Guaranteed Safe Online Learning of a Bounded System

Jeremy H. Gillula[†] and Claire J. Tomlin

Abstract— **For some time now machine learning methods have been widely used in perception for autonomous robots. While there have been many results describing the performance of machine learning techniques with regards to their accuracy or convergence rates, relatively little work has been done on developing theoretical performance guarantees about their stability and robustness. As a result, many machine learning techniques are still limited to being used in situations where safety and robustness are not critical for success. One way to overcome this difficulty is by using reachability analysis, which can be used to compute regions of the state space, known as reachable sets, from which the system can be guaranteed to remain safe over some time horizon regardless of the disturbances. In this paper we show how reachability analysis can be combined with machine learning in a scenario in which an aerial robot is attempting to learn the dynamics of a ground vehicle using a camera with a limited field of view. The resulting simulation data shows that by combining these two paradigms, one can create robotic systems that feature the best qualities of each, namely high performance and guaranteed safety.**

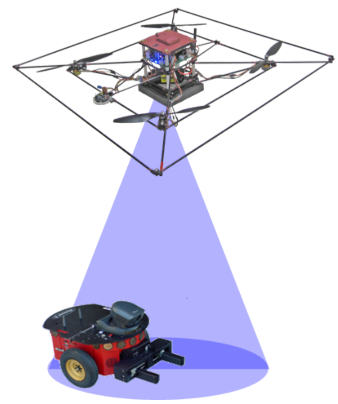

Fig. 1. Tracking a ground target using an aerial vehicle is just one example of an application in which machine learning techniques can be combined with reachability-based safety analysis to yield both superior performance and safety.

I. INTRODUCTION

For some time now machine learning methods have been widely used in perception for autonomous robots. For example, the winner of the 2005 DARPA Grand Challenge used a novel online machine learning algorithm to learn drivable terrain [1]. Machine learning has also been extended to controlling autonomous vehicles, as in the recent work done on using apprenticeship learning to control an RC helicopter [2]. And of course, machine learning has been used for over a decade in complicated path planning problems [3].

While there have been many results describing the theoretical performance of machine learning techniques with regards to their accuracy or convergence rates [4], [5], relatively little work has been done on developing theoretical performance guarantees about their stability and robustness. (One notable exception is recent work analyzing how controller parameterizations affect convergence rates and performance in reinforcement learning algorithms [6].) As a result, many machine learning techniques are still limited to being used in situations where either safety and robustness are not critical for success, or a large number of trials can be conducted to measure the safety of the system prior to deploying it in the field. This is a particularly difficult limitation for online machine learning algorithms, for example in a reinforcement learning scenario where a robot is learning a model and must simultaneously make decisions about how to act given that model. In such a scenario a spurious sample could cause the robot's model to temporarily be incorrect, causing it to take an unsafe or even catastrophic action.

One way to prevent such a catastrophic action is through the use of a technique from the control theory literature known as reachability analysis. Reachability analysis treats disturbances and controls to the system as opposing players in a dynamic game, where each player has different goals (i.e. driving the system to some unsafe region of the state space or staying outside of such an unsafe region). By using models of the system dynamics, as well as assuming that the control inputs and disturbances are bounded, one can compute regions of the state space, known as reachable sets, from which the system can be guaranteed to remain safe over some time horizon regardless of the disturbances [7]. This technique has been previously used in the robotics literature to guarantee the safety of a wide variety of tasks, including generating guaranteed safe switching regions for a collision avoidance controller for manned and unmanned aircraft [8], [7], [9], analyzing autoland systems and ultra-close formation flight [10], [11], and designing guaranteed safe aerobatic maneuvers for a quadrotor helicopter [12]. This wide range of applications demonstrates that in addition to being a rigorous way to provide provable guarantees about system safety, reachability methods are flexible enough to be adapted to many different system models and applications. With that said, reachability techniques also have some drawbacks in the same way that machine learning techniques do. For example, posing a problem successfully in a reachability

[†]Funding provided by a National Science Foundation graduate research fellowship.

J. Gillula is with the Computer Science Department, Stanford University, Stanford, CA, 94305-4035 (email: jgillula@cs.stanford.edu)

C. Tomlin is with the Department of Electrical Engineering and Computer Science, UC Berkeley, Berkeley, CA, 94720-1770 (email: tomlin@eecs.berkeley.edu)

framework usually requires some amount of domain-specific knowledge [12]. Additionally, the number of calculations required to compute the reachable set suffers from the curse of dimensionality, growing exponentially as the number of state dimensions increases [7].

