantage of the supervised learning approach is its ease of implementation, as observed in [6].

Our paper is motivated by the need to formally analyze such systems and provide guarantees for the behavior of these systems. The contributions of this paper are as follows.

- We show that classifier-in-the-loop systems can be modeled using switched systems, where the switching surfaces are uncertain.
- We define a notion of *generalization* in classifier-in-the-loop systems.
- We analyze the control task of making a mobile robot follow a path in two-dimensions using three velocity commands: move forward, turn left, and turn right. We simulate a sensor based on structured lighting, and train a classifier that chooses one of the three control commands

This work is supported by the National Science Foundation (NSF 1652113 and NSF 1617639) and Air Force Research Labs (FA8650-15-C-2546).

Hasan A. Poonawala is with the Institute for Computational Engineering and Science, University of Texas, Austin, TX 78712, USA. hasanp@utexas.edu

Ufuk Topcu is with the Department of Aerospace Engineering, University of Texas, Austin, TX 78712, USA. utopcu@utexas.edu

from the sensor measurement. The resulting nominal switched system is shown to generalize through analysis and simulations.

II. Preliminaries

A. Dynamical Systems

Consider a dynamical system with state $x \in X \subseteq \mathbb{R}^n$. The dynamics of the system are given by

$$\dot{x} = f(x,p) + g(x,p)u \tag{1}$$

where $u \in U \subseteq \mathbb{R}^p$ is the control input, $p \in P \subseteq \mathbb{R}^q$ is a parameter, and $f\colon \mathbb{R}^n \to \mathbb{R}^n$ and $g\colon \mathbb{R}^n \to \mathbb{R}^n$ are functions that define the drift and input vector fields over X respectively, given p. In this paper, the parameter p will represent the environment that an agent is interacting with or operating in. The available sensor obtains a feature $\phi \in \Phi \subseteq \mathbb{R}^m$. The (unknown) relationship between the feature, state, and environment is represented as a map $\mathcal{H}\colon X \times P \to \Phi$, that is $\phi = \mathcal{H}(x,p)$.

B. Feature Classification

A classifier $C\colon \Phi \to L$ is a map that assigns a unique label $b \in L$ to a feature $\phi \in \Phi$, where Φ is the set of features. The goal of machine learning algorithms is to learn a set of weights $w \in \mathbb{R}^m$ (which parameterize the classifier) that minimizes the expected error between the label associated with feature ϕ in the data and the predicted label $C(\phi)$.

The set D of training data consists of N triples (x^i, ϕ^i, b^i) where i denotes the index of an element of the training data. The training algorithm that determines the classifier C will attempt to learn the relationship between the features and labels. Since the feature depends on the state and environment, and the control depends on the label, the classifier is indirectly learning a mapping from the state to the control.

C. Topology of Functions

Definition 1 (Function of class k)**.** A function f is said to be a *function of class* k, where $k \in \mathbb{N}$, if it is continuous and continuously differentiable up to order k , inclusive, in its domain of definition. The set consisting of all functions of class k is denoted as $\mathcal{C}^k$.

Definition 2 (δ-closeness to rank r [2])**.** A function $f \in \mathcal{C}^k$ defined over a region W is said to be *δ-close to rank r* ($r \leq k$) to a function g (also defined on W) if $\|f - g\| < \delta$ and $\|f^{(l)} - g^{(l)}\| < \delta$ for $1 \leq l \leq r$, where $f^{(l)}$ is the l^{th} order partial derivative of f.

Let f_1 and f_2 be two functions of class $\mathcal{C}^k, k \in \mathbb{N}$. The distance $d_{\mathcal{C}^1}$ between f_1 and f_2 over a domain W is

$$d_{\mathcal{C}^1}(f_1, f_2) = \sup_{x \in W} \|f_1(x) - f_2(x)\| + \|f_1'(x) - f_2'(x)\|. \tag{2}$$

D. Path-Following

We will use path-following in the plane as an example in order to illustrate the concepts presented in the paper. The configuration of a mobile agent in the plane is given by (x_1, x_2, θ), where $(x_1, x_2) \in \mathbb{R}^2$ denotes the location of the centroid of the robot and $\theta \in \mathcal{S}^1$ denotes the heading of the robot (see Figure 1). The kinematics are given by

