Formally Verified Switching Logic for Learning Aircraft Controller

Ratan Lal[1], Daksh Shukla[2], Shawn Keshmiri[2], and Pavithra Prabhakar[1]

[1] Kansas State University
[2] Department of Aerospace, University of Kansas

Abstract

In this paper, we consider a design of hybrid controller for the linear model of aircraft dynamics to learn a better controller. The hybrid controller consists of learning based controller supported by a safety controller. Learning based controllers alone is not allowed to learn while flying the aircraft because of the following two problems. First, it may lead the system to be unsafe because of spurious training. Second, if it has been learned under the conservation safe zone, it may lead the system to be in a catastrophic failure when it will attempts to learn from less conservative safe zone. We address both the problems in the design of hybrid controller which ensures that the learning based controller is allowed most of the time to learn a better controller on the fly while guaranteeing its safe operation.

1 Indroduction

Cyber physical systems are those systems where physical systems are governed by the controllers. The physical systems are captured by differential equations, however, controllers are implemented by software. The main challenge in the cyber physical systems is to apply the controllers to the physical systems appropriately to ensure its safety, specially in safety critical systems. The crux of the problem is reach-ability analysis and finding appropriate switching between controllers.

There has been growing interest in reach-ability analysis for different class of systems. In general, it is undecidable, however, it is decidable for simple class of systems [27]. In the literature, over-approximation techniques have been studied [6, 14, 4, 21, 26, 9, 12, 19, 28, 29] that compute a super set of the exact set of reachable states. The focus is to reduce the over-approximation error. However, there is trade off between precision and computation complexities. Let a safe region be considered as the set of all states reached within time $T > 0$. Then the over-approximation of the exact reach states does contain a point that does not reach within time T. Hence, it may violate safety guarantee. In fact, an under approximation would guarantee safety, albeit be more conservative. In general, computing under-approximations is also challenging. However, we find a way for an under-approximation of the exact reach set which provides a practically interesting under-approximations of the reachable set.

Learning based controllers have been widely used in safety critical systems. It is not directly used while the system is operating due to the possibilities of making the system to be in unsafe zone. However, it has been used provided a safe zone for its application based on reach-ability analysis [11, 13]. If learning based controller has been applied for conservative safe zone, then the system may reach unsafe zone when the controller attempts to learn in a less conservative safe zone. To learn a better controller, reinforcement based controller [3] has been developed using an integration of safety and learning.

We consider a dynamical model of the aircraft dynamics which is a linear dynamics comprising of longitudinal and lateral dynamics. Our broad goal is to design hybrid controller consists of learning based controller supported by the safety controller which provides a safe zone for the learning based controller to learn better controller. To achieve this, we design the hybrid controller in such a way that the learning based controller is allowed the maximum duration of time in the safe zone while guaranteeing its operation in the safe zone.

2 Problem

We consider a linear dynamics of an aircraft comprising of longitudinal and lateral dynamics. The state variables for the longitudinal dynamics are velocity (V), Angle of attack (α), pitch angle (θ), and pitch rate (Q), and control inputs for the dynamics are throttle (δ_t) and elevator deflection (δ_E). The state variables for the lateral dynamics are angle of side-slip (β), roll angle (ϕ), roll rate (P), yaw rate (R), and control inputs for the dynamics are aileron deflection (δ_a) and rudder deflection (δ_r). Let us consider a general form for the linear aircraft dynamics as given below,

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{1}$$

where A is an $n \times n$ square matrix, and B is a $n \times d$ rectangular matrix; $\mathbf{x}$ is a vector of n state variables and $\mathbf{u}$ is a vector of d control inputs.

Problem 1. *Given System 1, a learning based controller and a safety controller, design a hybrid controller such that learning based controller is operated for a long time while its operation is guaranteed to be safe by the safety controller.*

The crux of Problem 1 is to decide the switching between the controllers such that ANN controller has been used most of the time while guaranteeing its operation to be safe. We consider Artificial Neural Network (ANN) controller for the learning based controller and Linear Quadratic Regulator (LQR) controller for the safety controller. Later, we provide the details of LQR and ANN controller. We assume that we have a known safe region, say $\mathcal{S}$, from which we know that LQR controller are safe to apply, that is, the system is safe. Hence, ANN controller is allowed to operate in the safe zone $\mathcal{S}$. Next, we need to provide the largest possible safe region for ANN controller to learn a better controller. In other words, we need to extend $\mathcal{S}$ to $\mathcal{S}'$ that we do with the help of LQR controller. This is formally formulated as follows.

