



Formally Verified Switching Logic for Recoverability of Aircraft Controller

Ratan Lal^{1(✉)}, Aaron McKinnis², Dustin Hauptman², Shawn Keshmiri²,
and Pavithra Prabhakar¹

¹ Kansas State University, Manhattan, KS, USA
{ratan,pprabhakar}@ksu.edu

² Department of Aerospace, University of Kansas, Lawrence, KS, USA
{amckinnis,dhauptman,keshmiri}@ku.edu

Abstract. In this paper, we investigate the design of a safe hybrid controller for an aircraft that switches between a classical linear quadratic regulator (LQR) controller and a more intelligent artificial neural network (ANN) controller. Our objective is to switch safely between the controllers, such that the aircraft is always recoverable within a fixed amount of time while allowing the maximum time of operation for the ANN controller. There is a *priori* known safety zone for the LQR controller operation in which the aircraft never stalls, over accelerates, or exceeds maximum structural loading, and hence, by switching to the LQR controller just before exiting this zone, one can guarantee safety. However, this *priori* known safety zone is conservative, and therefore, limits the time of operation for the ANN controller. We apply reachability analysis to expand the known safety zone, such that the LQR controller will always be able to drive the aircraft back to the safe zone from the expanded zone (“recoverable zone”) within a fixed duration. The “recoverable zone” extends the time of operation of the ANN controller. We perform simulations using the hybrid controller corresponding to the recoverable zone and observe that the design is indeed safe.

1 Introduction

Different types of controller designs have been investigated for aircraft control, such as Linear Quadratic Regulators [28], Fuzzy Logic (FL) [8], and Artificial Neural Networks [26]. The LQR controllers provide an optimal controller for linear time invariant (LTI) systems that minimizes a quadratic cost function and guarantees stability and robustness. Though the LQR design is not directly applicable to non-linear systems, often non-linear systems are approximated by linear systems via linearization around the equilibrium point, thus enabling the application of the LQR based design. Although the LQR controller provides good performance for LTI systems [28], studies have shown that the ANN controllers have better performance in the presence of uncertain environments [26]. The ANN controller is especially suitable for adaptive flight control applications,

where system dynamics are dominated by unknown nonlinearities [19]. An aircraft can experience a number of issues that may cause failures in the system. Things like over-acceleration can cause the aircraft to gain too much energy and enter into unstable modes, while rapid de-acceleration and hard maneuvers will cause increased structural loading, leading to broken lifting platforms. Another issue is that of stall, in which the airflow over the lifting section crosses a “critical angle of attack”, compromising the lift generation. All of these problems can occur as a function of the control input or as external disturbances, such as high wind gust, further complicating the problem. Though ANN-based adaptive controllers are capable of handling these situations, guaranteeing safe functionality of these systems remains a challenge due to the complexity of these controllers. So, we have LQR-based controllers on one hand, that are efficient in nominal conditions, and are simple enough to be amenable to analysis, and sophisticated ANN-based controllers on the other hand that can handle difficult environmental conditions, but are, at the same time, too complex to be amenable to analysis. Our solution is a “hybrid controller” consisting of a simplex like architecture [7], wherein, we switch between the ANN and LQR controller in such a way that safety is guaranteed by the switching logic, that is, the aircraft is always recoverable from a stall within a fixed amount of time if it occurs.

Our broad objective is to find an ANN-based controller that can improve performance in uncertain environments. To achieve this goal, we need to train the ANN-controller, however, it is risky to train an ANN controller during a real flight test as it poses a safety risk. Hence, the solution we propose is to switch between a traditional LQR controller and the ANN controller in such a way that safety is guaranteed. More precisely, we allow the ANN controller to operate while the aircraft remains within a “safe zone” from which the LQR controller can guarantee that the aircraft never stalls. When the ANN controller is on the verge of leaving the safe zone, we switch to the LQR controller. However, these expert determined safe zones are often too conservative (small), thereby not providing sufficient time of operation for the ANN controller. A longer duration of operation for the ANN controller is desirable for the learning process, so we provide a method to extend the safe zone to a larger set (“recoverable zone”), which guarantees that the aircraft recovers within a fixed amount of time if a stall occurs. The recoverable zone computation is performed using formal methods based reachable set computation, thereby providing a formally verified switching component decision procedure that guarantees the safe operation of the aircraft.

We consider a dynamic model of a fixed-wing aircraft, with six-degrees-of-freedom (6-DOF), which is used as an experimental platform to employ a hybrid controller that consists of an intelligent and automatic switching between an LQR and an ANN based controller. The aircraft dynamics consists of a decoupled longitudinal and lateral linear time invariant dynamics, with a decoupled state-feedback LQR controller for each component. For our simulations, we consider an ANN controller that combines aircraft guidance and control systems and performs end-to-end mapping from error states to control surface values, in order to fly along a straight line with steady state wings-level and altitude hold.