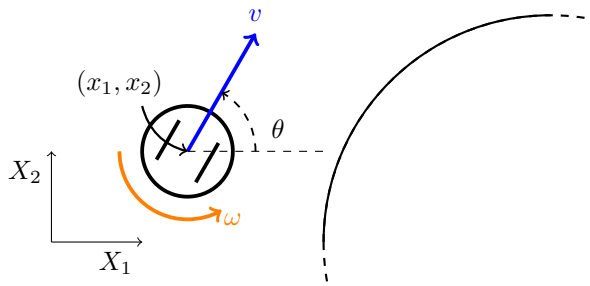


Fig. 1: A mobile robot with configuration (x_1, x_2, θ), linear speed v, and angular velocity ω. The curved black line represents a local segment of a path that the robot must follow.

$$\dot{x}_1 = v\cos(\theta), \tag{3a}$$
$$\dot{x}_2 = v\sin(\theta), \tag{3b}$$
$$\dot{\theta} = \omega, \text{ and} \tag{3c}$$

where $v \in \mathbb{R}$ and $\omega \in \mathbb{R}$ are the forward speed and angular velocity respectively. The differential equations (3) are often used to model wheeled mobile robots, though they are applicable to any system that moves in the plane with a velocity directed along its instantaneous heading only. This model will be controlled using only three values of the control input (v, ω). The set of labels is $L = \{b_F, b_R, b_L\}$ corresponding to moving forward, turning right, and turning left respectively. We choose the map $G\colon L \to U$ to be

$$G(b) = \begin{cases} (v^*, 0) & \text{if } b = b_F \\ (0, \omega^*) & \text{if } b = b_L \\ (0, -\omega^*) & \text{if } b = b_R, \end{cases} \tag{4}$$

where $v^* > 0$ and $\omega^* > 0$ are constants. The curvature ρ of the path is the environment parameter p.

III. Classifier-in-the-Loop Systems

A classifier can be used to generate the control input u to the system in (1) by associating a label $b \in L$ to the feature ϕ obtained from the sensor. This control is given by a map $G\colon L \to U$, so that $u_b = G(b)$. This process results in a feedback loop that is depicted in Figure 2. We call such systems as classifier-in-the-loop systems.

The classifier defines decision surfaces in Φ. We can represent the i^{th} decision surface by a function $\beta_i\colon \mathbb{R}^m \to \mathbb{R}$, through the constraint $\beta_i(\phi) = 0$. Since $\phi = \mathcal{H}(x,p)$, these decision surfaces in Φ define switching surfaces in the space $X \times P$ corresponding to the piece-wise constant control $u = (G \circ C)(\phi)$. The switching surface in $X \times P$ corresponding to β_i is $(\beta_i \circ \mathcal{H})(x,p) = 0 \coloneqq \gamma_i(x,p) = 0$. The closed-loop system becomes a switched dynamical system under the switching control u.

We will consider the case of static environments. The closed-loop dynamics near a switching surface in X are

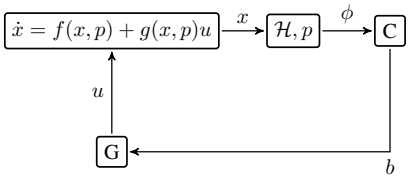


Fig. 2: A dynamical system with an end-to-end classifier $G \circ C$ as the controller. The class label b is mapped to the control u. The feature ϕ obtained by the sensor in a state x and environment p depends on the (unknown) map $\mathcal{H}$.

given by

$$\dot{x} = \begin{cases} f^1(x,p) & , \text{ if } \gamma(x,p) \geq 0, \text{ and} \\ f^2(x,p) & , \text{ if } \gamma(x,p) \leq 0, \end{cases} \tag{5}$$

where each of the vector fields f^1 and f^2 is obtained when choosing the control u as u_a and u_b for two labels $a, b \in L$. The switched system may consist of multiple switching surfaces, each one corresponding to one of the possible pairs of labels in L.