Problem 2. *Given system 1, time horizon $T > 0$ and safe zone $\mathcal{S}$, find the largest safe zone $\mathcal{S}'$ such that $\mathcal{S}'$ is guaranteed to be safe zone for ANN controller, that is, the next state within T using control inputs provided by ANN controller is in $\mathcal{S}'$.*

Next, we address both the problems in the following sections.

3 Hybrid Controller Design

In this section, we address Problem 1. The hybrid controller design consists of mainly four components (a) LQR Controller (b) ANN Controller (c) Switching Logic (d) Aircraft Dynamics as shown in Figure 1 . Both LQR and ANN controller receive the current state of the aircraft and output control inputs. The crucial part of the design is the Switching Logic which ensures that ANN controller has been operated most of the time to learn a better controller. The implementation of Switching Logic is the following.

1. If the next state is within the safe zone, then control inputs produced by ANN controller are used to aircraft dynamics;

2. If the next state is outside the safe zone, then the control inputs produced by LQR controller are used to aircraft dynamics.

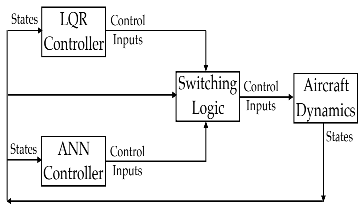

Figure 1: Hybrid Controller

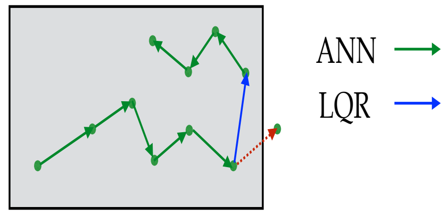

Figure 2: Safe Zone $\mathcal{S}$

The Switching Logic guarantees that the system is always in the safe zone under the operation of hybrid controller. The first point in the implementation of Switching Logic ensures that ANN controller is used most of the duration within the safe zone. The second point ensures that the system is always in the safe zone.

We illustrate a sample behavior of hybrid controller as shown in Figure 2, where rectangular region represents a safe zone $\mathcal{S}$; Arrow shows the next state from the current state. All green arrows show that control input obtained by ANN controller has been adopted from the current state. Blue arrow shows that control input obtained by LQR controller has been adopted from the current state. Note that red line has not been executed because the next state is outside the safe zone $\mathcal{S}$. Hence, LQR controller is applied at the current state for not entering the system to be in unsafe zone. Next, we provide the details about the extension of the safe zone $\mathcal{S}$ to learn a better controller.

4 Linear Quadratic Regulator Controllers

In this section, we address Problem 2. Assume that we have Linear Quadratic Regulator (LQR) controller with gain matrix K that generates control inputs $\mathbf{u}$ for a given value of $\mathbf{x}$, that is, $\mathbf{u} = -K\mathbf{x}$. We use this in Equation 1, and obtain the following equation,

$$\dot{\mathbf{x}} = (A - BK)\mathbf{x} \tag{2}$$

The solution of Equation 2 is given by $\mathbf{x}(\mathbf{x}_0, t) = e^{(A-BK)t}\mathbf{x}_0$, $\mathbf{x}_0 \in \mathcal{I}$, $t \in [0, T]$ and $\mathbf{x}(\mathbf{x}_0, 0) = \mathbf{x}_0$ for $\mathbf{x}_0 \in \mathcal{I}$, $\mathcal{I}$ is a set of initial states, where $\mathbf{x}(\mathbf{x}_0, t)$ denotes the state of the system at time t starting from state $\mathbf{x}_0$.

Our broad goal is to find the largest set $\mathcal{S}'$ of all values of state variables of the aircraft from Equation 2 such that a known safe zone $\mathcal{S}$ can be reached from $\mathcal{S}'$ within a given time horizon $T > 0$. Here T represents one sample time period or any other time for which we are willing to allow the LQR controller to operate to bring it back to the safe zone $\mathcal{S}$. The crux of the problem is backward reach-ability, which can be alternatively tackled using a forward reach-ability analysis on the following transformed equation:

$$\dot{\mathbf{x}} = -(A - BK)\mathbf{x} \tag{3}$$

Equation 3 is obtained from Equation 2 by negating the right hand side. The effect of the transformation is that the system now evolves backward in time. Our claim is that the set of states reached from Equation 3 from $\mathcal{S}$ for a given time horizon $T > 0$ is equal to the set of states $\mathcal{S}'$ that can reach $\mathcal{S}$ within time T from Equation 2.

