Faculty of Aerospace Engineering

Adaptive Critic Control For Aircraft Lateral-Directional Dynamics

An Evaluation of J-SNAC Algorithm as a Fault Tolerant Flight Control Framework

Imrul Kayesh Ashraf

November 13, 2018

TU Delft Delft University of Technology

Challenge the future

Adaptive Critic Control For Aircraft Lateral-Directional Dynamics

An Evaluation of J-SNAC Algorithm as a Fault Tolerant Flight Control Framework

Master of Science Thesis

Imrul Kayesh Ashraf

November 13, 2018

Faculty of Aerospace Engineering · Delft University of Technology

Delft University of Technology

Delft University Of Technology
Department Of
Control and Simulation

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **"Adaptive Critic Control For Aircraft Lateral-Directional Dynamics"** by **Imrul Kayesh Ashraf** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: November 13, 2018

Readers:

Dr.Ir. Q. P. Chu

Dr.Ir. E. van Kampen

Dr. A. Sharpanskykh

Preface

What you have in your hand is my MSc thesis project report. Although this report is only disclosing my thought process regarding one Reinforcement Learning flight controller, it represents more. It is a tiny step towards artificial general intelligent aircraft, which I aspire to see in reality. It encompasses three concepts that intrigue me the most: *time, consciousness, and flying*. Once you have read it, I believe that you will have more ideas on Reinforcement Learning, Adaptive Critics, Fault Tolerant Flight Controllers, and the direction to realize intelligent aircraft in the near future.

I have worked on this piece for more than one and a half years, but I still feel that it is incomplete. While writing this, I have learned a lot about aircraft, macro world dynamics, optimizations, motion control, machine learning, and system identification. It has made me appreciate academia, the systematic process of knowledge creation, and approaching the unknown with realistic means. It has given me a glimpse into the universe' working. It has pointed out strengths and weaknesses. Lastly and most importantly, it has killed a schoolboy and given birth to a scientist.

Working on this project was not as easy as it might seem. It required several trials and tribulations. The literature concerning the topics in this report are not coherent to each other; they use different terminologies to make the life difficult for a perfectionist and information hoarder. Besides that, I had accommodated demons within me that impeded this report's completion. Only when I started reading the literature with my terms and had purged the demons, this report wrote itself through me.

This piece would not have been completed without support from great people. First of all, I would like to thank my parents, for being themselves and patient with me. Secondly, my utmost gratitude goes to my supervisor Dr. Ir. Erik-Jan Van Kampen. I do not have enough words to explain how much I have appreciated his guidance, encouragements and ever so slight pushes. Then I would like to thank my sisters Habiba, Esem, and Nusrat for reminding me what is most important. A ton of thanks goes to my colleague and dear friend, Ir. Manan Siddiquee, for decluttering my thoughts and showing me the way to be a true academician. After that, I would like to thank Anton, Tigran, Ishaq, and Malik for sharing their experience of writing MSc thesis. I would also like to thank my friend

Rakesh for his proofreading and edits. Then I would like to thank Tariqul, Kenan, Vashish, Usama, Niek, Valentin, Roberto and G. for their moral support. Last and not least, I would like to convey deep gratitude towards past and present C&S upper house members for the stimulating conversations, encouragements and engaging workplace. I would also like to thank my unmentioned friends and family for their empathy and kindness.

Imrul Kayesh Ashraf

Delft, November 2018

Acronyms

AC	Adaptive Critic
ACD	Adaptive Critic Design
AD	Action Dependent
AFTFC	Active FTFC
ANN	Artificial Neural Network
DHP	Dual Heuristic Programming
DP	Dynamic Programming
FCS	Flight Control Systems
FDD	Fault Detection and Identification
FDI	Fault Detection and Isolation
FTFC	Fault Tolerant Flight Control
GDHP	Generalized Dual Heuristic Programming
HDP	Heuristic Dynamic Programming
LOC-I	Loss of Control-In Flight
MDP	Markov Decision Process
MDPs	Markov Decision Processes
PFTFC	Passive FTFC
RL	Reinforcement Learning
RM	Control Reconfiguration Mechanism
SNAC	Single Network Adaptive Critic
TD	Temporal Difference

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Improvement of safety and performance are two essential criteria for the development of new technology. According to recent statistics, Loss of Control-In Flight (LOC-I) causes most of the catastrophic aircraft accidents [1, 2]. Currently, advanced methodologies are being investigated to prevent and recover aircraft from its precursors [3–5]. One such promising technology is adaptive control system. Such system promises to improve the fault tolerance by analytically adjusting the onboard control strategies [3, 4].

In so far, multiple methods have been proposed to design adaptive control systems [4, 6–13]. Among these methods, one promising group is Adaptive Critic Design (ACD)[14, 15]. These are a class of Reinforcement Learning (RL) algorithms that can learn control law for any arbitrary system [16]. The learning capability make these techniques better suitable for accommodating a wide range of non-nominal conditions [17]. However, due to the recency of their emergence, these methods are yet to be validated for Flight Control Systems (FCS) design.

In previous research, few of ACD algorithms were used to develop task-specific flight control laws [16, 18–25]. These works demonstrated that such algorithms could learn to attain different flight objectives, e.g., aircraft landing, pitch tracking and pull up maneuver, and could adapt the control law when warranted. However, most of these research considered only to control the longitudinal dynamics and have used relatively computationally heavier ACD architectures [26–28]. This thesis project is set out to address these two limitations by recommending a lateral-directional flight controller with a computationally simpler ACD architecture.

1-1 Thesis Objective

The objective of this thesis is to improve the fault-tolerance of fixed-wing aircraft by investigating the applicability of J-SNAC algorithm for the design of an adaptive lateral-directional flight controller.

1-2 Research Questions

Four sets of tasks were defined to attain the stated objective. These tasks concerned with determining the working principle of ACD algorithms, developing a controller with the J-SNAC algorithm, identifying the conditions for its successful implementation and evaluating the controller for driving lateral-directional dynamics of fixed-wing aircraft. Following research questions and subquestions were used to guide these tasks.

R.Q.1 What is ACD, specifically J-SNAC, in state-of-the-art literature?

- R.Q.1.1 What is ACD?
- R.Q.1.2 What are the theoretical differences between the different ACD architectures?
- R.Q.1.3 How does J-SNAC algorithm perform in comparison to other ACD architectures?

R.Q.2 What are the conditions for successful implementation of a J-Single Network Adaptive Critic (SNAC) controller?

- R.Q.2.1 How to use J-SNAC for the design of a controller for a nonlinear system?
- R.Q.2.2 What are the hyper-parameters of a controller designed with J-SNAC?
- R.Q.2.3 How sensitive is the controller performance to changes in hyper-parameters?

R.Q.3 To what extent does the proposed controller improve the performance and survivability of fixed wing aircraft?

- R.Q.3.1 How does the proposed controller perform in comparison to a traditional fixed-gain linear controller?
- R.Q.3.2 How does the proposed controller perform as an adaptive controller?
- R.Q.3.3 To what extent can the controller performances be generalized for fixed-wing aircraft control?

1-3 Research Approach

The research activities are distributed in three phases, namely, literature review, preliminary study, and the aircraft implementation phases.

Literature Review

In the first phase, state-of-the-art literature was studied to answer research question R.Q.1.

Preliminary Study

In the preliminary study phase, the J-SNAC algorithm was analyzed. The goal of this study was to confirm the design methodology and identify the hyperparameters of the algorithm. This study aimed to answer research question R.Q.2.

Controller Implementation and Evaluation

Subsequently, a lateral-directional flight controller was developed with the J-SNAC algorithm. The controller was implemented to control a simulation model of F16 [29]. The performance of the controller was evaluated for tracking tasks and adaptation to unanticipated changes in the aircraft dynamics. The tracking performance was compared with a traditional flight controller, designed with linear control theory. The adaptability of the controller was investigated by measuring its efficacy with a specified performance index. Activities of this phase aided to answer the remaining research questions and conclude the research.

1-4 Report Outline

Following this introduction is an article, summarizing most of this research work. After that, Part II presents the results from the literature review and preliminary analysis phase. Then, Part III presents additional findings from the implementation and evaluation phase. Finally, the report concludes in Part IV with conclusions and recommendation.

For the reader who needs familiarization with fault-tolerant flight control and reinforcement learning, please refer to Chapter 2 and 3 in part II.

Part I

Article

Adaptive Critic Control For Aircraft Lateral-Directional Dynamics

Imrul K. Ashraf*

Delft University of Technology, PO Box 5, 2600 AA 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

Until now several RL Flight Controllers have been proposed with different ACD frameworks.[12–20] 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,[21–24] 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,

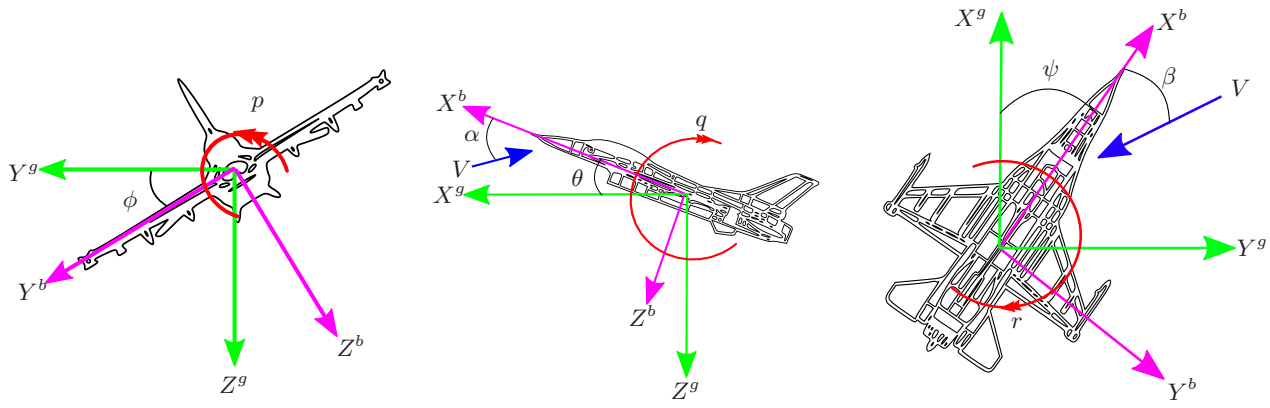
$$\begin{aligned}
\dot{\psi} &= \tfrac{1}{\cos\theta}(q\ \sin\phi + r\ \cos\phi) \\
\dot{\phi} &= p + \tan\theta(q\sin\phi + r\cos\phi) \\
\dot{\beta} &= \tfrac{Y}{m} + p\sin\alpha - r\cos\alpha + \tfrac{g}{V}\cos\beta\sin\phi\cos\theta + \tfrac{\sin\beta}{V}\left(g\cos\alpha\sin\theta - g\sin\alpha\cos\phi\cos\theta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

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.[10,25] The type of optimal control problem considered for the development of flight controller is called "Infinite Horizon Discounted Return Problem".[21] This problem is defined as follows.



(a) Front view (b) Left view (c) Top view

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,

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

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 $r(t)$,

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

The objective is to find a state feedback control law,

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

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

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

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,

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

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 $V^h(\mathbf{x})$,

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

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

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

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,[26] at time t, the optimal value function satisfies following self-consistency property.[21]

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

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.,

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

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 in[21] 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.

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

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 $\delta(t)$.

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

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

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

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

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

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,

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

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.[21,22,24,33]

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

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,

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

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

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

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

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

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.[21]

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[21,27,28] 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

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

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 signal[21] and it is generated with the following system of equations.

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

Where, σ_0 is the maximum perturbing action, $N(t)$ 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 $r(t)$.

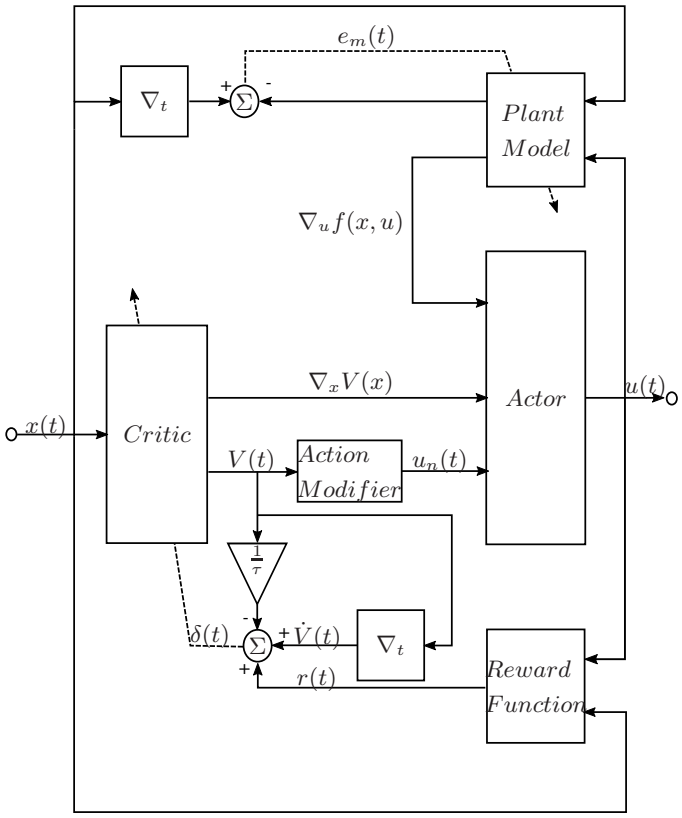


Figure 2. J-SNAC control algorithm. At time t, $x(t)$ is the state measurements, $u(t)$ is the action to be applied, $V(t)$ is value of the state x, $r(t)$ is the reward for being in state x and applying action u, $u_n(t)$ is an additive noise signal, $\delta(t)$ is the temporal difference, $f(x,u)$ is the system dynamics, and ∇ is system/operator to calculate partial derivatives (e.g. $\nabla_t x$ is the partial derivatives of x with respect to 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.[29,30] 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,

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

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,

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

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.[31] The system of equations

for the RLS estimator is as follows,

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

Where $\Delta\hat{\dot{x}}(t)$ 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 ($\partial V/\partial x \cdot \dot{\mathbf{x}} \equiv \partial V/\partial t$). 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

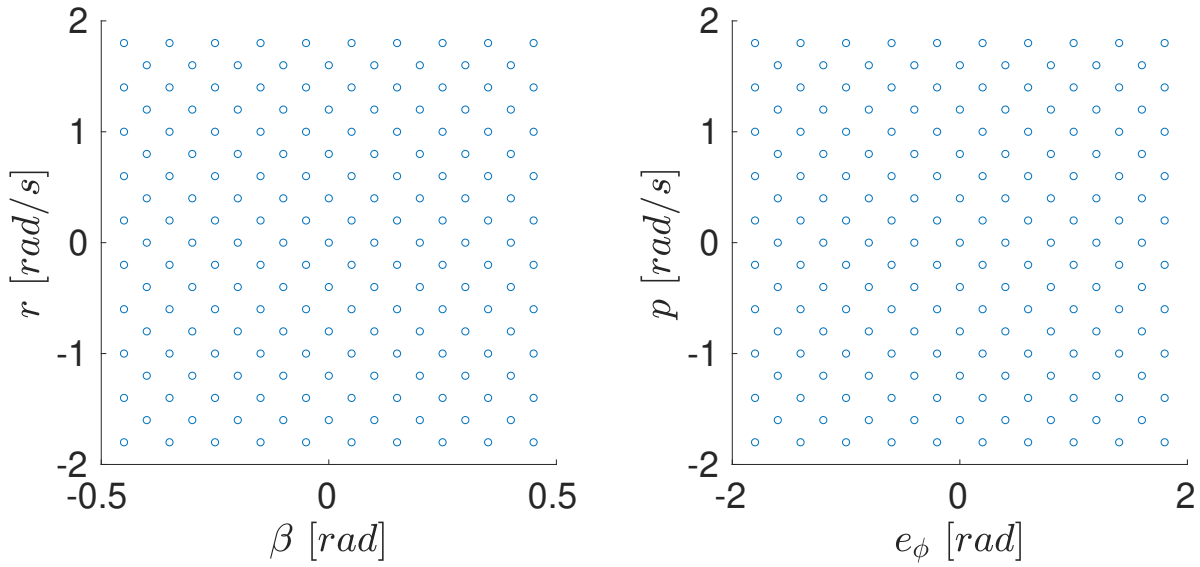
This section explains the objective of the proposed lateral-directional flight control system. Furthermore, this section elaborates the use of J-SNAC for the design of the flight control system.

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.



(a) side slip controller (b) roll angle controller

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,

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

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

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

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.

$$r_i = \frac{1}{\sqrt{2}\zeta_i} \tag{28}$$

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

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

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

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

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

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

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

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

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 aircraft[34] 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.

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

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).

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

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.

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

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.

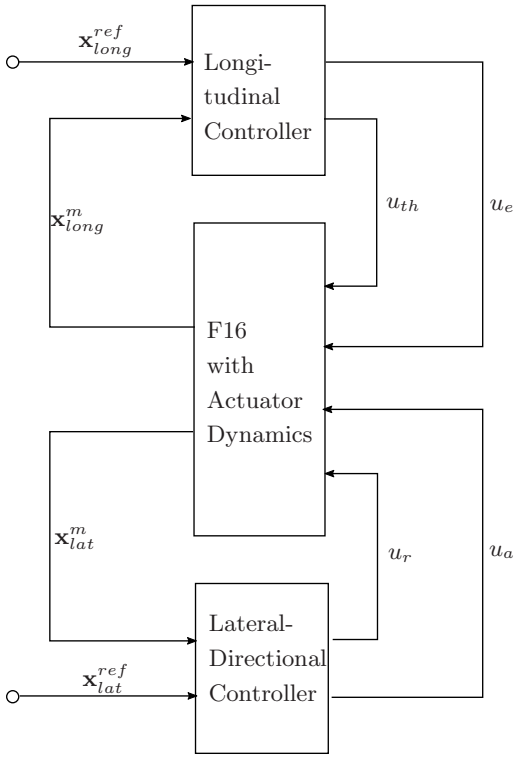
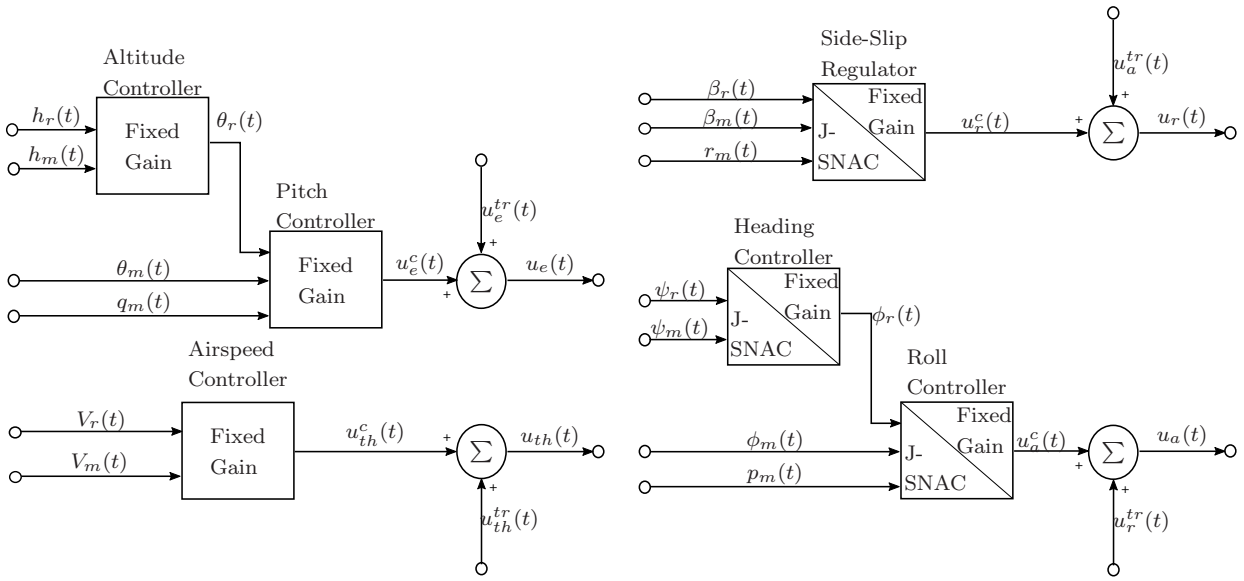


Figure 4. F16 aircraft model with flight controllers. $\mathbf{x}_{long}^{ref}$ and $\mathbf{x}_{lat}^{ref}$ are the external command signals for longitudinal and lateral states respectively. $\mathbf{x}_{long}^{m}$ and $\mathbf{x}_{lat}^{m}$ are the measured/estimated signals for longitudinal and lateral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



(a) Longitudinal Flight Controller

(b) Lateral-Directional flight controller

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.

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

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.

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

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.

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

$$e_\beta(t) = \beta_r(t) - \beta_m(t) = -\beta_m(t) \tag{39}$$

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

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.

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

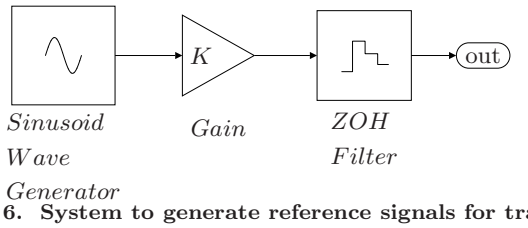


Figure 6. System to generate reference signals for training.

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,

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

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.

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

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.

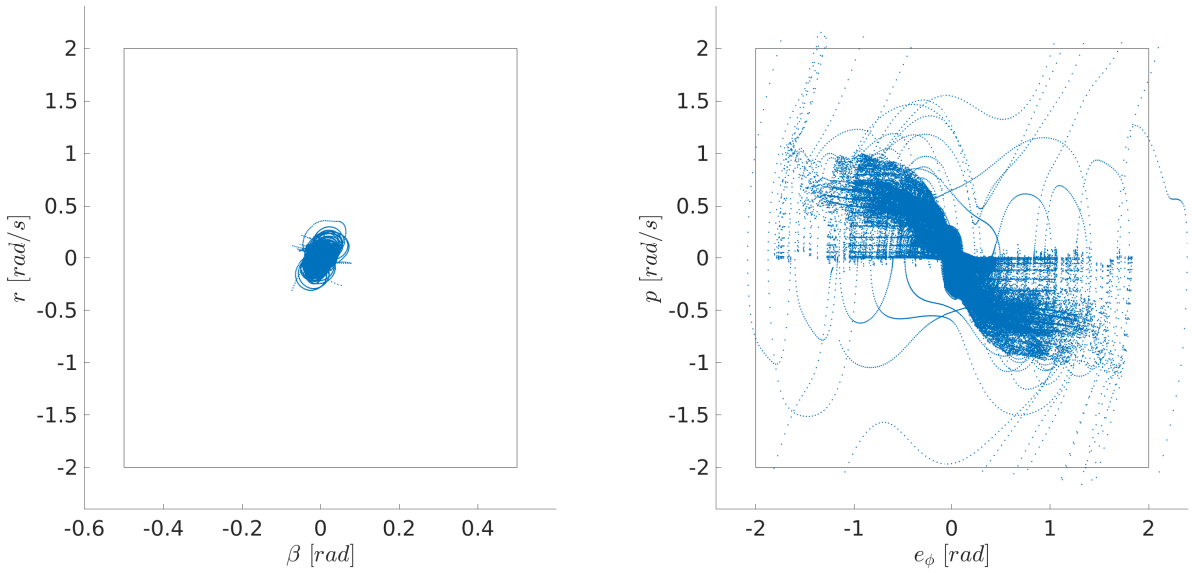
Then the controller was evaluated for robustness against sensor noise and partial failure of the aileron. The sensor noise is simulated by corrupting the rotational rate signals (i.e., roll rate p and yaw rate r) with zero mean Gaussian noise. The partial loss of aileron was simulated by halving the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



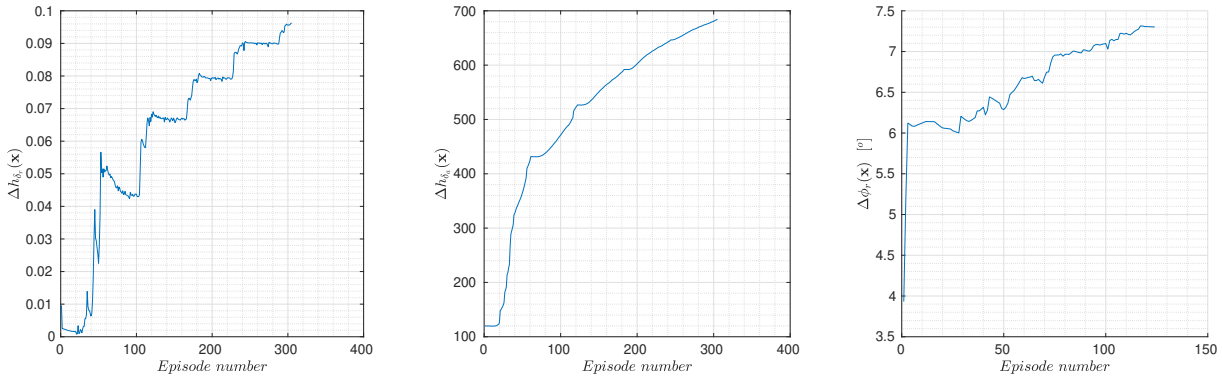
(a) side slip controller

(b) roll angle controller

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.

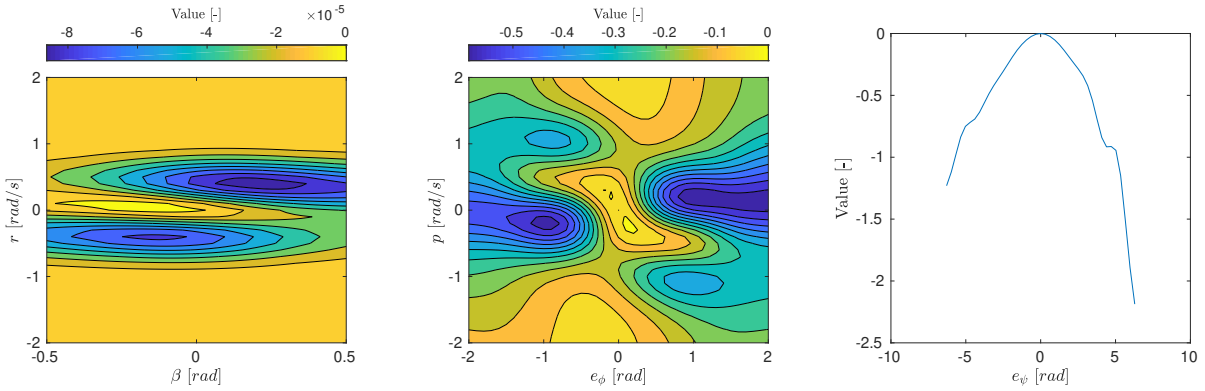
According to Figure 8(c), the heading angle policy changed rapidly during the first episode and afterward there is a slow increase in the change of policy with some fluctuations. Rapid change in the first episode is due to high TD error in the first episode, and small variations after that are due to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



(a) side slip controller (b) roll angle controller (c) heading angle controller

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.



(a) side slip controller (b) roll angle controller (c) heading angle controller

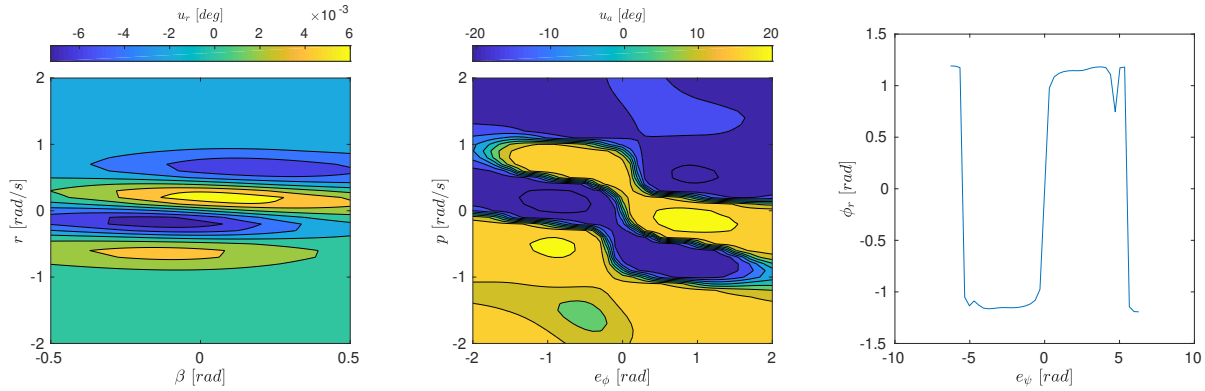
Figure 9. Value functions after training.

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



(a) side slip controller (b) roll angle controller (c) heading angle controller

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 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
	Non-trained J-SNAC	-52.0257
sin wave	Trained J-SNAC	-1.5802
	PID	-0.1839
	Non-trained J-SNAC	-4.8565
smooth-step	Trained J-SNAC	-0.3440
	PID	-0.1623

C. Robustness Against Sensor Noise

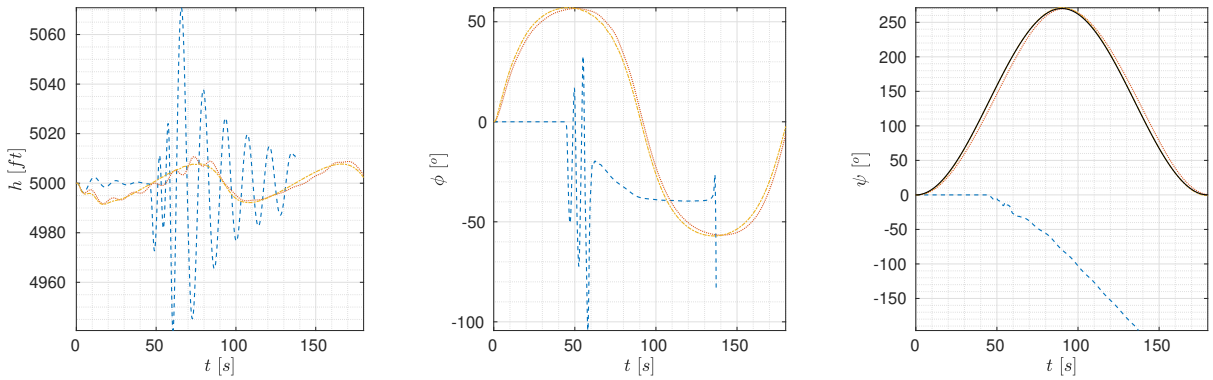
Figure 14 shows the aircraft state evolution while tracking sinusoidal heading commands in the presence of noise in the rate measurements. The sensor noise is simulated by adding zero-mean noise signals with the roll and yaw rate signals. The noise signals have a standard deviation of 5 degrees/s.

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

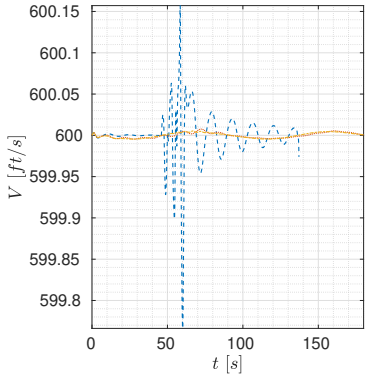
Figure 14 shows the aircraft state evolution while tracking sinusoidal heading commands in the presence of aileron actuator failure.



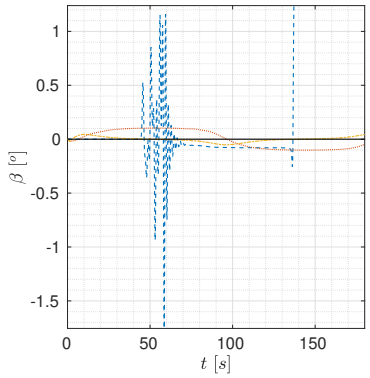
(a) Altitude

(b) Roll Angle

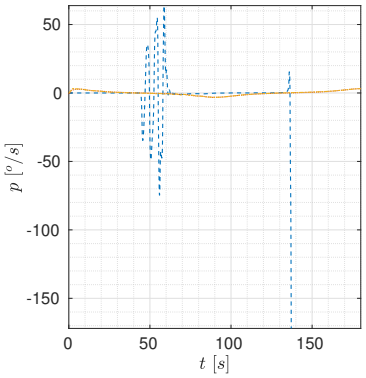
(c) Heading Angle



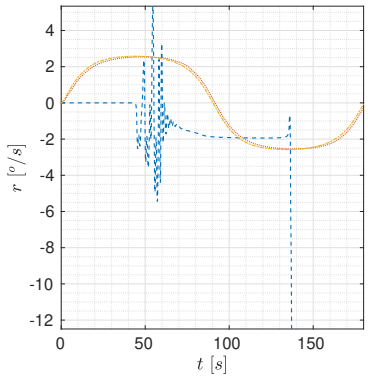
(d) Airspeed



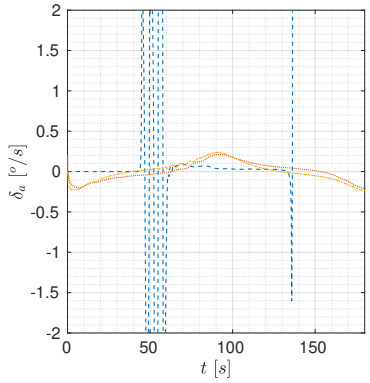
(e) Angle of Side-Slip



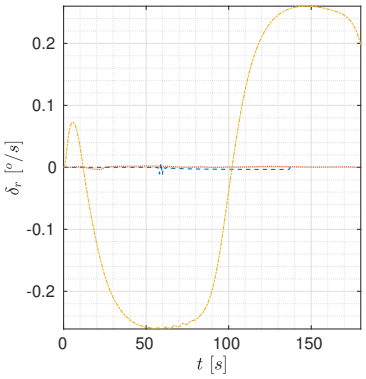
(f) Roll Rate



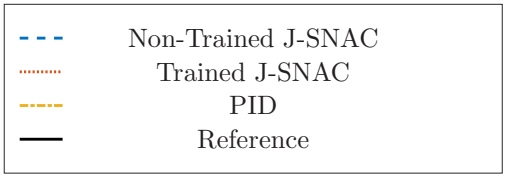
(g) Yaw Rate



(h) Aileron Angle

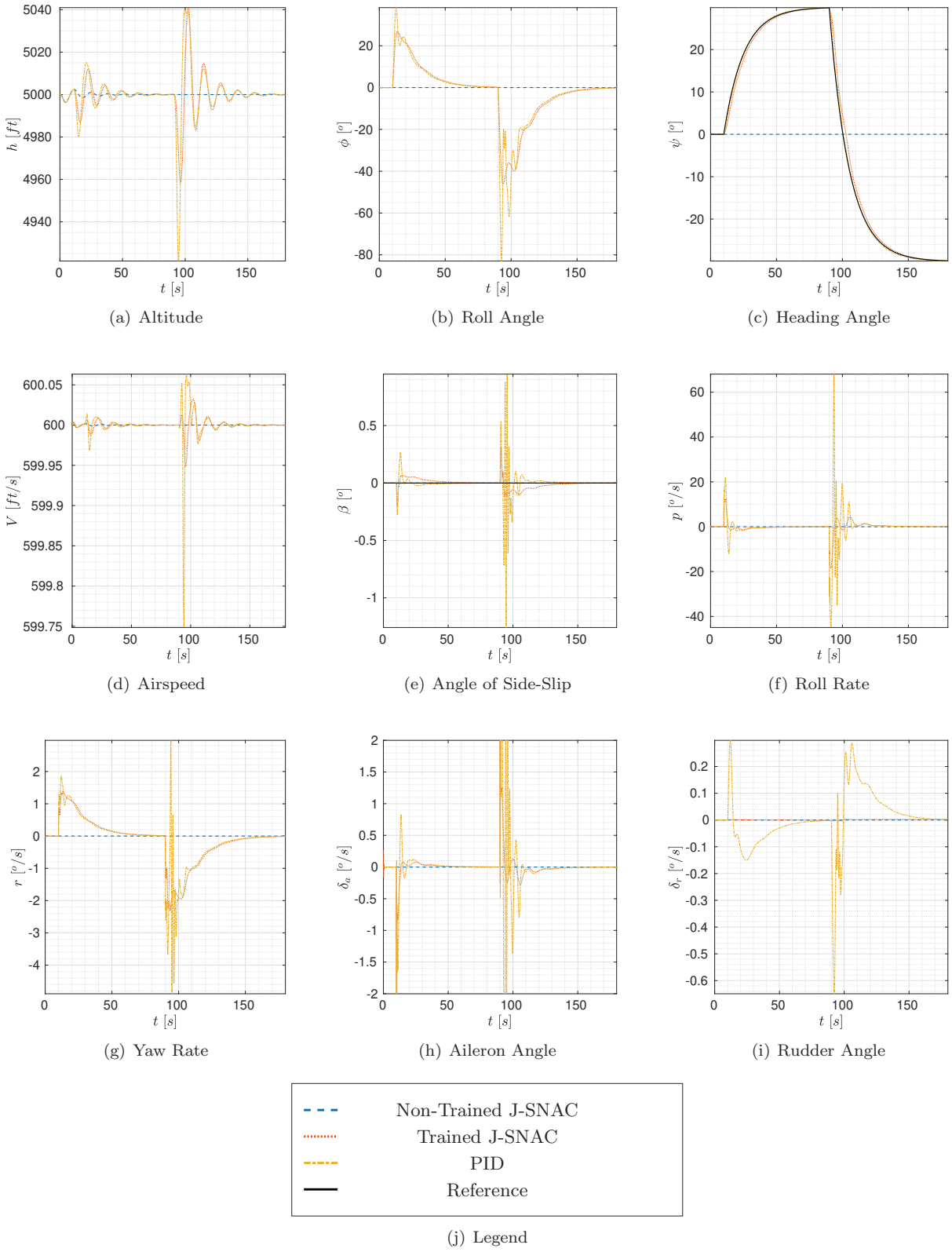


(i) Rudder Angle



(j) Legend

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



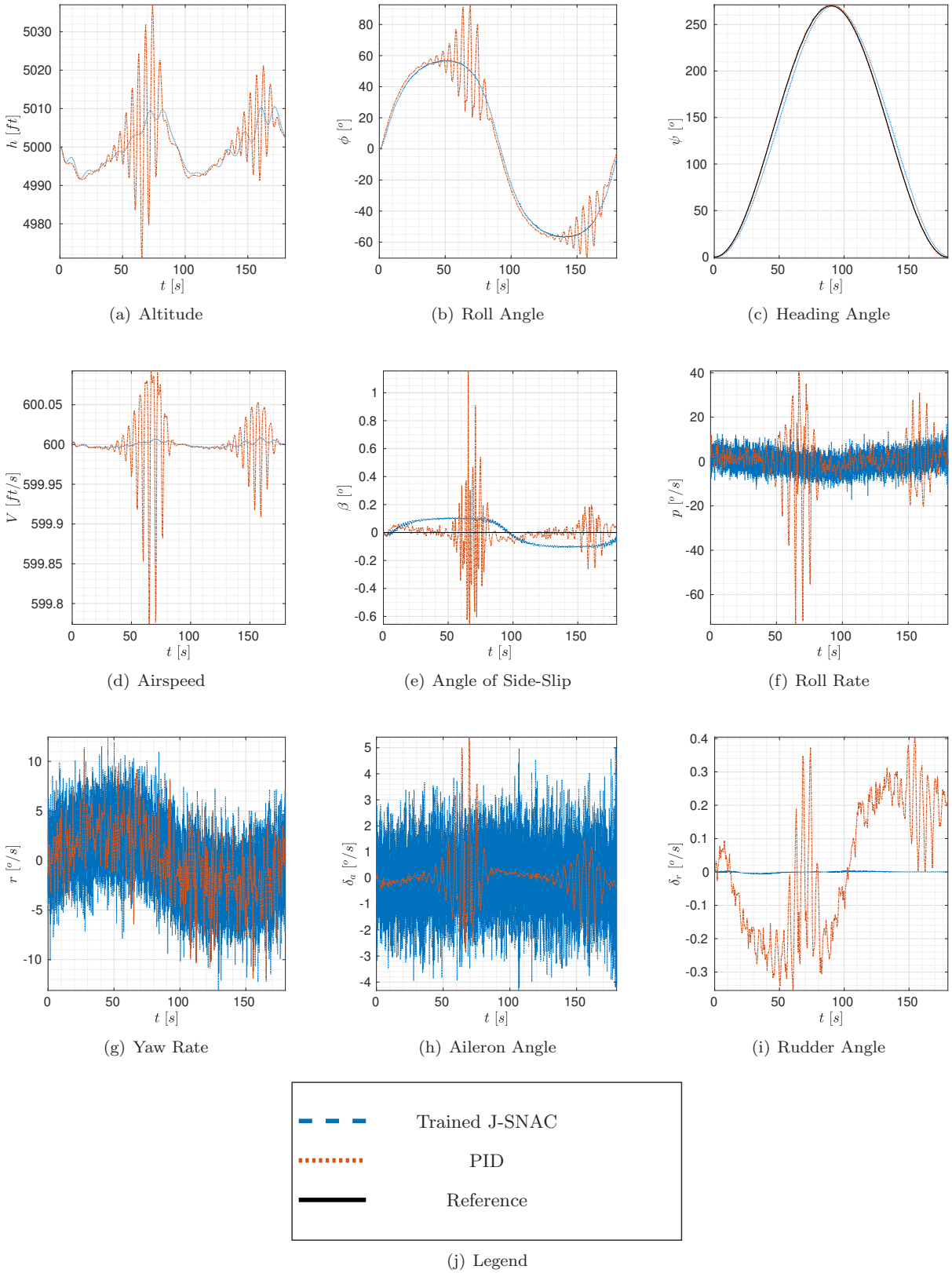
(a) Altitude (b) Roll Angle (c) Heading Angle

(d) Airspeed (e) Angle of Side-Slip (f) Roll Rate

(g) Yaw Rate (h) Aileron Angle (i) Rudder Angle

(j) Legend

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



(a) Altitude (b) Roll Angle (c) Heading Angle

(d) Airspeed (e) Angle of Side-Slip (f) Roll Rate

(g) Yaw Rate (h) Aileron Angle (i) Rudder Angle

(j) Legend

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 flight-envelope protection while learning; investigating the controller performance for other fault scenarios and validating the simulation studies with experimental studies.