We have performed Hardware in The Loop (HITL) simulation of the hybrid controller in conjunction with the the 6-DOF differential equations, on the aircraft avionics using the open source software, QGroundControl. Our simulations exhibit that the number of sample iterations for which ANN controller actions are performed while ensuring safe flying, increases as the learning space (recoverable zone) is expanded.

2 Related Work

Artificial Neural Networks have been widely used in many control applications, such as automatic generation control of interconnected power systems [41], irrigation scheduling [37], micro-turbine power plant [36], solar binding [4], robotics [1, 6], and aircraft control [17]. ANN is popularly used in flight control [19], robot control [25] as well as for non-linear systems [42].

Verification has been extensively applied to dynamical systems, and focus on over-approximation based methods including predicate abstraction [3, 22], state-space exploration based fix-point computation [14], Hamilton-Jacobi based methods [2], symbolic state space exploration based methods [16], Satisfiability Modulo Theory (SMT) based methods [20, 21, 23, 38], and counter-example guided abstraction-refinement based methods [24, 31, 32].

Recent studies [40] compare several neural network verification algorithms. Formal verification of feedforward neural networks with different activation functions, such as ReLU [18] and Lipschitz-continuous functions [33], have been studied. Different verification problems have been considered including output range analysis [10], and robustness analysis [15]. Verification methods include those based on reduction to satisfiability solving [18], optimization solving [12], abstract interpretation [35], abstraction-refinement [30], and linearization [13]. Verification of ANN with feedback controllers has been explored [11].

In this paper, one of the problems we study is stall. The stall could occur due to many reasons. Researchers have developed different techniques to avoid or recover from the stall. Deep stall has been studied [27], which is an uncontrollable state at which the angle of attack (AOA) increases automatically and will be locked at a certain AOA which is far beyond the critical angle of attack. A stall due to wing has been studied [39]. The stall avoidance/recovery have been studied [9]. Here, we present a hybrid controller consisting of ANN and LQR controller similar to simplex design [7], which will not only recover, but also provide more learning space for the ANN controller to explore. Our hybrid controller is different from the simplex design [7] in many perspectives. Our hybrid controller makes the decision between ANN and LQR control input via safety checking performed based on an under-approximation reach set, which is computed off-line. However, in the work [7], the analysis is performed based on an over-approximation reach set. Also, in the work [7], an initial set is known; however, in our work, a target set (“safe zone”) is known and the initial set is unknown.

3 Hybrid Controller Architecture

In this section, we provide details of the hybrid controller architecture which is shown in Fig. 1. It has mainly four components: (a) Aircraft dynamics, (b) LQR controller (c) ANN controller, and (d) Switching logic. For the aircraft dynamics, we consider a 6-DOF model of the fixed wing aircraft. The hybrid controller consists of the LQR and the ANN controller, and the switching logic; the LQR and the ANN controller each receive the state of the aircraft periodically (which is obtained from the aircraft dynamics model in the simulations) and compute the inputs to the aircraft. The switching logic decides which input is fed back to the aircraft (dynamics) at each sample time, based on the current state of the system. The state of the system (dynamics) is updated according to the input selected. We note that the details of the ANN controller is not important for the correctness of this work, since the safety is guaranteed even when the ANN control is considered as a black box. However, we adapt the ANN controller from the work [34] for the ANN component of the hybrid controller. We briefly describe the important aspects of the aircraft dynamics, LQR controller and the switching logic.

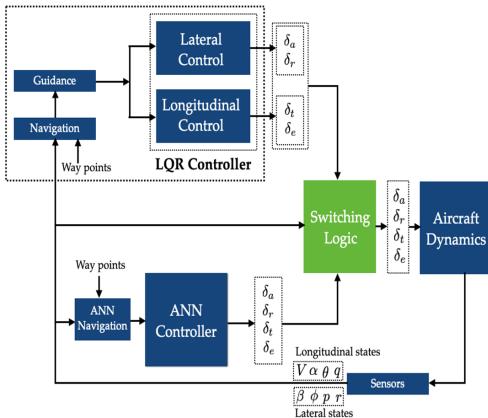


Fig. 1. Hybrid controller architecture

Input: : $\text{Switch}(x, \tau, \mathcal{S})$:
 x - Current state,
 τ - Sample time interval,
 \mathcal{S} - Safe zone

Output: u - Control input

- 1: $u_{ann} = \text{ANN}(x)$
- 2: $x' = \text{nextstate}(x, u_{ann}, \tau)$
- 3: if $x' \in \mathcal{S}$ then
- 4: return u_{ann}
- 5: else
- 6: $u_{lqr} = \text{LQR}(x)$
- 7: return u_{lqr}
- 8: end if

Fig. 2. Switching Logic for LQR and ANN controller

3.1 Aircraft Dynamics

We start with a brief description of the aircraft states and motion. The aircraft has 3 axes, the roll axis (I), pitch axis (J) and yaw axis (K) as shown in Fig. 3. Motion occurs in two planes, the longitudinal, axes (I) and (K), and lateral, axes (I) and (J), which are often considered to be decoupled.

