Adaptive Critic Control For Aircraft Lateral-Directional Dynamics

Ashraf, Imrul; van Kampen, Erik-jan

DOI

[10.2514/6.2020-1845](https://doi.org/10.2514/6.2020-1845)

Publication date

2020

Document Version

Final published version

Published in

AIAA Scitech 2020 Forum

Citation (APA)

Ashraf, I., & van Kampen, E. (2020). Adaptive Critic Control For Aircraft Lateral-Directional Dynamics. In *AIAA Scitech 2020 Forum: 6-10 January 2020, Orlando, FL* (pp. 1-23). [AIAA 2020-1845] (AIAA Scitech 2020 Forum; Vol. 1 PartF). American Institute of Aeronautics and Astronautics Inc. (AIAA).
<https://doi.org/10.2514/6.2020-1845>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Adaptive Critic Control For Aircraft Lateral-Directional Dynamics

Imrul K. Ashraf*, E. van Kampen†

Delft University of Technology, PO Box 5058, 2600 GB Delft, The Netherlands

Loss of control-in flight (LOC-I) is one of the causes of catastrophic aircraft accidents. Fault-tolerant flight control (FTFC) systems can prevent LOC-I and recover aircraft from LOC-I precursors. One group of promising methods for developing Fault-Tolerant Control (FTC) system is the Adaptive Critic Designs (ACD). Recently one ACD algorithm, called value function based single network adaptive critic (J-SNAC), has emerged and it promises to make applications of ACD more practical by reducing the required amount of computations. This paper discusses the implementation of this framework for the design of a lateral-directional flight controller. The proposed flight controller is trained to perform coordinated-turns with an F16 simulation model. The trained controller was evaluated for tracking two different heading command signals, robustness against sensor noises and partial failure of the ailerons. The controller is found to be effective for the considered assessments.

Nomenclature

ACD	Adaptive Critic Designs
ADP	Approximate Dynamic Programming
CE	Control Effectiveness
FA	Function Approximator
FCS	Flight Control System
H.O.T	Higher Order Terms
J-SNAC	Value Function Based Single Network Adaptive Critic
LOC-I	Loss Of Control-In Flight
PI	Performance Index
PID	ProportionalIntegralDerivative
RL	Reinforcement Learning
RLS	Recurssive Least Square
RMS	Root-Mean-Square
TD	Temporal Difference

I. Introduction

Loss of control is one of the causes of catastrophic aircraft accidents.¹⁻⁴ Enhanced dynamics control strategies, that can accommodate onboard system failures and persist in adverse operational environment,² can be employed to diminish this cause. “*Adaptive Critic Design*” (ACD) algorithms are a group of such strategies.⁵⁻¹¹ These are a class of *Reinforcement Learning* (RL) algorithms, that uses *function approximators* (FA) and *Approximate Dynamic Programming* (ADP) technique to learn solutions to complex control problems autonomously. Their learning capability may enable Flight Control Systems (FCS) to adapt in response to unanticipated changes in the aircraft sub-systems or operating conditions. However, due to lack of maturity, these algorithms are yet to be implemented in FCS.

*MSc. Student, Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology

†Assitant Professor, Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology

Until now several RL Flight Controllers have been proposed with different ACD frameworks.^{12–21} One limitation of these controllers is that they have an exorbitant computational requirement. This requirement comes from learning two different functions with separate function approximation structures. Utilizing one of the modern ACD architectures,^{22–25} can circumvent the computational burden. These modern frameworks make half of the required computation in ACD algorithms superfluous by eliminating one of the function to be learned. Furthermore, most of the research mentioned above have focused their work on the control of aircraft longitudinal dynamics. Control of the lateral-directional flight dynamics with ACD would reveal the efficacy of these algorithms to control the coupled roll and yaw motions and thus facilitating their implementation in future FCS.

This article contributes by addressing the mentioned limitations of the previous studies. It focuses on the theoretical development and performance analyses of a lateral-directional flight controller designed with Value Function based Single Network Adaptive Critic (J-SNAC) algorithm. The organization of this article is as follows. Section II introduces the preliminaries to rest of the article. Next, Section III presents the objective of the proposed controller and its design. Then Section IV gives the controller training schedule and performance evaluation strategies. Subsequently, Section V presents the results and discussions from the training and evaluation processes. Finally, Section VI concludes the article with the implications of this paper and future research directions.

II. Preliminaries

This section presents the preliminaries to the development of lateral-directional flight control law with the J-SNAC algorithm. Firstly, it describes the lateral-directional flight dynamics. Next, it presents the *Infinite Horizon Discounted Return Problem* and few essential concepts required to solve this problem. Finally, it provides an overview of the J-SNAC algorithm.

A. Lateral-Directional Flight Dynamics

The objective of this work is to synthesize a reinforcement learning controller to drive aircraft heading angle ψ , roll angle ϕ , side slip angle β , roll rate p and yaw rate r (see Figure 1 for the definitions) by manipulating of the aileron δ_a and rudder δ_r deflections. The system of equations that governs these dynamic states is as follows,

$$\begin{aligned}\dot{\psi} &= \frac{1}{\cos \theta}(q \sin \phi + r \cos \phi) \\ \dot{\phi} &= p + \tan \theta(q \sin \phi + r \cos \phi) \\ \dot{\beta} &= \frac{Y}{m} + p \sin \alpha - r \cos \alpha + \frac{g}{V} \cos \beta \sin \phi \cos \theta + \frac{\sin \beta}{V} (g \cos \alpha \sin \theta - g \sin \alpha \cos \phi \cos \theta + \frac{T \cos \alpha}{m}) \\ \dot{p} &= \frac{1}{I_{xx}I_{zz} - I_{xz}^2} (I_{zz}L + I_{xz}N + (I_{xz}(I_{xx} - I_{yy} + I_{zz}))pq + (I_{zz}(I_{yy} - I_{zz}) - I_{xz}^2)qr) \\ \dot{r} &= \frac{1}{I_{xx}I_{zz} - I_{xz}^2} (I_{xz}L + I_{xx}N - (I_{xz}(I_{xx} - I_{yy} + I_{zz}))qr + (I_{xx}(I_{xx} - I_{yy}) - I_{xz}^2)pq)\end{aligned}\quad (1)$$

The dynamics of the lateral-directional state variables are coupled with longitudinal state variables (e.g., the involvement of airspeed V , body pitch rate q , pitch angle θ and angle of attack α in Eq. 1). The aerodynamic forces and moments that influences the lateral-directional dynamics most are side-force Y , rolling moment L and yaw moment N . These forces and moments depend on the Mach number, aerodynamic angles (α and β) and deflections of the aerodynamic surfaces (δ_a , δ_e and δ_r). Next to these force and moments, gravitational attraction g influences the lateral-directional dynamics. Last but not least, the state variables are dependent on aircraft inertial properties, i.e., mass m , and mass moment of inertia I_{xx} , I_{yy} , I_{zz} and I_{xz} .

This work assumes that airspeed and altitude controller are in-place so that the cross-coupling between longitudinal and lateral-directional state variables are negligible. Additionally, the effects of thrust T on side-slip dynamics $\dot{\beta}$ is considered to be weak.

B. Infinite Horizon Discounted Return Problem

Reinforcement Learning (RL) algorithms are a group of data-driven approaches to solving optimal control problems.^{10,26} The type of optimal control problem considered for the development of flight controller is called “Infinite Horizon Discounted Return Problem”.²² This problem is defined as follows.

Figure 1. Definition of aircraft state variables with body reference frame F^b and geodetical reference frame F^g .