References

[1]Belcastro, 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.

[2]Belcastro, 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*.

[3]Safety, B. A., "Statistical Summary of Commercial Jet Aircraft Accidents: Worldwide Operations, 1959-2016," *Boeing Commercial Airplane, Seattle, WA*.

[4]Safety, I., "Safety Report," techreport, International Civil Aviation Organization, Montreal, Canada, 2017.

[5]Werbos, P. J., "Approximate dynamic programming for real-time control and neural modeling," , 1992.

[6]Werbos, P. J., "Neurocontrol and supervised learning: An overview and evaluation," *Handbook of intelligent control*, Vol. 65, 1992, p. 89.

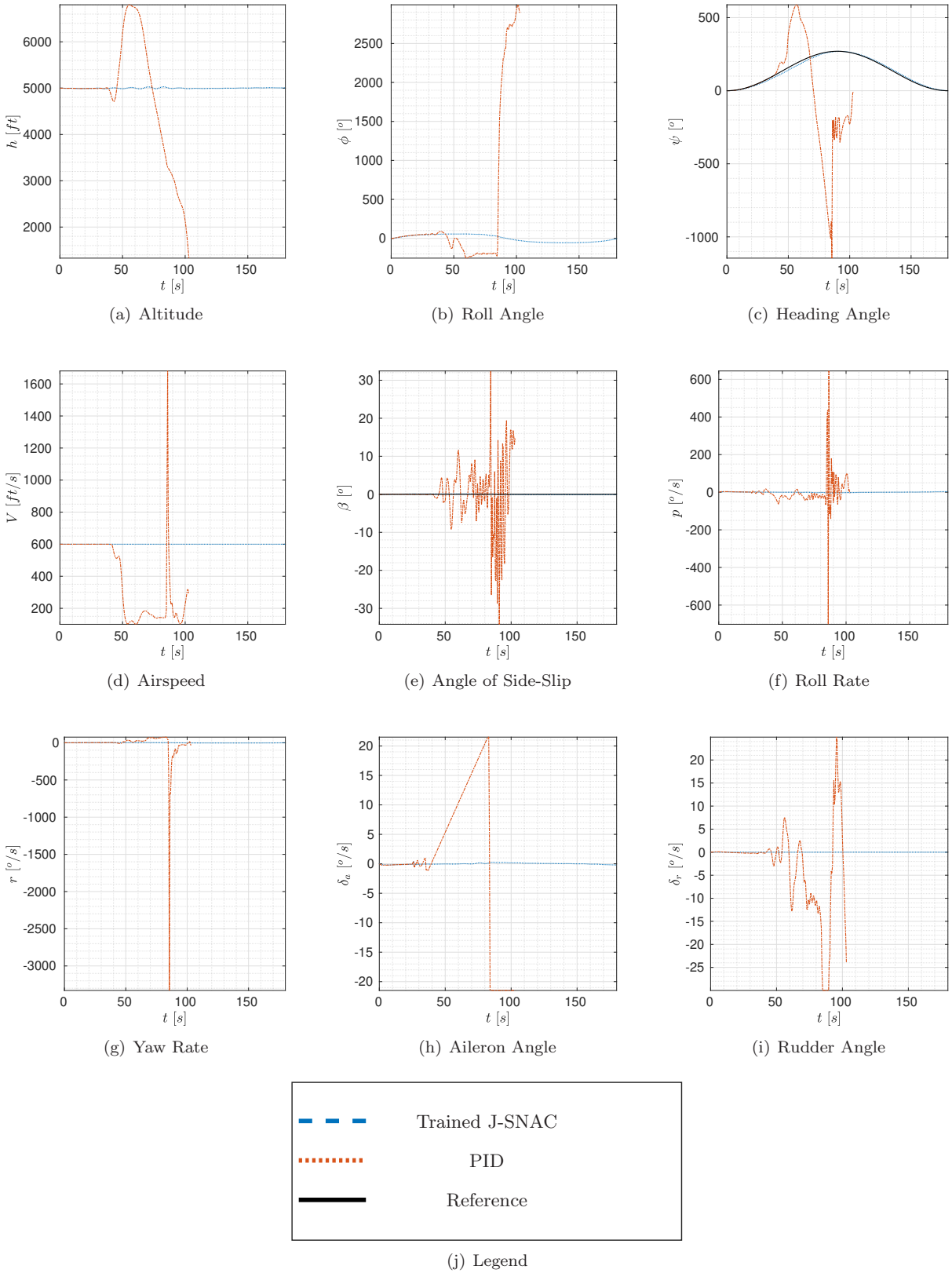
[7]Lewis, F. L. and Vrabie, D., "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, Vol. 9, No. 3.

[8]Si, J., *Handbook of learning and approximate dynamic programming*, Vol. 2, John Wiley & Sons, 2004.

[9]Werbos, 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.

[10]Lewis, 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.

[11]Lewis, F. L. and Liu, D., *Reinforcement learning and approximate dynamic programming for feedback control*, Vol. 17, John Wiley & Sons, 2013.



(a) Altitude (b) Roll Angle (c) Heading Angle

(d) Airspeed (e) Angle of Side-Slip (f) Roll Rate

(g) Yaw Rate (h) Aileron Angle (i) Rudder Angle

(j) Legend

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

[12]Enns, 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.

[13]Han, 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.

[14]Han, 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.

[15]Han, 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.

[16]Han, 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.

[17]Ferrari, 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.

[18]Ferrari, S. and Stengel, R. F., "Online adaptive critic flight control," *Journal of Guidance Control and Dynamics*, Vol. 27, No. 5, 2004, pp. 777–786.

[19]Van Kampen, E., Chu, Q., and Mulder, J., "Online adaptive critic flight control using approximated plant dynamics," in "Machine Learning and Cybernetics, 2006 International Conference on," IEEE, 2006, pp. 256–261.

[20]Nobleheart, 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.

[21]Doya, K., "Reinforcement learning in continuous time and space," *Neural computation*, Vol. 12, No. 1, 2000, pp. 219–245.

[22]Ding, 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.

[23]Padhi, 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.

[24]Ding, 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.

[25]Sutton, 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.

[26]Bellman, R. E., *Dynamic Programming*, Princeton University Press, 1957.

[27]Moody, J. and Darken, C. J., "Fast learning in networks of locally-tuned processing units," *Neural computation*, Vol. 1, No. 2, 1989, pp. 281–294.

[28]Rao, 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.

[29]Zhou, 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.

[30]Zhou, Y., van Kampen, E.-J., and Chu, Q. P., "Launch Vehicle Adaptive Flight Ccontrol Wwith Incremental Model Based Heuristic Dynamic Programming," .

[31]Morelli, E. A. and Klein, V., *Aircraft system identification: Theory and practice*, Sunflyte Enterprises Williamsburg, VA, 2016.

[32]Werbos, P. J., *Reinforcement Learning and Approximate Dynamic Programming (RLADP)-Foundations, Common Misconceptions, and the Challenges Ahead*, John Wiley & Sons, chap. 1, pp. 9–12, 2013.

[33]Ding, 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.

[34]Russell, R. S., "Non-linear F-16 simulation using Simulink and Matlab," *University of Minnesota, Tech. paper*.

[35]Moorhouse, D. and Woodcock, R., "Us military specification mil–f–8785c," Tech. rep., US Department of Defense Arlington County, 1980.

Part II

Preliminary Research

Chapter 2

Background Study

The main goal of this chapter is to situate the current research in the field of aircraft guidance, control, and navigation. It presents an overview of LOC-I and Fault Tolerant Flight Control (FTFC). Furthermore it gives the justifications for using ACD in future FTFC.

Section 2-1 defines the LOC-I and lists some potential technologies for its mitigation. Next, section 2-2 defines FTFC and gives a short summary of present and future FTFCs. Then in section 2-3, various approaches for developing FTFC is presented. Finally, section 2-4 justifies for the use of ACD in future FTFCs.

2-1 Loss of Control- In flight (LOC-I)

LOC-I is one of the three high-risk aircraft accident categories [30], that causes the most onboard fatalities [1, 2, 30]. Unlike the other high-risk accident categories, the number of LOC-I occurrences has not decreased in recent years [1, 2]. LOC-I accounted for 42.9 % of all fatal accidents in the year 2016 [2].

LOC-I is a complex problem, caused by a plethora of causes that either act individually or in combination [31]. As a result, there is no standard method for preventing or recovering from LOC-I events. However, a collection of new technologies are in the research and development phase that has the potential to accomplish this goal [31]. According to [31], below are few of the important technologies that are currently being considered.

- Advanced mathematical models for characterizing LOC-I conditions and their effects on aircraft dynamics and control characteristics.
- Onboard systems technologies that can detect the effects of LOC-I hazards on vehicle dynamics and control.
- Onboard systems technologies that can assess and predict the flight safety.
- Onboard systems technologies that can mitigate the effects of LOC-I hazards.

- Onboard systems technologies that can provide guidance or automatic upset recovery.
- Onboard systems technology that enables improved situational awareness and decision support to the flight crew.
- Onboard systems technology that provides enhanced dynamics and control capabilities under LOC-I precursor conditions.
- Advanced simulation model and technologies to simulate an impaired vehicle in real time for training flight-crew under LOC-I hazards.
- Advance tools to perform validation and verification (V&V) of the technologies mentioned above to aid their certification.

2-2 Fault Tolerant Flight Control System (FTFC)

FTFCs are the onboard systems that can mitigate the LOC-I hazards by providing enhanced dynamics and control capabilities under the LOC-I precursor conditions. The current generation of FTFCs employs redundancies in hardware, distributed systems, and other controls and sensors to handle faults in critical components [3]. These FTFCs accommodate a specific class of faults named as *additive faults*.

However, recent aircraft accident analyses show that parametric failures cause the most number of LOC-I [3]. Parametric failures can be accommodated with advanced control methods that have improved robustness properties and real-time reconfiguring or adaptive capabilities [3]. Although several of such methods already exist (see [6, 7, 10–12]), lack of experimental evidence have deemed these methods unreliable [9]. Simulation and scaled model experimental studies are required for maturing these technologies and introduce them in the future aircraft.

2-3 Classification of FTFC Design Methods

While there are a plethora of techniques that can be used in reconfigurable FTFCs (see [6, 7, 10–12]), many new methods are being developed at the moment. The main drive for such development is to improve on previous systems regarding efficiency, performance and design simplicity. Figure 2-1 shows the relevant and currently considered methods for FTFC design. This illustrated classification is neither unique nor exhaustive. Literature such as [3, 6, 7, 10–12, 32] can provide a detailed perspective on these methods.

As in the Figure 2-1, FTFCs can be classified into two categories: Active FTFC (AFTFC) and Passive FTFC (PFTFC). PFTFC are designed with robust control methods and are suitable for addressing the failures that can be modeled as uncertainty around a nominal model. Whereas AFTFC design techniques employs either a *on-line redesign* method or a *projection based method*. The main differences between PFTFC and AFTFC are:

- AFTFC methods account fault informations explicitly and do not assume a static nominal model.

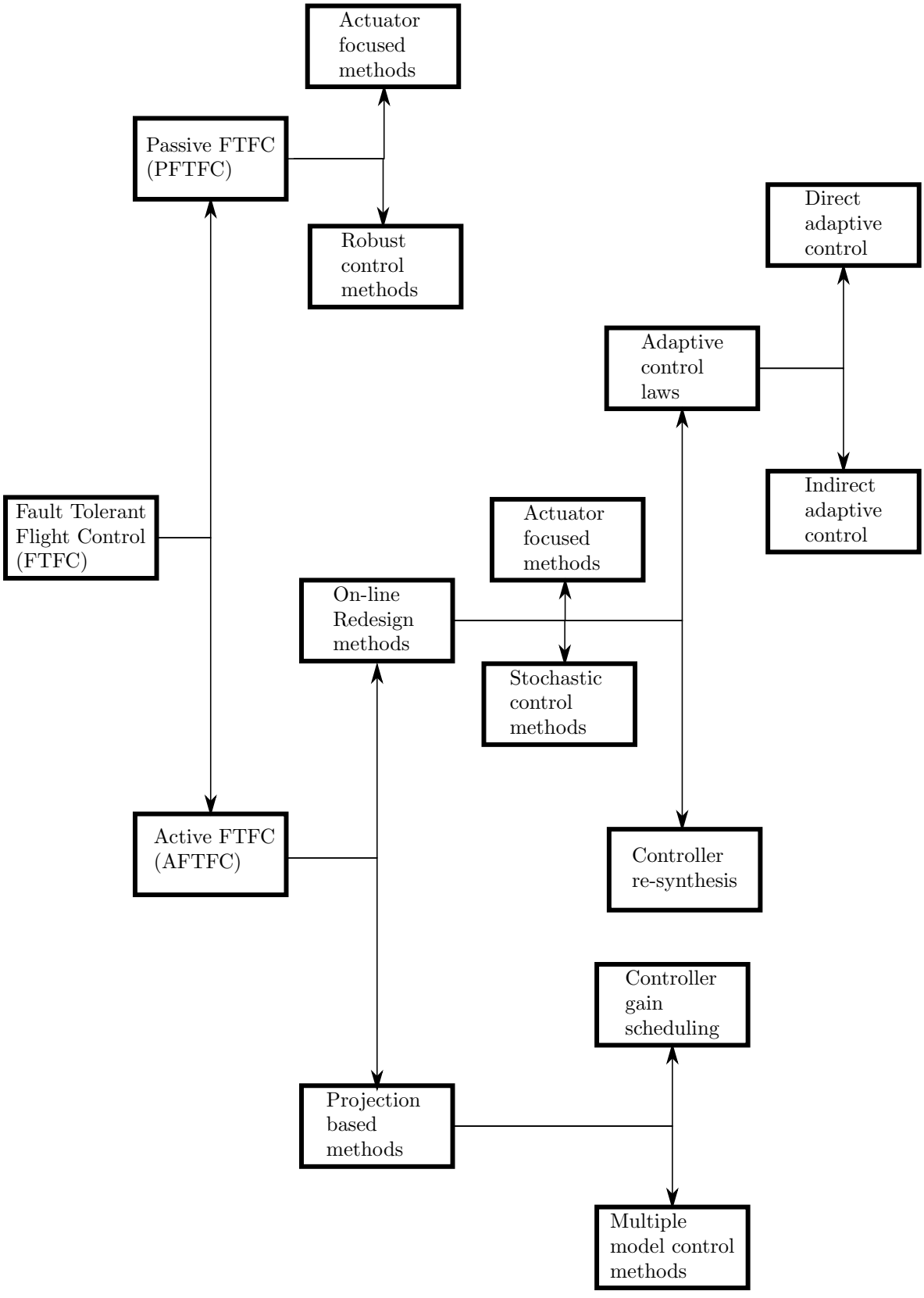


Figure 2-1: Classification of the state-of-the-art FTFC design methods

- Unlike PFTFC, AFTFC generally requires other supporting systems such as FDI/FDD and RM [11].

The controllers designed with PFTFC methods can provide satisfactory stability and performance guarantees for the failures that do not make the closed-loop system cross the controller stability boundaries [11]. In [10], the authors presented few methods used for the design of PFTFC systems. Two advantages of using PFTFC are: 1) they are easy to develop and implement, 2) they do not require any fault isolation and identification schemes. The main disadvantage of using PFTFC techniques is that they are only suitable for a certain class of failure modes; the vast majority of the faults cannot be modeled as uncertainty around a nominal model. [11]

The AFTFC systems designed with projection based methods perform its control reconfiguration tasks by either selecting one or mixing multiple controllers from a set of pre-designed controllers. Ordinarily, each controller in the set is designed for a particular fault and is selected by the Control Reconfiguration Mechanism (RM) when the Fault Detection and Identification (FDD) system has identified the designated fault. Similar to PFTFC, these controllers can deal with a finite and known number of faults [11].

The AFTFC systems designed with on-line redesign methods reconfigure their control laws by either recomputing the controller parameters or recalculating the structure and the parameters of the controller. Online redesign methods are more computationally expensive; as they often turn out to be on-line optimization procedures. Additionally, similar to projection based methods most of the online redesign methods require RM and FDD/Fault Detection and Isolation (FDI) systems. The attainable post-fault system performance of the on-line redesign methods surpasses that of PFTFC and projection-based AFTFC systems[11]. Some of the methods for designing AFTFC can be found in [10–12].

2-4 Use of Adaptive Critic Design (ACD) in FTFC design

Considering the goal of creating a general-purpose FTFC that can accommodate parametric failures attain better post-fault performances, it would be wise to mature one of the on-line redesign methods. Specifically, one of the direct adaptive control design methods, because these types of controllers have the inherent ability to adapt to changes in the system parameters while remaining free of the issues faced by indirect methods [11]. Many methods can be used to design direct adaptive controllers. Some of these methods are model reference adaptive control, self-tuning control, adaptive neuro controller, adaptive critic designs, L_1 adaptive controller, adaptive back-stepping and adaptive nonlinear dynamic inversion [3, 11].

Although all of the techniques mentioned above can be used to design direct-adaptive controllers, there is a lack of trust in ACD based FTFCs in the FCS design community. The main reason for this is the limited information on the performances of these controllers. ACD methods pose severe computational load which was not previously matched by flight computers' capability [3]. However, the current state of computing technology allows reconsidering this view.

Also considering that when an aircraft incur unanticipated failures, the parametric properties of the aircraft moves far away from nominal set points. The goal of a reconfigurable FTFC

is then to find the best control strategy rapidly and make the aircraft stable and controllable again. This reconfiguration is a dynamic optimization problem, characterized by stochastic and nonlinear aspects. Dynamic programming proposed by Bellman [33] is the only exact method for solving stochastic dynamic optimization problems [17]. However, dynamic programming cannot be applied online. ACD solve dynamic optimization problems efficiently by computing an approximate solution to the problem in real and forward in time [17].

Furthermore, ACD methods are based on MIMO, nonlinear and optimal control design principles. These principles as a combination make ACD an elegant method from the design point of view. These principles allow having lesser assumptions and approximations during the design phase. They enable to consider more realistic cases of nonlinearities. Optimal nature of these type of control laws also allows incorporating the model uncertainties (parameter variation, external disturbances, un-modeled dynamics) and the optimality requirements. Thus such control laws also allow meeting the limits of states, inputs, and outputs.

2-5 Conclusion

This chapter presented a summary of the literature that aided to frame the current research. The initial research entailed an exploration of state-of-art literature to identify and situate the current work in the field of aircraft guidance, control, and navigation. From the background study, it was found that there is a strong desire for advanced FTFC systems in the aviation community. Moreover, it was found that ACD has the potential to improve on existing FTFC by simplifying the design procedure and accommodate unanticipated faults. However, in order to apply ACD in FCS, these algorithms are required to be evaluated.

Chapter 3

Review on Adaptive Critic Designs

In the previous chapter, it was motivated that ACDs have the potential to overcome limitations of current generation FTFCs while simplifying the overall design effort. This chapter presents a review of this class of algorithms and their implementation on FCS.

In the following section, the preliminary information concerning ACDs is presented. Next in Section 3-2, the notable ACD architectures are presented. Then in Section 3-3, existing applications of ACD in FCS are given. Finally the chapter concludes in Section 3-4, where the notable ACD architectures are contrasted with existing FCS applications .

3-1 Preliminaries

ACDs are a class of algorithms that combine the concept of Dynamic Programming (DP), Temporal Difference (TD) learning and function approximation to find approximate solution to large-scale Markov Decision Processes (MDPs). Before delving into ACDs, these preliminaries are elucidated in this section. The source materials for these preliminary information include [34–40].

3-1-1 Markov Decision Processes (MDP)

ACD concerns with sequential decision-making problems. Such problems are generally formalized with Markov Decision Process (MDP). Figure 3-1 depicts a control theoretic perspective on MDP. In this setting, there is a controller, a dynamic plant, and a reward function. At a given time instant t_k, the controller observes the state of the plant $x[t_k]$ and applies an action $u[t_k]$. The action changes the state of the plant to $x[t_{k+1}]$ and in response to this change, the reward function rewards the controller with a scalar reward $r[t_{k+1}]$. The goal of the controller is to continue this process of state observation, action application and reward reception for a duration of time; and uses its observations (i.e., state transitions and rewards) to learn a control policy that allows it to accrue maximum possible rewards.

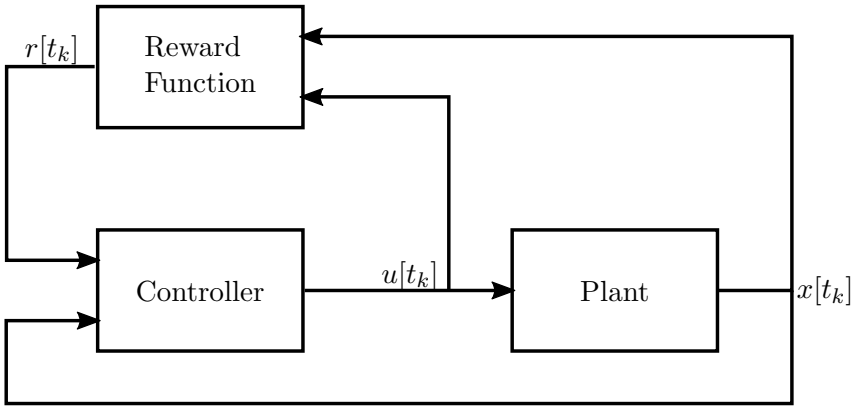


Figure 3-1: A control theoretic perspective on Markov Decision Process

Formal Framework of MDP

Formally, an MDP is modeled with the tuple $\mathbb{M} = \{X, U, f, \rho\}$. In this tuple,

- X is the set of Markovian states.
- U is the set of control actions.
- f is the mapping $f : X \times U \times X \rightarrow [0, 1]$, describing the one-step state transitions as conditional probabilities $f(x, u, x') = Pr(x'|x, u)$ of moving to state x', given that controller applied action u when it observed the state x.
- ρ is the reward function $\rho : X \times U \times X \rightarrow \mathbb{R}$, giving the expected reward $\rho(x, u, x') = \mathbb{E}(\mathbb{R}|x, u, x')$ for moving to state x', given that controller applied action u when it observed the state x.

A state is called "Markovian" if the transition probabilities f and the reward function ρ depend only on the current state x and not on the past state trajectory of the MDP.

The transition function f and reward function ρ together are called the model of MDP.

Policies of MDP

A policy h is a rule of behavior that the controller follows to interact with the plant under all circumstances. A policy is the mapping $h : X \times U \rightarrow [0, 1]$, describing the probabilities $h(x, u) = Pr(u|x)$ of taking action u when the controller observes the state x. This probabilistic distribution makes the policies of MDP stochastic. However, there are also deterministic policies for MDP. A policy is called deterministic if only one action is mapped to each state; a deterministic policy is generally expressed as a direct mapping of the states to action, i.e., $h : X \rightarrow U$. Furthermore, the policy is called stationary if $h(x, u)$ is independent of the time and otherwise it is called non-stationary.

Optimal Control of MDP

The goal of the controller is to find a stationary-policy that allows it to maximize a measure of *rewards to be accrued*. This measure is called *the return* value in the literature. The maximization of the return value makes this type of problems an optimal control problem. From the perspective of control-system synthesis, the return value abstracts the overall control objective. Although several definitions of the return exist, most works in the literature advocate the use of the discounted-infinite-horizon formulation. This formulation is used because most convergence analysis and stability proofs are done for this class of objective functions.

$$R[t_k] = \sum_{i=k}^{\infty} \gamma^{i-k} r[t_{i+1}] \tag{3-1}$$

Equation 3-1, shows the definition of discounted-infinite-horizon return. There, $r[t_i]$ is the reward received at time instant t_i and $\gamma \in [0, 1]$ is the discount factor. The discount factor determines the length of the time horizon that the controller considers for optimizing its actions. If the desire is to optimize the controller's actions for a near future event, γ is set closer towards 0. On the contrary cases, γ is set closer towards 1.

Value Functions and Bellman Equations

The suitability of a given policy h for an MDP can be determined by finding the state-values of the policy. The state-value of policy h for a given state $x \in X$ is defined as the expected return to be received by the controller, given that it starts to execute the policy from the state x. State values of policy for all states of the plant are generally stored in a functional or tabular form. This stored form of values is called the state-value function. State value function is denoted with $V^h(x)$ in this work.

$$V^h(x) = \mathbb{E}^h \left\{ R[t_k] \middle| x[t_k] = x \right\} = \mathbb{E}^h \left\{ \sum_{k=0}^{\infty} \gamma^k r[t_{k+1}] \middle| x[t_k] = x \right\} \tag{3-2}$$

Alternatively, the value of a policy can also be determined by finding its action values. Action values of a policy h for a given state action pair (x, u) is defined as the expected return to be received by the controller, given that the controller takes action u in the state x and follows the policy h after that. Similar to state values, action values are also stored in a functional or tabular form. This stored form of action values is called action-value function, which is denoted with $Q^h(x, u)$ in this work.

$$Q^h(x, u) = \mathbb{E}^h \left\{ R[t] \middle| x[t_k] = x, u[t_k] = u \right\} = \mathbb{E}^h \left\{ \sum_{k=0}^{\infty} \gamma^k r[t_{k+1}] \middle| x[t_k] = x, u[t_k] = u \right\} \tag{3-3}$$

The definitions of the value functions can be restructured to find a recursive nature within them. This recursive relation is presented in Equation 3-4 and 3-5. These equations are referred to as Bellman Equations.

$$V^h(x) = \sum_u h(x,u) \sum_{x'} f(x,u,x') \left[\rho(x,u,x') + \gamma V^h(x')\right] \tag{3-4}$$

$$Q^h(x,u) = \rho(x,u,x') + \gamma \sum_u h(x,u) Q^h(x',u) \tag{3-5}$$

The state value function and the action value function for a given policy are inter-related. This corollary comes from the definitions of the value functions. For a given policy h, determining any one of the value functions is sufficient to characterize the policy. The availability of the MDP model dictates the choice of the value function. If the model is available, the state value function is preferable. In the absence of the model of the MDP, action value function is preferred as it implicitly stores the MDP model.

Optimal Value Functions and Optimal Policies

Optimal value function, $V^*(x)$ (or $Q^*(x,u)$), is defined as the mapping of the states (or state-action pairs) of the MDP to the maximum possible return that can be obtained from any policy. The definitions of the optimal value functions are given in equations 3-6 and 3-7. Since optimal value functions are unique to a given problem they can simplified, as in Equations 3-8 and 3-9. These simplified equations are called Bellman optimality Equations.

$$V^*(x) \equiv \max_h V^h = \max_h \left\{ \sum_u h(x,u) \sum_{x'} f(x,u,x') \left[\rho(x,u,x') + \gamma V^h(x')\right] \right\} \tag{3-6}$$

$$Q^*(x,u) \equiv \max_h Q^h = \max_h \left\{ \rho(x,u,x') + \gamma \sum_u h(x,u) Q^h(x',u) \right\} \tag{3-7}$$

$$V^*(x) = \max_u \left\{ \sum_{x'} f(x,u,x') \left[\rho(x,u,x') + \gamma V^*(x')\right] \right\} \tag{3-8}$$

$$Q^*(x,u) = \rho(x,u,x') + \gamma \max_{u'} Q^*(x',u') \tag{3-9}$$

Optimal policies, denoted with $h^*(x)$, are the policies that correspond to the optimal value functions. Moreover, by definition, optimal policies allow accruing maximum possible reward from any given state.

3-1-2 Dynamic Programming (DP)

Dynamic Programming is a collection of model-based methods for solving MDPs. The main idea in DP is to use the model of the MDP to find the optimal value function and then synthesize the optimal policy from the computed optimal value function. Although the processes of DP are intuitive, their applications are limited to problems with a finite and small number of states and actions. This is because of the following three reasons:

- DP requires a stationary model of the MDP. For large and complex system it may not be possible to obtain this model.
- DP is computationally intractable for large state-action space systems as it stores values for each of the states (and actions).
- DP is a backward search technique and thus precludes its use in real-time control applications.

Although ACD and other approximate dynamic programming techniques are developed to overcome the limitations of DP, their working principles are closely related to that of DP. DP techniques consist of two computational processes, *policy evaluation* and *policy improvement*. These processes are executed iteratively to find optimal value function and optimal policy. Below are an account on these processes and few relevant DP techniques.

Policy Evaluation

Policy evaluation is the process of computing a value function corresponding to a given policy $h(x,u)$. Value functions of a given policy are found by solving Bellman Equations (see Equation 3-4 and 3-5). The main idea is to assume an initial value function for a given policy and then use the appropriate Bellman Equation to iterate over the assumed value function until convergence.

Policy Improvement

Value functions define a partial ordering over the set of all possible policies. A policy h' is said to be equal or better than the policy h, if, for example, the value function $V^{h'}(x)$ is greater than or equal to the value function $V^h(x)$. This partial ordering of policies by a given value function can be utilized to obtain a better or equal policy. The process of obtaining improved policy from a given value function, $V^h(x)$ or $Q^h(x,u)$, is called *policy improvement*. This improved policy is often referred to as *greedy policy*. Equations for finding the improved policies from given value functions are presented in Equations 3-10 and 3-11.

$$h'(x,u) = \underset{u}{\operatorname{argmax}} \left\{ \sum_{x}^{'} f(x,u,x') \left[\rho(x,u,x') + \gamma V^h(x) \right] \right\} \tag{3-10}$$

$$h'(x,u) = \underset{u}{\operatorname{argmax}}\, Q^h(x,u) \tag{3-11}$$

Policy Iteration Algorithm

In the policy iteration algorithm, policy evaluation and policy improvements steps are executed in sequential manner. This procedure is repeated until both the value function and the policy function becomes stationary. These stationary functions are the optimal policy and value functions.

Value Iteration Algorithm

The main idea of value iteration algorithm is to assume a value function and iteratively execute the Bellman optimality equation (Equation 3-8 or 3-9) until the convergence of value function. The converged value function is the optimal value function. Once this function is found, optimal policy can be computed using appropriate policy improvement Equations (i.e. Eq. 3-10 or 3-11).

The use of the Bellman optimality equation can be seen as executing one iteration of policy evaluation and immediately executing the policy improvement process.

Asynchronous Policy/Value Iteration Algorithm

The standard policy iteration and value iteration requires performing the value evaluation for all states of the MDP in one iteration of policy evaluation process. In asynchronous methods, the idea is to perform policy evaluation and policy improvement in a sub-space of the state-action-space. This is useful when state space is large, and a limited model or data is available.

Generalized Policy Iteration Algorithm

Policy iteration was introduced as performing policy evaluation until the convergence of the value function and value iteration was introduced as performing only one step of policy evaluation. GPI presents the spectrum in-between value iteration and policy iteration. The main idea is to stop the evaluation step earlier for the sake of convergence of the policy by compromising the accuracy of the converged policy as the optimal policy.

GPI represents the idea of interleaving of the policy evaluation and policy improvement steps at any time, without completing the policy evaluation and policy improvement steps. This gives further granularity over the DP techniques, which can be harnessed in application-specific tasks.

3-1-3 Temporal Difference (TD) Learning

TD learning refers to a class of model-free methods for solving MDP. The main idea of TD learning is to estimate the optimal value function for a given MDP and adapt this estimation with samples from the state and reward trajectories. The use of samples removes the requirement for the model of the MDP to find the optimal policy.

The main benefits of using TD in comparison to DP are:

- The model of MDP is not required.
- TD can be used to improve the estimation of value function online and thus usable in real-time control.
- It is an incremental learning scheme and thus allows to update value function locally and before the end of a learning cycle.

TD(0)

The basic form of TD learning algorithm is called TD(0). The equation for updating state value function with this algorithm is given in Equation 3-12. In the equation r_k is the current reward received by the controller; $V_{k+1}(x)$ and $V_k(x)$ stands for the new and the current estimate of the value of the state x; $V_k(x')$ stands for the current estimate of value of the state x' that follows from the state x; and $\alpha \in [0, 1]$ is the learning rate that determines increment to the new estimate.

$$V_{k+1}(x) := V_k(x) + \alpha(r_k + \gamma V_k(x') - V_k(x)) \tag{3-12}$$

TD(λ)

One limitation of TD(0) is that it only updates the value function for one state at a time. This makes this scheme a slow learning process. This limitation can be overcome by the use of the so-called eligibility trace. The main idea is to assign trace parameters to the states of the MDP and exponentially decay this parameter according to the recency of visits. Every time a state is visited, the respective parameter is incremented and subsequently decayed exponentially with time. When a new reward is received, the credit of this reward is shared with recently visited states by updating the value function for states according to their eligibility trace.

The TD learning scheme that uses this trace is called TD(λ). In this scheme the value function gets updated with some form of Equation 3-13. In this equation, α is still the learning rate; δ is the temporal difference (see Equation 3-14); and $e_k(x)$ is the vector of eligibility traces.

$$V_{k+1}(x) := V_k(x) + \alpha\delta_k e_k(x) \tag{3-13}$$

$$\delta_k = r_k + \gamma V_k(x') - V_k(x) \tag{3-14}$$

The update of eligibility traces depends on the parameter $\lambda \in [0, 1]$. These traces can be updated in a variety of ways; the most used scheme in the literature is called the *replacing traces*. In this scheme, $e_k(x)$ is updated with Equation 3-15.

$$e_k(x) = \begin{cases} \gamma\lambda e_{k-1}(x), & \text{if } x \neq x_k \\ 1, & \text{if } x = x_k \end{cases} \tag{3-15}$$

Need for Exploration

One advantageous aspect of TD methods is that they do not require any model. However, to guarantee that TD methods would find the optimal policy or at least get close to it, exploration of the state-action space becomes an imperative. State-action space exploration is incorporated by adopting a stochastic policy. Stochasticity can be brought in many ways; for continuous time systems, this is brought by adding a noise signal to the computed action signal.