Fortunately, the strengths and weaknesses of machine learning and reachability analysis are mostly complementary. Online machine learning, for example, is very successful at achieving high performance on high dimensional tasks without needing a great deal of a priori domain-specific knowledge, but suffers from an inability to make guarantees about its safety and robustness except in a statistical sense. Meanwhile reachability techniques, as mentioned above, have difficulty with high dimensional systems and require a reasonable amount of domain-specific knowledge to be useful, but provide the ability to make absolute safety guarantees. This paper proposes combining these two paradigms to create a robotic system that features the strengths of each. While the general idea of combining statistical learning techniques with control theory techniques has been proposed before [13] we believe the method proposed in this paper is more general. In particular we will show how with relatively modest assumptions about the system in question one can use reachability analysis to prevent a system from taking unsafe actions, while using machine learning to improve the performance of the system online.

We will apply this idea to the problem of autonomously learning the dynamics of a target vehicle. We will assume a scenario in which we have an aerial vehicle equipped with a downward pointing camera with a limited field of view, and this aerial vehicle is attempting to use this camera to learn the initially unknown dynamics of a target ground vehicle (see Figure 1). In this scenario, safety is defined as always keeping the target vehicle inside the camera's field of view, even under the presence of disturbances (such as measurement noise and the initially unknown dynamics of the target).

While there is a significant body of literature on autonomous tracking applications, to our knowledge none of the previous work simultaneously combines online learning of a target's dynamics with guarantees about keeping the target in view. For example, some prior work has focused on reactive motion strategies for tracking unpredictable targets [14], [15]. Other work has been done which formally accounts for uncertainty in the position and trajectory of the vehicle being tracked [16]. Finally, the prior work most similar to our approach is that of Guibas et al., which uses multiple observers and computes a visibility region for each one, and seeks to guarantee that the evader (target) will eventually lie in one of the visibility regions [17]. This paper can be seen as an extension of that work, however in addition to incorporating machine learning for better tracking of the target, we will also use a different methodology for computing the visibility regions (and will formally take into account the dynamics of the target and observer).

The organization of the rest of this paper is as follows. Section II describes the model we use for our example application. Section III then briefly reviews the machine learning and reachability analysis techniques we use later in the paper, and describes how we combine the two methodologies in order to provide superior safety and performance. Section IV explains the details of several simulations of the example application, and provides a comparison of the performance of three different control schemes (using machine-learning-based control only, using safety-from-reachability-analysis-based control only, and using a hybrid scheme). Finally, Section V concludes with some general statements about the presented methods, as well as what directions we expect to explore in the future.

II. PROBLEM SETUP

A. System Dynamics

To show how to combine these two disparate methodologies, it is helpful to have an illustrative example. As described at the end of the Introduction, we will be using a scenario in which we have an aerial vehicle equipped with a downward pointing camera (hereafter referred to as the observer) attempting to track and learn the dynamics of a ground vehicle (hereafter referred to as the target). For simplicity, we assume that the horizontal dynamics of the observer can be described by a second-order integrator and that the altitude dynamics of the observer can be described by a first-order integrator, i.e.

$$\ddot{x}_o = u_{o_x},\, \dot{y}_o = u_{o_y} \tag{1}$$

where x_o is the observer's horizontal position, y_o its vertical altitude, and $\mathbf{u_o} = \begin{bmatrix} u_{o_x} & u_{o_y} \end{bmatrix}^\top \in \mathcal{U}_o$ are its control inputs which are bounded and may be functions of the observer's current state, the current measurement and measurement history of the target's position as defined below, and time.