Let the training data be collected in an environment p^*. We estimate the surface $\gamma(x,p^*) = 0$ using these data (see Section IV). Let the estimated surface for $p \equiv p^*$ be denoted as $\alpha(x) = 0$. We define the following switched system:

$$\dot{x} = \begin{cases} f^1(x,p^*) & , \text{ if } \alpha(x) \geq 0, \text{ and} \\ f^2(x,p^*) & , \text{ if } \alpha(x) \leq 0. \end{cases} \tag{6}$$

We will refer to (6) as the nominal system. When the vector fields $f^1(x,p^*)$ and $f^2(x,p^*)$ are known, one can use methods from [5], [7] to investigate the behavior of the trajectories of (6). We also define the system

$$\dot{x} = \begin{cases} f^1(x,p) & , \text{ if } \alpha(x) \geq 0, \text{ and} \\ f^2(x,p) & , \text{ if } \alpha(x) \leq 0, \end{cases} \tag{7}$$

which is an estimate of the closed-loop dynamics in (5) when using the classifier defined from the training data in environments different from p^*.

The main concern of this paper is whether the conclusions derived for systems (6) or (7) are valid for system (5). There are two reasons that this concern arises. First, the surface $\alpha(x) = 0$ is estimated using limited training data, and may not be identical to $\gamma(x,p) = 0$, even for $p = p^*$. Second, the system may need to operate in environments p that are different from p^*. In this case, the switching surfaces *and* the vector fields of the closed-loop system (5) may be different from those in (6).

In the path-following problem, we want a robot with dynamics (3) to follow a continuous and connected path (defined by the zero-level set of a function $\sigma\colon \mathbb{R}^2 \to \mathbb{R}$) in the plane. To any point $a \in \mathbb{R}^2$ close enough to the path σ we can associate a unique Frenet-Serret frame corresponding to the point b (where $\sigma(b) = 0$) that is closest to a. Then, the configuration of the agent in this frame is $z = (\psi, w)$, where ψ is the difference between θ and the direction of the tangent to σ at b and w is the distance between a and b. The dynamics of z depend on the curvature of the path, and therefore the environment variable p is the curvature ρ.

The dynamics are given by

$$\begin{bmatrix} \dot{\psi} \\ \dot{w} \end{bmatrix} = \begin{cases} \begin{bmatrix} \omega^* & 0 \end{bmatrix}^T, & \text{if } b = b_L, \\ \begin{bmatrix} \frac{v^* \rho \cos(\psi)}{1 - \rho w} & v^* \sin(\psi) \end{bmatrix}^T, & \text{if } b = b_F, \text{ and} \\ \begin{bmatrix} -\omega^* & 0 \end{bmatrix}^T, & \text{if } b = b_R. \end{cases} \tag{8}$$

Note that the conditions in (8) are expressed in terms of the class label instead of the value of functions of the form $\beta_i\colon X \to \mathbb{R}$. In the next section, we provide a method to estimate these switching surfaces.

IV. Switching Surfaces

In this section, we show how to compute an approximation $\alpha_i(x) = 0$ of the switching surface $\gamma_i(x,p^*) = 0$, where p^* corresponds to value of p represented in the data. Recall that the decision surfaces in Φ defined by C are denoted as $\beta_i(\phi) = 0$, and therefore $\gamma_i = \beta_i \circ \mathcal{H}$. We assume that the surfaces β_i have co-dimension 1 almost everywhere in Φ. This assumption is required for the switched system obtained from these surfaces to be well defined.

Let ϕ_j be the j^{th} element of ϕ. Using the data $D = \{(x^k, \phi^k, b^k)\}$, $k \in \{1, \ldots, N\}$ corresponding to the environment p^*, we can compute interpolating functions h^j, $1 \leq j \leq m$ such that $\phi_j^k = h^j(x^k)$ for $k \in \{1, \ldots, N\}$. The functions h^j for $j \in \{1, \ldots, m\}$ together are denoted by the map h.