3-1-4 Function Approximation

Function approximation is the data-driven approach to approximate the underlying relationship between a given set of input and output signals. A function can be approximated with a parameterized functionals such polynomial functions or with non-parameterized functionals such as decision tree. In ACD, function approximation is used to approximate the value function, policy function and sometimes the model of the system that the controller is trying to control.

ACDs, are a type of GPI algorithms. Therefore these methods improve the approximation of the value and policy function iteratively. These improvements of approximations are often made with gradient-based algorithms. As a result, ACD poses the need for a function approximator that is differentiable with respect to their parameters and the states. In the literature, ACDs are often deployed with Artificial Neural Network (ANN). This choice of function approximation is rationalized by the fact that ANNs are universal function approximators, and are infinitely differentiable.

3-2 Adaptive Critic Designs

ACD are a class of algorithms that attempts to circumvent "the curse of dimensionality" in DP by *approximating its solution in the most general case* [41]. The utility of this approximation is that complex optimization over time problems can be addressed in a tractable manner.

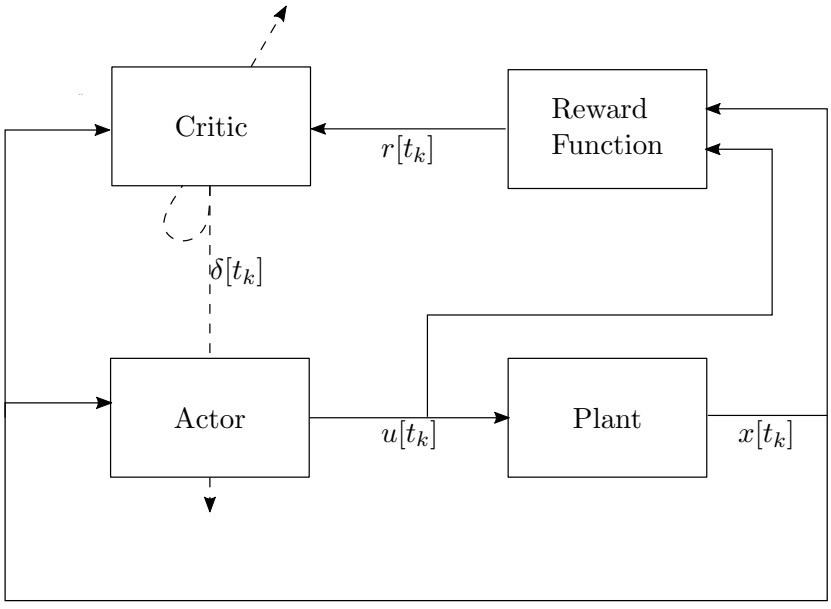


Figure 3-2: A pictorial depiction of working mechanism of ACD algorithms

Figure 3-2 presents an overview of the ACD algorithms. ACDs utilizes function approximation and temporal difference learning to approximate DP. ACDs consist of two "entities" called the *actor* and the *critic*. The actor uses a parametric function to approximate the optimal policy function (h^*). Moreover, the critic uses a parametric function to approximate the optimal

value function (V^* or Q^*). The critic improves its approximation of the optimal value function using a suitable TD learning method. Simultaneously the actor improves its approximation of the optimal policy by tuning its parameters to maximize the expected return according to the newly adapted value function. This process of adaptation is a GPI scheme, which overtime contract the estimated value and policy functions to their (near) optimal forms. The use of TD learning enables ACDs to learn to make better decisions over time. The use of function approximator reduces the number of parameters to be updated in policy evaluation and policy improvement steps drastically and aids to generalize the control policy for similar states.

ACD encompasses a plethora of architectures. Few studies, namely [15, 42–46], surveys a great portion of these algorithms and their recent advancements. Among the existing ACD algorithms, four are the fundamental as all others are either extended or modified versions of these four. The theories and design procedures for various ACD are found in [14, 15, 36, 38, 43, 47–49] and the references therein. Below is an account of the four basic ACD algorithms and how they are extended or modified to build other ACD algorithms.

3-2-1 Fundamental ACD Architectures

There are four basic ACD algorithms with improving performance and complexity. These structures are derived for discrete-time systems and are called Heuristic Dynamic Programming (HDP), Dual Heuristic Programming (DHP), Action Dependent (AD)HDP and ADDHP. Following aspects can distinguish these architectures,

- **Critic Output:** The critics are designed output either the approximated state values ($V(\mathbf{x})$) or action values $Q(\mathbf{x}, \mathbf{u})$) or the derivatives of the value functions ($\partial V(\mathbf{x})/\partial \mathbf{x}$ or $[\partial Q(\mathbf{x})/\partial \mathbf{x}\ \partial Q(\mathbf{x})/\partial \mathbf{u}]^T$).
- **Critic Input:** When the critic is designed to approximate the state value function or its derivatives, the critic takes the state measurements as its inputs. On the other hand, when the critic is designed to approximate the action value function or its derivatives, the critic takes in both the state and action measurements.
- **Requirement of Plant Model Derivatives:** ACDs adapt the parameters of the critic and actor with gradient-based methods. Depending on the architectures, plant model derivatives are required to update either the actor or the critic or both of them.

Heuristic Dynamic Programming (HDP)

In HDP, the critic approximate the optimal state value with a function approximator. Because of this choice of value function, the HDP critic takes the state measurements as its input. The critic's approximation is improved incrementally by minimizing following temporal difference error,

$$E_c = \frac{1}{2}||e_c||^2 = \frac{1}{2}\sum_k e_c^2[t_k] = \frac{1}{2}\sum_k \left[\hat{V}[t_k] - r[t_k] - \gamma \hat{V}[t_{k+1}]\right]^2 \tag{3-16}$$

Where, $\hat{V}[t_k] = \hat{V}(\mathbf{x}[t_k], W_c)$ with W_c being the critic parameters. This architecture requires a plant model to train the actor. This is because the actor parameter update requires an estimation of $\partial V(\mathbf{x}[t_{k+1}])/\partial \mathbf{u}[t_k]$. Since there is no direct link between the critic and the actor, this estimation is done by back-propagating the signal $\partial V(\mathbf{x})/\partial \mathbf{x}$ through the plant model. The plant model does not have to be an accurate one, when there is a mismatch between actual plant and model output HDP uses an online system identification procedure to adapt the model.

Dual Heuristic Programming (DHP)

In this structure, critic directly estimates the derivative of the state value function $\partial V/\partial \mathbf{x}$. The identity for this derivative is given by,

$$\frac{\partial V[t_k]}{\partial \mathbf{x}[t_k]} = \frac{\partial r[t_k]}{\partial \mathbf{x}[t_k]} + \frac{\partial r[t_k]}{\partial \mathbf{u}[t_k]}\frac{\partial \mathbf{u}[t_k]}{\partial \mathbf{x}[t_k]} + \frac{\partial V[t_{k+1}]}{\partial \mathbf{x}[t_{k+1}]}\left[\frac{\partial \mathbf{x}[t_{k+1}]}{\partial \mathbf{x}[t_k]} + \frac{\partial \mathbf{x}[t_{k+1}]}{\partial \mathbf{u}[t_k]}\frac{\partial \mathbf{u}[t_k]}{\partial \mathbf{x}[t_k]}\right] \tag{3-17}$$

To evaluate the right hand side of this equation, a model of the plant dynamics is needed. This includes all the terms of Jacobian matrix of the coupled plant-controller system, i.e., $\frac{\partial \mathbf{x}[t_{k+1}]}{\partial \mathbf{x}[t_k]}$ and $\frac{\partial \mathbf{x}[t_{k+1}]}{\partial \mathbf{u}[t_k]}$. The TD error for this architecture is given by,

$$E_c = \frac{1}{2}||e_c||^2 = \frac{1}{2}\sum_k e_c^2[t_k] = \frac{1}{2}\sum_k \left[\frac{\partial \hat{V}[t_k]}{\partial \mathbf{x}[t_k]} - \frac{\partial r[t_k]}{\partial \mathbf{x}[t_k]} - \gamma\frac{\partial \hat{V}[t_{k+1}]}{\partial \mathbf{x}[t_k]}\right]^2 \tag{3-18}$$

Where, $\frac{\partial \hat{V}[t_k]}{\partial \mathbf{x}[t_k]} = \hat{\lambda}(\mathbf{x}[t_k], W_c)$ with W_c as the critic parameters. The actor training is much like that in HDP, except that the actor training loop directly utilizes the critic outputs $(\partial \hat{V}[t_k]/\partial \mathbf{x}[t_k])$ along with the system model. Thus, DHP uses models for both critic and actor training.

Action Dependent Heuristic Dynamic Programming (ADHDP)

This is the continuous state analog of well known Q-learning [35]. It is similar to HDP except that critic approximates the action values $(Q(\mathbf{x}, \mathbf{u}))$ instead of the state value $V(\mathbf{x})$. The use Q-function replaces the TD error to be minimized with

$$E_c = \frac{1}{2}||e_c||^2 = \frac{1}{2}\sum_k e_c^2[t_k] = \frac{1}{2}\sum_k \left[\hat{Q}[t_k] - r[t_k] - \gamma\hat{Q}[t_{k+1}]\right]^2 \tag{3-19}$$

Where, $\hat{Q}[t_k] = \hat{Q}(\mathbf{x}[t_k], \mathbf{u}[t_k], W_c)$ and W_c is the parameters of the critic. In this architecture, there is a direct link between the critic and the actor. As a result, a system model is not be required to train the actor. To train the actor, the derivative $\partial Q(\mathbf{x}, \mathbf{u})/\partial \mathbf{u}$ is required, and this term is directly computed by back-propagation. Thus ADHDP requires no model for the training of the critic or the actor.

Action Dependent Dual Heuristic Programming (ADDHP)

It is similar to DHP except that critic approximates the derivatives of the action values ($[\frac{\partial Q(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}} \quad \frac{\partial Q(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}]^T$) instead of the derivatives of state value $V(\mathbf{x})$. Similar to DHP, this architecture also requires a system model to estimate the derivative. The TD error equation for this architecture can be derived by using a *chain rule* that considers all the contributing pathways to the derivatives. It is not presented here for the sake of brevity.

Since critic is already approximating $\partial Q/\partial \mathbf{u}$, no model is needed for training the actor in this architecture. Therefore ADDHP uses a plant model for critic training but not for the actor training.

3-2-2 Modified ACD Architectures

On top of the mentioned four architectures, there are many more ACD architectures. However, the majority of these architectures are modified versions of the basic four algorithms. Among these, two promising architectures are SNAC [50] and J-SNAC [51]. These architectures have been proposed to alleviate the computation load in basic four ACD algorithms. Below are overviews of these two modified architectures.

Single Network Adaptive Critic (SNAC)

This architecture is a modified version of DHP. The main modification is the replacement of the parametric function of the actor with a closed form function. This replacement brings about three advantages,

- Simpler architecture.
- Lesser computational load.
- Elimination of the approximation error associated with the eliminated actor function.

The elimination of the actor-network makes this architecture only valid for input affine systems. However, this limitation is not an issue from the perspective of aircraft control as aircraft is an input affine system.

The critic training in this architecture is the same as the DHP architecture. There is no need to train the actor in this architecture, however the closed-form actor function do require the term $\frac{\partial V(x_{t_{k+1}})}{\partial u_{t_k}}$. In order to compute this derivative, a model of the system is required.

J-Single Network Adaptive Critic (J-SNAC)

This architecture is similar to SNAC, except that it is a modified version of HDP. This architecture provides the same benefits as SNAC and has the same limitation. One additional advantage of this architecture is that it outputs the state values. State values have physical meaning as opposed to its state value derivatives. Furthermore, learning the state value alone makes this architecture a faster learning algorithm.

The training of the critic is similar to HDP, and there is no need of training the actor. Similar to SNAC, this architecture requires a model of the system to compute $\frac{V(x[t_{k+1}])}{\partial u[t_k]}$ and it does so by backpropagation through the critic and the model of the system.

3-2-3 Extended ACD Architectures

Two important extended ACD architectures that are also derived for discrete time systems are Generalized Dual Heuristic Programming (GDHP) and ADGDHP [15]. Other extended ACD architectures include continuous time analogues of the discrete time ACDs [16, 27, 43, 52, 53].

Below are an overview on GDHP and ADGDHP. Other extended ACDs are precluded from the discussion for the sake of brevity.

Generalized Dual Heuristic Programming (GDHP)

This architecture combines HDP and DHP into one. The critic approximates both $V(x)$ and its gradients $\partial V(x)/\partial \mathbf{x}$. This combination is done to improve the approximation of the state value function and bring about superior performance. In this architecture, the critic training is done through a procedure that reduces the temporal difference error for both the value function and its gradient. Combination of the two TD error introduce additional complexities and hence requires relatively higher computational memory and processing. The actor training in this architecture is the same as that of DHP. Therefore GDHP uses models for both critic and actor training.

Action Dependent Generalized Dual Heuristic Programming (ADGDHP)

Similar to GDHP, ADGDHP also combines two other ACD architecture. This architecture combines ADHDP and ADDHP. Thus the critic of ADGDHP estimates the $Q(\mathbf{x}, \mathbf{u})$ and its gradient with respect to states $\partial Q(x)/\partial \mathbf{x}$ and controls $\partial Q(x)/\partial \mathbf{u}$. As with GDHP, critic training utilizes both the ADHDP and ADDHP procedures, and actor training is similar to that of ADDHP. Therefore ADGDHP uses a model for critic training but not for controller training.

3-2-4 Comparison Between Different ACD Algorithms

Different ACD algorithms can be compared to each other for their learning and control performance. Until now there has been no study that compared all ACD algorithms in one single setup. However, there has been some comparison studies where a few of the architectures have been compared.

One such study is that of Prokhorov et al. [15]. In their work, the authors have compared HDP, ADHDP, DHP, and GDHP for an aircraft auto-landing problem. From their findings, they have concluded that HDP and ADHDP do not improve the controller performance after a finite number of training iterations. Furthermore, they have noted that GDHP and DHP achieve similar control performance and these architectures have superior control performances compared to HDP and ADHDP.

In [24], Van Kampen et al. have compared HDP and ADHDP for the control of longitudinal dynamics of F16. Their comparison of learning performance showed HDP has a higher chance of converging to the correct control behavior than ADHDP. Their control performance comparison showed that both of the architectures could adapt to changes in the plant parameters and external disturbances. However, they have noted that HDP is better in adapting when there are changes in plant parameter, and ADHDP is more robust towards external disturbances.

Another comparison study is that of Venayagamoorthy et al. [54]. In their study, the authors have compared HDP and DHP for the control of a Turbogenerators. Their findings agree with that Prokhorov et al., as in they have found better control performance with DHP. However, they did not compare any learning performances of these architectures.

In [26], Pohl has researched on the same problem as Van Kampen et al. except that he has compared HDP and DHP architectures. His results demonstrated that DHP converges faster and takes much lower time to learn desired behavior. Furthermore, his control performance comparison showed that DHP has superior control and adaptation performance than HDP. In so much his results support the findings of the Venayagamoorthy et al. regarding control performance.

In [50], Padhi et al. have compared DHP and SNAC for controlling Piezoelectric microactuators and van der Pol oscillators. The comparison showed that SNAC performed as good as DHP while taking half the time needed for learning. And in [51]. Ding et al. researched the same application as Padhi et al. except that they have compared learning and control performance of the HDP and J-SNAC algorithms. Their analysis had demonstrated that J-SNAC performs as good as HDP while taking half the time to learn desired control behavior. When the studies of Padi et al. and Ding et al. are compared, it can be found that SNAC is the best architecture regarding learning and control performances. Furthermore, it was observed that DHP and J-SNAC have similar learning performances, but DHP has better control performances than J-SNAC. Lastly, it is observed that among these four architectures HDP is the slowest learning algorithm and has similar performance as J-SNAC.

3-3 Application of ACD in Flight Control Systems

The ADHDP algorithm has been used for trimming and adaptive control of Apache helicopter under the name of direct neuro-dynamic-programming in [18, 55].

DHP is used to control a wide variety of aircraft, from nimble fighter planes to large transport aircraft. It has been used to improve controller performance when unexpected changes in aircraft parameters occur [23] or to tune parameters of a reference model [56]. DHP has been used to control the longitudinal dynamics of a linearized aircraft model [57] and to expand the stable region of operations of a simplified model of F8 [58]. DHP has also used in developing controllers for missiles and helicopters [20, 20–22, 59] and in a guidance trajectory generator for reusable launch vehicles [60].

SNAC has been used for optimizing longitudinal dynamics controllers either directly or through a reference model[61, 62] and for controlling a morphing fighter during a pull-up maneuver [63]. It was also used to control aircraft landing on a carrier in the presence of control system model errors [64].

The J-SNAC algorithm has been used to solve nonlinear control problems with model uncertainties. It was only demonstrated to control short period dynamics of a fighter plane and the attitude of a spacecraft with reaction wheel [65, 66]. J-SNAC controllers have not been validated fora large variety of systems, but it is expected to control the full dynamics of an aircraft.

3-4 Conclusion

This chapter presented a review of relevant literature on ACD's. The goal of this chapter was to discuss various ACD algorithms, explore existing applications of ACD's in FCS design and delineate a research and design problem. In so much, this chapter served to answer **R.Q.1.1: What is ACD?**, **R.Q.1.2: What are the theoretical differences between the different ACD architectures?** and **R.Q.1.3: How does J-SNAC algorithm perform in comparison to other ACD architectures?**

State-of-the-art literature was studied to get insights on ACD. In this chapter basis of ACD algorithms and overview on eight ACD algorithms has been presented. Other ACD architectures were left from this discussion because they are either extended or modified versions of these eight architectures. In the literature, it was found that DHP is the most popular architecture for FCS design. However, there has been a growing interest in SNAC and J-SNAC algorithms in the recent studies, because of their faster learning capabilities.

Being the simplest ACD algorithm, J-SNAC promises to be fast learning and rapidly deployable. However, until now J-SNAC has only been validated for controlling the short-period dynamics of F-16. It is expected that this controller is also capable of controlling the full dynamics of an aircraft. However, there is no such validation at the moment. Therefore, in this research, an attempt will be made to validate this architecture for lateral-directional dynamics control.

Chapter 4

Preliminary Analysis on J-SNAC

The previous chapter presented a review of ACD algorithms. There, several ACD architectures were presented. Furthermore, it was concluded that this research would focus on the least computationally expensive ACD architecture, called J-SNAC. This chapter gives the design and analysis of a controller, based on this algorithm. Functionally, this chapter serves to answer the *Research Question 2: What are the conditions for successful implementation of a J-SNAC algorithm.*

Section 4-1 describes the control problem, used for defining the paradigm for this design and analysis. Next, Section 4-2 describes the controller, its sub-systems, the learning procedure, and the implemented algorithm. Then, Section 4-3 presents the training and post-training performance of the controller. Afterward, Section 4-4 presents sensitivity analysis on the controller hyper-parameters. Finally, Section 4-5 presents concluding remarks from this preliminary analysis.

4-1 Control of an Under-Actuated Pendulum

In this phase of the study, J-SNAC is used to develop a learning control law for an under-actuated pendulum. The pendulum consists of a pole and a torque motor attached to a pivot point. The motor can exert a limited amount of torques to rotate the pole about the pivot. The control objective is to use the limited torques to hold the pendulum in the upright position. Restricted torques make this task difficult when the pendulum is in a downright position. In such case, the controller has to swing the pendulum back and forth to gain kinetic energy and use this energy to drive the pendulum to the upright position. Learning to attain this swing-up objective demonstrates the applicability of the algorithm, for solving non-trivial control problems.

Figure 4-1 shows kinematics and dynamics of this setup. Equation 4-1 gives the equation of motion of the pendulum. Moreover, relevant system parameters are given in Table 4-1.

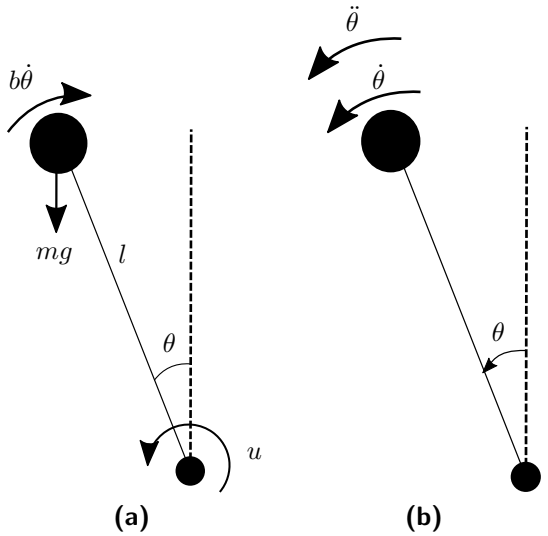


Figure 4-1: Dynamics of the pendulum states. Figure (a) shows the forces and moment acting on the pendulum and Figure (b) shows its sign conventions for the pendulum states.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \frac{mgl}{J}\sin(\theta) - \frac{b}{J}\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} u \tag{4-1}$$

Table 4-1: System parameters for the pendulum swing up and balance problem

Parameter	Description	Values	
J	Lumped mass moment of inertia	1.00	kgm^2
m	Lumped mass of the pendulum	1.00	kg
g	Gravitational acceleration	9.81	ms^{-2}
l	Length of the pole	1.00	m
b	Lumped damping coefficient	0.01	kgm^2s^{-1}

4-2 The J-SNAC Controller

The Figure 4-2 shows the architecture of the J-SNAC controller. It consists of four subsystems, namely *critic, actor, plant model* and *the reward function*, and a parameter adaptation process that updates the critic system. Below are descriptions of the subsystems, the critic adaptation process, hyper-parameters of the controller and the J-SNAC algorithm. The derivation of this algorithm can be found in [27].

4-2-1 Elements of the Controller

Following are the description of subsystems in the J-SNAC controller.

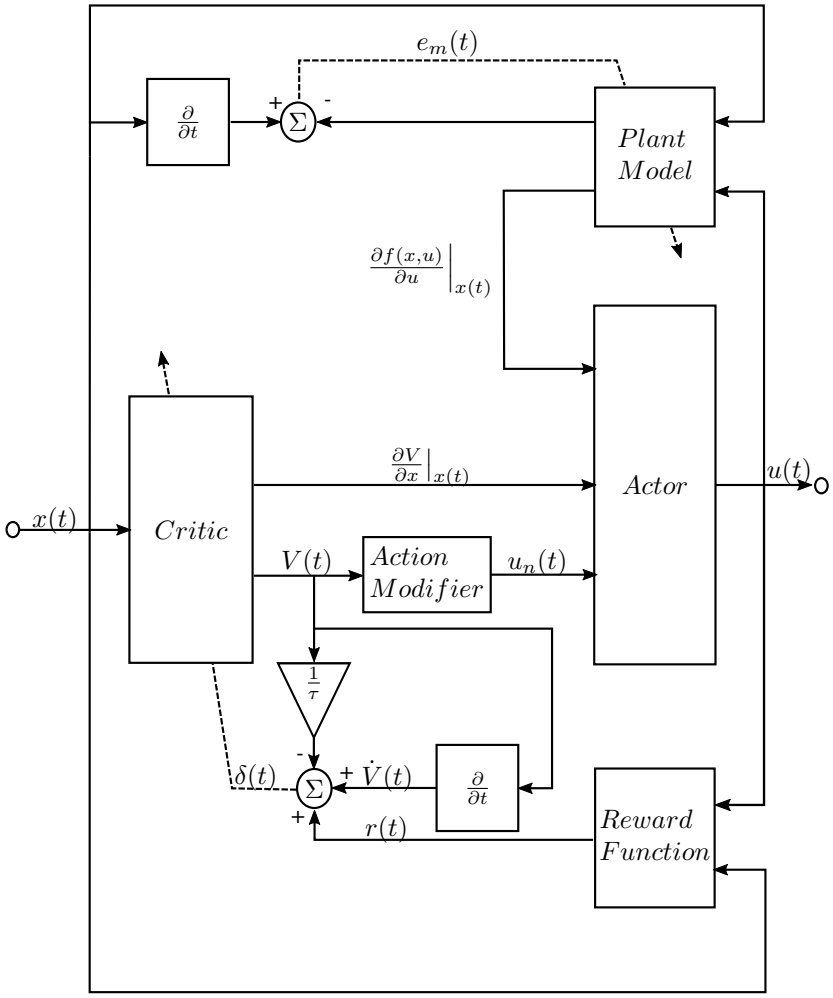


Figure 4-2: J-SNAC control architecture

Critic

The critic has two functions, namely, store the state-value function $V(x)$ and transmit state-values $V(x(t))$ and co-state-values $\frac{\partial V}{\partial x}\Big|_{x(t)}$ to other subsystems of the controller. The critic system stores the state-value-function with a function approximation structure. Additionally, it reads out the state values from the approximated state value function and calculates the co-state-values by performing a backpropagation on the approximated function.

For the pendulum problem, the critic was developed with a Normalized-Radial-Basis-Function (NRBF) network. The choice of this parametric structure is motivated by the fact that NRBF allows changing the approximated function in a local region of the state-space. The system of equations for this structure is given in Equation 4-2, 4-3 and 4-4. Where, $\mathbf{x}$ represent the state of the pendulum, K is the total number of basis functions in the function approximator and $\mathbf{a_k}, \mathbf{r_k}, \mathbf{c_k}$ are the amplitude, spread, and location of the basis functions respectively.

$$V(\mathbf{x};\mathbf{a}) = \sum_{k=1}^{K} a_k v_k(\mathbf{x}) \tag{4-2}$$

$$v_k(\mathbf{x}) = \frac{u_k}{\sum_{l=0}^{K} u_l(\mathbf{x})} \tag{4-3}$$

$$u_k(\mathbf{x}) = e^{\|r_k^T(\mathbf{x}-\mathbf{c_k})\|} \tag{4-4}$$

Plant model

Computation of control actions in ACD architectures requires calculation of Jacobian of the system dynamics with respect to the control input (control effectiveness matrix). A plant model is needed to perform this calculation. If the control effectiveness matrix is not known, then the plant model has to estimate this matrix with state measurements and actions applied by the controller. Typically, plant model is approximated with a function approximation structure, and control effectiveness is computed by performing backpropagation. However, other ways of obtaining the control effectiveness matrix would suffice for the function of the plant model.

For the considered pendulum, the control effectiveness matrix is time-invariant, see Equation 4-5. Hence, no estimation procedure is developed for the plant model, and rather this constant matrix is directly provided to the controller.

$$\left.\frac{\partial f(x,u)}{\partial u}\right|_{x(t)} = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \tag{4-5}$$

Reward function

The purpose of the reward function is to compute the one step performance of the controller. The most important task in developing an ACD controller is to define this function, as rewards are the primary means to shape the controller behaviors.

The chosen reward function for pendulum problem is given in Equations 4-2 to 4-4. The state-dependent part of the reward function penalizes the controller quadratically for deviating from the desired upright position. The action dependent part of the reward function penalizes the controller for crossing the limits of the torque motor. In Equation 4-8, c is the control cost parameter and u_{max} is the maximum available torque.

$$r(\mathbf{x},u) = r(\mathbf{x}) + r(u) \tag{4-6}$$

$$r(\mathbf{x}) = -0.5 \cdot \frac{4}{\pi^2} \cdot \theta^2 \tag{4-7}$$

$$r(u) = -c \cdot \frac{4}{\pi^2} \cdot u_{max} \cdot \ln\left(\left|\sec\left(\frac{\pi^2}{4} \cdot \frac{u}{u_{max}}\right)\right|\right) \tag{4-8}$$

Action modifier

The purpose of the action modifier is to add exploratory actions to the approximated optimal actions. These exploratory actions are needed for the identification of the optimal value function. In classical control literature, this process is known as *persistence of excitation*. Exploratory actions can be synthesized in different ways; the only requirement is that they have to be rich enough for the identification of the value function.

For the pendulum problem, a modulated-filtered-Gaussian-noise is used as the exploratory action. The modulation is performed based on the estimate of the state values. This modulation allows stopping the exploratory actions when the controller is close to the objective state. A filter is used to produce colored noise.

The action modifier system receives the approximated state value from the critic and outputs the exploratory action for the actor function. It uses Equations 4-9, 4-10 and 4-11 to generate the exploratory actions. Where, $u_n(t)$ is the exploratory noise, σ is the modulation factor, $n(t)$ is filtered noise, τ_n is filter time constant, $N(t)$ is the Gaussian noise, σ_0 is the size of the modulation, $r_{\max}$ is maximum expected reward, $r_{\min}$ is the minimum expected reward and $V(t)$ is the value of the current state.

$$u_n(t) = \sigma n(t) \tag{4-9}$$

$$\tau_n \dot{n}(t) = -n(t) + N(t) \tag{4-10}$$

$$\sigma = \sigma_0 \min\left[1, \max\left[0, \frac{r_{\max} - V(t)}{r_{\max} - r_{\min}}\right]\right] \tag{4-11}$$

Actor

The function of the actor is to command the torque motor. The action signals are a function of co-state, control effectiveness matrix, and exploratory input. These signals come from the critic, the model, and the action modifier systems.

The actor system of the pendulum controller generates the command signals with Equation 4-12 (see [27] for the derivation).

$$u(t) = \frac{2 \cdot u_{\max}}{\pi} \arctan\left(\frac{\pi}{2}\left(\frac{1}{c}\left.\frac{\partial v}{\partial x}\right|_{x(t)} \cdot \left.\frac{\partial f}{\partial u}\right|_{x(t)} + u_n(t)\right)\right) \tag{4-12}$$

4-2-2 Critic Update Scheme

In the J-SNAC algorithm, the control policies are learned by estimating the state-value function and improving this estimation with a temporal difference learning scheme. The update scheme can be made simple for faster learning or complex for better performance.

For the pendulum problem, the value function is updated with a TD scheme that uses an eligibility trace. The eligibility trace is used to resolve the credit assignment problem and speed up the learning. Furthermore, the value function is updated by manipulating the

amplitudes of the RBFs (i.e., $\mathbf{a_k}$) alone. This was done to simplify the computations in the learning scheme. Equations 4-13 to 4-16 are used to execute this learning scheme. In these equations, $\mathbf{w}$ and $\mathbf{w}'$ are the vector of critic function approximator parameters before and after the updates, $\Delta\mathbf{w}$ is the change in function approximator parameters, α is the learning rate, $\delta(t)$ is the continuous time temporal difference, $\mathbf{e}(t)$ is the eligibility trace vector at time t, $r(t)$ is the reward received at time t, τ is the time horizon used in the definition of the return value, $V(t)$ is the current state value, $\dot{V}(t)$ is the current time derivative of the state value and κ is the time constant for decay of the eligibility trace.

$$\mathbf{w}' = \mathbf{w} + \Delta\mathbf{w} \tag{4-13}$$

$$\Delta\mathbf{w} = \alpha\delta(t)\mathbf{e}(t) \tag{4-14}$$

$$\delta(t) = r(t) - \frac{1}{\tau}V(t) + \dot{V}(t) \tag{4-15}$$

$$\dot{\mathbf{e}}(t) = -\frac{1}{\kappa}\mathbf{e}(t) + \frac{\partial V(\mathbf{x}(t);\mathbf{w})}{\partial\mathbf{w}} \tag{4-16}$$

4-2-3 Hyper-parameters of J-SNAC

Hyper-parameters of the implemented controller comes from its subsystems. The identified hyper-parameters of this controller are given in Table 4-2. The critic performance depends on the number of its radial basis functions (K). The learning scheme depends on the sampling time Δt, eligibility trace time constant κ, the time constant of return functional τ and the learning rate α. Action modifier outputs depend on the intensity of the exploratory action σ_0 and time constant of noise filter τ_n. Moreover, c shapes the reward function and gradient of the estimated policy function.

Table 4-2: Hyper-parameters of the implemented HDP algorithm

Parameter	Description
K	Number of basis functions in critic network
τ	Time constant of return functional
κ	Time constant for eligibility trace
Δt	State sampling time
α	Value function learning rate
σ_0	Intensity of exploration noise
τ_n	Time constant of action modulator
c	Control cost coefficient

4-2-4 The J-SNAC algorithm

Algorithm 1 presents the implemented J-SNAC algorithm for controlling the pendulum.

Data: Parametric structure for critic; Reward function; Plant model; and Actor Function
Result: Improved value function and control policy
Initialize the hyper-parameters ;
Initialize the duration of episode T ;
Initialize the initial states $[\theta_0 \ \dot{\theta_0}]$;
for $t = 0 : \Delta t : T$ **do**
  Observe the state measurement $x(t)$;
  Compute critic and the model outputs $V(t)$, $\frac{\partial V}{\partial x}\big|_{x(t)}$ and $\frac{\partial f(x,u)}{\partial u}\Big|_{x(t)}$, using the observed state $x(t)$;
  Compute the exploration noise $u_n(t)$ using $V(t)$;
  Compute the action $u(t)$ using $\frac{\partial V}{\partial x}\big|_{x(t)}$, $\frac{\partial f(x,u)}{\partial u}\Big|_{x(t)}$ and $u_n(t)$;
  Compute the reward $r(t)$ with $x(t)$ and $u(t)$;
  Compute temporal difference $\delta(t)$ using $r(t)$, $V(t)$ and an estimate of $\dot{V}(t)$;
  Compute derivative of the value function with respect to its parameters $\frac{\partial V(\mathbf{x}(t);\mathbf{w})}{\partial \mathbf{w}}$;
  Compute the eligibility trace $e(t)$ using $\frac{\partial V(\mathbf{x}(t);\mathbf{w})}{\partial \mathbf{w}}$;
  Compute the increment in value function parameters $\Delta \mathbf{w}$;
  Apply the value function increment with $\mathbf{w} \leftarrow \mathbf{w} + \Delta \mathbf{w}$;
end

Algorithm 1: J-SNAC algorithm

4-3 Controller Implementation and Results

The closed-loop system consisting the J-SNAC controller and the pendulum is shown in Figure 4-3. This setup was produced in the Simulink environment, and subsequently, the controller was trained to perform the pendulum control task. Once the initial training was over, the robustness of the learned policy was tested with two experiments. In the first experiment, the controller was tested for its sensitivity to high-frequency sensor noise. Also, in the second experiment, the available torque was further reduced to see if the controller could perform the same task with reduced torques.

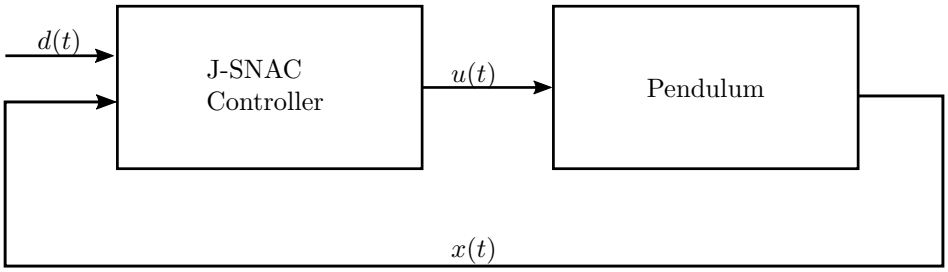


Figure 4-3: Closed loop system with pendulum and the J-SNAC controller. $d(t)$ is the desired state, which is **0** vector. $x(t)$, $u(t)$ are the state vector and the controlled actions

4-3-1 Learning of the Control Policy

The controller was initialized with hyper-parameters in Table 4-3 and was set to be trained for 1000 episodes. Each of the training episodes lasted for 20 seconds. The initial position of the pendulum was randomized during each of the training episodes, to ensure that all states are visited. The training was stopped when there was no significant change in the policy function.

Table 4-3: Hyper-parameters values used in the pendulum controller

Parameter	Value
K	676
τ	10 s
κ	0.1 s
Δt	0.01 s
α	2
σ_0	50
τ_n	0.1
c	0.01

Figure 4-4 shows the trajectory of the root mean square of the change in RBF amplitudes and accumulated rewards received by the agent in each episode. It can be seen that the learned policy had fully converged after 538 training episodes. The converging trend in these figures shows the improvement of the control law from episode to episode. The noise in this measurements comes from the fact that each episode started from a random initial position and the controller was exploring the state and action-spaces while it was being trained.

Figure 4-5 shows explored the state and action space during the training sessions. The controller had encountered most of the states within the space of $[-\pi,\ \pi]$ rad $\times$ $[-9,\ 9]$ rad/s and have tried almost all admissible torques (i.e. within $[-5,\ 5]$ Nm.) in the visited states.

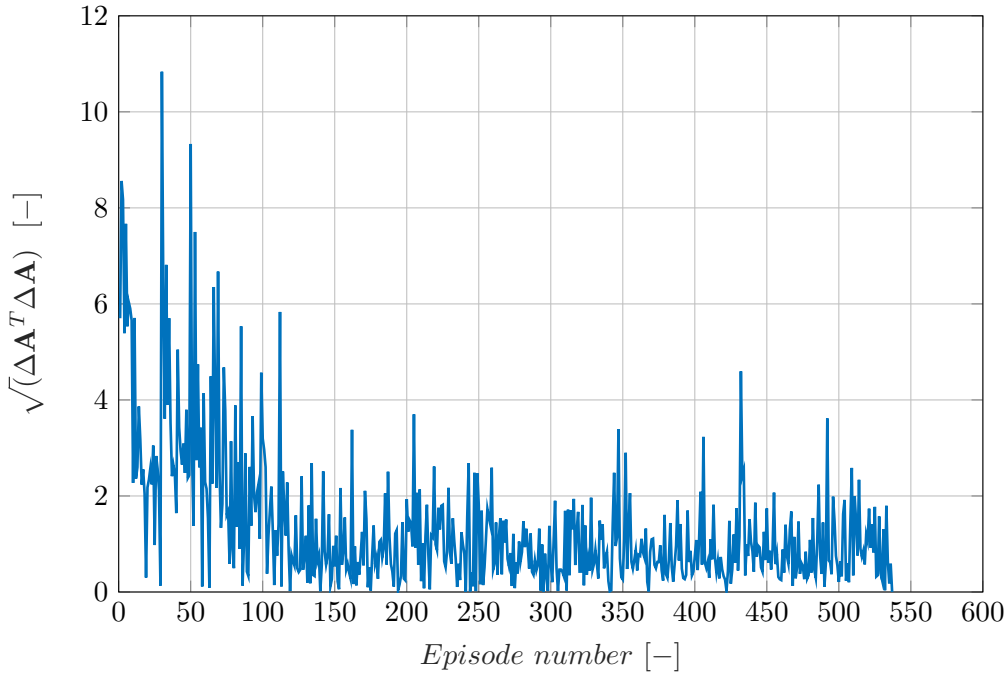
Figures 4-6 and 4-7 shows the surfaces of value and policy functions, before and after the training.