For the target we also assume that the dynamics are a second-order integrator, with some bounded, potentially time-varying input $u_t(x_t, \dot{x}_t, t) \in \mathcal{U}_t$:

$$\ddot{x}_t = u_t(x_t, \dot{x}_t, t) \tag{2}$$

where x_t is the target's horizontal position (with an assumed constant altitude of zero). [1]

B. Measurement Model

Next, we assume that the observer's measurement model is such that it has a limited field of view, θ, so that the target is only in view when $x_t \in \mathcal{V}$ with $\mathcal{V} = \{x_t : \|x_o - x_t\| \leq y_o \tan\theta\}$. We also assume that the measurements are subject to truncated additive Gaussian noise proportional to the observer's altitude. Then the measurement model can be described by:

$$\tilde{\mathbf{x}}_\mathbf{t} = \begin{cases} \mathbf{x_t} + \eta & \text{if } x_t \in \mathcal{V} \\ \emptyset & \text{otherwise} \end{cases} \tag{3}$$

[1]Note that for the purposes of this example we have assumed that the vehicles travel in a 2D world, with the ground vehicle constrained to a 1D line at $y = 0$. Although the methods described would easily generalize to a full 3D world, using a 2D world will make the visualization of the reachable sets found in Section III-B considerably easier.)

where $\eta \sim \mathcal{N}(0, \sigma)$ is the noise, with η bounded to between $\pm 3\sigma$, and $\sigma \propto y_o$. Note that under this measurement model there is a natural tension between the goal of safety (i.e. keeping the target within the field of view by keeping the field of view as large as possible, by increasing altitude y_o since $\|\mathcal{V}\| \propto y_o$), and reducing the noise on measurements (by reducing the observer's altitude, since $\sigma \propto y_o$).

C. Problem Statement

This natural tension also arises in the problem statement, which can be phrased in two parts:

1) Minimize the prediction error $e = \|\hat{x}_t - x_t\|$ between the predicted position of the target and its actual position, and,
2) Maintain the target in view at all times, i.e. $x_t \in \mathcal{V}$.

Part (1) can be readily solved using machine learning (i.e. by learning the system dynamics of the observer, and then using the learned model to predict the target's motion), while part (2) can be solved using reachable sets to determine which control inputs will maintain safety. The next section describes in more detail how we apply these two paradigms to the problem at hand, and how they can be successfully combined.

III. FUSING MACHINE LEARNING AND REACHABILITY ANALYSIS

This section briefly reviews the relevant tools from reachability analysis and machine learning which were used in the example system. Due to space constraints only a limited amount of material can be presented, and the reader is encouraged to peruse the cited references for more detail.

A. Machine Learning

The purpose of the learning algorithm in this scenario is to enable better prediction of the target vehicle's future position by identifying its underlying model. To accomplish this task a very simple form of online machine learning was used based very closely on the work done by Hoburg and Tedrake [18]. We began by assuming that the target's motion model could be described by a linear combination of N basis functions $\phi_i(\mathbf{x_t}, t)$, each of which could be a potentially nonlinear function of the target's state variables as well as time. The target's motion model can then be written as

$$\ddot{x}_t = \sum_{i=0}^{N} W_i \phi_i(\mathbf{x_t}, t) = [W_1 \dots W_N] \begin{bmatrix} \phi_1 \\ \dots \\ \phi_N \end{bmatrix} \quad (4)$$

where the W_i are weights indicating how much the ith basis function ϕ_i contributes to the target's control input u_t. We then determine the optimal values of these weights by taking a random subset of all the measurements $\tilde{\mathbf{x}}_\mathbf{t}$ (the training set), calculating a least squares fit for every possible combination of the N basis functions on this training set, and then choosing the set of basis functions (and the corresponding weights) which produce the smallest mean-squared error on the remainder of the measurements (the testing set).

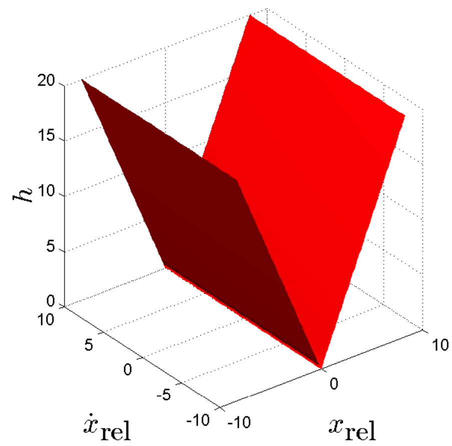

Fig. 2. The initial keep-out $\mathcal{K}$ set for the augmented system. Points between the two half-planes are safe, while points outside are unsafe.