In the longitudinal plane, the states are, velocity (V), angle of attack (α), pitch angle (θ) and pitch rate (q), and control inputs are thrust (δ_t) and elevator

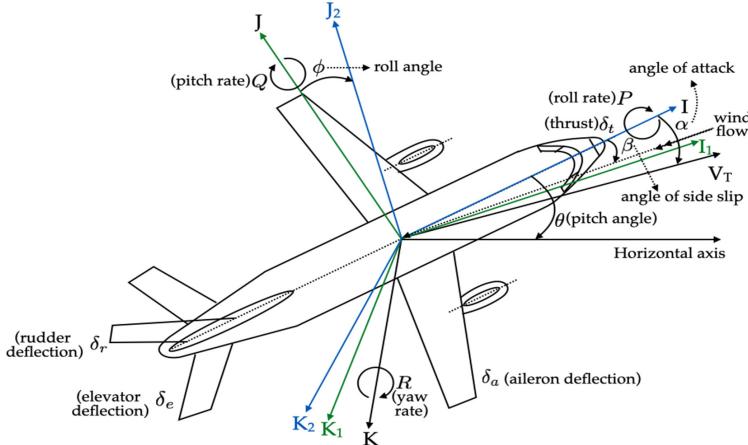


Fig. 3. Overview of aircraft

deflection δ_e . All the states and control inputs are shown in Fig. 3. The angle of attack (α) is the angle between the roll axis (I) and the direction of velocity (V). The pitch angle (θ) is the angle between the roll axis (I) and the horizontal axis. The pitch rate (q) is the rate of change in the pitch angle θ . When the pitch angle (θ) changes, the lateral plane rotates and the roll and yaw axes will change to I_1 and K_1 , respectively. The thrust (δ_t) generates a force that is used to move the aircraft forward along the roll axis, and the elevator deflection (δ_e) is a control surface located at the rear of the aircraft which primarily controls the pitch angle (θ). The longitudinal dynamics is a linear dynamics of the form $\dot{\mathbf{x}}_{lon} = A_{lon}\mathbf{x}_{lon} + B_{lon}\mathbf{u}_{lon}$, where $\mathbf{x}_{lon} = [V, \alpha, \theta, q]'$, $\mathbf{u}_{lon} = [\delta_t, \delta_e]'$, and A_{lon} and B_{lon} are specific matrices.

In the lateral plane, the states are, side-slip angle (β), roll angle (ϕ), roll rate (p) and yaw rate (r), and control inputs are aileron deflection (δ_a) and rudder deflection (δ_r). The states and control inputs are shown in Fig. 3. The angle of side-slip (β) is the angle between the roll axis (I) and the direction of incoming airflow. When the roll axis I rotates, the pitch axis (J) and the yaw axis (K) will change to J_2 and K_2 , respectively. The roll angle (ϕ) is the angle between J and J_2 . The roll rate (p) is the rate of change in the roll angle (ϕ). The yaw rate (r) is the rotational rate of change in the yaw axis (K). The aileron deflection (δ_a) is the control surface which is used to control the rotation of the roll axis (I). The rudder deflection (δ_r) is the control surface which is used to control the rotation of the yaw axis (K). The lateral dynamics is a linear dynamics of the form $\dot{\mathbf{x}}_{lat} = A_{lat}\mathbf{x}_{lat} + B_{lat}\mathbf{u}_{lat}$, where $\mathbf{x}_{lat} = [\beta, \phi, p, r]'$, $\mathbf{u}_{lat} = [\delta_a, \delta_r]'$, and A_{lat} and B_{lat} are specific matrices.

3.2 LQR Controller

Linear Quadratic Regulator (LQR) controller for a linear dynamics $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ is an optimal controller that minimizes a quadratic cost function (J). It is a linear state feedback controller of the form $-K\mathbf{x}$, where K is referred to as the gain matrix. The closed loop dynamics is given by $\dot{\mathbf{x}} = (A - BK)\mathbf{x}$; which is the system behavior when controller by the LQR controller. Since the longitudinal and lateral dynamics of the aircraft are decoupled, we have an LQR controller for each component with gains K_{lon} and K_{lat} , resulting in corresponding closed loop systems, $\dot{\mathbf{x}}_{lon} = (A_{lon} - B_{lon}K_{lon})\mathbf{x}_{lon}$ and $\dot{\mathbf{x}}_{lat} = (A_{lat} - B_{lat}K_{lat})\mathbf{x}_{lat}$.