Proposition 1. *Let* $A = \{\boldsymbol{x} \mid \boldsymbol{x} = e^{-(A-BK)t}\boldsymbol{x}_0,\ \boldsymbol{x}_0 \in \mathcal{S},\ t \in [0,T]\}$ *and* $B = \{\boldsymbol{x}_0 \mid e^{(A-BK)t}\boldsymbol{x}_0 \in \mathcal{S},\ for\ t \in [0,T]\}$. *Then we have* $A = B$.

Since under-approximations guarantee safety, we obtain an under-approximation of the set of points reachable from a set $\mathcal{S}$ within time T, by computing the reach set at some times $r, 2r, \ldots, kr = T$ and taking their union. Thus, we obtain $\mathcal{S}'$. We illustrate the safe zone

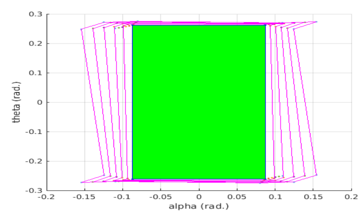

Figure 3: Longitudinal System

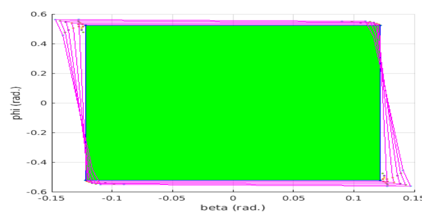

Figure 4: Lateral System

S and extended safe zone $\mathcal{S}'$ within $T = 0.05$ for angle of attach α and pitch angle (θ) of longitudinal system and angle of side slip β and roll angle ϕ of lateral system as shown in Figure 3,4, respectively. In both Figure 3,4, green region represents $\mathcal{S}$ and green region with magenta region shows $\mathcal{S}'$.

5 Artificial Neural Networks Based Flight Controller

Using neural networks for aircraft control goes far back to the year 1991 [33]. Most of the research has been concentrated around the idea of designing only the low-level control systems [33, 23, 31, 7]; others are more focused on specific topics related to system identification including parameter and model identification or inverse plant dynamics approximation [18], nonlinear aerodynamic modeling [32, 20, 17], modeling unknown and uncertain aircraft dynamics [25], unmanned aerial system (UAS) navigation or path planning [34], trajectory optimization [15], sensor validation [8], and fault detection systems [22]. Though there is ample research on application of machine learning methods or some form of learning-based control that imparts some level autonomy to drones, it is mostly concentrated around the application on rotary-wing aircraft (e.g. quadcopters) which can take a "stop-think-act" approach as part of control decisions e.g. [10, 16, 2, 1]. These systems behave as point mass objects and do not have the complexity that a fixed-wing aircraft has. A fixed-wing aircraft has to maintain a minimum velocity (above stall), has to fly at small air flow angles (e.g. stall angle of attack), and should meet the restrictive load factor which presents a great deal of challenge compared to quadcopters. The main goal of this research is develop a learning artificial neural network (ANN) autopilot for fixed-wing UASs. In standard imitation learning paradigms, researchers use expert policy decisions to train a controller. This type of approach has resulted in state-of-the-art controllers in a variety of different applications including in robotics and vehicle control e.g. [5, 24]. Here we aim to adapt this approach to fixed-wing unmanned aerial systems (UASs) where policy decisions jointly capture guidance, navigation, and control (GNC) decisions. While previous research has considered using imitation learning [30] or reinforcement learning e.g. [10, 16, 34] for parts of UAS flight control, this work is the first attempt that the authors know of to jointly apply this type of an approach to the full GNC as the main ANN autopilot for a fixed wing UAS.Figure 5 shows the structure of ANN base algorithm used in this research. Our research goal is to construct a robust neural network training algorithm that allows for imitation learning of a joint GNC UAS controller.