B. Reachability Analysis

To guarantee safety using reachability analysis, we use the same approach as in [7]. We begin by defining an augmented system $\mathbf{x} = \begin{bmatrix} x_o - x_t & \dot{x}_o - \dot{x}_t & y_o \end{bmatrix}^\top$, with system dynamics

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, d) = \begin{bmatrix} \dot{x}_o - \dot{x}_t \\ u_{o_x} - u_t \\ u_{o_y} \end{bmatrix} \quad (5)$$

where the control input to the augmented system is the bounded control to the observer $\mathbf{u} = \begin{bmatrix} u_{o_x} & u_{o_y} \end{bmatrix}^\top \in \mathcal{U}_o$, and the disturbance input to the augmented system is the bounded control to the target $d = u_t \in \mathcal{U}_t$.

We can then describe a keep-out set $\mathcal{K} = \{\mathbf{x} : x_t \notin \mathcal{V}\}$, which represents all the points in the state space of the augmented system in which the target is outside of the field of view of the observing vehicle (see Figure 2). The backwards reachable set $Pre_\tau(\mathcal{K})$ is then defined as the set of all states $\mathbf{x}$ such that, for any control input $\mathbf{u}$, the disturbance d can drive the system state into the avoid set $\mathcal{K}$ in some time τ. Thus, as long as we guarantee the state of the augmented system always remains outside $Pre_\tau(\mathcal{K})$, we can guarantee the system will stay safe (the target will remain in the observer's field of view) for the next τ units of time.

To calculate $Pre_\tau(\mathcal{K})$, we define a cost function $l(\mathbf{x})$ such that $l(\mathbf{x}) < 0$ for $\mathbf{x} \in \mathcal{K}$ and $l(\mathbf{x}) > 0$ for $\mathbf{x} \notin \mathcal{K}$. Following [7], we can then formulate the problem as a dynamic game with cost function

$$J(x, t) = \max_{\mathbf{u} \in \mathcal{U}_o} \min_{d \in \mathcal{U}_t} l(\mathbf{x}(0)) \quad (6)$$

where the disturbance is trying to steer the augmented system state into the keep-out set by minimizing $l(\mathbf{x})$, and the control is trying to keep it out of $\mathcal{K}$ by maximizing $l(\mathbf{x})$. To solve this game one needs the optimal Hamiltonian

$$H^*(\mathbf{x}, p) = \max_{\mathbf{u} \in \mathcal{U}_o} \min_{d \in \mathcal{U}_t} p^\top f(\mathbf{x}, \mathbf{u}, d) \quad (7)$$

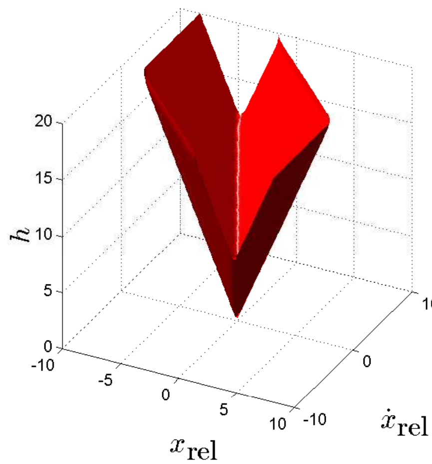

Fig. 3. The unsafe set $Pre_\tau(\mathcal{K})$ for the augmented system after $\tau = 1$ second. Points within the surface are safe, while points outside the surface are unsafe.

where $p = \frac{\partial J}{\partial \mathbf{x}}(\mathbf{x}, t)$ is the Hamiltonian costate. Note that solving for the optimal Hamiltonian provides the optimal control actions $u^*_{o_x}$ and $u^*_{o_y}$ which must be taken to maintain safety (as well as the optimal disturbance action u^*_t which the target will use to try to make the system unsafe). Equipped with these expressions, we can then make use of the Level Set Toolbox developed at the University of British Columbia [19] to calculate the unsafe set $Pre_\tau(\mathcal{K})$ (see Figure 3).