We can associate a class label b_x to each state $x \in X$ as $b_x = C(h(x))$. We can estimate the switching surfaces between regions of the state space – each region being associated with a unique class label – by solving a binary classification problem [1] for all possible pairs of labels in L. The input data for each binary classification problem is a set of pairs (x, b_x), where b_x is one of the two labels associated with the switching surface being estimated. Standard algorithms [1] can be used to learn the switching surfaces. We model the switching surface using a two-input one-output neural network with one hidden layer, which is trained using back propagation [1]. If the functions β_i and $\mathcal{H}$ are continuous then α_i will be a continuous surface of co-dimension 1 in $\mathbb{R}^n$.

In Section VI, we describe a sensing scheme that produces a feature for any pose of the path-following robot in the world frame. Using features corresponding to a limited subset of the state space, we can construct the classifier and switching surfaces α_i for the system in (8). Figure 3 shows the the true switching surfaces (solid lines) and the estimated switching surfaces (dashed lines) determined by the classification scheme presented in Section VI. The estimated surfaces are close to the true ones, but do not match them exactly. This mismatch demonstrates the need for robustness to uncertainty in the switching surfaces. Figure 3 also depicts the vector fields induced by the classifiers when the path has zero curvature ($\rho = 0$).

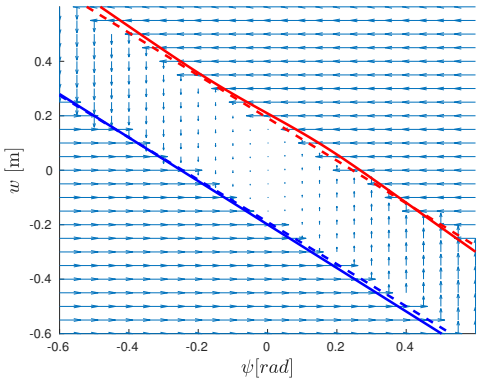


Fig. 3: The switching surfaces divide the state space into three regions, one for each label b_F, b_L and b_R. The red solid line is the true switching surface between labels b_F and b_L; the dashed red line is the estimate of that surface obtained from limited data. The blue solid line is the true switching surfce between labels b_F and b_R; the dashed blue line is the estimate of that surface from limited data. We also depict the resulting discontinuous closed-loop vector field when $\rho = 0$.

V. Performance In Unseen Environments

The previous section shows how to obtain the switching surfaces $\alpha_i(x) = \gamma_i(x, p^*) = 0$ in X for a fixed environment p^*, which yields the nominal system (6). Analysis of the nominal system may lead to the conclusion that the closed-loop system possesses well-behaved trajectories. For example, the trajectory $x(t)$ starting at any initial state $x(t_0)$ may be such that $x(t)$ approaches a desired equilibrium point, or $x(t)$ never reaches an unsafe region of the state space. As described at the end of Section III, this analysis may not hold due to the limitations of estimating the closed-loop dynamics from finite data.

We refer to the ability of a classifier-based control to yield similar behavior in environments not represented in the data as *generalization*.

Definition 3 (Generalization)**.** Let the controller G and classifier C result in a closed-loop system (6) based on the data D. Let this system satisfy control property M. We say that the classifier-in-the-loop control defined by G and C *generalizes* to the set P if the resulting closed-loop systems (5) defined for $p \in P$, where $P \ni p^*$ also satisfy property M.

In this section, we will describe conditions under which the control G and the classifier C defined by data D generalizes.

The vector fields and switching surfaces for (5) may be viewed as small perturbations of those in (6), when p and p^* are sufficiently close. Recall that since switching surface α is estimated from data, it is possible that the (6) is also a perturbation of the true switched system corresponding to p^*. If the system (6) is *structurally stable* [2], [5], [13] to small perturbations of its vector fields and switching surfaces, then we can expect that the classifier defined using data collected in p^* will yield similar behavior in (5).

Consider a nominal system

$$\dot{x} = \begin{cases} f^1(x) & , \text{ if } \alpha(x) \geq 0 \\ f^2(x) & , \text{ if } \alpha(x) \leq 0, \end{cases} \tag{9}$$

and the perturbed system

$$\dot{x} = \begin{cases} f^1(x) & , \text{ if } \tilde{\alpha}(x) \geq 0 \\ f^2(x) & , \text{ if } \tilde{\alpha}(x) \leq 0. \end{cases} \tag{10}$$

We quantify the size of perturbations in the next two definitions.