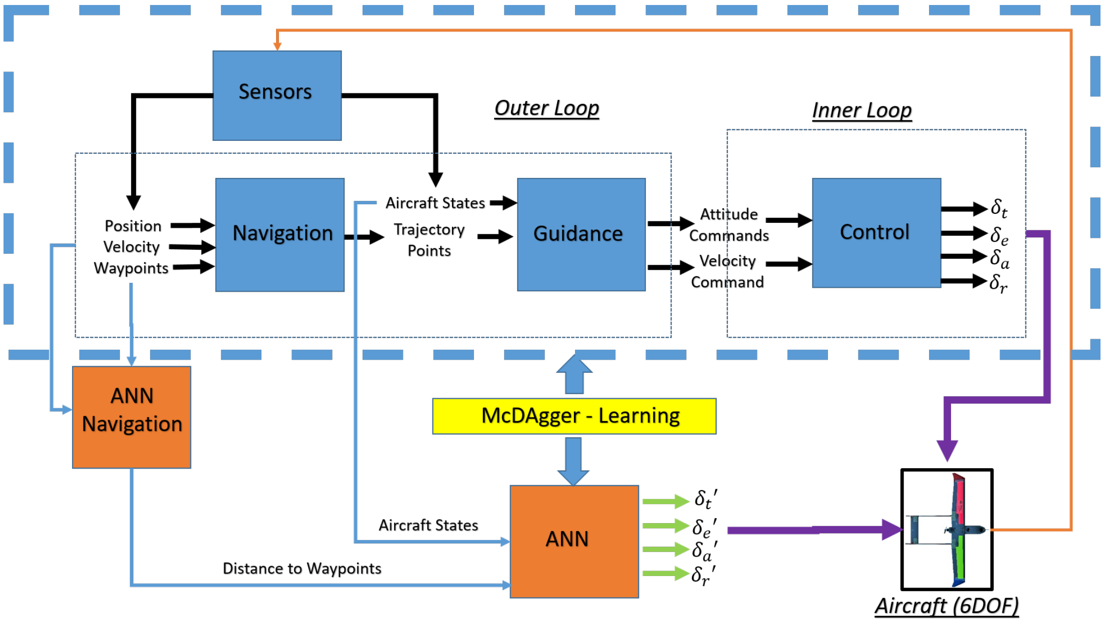

Figure 5: Standard autopilot for fixed-wing aircraft has three components that operate in sequence, requiring different inputs (arrows in) and providing different outputs (arrows out).

Our research goal is to construct a robust neural network training algorithm that allows for imitation learning of a joint GNC UAS controller. We begin by adapting an imitation learning paradigm to the context of fixed-winged aircraft, highlighting limitations of the widely used DAgger algorithm when extended to this particular autopilot scenario. We then introduce a variant of the DAgger algorithm that uses Monte Carlo sampling of initial flight conditions to build up a more robust set of training examples and evaluate the improved performance of our Monte Carlo variant (McDAgger) in our specific application. We show that our variant of McDAgger (GNC McDAgger) algorithm has a consistent increase in learning in terms of flight path, indicated by its slope and the neural network trained by this algorithm can fly indefinitely.

In the baseline model, the expert policy (GNC) is simulated to fly the aircraft using 6DOF equations, to guide the aircraft in a racetrack type trajectory following the 4 way-points as shown in figure 6. The coordinates of the 4 way-points are selected and converted from geodetic coordinates from a flight test location at Lawrence, Kansas which is utilized for research flight tests by KU flight systems team. Looking at the distance that the UAV has to cover ($\approx$ 7000ft), it takes about $\approx$ 140 to 150 seconds at a cruise speed of 50 ft/s to complete one loop. It was determined using simulations, that the GNC+6DOF system more or less converges to a repetitive state after about three and a half loops or about 525 seconds. Depending on the initial condition that the aircraft starts flying at, with respect to the way-points, it takes about 50 to 100 seconds for the aircraft to converge on to the desired trajectory and altitude. Therefore, a conservative flight time of 750 seconds was chosen as the target flight time for validation of the neural network. To collect training data, we include a GNC flight of 750 seconds sampling at 20 Hz. This initial training data is available to all of the models we consider. Regardless of training