Figure 4-8 shows the performance of the controller after its training. The controller had successfully learned to swing up the pendulum to its unstable equilibrium state and hold this state for a long duration.

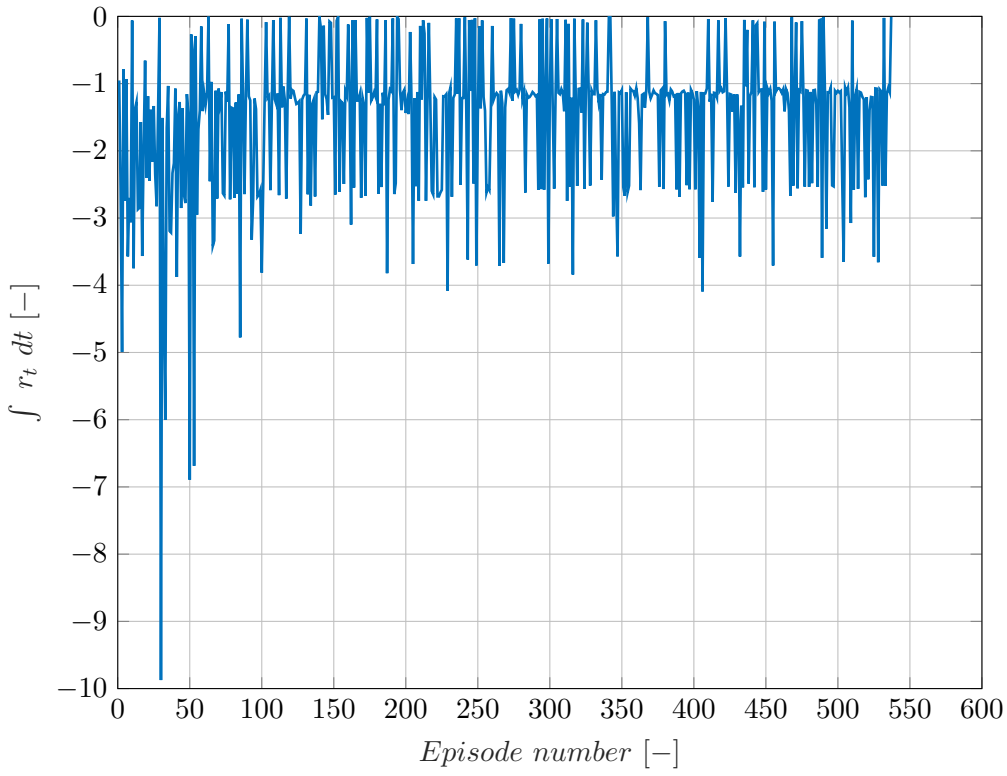
4-3-2 Robustness of the learned control policy

Figure 4-9 shows the performance of the controller in two robustness tests. In the first test, the pendulum state measurements were corrupted by noise. The learned policy of the controller allowed it to overcome this noise and got state measurements to the desired value.

The second robustness test concerned with performing the task with further limited torques. To cause this the controller's maximum torque was reduced by 40%, i.e., $u_{max} = 3$ Nm. The test was to observe if the controller could perform the same task with limited torque or if it could adapt the control policy to perform the task. The controller could perform the swing up with this limited torque, without any further training.



(a) RMS of change in RBF amplitudes



(b) Cumulative rewards

Figure 4-4: Learning performance measurements while being trained. Figure (a) shows the RMS of change in RBF amplitudes and Figure (b) shows cumulative rewards collected by the controller, across the training episodes.

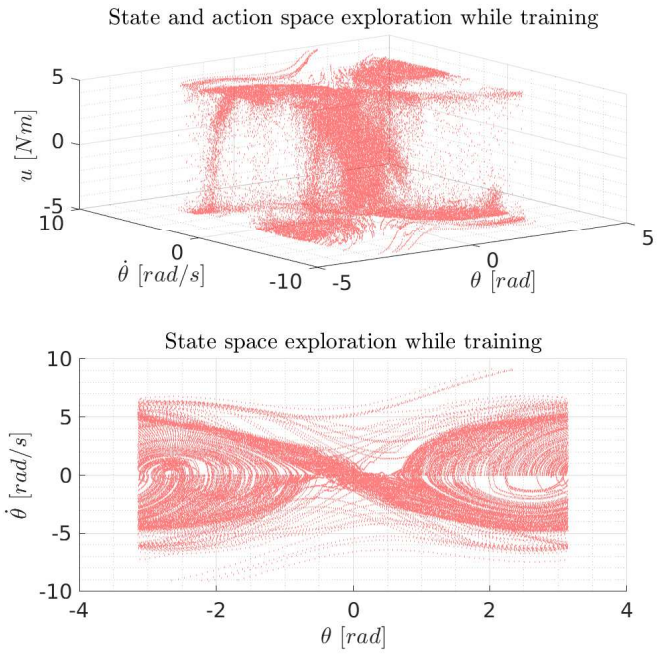


Figure 4-5: Figures showing the exploration of the state and action space by the agent while it was being trained

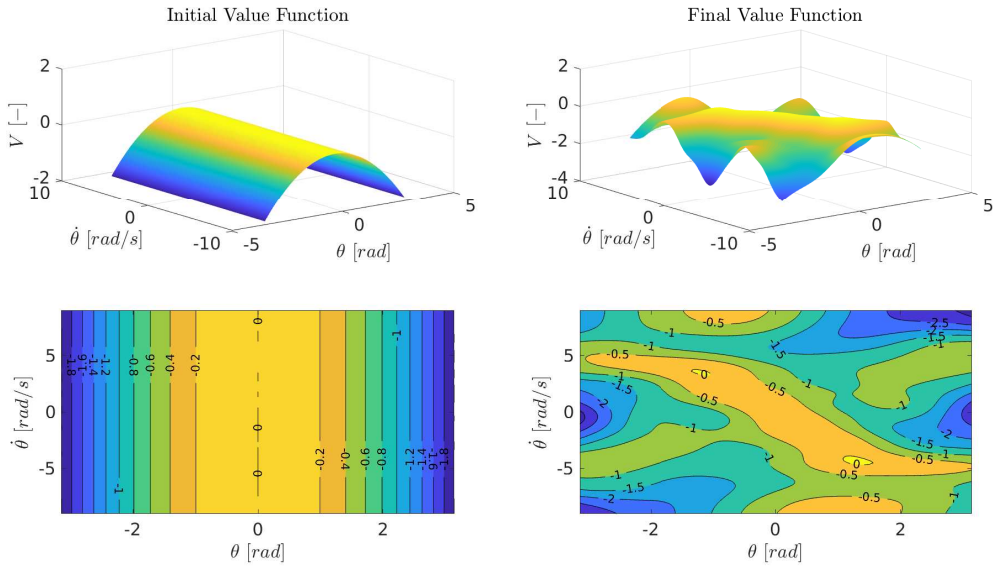


Figure 4-6: Surface of the value function before and after training

4-4 Effects of Hyper-parameters Values on Controller Performance

To determine how the variation of the training routine and hyper-parameters effects the controller performance, 26 controllers were trained and their learning and post-learning control performance were compared. The controllers were trained for 1000 episodes, with each set of

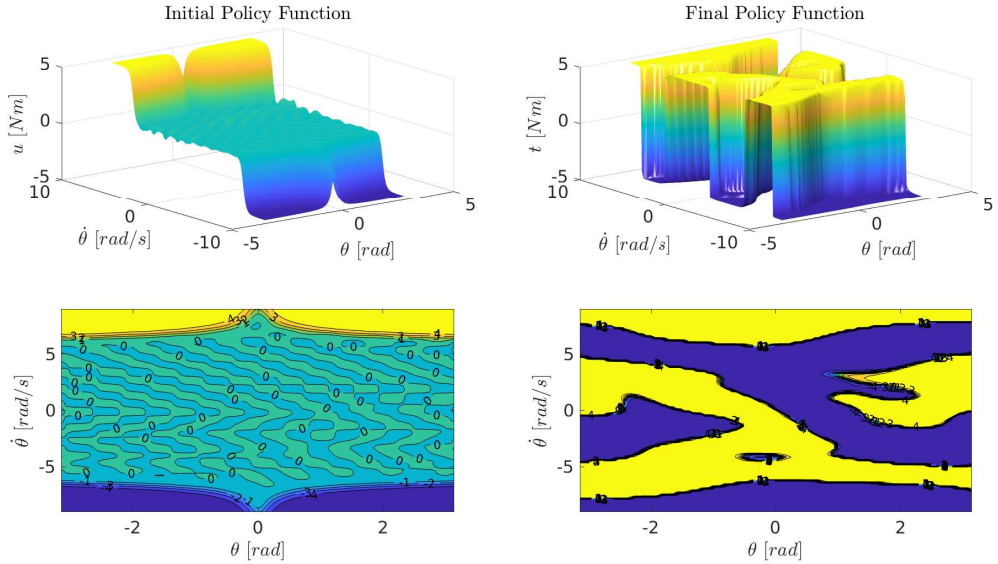


Figure 4-7: Surface of the policy function before and after training

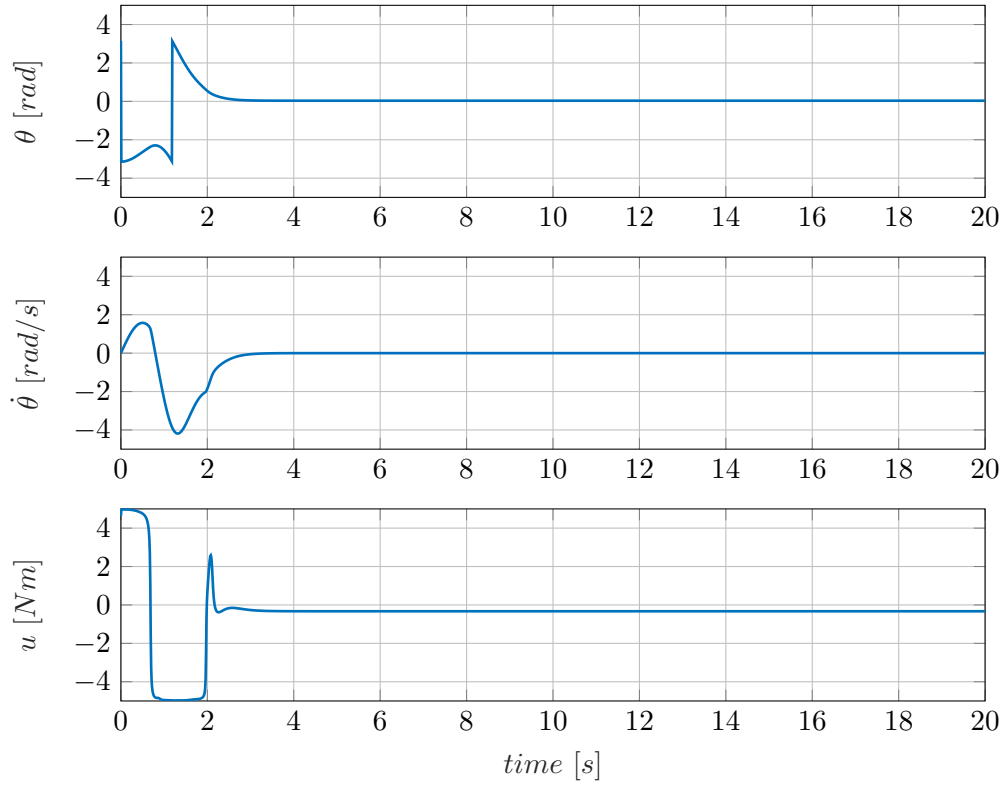


Figure 4-8: Performance of the agent after its training on the swing-up and balance task

the hyper-parameters. No stopping criteria were used to stop the training, to keep the number of training episode as a control variable. Each episode started with random initial position and lasted for 20 seconds. The effect on learning performance was determined by comparing

the rate of change of policy over episodes. The rate of change of policy over episode was measured by calculating the root mean square (RMS) of changes of control actions assigned to 10201 preselected states. The changes in RMS value over episodes have a stochastic nature; therefore a moving average filter was used to smoothened this measure before they were compared. The effect on post learning control performance was determined by comparing the efficacy of the controllers to perform the swing-up-task, cumulative rewards collected while using the learned control policy to perform the swing up task and state of the pendulum after 20 s.

Table 4-4 shows the hyper-parameters values used for the sensitivity analysis. In the table, the second column gives the value of the parameters when they were kept constant, and other columns give the values when they were changed.

Table 4-4: Hyper-parameters values used for sensitivity analysis

Parameter	Control Value	Changed Value 1	Changed Value 2	Changed Value 3
K	1600	100	400	6400
τ	1 s	0.1 s	0.01 s	10 s
κ	0.1 s	0.05 s	1 s	10 s
Δt	0.1 s	0.005	0.01	0.2
α	0.5	0.005	0.05	5
σ_0	5	0.05	0.5	50
τ_n	1	0.02	1	10
c	0.1	0.01	1	10

Firstly, the effect of changing the number of basis functions in the value function architecture was determined. Figure 4-10 and Table 4-5 compares the learning and post learning performance of four controllers, with varying number of basis functions. The policy converging rate seemed independent of the number of basis function used, as initial, final and global rate of change of policy does not follow any trend when the number of basis function is increased. However, the post-learning performance of the controller was best when 1600 RBF were used, after that is the controller with 400 RBF. The controller with 100 and 6400 basis-functions failed to perform the swing- up-task. From this observation, it can be said that the number of basis functions affects the learned knowledge non-linearly but does not affect the learning process.

Secondly, the effect of changing the time constant of return functional τ was determined. Figure 4-11 and Table 4-6 compares the learning and post learning performance of the controller, trained with four different τ values. The policy converging rate seemed to decrease with the increasing τ values. Also, final change in policy also reduces with increasing τ values. The post learning performance is also better with increasing τ values. Therefore it can be said that the post-learning performance of the controller can be improved, at the expense of the learning rate, by increasing the time constant of return functional.

Thirdly, the effect of changing the time constant of eligibility trace κ was determined. Figure 4-11 and Table 4-6 compares the learning and post learning performance of the controller, trained with four different κ values. The policy converging rate seemed to increase with the increasing κ values. There are more numbers of sudden spikes in the change in policy when κ

is increased. The controllers with κ of 0.1 s, 1 s and 10 s learned to perform the swing-up-task. However, the performance is best when κ had the value of 1 s. From these observations, it can be said that the learning rate improves when κ has higher values, but it does not necessarily improve the controller performance.

Fourthly, the effect of changing the state sampling time Δt was determined. Figure 4-13 and Table 4-8 compares four controllers trained with four different sampling rate. The learning and control performance deteriorates with the increase in sample time. This can be attributed to the fact that with higher sampling time, less data is available for training the controller. One anomaly in this trend is that post-training performance is better with $\Delta t = 0.01$ s than that of $\Delta t = 0.005$. Therefore, it can be concluded that sampling time needs to be small, but the exact value has to be tuned with diligence.

Fifthly, the effect of changing the value function learning rate α was determined. Figure 4-14 and Table 4-9 compares the sensitivity of the learning and post learning performance for four different values of learning rate. The learning is faster with higher value learning rate, which is a logical consequence. However, the post-training performance deteriorates when the learning rate is increased from 0.5 to 5. This deterioration is due to over-training. Therefore, if this learning scheme is used, the learning rate has to be tuned beforehand or adapted on-line, in such a manner that it learns relatively fast but avoids overtraining issues.

Sixthly the effect of changing the intensity of exploratory noise was determined. Figure 4-15 and Table 4-10 compares the sensitivity of the learning and post learning performance for four different noise intensities. Although the controller learned to perform the swing-up-task with all of the noise settings, the initial learning rate is higher with lower noise intensities. Furthermore, there is no direct correlation between the level of noise intensity and controller performance. Anyhow, the performance is best when noise intensity is 5. From this, it can be concluded, similar to previous parameters, this parameter effect the performance nonlinearly and hence has to be tuned or adapted with diligence.

Subsequently, the effect of changing time constant of the action modulator filter, τ_n, was determined. Figure 4-16 and Table 4-11 compares the sensitivity of the learning and post learning performance for four different filter time constant. τ_n is related to the cutoff frequency of the exploration noise filter. The change in bandwidth of the noise frequency does not affect the learning rate. It is seen that the controller performs worse, with both high and low frequency of noise. This non-linearity prohibits making a conclusive remark on what bandwidth of noise makes the exploratory signal most useful.

Next, the effect of changing the control cost coefficient parameter c was determined. Figure 4-17 and Table 4-12 shows the learning and post learning performance of four controller with varying values of c. The changes in learned control law are higher with lower c values. Furthermore, the controller also performs better when c has lower values. This can be attributed to the fact that with lower c, high torques are cheap. This allows the controller to apply high torques, leading to more exploration of the state-action space. As a consequent, the approximation of the optimal policy gets better.

Lastly, the effect of randomization of the initial position in training episodes was determined. For this, learning and post learning performance of two controllers, trained with same hyper-parameters (listed second column of Table 4-4) but one was trained by randomizing the initial position, and other one started all its training episode at downright position, was compared.

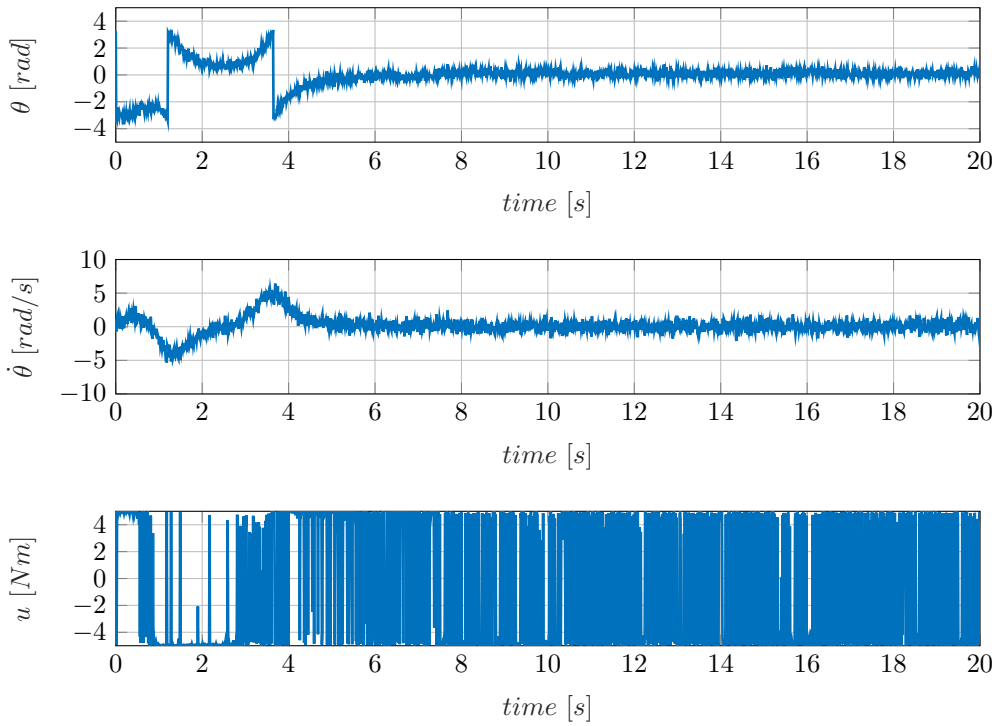
Figure 4-18 shows the change in policy over episodes for these controllers and Table 4-13 shows the post-training performance of the controllers. As can be seen in the figure, the rate of change in policy is higher, when the initial position is randomized. Furthermore, the rate of learning converged later when the initial position is randomized. The final rate of change of policy seemed to stabilize at the same value for both conditions. After training, both of the controllers could learn to achieve the swing-up-task. However, rewards collected is higher, and the final state is closer to the desired zero state when the initial position is randomized. From this, it can be said that encouraging exploration by randomizing the initial state, improves the learning and post learning performance of the controller.

4-5 Conclusion

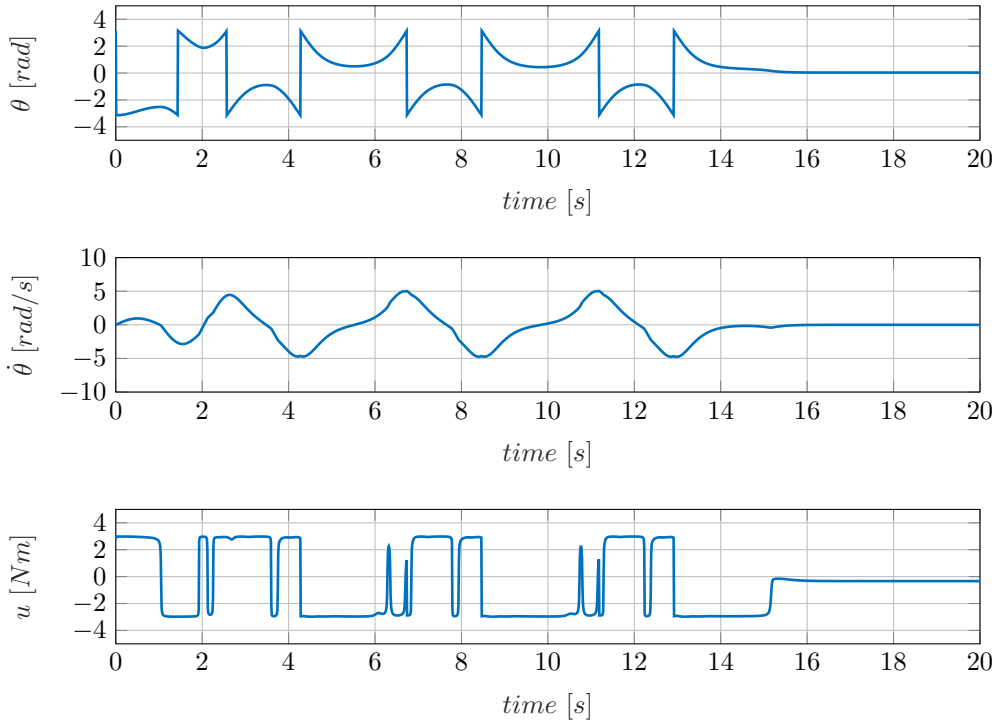
This chapter presented the development, implementation, and analyses of J-SNAC controller. The goal of this chapter was to answer **R.Q.2.1: How to use J-SNAC for the design of the controller for a nonlinear system?**, **R.Q.2.2: What are the hyper-parameters of a controller designed with J-SNAC?** and **R.Q.2.3: How sensitive is the controller performance to changes in hyper-parameters?**

A J-SNAC controller was developed and successfully implemented for a swing up pendulum problem. Eight hyper-parameters were identified, and sensitivity analysis was performed for them. The implementation results confirm that J-SNAC algorithm is suitable for learning and performing non-linear control tasks. Sensitivity analysis on hyper-parameters shows that encouraging exploration, either by decreasing the control cost co-efficient or by starting randomly or by increasing the intensity of the exploration noise, improves the controller performance. However, encouraging exploration delays the convergence of the learned policy. The convergence property could be enhanced by improving the training procedure. The sensitivity study also showed that the effects of change in hyper-parameters are non-linear. For the pendulum problem, hyper-parameters were tuned with error and trial method, until satisfactory results were obtained. Better search methods could be used to find the optimal set of hyper-parameters.

The training of the controller could be improved by scheduling the hyper-parameters values or by tuning them with more sophisticated methods like grid search algorithm. However, such a task is another research in itself. For the sake of brevity, no improvement in *learning scheme* would be made on this research. Instead, it would be considered as a limitation of the design, and further improvements will be proposed at the end of the research.



(a) State action trajectory in the first robustness test



(b) State action trajectory in the second robustness test

Figure 4-9: Results from the tests of robustness of the control law . Figure (a) shows the performance of the controller when there are additive noise on the state measurements. Figure (b) shows the the performance of the controller when the effectiveness of the torque motor is reduced by 40 %.

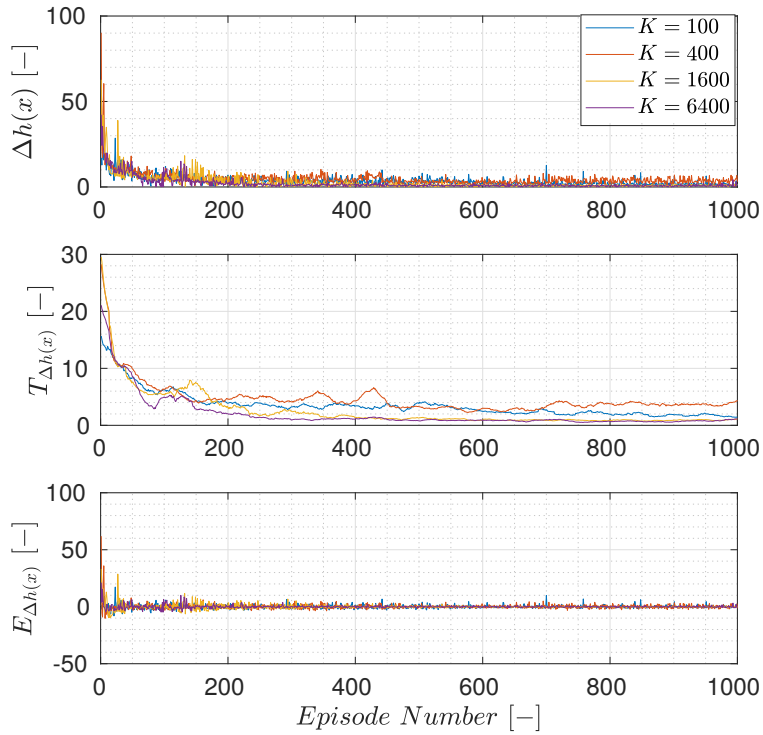


Figure 4-10: Change of policy across training episodes when the number of RBF K in the critic structure is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-5: Performance indices when the number of RBF K in the critic structure is changed.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
Number of RBF $= 100$	0	-05.3304	$(-2.1222, -09.2641)$
Number of RBF $= 400$	1	-07.6863	$(-0.1321, -00.0001)$
Number of RBF $= 1600$	1	-04.5552	$(-0.0048,\ 00.0000)$
Number of RBF $= 6400$	0	-11.2300	$(\ 2.1506,\ 10.1114)$

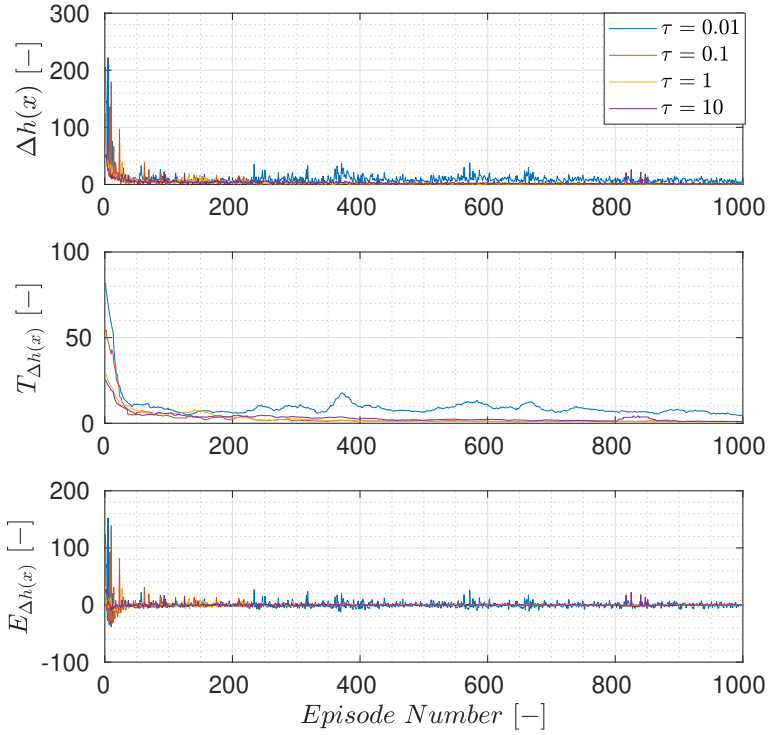


Figure 4-11: Change of policy across training episodes when the time constant of return functional τ is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-6: Performance indices when the time constant of return functional τ is varied.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
$\tau = 00.01$	0	-37.7079	$(\ 3.0398,\ 0.0000)$
$\tau = 00.10$	0	-45.0875	$(-3.1363, -0.0337)$
$\tau = 01.00$	1	-04.5551	$(-0.0048,\ 0.0000)$
$\tau = 10.00$	1	-04.2549	$(\ 0.1670,\ 0.0000)$

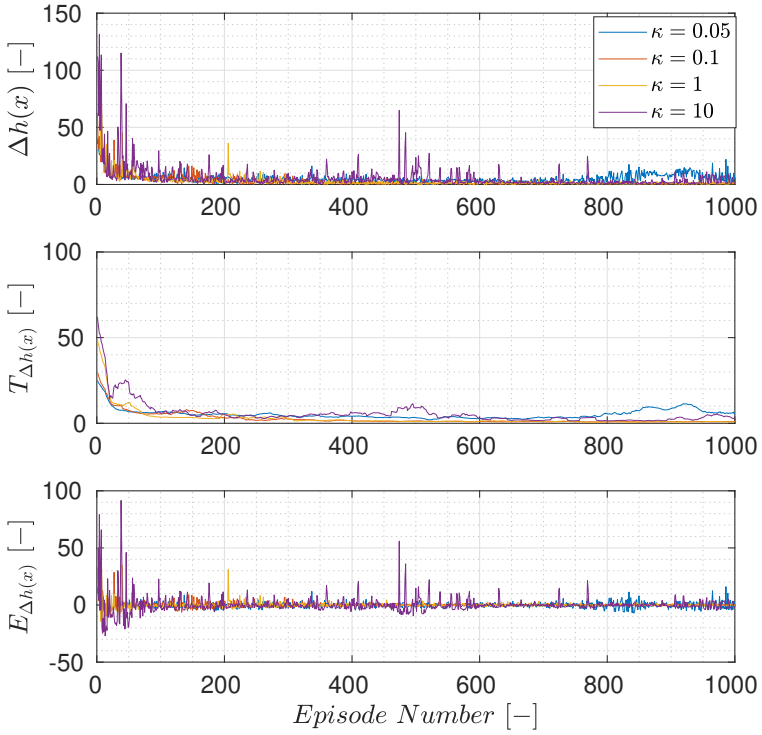


Figure 4-12: Change of policy across training episodes when the time constant of eligibility trace κ is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-7: Performance indices when the time constant of eligibility trace κ is varied.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
$\kappa = 00.05$	0	-35.1633	$(-2.6848,\ 0.0027)$
$\kappa = 00.10$	1	-04.5552	$(-0.0048,\ 0.0000)$
$\kappa = 01.00$	1	-04.3853	$(\ 0.0195,\ 0.0000)$
$\kappa = 10.00$	1	-06.1716	$(-0.0576, -0.0326)$

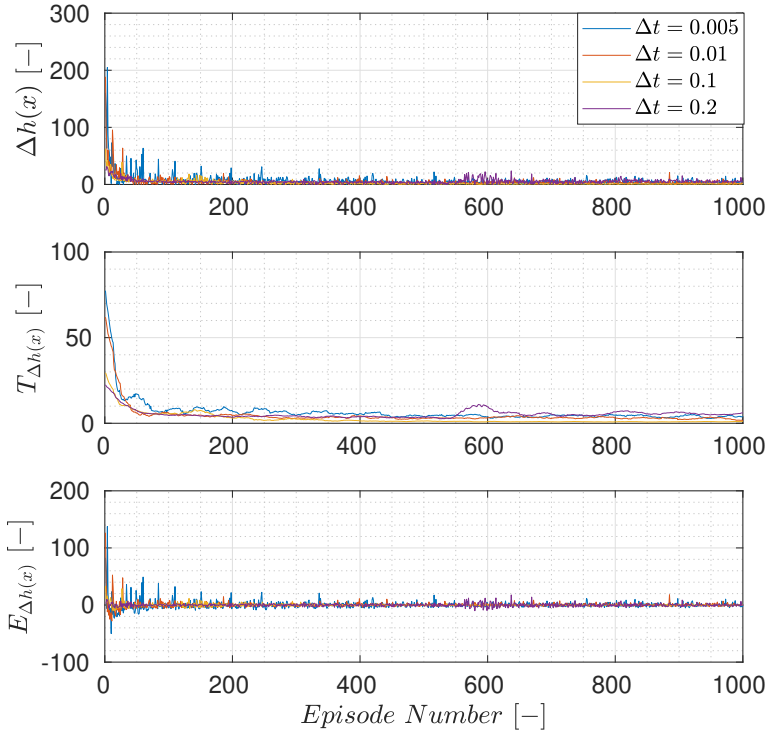


Figure 4-13: Change of policy across training episodes when the state sampling time Δt is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-8: Performance indices when the state sampling time Δt is varied.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
$\Delta t = 0.005$	1	-4.6094	$(\ 0.0101, 0.0000)$
$\Delta t = 0.010$	1	-4.4028	$(\ 0.0077, 0.0000)$
$\Delta t = 0.100$	1	-4.5552	$(-0.0048, 0.0000)$
$\Delta t = 0.200$	0	-9.1419	$(\ 2.5468, 9.0313)$

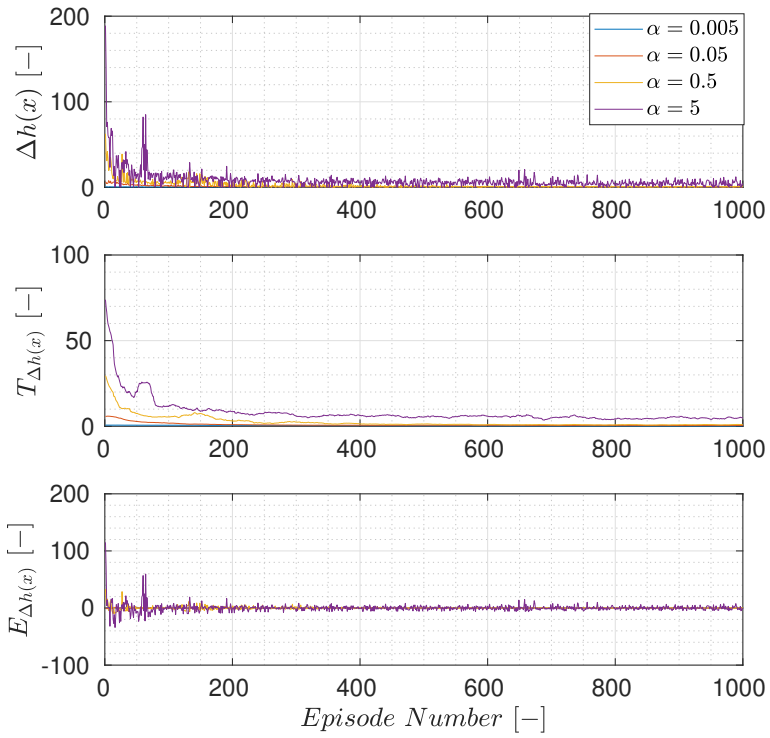


Figure 4-14: Change of policy across training episodes when the value function learning rate α is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-9: Performance indices when the value function learning rate α is varied.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
$\alpha = 0.005$	0	-18.8144	$(-0.1290, 1.4636)$
$\alpha = 0.050$	1	-05.1194	$(-0.0029, 0.0003)$
$\alpha = 0.500$	1	-04.5552	$(-0.0048, 0.0000)$
$\alpha = 5.000$	1	-04.7486	$(0.0420, 0.0000)$

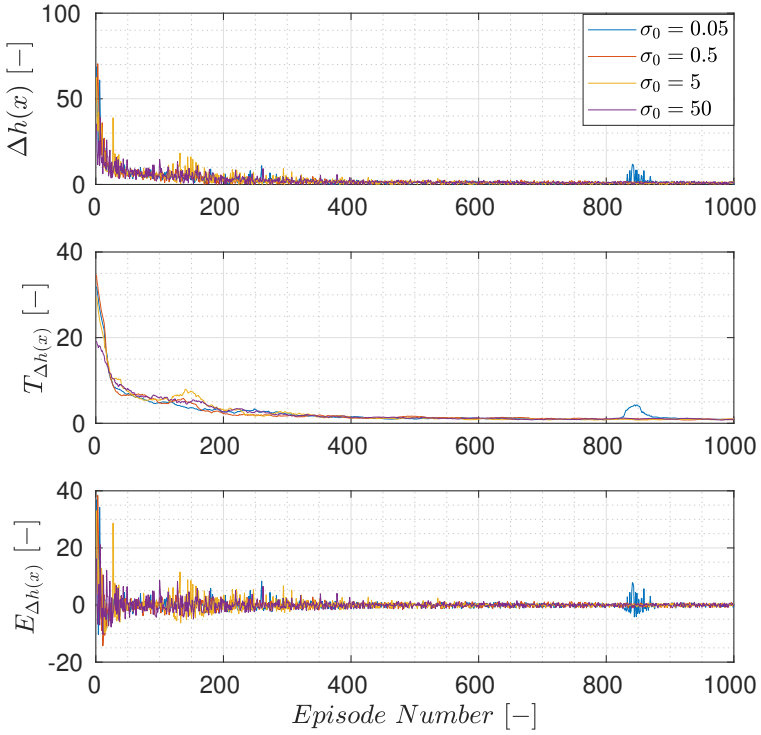


Figure 4-15: Change of policy across training episodes when the maximum value of exploration noise σ_0 is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-10: Performance indices when the maximum value of exploration noise σ_0 is varied.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
$\sigma_0 = 00.05$	1	-05.9064	$(\ 0.0632, 00.0000)$
$\sigma_0 = 00.50$	1	-06.1980	$(-00.0952, 00.0000)$
$\sigma_0 = 05.00$	1	-04.5552	$(-00.0048, 00.0000)$
$\sigma_0 = 50.00$	1	-04.9555	$(\ 0.1168, 00.0000)$

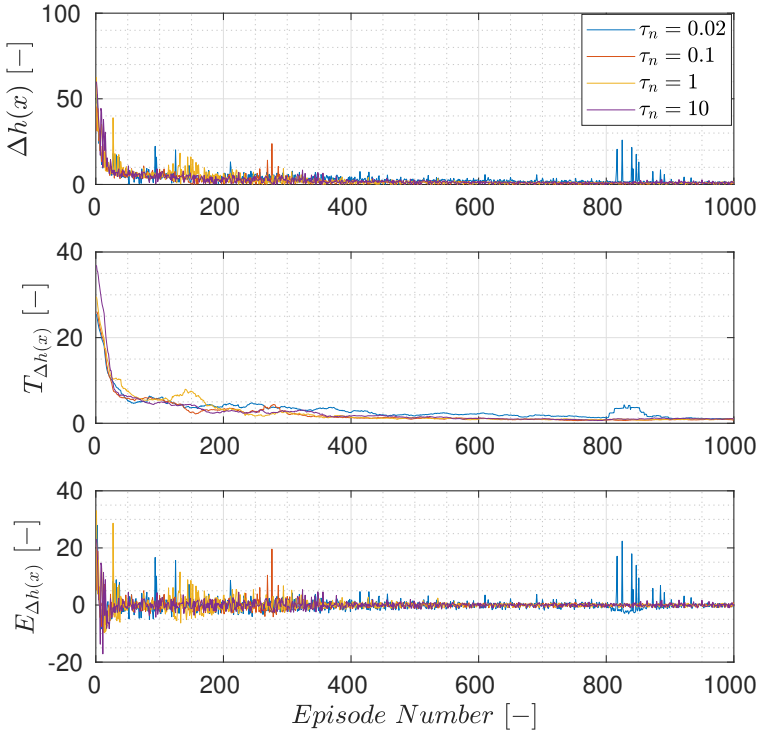


Figure 4-16: Change of policy across training episodes when the time constant of action modulator τ_n is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-11: Performance indices when the time constant of action modulator τ_n is varied.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
$\tau_n = 00.02$	0	-07.7088	$(\ 1.8909, 9.6632)$
$\tau_n = 00.10$	1	-06.6351	$(-0.1380, 0.0000)$
$\tau_n = 01.00$	1	-04.5552	$(-0.0048, 0.0000)$
$\tau_n = 10.00$	1	-06.3426	$(\ 0.1234, 0.0000)$

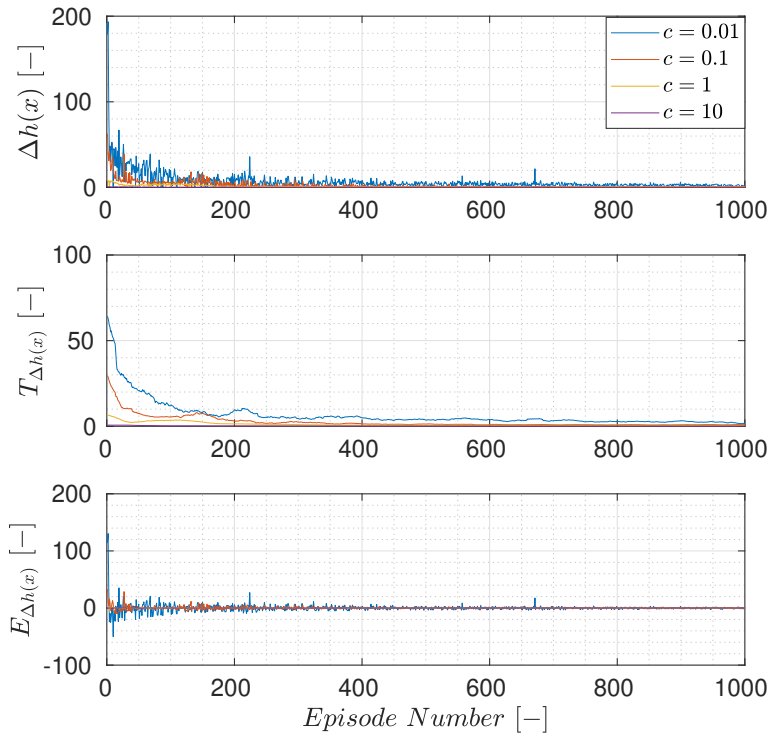


Figure 4-17: Change of policy across training episodes when the control cost parameter c is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter.