Definition 4 (δ-Perturbation of a Vector Field)**.** A vector field $\tilde{f}$ is a *δ-perturbation of a vector field f* if $d_{\mathcal{C}^1}(\tilde{f}, f) < \delta$.

Definition 5 (δ-Perturbation of a Switching Surface)**.** A switching surface $\tilde{\alpha}(x) = 0$ is a *δ-perturbation of a switching surface* $\alpha(x) = 0$ if α and $\tilde{\alpha}$ are δ-close to rank r, where $r \geq 2$.

Based on the definitions of δ-perturbations above, we define what it means for two systems to be similar.

Definition 6 (δ-closeness of systems)**.** The systems in (9) and (10) are δ-close if

- $\tilde{f}^1$ is a δ-perturbation of f^1,
- $\tilde{f}^2$ is a δ-perturbation of f^2, and
- $\tilde{\alpha}$ is a δ-perturbation of α.

The notion of δ-closeness can be interpreted as the definition of a topology on the space of switched dynamical systems [2]. The definition of structural stability follows [2].

Definition 7 (Structural Stability)**.** System (9) is structurally stable in region $\overline{W}$ bounded by a cycle without contact Γ if $\exists \delta > 0$ such that any dynamic system $(\tilde{A})$ δ-close to (9) has the same topological structure as (9) in $\overline{W}$.

We now present the main result that connects the structural stability of closed-loop systems corresponding to data and the ability of the control and classifier defined using that data to generalize.

Theorem V.1. *Let* (6) *be the closed loop system due to classifier C and control G. Let* (6) *be structurally stable. If f^i and γ are in $\mathcal{C}^1$, then the control G and classifier C generalizes to the set $B_\epsilon(p^*)$ for some $\epsilon > 0$.*

Proof. For sufficiently small $\epsilon > 0$ there exists $\delta > 0$ such that the vector fields $f^i(x, p)$ and switching surface $\gamma(x, p) = 0$ are δ-perturbations of $f^i(x, p^*)$ and $\gamma(x, p^*) = 0$ respectively, if $\|p - p^*\| < \epsilon$. That is, for $p \in B_\epsilon(p^*)$, (5) is a sufficiently small perturbation of (6). Since (6) is structurally stable, the trajectories of (5) are similar to that of (6), and therefore satisfy the same control properties as (6). □

Remark 1. The work in [5] provides conditions under which a two-dimensional switched system is structurally stable under arbitrary perturbations of the vector fields. The case of perturbations of the switching surface is not dealt with in [2], [5]. In general, structural stability to perturbations of switching surfaces has received little attention when compared

to structural stability to perturbations of the vector field [11], [12]. Some work exists for variable switching surfaces [4], [12], but these works assume that sliding motions do not occur. For a smooth-enough switching surface α and vector fields f^1 and f^2, structural stability of (9) to perturbations of vector fields implies structural stability of (9) to small perturbations of the switching surfaces. This conclusion allows us to use the existing work in [5] in the example we provide.

The dynamics in (8) defined by the estimated surfaces in Figure 3 will possess a unique equilibrium on one of the switching surfaces for all environments with non-zero curvature. Using techniques from [5], we can determine that the equilibrium is a structurally stable focus. Moreover, the asymptotic stability of the equilibrium is global. Therefore, we can expect that the dynamics in nearby curvatures of the same sign will also be asymptotically stable.

For the case of zero curvature, there is an infinite set X_e of equilibria lying on the line $\psi = 0$ in the region corresponding to the label b_F. These equilibria are not structurally stable to perturbations of the vector field, and therefore Theorem V.1 does not apply. The intersections of the two switching surfaces with the line $\psi = 0$ also belong to X_e. These two equilibria are attractive, in fact all trajectories not starting in X_e will reach one of these two equilibria. They are not stable in the sense of Lyapunov. The case of zero curvature represents a bifurcation of the closed-loop dynamics corresponding to the curvature. The unique equilibrium point that exists for positive curvature undergoes a bifurcation into a linear singularity, with the end points on the switching surfaces, when the curvature becomes zero. When the curvature is negative, the line of singularity disappears and the equilibrium on the opposite surface persists. In effect, the equilibrium 'jumps' from one switching surface to the other as the curvature changes sign.