Given a continuous-time-nonlinear system,

$$\dot{\mathbf{x}}(t) = f[\mathbf{x}(t), \mathbf{u}(t)] \quad (2)$$

with $\mathbf{x} \in X \subset \mathbb{R}^n$ being the states, $\mathbf{u} \in U \subset \mathbb{R}^m$ being the control inputs. An associated one-step-control performance for this system is given by the *reward* function $r(t)$,

$$r(t) = \rho[\mathbf{x}(t), \mathbf{u}(t)] \quad (3)$$

The objective is to find a state feedback control law,

$$\mathbf{u}(t) = h[\mathbf{x}(t)] \quad (4)$$

such that the following performance measure is maximized for any initial state $\mathbf{x}(t_0) \in X$.

$$R[\mathbf{x}(t)] = \int_t^\infty e^{-\frac{s-t}{\tau}} \rho[\mathbf{x}(s), \mathbf{u}(s)] ds \quad (5)$$

In Eq. (5), $R[\mathbf{x}(t)]$ is the *return* of the state $\mathbf{x}$ and τ is the time constant to discount future rewards.

C. Value and Policy Functions

ACD compute solutions to control problems (i.e. *optimal control policy*) through the *optimal value* function. Below are definitions of policy function, value function and their optimal forms.

A policy $h(\mathbf{x})$ is defined as the stationary mapping of states to control actions,

$$h(\mathbf{x}) : \mathbf{x} \rightarrow \mathbf{u}, \quad \forall \mathbf{x} \in X \quad (6)$$

The stationary mapping of return $R(\mathbf{x})$ from each state $\mathbf{x} \in X$ for a given control policy $h(\mathbf{x})$ is defined as the value function $V^h(\mathbf{x})$,

$$V^h(\mathbf{x}) : \mathbf{x} \rightarrow R(\mathbf{x}), \quad \forall \mathbf{x} \in X, \quad u(t) = h[x(t)] \quad (7)$$

The optimal value function $V^*(\mathbf{x})$ is that corresponds to the optimal control policy $h^*(x)$. It is defined as following,

$$\begin{aligned} V^*(\mathbf{x}) &= \int_t^\infty e^{-\frac{s-t}{\tau}} \rho[\mathbf{x}(s), h^*[\mathbf{x}(s)]] ds \\ &= \max_{\mathbf{u}_{[t, \infty)}} \left[\int_t^\infty e^{-\frac{s-t}{\tau}} \rho[\mathbf{x}(s), \mathbf{u}(s)] ds \right] \end{aligned} \quad (8)$$

Where $\mathbf{u}_{[t,\infty)}$ is the time course of $\mathbf{u}(s) \in U$ for $t \leq s < \infty$. According to the principle of optimality,²⁷ at time t , the optimal value function satisfies following self-consistency property.²²

$$\frac{1}{\tau} V^*(\mathbf{x}) = \max_{\mathbf{u}(t) \in U} \left[\rho[\mathbf{x}(t), \mathbf{u}(t)] + \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} f[\mathbf{x}(t), \mathbf{u}(t)] \right] \quad (9)$$

Eq. (9) presents the Hamilton-Jacobi-Bellman (HJB) equation for the Infinite Horizon Discounted Return problem. The optimal policy consists of actions that maximize the right-hand side of the HJB equation, i.e.,

$$\mathbf{u}^*(t) = h^*[\mathbf{x}(t)] = \arg \max_{\mathbf{u} \in U} \left[\rho[\mathbf{x}(t), \mathbf{u}] + \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} f[\mathbf{x}(t), \mathbf{u}] \right] \quad (10)$$

D. Policy Evaluation and Improvement

Policy Evaluation and *Policy Improvement* are two interactive processes through which ACD algorithms learn the optimal value and policy function. Below are descriptions of policy evaluation and policy improvement processes in J-SNAC algorithms. Detailed descriptions of these processes can be found in²² for complete and their derivation.

1. Policy Evaluation

Policy Evaluation is the process of estimating the value function $V^h(\mathbf{x})$ corresponding to the policy $h(\mathbf{x})$. Given, a parametric function $\hat{V}(\mathbf{x}(t); \mathbf{w})$ that approximates the $V^h(\mathbf{x})$, with $\mathbf{w}$ being a set of function approximator parameters. When the estimated value function $\hat{V}(\mathbf{x}(t))$ is a equivalent to $V^h(\mathbf{x})$, it satisfies following consistency condition.

$$\dot{V}^h(\mathbf{x}(t)) = \frac{1}{\tau} V^h(\mathbf{x}(t)) - r(t) \quad (11)$$

When the consistency condition is not satisfied, the disparity between the predicted and the real function can be reduced by minimizing the *Temporal Difference* (TD) error $\delta(t)$.

$$\delta(t) \equiv r(t) - \frac{1}{\tau} \hat{V}(t) + \dot{\hat{V}}(t) \quad (12)$$

TD error diminishes when the loss function $E_c(t)$ is minimized by adjusting the parameters of the value function approximator.

$$E_c(t) = \frac{1}{2} \delta^2(t) \quad (13)$$

One approach to adapting the function approximator is to utilize the TD(0) algorithm, where parameters are adjusted with the following gradient estimate.

$$\frac{\partial E_c(t)}{\partial w_i} = -\delta(t) \frac{1}{\tau} \frac{\partial \hat{V}(t)}{\partial w_i} \quad (14)$$

However, further improvement in the learning performance can be made by adding *eligibility traces* in the parameter update law (TD(λ) algorithm). Eligibility traces *smoothen the descending gradient* and distributes the credits of receiving rewards to the visited states according to their the recency of visits. The weight update law with eligibility trace is given by,

$$\begin{aligned} w_i &= w_i - \alpha(t) \delta(t) e_i \\ \dot{e}_i(t) &= -\frac{1}{\kappa} e_i(t) + \frac{\partial \hat{V}(\mathbf{x}(t); \mathbf{w})}{\partial w_i} \end{aligned} \quad (15)$$

Where $\alpha(t)$ is a variable learning rate, and $0 < \kappa \leq \tau$ is the time constant of the eligibility trace.

2. Policy Improvement

Policy improvement is the process of improving the policy $h(\mathbf{x})$ by making the policy *greedy* with respect to the current estimate of the value function $V^h(\mathbf{x})$. This process entails searching for value function optimizing actions (greedy actions). When the system dynamics $\dot{x}$ is affine-in-input (see Eq. (16)) and the reward function $\rho(\mathbf{x}, \mathbf{u})$ is convex with respect to the action $\mathbf{u}$, the searching operation has a unique solution and it can be expressed in a closed form function.^{22,23,25,28}

$$\dot{\mathbf{x}}(t) = f[\mathbf{x}(t)] + g[\mathbf{x}(t)]\mathbf{u}(t) \quad (16)$$

Assuming that reward function can be separated into state dependent $\rho_{\mathbf{x}}(\mathbf{x})$ (defined to encompass the control objective) and action dependent $\rho_{\mathbf{u}}(\mathbf{u})$ parts (defined to engrave physical limits and/or learning strategy). The reward function can be expressed as,

$$\rho(\mathbf{x}, \mathbf{u}) = \rho_{\mathbf{x}}(\mathbf{x}) - \sum_{i=1}^m \rho_{u_i}(u_i) \quad (17)$$

From the definition of optimal policy in Eq. (10), an action is said to be greedy if it satisfies,

$$\begin{aligned} 0 &= \frac{\partial}{\partial \mathbf{u}} \left[\rho[\mathbf{x}(t), \mathbf{u}] + \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} f[\mathbf{x}(t), \mathbf{u}(t)] \right] \\ &= \frac{\partial}{\partial \mathbf{u}} \left[\rho(\mathbf{x}(t), \mathbf{u}) + \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} (f[\mathbf{x}(t)] + g[\mathbf{x}(t)]\mathbf{u}) \right] \\ &= -\rho'_{u_i}(u_i) + \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}(t)) \quad (i = 1, \dots, m) \end{aligned} \quad (18)$$

From this derivation, the closed form function for greedy policy (named as *the actor*) is given as,

$$\mathbf{u}(t) = \rho'_{\mathbf{u}}^{-1} \left(\frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} g[\mathbf{x}(t)] \right) \quad (19)$$

As per Eq. (19), the computation of greedy actions requires an estimate of *Control Effectiveness* (CE) parameters and the co-states.