3.3 Switching Algorithm for the Safety of ANN Controller

Stall is one of the important issues for any aircraft. Stall is a condition in which the angle of attack surpasses a critical bound and greatly decreases lift generation. Consequently, the aircraft will start rapidly descending. Additional problems occur when the aircraft encounters large accelerations, primarily about the roll and yaw axes, which can lead the aircraft into an unstable spiral mode, a dangerous and usually unrecoverable event. Finally, rapid maneuvers can lead to large loads on the aircraft structure, causing permanent deformation or breaking the structure altogether. Generally, exact constraints for these problems cannot be found due to the complexity of aircraft motion. However, a set of safe constraints has been generated for the testbed aircraft by examining previous flight test data in which problems did not occur.

The objective of the switching logic is to arbitrate the switching between the LQR and ANN based controllers, while maintaining safety and at the same time providing ANN controller the maximum opportunity to operate, and thereby learn. Our premise is that we have some known safe zone \mathcal{S} give by an expert in which LQR controller actions are safe, that is, if we apply control input $\mathbf{u} = -K\mathbf{x}$, when $\mathbf{x} \in \mathcal{S}$, to the LTI dynamics of the aircraft, then the aircraft never stalls. However, if we apply control input \mathbf{u}' obtained by the ANN controller at a state $\mathbf{x} \in \mathcal{S}$, we cannot ensure that the system never stalls. Computing such a safe zone for an ANN controller would be computationally hard. Hence, the switching algorithm computes the effect of applying \mathbf{u}' computed by the ANN controller, and decides to pass it on to the system, if it infers that the system will be safe in the next step. Otherwise, it outputs the input suggested by the LQR controller. In either case, it ensures that the system is in the \mathcal{S} region at all times during the operation of the flight. The details of the switching algorithm are provided in Fig. 2.

The performance of the hybrid controller depends on the safe zone. The safe zone obtained by expert advice is often conservative. Hence, we provide a method to extend the safe zone (“recoverable zone”) for which the switching algorithm guarantees that the system is always recoverable within the fixed duration if it occurs. Next, we provide the details of computing the recoverable zone.

4 Computation of Recoverable Zone

In this section, we provide the details of computing a recoverable zone for the fixed time $T > 0$. Our broad goal is to compute all those states from which the given safe zone \mathcal{S} can be reached within the time $T > 0$ for an LTI dynamics of aircraft which is in the form of $\dot{\mathbf{x}} = (A - BK)\mathbf{x}$, where K is an LQR control gain matrix. This is the problem of computing the backward reach set of a linear system

$$\dot{\mathbf{x}} = C\mathbf{x} \quad (1)$$

where $C = A - BK$. The solution of a linear system $\dot{\mathbf{x}} = C\mathbf{x}$ is given by $x(t) = e^{Ct}x(0)$, where $x(t)$ is the state of the system at time t . Hence, we define the backward reach set for a given linear closed loop system as follows:

Definition 1. [Backward Reach Set] *Given a linear closed loop system $\dot{\mathbf{x}} = A\mathbf{x}$, a time horizon $T > 0$, and a final set of states \mathcal{X}_f , the backward reach set $\text{Reach}_B(\mathcal{X}_f, A, [0, T])$ is defined as follows:*

$$\text{Reach}_B(\mathcal{X}_f, A, [0, T]) = \{\mathbf{x} \mid \exists t \in [0, T], e^{At}\mathbf{x} \in \mathcal{X}_f\}.$$

Next, we formally define the recoverable zone in terms of backward reach set.

Definition 2. [Recoverable Zone] *Given system in Eq. (1), a time horizon $T > 0$, and a safe zone \mathcal{S} , a recoverable zone \mathcal{S}' is defined as follows:*

$$\mathcal{S}' = \text{Reach}_B(\mathcal{S}, C, [0, T]).$$

The computation of the recoverable zone \mathcal{S}' can be alternatively tackled using a forward reachability analysis on the following transformed equation.

$$\dot{\mathbf{x}} = -C\mathbf{x} \quad (2)$$

We define forward reach set for a given linear closed loop system as follows:

Definition 3. [Forward Reach Set] *Given a linear closed loop system $\dot{\mathbf{x}} = A\mathbf{x}$, a time horizon $T > 0$, and an initial set of states \mathcal{X}_0 , forward reach set $\text{Reach}_F(\mathcal{X}_0, A, [0, T])$ is defined as follows:*

$$\text{Reach}_F(\mathcal{X}_0, A, [0, T]) = \{e^{At}\mathbf{x}_0 \mid \exists t \in [0, T], \exists \mathbf{x}_0 \in \mathcal{X}_0\}.$$