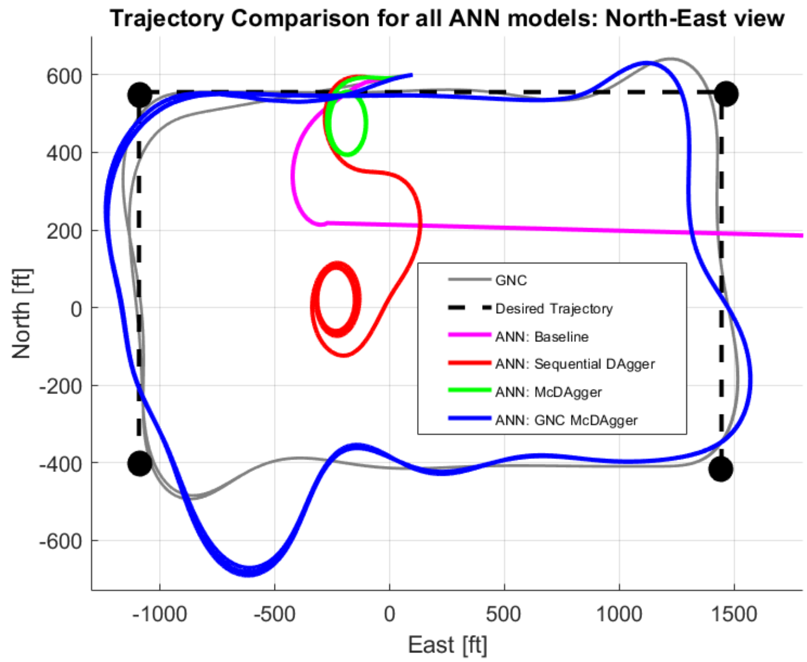

Figure 6: All models are evaluated in a simulated flight test setting to follow the dashed desired way-point path, and the corresponding aircraft trajectories are compared.

procedure, which vary based on the assumptions of our models (see below), all models are validated on their ability to fly the aircraft by sequentially choosing control states that are then fed to the 6DOF model which returns the aircraft state after executing those controls—this new state is fed back to the neural network and this loop is run until the aircraft becomes unstable or the 6DOF model cannot make a prediction given the control inputs (e.g an ANN+6DOF system). Although the GNC McDAggr ANN was capable to combine and imitate functionality of GNC modules, low frequency oscillations in the control surfaces (and consequently aircraft states) raised serious concerns on the reliability and safety of such algorithms for real world applications. During our research we found that if ANN excitation of 6DOF is limited to a portion of trajectory there will be no oscillations. Or, in other words, if the ANN is allowed to fly the aircraft for a limited time window, and data is continuously added over this window, by querying the GNC policy, then the neural network outputs have no oscillations. One other main observation was that the neural network training must be restricted to a time window until all the oscillations disappear and the ANN has learned that portion of trajectory very well. This is how the moving window DAgger concept was developed, in which the ANN was implemented to learn 10 seconds portion of trajectory around four waypoints.

References

[1] Pieter Abbeel, Adam Coates, Timothy Hunter, and Andrew Y Ng. Autonomous autorotation of an rc helicopter. In *Experimental Robotics*, pages 385–394. Springer, 2009.

[2] Pieter Abbeel, Adam Coates, and Andrew Y Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[3] Anayo K Akametalu, Shahab Kaynama, Jaime F Fisac, Melanie Nicole Zeilinger, Jeremy H Gillula, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *CDC*, 2014.

[4] Matthias Althoff, Olaf Stursberg, and Martin Buss. Reachability analysis of linear systems with uncertain parameters and inputs. In *Conference on Decision and Control*, 2007.

[5] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[6] Eugene Asarin, Olivier Bournez, Thao Dang, and Oded Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *International Workshop on Hybrid Systems: Computation and Control*, 2000.

[7] Haitham Baomar and Peter J Bentley. An intelligent autopilot system that learns piloting skills from human pilots by imitation. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pages 1023–1031. IEEE, 2016.

[8] Giampiero Campa, Mario Luca Fravolini, Marcello Napolitano, and Brad Seanor. Neural networks-based sensor validation for the flight control system of a b777 research model. In *American Control Conference, 2002. Proceedings of the 2002*, volume 1, pages 412–417. IEEE, 2002.

[9] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *International Conference on Computer Aided Verification*, 2013.

[10] Seungwon Choi, Suseong Kim, and H Jin Kim. Inverse reinforcement learning control for trajectory tracking of a multirotor uav. *International Journal of Control, Automation and Systems*, 15(4):1826–1834, 2017.

[11] Jaime F. Fisac, Anayo K. Akametalu, Melanie Nicole Zeilinger, Shahab Kaynama, Jeremy H. Gillula, and Claire J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *CoRR*, 2017.