E. Value Function Based Single Network Adaptive Critic

Figure 2 presents a pictorial depiction of the Value Function Based Single Network Adaptive Critic (J-SNAC) algorithm. It solves infinite horizon discounted return problem defined for an input-affine system, forward in time. It consists of five subsystems, namely the critic, the plant model, the reward function, the action modifier, and the actor. The derivation of this algorithm can be found in.²²

1. The Critic

The critic learns the optimal value function $V^*(\mathbf{x})$ and reads out the state values $V(\mathbf{x})$ and the co-state-values $\partial V(\mathbf{x})/\partial \mathbf{x}$ to other subsystems of the controller. The critic system uses a TD(λ) algorithm to learn the optimal value function. It reads out the state value from the learned function and calculates co-states by performing backpropagation on the approximated function.

In this work Normalized Radial Basis Function (NRBF) network^{22,29,30} is used for the critic. The choice of this parametric structure is motivated by its ability to alter the estimated function in a local region of the state-space without altering the global shape. Assuming K basis functions in the network, output V from the NRBF structure for a given input $\mathbf{x}$ is given by

$$\begin{aligned} V(\mathbf{x}; \mathbf{a}) &= \sum_{k=1}^K a_k v_k(\mathbf{x}) \\ v_k(\mathbf{x}) &= \frac{u_k}{\sum_{l=0}^K u_l(\mathbf{x})} \\ u_k(\mathbf{x}) &= e^{-\|r_k^T(\mathbf{x} - \mathbf{c}_k)\|} \end{aligned} \quad (20)$$

Where a_k , c_k and r_k are the amplitude, location and spread of the k^{th} basis function.

2. The Reward Function

The reward function computes the one step performance of the controller. It is a user defined function to encapsulate the control objective and physical constraints. J-SNAC algorithm assumes that the reward function is action-dependent, i.e., $r(\mathbf{x}, \mathbf{u})$ and convex with respect to the action $\mathbf{u}$.

3. Action modifier

To learn a stationary, near-optimal value function and to estimate the control effectiveness parameters, the action applied by the actor needs to excite the system-to-be controlled persistently. This excitation signal is called exploration action signal. J-SNAC uses a filtered and modulated noise signal as its excitation signal²² and it is generated with the following system of equations.

$$\begin{aligned} \mathbf{u}_n(t) &= \sigma(t)\mathbf{n}(t) \\ \tau_n \dot{\mathbf{n}}(t) &= -\mathbf{n}(t) + \mathbf{N}(t) \\ \sigma(t) &= \sigma_0 \min \left[1, \max \left[0, \frac{r_{\max} - V(t)}{r_{\max} - r_{\min}} \right] \right] \end{aligned} \quad (21)$$

Where, σ_0 is the maximum perturbing action, $\mathbf{N}(t)$ is a zero-mean Gaussian noise signal, $V(t)$ is the estimated value of the state at time t , $r_{\max}$ and $r_{\min}$ are the maximum and minimum value of expected rewards $r(t)$.

4. The Plant Model

The plant model estimates of the *Control Effectiveness* (CE). In this work, CE is approximated incrementally with Recursive Least Square (RLS) estimator.^{31,32} The central idea in this estimation process is to linearize the plant locally in time and space and use sampled input-output data to estimate the parameters of the linearized plant.

Given a continuous-time nonlinear system (e.g., Eq. (2)), it can be linearized around a time t_0 using Taylor series expansion,

$$\dot{x}(t) = \dot{x}(t_0) + \left. \frac{\partial f(x(t), u(t))}{\partial x(t)} \right|_{x(t_0), u(t_0)} (x(t) - x(t_0)) + \left. \frac{\partial f(x(t), u(t))}{\partial u(t)} \right|_{x(t_0), u(t_0)} (u(t) - u(t_0)) + \text{H.O.T} \quad (22)$$

Truncating the expansion up-to linear terms and rewriting the terms $(\dot{x}(t) - \dot{x}(t_0))$, $(x(t) - x(t_0))$, $(u(t) - u(t_0))$, $\left. \frac{\partial f(x(t), u(t))}{\partial x(t)} \right|_{x(t_0), u(t_0)}$, $\left. \frac{\partial f(x(t), u(t))}{\partial u(t)} \right|_{x(t_0), u(t_0)}$ as $\Delta \dot{x}(t)$, $\Delta x(t)$, $\Delta u(t)$, $F[x(t_0), u(t_0)]$, $G[x(t_0), u(t_0)]$ respectively, following linear system can be approximated,

$$\Delta \dot{x}(t) \approx F[x(t_0), u(t_0)] \Delta x + G[x(t_0), u(t_0)] \Delta u \quad (23)$$

Assuming that states and actions are sampled at a fast rate, the linearized drift dynamics $F[x(t_0), u(t_0)]$ and control effectiveness $G[x(t_0), u(t_0)]$ can be estimated with an RLS estimator.³³ The system of equations

Figure 2. J-SNAC control algorithm. At time t , $x(t)$ is the state measurements, $u(t)$ is the action to be applied, $V(t)$ is value of the state x , $r(t)$ is the reward for being in state x and applying action u , $u_n(t)$ is an additive noise signal, $\delta(t)$ is the temporal difference, $f(x, u)$ is the system dynamics, and ∇ is system/operator to calculate partial derivatives (e.g. ∇_{tx} is the partial derivatives of x with respect to t).

for the RLS estimator is as follows,

$$\begin{aligned}
\Delta \hat{x}(t) &= X(t)^T \hat{\Theta}(t-1) \\
e(t) &= \Delta \dot{x}(t) - \Delta \hat{x}(t) \\
\hat{\Theta}(t) &= \hat{\Theta}(t-1) + K(t)e(t) \\
K(t) &= Q(t)X(t) \\
Q(t) &= \frac{P(t-1)}{\Lambda + X(t)^T P(t-1) X(t)} \\
P(t) &= \frac{1}{\Lambda} \left[P(t-1) - \frac{P(t-1)X(t)X(t)^T P(t-1)}{\Lambda + X(t)^T P(t-1) X(t)} \right]
\end{aligned} \tag{24}$$

Where $\Delta \hat{x}(t)$ is the estimation of the incremental change in state rate $\Delta \dot{x}(t)$, X is the regression vector $[\Delta x \ \Delta u]^T$, $\hat{\Theta}(t)$ is the concatenated matrix of estimated drift dynamics and control effectiveness $[\hat{F}^T \ \hat{G}^T]^T$ at time t , K is the estimator gain, Q is the innovation matrix, P is the estimator covariance matrix and finally $\Lambda \in [0, 1]$ is the data forgetting factor of the estimator.

5. The actor

The actor commands the control effectors. In the J-SNAC algorithm, its definition comes the reward function and requires values of the co-state, control effectiveness, and exploratory actions to compute the control signal. These signals come from the critic, the model, and the action modifier systems.

6. Partial Derivative Estimation

In Figure 2, it can be seen that J-SNAC algorithm requires co-states (partial derivative of the value function with respect to the state measurements $\partial V / \partial x$) and time derivative of the value ($\partial V / \partial x \cdot \dot{x} \equiv \partial V / \partial t$). Furthermore in order to update estimate the control effectiveness parameter the time rate of the state measurements $\partial x / \partial t$ are required. A *back-propagation* through the function approximator is used for estimating the derivative $\partial V / \partial x$. The time derivatives of the states and the value function is estimated by using a derivative filter. The equation for this derivative filter in Laplace domain is given as,

$$Y(s) = \frac{s}{d \cdot s + 1} U(s) \tag{25}$$

with Y being the estimated time derivative of the signal U , s being the Laplace variable and d being an adjustable filter coefficient.