Equation (2) is obtained from Eq. (1) by negating the right hand side. The effect of the transformation is that the system now evolves backward in time. We notice that the set of states that can reach \mathcal{S} within time T from Equation (1) ($\text{Reach}_B(\mathcal{S}, C, [0, T])$) is equal to the set of states reached using Equation (2) from \mathcal{S} in a given time horizon $T > 0$ ($\text{Reach}_F(\mathcal{S}, C, [0, T])$). Next, we formulate this equivalence of forward and backward reach sets of the two systems, namely Equations (1), (2) in Theorem 1.

Theorem 1. *Given systems in Equation (1) and Equation (2), a time horizon $T > 0$, a safe zone \mathcal{S} , we have $\text{Reach}_F(\mathcal{S}, -C, [0, T]) = \text{Reach}_B(\mathcal{S}, C, [0, T])$.*

The computation of the exact recoverable zone is complex because the solution of Equation (2) consists of exponential function, and there are no known algorithms for solving constraints with exponential functions, unlike solvers for linear and polynomial functions. Hence, several over-approximation methods have been investigated [5, 16, 20, 29, 31, 32]. An over-approximated recoverable zone violates the property of the recoverable zone, that is, it contains point that are not guaranteed to reach the safe zone within the time bound. In this situation, the stall may not be recoverable if it occurs. Therefore, we compute an under-approximation of the exact recoverable zone \mathcal{S}' which is conservative, nevertheless, ensures the safety of the switching algorithm.

4.1 Under-Approximation of Recoverable Zone

In this section, we provide a method to compute an under-approximation of the exact recoverable zone \mathcal{S}' . While computing under-approximations are in general hard, we use a simple idea that provides a practically viable under-approximation for our purposes. Our broad approach is based on sampling, and consists of an under-approximate reach set which is the union of the reach set at certain time points, as opposed to all the points in the given interval. We sample the time interval $[0, T]$ at sample times that are multiples of r . Then, we compute forward reach set from safe zone \mathcal{S} under Equation (2) at sample times $r, 2r, \dots, kr = T$ and take their union, that is, the under-approximation of the recoverable zone denoted $\text{Approx}(\mathcal{S})$ is $\bigcup_{i=0}^k \text{Reach}_F(\mathcal{S}, -C, ir)$, where $\text{Reach}(\mathcal{S}, -C, ir)$ denotes the forward reach set from \mathcal{S} at time ir . Next, we show that $\text{Approx}(\mathcal{S})$ is an under-approximation of the recoverable zone \mathcal{S}' . We formulate this in Theorem 2.

Theorem 2. *Given system in Equation (2), a time horizon $T > 0$, a safe zone \mathcal{S} , we have $\text{Approx}(\mathcal{S}) \subseteq \text{Reach}_F(\mathcal{S}, -C, [0, T])$.*

Note that $\text{Approx}(\mathcal{S})$ converges to the exact recoverable zone \mathcal{S}' as $r \rightarrow 0$.

5 Experimental Analysis

In this section, we provide the details of our implementation of hybrid controller architecture. Then, we present the experimental results.

5.1 Experimental Setup

The experimentation method for preliminary concept testing is a Hardware in The Loop (HITL) simulation. The HITL runs the 6-DOF differential equations, on the aircraft avionics, which are then propagated using a Runge-Kutta fourth order integration method.

**Fig. 4.** AFS 6.0**Fig. 5.** HITL aggressive trajectory

This technique generates all aircraft states and control inputs that are necessary to the operation of the switch. The main advantage of conducting these simulations as an HITL rather than software simulations is that all the codes will be tested on the actual hardware used for flight, showcasing any shortcomings in computation power or integration missteps, which may impact flight test success.

The current avionics, Autopilot Flight System (AFS) 6.0, consists of three main components. Sensor data and outputs are handled by the Pixhawk 2.1 cube. The onboard computer which runs the in-house designed guidance, navigation and control (GNC) algorithms, as well as handles the state emulation is the Nvidia Tegra Nano. The Tegra Nano is a low cost system, with a quad-core CPU and a 128 core GPU. The final component is a 900 MHz telemetry unit which serves as the communication between the aircraft and the ground station, where the ground station provides a visual representation of the current aircraft state as well as relevant GNC information. The ground station used for these simulations is a modified version of the open source software, QGroundControl, which is also used to generate way-points for the given area of operation. Figure 4 shows both the front and back sides of the custom avionics boards.