VI. Simulation

We simulate a robot moving through a terrain using a camera and structured light for sensing. The control is implemented as an end-to-end machine learning solution as depicted in Figure 2. The structured light involves a planar pencil of light rays, where the plane containing all the light rays makes an angle with respect to the horizontal (see Figure 4). The image captured by the camera will contain the reflections of the light rays off the objects on the terrain. These reflections are captured as pixels in the image, and it the image coordinates of these pixels are used for classification. This feature vector has dimension 420. We implement the simulation in Matlab.

We define the terrain in terms of the elevation at each point $(x_1, x_2) \in \mathbb{R}^2$. The shape of the terrain is such that the lowest points of the terrain define a path in two dimensions (see Figures 5b and 5a). An example of a terrain is determined by the elevation map $Te\colon \mathbb{R}^2 \to \mathbb{R}$ given by

$$Te(x_1, x_2) = x_1^2. \tag{11}$$

We assume that a camera attached to a mobile robot is moving in the horizontal plane, with optical axis in the horizontal plane. As mentioned, the camera is used to capture the reflections of a pencil of lasers beamed onto the terrain. The feature to be classified corresponds to the set of coordinates of pixels corresponding to the reflected structured light. We assume that the number of such pixels is constant.

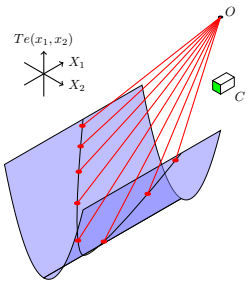
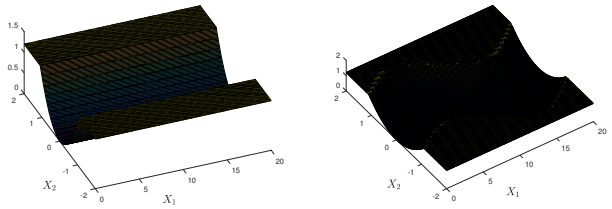


Fig. 4: A sketch of the sensing mechanism used in the simulation. A planar pencil of rays are emitted from the point O. These rays intersect the terrain (blue surface) at the red points, which can be seen by the camera C (located directly below O). Figure is best viewed in color.



(a) Training (b) Testing

Fig. 5: Visualization of a) the terrain in which we collect data to train the classifier and b) the terrain in which we test that classifier.

The terrain represented by (11) is used to collect the training data (see Figure 5a). Recall that the training data consists of triples of the form (x, ϕ, b). The mobile robot moves in a direction parallel to the path defined (11). At each point along this path, three features are collected. The three data sets correspond to the angle between the camera's optical axis and the tangent to the path being $+30°$, $0°$ and $-30°$ respectively. The features obtained in these positions are labeled as b_R, b_F and b_L respectively. This simulates the states corresponding to the data collected in [6].

The features corresponding to each label are observed to be clustered together. The clusters corresponding to the labels are easily separated, and thus a simple nearest-neighbor classification scheme is sufficient.

We then test the classifier obtained from the training data on a path defined by the equation

$$Te(x_1, x_2) = \left(x_2 - 0.5 \sin \frac{x_1}{3}\right)^2. \tag{12}$$

Figure 5b depicts the resulting terrain. This terrain defines a path consisting of sections with alternating sign of curvature. We simulate a mobile robot using this classifier to navigate in the terrain defined by (12). The resulting trajectory in

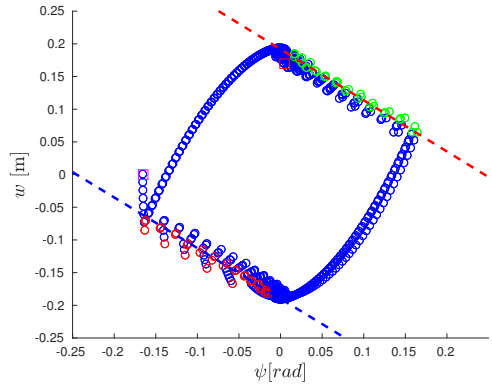


Fig. 6: Trajectories in the state space for the simulation of path-following control. Each circle represents the state at a unique instant of time. The color of the circles indicates the classifier output for the feature observed in that state. The blue circles correspond to b_F, the red circles to b_L and the green circles to b_R. The dashed lines represent the switching surfaces estimated from the training data.