Table 4-12: Performance indices when the control cost parameter c is varied.

Parameter setting	Efficacy index	Cumulative rewards	Final state $(\theta, \dot{\theta})$
$c = 00.01$	1	-03.9959	$(\ 0.1181,\ 0.0311)$
$c = 00.10$	1	-04.5552	$(-0.0048,\ 0.0000)$
$c = 01.00$	0	-20.8582	$(\ 0.8195,\ 1.3120)$
$c = 10.00$	0	-39.9890	$(\ 3.1331, -0.0021)$

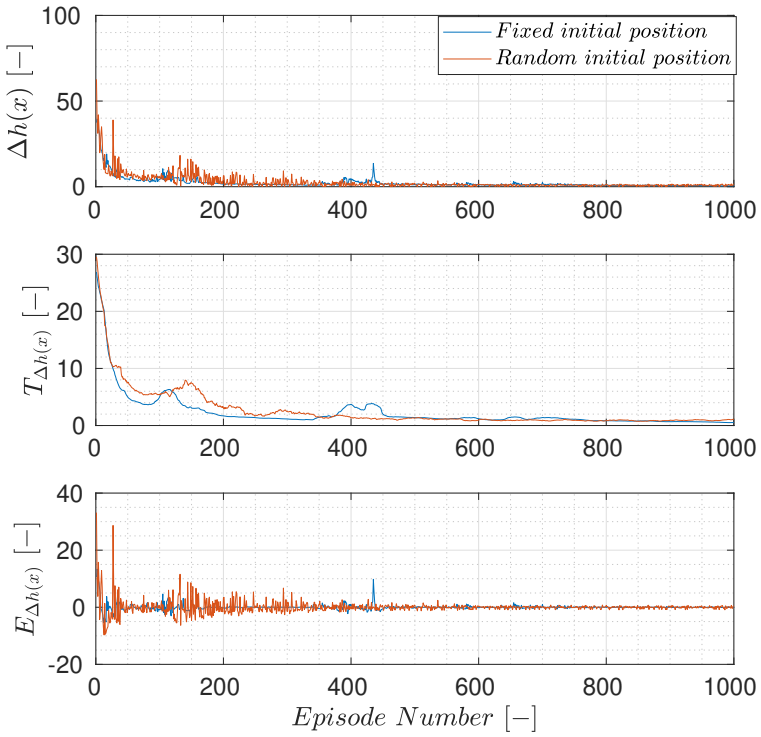


Figure 4-18: Change of policy across training episodes when the number of RBF K in the critic structure is changed. The top figure shows the measured RMS of changes in the policy. The middle figure shows the global trend, obtained by passing a moving average filter through the measured data. The bottom figure shows the residuals from the filter

Table 4-13: Difference between the controller performance indices. In the first condition, each of the training episodes started from the downright position. While, in the second condition, the initial positions were randomized across the episodes.

Parameter setting	**Efficacy index**	**Cumulative rewards**	**Final state** $(\theta, \dot{\theta})$
Random initial state	1	-04.5552	$(-0.0048, 0.0000)$
Fixed initial state	1	-08.3406	$(-0.1060, 0.0000)$

Part III

Additional Results

Chapter 5

Linear Flight Control Systems

In the article, it was stated that a longitudinal and lateral-directional flight controllers were designed with linear control theory. The longitudinal flight controllers served to hold the airspeed and flight altitude of the F16 model at a reference value of 600 feet and 500 feet per second respectively. The lateral-directional flight controllers served as a benchmark to the proposed J-SNAC based flight controller. This chapter elaborates on the design and the performances of these controllers. Section 5-1 and 5-2 presents the structure and the design of these controllers. Then Section 5-3, presents the performance of these controllers for tracking tasks under various condition.

5-1 Longitudinal Dynamics Controllers

This section presents the structure and design of the longitudinal dynamics controller for the F16 aircraft. The objective of this controller is to hold a specified altitude level and airspeed of the F16 model at 5000 feet and 600 feet per seconds, by manipulating the elevator deflection and the throttle settings.

5-1-1 Controller Structure

Figure 5-1 depicts the chosen structure for longitudinal dynamics controller. This controller consist of three linear control laws, namely *Altitude Regulator*, *Pitch Regulator* and *Airspeed Regulator*.

The Altitude Regulator takes 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 is defined with Equations 5-1 and 5-2. 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.

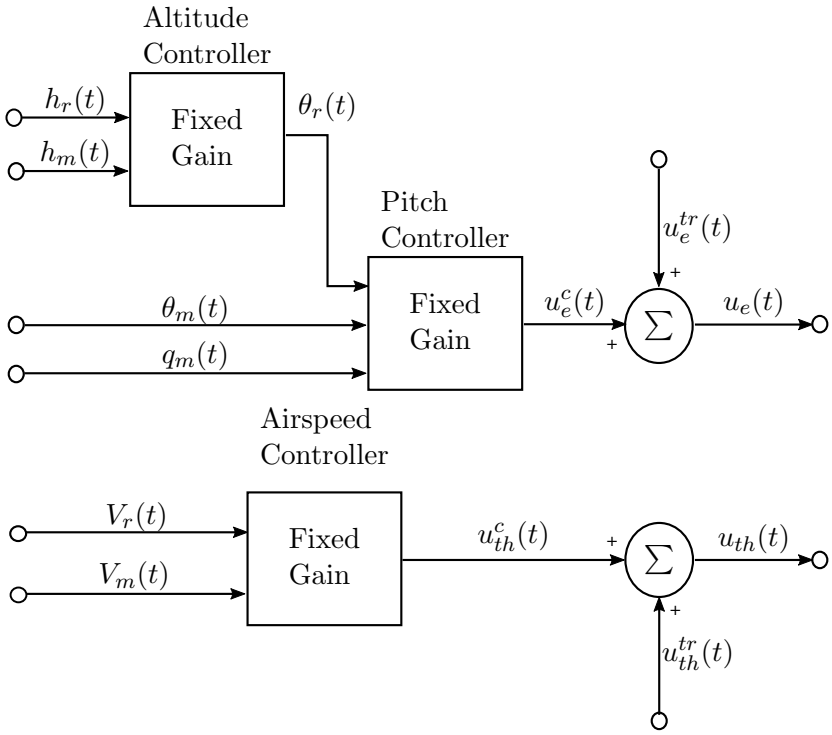


Figure 5-1: The longitudinal dynamics controller structure

$$\theta_r(t) = K_{P_{e_h}} e_h(t) + K_{I_{e_h}} \int_{t_0}^{t} e_h(\tau)d\tau + K_{D_{e_h}} \dot{e}_h(t) \tag{5-1}$$

$$e_h(t) = h_r(t) - h_m(t) \tag{5-2}$$

The Elevator Regulator takes the desired pitch angle, measured pitch angle and pitch rate from the Altitude Regulator and the sensors. The policy of this controller is given in Equation 5-3. In this equation q_m stands for measured pitch rate, θ_m stands for measure pitch angle, and $u_e^c(t)$ stands for dynamic command for elevator deflections.

$$u_e^c(t) = \theta_r(t) - K_\theta \theta_m(t) - K_q q_m(t) \tag{5-3}$$

The combination of two signals determines elevator deflection. The first signal is a dynamic signal ($u_e^c(t)$) generated by the elevator regulator and the second signal is a static signal $u_e^{tr}(t)$ determined from the 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 . The control law of this controller is given in Equations 5-4 and 5-5. 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.

$$u_{th}^c(t) = K_{P_{e_V}} e_V(t) + K_{I_{e_V}} \int_{t_0}^{t} e_V(\tau)d\tau + K_{D_{e_V}} \dot{e}_V(t) \tag{5-4}$$

$$e_V(t) = V_r(t) - V_m(t) \tag{5-5}$$

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.

5-1-2 Controller Gain Determination

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-controller that needs tuning for achieving the control objective.

These parameters were tuned with root locus and successive loop closure methods, to meet the specifications in MIL-F-8785C [67] for the category B flight and level 1 flying qualities. The determined gain values are given in Table 5-1.

Table 5-1: Longitudinal controller parameter values for holding F16 at an altitude of 5000 feet and with an airspeed of 600 feet per second.

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

5-2 Lateral-Directional Dynamics Controllers

This section presents the structure and design of the PID lateral-directional dynamics controller for the F16 aircraft. The objective of this controller is to make coordinated turns to track heading commands by manipulating the aileron and rudder deflections. The controller parameters have been designed to attain its objective at an altitude of 5000 feet and with an airspeed of 600 feet per second.

5-2-1 Controller Structure

Figure 5-2 depicts the chosen structure for lateral-directional dynamics controller. This controller consist of three linear control laws, namely *Heading Regulator*, *Aileron Regulator* and *Rudder Regulator*.

The Heading Regulator takes the 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 logic is defined in Equations 5-6 and 5-7. 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.

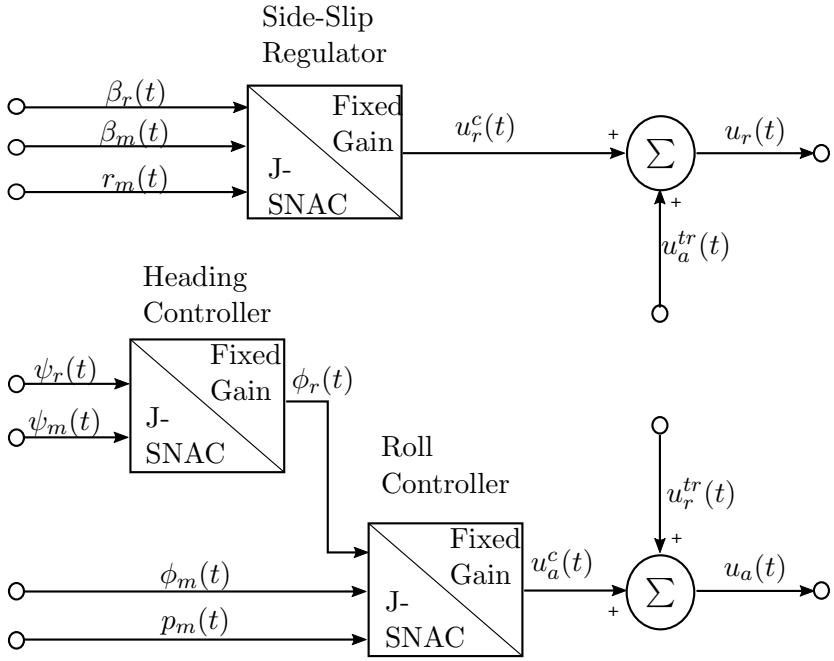


Figure 5-2: The lateral-directional dynamics controller structure

$$\phi_r(t) = K_{P_{e_\psi}} e_\psi(t) + K_{I_{e_\psi}} \int_{t_0}^{t} e_\psi(\tau) d\tau + K_{D_{e_\psi}} \dot{e}_\psi(t) \tag{5-6}$$

$$e_\psi(t) = \psi_r(t) - \psi_m(t) \tag{5-7}$$

The Aileron Regulator takes the desired roll angle, measured roll angle (ϕ_m) and roll rate (p_m) from the Heading Regulator and the sensors/estimator. The control logic for this controller is given by Equations 5-8 and 5-9. 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.

$$u_a^c(t) = K_{P_{e_\phi}} e_\phi(t) + K_{I_{e_\phi}} \int_{t_0}^{t} e_\phi(\tau) d\tau + K_{D_{e_\phi}} \dot{e}_\phi(t) - K_p p_m(t) \tag{5-8}$$

$$e_\phi(t) = \phi_r(t) - \phi_m(t) \tag{5-9}$$

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.

The Rudder 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 Equations 5-10, 5-11 and 5-12. 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.

$$u_r^c(t) = K_{I_{e_\beta}} \int_{t_0}^{t} e_\beta(\tau) d\tau + K_w w(t) \tag{5-10}$$

$$e_\beta(t) = \beta_r(t) - \beta_m(t) = -\beta_m(t) \tag{5-11}$$

$$\dot{w}(t) = -w(t) + r_m(t) \tag{5-12}$$

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

5-2-2 Controller Gain Determination

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 lateral-directional-controller that needs tuning for achieving the control objective.

Similar to the longitudinal controller, these parameters were also tuned with root locus and successive loop closure method. Again, the parameters are tuned to meet the specifications in MIL-F-8785C [67], for the category B flight and level 1 flying qualities. The determined gain values are given in Table 5-2.

Table 5-2: 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.

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

5-3 Performance Of The Linear Controllers for Tracking Tasks

Figure 5-3 presents the configuration for implementing the designed controllers on F16 model. Before the deployment of the controllers, the aircraft is trimmed at the mentioned operating point with the trimming program provided with the F16 model [29]. Subsequently, the controller is made to track reference signals under different conditions. The performance of the controller is quantified with the weighted root mean square errors in relevant state variables (defined in Equation 5-13).

$$\begin{aligned} PI &= -0.1 \cdot \sqrt{\frac{1}{T}\int_0^T \left(\frac{h(t)-h_r(t)}{25}\right)^2 dt} - 0.1 \cdot \sqrt{\frac{1}{T}\int_0^T \left(\frac{V(t)-V_r(t)}{10}\right)^2 dt} \\ &\quad -0.4 \cdot \sqrt{\frac{1}{T}\int_0^T \left(\frac{\beta(t)-\beta_r(t)}{2}\right)^2 dt} - 0.4 \cdot \sqrt{\frac{1}{T}\int_0^T \left(\frac{\psi(t)-\psi_r(t)}{2}\right)^2 dt} \end{aligned} \tag{5-13}$$

5-3-1 Tacking Of A Sinusoid Reference Signal Under Ideal Conditions

The linear flight controller was made to track a sinusoidal heading reference signal. This reference signal is given in Equation 5-14 and it is chosen to observe turning performance in all possible directions. Figures 5-4 to 5-9 presents the response of the linear controllers in this tracking task. According to the defined performance index, the controller's performance is -0.1839.

$$\psi_r(t) = \frac{3}{4}\pi \sin(\frac{2\pi}{180}t - \frac{\pi}{2}) + \frac{\pi}{2} \tag{5-14}$$

5-4 Conclusion

This chapter served to elaborate on the partial answer to research question **R.Q.3.1: How does the proposed controller perform in comparison to a traditional fixed-gain linear controller?**. It has presented the design of the linear flight controllers and its performances for tracking of a sinusoid reference signal.

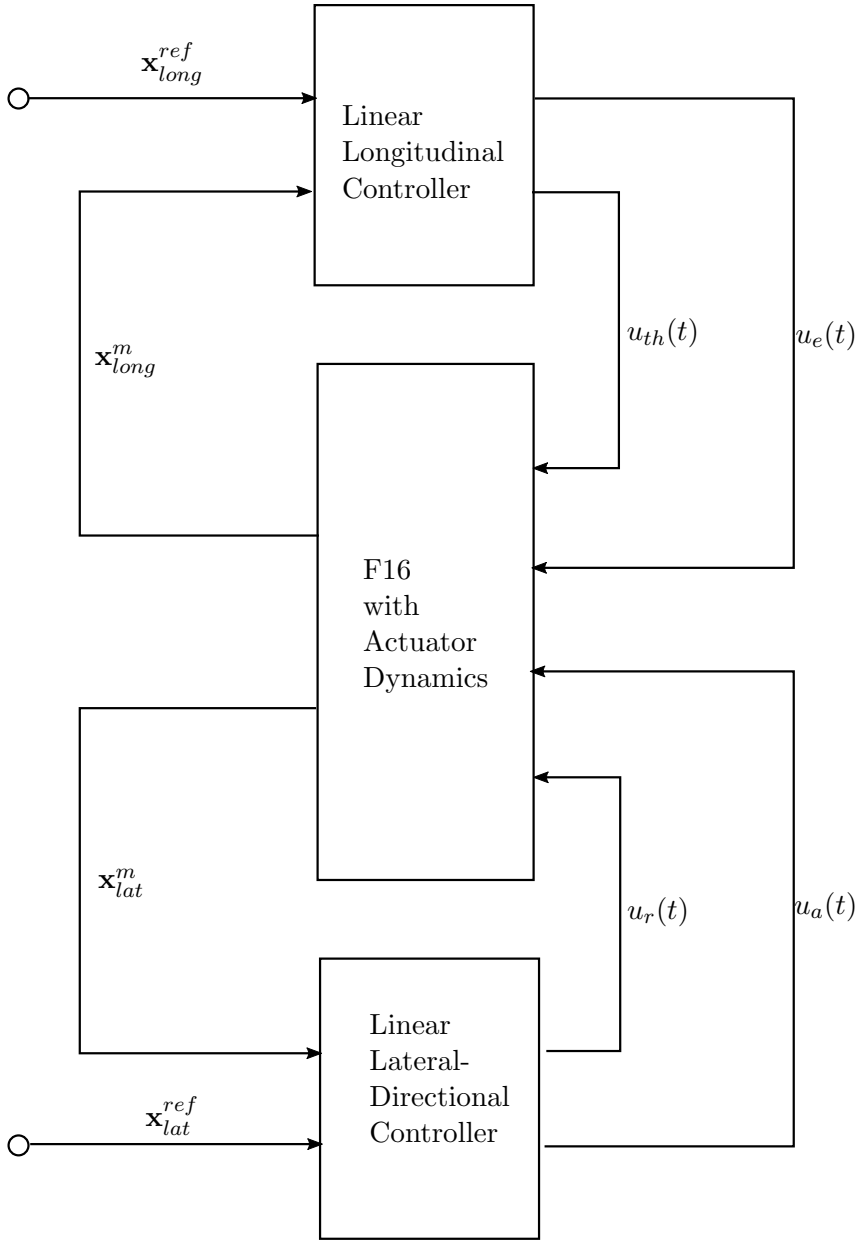


Figure 5-3: F16 aircraft model with linear flight controllers. $\mathbf{x}_{long}^{ref}$ and $\mathbf{x}_{lat}^{ref}$ are the external command signals for longitudinal and lateral states respectively. $\mathbf{x}_{long}^{m}$ and $\mathbf{x}_{lat}^{m}$ are the measured/estimated signals for longitudinal and lateral states. $u_{th}(t)$, $u_e(t)$, $u_a(t)$ and $u_r(t)$ are the command signals for the flight control surface deflection and throttle setting.

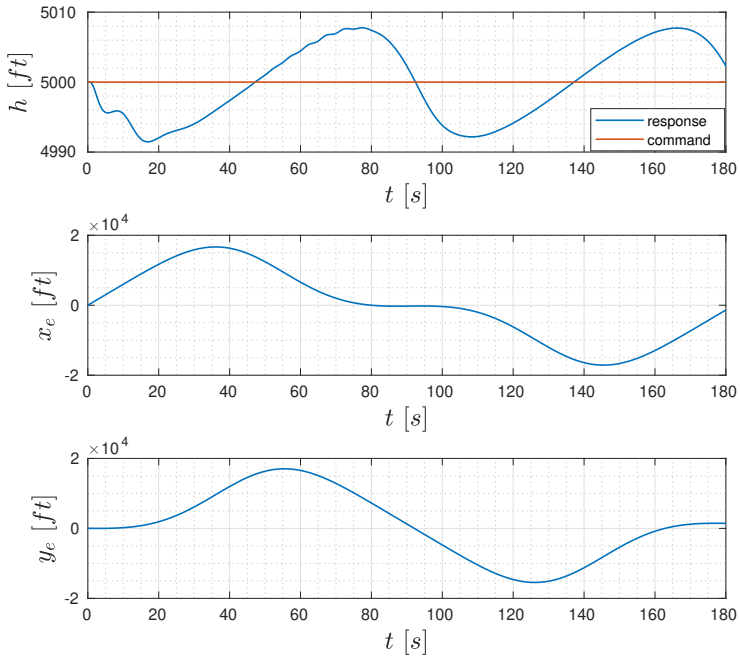


Figure 5-4: Altitude and position response

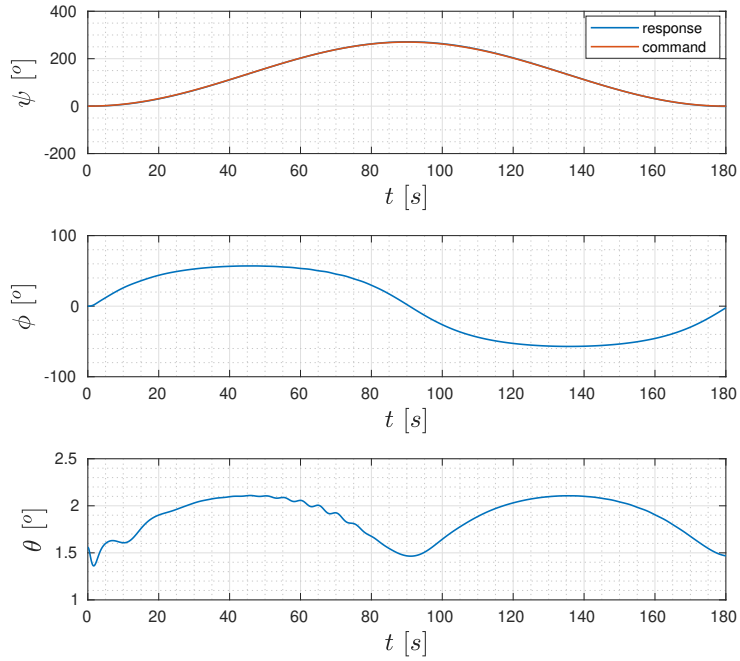


Figure 5-5: Heading and attitude angle response

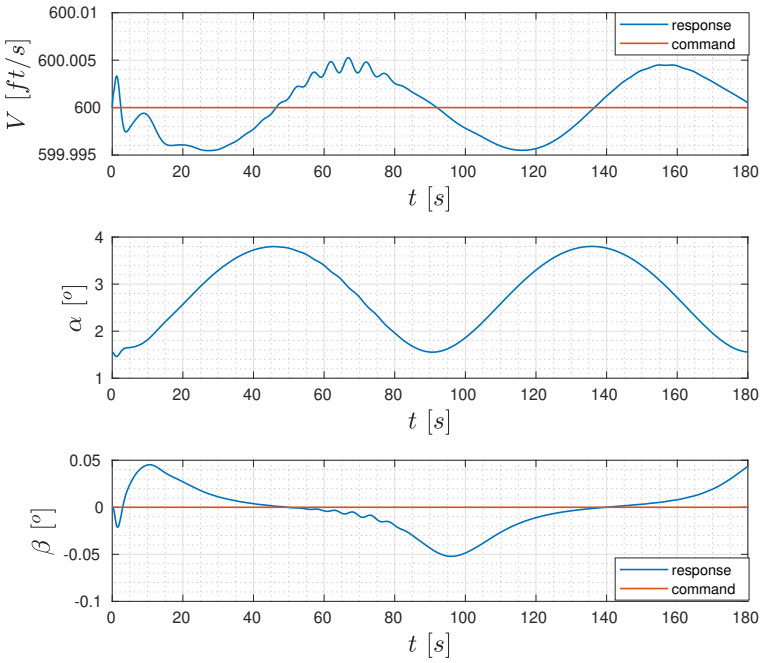


Figure 5-6: Airspeed and aerodynamic angles response

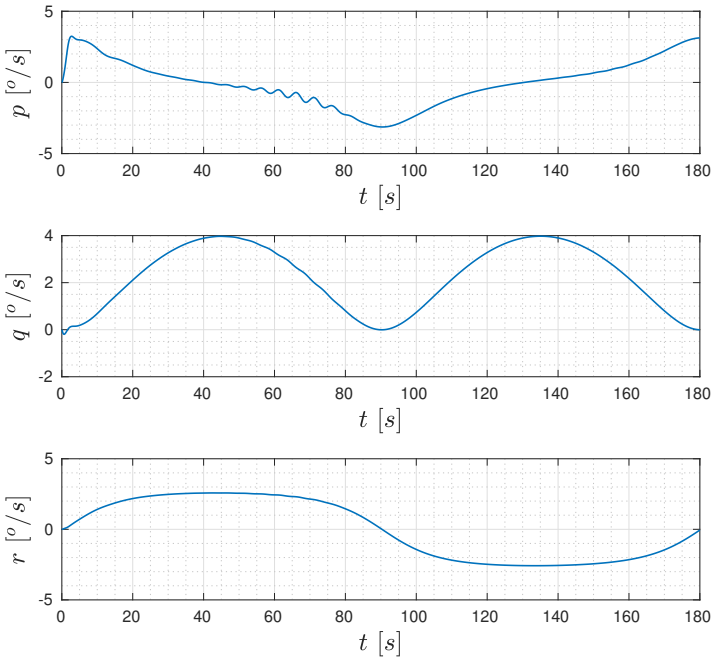


Figure 5-7: Angular rate response

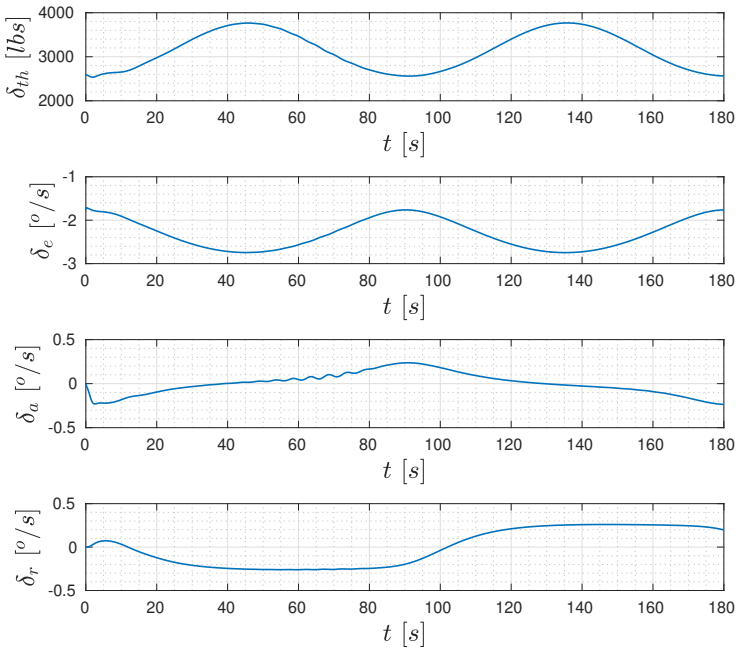


Figure 5-8: Actuator response

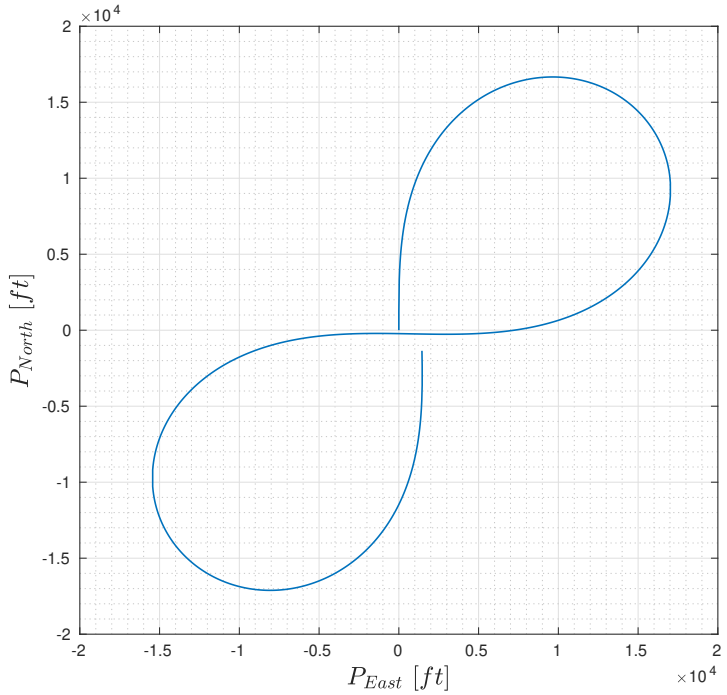


Figure 5-9: Ground Track

Chapter 6

Additional Results And Discussion On J-SNAC Flight Control System

In the article, the design and performance of the J-SNAC based lateral-directional flight controllers were presented. This chapter elaborates further on this flight controller and presents some additional results. Section 6-1 and 6-1 presents discussions on the structure of the controller and its hyper-parameters. Then Section 5-3, presents further performance evaluations results. Afterwards Section 6-4 discusses verification and validation of this flight controller.

6-1 Discussion On The Controller Structure

As stated in the article, a distributed architecture with three J-SNAC controllers is chosen for the lateral-directional flight controller. This distributed architecture brought modularity in the flight control system and reduced the number of state variables for each sub-controller. Splitting of the controller into sub-controllers made the state definitions in the sub-controllers non-Markovian. For example, the proposed side slip regulator took in the body yaw rate r and side-slip angle β measurements as its input. The choice of these variables was motivated from the fact that rudder deflection δ_r changes the yaw rate r and yaw rate can be chosen as pseudo-control for side-slip angles. However, as per the equations of motion, the side-slip dynamics depends on other state variables too, e.g., roll rate p and velocity V. If the state definition for the side slip regulator consisted all state variables, it could perhaps use the other state variables as a pseudo-control to side-slip angles. Including other state variables in the input state can restore the Markovian property and can bring better adaptability. However, it will expand the hyperspace of the value function and makes learning more complicated.

Another limitation of the proposed flight controller is in the state definition of the roll and heading angle trackers. A tracking problem is fundamentally different from a regulation problem. It requires knowledge of the error dynamics, implying knowing the reference signal dynamics. In the proposed trackers, full error dynamics was not utilized. Working with full error dynamics can restore the Markovian property for the trackers' input states and can improve the controller performance.

6-2 Discussion on Hyper-parameters Selection

Assuming that NRFB network is chosen as the value function approximator and only the amplitudes of these basis function are updated to learn the value function, there are eight hyper-parameters in the J-SNAC algorithm. These parameters include the number of basis function, the time constant to discount future rewards, the time constant for eligibility trace, the time constant for exploration noise filter, the time constant for the derivative filter, exploration noise intensity, learning rate, and action cost parameter. The learning and the control performance of the J-SNAC controller are highly dependent on these parameters' value. However, tuning these variables for best performance can be tedious. In this work, a coarse golden section search was used to determine the parameter values. However, the optimality of these values cannot be guaranteed as no evaluation was performed on this aspect. To improve the convergence and to ensure the optimality of the learned policy, other techniques like grid search and Bayesian optimization can be implemented to tune the hyperparameters.

6-3 Control Performance Evaluation

Six tracking tasks were used to evaluate the control performances of the proposed flight controller. First three of the tasks concerned with tracking under ideal condition, then one task concerned with the effects of sensor noise and then two tasks concerned with adaptability. The Performance Index (PI), defined in Equation 5-13, is used to quantify control performances. Below are the results and discussion of these evaluation process.

6-3-1 Tacking Of A Sinusoid Reference Signal Under Ideal Conditions

The first evaluative task was to track a sinusoid reference signal. Figure 7-1 to 7-6 depicts the state trajectories of the aircraft with PID, untrained and trained J-SNAC controllers. Visually, it can be seen that the untrained controller could hardly follow the command signal, but both PID and trained J-SNAC controller have an almost identical response. However, there are two notable difference between the PID and trained J-SNAC controller responses. The first difference is that The PID law produces better side slip attenuation. The second difference is that the J-SNAC controller has a delay in tracking when compared to the PID controller.

As per the Performance Index the PID, untrained, and trained J-SNAC controller has the score of -0.1839, -52.0257, and -1.5802 respectively. The higher score of PID controller can be directly attributed to better slip attenuation and faster response.