[12] Goran Frehse, Rajat Kateja, and Colas Le Guernic. Flowpipe approximation and clustering in space-time. In *International Conference on Hybrid Systems: Computation and Control*, 2013.

[13] Jeremy H Gillula and Claire J Tomlin. Reducing conservativeness in safety guarantees by learning disturbances online: iterated guaranteed safe online learning. In *Robotics: Science and Systems*, 2013.

[14] Antoine Girard. Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*, 2005.

[15] Joseph F Horn, Eric M Schmidt, Brian R Geiger, and Mark P DeAngelo. Neural network-based trajectory optimization for unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 35(2):548–562, 2012.

[16] Jemin Hwangbo, Inkyu Sa, Roland Siegwart, and Marco Hutter. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4):2096–2103, 2017.

[17] Shawn S Keshmiri, Edward Lan, and Richard Hale. Nonlinear aerodynamics of an unmanned aircraft in wind shear. *Aircraft Engineering and Aerospace Technology*, 89(1):39–51, 2017.

[18] Byoung S Kim and Anthony J Calise. Nonlinear flight control using neural networks. *Journal of Guidance, Control, and Dynamics*, 20(1):26–33, 1997.

[19] Ratan Lal and Pavithra Prabhakar. Bounded error flowpipe computation of parameterized linear systems. In *International Conference on Embedded Software, EMSOFT*, 2015.

[20] Chuan-Tau Lan and Shawn Keshmiri. Analysis of the crash of a transport aircraft and assessment of fuzzy-logic stall recovery. *Aerospace Science and Technology*, 71:231–244, 2017.

[21] Colas Le Guernic and Antoine Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 2010.

[22] Marcello R Napolitano, Charles Neppach, Van Casdorph, Steve Naylor, Mario Innocenti, and Giovanni Silvestri. Neural-network-based scheme for sensor failure detection, identification, and

accommodation. *Journal of Guidance, Control, and Dynamics*, 18(6):1280–1286, 1995.

[23] Marsello R Napolitano and Michael Kincheloe. On-line learning neural-network controllers for autopilot systems. *Journal of Guidance, Control, and Dynamics*, 18(5):1008–1015, 1995.

[24] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile off-road autonomous driving using end-to-end deep imitation learning. *arXiv preprint arXiv:1709.07174*, 2017.

[25] Urpo J Pesonen, James E Steck, Kamran Rokhsaz, Hugh Samuel Bruner, and Noel Duerksen. Adaptive neural network inverse controller for general aviation safety. *Journal of Guidance, Control, and Dynamics*, 27(3):434–443, 2004.

[26] Pavithra Prabhakar and Mahesh Viswanathan. A dynamic algorithm for approximate flow computations. In *International Conference on Hybrid Systems: Computation and Control*, 2011.

[27] Anuj Puri and Pravin Varaiya. Decidability of hybrid systems with rectangular differential inclusion. In *International Conference on Computer Aided Verification*, 1994.

[28] Nima Roohi, Pavithra Prabhakar, and Mahesh Viswanathan. Hybridization based CEGAR for hybrid automata with affine dynamics. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2016.

[29] Nima Roohi, Pavithra Prabhakar, and Mahesh Viswanathan. HARE: A hybrid abstraction refinement engine for verifying non-linear hybrid automata. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2017.

[30] Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J Andrew Bagnell, and Martial Hebert. Learning monocular reactive uav control in cluttered natural environments. In *2013 IEEE international conference on robotics and automation*, pages 1765–1772. IEEE, 2013.

[31] Malcolm Ryan and Mark Reid. Learning to fly: An application of hierarchical reinforcement learning. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer, 2000.

[32] James Steck, Kamran Rokhsaz, James Steck, and Kamran Rokhsaz. Some applications of artificial neural networks in modeling of nonlinear aerodynamics and flight dynamics. In *35th Aerospace Sciences Meeting and Exhibit*, page 338, 1997.

[33] Terry Troudet, Sanjay Garg, and Walter Merrill. Neural network application to aircraft control system design. In *Navigation and Control Conference*, page 2715, 1991.

[34] Baochang Zhang, Zhili Mao, Wanquan Liu, and Jianzhuang Liu. Geometric reinforcement learning for path planning of uavs. *Journal of Intelligent & Robotic Systems*, 77(2):391–409, 2015.