C. Fusing Machine Learning and Reachability Analysis

The key observation that enables us to combine these two paradigms is that in order to maintain safety, the system only needs to use the optimal control actions $u^*_{o_x}$ and $u^*_{o_y}$ when the augmented system state is near the border of the unsafe set $Pre_\tau(\mathcal{K})$ [9]. At all other times the observer vehicle is free to use whatever control actions it likes, including taking the action that will optimally reduce the amount of sensor noise in its measurements so that it can learn a better dynamic model of the target vehicle's motion. As would be intuitively expected in this particular scenario the action that will optimally reduce the observer's measurement noise as much as possible is to reduce the observer's altitude as quickly as possible. This can be easily verified by computing the conditional entropy as in [20], which is given by the entropy formula for Gaussians [21]:

$$H(x_t|x_o) = \frac{1}{2}\log\left(2\pi e\left|\Sigma^{-1} + \left(\sigma^{-2}y_o^{-2}\right)\right|^{-1}\right) \quad (8)$$

where $p(x_t)$ is approximated as a Gaussian with covariance Σ. Since the conditional entropy can be rewritten as:

$$H(x_t|x_o) = \frac{1}{2}\log 2\pi e - \frac{1}{2}\log\left|\Sigma^{-1} + \sigma^{-2}y_o^{-2}\right| \quad (9)$$

and since the logarithm is an increasing function, one can then minimize the conditional entropy (and thus reduce the uncertainty in the position of the target) by maximizing the quantity $\Sigma^{-1} + \sigma^{-2}y_o^{-2}$ which can be achieved by making the altitude y_o as small as possible.

IV. SIMULATION RESULTS

A. Simulation Setup

To show the benefits of this new paradigm, several simulations of the above system were conducted. In each simulation, the observer and target dynamics were simulated using a numerical integration scheme, and the observer was allowed to take measurements at fixed time intervals. For simplicity the target's controller was chosen to be a subset of the basis functions ϕ_i which the observer would use for its learning algorithm. For comparison, three different control schemes for the observer were run on the same set of measurement data:

1) Control based solely off online learning of the target's motion,
2) Control based solely off the reachability analysis for safety, and
3) A hybrid controller based off online learning of the target's motion, except when the optimal safety inputs were required (as determined by the reachable set calculations described in the previous section).

For the first control scheme the observer was allowed to keep a history of all of its previous measurements of the target's position and velocity. Then after each sensor measurement the observer ran the machine learning algorithm described in Section III-A to estimate the target's motion model. It then predicted what the position of the target would be at the next measurement time, and used a simple PD controller in the horizontal direction to control to that point (until the next measurement could be made). In the vertical direction, it would reduce its altitude as much as possible in order to reduce the noise on its future sensor measurements.

For the second control scheme the observer used the same controller in both the horizontal and vertical direction as in the learning-only control scheme, except that in this case, the reference point for the horizontal PD controller was simply the last (noisy) measurement of the target's position and velocity. Additionally, when the target was detected as being within some threshold of the boundary of the unsafe set calculated in Section III-B, the optimal control input from the optimal Hamiltonian was instead substituted until the next sensor measurement could be taken.

Finally, the third control scheme constituted a hybrid of the two previous control schemes. As with the first and second control scheme, a PD controller was used in the horizontal direction, and the target attempted to decrease altitude in the vertical direction. As with the first control scheme, the machine learning algorithm was run after every sensor measurement and the result was used to better track the target. As with the second control scheme, the optimal control input from the optimal Hamiltonian was used instead of the nominal control whenever the target was detected as being within a given threshold of the boundary of the unsafe set.

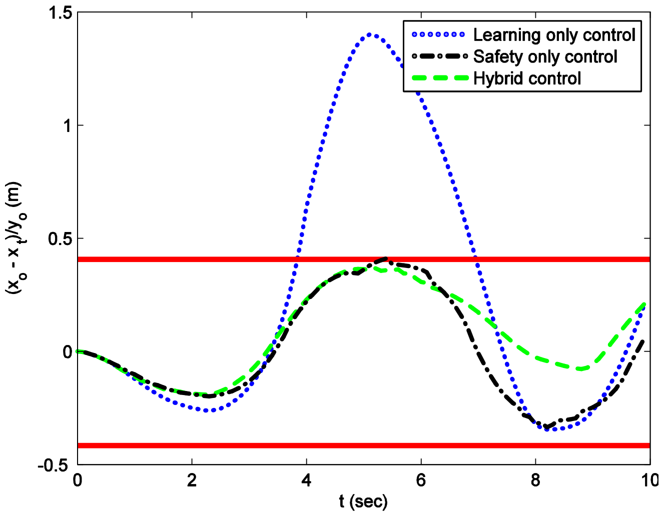

Fig. 4. Graph of the relative position of the target within the observer's field of view, normalized by the observer's altitude. The solid red lines indicate the boundaries of the field of view.