6-3-2 Tacking Of A Smoothened Step Reference Signal Under Ideal Conditions

The second evaluative task was to track a smoothened step reference signal. This signal is chosen to verify the control performance on a task that was not in the training program. Figure 7-7 to 7-12 shows the state trajectories of the aircraft with PID, untrained, and trained J-SNAC controller for this tracking task. Similar to the tracking of the sinusoid, non-trained J-SNAC controller failed to perform the tracking while the trained J-SNAC and PID controller

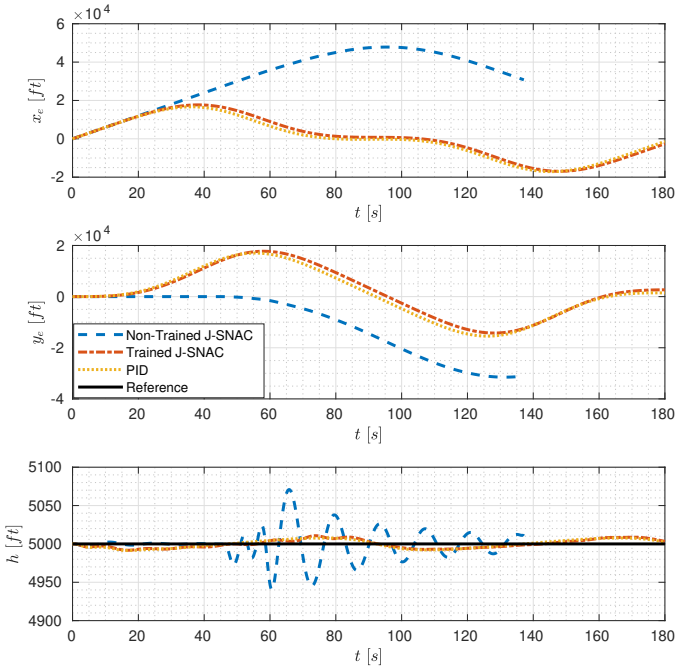


Figure 6-1: Altitude and position responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

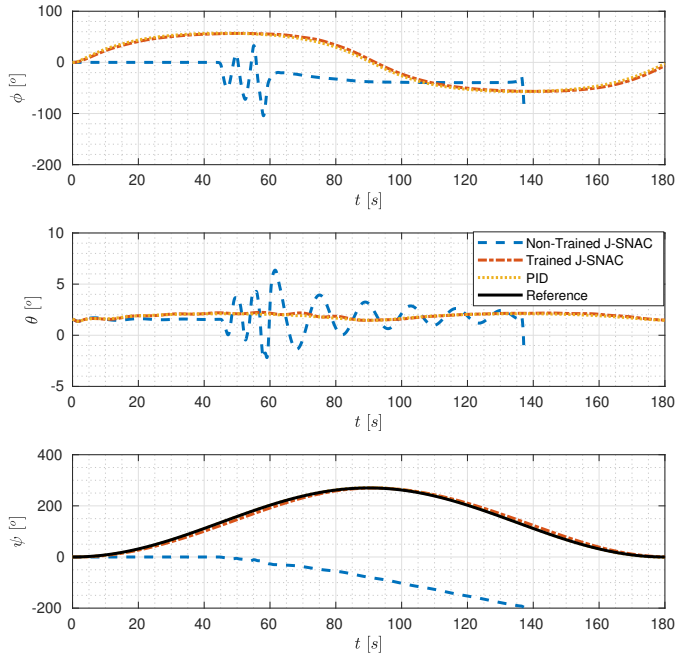


Figure 6-2: Heading and attitude angle responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

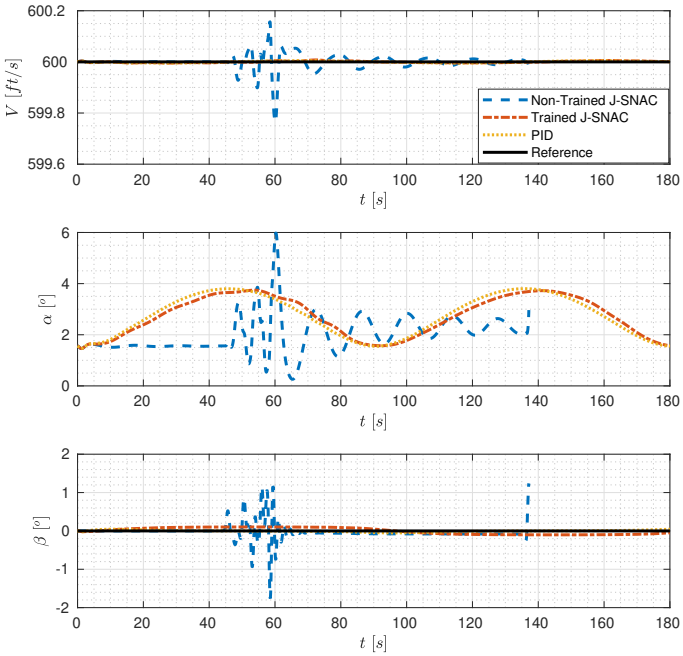


Figure 6-3: Airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.e

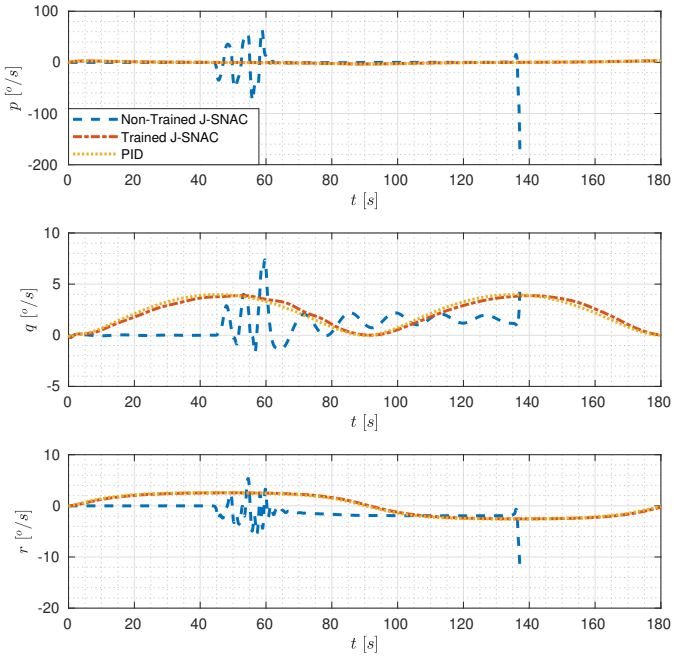


Figure 6-4: Angular rate responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

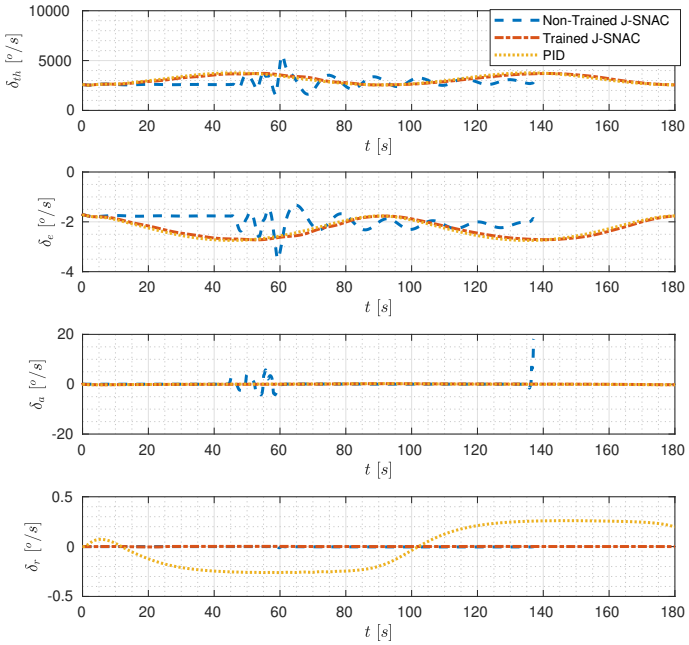


Figure 6-5: Control actuator responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

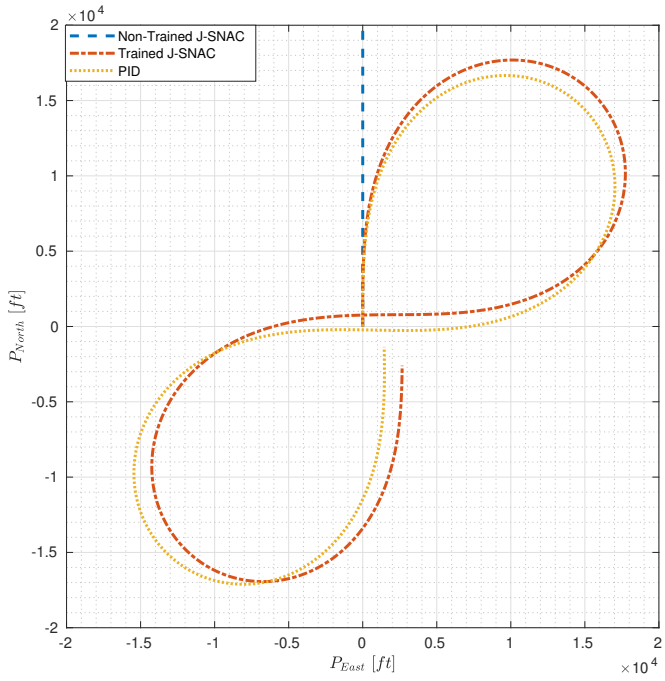


Figure 6-6: Ground track produced by the aircraft while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

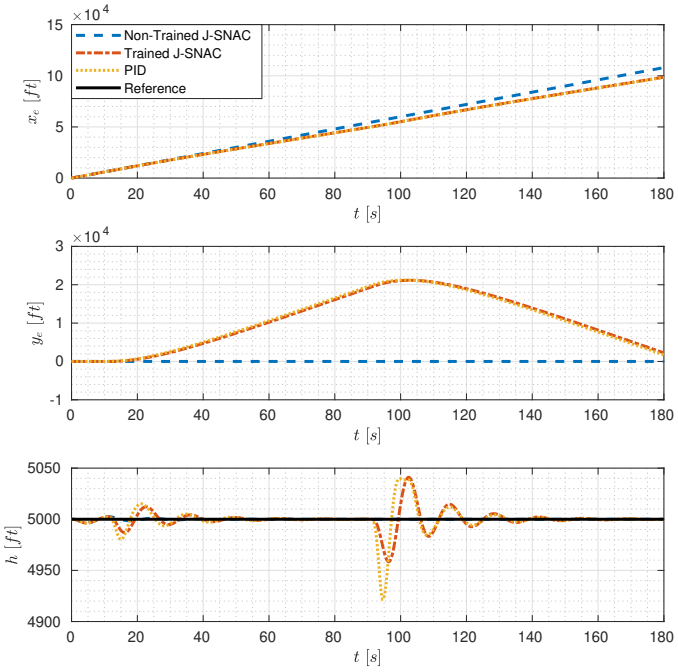


Figure 6-7: Altitude and position responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

performs almost similarly. Again, PID side-slip regulator attenuates incurred side-slip better than the J-SNAC controller, and it responds faster in tracking commands. One additional difference is that PID controllers create more aggressive controls for the aerodynamic surface actuators.

As per the Performance Index the PID, untrained, and trained J-SNAC controller has the score of -0.1623, -4.8565, and -0.3440 respectively. The higher score for PID controller is still due to better slip attenuation and faster response.

6-3-3 Tacking Of A Ramp Signal Under Ideal Conditions

The third evaluative task was tracking of a ramp signal. This signal is also chosen to verify the control performance on a task that was not in the training program. Figure 7-13 to 7-18 shows the state trajectories of the aircraft with PID, untrained, and trained J-SNAC controller for this tracking task. In these figures, it can be seen that the PID controller creates more oscillations in the state responses. These oscillations are because of higher aerodynamic surface deflection commands. Furthermore, the J-SNAC controller does not compensate for a small side-slip angle whereas the PID controller attenuates the side-slip angle over time.

As per the Performance Index the PID, untrained, and trained J-SNAC controller has the score of -0.1261, -57.2332, and -1.1436 respectively.

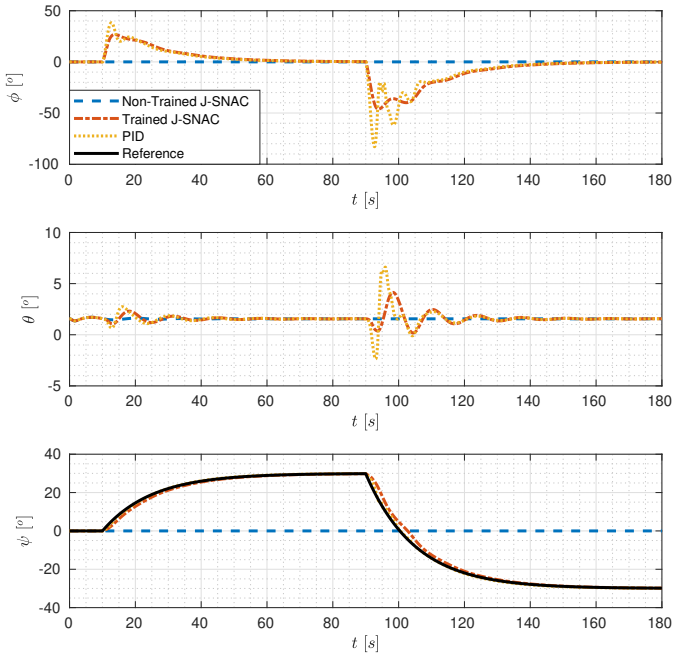


Figure 6-8: Heading and attitude angle responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

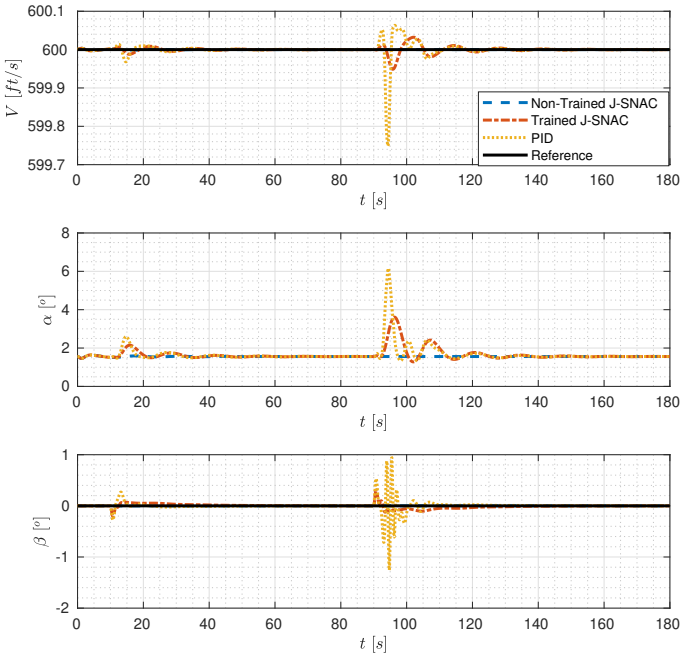


Figure 6-9: Airspeed and aerodynamic angles responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

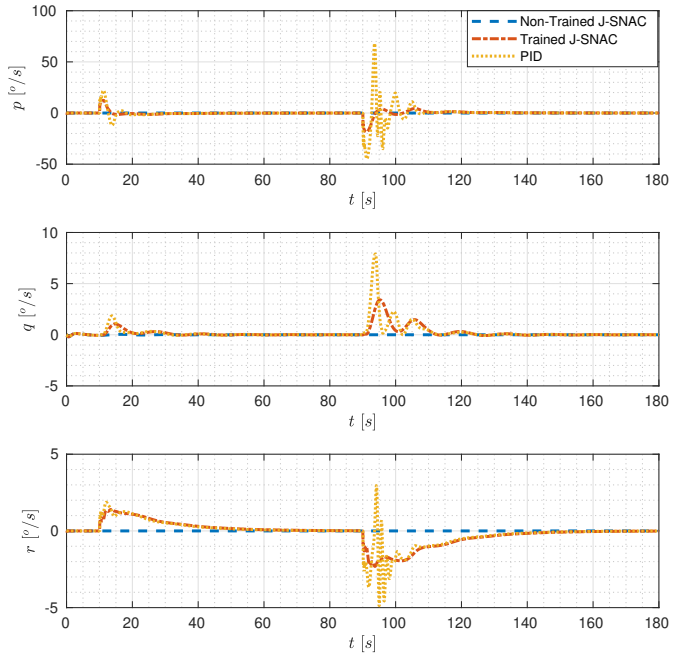


Figure 6-10: Angular rate responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

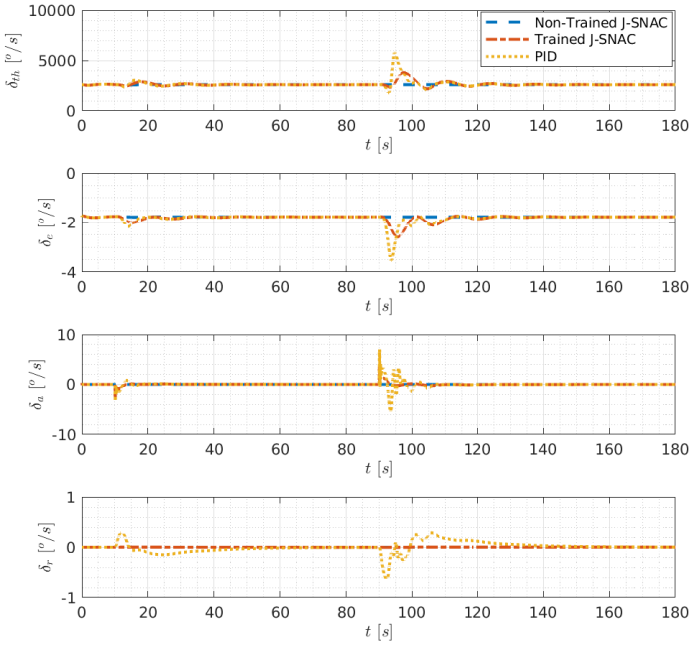


Figure 6-11: Actuator responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

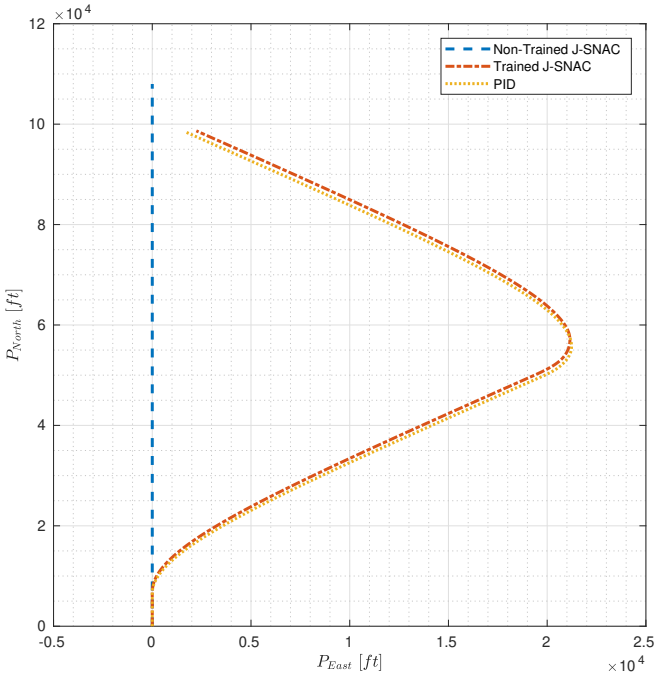


Figure 6-12: Ground track produced by the aircraft while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

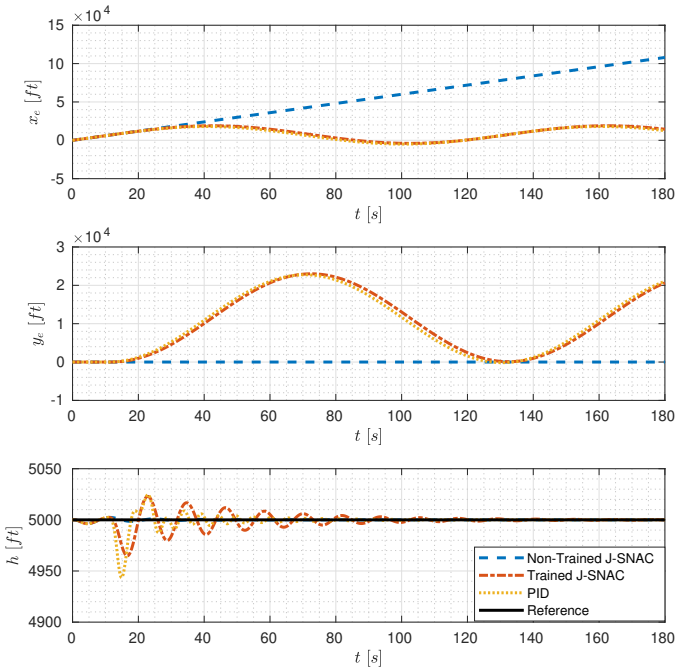


Figure 6-13: Altitude and position responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

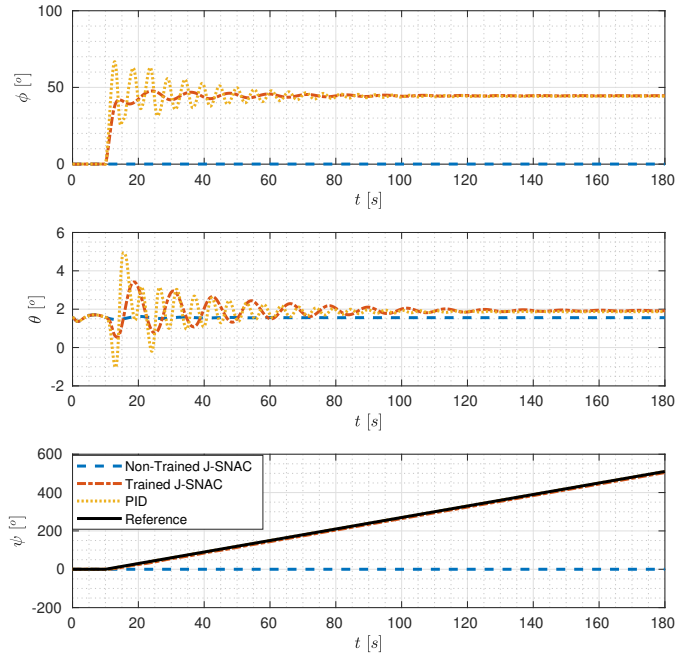


Figure 6-14: Heading and attitude angle responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

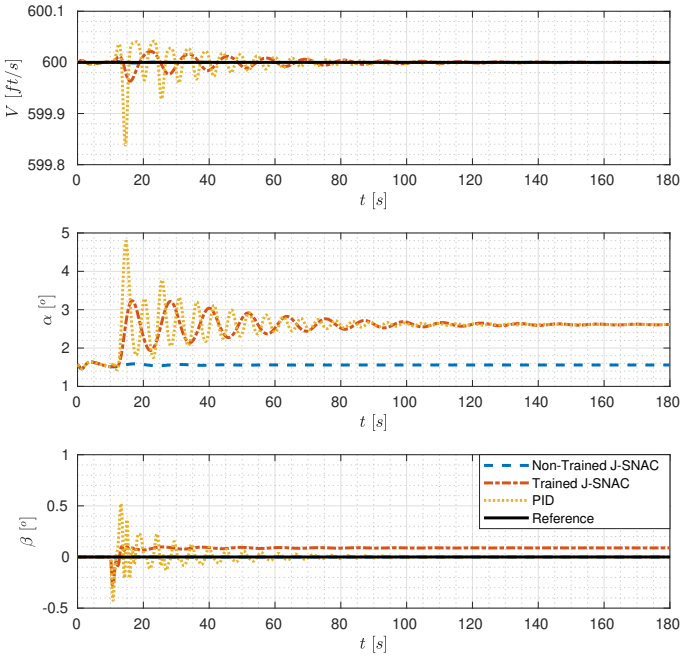


Figure 6-15: Airspeed and aerodynamic angles responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

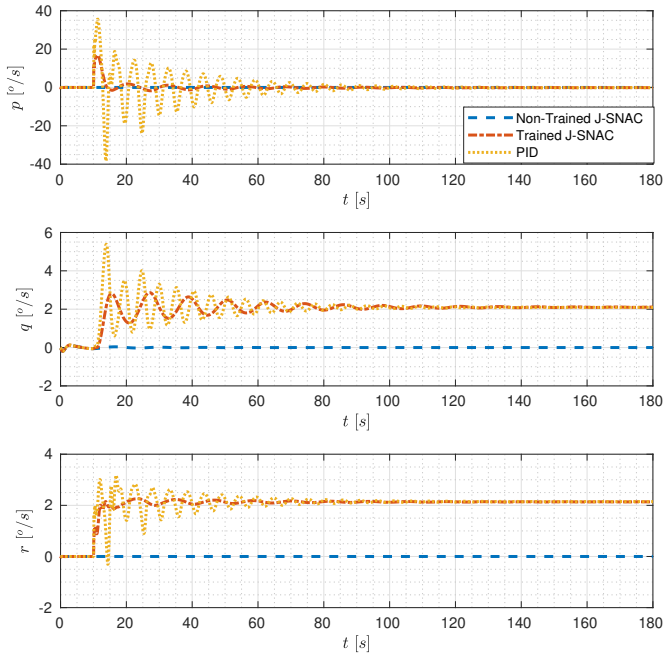


Figure 6-16: Angular rate responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

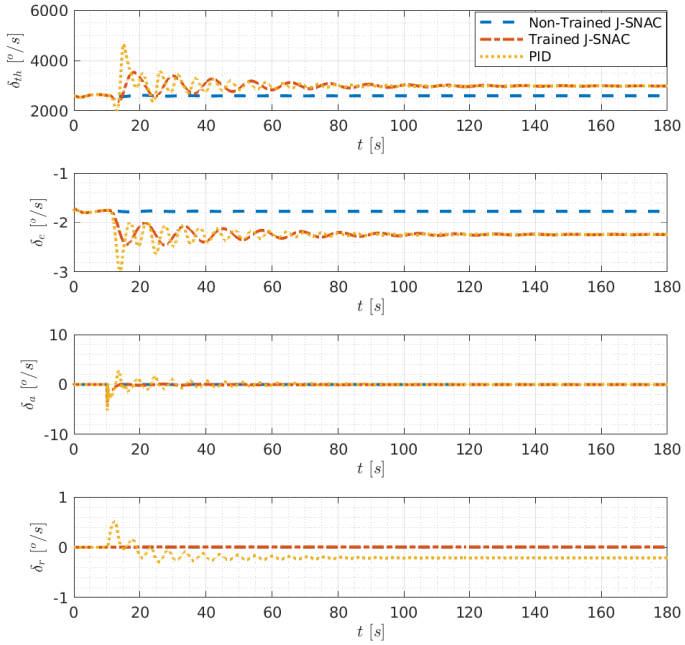


Figure 6-17: Actuator responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

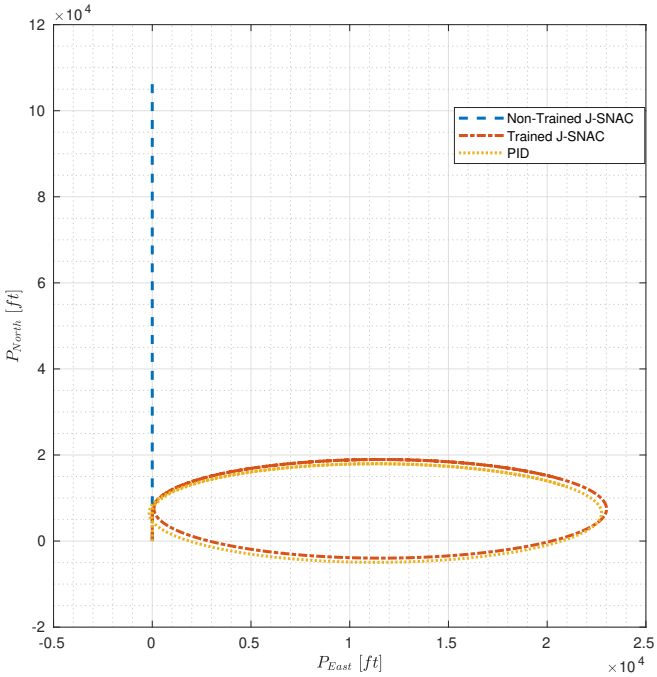


Figure 6-18: Ground track produced by the aircraft while tracking smoothened ramp reference signal with PID, untrained, and trained J-SNAC controllers.

6-3-4 Tacking Of A Sinusoid Reference Signal With Sensor Noise

The fourth evaluative was tracking the sinusoid reference signal, while the angular rate measurement was corrupted with noise. This signal is chosen to evaluate the robustness of the learned control policy to sensor noise. Figure 7-7 to 7-12 shows the state trajectories of the aircraft with PID and the trained J-SNAC controller for this tracking task. One of the main differences in the state responses in this scenario is that the PID controller makes aggressive maneuvers in the presence of the noise, whereas the J-SNAC controller does not. Another difference is in the command signals sent by these controllers. The PID control law sends smoother commands to the aileron actuators in comparison to the J-SNAC controller.

As per the Performance Index, the PID and the trained J-SNAC controller has the score of -0.2719, and -1.5917 respectively.

6-3-5 Tacking Of A Sinusoid Reference Signal With Aileron Handover

The fifth evaluative task was to track the sinusoid reference signal, while the aircraft incurs an aileron hard-over. Aileron hard-over was simulated by dividing the command signal with 2 and then adding 7 degrees bias to the command signal. The hardover is set to onset at $t = 25$ s. Figure 7-25 to 7-30 shows the state trajectories of the aircraft with PID and the trained J-SNAC controller for this scenario. As expected, the PID controller fails to track the reference signal after a few seconds of failure and eventually crash the aircraft. However,

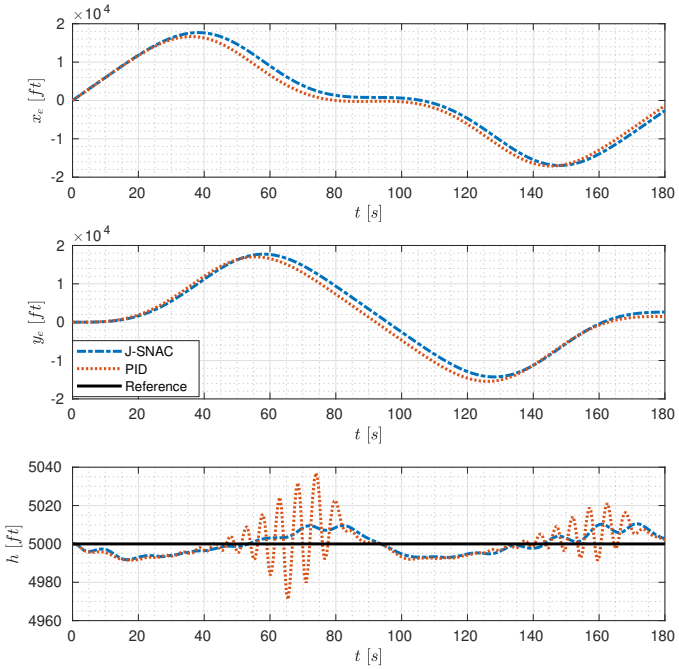


Figure 6-19: Effect of sensor noise on the altitude and position responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

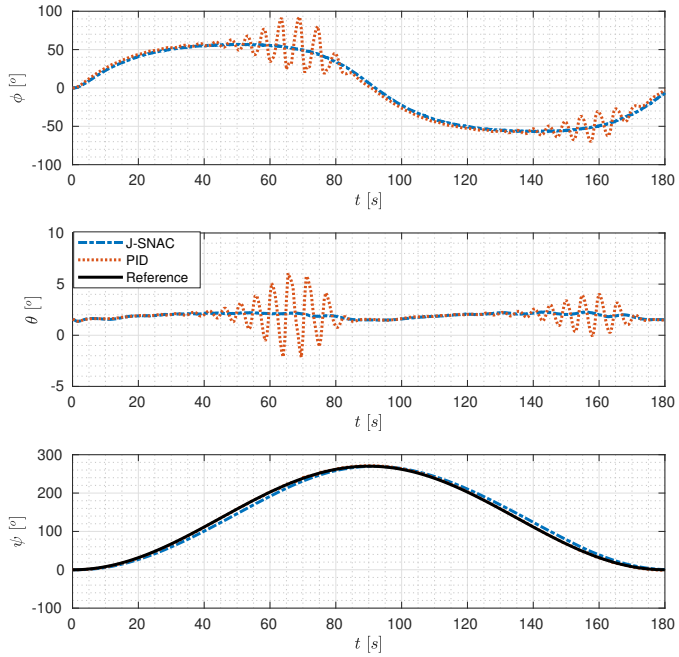


Figure 6-20: Effect of sensor noise on the heading and attitude angle responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

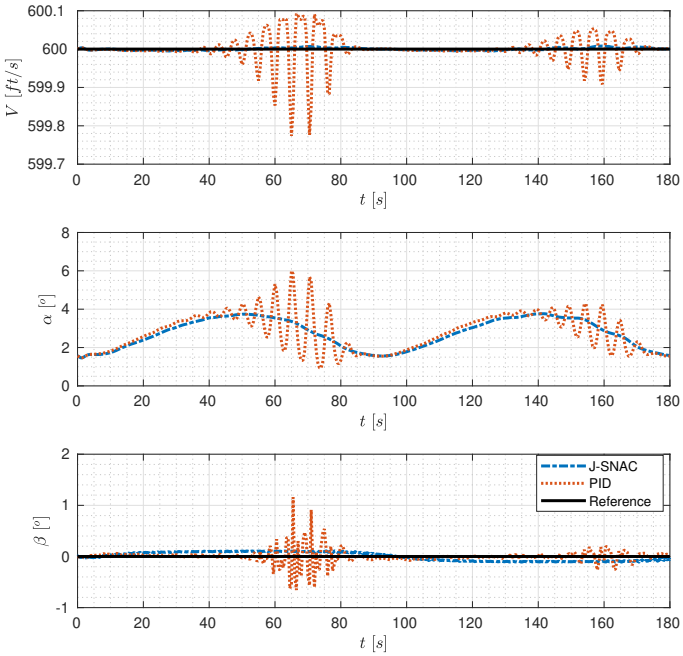


Figure 6-21: Effect of sensor noise on the airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

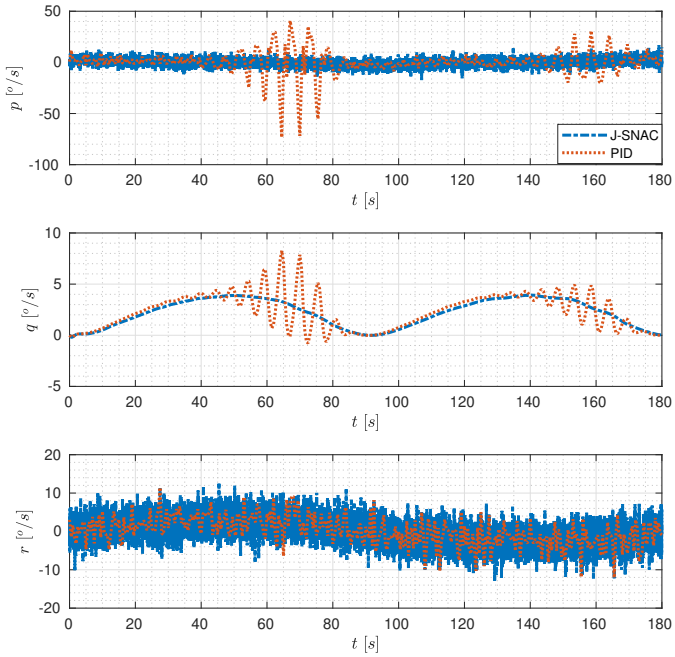


Figure 6-22: Effect of sensor noise on the angular rate responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

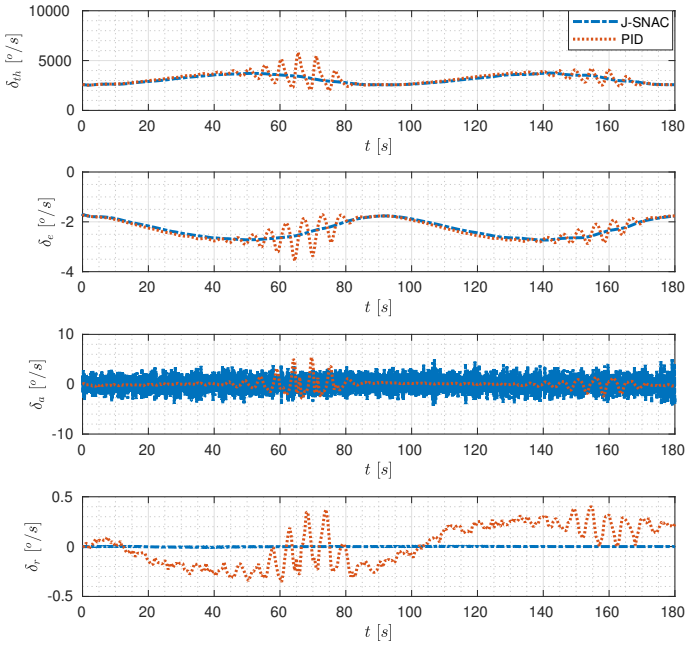


Figure 6-23: Effect of sensor noise on the actuator responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

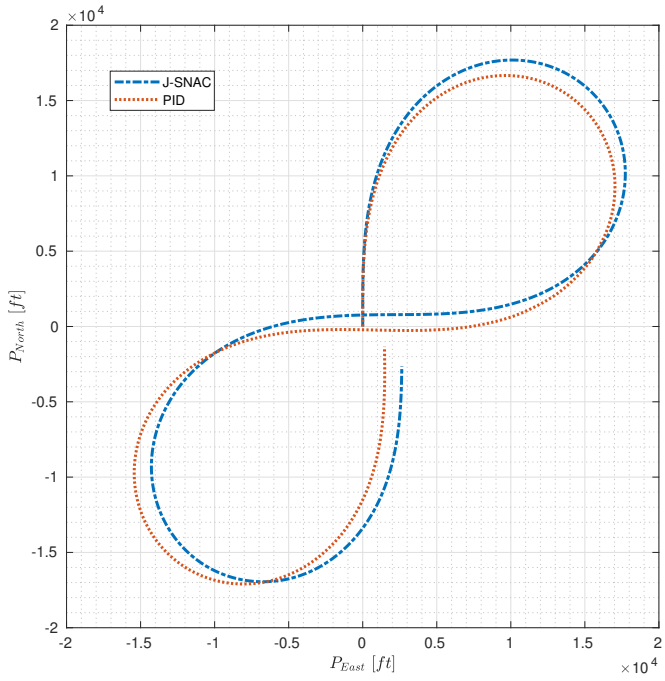


Figure 6-24: Effect of sensor noise on the ground track while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

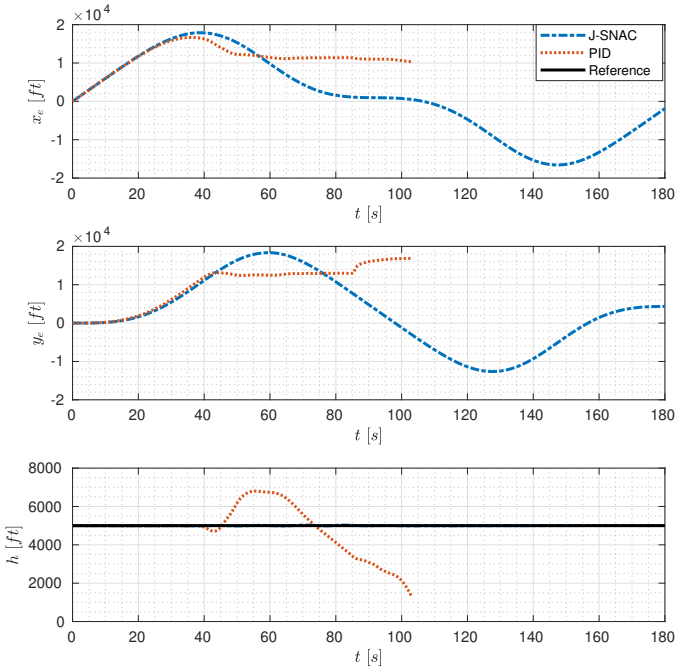


Figure 6-25: Effect of aileron hardover on the altitude and position responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

the J-SNAC controller provides a smooth tracking performance. The continuous tracking by J-SNAC is due to the immediate identification of the reduced control effectiveness and instantaneous adaptation of the control law according to this new control effectiveness.