the state space is plotted in Figure 6. Each circle denotes the state (ψ, w) at a particular instant of time. The color of the circle indicates the class label obtained at that instant of time. The estimated switching surfaces are also plotted. The use of a classifier to control a continuous-state dynamical system indeed results in a switched dynamical system as predicted. Moreover, the switching surface on which the trajectory slides appears to vary by a small amount, in the sense of Definition 5 for small values of δ. During sections of the path that have nearly non-zero constant curvature, the state tends to the unique equilibrium on one of the switching surfaces. Since the path is sinusoidal, it will straighten out briefly after, during which the robot moves according to u_{b_F} (blue circles). This forward motion continues until it reaches a portion of the terrain with the opposite curvature. The equilibrium is now on the opposite switching surface, and the trajectory approaches this equilibrium until the curvature changes sign once more. Note that the training data for the classifier does not contain data points corresponding to non-zero curvature at all. The structural stability of the non-zero curvature dynamics consisting of switching surfaces defined by the training data predicts the observed asymptotic stability of the equilibria.

VII. Conclusion

We showed how a classifier-in-the-loop system can be modeled as a switched dynamical system. Moreover, we proposed a method to estimate the switching surfaces in the state space that are defined by the classifier using the training data. Due to the potential errors in estimation, and the finite amount of training data, the stability properties of the estimated switched system do not automatically hold for states or environments not represented in the data. In order to address this issue, we proposed using the concept of structural stability to determine whether the results of the analysis of the nominal system will hold for environments not represented in the data. The usefulness of this approach is demonstrated in a simulation example consisting of a robot following a path in two dimensions.

This paper has presented one approach to the analysis of systems that are controlled using data. There are a few issues that we would like to address in the future. First, we want to develop methods that quantify the structural stability of the dynamics. For example, we would like to predict how small the curvature of the two-dimensional path can be without losing closed-loop asymptotic stability, given a classifier trained in paths with large curvature. Second, a theory that addresses how the structure of the classifier influences the structural stability of the closed loop dynamics is required. Third, a method to synthesize the values of the control assigned to each class label may improve the structural stability of the closed-loop system. Finally, a notion of robustness of classifier-in-the-loop systems to perturbations of the switching surfaces only may provide more appropriate tools for certifying the behavior of classifier-in-the-loop systems.

References

[1] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[2] A. A. Andronov. *Theory of bifurcations of dynamic systems on a plane*. Wiley, 1971.

[3] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM.

[4] M. di Bernardo and K. Camlibel. Structural stability of boundary equilibria in a class of hybrid systems: Analysis and use for control system design. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 215–220, Dec 2005.

[5] Aleksej Fedorovi Filippov and F. M. Arscott. *Differential equations with discontinuous righthand sides*. Mathematics and its Applications. 1988.

[6] A. Giusti, J. Guzzi, D. C. Cirean, F. L. He, J. P. Rodrguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, July 2016.

[7] R. Goebel and R.G. Sanfelice. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012.

[8] Gregory Kahn, Tianhao Zhang, Sergey Levine, and Pieter Abbeel. PLATO: policy learning using adaptive trajectory optimization. *CoRR*, abs/1603.00622, 2016.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[10] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17(1):1334–1373, January 2016.

[11] S. Pettersson and B. Lennartson. Stability and robustness for hybrid systems. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 2, pages 1202–1207 vol.2, Dec 1996.

[12] S. Prajna and A. Papachristodoulou. Analysis of switched and hybrid systems - beyond piecewise quadratic methods. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 4, pages 2779–2784 vol.4, June 2003.

[13] Jorge Sotomayor and Ana Lúcia F. Machado. Structurally stable discontinuous vector fields in the plane. *Qualitative Theory of Dynamical Systems*, 3(1):227–250, 2002.