While in HITL, the ANN controllers are very stable due to being trained with similar dynamic models to those that are used to propagate the simulation. This makes it unlikely to see the switching logic in action as no control inputs would be deemed unsafe, especially in grid or racetrack patterns that make up the majority of flight test operations. To circumvent this, an oddly shaped trajectory, shown in Fig. 5, with multiple sharp turns is used to ensure previously un-visited states are achieved. The simulation is run for approximately one lap of the given trajectory for each value of the time horizon shown in the following section.

5.2 Experimental Results

In this section, we present the simulation results for the performance of hybrid controller. For the simulation, we consider the safe zone provided by experts, which are given in Table 1. We run the simulation for different recoverable zones, which are computed for different values of time horizon T , namely, $T = 0.05$,

$T = 0.15$, $T = 0.25$, and $T = 0.35$ with time step $\tau = 0.05$ unit. The simulation results are shown in Figs. 6 and 7. The simulation results are plotted in Fig. 6 and Fig. 7 for longitudinal velocity and lateral angle of side-slip, respectively.

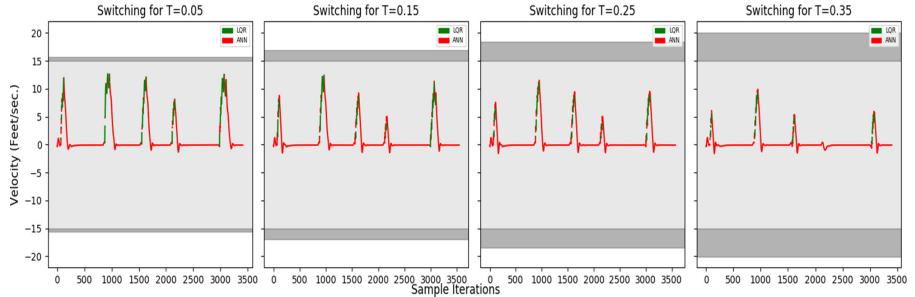


Fig. 6. Switching between ANN and LQR controller for the longitudinal velocity

Table 1. Safe zone for longitudinal and lateral state variables

Safe Zone	V (Feet/sec.)	α (Radians)	θ (Radians)	Q (Radians/sec.)	β (Radians)	ϕ (Radians)	P (Radians/sec.)	R (Radians/sec.)
Min	-15	-0.087	-0.262	-0.262	-0.122	-0.785	-0.873	-0.349
Max	15	0.087	0.262	0.262	0.122	0.785	0.873	0.349

In both Figs. 6 and 7, we observe that the recoverable zone expands when the time horizon T increases.

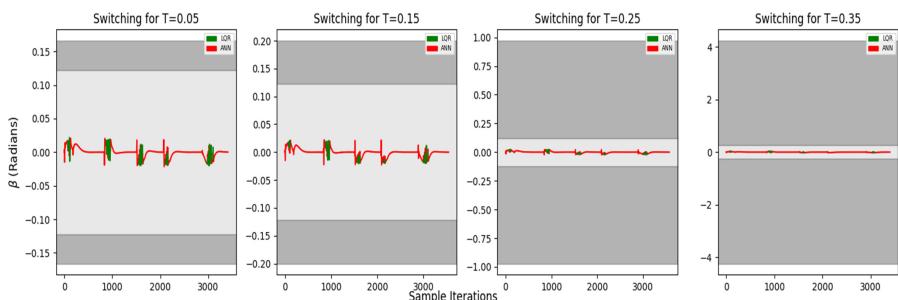


Fig. 7. Switching between ANN and LQR controller for the lateral angle of side-slip

Also, we observe that the number of sample iterations in which ANN controller actions are performed, increases when the recoverable zone is expanded. For instance, in Figs. 6 and 7, for $T = 0.35$, ANN controller actions have been

performed from the sample iteration 1500 to 3000, which was not the case for $T = 0.25$. For clarity, in Table 2, we present the number of sample iterations N for both ANN and LQR controller in which their actions have been performed, for different values of time horizon T .

Table 2. Number of sample iterations for ANN and LQR controller

	N for $T = 0.05$	N for $T = 0.15$	N for $T = 0.25$	N for $T = 0.35$
<i>ANN</i>	3181	3269	3295	3345
<i>LQR</i>	220	132	106	56
<i>Total</i>	3401	3401	3401	3401

In Table 2, we observe that N grows for ANN controller when the time horizon T increases, that is, the recoverable zone is expanded. However, N decreases for LQR controller when T increases. This validate the fact that hybrid controller framework provides ample time for the ANN controller to learn while ensuring a safe flight.