III. Flight Control Systems Design

This section explains the objective of the proposed lateral-directional flight control system. Furthermore, this section elaborates the use of J-SNAC for the design of the flight control system.

A. Control Objective

The control objective considered here is to perform coordinated turns at a given flight altitude and airspeed. Such a task entails maintaining a zero side-slip condition (regulation problem) and tracking the desired aircraft heading angles (tracking problem). The strategy is to manipulate the rudder deflections δ_r to regulate the side-slips ($\beta = 0$) and produce desirable roll angles ϕ_r to track the heading angles ψ_r . The desired roll angles ϕ_r are attained by manipulating the aileron deflections δ_a .

B. Lateral-Directional Flight Control System Design with J-SNAC

In this work, a distributed architecture is chosen for the design lateral-directional flight control system. Its modularity and minimization of dimensionality motivate the choice of the architecture. The proposed flight control system consists of three J-SNAC controllers, one for regulating side-slip (β) angle, one for tracking desired roll angle ϕ_r and the other one is for producing desired roll angle ϕ_r to track desired heading angle ψ_r . All three controllers have the structure depicted in Figure 2.

Figure 3. Placement of Normalized Radial Basis Functions in the state-space for the side-slip regulator and the roll tracker

1. Side-Slip Regulator Design

The J-SNAC side slip regulator takes the vector signal $[\beta_m \ r_m(t)]^T$ as its input and outputs the scalar signal $u_r(t)$. β_m is the measured/estimated side-slip angle, r_m is the measured body yaw rate and $u_r(t)$ is the command signal for the rudder actuator.

The reward function for this regulator is defined as,

$$\rho(\beta_m, r_m, u_r) = -2\beta_m^2 - c_r \frac{4}{\pi^2} u_{r_{\max}} \log \left(\left| \frac{1}{\cos \left(\frac{\pi^2}{4} \frac{u_r}{u_{r_{\max}}} \right)} \right| \right) \quad (26)$$

The action-depended part in the reward function implies following the actor function,

$$u_r(t) = \frac{2 \cdot u_{r_{\max}}}{\pi} \arctan \left(\frac{\pi}{2} \left(\frac{1}{c_r} [\partial V / \partial \beta \ \partial V / \partial r] \begin{bmatrix} \partial \dot{\beta} / \partial u_r \\ \partial \dot{r} / \partial u_r \end{bmatrix} + u_{n,\beta} \right) \right) \quad (27)$$

Table 1. Hyper-Parameters for side-slip controller

Variable	Value	Units
Maximum surface deflections ($u_{r_{\max}}$)	30	degrees
Discounting time horizon (τ_β)	0.1	s
Eligibility trace time constant (κ_β)	0.01	s
Action cost parameter (c_β)	0.1	-
Exploration noise filter time constant ($\tau_{n,\beta}$)	5	s
Learning rate ($\alpha_\beta(t)$)	1	-
Exploration noise intensity ($\sigma_{0,\beta}$)	30	degrees
Derivative filter time constant (d_β)	0.02	s

The NRBF network used in side-slip regulator for learning the value function consists of 181 basis functions distributed in a hexagonal pattern (see Figure 3). The spreads of each basis function are the

defined with Eq (28), where r_i is the spread of i^{th} basis function and ζ_i is the Euclidean distance to the nearest basis function. The learning process only updates the amplitudes of the basis functions to reduce the required computations further.

$$r_i = \frac{1}{\sqrt{2}\zeta_i} \quad (28)$$

The control effectiveness parameters has been estimated with the incremental identification procedure (see Eq. (24)). The state vector and control vector for the estimator are $[\Delta\phi \ \Delta\beta \ \Delta p \ \Delta r]^T$ and $[\Delta u_a \ \Delta u_r]^T$.

Implemented hyper-parameters for this controller are given in Table 1.

2. Roll Angle Controller

The J-SNAC roll angle controller takes the vector signal $[e_\phi \ p_m(t)]^T$ as its input and outputs scalar signal $u_a(t)$. e_ϕ is the difference between the reference for roll angle ϕ_r and the measured roll angle ϕ_m . p_m is the measured body roll rate and u_a is the command signal for the aileron actuator. The reward function for this tracker is defined as

$$\rho(e_\phi, p_m, u_a) = -e_\phi^2 - \frac{p_m^2}{8} - c_a \frac{4}{\pi^2} u_{a_{\max}} \log \left(\left| \frac{1}{\cos \left(\frac{\pi^2}{4} \frac{u_a}{u_{a_{\max}}} \right)} \right| \right) \quad (29)$$

The action-depended part in the reward function implies following actor function for the roll tracker,

$$u_a(t) = \frac{2 \cdot u_{a_{\max}}}{\pi} \arctan \left(\frac{\pi}{2} \left(\frac{1}{c_a} [\partial V / \partial e_\phi \partial V / \partial p] \left[\frac{\partial \dot{e}_\phi / \partial u_a}{\partial \dot{p} / \partial u_a} \right] + u_{n,\phi} \right) \right) \quad (30)$$

Table 2. Hyper-Parameters for roll controller

Variable	Value	Units
Maximum surface deflections ($u_{\max}$)	21.5	degrees
Discounting time horizon (τ_ϕ)	0.1	s
Eligibility trace time constant (κ_ϕ)	0.01	s
Action cost parameter (c_ϕ)	0.1	-
Exploration noise filter time constant ($\tau_{n,\phi}$)	5	s
Learning rate (α_ϕ)	1	-
Exploration noise intensity ($\sigma_{0,\phi}$)	21.5	degrees

The NRBF network and control effectiveness identification for roll tracker is identical to that of the side-slip regulator. Implemented hyper-parameters for roll tracker are listed in Table 2.

3. Heading Angle Controller

The J-SNAC heading angle controller takes the scalar signal $e_\psi(t)$ as its input and outputs the scalar signal $\phi_r(t)$. $e_\psi(t)$ is the difference between the reference for heading angle $\psi_r(t)$ and the true heading angle $\psi_m(t)$. $\phi_r(t)$ is the reference signal for the roll angle controller. The reward function for this tracker is defined as

$$\rho(e_\psi, \phi_r(t)) = -0.5e_\psi^2 - c_{\phi_r} \frac{4}{\pi^2} \phi_{r_{\max}} \log \left(\left| \frac{1}{\cos \left(\frac{\pi^2}{4} \frac{\phi_r}{\phi_{r_{\max}}} \right)} \right| \right) \quad (31)$$

The action-depended reward part implies following actor function for the heading angle tracker,

$$u_r(t) = \frac{2 \cdot \phi_{r_{\max}}}{\pi} \arctan \left(\frac{\pi}{2} \left(\frac{1}{c_{\phi_r}} \frac{\partial V}{\partial e_\psi} \frac{\partial \dot{e}_\psi}{\partial \phi_r} + u_{n,\psi} \right) \right) \quad (32)$$

The NRBF network for heading angle tracker consisted of 25 basis function evenly distributed in within the space of $[-2\pi \ 2\pi]$. The spread of each basis function is according to Eq. (28). Since the kinematic

equation that determines the heading angle is non-changing, the control effectiveness is set with a desired value of $\partial\psi/\partial\phi_r = 0.5$. Implemented hyper-parameters for this controller are listed in Table 3.