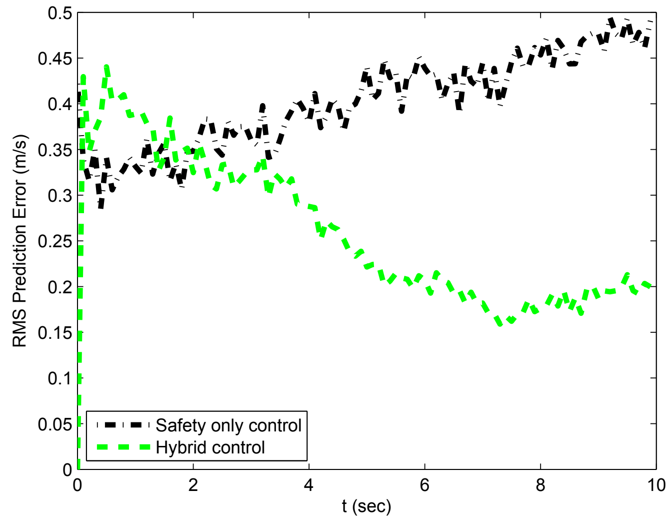

Fig. 5. The RMS prediction error of the target's velocity as a function of time, averaged over 300 simulation runs.

Additionally it should be noted that in order to prevent the tracking problem from becoming trivial, the target was allowed a greater maximum horizontal acceleration than the observer, but not so great that the observer would not be able to keep the target in its field of view by increasing altitude.

B. Simulation Results

Three hundred simulations runs were conducted using the framework described above, and the results are described below.

1) Safety: Figure 4 shows the safety results of a typical run of the simulated system under each of the three control schemes, by plotting the relative position of the target and the observer, normalized by the observer's altitude, versus time. The solid red lines indicate the boundaries of the observer's field of view, $\arctan \pm\theta$; the system is safe when the relative position is between these two lines. As would be expected given that the target vehicle had a greater potential horizontal acceleration than the observer, safety was only maintained when the observer's controller included the reachability-based control scheme. This can be seen by the failure of the learning-only control scheme to maintain safety: the observer lost track of the target around $t = 4$ sec because the target was able to accelerate in the horizontal direction faster than the observer, and the observer did not attempt to maintain safety by increasing its altitude. Conversely, in both the safety-only control scheme and the hybrid control scheme, the observer was able to detect when the relative position and velocity of the target was near the boundary of the unsafe reachable set pictured in Figure 3 and take the appropriate control action to maintain safety. (Note that this behavior was consistent throughout all 300 simulation runs.)

2) Model Learning: Figure 5 shows the RMS prediction error of the hybrid control scheme and the safety-only control scheme (averaged over the 300 simulation runs) as a function of time. (Note that the prediction error for the learning-only control scheme is not plotted, since the learning-only control scheme always lost track of the vehicle and its prediction error thus diverged due to the lack of measurements.) Due to the time-varying dynamics chosen for the target, the target increases the magnitude of its acceleration (within the specified bounds $\mathcal{U}_t$) as time goes on. As a result the prediction error for the safety-only control scheme actually increases over time, since it is simply predicting the target to have the same velocity as the most recent measurement (which becomes a less and less accurate assumption as t increases). The hybrid control scheme, on the other hand, initially has high error due to the lack of sensor measurements and the increased noise on the few measurements available. However, as the observer is able to decrease its altitude and collect more measurements the model improves, until it reaches a more accurate level of performance than would be possible without the machine learning.

V. CONCLUSIONS AND FUTURE WORK

This paper describes the successful combination of two major complementary methodologies in robotics: machine learning and reachability analysis. By combining these two paradigms, one can create robotic systems that feature the best qualities of each, namely high performance and guaranteed safety. We have demonstrated this combination on an example application in which an aerial vehicle is attempting to learn online the a priori unknown dynamics of a ground vehicle. The resulting simulation data shows the benefits of such a hybrid robotic control system, by showing that safety is maintained by the reachability analysis while the system's performance simultaneously improves due to the machine learning.

It is important to note that the results presented in this paper are not restricted to the specific scenario we have presented, but are a general characteristic of reachability analysis; regardless of the system, as long as both the control inputs and disturbances are bounded, the safety-enforcing control action only needs to be taken when the system is near the border of the unsafe set, and the system is free to use any other controller at other times, including any one of a variety of controllers that could be developed using machine learning.