5.3 Practical Challenges

The implementation of the hybrid controller proved to be complex in two ways. First, the timing of the switching logic was important to the overall safety of the project. When delays are introduced into the system, the current state of the aircraft and the information the switch is making the decision on can become out of sync. If the switching logic is behind the aircraft states it can make incorrect calls on whether or not the aircraft is still safe, and cause the ANN to overextend its operation, leading to a loss of control. This is made worse as aircraft have large inertias and relatively slow time constants on control inputs meaning they can become uncontrollable much quicker than most dynamic systems. This need for extreme low latency operation caused many changes in the code structure including a rewrite from Python to C++ and parallelization of applicable code. The second practical problem is that the lack of full state feedback and low-quality sensor data. Two of the aircraft states, angle of attack and sideslip angle, cannot be directly measured by low cost systems. The easiest solution is to employ a Kalman filtering technique to estimate these two states. However, if the aircraft is experiencing a large perturbation away from the trim point, the Kalman Filter can diverge very rapidly and feed incorrect information to the switch about the relevant states. On top of this, many of the measured states are taken using low-cost, off the shelf components. In a similar way, the use of these components may introduce noise or a bias which could allow the aircraft to go into the uncontrollable region without alerting the switch or the aircraft operator. Low pass filtering is applied to attempt to deal with the noise, but the imparted delay to the sensor data must also be taken into consideration.

6 Conclusions

We have developed a hybrid controller for an aircraft dynamics which provides considerable amount of time to the ANN controller to operate and learn, while at the same time guarantees the safe operation of the flight at all times. In future, we will consider more sophisticated ANN controllers and investigate methods for computing larger recoverable zones that allow for further increase of the ANN operation time. Additionally, experimentation will be done with real flight tests, moving past HITL simulations.

Acknowledgements. Pavithra Prabhakar was partially supported by NSF CAREER Grant No. 1552668, NSF Grant No. 2008957, ONR YIP Grant No. N000141712577 and USDA Grant No. 2017-67007-26153. Also, this work was partially supported with funding from National Aeronautics and Space Administration (NASA) Grant No. NNX15AJ97H and Grant No. 80NSSC19C0102.

References

1. Abdelhameed, M.M.: Adaptive neural network based controller for robots. *Mechatronics* **9**, 147–162 (1999)
2. Akametalu, A.K., Fisac, J.F., Gillula, J.H., Kaynama, S., Zeilinger, M.N., Tomlin, C.J.: Reachability-based safe learning with gaussian processes. In: Conference on Decision and Control (2014)
3. Alur, R., Dang, T., Ivancic, F.: Predicate abstraction for reachability analysis of hybrid systems. *ACM Trans. Embedded Comput. Syst. (TECS)* **5**, 152–199 (2006)
4. Argiriou, A.A., Bellas-Velidis, I., Kummert, M., André, P.: A neural network controller for hydronic heating systems of solar buildings. *Neural Netw.* **17**, 472–440 (2004)
5. Asarin, E., Bournez, O., Dang, T., Maler, O.: Approximate reachability analysis of piecewise-linear dynamical systems. *Computation and Control*. In: International Workshop on Hybrid Systems (2000)
6. Azhar, M.A.H.B., Dimond, K.R.: Design of an FPGA based adaptive neural controller for intelligent robot navigation. In: Proceedings Euromicro Symposium on Digital System Design. Architectures, Methods and Tools (2002)
7. Bak, S., Johnson, T.T., Caccamo, M., Sha, L.: Real-time reachability for verified simplex design. In: IEEE Real-Time Systems Symposium (2014)
8. Bhangal, N.S.: Design and performance of LQR and LQR based fuzzy controller for double inverted pendulum system. *J. Image Graph.* **1**, 143–146 (2013)
9. Brown, J.A.: Stall avoidance system for aircraft, 1986. US Patent 4,590,475 (1986)
10. Dutta, S., Chen, X., Sankaranarayanan, S.: Reachability analysis for neural feedback systems using regressive polynomial rule inference. In: International Conference on Hybrid Systems: Computation and Control, (2019)
11. Dutta, S., Jha, S., Sankaranarayanan, S., Tiwari, A.: Output range analysis for deep feedforward neural networks. In: NASA Formal Methods Symposium (2018)
12. Dvijotham, K., Stanforth, R., Gowal, S., Mann, T.A., Kohli, P.: A dual approach to scalable verification of deep networks. *CoRR*, abs/1803.06567 (2018)
13. Ehlers, R.: Formal verification of piece-wise linear feed-forward neural networks. In: D’Souza, D., Kumar, K.N. (eds.) *Automated Technology for Verification and Analysis* (2017)