Table 3. Hyper-Parameters for heading controller

Variable	Value	Units
Maximum roll command ($\phi_{r_{\max}}$)	68.76	degrees
Discounting time horizon (τ_ψ)	0.1	s
Eligibility trace time constant (κ_ψ)	0.01	s
Action cost parameter (c_ψ)	0.001	-
Exploration noise filter time constant ($\tau_{n,\psi}$)	5	s
Learning rate (α_ψ)	0.002	-
Exploration noise intensity ($\sigma_{0,\psi}$)	68.76	degrees

IV. Controller Training and Evaluation Method

This section presents the simulation setup, the controller training, and evaluation methods. Furthermore, it gives the design of the PID flight controllers, used for stabilizing the longitudinal flight dynamics and benchmarking the proposed J-SNAC flight controller.

A. Aircraft Model and Simulation Setup

The proposed lateral-directional flight control system was trained and evaluated in a Simulation environment made with MATLAB and Simulink. This setup used *Fourth-Order Runge-Kutta Solver* with a fundamental time step of 0.02s to calculate the state evolution. The simulation setup consisted a nonlinear model of the F16 aircraft³⁴ and the controllers (see Figure 4).

The aircraft model used in the setup has traditional aerodynamic control surfaces (i.e., aileron, elevator, and rudder) and a single engine. Furthermore, the model consists first order lag filters with bounded rate and values to model the aerodynamics surface actuators and the engine.

The aircraft is initialized at a *steady-symmetric flight* condition at an altitude of 5000 ft and airspeed of 600 ft/s. The state values at this trim conditions are given in Table 4.

Table 4. Trim condition for the simulation setup

Variable	Value	Units
Altitude (h)	5000	ft
Airspeed (V)	600	ft/s
Mach number (M)	0.5470	-
Angle of attack (α)	1.5579	degrees
Angle of Side slip (β)	0	degrees
Pitch angle (θ)	1.5579	degrees
Throttle Setting (δ_{th})	2.5942×10^3	lbf
Elevator Deflection (δ_e)	1.7640	degrees
Rudder Deflection (δ_r)	0	degrees
Aileron Deflection (δ_a)	0	degrees

B. Fixed Gain Controller Design

In Section II, it was assumed that the effects of longitudinal state variable on lateral-directional state dynamics are minimum. For this assumption to hold, longitudinal dynamics controllers are necessary. Here, a set of fixed gain linear controllers were designed to hold the longitudinal states close to their trimmed

values. Furthermore, to provide a benchmark for the proposed J-SNAC based lateral-directional flight controller, another set of fixed-gain linear controllers were designed for controlling the lateral-directional flight controller. Figure 4 depicts how longitudinal flight controllers work in tandem with the lateral-directional flight controller.

1. Longitudinal Dynamics Controller Design

The function of the longitudinal flight controller is to hold a longitudinal state (i.e., altitude h , airspeed V , pitch angle θ , the angle of attack α , pitch rate q) at a constant value. Figure 5(a) shows the structure of the longitudinal flight controller used in this work.

This flight controller consists of three PID control laws, two of which work together to hold a reference flight altitude h_r and the other one holds a reference airspeed $V_r(t)$. The altitude regulator takes in desired altitude $h_r(t)$ and measured altitude $h_m(t)$ as its input and outputs a desired pitch angle $\theta_r(t)$. The control law for this controller is defined with Eq. (33). In these equations θ_r , $K_{P_{e_h}}$, $K_{I_{e_h}}$, $K_{D_{e_h}}$ stands for desired pitch angle and PID gains of the controller.

$$\begin{aligned}\theta_r(t) &= K_{P_{e_h}} e_h(t) + K_{I_{e_h}} \int_{t_0}^t e_h(\tau) d\tau + K_{D_{e_h}} \dot{e}_h(t) \\ e_h(t) &= h_r(t) - h_m(t)\end{aligned}\quad (33)$$

The pitch controller takes in the desired pitch angle $\theta_r(t)$ from the altitude regulator, measured pitch angle $\theta_m(t)$ and pitch rate $q_m(t)$ from the sensors as its input and outputs dynamic command for elevator deflections $u_e^c(t)$. The control law for this controller is defined in Eq. (34).

$$u_e^c(t) = \theta_r(t) - K_\theta \theta_m(t) - K_q q_m(t) \quad (34)$$

The combination of two signals determines the actual elevator deflection. The first signal is a dynamic signal $u_e^c(t)$ generated by the pitch controller and the second signal is a static signal $u_e^{tr}(t)$ determined from trimming routine.

The airspeed regulator takes in the desired airspeed $V_r(t)$ and the measured airspeed $V_m(t)$ as its input and outputs a dynamic throttle command signal determined with Eq. (35). In these equations, $u_{th}^c(t)$ stands for dynamic throttle command signal, $K_{P_{e_V}}$, $K_{I_{e_V}}$ and $K_{D_{e_V}}$ stands for the PID gains.

$$\begin{aligned}u_{th}^c(t) &= K_{P_{e_V}} e_V(t) + K_{I_{e_V}} \int_{t_0}^t e_V(\tau) d\tau + K_{D_{e_V}} \dot{e}_V(t) \\ e_V(t) &= V_r(t) - V_m(t)\end{aligned}\quad (35)$$

Similar to the elevator, the throttle setting is determined by the combination of a dynamic u_{th}^c and a static signal u_{th}^{tr} . The dynamic signal comes from the airspeed controller, and the static signal comes from the trimming routine.

There are eight parameters, namely $K_{P_{e_h}}$, $K_{I_{e_h}}$, $K_{D_{e_h}}$, K_θ , K_q , $K_{P_{e_V}}$, $K_{I_{e_V}}$ and $K_{D_{e_V}}$, in the longitudinal flight controller. These parameters were tuned with *root locus* and *successive loop closure methods*, to meet the specifications for the category B flight phase and level 1 flying qualities, as stipulated in MIL-F-8785C.³⁵ The determined gain values are given in Table 5.

Figure 4. F16 aircraft model with flight controllers. x_{long}^{ref} and x_{lat}^{ref} are the external command signals for longitudinal and lateral states respectively. x_{long}^m and x_{lat}^m are the measured/estimated signals for longitudinal and lateral states. u_{th} , u_e , u_a and u_r are the command signals for the flight control surfaces and the engine.

Figure 5. Internal structure of decoupled flight controllers. The purpose of the longitudinal flight control-system is to hold a specific altitude and flight velocity. The purpose of lateral-directional control-system is to perform coordinated turns. Sub-controllers in longitudinal flight controller consist of a PID law. Sub-controllers of the lateral-directional flight controller consist of either J-SNAC or PID control law.

2. Lateral-Directional Dynamics Controller Design

The purpose of lateral-directional flight control system is to perform the same control objective as J-SNAC flight controller, i.e., coordinated turns. This linear flight controller has a similar structure to the J-SNAC controller (see, Figure 5(b)).

Similar to the longitudinal-dynamics controller, these controllers were designed to meet the specification provided in MIL-F-8785C, with root-locus and successive loop closure methods.

The linear heading tracker takes desired heading angle $\psi_r(t)$ and measured heading angle $\psi_m(t)$ as its input and outputs a desired roll angle $\phi_r(t)$. The control law is defined with Eq. (36). In these equations ϕ_r , $K_{P_{e_\psi}}$, $K_{I_{e_\psi}}$, $K_{D_{e_\psi}}$ stands for desired roll angle and PID gains of the controller.

$$\begin{aligned}\phi_r(t) &= K_{P_{e_\psi}} e_\psi(t) + K_{I_{e_\psi}} \int_{t_0}^t e_\psi(\tau) d\tau + K_{D_{e_\psi}} \dot{e}_\psi(t) \\ e_\psi(t) &= \psi_r(t) - \psi_m(t)\end{aligned}\quad (36)$$

Table 5. Longitudinal controller parameter values for holding F16 at an altitude of 5000 feet and with an airspeed of 600 feet per second.