In the future, we hope to expand on these results in a variety of ways. We are currently working on implementing the resulting algorithms on a physical testbed, namely the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control, for real-world validation [22]. Additionally we hope to be able to incorporate more challenging environments (such as obstacles and occlusions) which would simultaneously increase the practicality of our system in the real world, while making the resulting reachable set calculations potentially more difficult. A final promising research direction that we are exploring involves combining reachability-based safety techniques with reinforcement learning algorithms for autonomous data collection (i.e. the E^3-family of algorithms [23]) in order to enable a vehicle to autonomously and safely learn to control itself, without the need for advanced domain-specific knowledge.

REFERENCES

[1] H. Dahlkamp, "Selfsupervised monocular road detection in desert terrain," in *In Proc. of Robotics: Science and Systems (RSS*, 2006.

[2] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in Neural Information Processing Systems 19*. MIT Press, 2007, p. 2007.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *IEEE International Conference on Methods and Models in Automation And Robotics*. MorganKaufmann, 1997, pp. 566–580.

[4] C. J. Geyer, "On the asymptotics of constrained m-estimation," *The Annals of Statistics*, vol. 22, no. 4, pp. pp. 1993–2010, 1994.

[5] K. Knight and W. Fu, "Asymptotics for lasso-type estimators," *The Annals of Statistics*, vol. 28, no. 5, pp. pp. 1356–1378, 2000.

[6] J. Roberts, I. Manchester, and R. Tedrake, "Feedback controller parameterizations for reinforcement learning," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2011.

[7] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[8] C. J. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998.

[9] G. M. Hoffmann and C. J. Tomlin, "Decentralized cooperative collision avoidance for acceleration constrained vehicles," in *Proc. 47th IEEE Conf. Decision and Control*, Cancun, Mexico, December 2008, pp. 4357–4363.

[10] M. Oishi, I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "Invariance-preserving abstractions of hybrid systems: Application to user interface design," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 229–244, 2008.

[11] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *Proc. 47th IEEE Conf. Decision and Control*, Cancun, Mexico, 2008, pp. 3706–3712.

[12] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Proc. 2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010.

[13] C. G. Claudel, A. Hofleitner, N. Mignerey, and A. M. Bayen, "Guaranteed bounds on highway travel times using probe and fixed data," in *Transportation Research Board 88th Annual Meeting*, vol. 1, Washington, D.C., January 2009.

[14] R. Murrieta-Cid, H. H. González-Baños, and B. Tovar, "A reactive motion planner to maintain visibility of unpredictable targets," in *ICRA*, vol. 4, Washington, D.C., May 2002, pp. 4242–4248.

[15] S. M. LaValle, H. H. González-Baños, C. Becker, and J.-C. Latombe, "Motion strategies for maintaining visibility of a moving target," in *ICRA*, vol. 1, Albuquerque, NM, April 1997, pp. 731–736.

[16] P. Fabiani, H. H. González-Baños, J. C. Latombe, and D. Lin, "Tracking an unpredictable target among occluding obstacles under localization uncertainties," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 31 – 48, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/B6V16-445B587-1/2/5a5a8e8a0048304b5b91838c714bad22

[17] L. J. Guibas, J. claude Latombe, S. M. Lavalle, D. Lin, and R. Motwani, "A visibility-based pursuit-evasion problem," *International Journal of Computational Geometry and Applications*, vol. 9, pp. 471–494, 1996.

[18] W. Hoburg and R. Tedrake, "System identification of post stall aerodynamics for uav perching," in *Proceedings of the AIAA Infotech@Aerospace Conference*. Seattle, WA: AIAA, April 2009.

[19] I. M. Mitchell, *A Toolbox of Level Set Methods*, 2009, http://people.cs.ubc.ca/ mitchell/ToolboxLS/index.html.

[20] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Mutual information methods with particle filters for mobile sensor network control," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, December 2006, pp. 1019–1024.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & sons, 1991.

[22] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. 2007 AIAA Guidance, Navigation, and Control Conf.*, Hilton Head, SC, August 2007.

[23] M. Kearns and S. Singh, "Near-optimal reinforcement learning in polynomial time," *Machine Learning*, vol. 49, pp. 209–232, 2002, 10.1023/A:1017984413808. [Online]. Available: http://dx.doi.org/10.1023/A:1017984413808