14. Frehse, G., et al.: SpaceEX: scalable verification of hybrid systems. In: International Conference on Computer Aided Verification (2011)
15. Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.T.: AI2: safety and robustness certification of neural networks with abstract interpretation. In: IEEE Symposium on Security and Privacy (2018)
16. Girard, A.: Reachability of uncertain linear systems using zonotopes. Computation and Control. In: International Workshop on Hybrid Systems (2005)
17. Julian, K.D., Kochenderfer, M.J., Owen, M.P.: Deep Neural Network Compression for Aircraft Collision Avoidance Systems. arXiv e-prints (2018)
18. Katz, G., Barrett, C.W., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: an efficient SMT solver for verifying deep neural networks. CoRR (2017)
19. Khosravani, M.R.: Application of neural network on flight control. Int. J. Mach. Learn. Comput. **2**, 882–885 (2012)
20. Lal, R., Prabhakar, P.: Bounded error flowpipe computation of parameterized linear systems. In: International Conference on Embedded Software (2015)
21. Lal, R., Prabhakar, P.: Safety analysis using compositional bounded error approximations of communicating hybrid systems. In: IEEE 56th Annual Conference on Decision and Control (CDC) (2017)
22. Lal, R., Prabhakar, P.: Hierarchical abstractions for reachability analysis of probabilistic hybrid systems. In: 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton) (2018)
23. Lal, R., Prabhakar, P.: Compositional construction of bounded error over-approximations of acyclic interconnected continuous dynamical systems. In: ACM-IEEE International Conference on Formal Methods and Models for System Design (2019)
24. Lal, R., Prabhakar, P.: Counterexample guided abstraction refinement for polyhedral probabilistic hybrid systems. ACM Trans. Embedded Comput. Syst. (TECS) **18**, 1–23 (2019)
25. Lewis, F.W., Jagannathan, S., Yesildirak, A.: Neural Network Control of Robot Manipulators and Non-linear Systems. CRC Press, Boca Raton (1998)
26. Mehri, M.: A comparison of neural network models, fuzzy logic, and multiple linear regression for prediction of hatchability. Poult. Sci. **92**, 1138–1142 (2013)
27. Montgomery, R.C., Moul, M.T.: Analysis of deep-stall characteristics of t-tailed aircraft configurations and some recovery procedures. J. Airc. **3**, 562–566 (1966)
28. Nair, V.G., Dileep, M.V., George, V.: Aircraft yaw control system using LQR and fuzzy logic controller. Int. J. Comput. Appl. **45**, 25–30 (2012)
29. Prabhakar, P., Viswanathan, M.: A dynamic algorithm for approximate flow computations. In: International Conference on Hybrid Systems: Computation and Control (2011)
30. Pulina, L., Tacchella, A.: An abstraction-refinement approach to verification of artificial neural networks. In: International Conference on Computer Aided Verification (2010)
31. Roohi, N., Prabhakar, P., Viswanathan, M.: HARE: A Hybrid Abstraction Refinement Engine for Verifying Non-linear Hybrid Automata. In: Legay, A., Margaria, T. (eds.) TACAS 2017. LNCS, vol. 10205, pp. 573–588. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54577-5_33
32. Roohi, N., Prabhakar, P., Viswanathan, M.: Hybridization based CEGAR for hybrid automata with affine dynamics. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems (2016)
33. Ruan, W., Huang, X., Kwiatkowska, M.: Reachability analysis of deep neural networks with provable guarantees. CoRR, abs/1805.02242 (2018)

34. Shukla, D., Lal, R., Hauptman, D., Keshmiri, S.S., Prabhakar, P., Beckage, N.: Flight test validation of a safety-critical neural network based longitudinal controller for a fixed-wing UAS. In: AIAA AVIATION (2020)
35. Singh, G., Gehr, T., Püschel, M., Vechev, M.T.: An abstract domain for certifying neural networks. PACMPL **3**(POPL), 41:1–41:30 (2019)
36. Sisworahardjo, N.S., El-Sharkh, M.Y., Alam, M.S.: Neural network controller for microturbine power plants. Electr. Power Syst. Res. **78**, 1378–1384 (2008)
37. Umair, S.M., Usman, R.: Automation of irrigation system using ANN based controller. Int. J. Electr. Comput. Sci. (2010)
38. Veanes, M., Bjørner, N., Raschke, A.: An SMT approach to bounded reachability analysis of model programs. In: International Conference on Formal Techniques for Networked and Distributed Systems (2008)
39. Wang, T.: Aircraft wing stall control device and method, 1987. US Patent 4,702,441 (1987)
40. Xiang, W., et al.: Verification for machine learning, autonomy, and neural networks survey. CoRR, abs/1810.01989 (2018)
41. Zeynelgil, H.L., Demiroren, A., Sengor, N.S.: The application of ANN technique to automatic generation control for multi-area power system. Int. J. Electr. Power Energy Systems (2002)
42. Zhang, T., Ge, S.S., Hang, C.C.: Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. Automatica **36**, 1835–1846 (2000)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