Parameter	Values	Parameter	Values
$K_{P_{e_h}}$	-0.0113	K_q	-0.0682
$K_{I_{e_h}}$	-0.0059	$K_{P_{e_v}}$	16759
$K_{D_{e_h}}$	-0.0328	$K_{I_{e_v}}$	9545
K_θ	-0.0367	$K_{D_{e_v}}$	5206

The side-slip regulator takes in the reference side slip angle $\beta_r(t) = 0$, measured side slip angle $\beta_m(t)$ and measured yaw rate r_m as its input and outputs a dynamic rudder command signal determined with Eq. (38), (39) and (40). This rudder controller contains a wash-out filter to augment yaw rate measurements. In the controller Equations the washed-out yaw rate measurement is given by $w(t)$. Furthermore, in the equations $u_r^c(t)$ stands for dynamic rudder deflection signal, $K_{I_{e_\beta}}$ and K_w stands for the controller gains.

The roll angle controller takes desired roll angle ϕ_r from the heading tracker, measured roll angle ϕ_m

and roll rate p_m from the sensors/estimators. The control logic for this controller is given by Eq. (37). In these equations p_m is the measured roll rate, ϕ_m is the measured roll angle, $u_a^c(t)$ is the dynamic command for aileron deflections, $K_{P_{e_\phi}}$, $K_{I_{e_\phi}}$, $K_{D_{e_\phi}}$ and K_p are the tunable controller parameters.

$$\begin{aligned} u_a^c(t) &= K_{P_{e_\phi}} e_\phi(t) + K_{I_{e_\phi}} \int_{t_0}^t e_\phi(\tau) d\tau + K_{D_{e_\phi}} \dot{e}_\phi(t) - K_p p_m(t) \\ e_\phi(t) &= \phi_r(t) - \phi_m(t) \end{aligned} \quad (37)$$

The combination of two signals determines aileron deflection. The first signal is a dynamic signal $u_a^c(t)$ generated by the aileron regulator and the second signal is a static signal $u_a^{tr}(t)$ determined from trimming routine.

$$u_r^c(t) = K_{I_{e_\beta}} \int_{t_0}^t e_\beta(\tau) d\tau + K_w w(t) \quad (38)$$

$$e_\beta(t) = \beta_r(t) - \beta_m(t) = -\beta_m(t) \quad (39)$$

$$\dot{w}(t) = -w(t) + r_m(t) \quad (40)$$

Similar to all other controllers, the combination of a dynamic u_r^c and a static signal u_r^{tr} determines the rudder deflection. The dynamic signal comes from the rudder regulator, and the static signal comes from the trimming routine.

There are nine parameters, namely $K_{P_{e_\psi}}$, $K_{I_{e_\psi}}$, $K_{D_{e_\psi}}$, $K_{P_{e_\phi}}$, $K_{I_{e_\phi}}$, $K_{D_{e_\phi}}$, K_p , $K_{I_{e_\beta}}$ and K_w , in the linear lateral-directional-flight controller that needs tuning. The determined gain values are given in Table 6.

Table 6. Lateral-directional-controller parameter values for making coordinated turns to track heading commands with F16 at an altitude of 5000 feet and with an airspeed of 600 feet per second.

Parameters	Values	Parameters	Values	Parameters	Values
$K_{P_{e_\psi}}$	27.40	$K_{P_{e_\phi}}$	-1.71	K_p	-0.07
$K_{I_{e_\psi}}$	1.45	$K_{I_{e_\phi}}$	-1.50	$K_{I_{e_\beta}}$	0.70
$K_{D_{e_\psi}}$	-16.54	$K_{D_{e_\phi}}$	-0.48	K_w	0.12

C. J-SNAC Flight Controller Training Method

The J-SNAC controller was initialized with zero knowledge about control task and then was trained in a two-step training procedure. In the first training sequence, the side-slip regulator and the roll angle controller was trained to track roll command signals with zero side-slips. Next, the heading angle controller was added to the flight control system and then trained together to follow heading angle commands.

1. Training of Side-Slip Regulator and Bank Angle Controller

During this phase of training, the slide slip regulator and roll angle controller is trained to track roll command signals with zero-side slips. The training session consisted of 305 episodes, where each episode lasted for 180 seconds. Each episode started at the trimmed condition mentioned earlier.

A cascaded system consisting of a sine wave generator, a static-gain, and a zero-order hold filter (see Figure 6) generates the commanded roll angles. Throughout training sessions, the sine wave generator produced a sine wave with an amplitude of $\pi/3$ radian and frequency of 1/180 Hz. The gain block is responsible for altering the sign of the sine signal randomly. This random switching is done to promote even exploration of the state-space. The zero-order hold filter is used to convert the sine signal into variable step signal. The variable step signals are generated by setting the sampling time of the zero-order filter with following law.

$$T = \text{mod}(N - 1, 61) \quad (41)$$

Figure 6. System to generate reference signals for training.

In Eq. 41, T stands for sampling time, and N is the episode number and “mod” stands for remainder operator. When the $T = 0$ the reference signal is a pure sine signal. When T is an integer, the reference generator produced block signals with varying levels.

These type of reference signals are chosen to make the tracking task gradually demanding across the training episodes and then repeating the tracking tasks five times.

2. Training of Heading Angle Training

Upon the completion of initial training of side-slip regulator and roll angle controller, the heading angle controller is added to the flight control system. The learning rate of the roll tracker and side-slip regulator is set to zero as it is desired to train the heading controller alone. The training session is similar to the previous training sequence, i.e., using the same reference signal generator. One of the differences between this and previous training session is that the heading angle controller was trained over 124 training episodes. Other difference is that the sinusoidal signal generator generated following the reference signal,

$$\psi_r(t) = \frac{3}{4}\pi \sin\left(\frac{2\pi}{180}t - \frac{\pi}{2}\right) + \frac{\pi}{2} \quad (42)$$

D. Controller Performance Evaluation

After the training, the proposed J-SNAC based lateral-direction flight controller was evaluated for its learning and control performance. At first, the controller is qualitatively assessed for its learning performances. Next, the controller is evaluated quantitatively for its control performances.

1. Training Performance Evaluation

The goal of this evaluation is to assess the training process and its effects on the value and policy functions. The training process is evaluated by observing the region of state-space covered by the controller and observing the change of policy function across the training episodes. Effects of training on the value and policy functions are evaluated by comparing their surfaces before and after the training processes.

2. Control Performance Evaluation

The goal of this evaluation is to quantify the control performance of the proposed controller before and after the training, then compare these performances with the performance of the benchmarking controller. Furthermore, control performance was also evaluated for robustness against sensor noise and partial failure of the aileron.

The performance of the proposed controller is quantified with the performance index PI defined in Eq. 43. The defined performance index is a weighted sum of normalized root mean squared (RMS) errors in desired altitude, airspeed, side-slip angle, and heading angle. Altitude and velocity are included in the PI to quantify the effects on the longitudinal flight controller. Side-slip and heading angles are included in the PI because they are the principal variables of interest. The error in altitude and airspeed are normalized with 25 feet and 10 feet per second. The error in heading and the side-slip angle is normalized with 2 degrees.

$$PI = -0.1 \cdot \sqrt{\frac{1}{T} \int_0^T \left(\frac{h(t) - h_r(t)}{25} \right)^2 dt} - 0.1 \cdot \sqrt{\frac{1}{T} \int_0^T \left(\frac{V(t) - V_r(t)}{10} \right)^2 dt} - 0.4 \cdot \sqrt{\frac{1}{T} \int_0^T \left(\frac{\beta(t) - \beta_r(t)}{2} \right)^2 dt} - 0.4 \cdot \sqrt{\frac{1}{T} \int_0^T \left(\frac{\psi(t) - \psi_r(t)}{2} \right)^2 dt} \quad (43)$$

The control performance of the controller was compared with the bench-marking fixed gain controller for tracking a sinusoid and a smoothened step signal under nominal conditions.