As per the Performance Index, the PID and the trained J-SNAC controller has the score of -92.9344, and -1.7953 respectively. The lower score of PID controller is attributed to its failure to adapt.

6-3-6 Tacking Of A Sinusoid Reference Signal With Partial Rudder Failure

The sixth and the last evaluative task was to track the sinusoid reference signal, while the aircraft incurs partial failure on the rudder. This failure was simulated by multiplying the rudder command signals with 0.1 and then adding 5 degrees bias to this reduced command signal. The failure is also set to onset at $t = 25$ s. Figure 7-31 to 7-36 shows the state trajectories of the aircraft with PID and the trained J-SNAC controller for this scenario. Similar to the aileron hardover scenario, the PID controller fails to track the reference signal after a few seconds of failure and eventually crash the aircraft. Whereas J-SNAC controller can track the command signal but with a small bias after the failure. Again, the immediate identification of the reduced control effectiveness and instantaneous adaptation of the control law according to this new control effectiveness is the main reason for this adaptability.

As per the Performance Index the trained J-SNAC controller has of -5.0370 and PID controller has a large negative score. The lower score of PID controller is again attributed to the non-

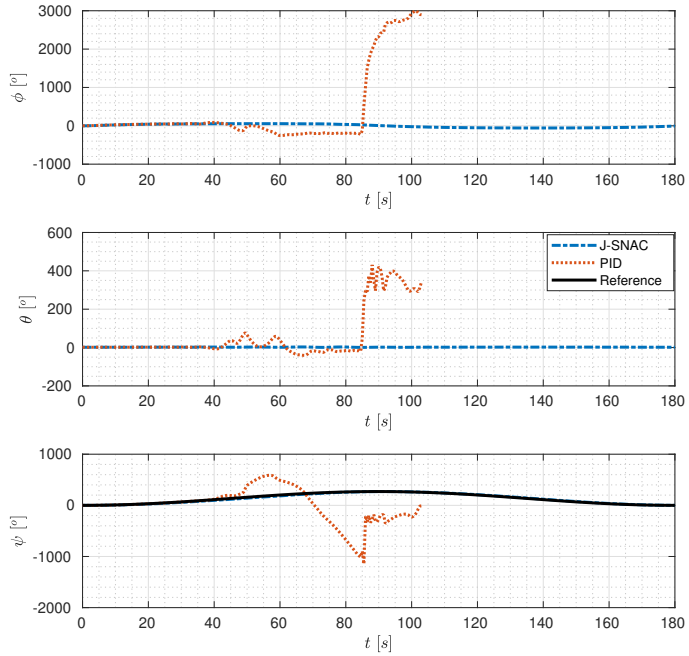


Figure 6-26: Effect of aileron hardover on the heading and attitude angle responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

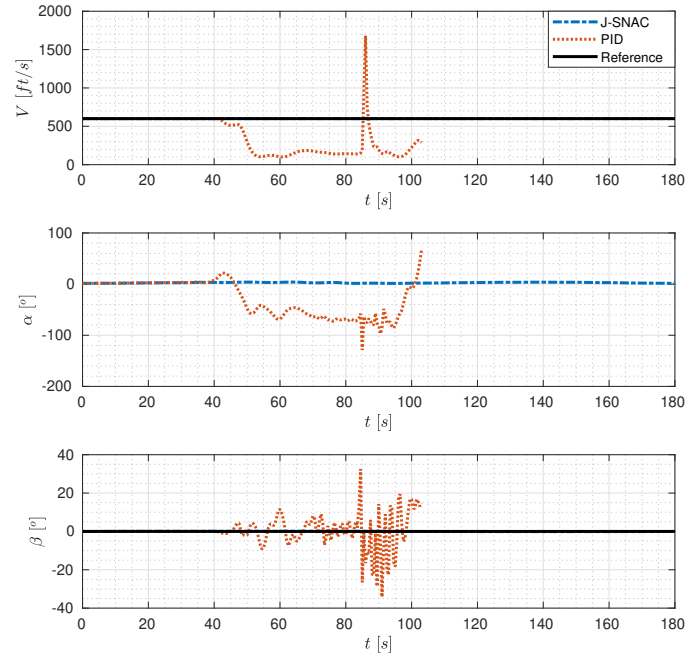


Figure 6-27: Effect of aileron hardover on the airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

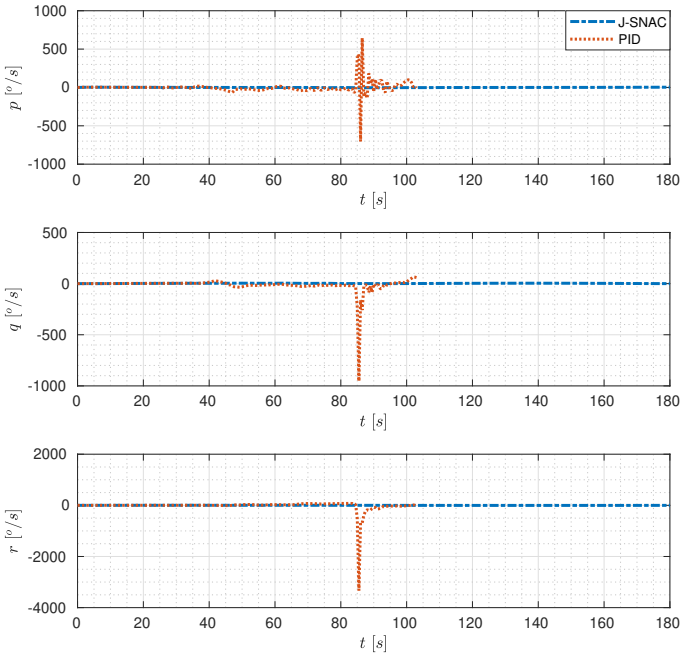


Figure 6-28: Effect of aileron hardover on the angular rate responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

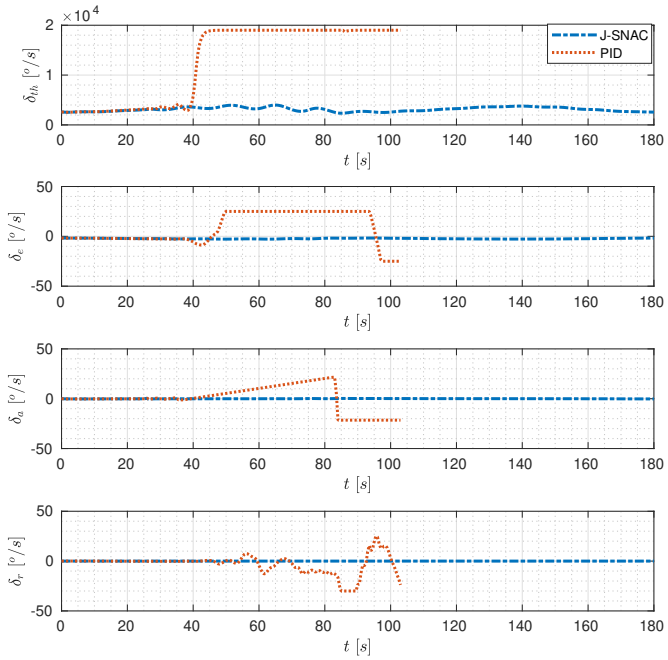


Figure 6-29: Effect of aileron hardover on the actuator responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

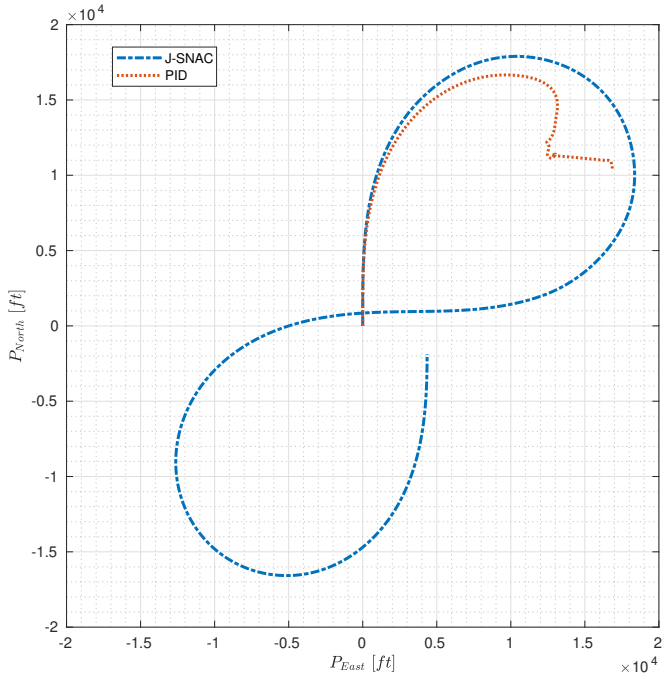


Figure 6-30: Effect of aileron hardover on the ground track while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

adaptive property.

6-4 On Verification And Validation Of The Proposed Flight Controller

One of the goals of this research project was to validate the J-SNAC algorithm for Flight Control System design. Results obtained from the training and control performance evaluation process have proven that this algorithm can learn control laws to drive lateral-directional state variables of F16 aircraft. Furthermore, the adaptability of the proposed controller was also verified by evaluating its performance under unanticipated conditions like sensor noise, aileron hardover and partial failure of the rudder. However in-order to generalize observed results and eventually implement the controller in a flight control system, further verification and validation studies are required. Below are some guidelines to verify and validate the proposed flight controller as an intelligent flight controller:

1. Verify the control performance for same tasks but in a different point of the flight envelope.
2. Verify the control performance for other possible faults.
3. Verify the optimality of the control policy by performing Monte-Carlo Simulation.

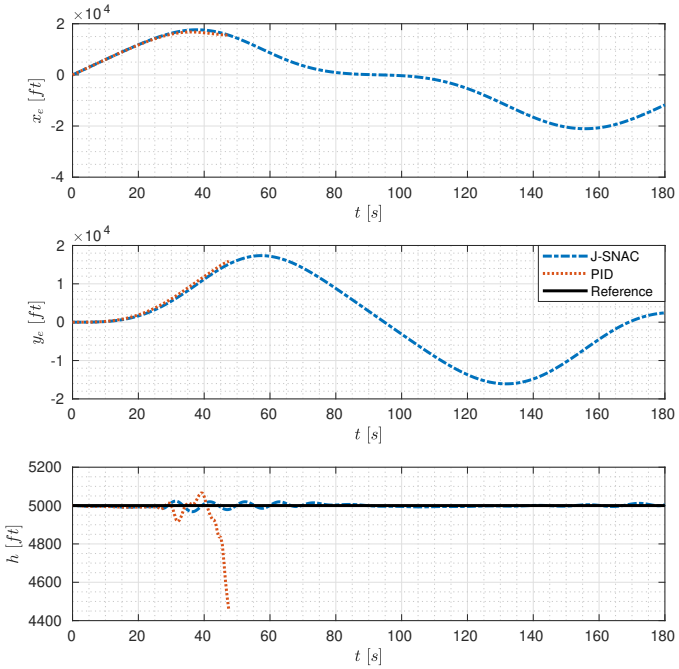


Figure 6-31: Effect of partial rudder failure on the altitude and position responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

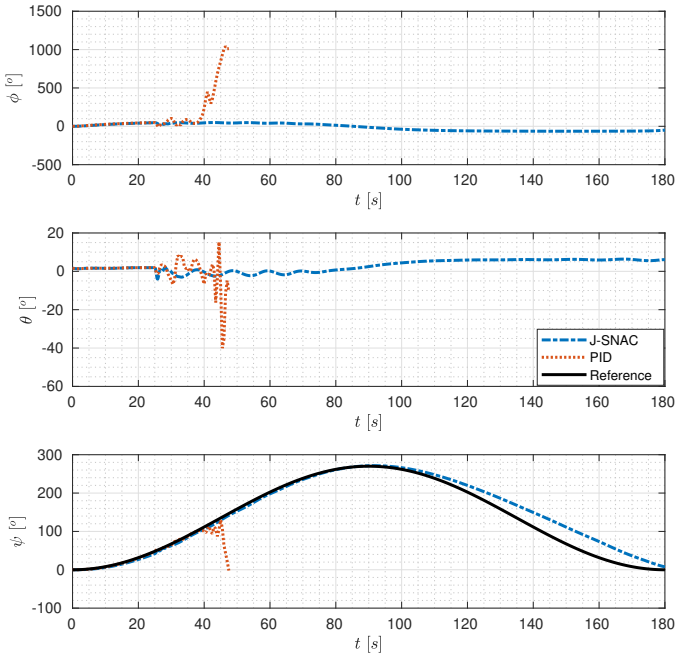


Figure 6-32: Effect of partial rudder failure on the heading and attitude angle responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

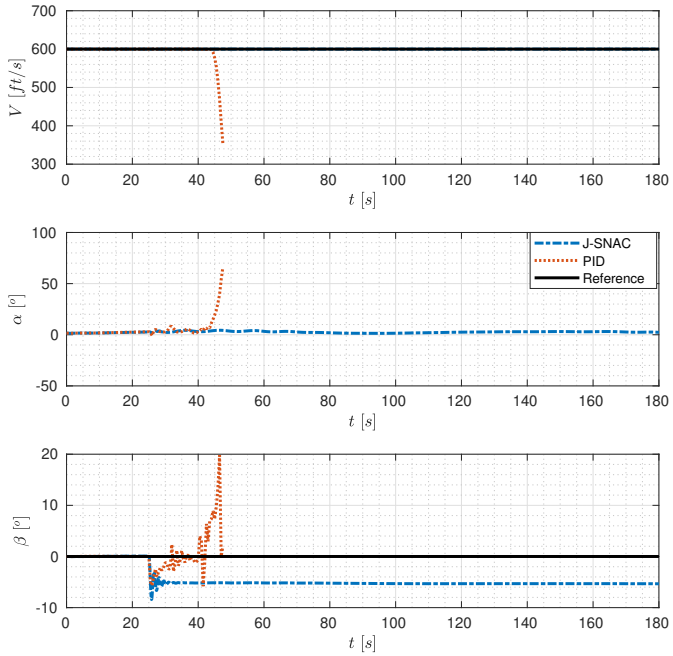


Figure 6-33: Effect of partial rudder failure on the airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

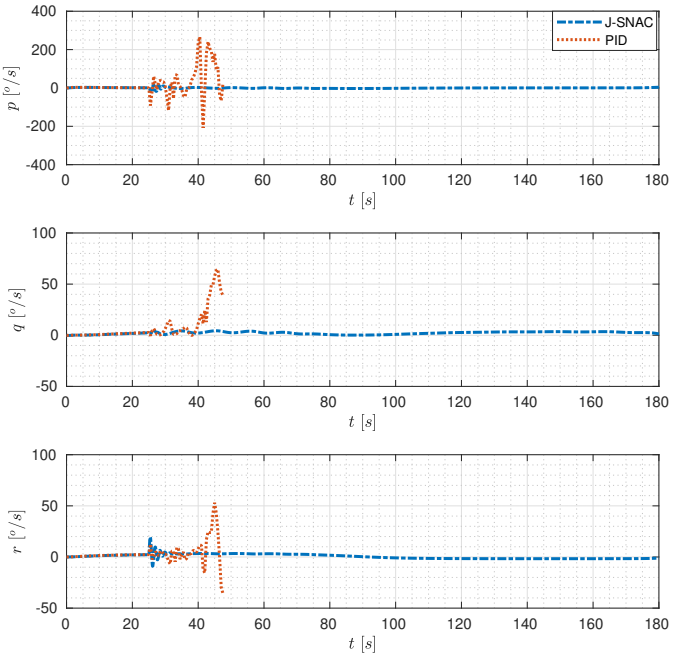


Figure 6-34: Effect of partial rudder failure on the angular rate responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

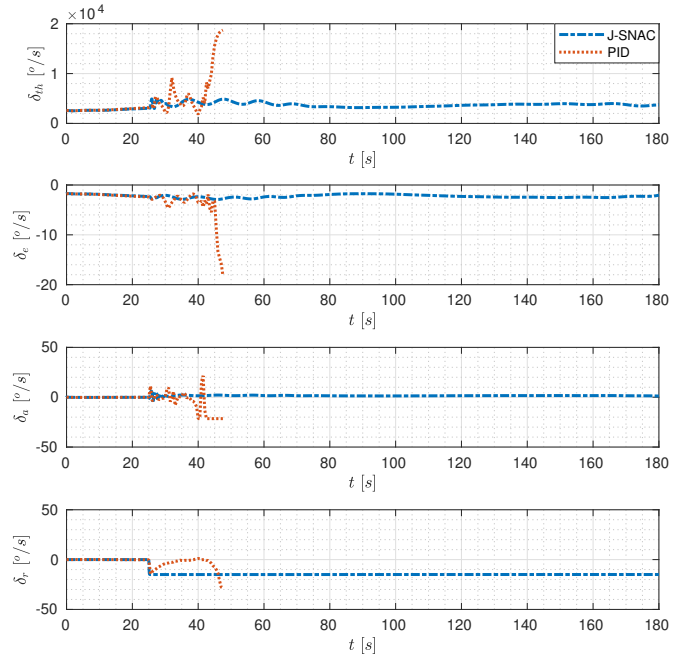


Figure 6-35: Effect of partial rudder failure on the actuator responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

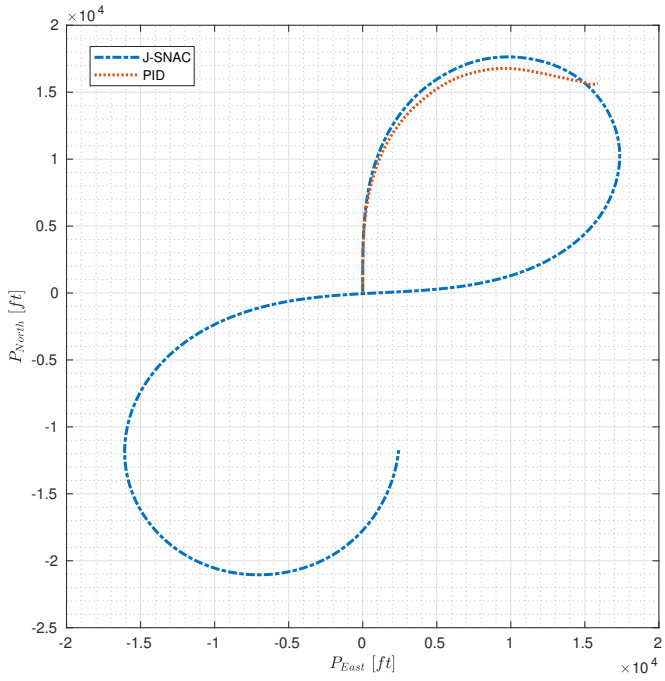


Figure 6-36: Effect of partial rudder failure on the ground track while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

4. Verify the benefits of using J-SNAC by comparing its performance with other ACD algorithms specifically for the flight controller design.
5. Verify the stability of the learning process by performing extended training.
6. Validate the control performance for other fixed-wing aircraft with simulation studies.
7. Validate the controller by implementing the controller in small-scale aircraft.

6-5 Conclusion

This chapter has presented discussions on the structure and hyper-parameters for the proposed flight controller. Then it has presented and discussed results from six control performance evaluation tasks. Additionally discussed on the verification and validation of the proposed flight controller. The goal of this chapter was given additional information on the proposed flight controller and to answer the research questions **R.Q.3.1: How does the proposed controller perform in comparison to a traditional fixed-gain linear controller?**, **R.Q.3.2: How does the proposed controller perform as an adaptive controller?** and **R.Q.3.3: To what extent can the controller performances be generalized for fixed-wing aircraft control?**.

Control performance evaluation showed that under nominal conditions the benchmarking PID controller performs slightly better than J-SNAC flight controller. This better performance is because of better side-slip attenuation property and faster response of the PID controller. However, when there are parametric failures in the aircraft, J-SNAC flight controller outperforms the PID controller by adapting the in-place control law. As per these results, J-SNAC is a viable algorithm for flight control design, however, to generalize the performance for fixed-wing aircraft, verification and validation studies are required to be performed.

Chapter 7

Detailed Control Performances Of The J-SNAC Flight Controller

The Performance Index (PI), defined in Equation 5-13, is used to quantify control performances. Detail explanation of the PI is given in the article.

7-1 Tacking Of A Sinusoid Reference Signal Under Ideal Conditions

The first evaluative task was to track sinusoid reference signal. Figure 7-1 to 7-6 shows the state trajectories of the aircraft with PID, untrained and trained J-SNAC controller. Visually, it can be seen that the untrained controller could hardly follow the command signal and both PID and trained J-SNAC controller have almost identical response. However, there are two notable difference between the PID and trained J-SNAC controller responses. The first difference is that The PID law produce better side slip attenuation. The second difference is that the J-SNAC controller has a delay on tracking when compared to the PID controller.

As per the Performance Index the PID, untrained, and trained J-SNAC controller has the score of -0.1839, -52.0257, and -1.5802 respectively. The higher score of PID controller can be directly attributed to better slip attenuation and faster response.

7-2 Tacking Of A Smoothened Step Reference Signal Under Ideal Conditions

The second evaluative task was to track a smoothened step reference signal. This signal is chosen to verify the control performance on a task that was not in the training program. Figure 7-7 to 7-12 shows the state trajectories of the aircraft with PID, untrained, and trained J-SNAC controller for this tracking task. Similar to the tracking of the sinusoid, non-trained

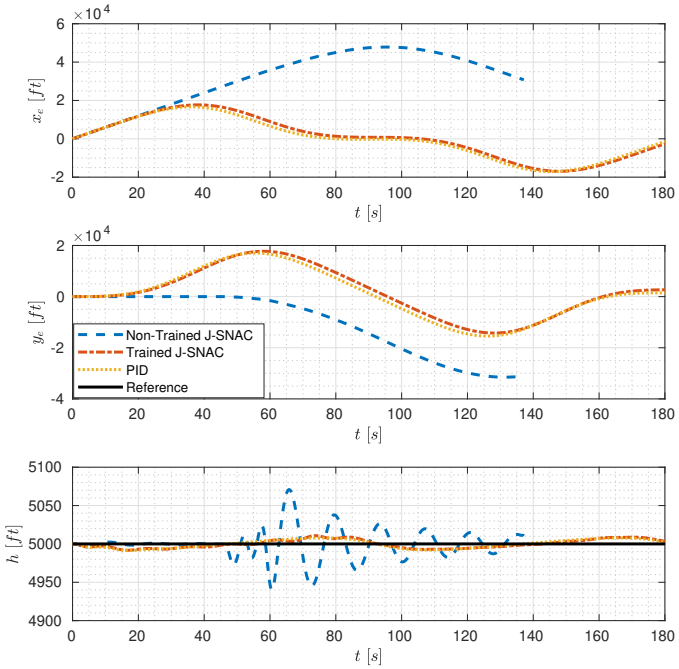


Figure 7-1: Altitude and position responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

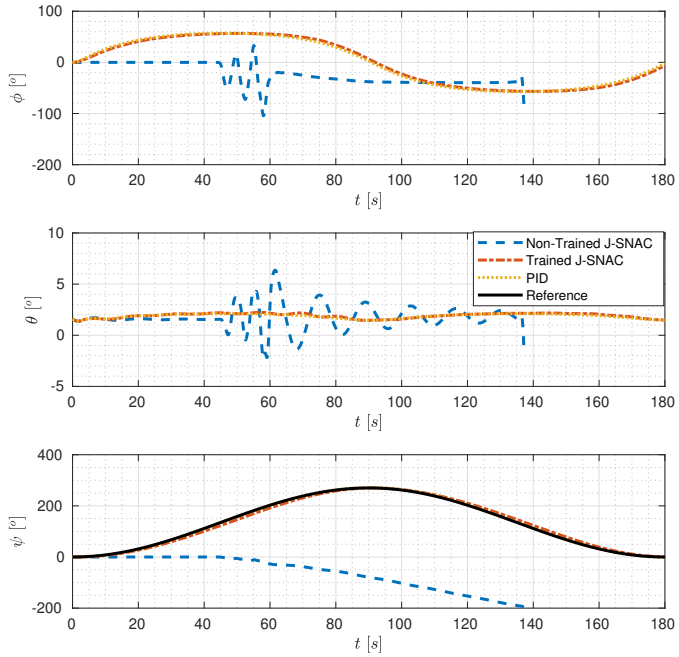


Figure 7-2: Heading and attitude angle responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

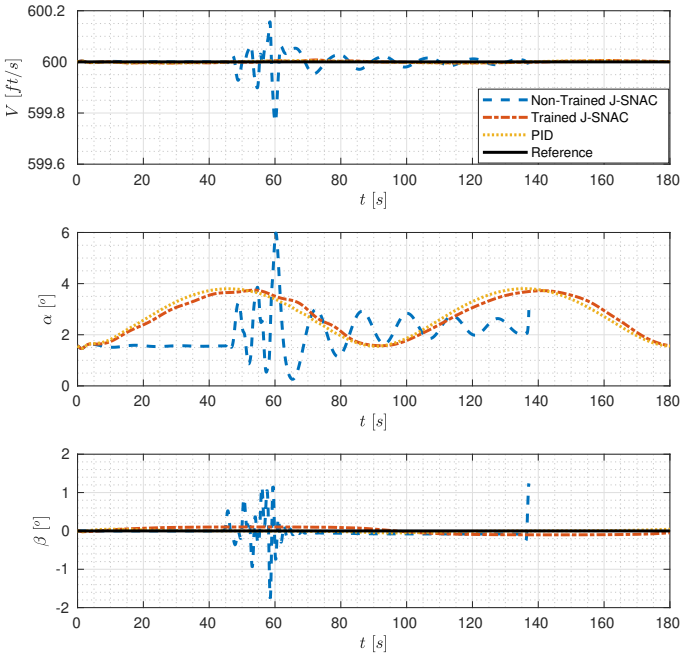


Figure 7-3: Airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.e

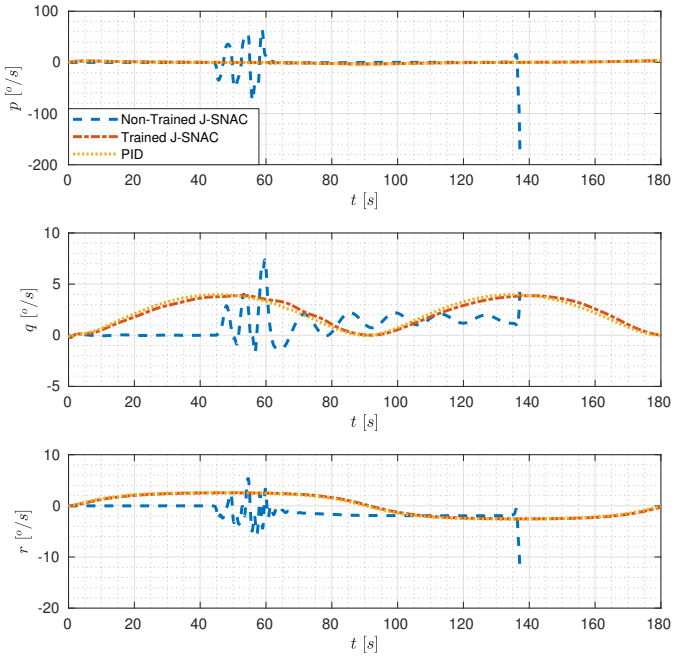


Figure 7-4: Angular rate responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

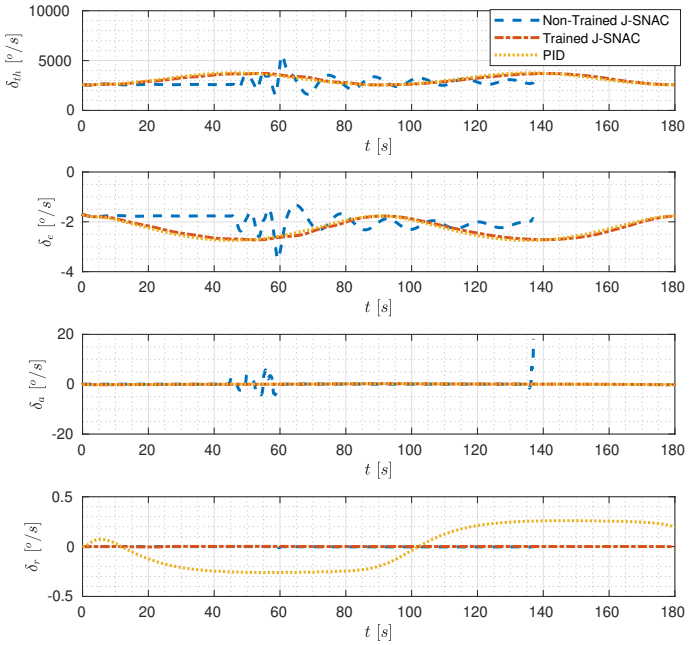


Figure 7-5: Control actuator responses while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

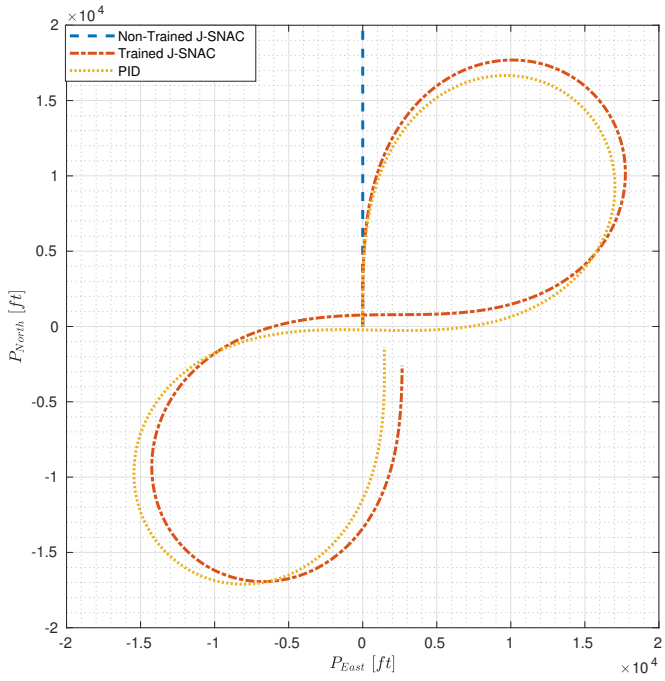


Figure 7-6: Ground track produced by the aircraft while tracking sinusoid reference signal with PID, untrained, and trained J-SNAC controllers.

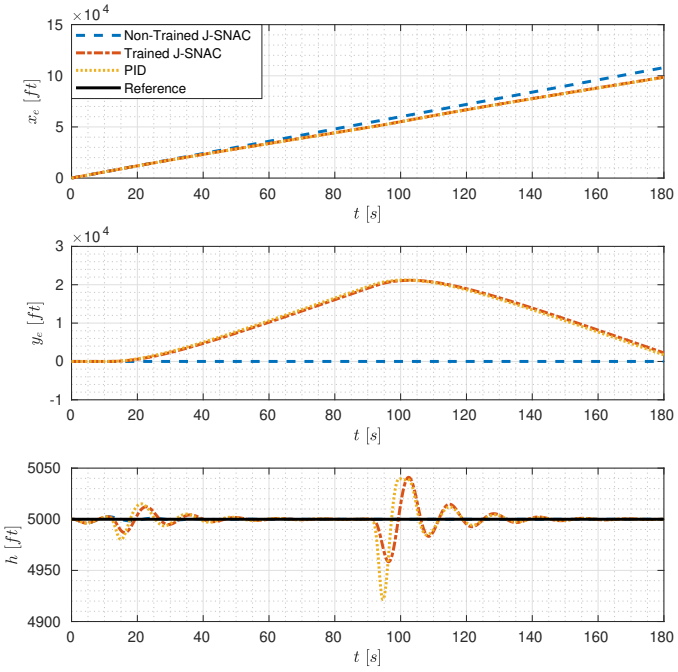


Figure 7-7: Altitude and position responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

J-SNAC controller failed to perform the tracking while the trained J-SNAC and PID controller performs almost similarly. 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.

As per the Performance Index the PID, untrained, and trained J-SNAC controller has the score of -0.1623, -4.8565, and -0.3440 respectively. The higher score for PID controller is still due to better slip attenuation and faster response.

7-3 Tacking Of A Ramp Signal Under Ideal Conditions

The third evaluative task was tracking of a ramp signal. This signal is also chosen to verify the control performance on a task that was not in the training program. Figure 7-13 to 7-18 shows the state trajectories of the aircraft with PID, untrained, and trained J-SNAC controller for this tracking task. In these figures it can bee seen that PID controller creates more oscillations in the state responses. This is because it commands higher aerodynamic surface deflections. Furthermore the J-SNAC controller do not compensate for a small side-slip angle where as the PID controller attenuates the side-slip angle over time.

As per the Performance Index the PID, untrained, and trained J-SNAC controller has the score of -0.1261, -57.2332, and -1.1436 respectively.