Then the controller was evaluated for robustness against sensor noise and partial failure of the aileron. The sensor noise is simulated by corrupting the rotational rate signals (i.e., roll rate p and yaw rate r) with zero mean Gaussian noise. The partial loss of aileron was simulated by halving the command signals and adding 7 degrees bias to this split signal.

V. Results and Discussion

This section presents and discusses the results from the training and performance evaluation procedures.

A. Affects of Training on the Value and Policy functions

Figure 7 shows the region of state-space that the J-SNAC flight controllers have explored while being trained. Although the roll and heading angle trackers have experienced most parts of the state-space, the side slip regulator has not experienced much of the state-space. This disparity between the explored regions by controllers is because of the training schedule. The reference signals used for training have made the roll and heading angle trackers explore most of the allowed state-space. However, since all training episodes started at zero-side-slip conditions, the exploration signal produced by J-SNAC side-slip regulator was insignificant. Furthermore, disturbances in side-slip angles while rolling was also small.

Figure 7. Depiction of parts of the state-space visited by the J-SNAC controllers during their training. The rectangular box represents the bounds in the state-space within which the controllers can learn its policy.

Figure 8 depicts the trajectory of policy function monitoring parameters ($\Delta h_{\delta_r}, \Delta h_{\delta_a}, \Delta \phi_r$) across the training episodes. The policy function monitoring parameters were defined with the RMS of changes in control actions assigned to a list of preselected states. In figure 8(a) and 8(b), it is observed that initially both the side-slip regulator and roll angle tracker changes rapidly. This rapid change is because of large initial TD errors. Next notable observation in these figures is that every 61 training episode there is a drop in the rate of change. This drop in the rate of change is because of the process of generating the tracking reference signal, which changed gradually over 61 episodes and then repeated after every 61 episodes. Additionally, the rate of change of side-slip and roll tracker policies are decreasing over the episodes, due to the declining TD error. The policies did not converge to a stationary form as there are unexplored regions in the state-space. With more training and possibly with better training scheme policy could converge.

According to Figure 8(c), the heading angle policy changed rapidly during the first episode and afterward there is a slow increase in the change of policy with some fluctuations. Rapid change in the first episode is due to high TD error in the first episode, and small variations after that are due to the exploration of state space and declining TD error.

Figure 8. Change in policy function tracking parameter across training episodes.

Figures 9 and 10 and shows the value and policy functions learned by the J-SNAC controllers after their training. Before the training, all of these functions have zero outputs for all input.

Figure 9. Value functions after training.

From these observations, it can be concluded that the J-SNAC algorithm could perform its learning function. However, the learned functions did not convergence due to the training program and the chosen hyper-parameters.

B. Difference in Performances Before and After Training

Figure 11 shows the state trajectories of the aircraft when it used benchmarking PID controller, non-trained and trained J-SNAC controllers for tracking sinusoidal reference signal. As expected, non-trained flight controller failed to follow the reference signal and eventually crash the aircraft after 50 seconds. The crash is due to unreasonable deflection of ailerons, causing high roll rate which then destabilizes the longitudinal controllers. After the training performance of PID and J-SNAC controller are almost similar. One of the differences between the performances of these controllers is that side-slip regulator designed with PID law attenuates incurred side-slips better. Furthermore, the J-SNAC controller has a delay in following heading commands compared to the PID controller.

Figure 12 depicts the state evolution of the aircraft for tracking a smoothened step signal. Similar to the tracking of the sinusoid, non-trained J-SNAC controller failed to perform the tracking while trained J-SNAC

Figure 10. Policies learned by each of the controllers

and PID controller performs almost the same. Again, PID side-slip regulator attenuates incurred side-slip better, and J-SNAC controller has a small delay in tracking. One additional difference is that PID controllers create more aggressive commands for the aerodynamic surface actuators.

Table 7 shows the performance score of PID, non-trained and trained J-SNAC controller according to the Eq.(43). The performance scores are in agreement with the visual analysis, i.e., the non-trained controller cannot perform the control task; trained controller performs almost similar but lower than that of the PID controllers. The lower score is due to the delay in tracking and lower attenuation of side-slips.

Table 7. Performance according to the Index given in Eq. (43)

Tracking task	Controller setting	PI value
sin wave	Non-trained J-SNAC	-52.0257
	Trained J-SNAC	-1.5802
	PID	-0.1839
smooth-step	Non-trained J-SNAC	-4.8565
	Trained J-SNAC	-0.3440
	PID	-0.1623

C. Robustness Against Sensor Noise

Figure 14 shows the aircraft state evolution while tracking sinusoidal heading commands in the presence of noise in the rate measurements. The sensor noise is simulated by adding zero-mean noise signals with the roll and yaw rate signals. The noise signals have a standard deviation of 5 degrees/s.

The tracking performance for both controllers was satisfactory, as both have tracked the reference heading angles. Although, J-SNAC controller produced a more noisy command signal for the aileron actuators and almost no commands for the rudder actuator. The noisy command signal is because the J-SNAC algorithm does not have any internal filtering procedures. Concerning the tracking, J-SNAC controller again has a delay. Also, J-SNAC controller did not compensate for a small increment in side-slips, because in the learned policy these small side-slips are mapped to no-rudder actions.

According to the defined performance index, the score of J-SNAC flight controller is -1.5917 and the score of the PID controller is -0.2719.

D. Control Adaptation During Partial Loss of Flight Control Surfaces

Figure 14 shows the aircraft state evolution while tracking sinusoidal heading commands in the presence of aileron actuator failure.

Figure 11. Tracking of sinusoidal reference signal with PID, non-trained and trained J-SNAC controller

Figure 12. Tracking of smooth step signal with PID, non-trained and trained J-SNAC controller.

Figure 13. Effect of noise in rate measurements for the PID and J-SNAC flight controllers.

As can be seen, the performance from the J-SNAC controller is smooth, and it provides an excellent tracking performance while PID controller fails to track after few seconds of failure. The continuous tracking by J-SNAC is due to the immediate identification of the reduced CE and adaptation of the control law according to this new CE. Where PID does not have any CE identification procedure, and due to the mismatch between the design and real model, the PID controller produces aggressive and high deflections for aileron which then destabilizes the aircraft flight.

According to the defined performance index, the score of J-SNAC flight controller is -1.7953 and the score of the PID controller is -92.9344.

VI. Conclusion

In this paper, design, and evaluation of a reinforcement-learning lateral-directional flight controller have been discussed. The proposed flight controller has a modular structure and is designed with the J-SNAC algorithm, incremental identification of control effectiveness and normalized radial basis function network. The proposed flight controller was applied to an F-16 non-linear model and trained to track heading commands with co-ordinated turns. The trained controller was evaluated for tracking tasks under the nominal condition, in presence sensor noise, and with aileron hard-over.

The simulation results confirm that J-SNAC algorithm along with incremental identification of control effectiveness is viable for the design of adaptive flight controllers. The control performance of a semi-trained J-SNAC flight controller close to a human-designed linear flight controller both with and without sensor noise. However, non-convergent policies make the tracking performance of the proposed controller lower. However, its autonomous learning and adaptability in the presence of uncertainty allow the proposed controller to adapt aileron hard-overs.

The tracking performance of the proposed controller can be further improved by adopting training procedures that facilitate more exploration of the state-space and guarantee the convergence of learned policies. Further improvement in the ACD based flight controller could be made by investigating on use of different function approximation structures with the J-SNAC algorithm and utilizing the best performing structure. In this work, control effectiveness was determined with an ad-hoc estimator, improvement in control-effectiveness determination can improve the learning and control performance. Also, the stability of the learning process was neglected in the current study. Before implementing on physical aircraft, the stability of the learning process is required to be ensured. Capabilities of the proposed flight controller can be expanded by combining it with reinforcement-learning longitudinal flight controllers; incorporating information exchange within sub-controllers; enlarging the training schedule to include the full-flight envelope (altitude and airspeed); incorporating safe learning;³⁶ investigating the controller performance for other fault scenarios and validating the simulation studies with experimental studies.

References

- ¹Belcastro, C. M. and Foster, J. V., "Aircraft loss-of-control accident analysis," in "Proceedings of AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Paper No. AIAA-2010-8004," , 2010.
- ²Belcastro, C. M., Foster, J. V., Shah, G. H., Gregory, I. M., Cox, D. E., Crider, D. A., Groff, L., Newman, R. L., and Klyde, D. H., "Aircraft Loss of Control Problem Analysis and Research Toward a Holistic Solution," *Journal of Guidance, Control, and Dynamics*.
- ³Safety, B. A., "Statistical Summary of Commercial Jet Aircraft Accidents: Worldwide Operations, 1959-2016," *Boeing Commercial Airplane, Seattle, WA*.
- ⁴Safety, I., "Safety Report," techreport, International Civil Aviation Organization, Montreal, Canada, 2017.
- ⁵Werbos, P. J., "Approximate dynamic programming for real-time control and neural modeling," , 1992.
- ⁶Werbos, P. J., "Neurocontrol and supervised learning: An overview and evaluation," *Handbook of intelligent control*, Vol. 65, 1992, p. 89.
- ⁷Lewis, F. L. and Vrabie, D., "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, Vol. 9, No. 3.
- ⁸Si, J., *Handbook of learning and approximate dynamic programming*, Vol. 2, John Wiley & Sons, 2004.
- ⁹Werbos, P. J., "Reconfigurable flight control via neurodynamic programming and universally stable adaptive control," in "American Control Conference, 2001. Proceedings of the 2001," IEEE, Vol. 4, 2001, pp. 2896-2900.
- ¹⁰Lewis, F. L., Vrabie, D., and Vamvoudakis, K. G., "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems*, Vol. 32, No. 6, 2012, pp. 76-105.
- ¹¹Lewis, F. L. and Liu, D., *Reinforcement learning and approximate dynamic programming for feedback control*, Vol. 17, John Wiley & Sons, 2013.

Figure 14. Effect of a faulty aileron actuator at $t = 25$ s for the PID and J-SNAC flight controllers.

- ¹²Enns, R. and Si, J., "Helicopter flight-control reconfiguration for main rotor actuator failures," *Journal of Guidance Control and Dynamics*, Vol. 26, No. 4, 2003, pp. 572–584.
- ¹³Han, D. and Balakrishnan, S., "Adaptive critic-based neural networks for agile missile control," *Journal of Guidance Control and Dynamics*, Vol. 25, No. 2, 2002, pp. 404–406.
- ¹⁴Han, D. and Balakrishnan, S., "Robust adaptive critic based neural networks for speed-constrained agile missile control," in "Proceedings of the AIAA Guidance, Navigation, and Control Conference," , 1999.
- ¹⁵Han, D. and Balakrishnan, S., "Adaptive critic based neural networks for control-constrained agile missile control," in "American Control Conference, 1999. Proceedings of the 1999," IEEE, Vol. 4, 1999, pp. 2600–2604.
- ¹⁶Han, D. and Balakrishnan, S., "State-constrained agile missile control with adaptive-critic-based neural networks," *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 4, 2002, pp. 481–489.
- ¹⁷Ferrari, S. and Stengel, R. F., "An adaptive critic global controller," in "American Control Conference, 2002. Proceedings of the 2002," IEEE, Vol. 4, 2002, pp. 2665–2670.
- ¹⁸Ferrari, S. and Stengel, R. F., "Online adaptive critic flight control," *Journal of Guidance Control and Dynamics*, Vol. 27, No. 5, 2004, pp. 777–786.
- ¹⁹Van Kampen, E., Chu, Q., and Mulder, J., "Continuous adaptive critic flight control aided with approximated plant dynamics," in "AIAA Guidance, Navigation, and Control Conference and Exhibit," AIAA, 2006, p. 6429.
- ²⁰Nobleheart, W., Shivanapura Lakshmikanth, G., Chakravarthy, A., and Steck, J. E., "Single network adaptive critic (SNAC) architecture for optimal tracking control of a morphing aircraft during a pull-up maneuver," in "AIAA Guidance, Navigation, and Control (GNC) Conference," , 2013, p. 5003.
- ²¹Zhou, Y., van Kampen, E., and Chu, Q., "Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2018, pp. 493–496.
- ²²Doya, K., "Reinforcement learning in continuous time and space," *Neural computation*, Vol. 12, No. 1, 2000, pp. 219–245.
- ²³Ding, J., Heydari, A., and Balakrishnan, S., *Single Network Adaptive Critics Networks-Development, Analysis, and Applications*, John Wiley & Sons, Inc., Hoboken, New Jersey, chap. 5, pp. 98–118, 2013.
- ²⁴Padhi, R., Unnikrishnan, N., Wang, X., and Balakrishnan, S., "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Networks*, Vol. 19, No. 10, 2006, pp. 1648–1660.
- ²⁵Ding, J., Balakrishnan, S., and Lewis, F. L., "A cost function based single network adaptive critic architecture for optimal control synthesis for a class of nonlinear systems," in "Neural Networks (IJCNN), the 2010 International Joint Conference on," IEEE, 2010, pp. 1–8.
- ²⁶Sutton, R. S., Barto, A. G., and Williams, R. J., "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems*, Vol. 12, No. 2, 1992, pp. 19–22.
- ²⁷Bellman, R. E., *Dynamic Programming*, Princeton University Press, 1957.
- ²⁸Ding, J. and Balakrishnan, S., "An online nonlinear optimal controller synthesis for aircraft with model uncertainties," in "AIAA Guidance, Navigation, and Control Conference," , 2010, p. 7738.
- ²⁹Moody, J. and Darken, C. J., "Fast learning in networks of locally-tuned processing units," *Neural computation*, Vol. 1, No. 2, 1989, pp. 281–294.
- ³⁰Rao, A. V., Miller, D., Rose, K., and Gersho, A., "Mixture of experts regression modeling by deterministic annealing," *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, 1997, pp. 2811–2820.
- ³¹Zhou, Y., van Kampen, E., and Chu, Q., "Incremental model based heuristic dynamic programming for nonlinear adaptive flight control," in "Proceedings of the International Micro Air Vehicles Conference and Competition 2016, Beijing, China," , 2016.
- ³²Zhou, Y., van Kampen, E.-J., and Chu, Q. P., "Launch Vehicle Adaptive Flight Ccontrol With Incremental Model Based Heuristic Dynamic Programming," *International Astronautical Congress, Adelaide Australia*.
- ³³Morelli, E. A. and Klein, V., *Aircraft system identification: Theory and practice*, Sunflyte Enterprises Williamsburg, VA, 2016.
- ³⁴Russell, R. S., "Non-linear F-16 simulation using Simulink and Matlab," *University of Minnesota, Tech. paper*.
- ³⁵Moorhouse, D. and Woodcock, R., "Us military specification mil-f-8785c," Tech. rep., US Department of Defense Arlington County, 1980.
- ³⁶Mannucci, T., van Kampen, E., de Visser, C., and Chu, Q., "Safe exploration algorithms for reinforcement learning controllers," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 4, 2018, pp. 1069–1081.