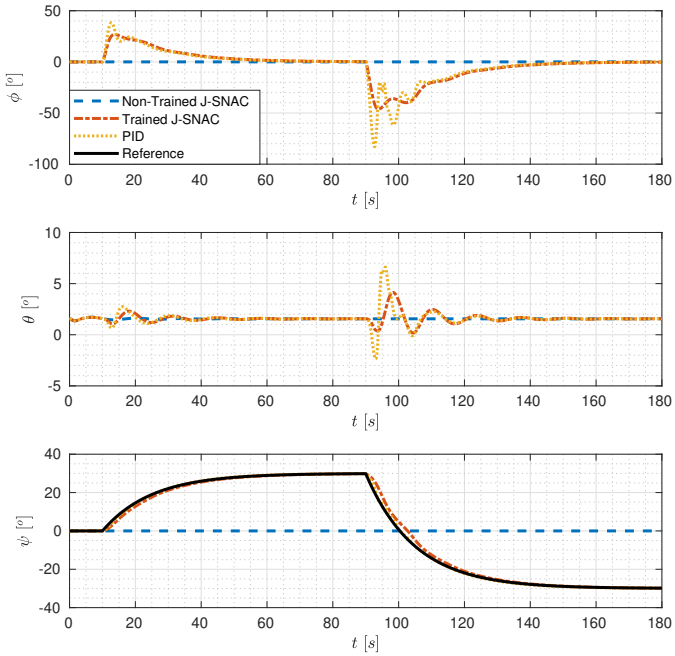


Figure 7-8: Heading and attitude angle responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

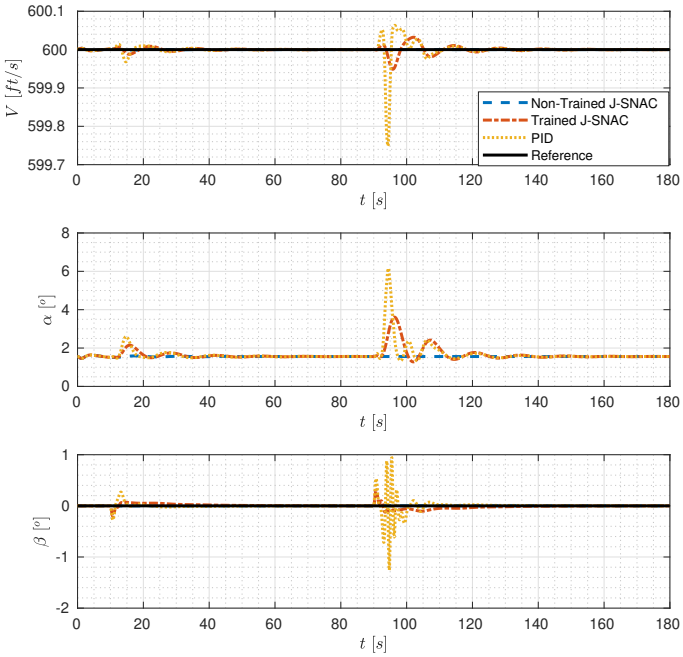


Figure 7-9: Airspeed and aerodynamic angles responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

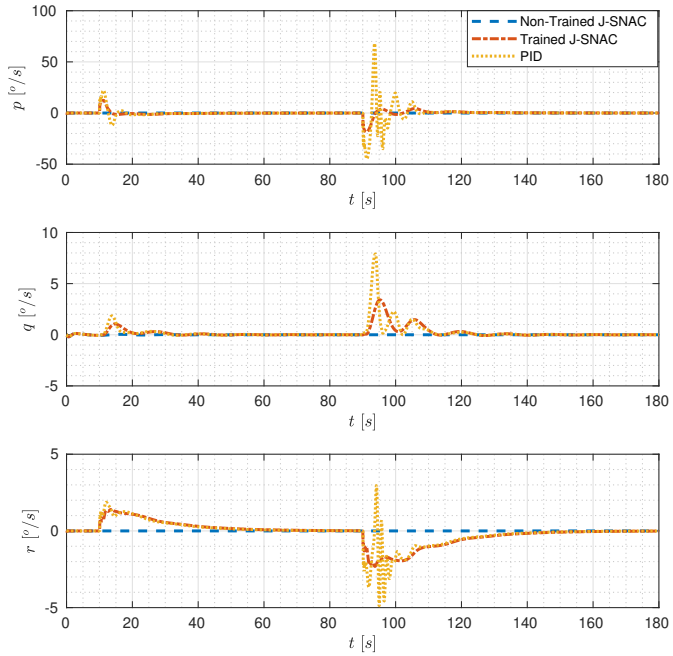


Figure 7-10: Angular rate responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

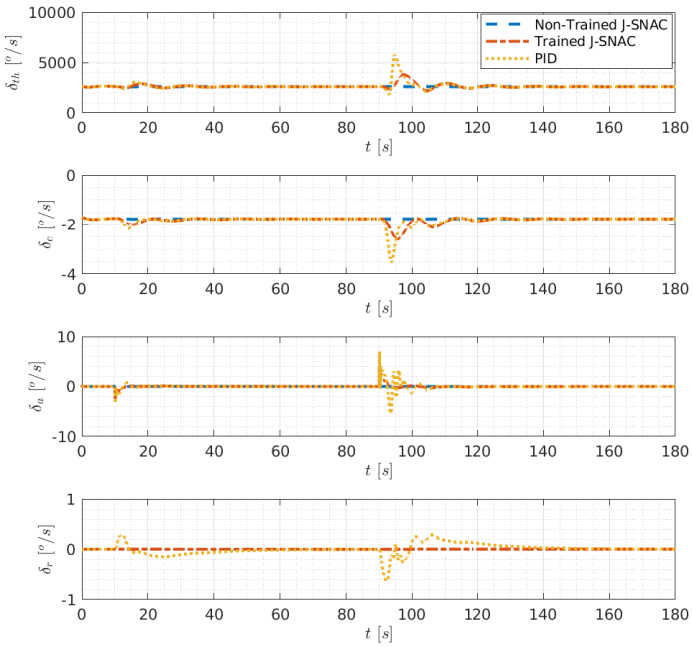


Figure 7-11: Actuator responses while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

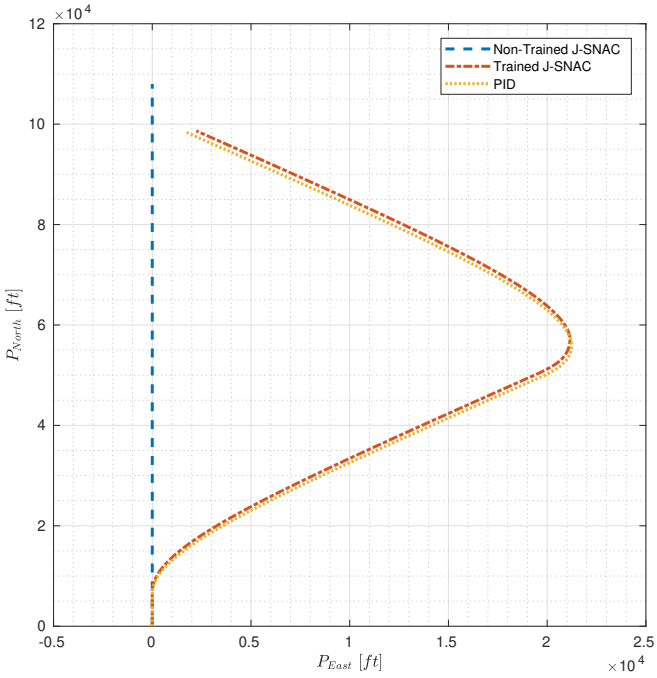


Figure 7-12: Ground track produced by the aircraft while tracking smoothened step reference signal with PID, untrained, and trained J-SNAC controllers.

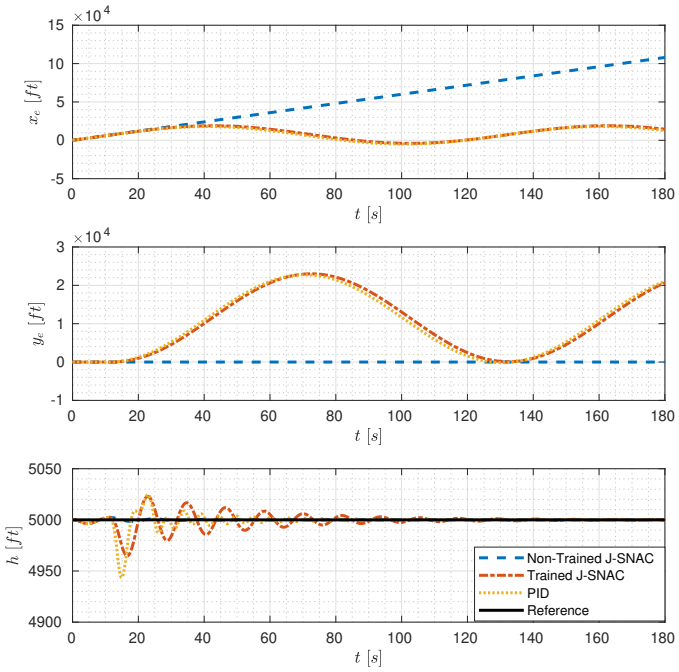


Figure 7-13: Altitude and position responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

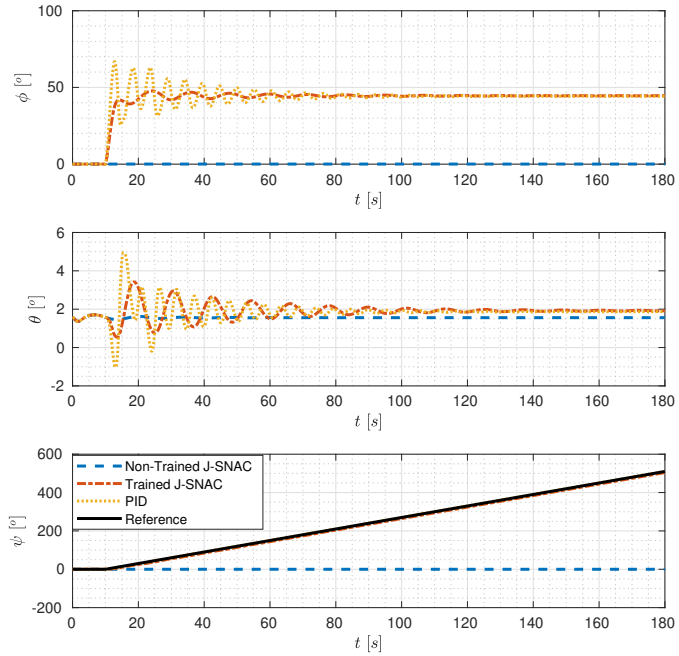


Figure 7-14: Heading and attitude angle responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

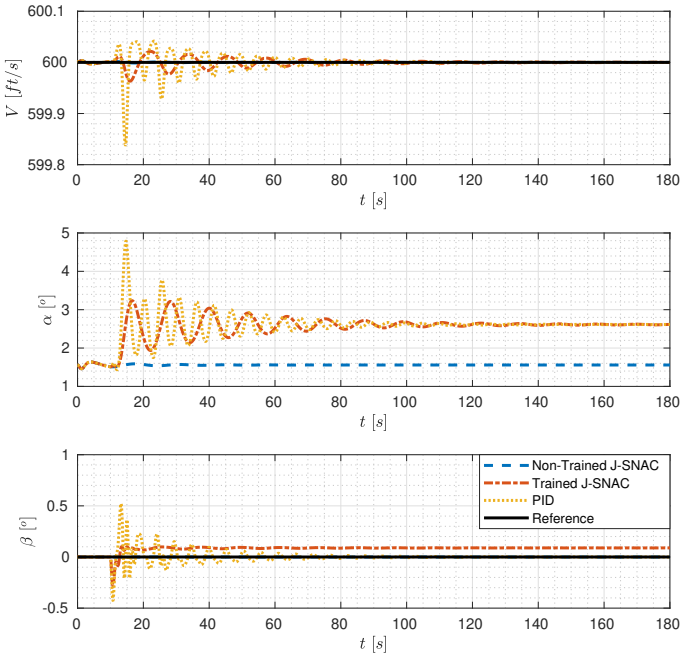


Figure 7-15: Airspeed and aerodynamic angles responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

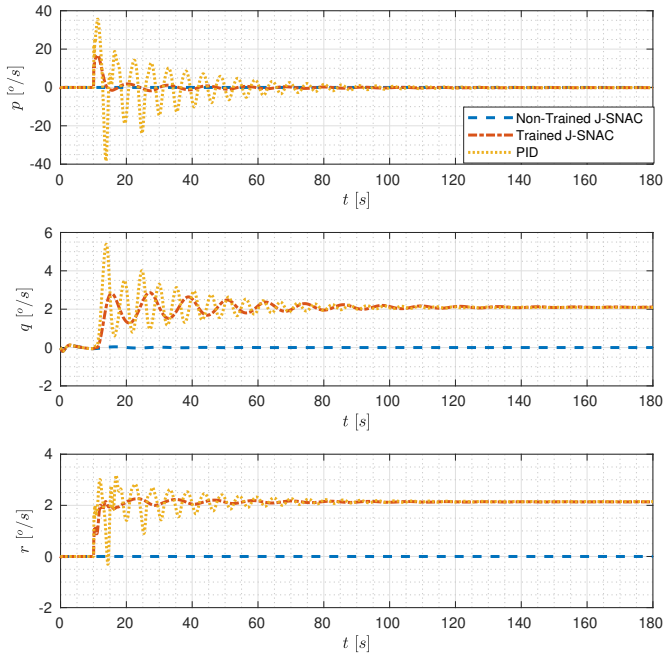


Figure 7-16: Angular rate responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

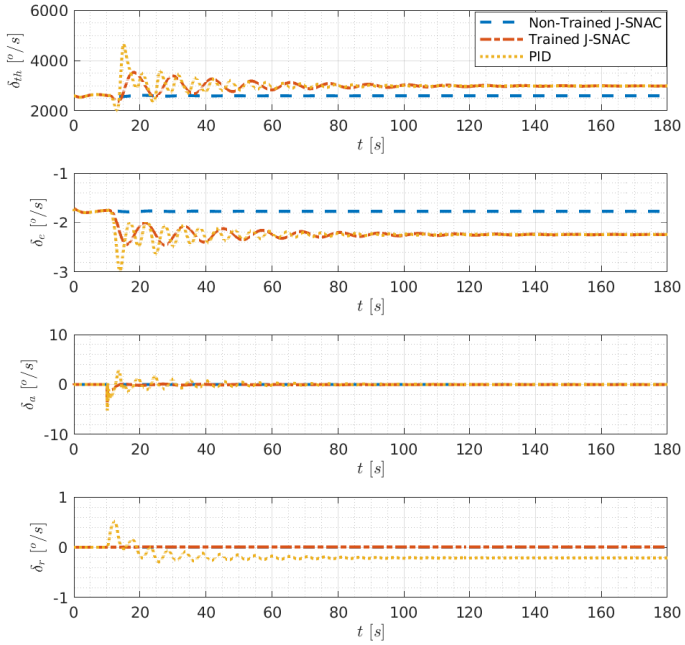


Figure 7-17: Actuator responses while tracking ramp reference signal with PID, untrained, and trained J-SNAC controllers.

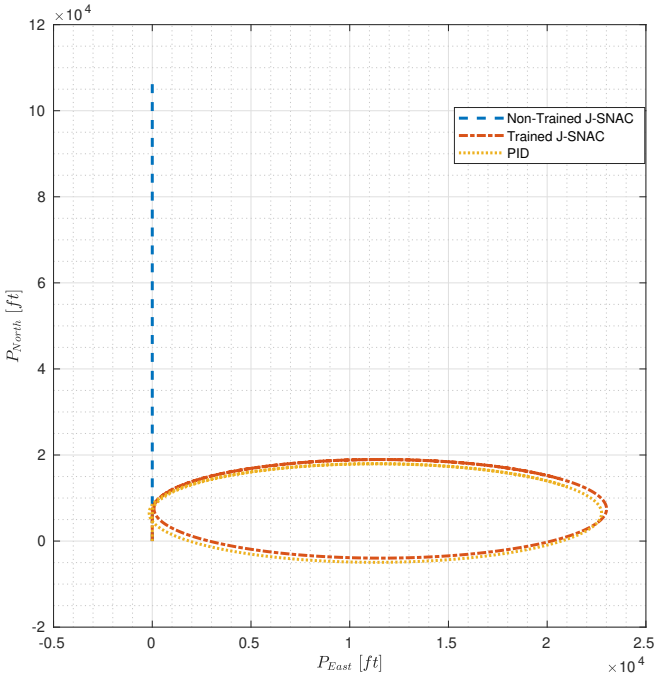


Figure 7-18: Ground track produced by the aircraft while tracking smoothened ramp reference signal with PID, untrained, and trained J-SNAC controllers.

7-4 Tacking Of A Sinusoid Reference Signal With Sensor Noise

The fourth evaluative was tracking the sinusoid reference signal, while the angular rate measurement were corrupted with noise. This signal is chosen to evaluate the robustness of the learned control policy to sensor noise. Figure 7-7 to 7-12 shows the state trajectories of the aircraft with PID, and the trained J-SNAC controller for this tracking task. One of the main differences in the state responses in this scenario is that the PID controller makes aggressive maneuvers in presence of the noise, where as the J-SNAC controller does not. Another difference is in the command signals sent by these controllers. The PID control law sends smoother commands to the aileron actuators in comparison to the J-SNAC controller.

As per the Performance Index the PID, and the trained J-SNAC controller has the score of -0.2719, and -1.5917 respectively.

7-5 Tacking Of A Sinusoid Reference Signal With Aileron Handover

The fifth evaluative task was track the sinusoid reference signal, while the aircraft incurs a aileron hardover. Aileron hardover was simulated by dividing the command signal with 2 and then add 7 degree bias to the command signal. The hardover is set to onset at $t = 25$ s. This scenario is chosen to evaluate the adaptability of the J-SNAC control policy. Figure 7-25 to

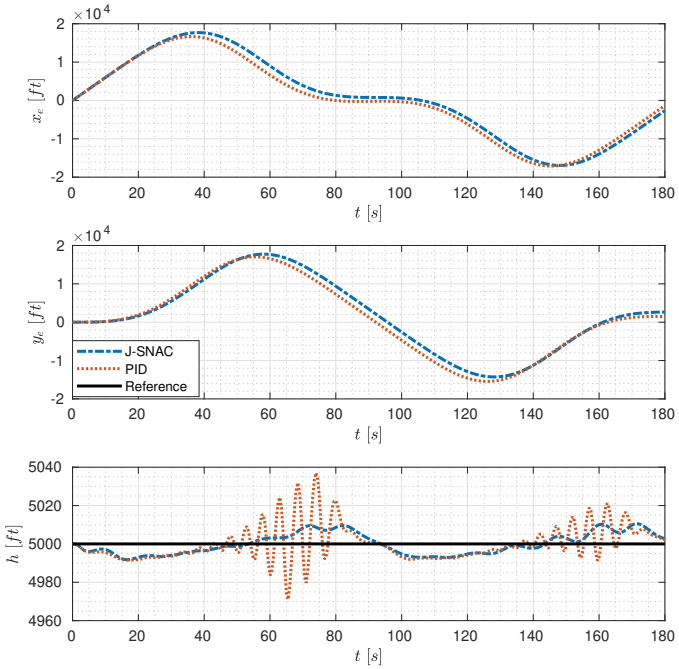


Figure 7-19: Effect of sensor noise on the altitude and position responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

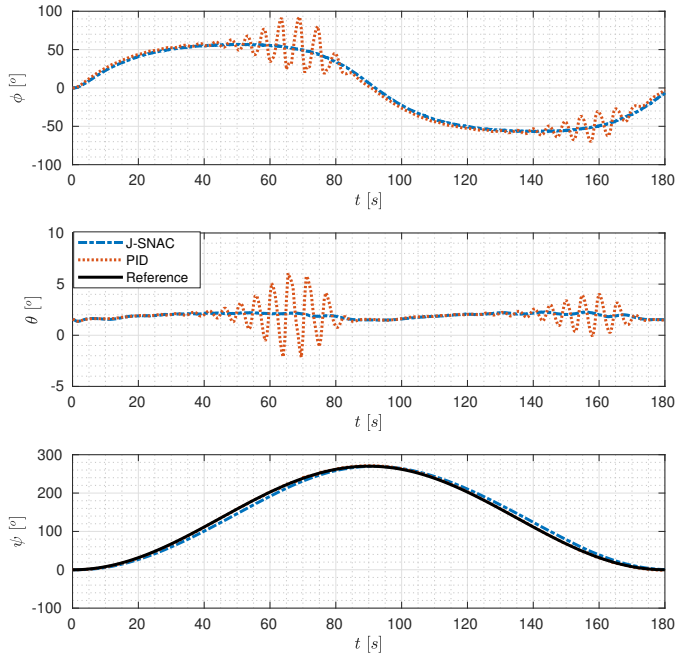


Figure 7-20: Effect of sensor noise on the heading and attitude angle responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

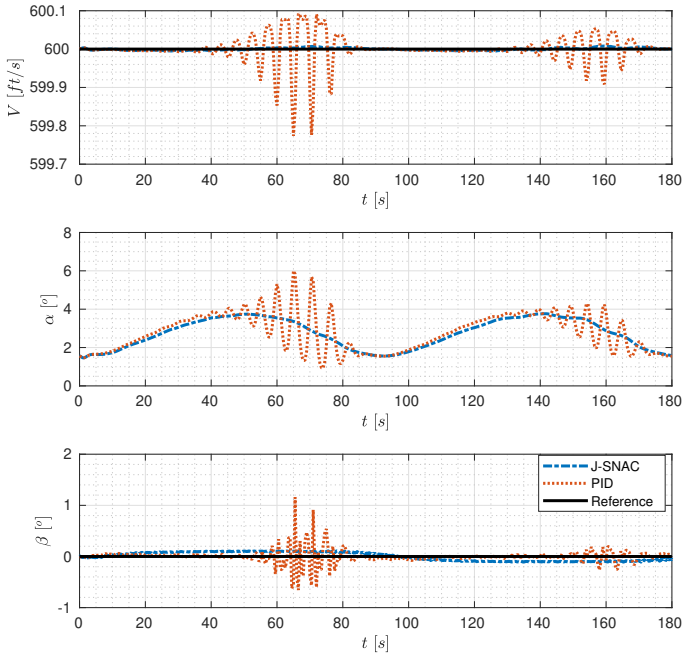


Figure 7-21: Effect of sensor noise on the airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

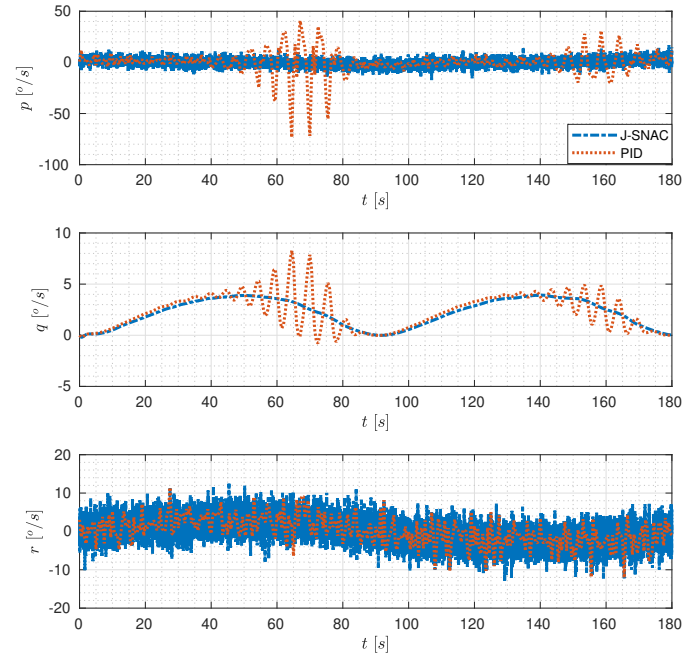


Figure 7-22: Effect of sensor noise on the angular rate responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

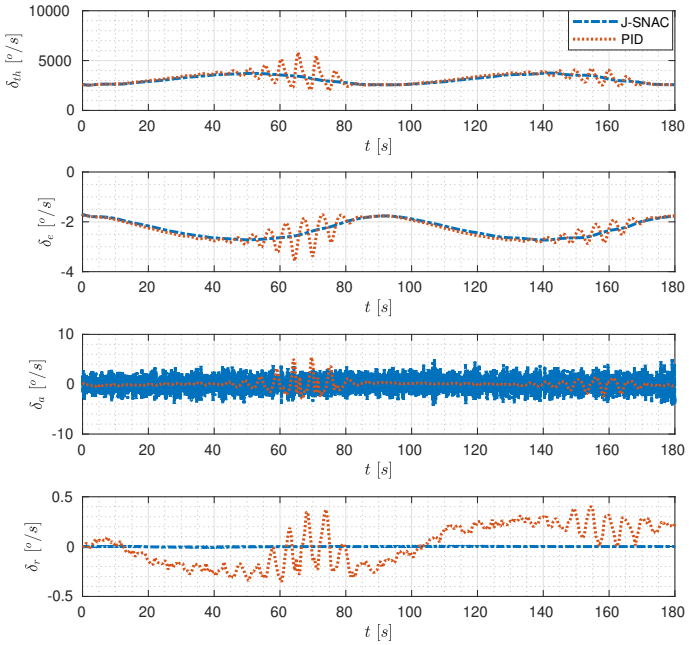


Figure 7-23: Effect of sensor noise on the actuator responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

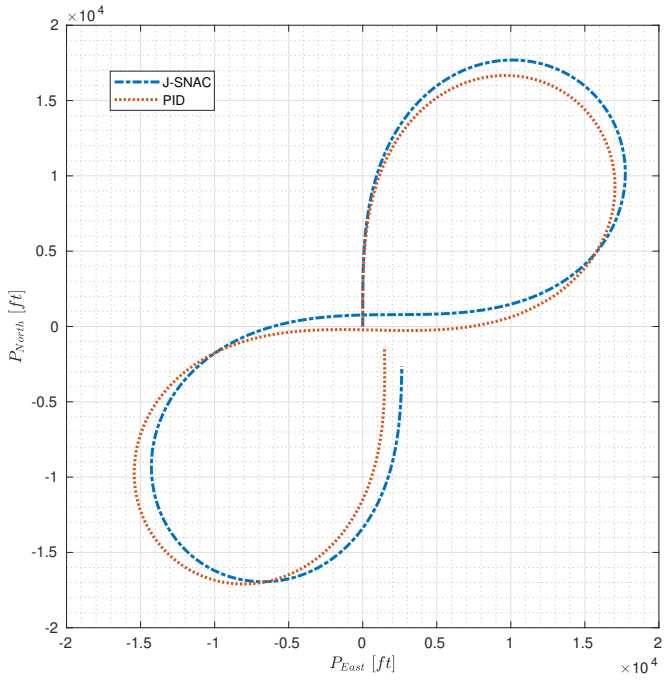


Figure 7-24: Effect of sensor noise on the ground track while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

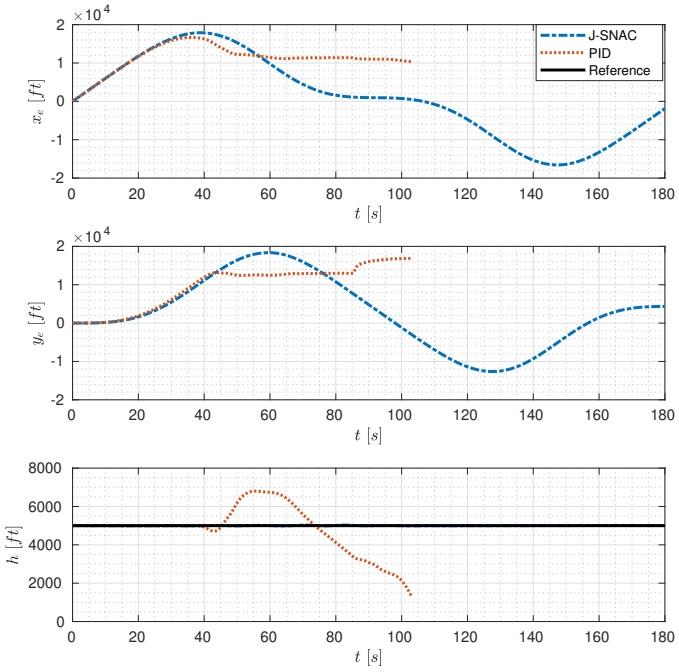


Figure 7-25: Effect of aileron hardover on the altitude and position responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

7-30 shows the state trajectories of the aircraft with PID, and the trained J-SNAC controller for this scenario. As expected, the PID controller fails to track the reference signal few seconds of failure and eventually crash the aircraft. Whereas J-SNAC controller provides an smooth tracking performance. The continuous tracking by J-SNAC is due to the immediate identification of the reduced control effectiveness and instantaneous adaptation of the control law according to this new control effectiveness.

As per the Performance Index the PID, and the trained J-SNAC controller has the score of -92.9344, and -1.7953 respectively. The lower score of PID controller is attributed to its failure to adapt.

7-6 Tacking Of A Sinusoid Reference Signal With Partial Rudder Failure

The sixth and the last evaluative task was track the sinusoid reference signal, while the aircraft incurs partial failure on the rudder. This failure was simulated by multiplying the rudder command signals with 0.1 and then adding 5 degree bias to this reduced command signal. The failure is also set to onset at $t = 25$ s. This scenario is also chosen to evaluate the adaptability of the J-SNAC controller. Figure 7-31 to 7-36 shows the state trajectories of the aircraft with PID, and the trained J-SNAC controller for this scenario. Similar to the aileron hardover scenario, the PID controller fails to track the reference signal few seconds of failure

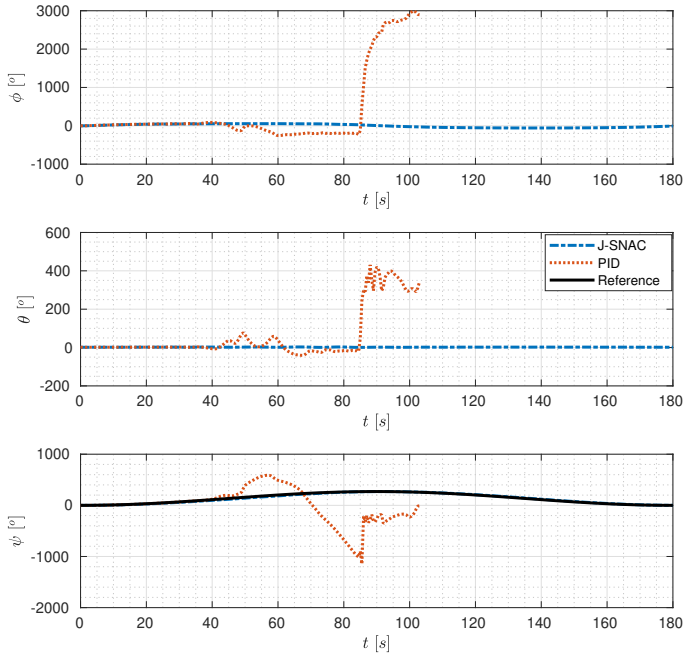


Figure 7-26: Effect of aileron hardover on the heading and attitude angle responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

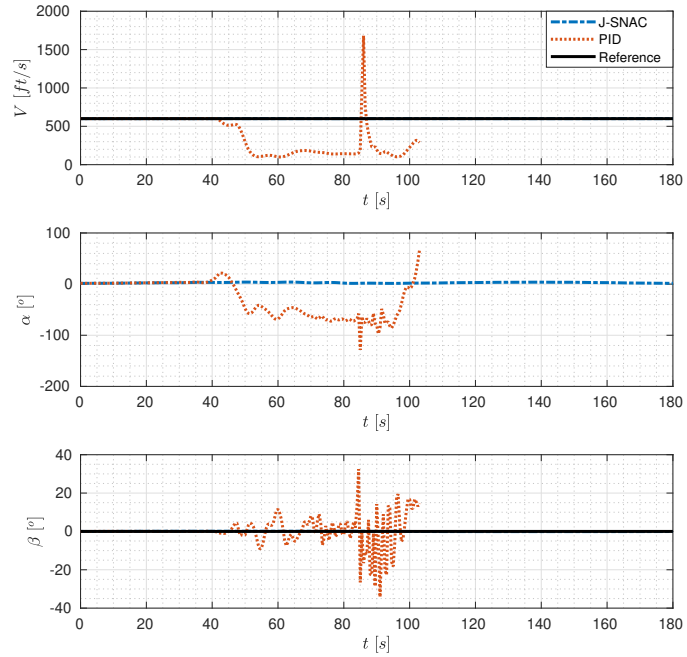


Figure 7-27: Effect of aileron hardover on the airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

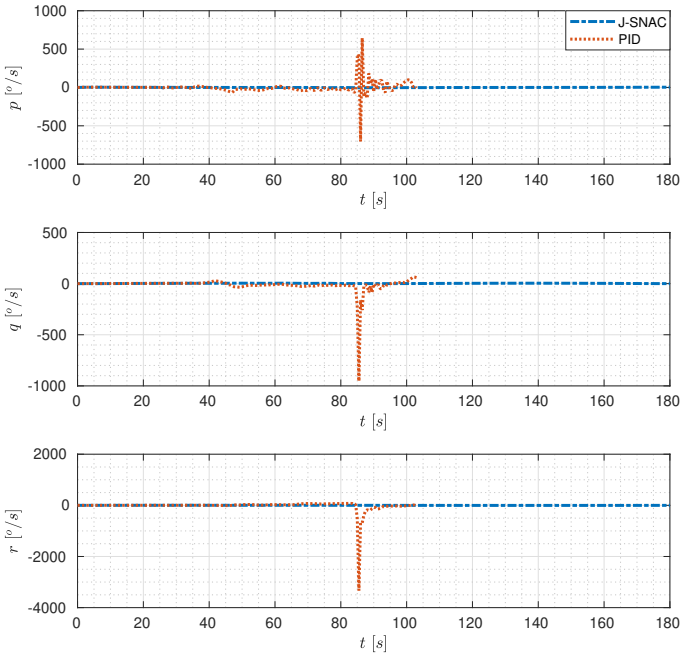


Figure 7-28: Effect of aileron hardover on the angular rate responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

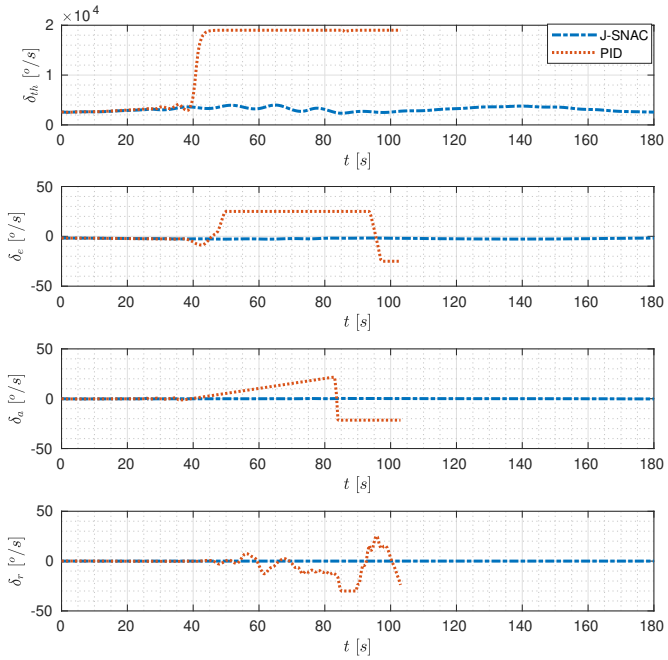


Figure 7-29: Effect of aileron hardover on the actuator responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

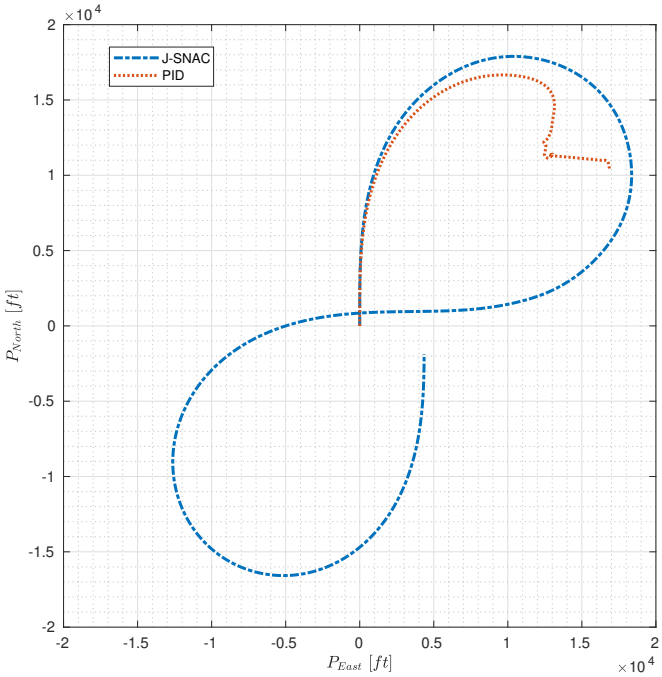


Figure 7-30: Effect of aileron hardover on the ground track while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

and eventually crash the aircraft. Whereas J-SNAC controller can track command signal but with a small bias after the failure. Again, the immediate identification of the reduced control effectiveness and instantaneous adaptation of the control law according to this new control effectiveness is the main reason for this adaptability.

As per the Performance Index the trained J-SNAC controller has of -5.0370 and PID controller produces a very large negative number respectively. The lower score of PID controller is again attributed to the non-adaptive property.

7-7 Conclusion

This section has presented and discussed results from six control performance evaluation tasks. The goal of this chapter was to answer the research question **R.Q.3.1: How does the proposed controller perform in comparison to a traditional fixed-gain linear controller?** and **R.Q.3.2: How does the proposed controller perform as an adaptive controller?**. Under nominal conditions, the benchmarking PID controller performs slightly better than J-SNAC flight controller because of its better side-slip attenuation property and faster response to command signals. However, when there are parametric failures in the aircraft, J-SNAC flight controller outperforms the PID controller by adapting the in-place control law.

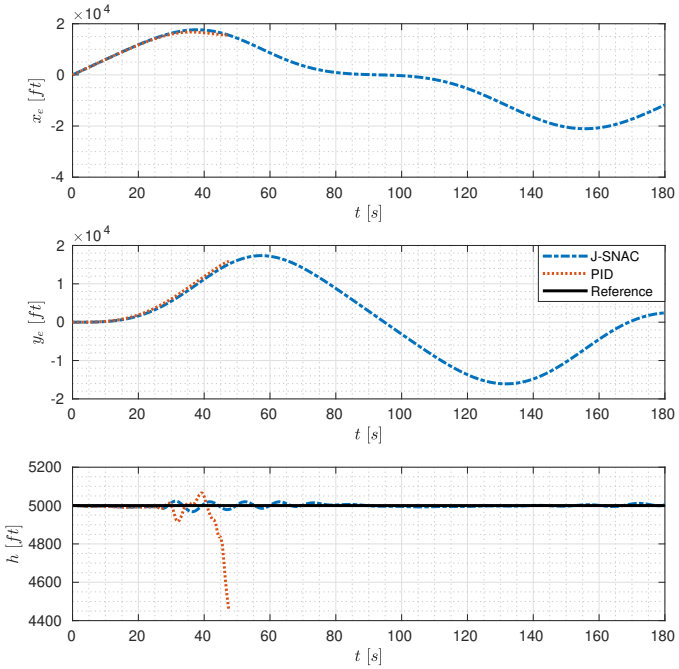


Figure 7-31: Effect of partial rudder failure on the altitude and position responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

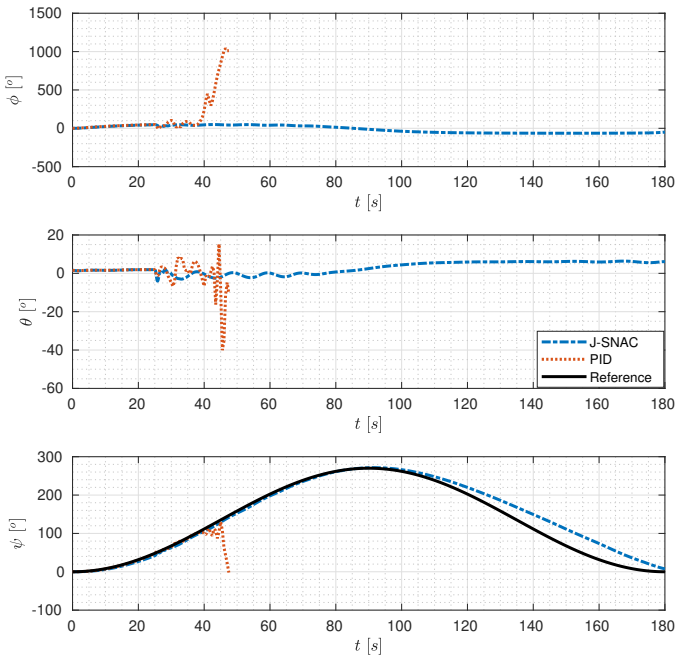


Figure 7-32: Effect of partial rudder failure on the heading and attitude angle responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

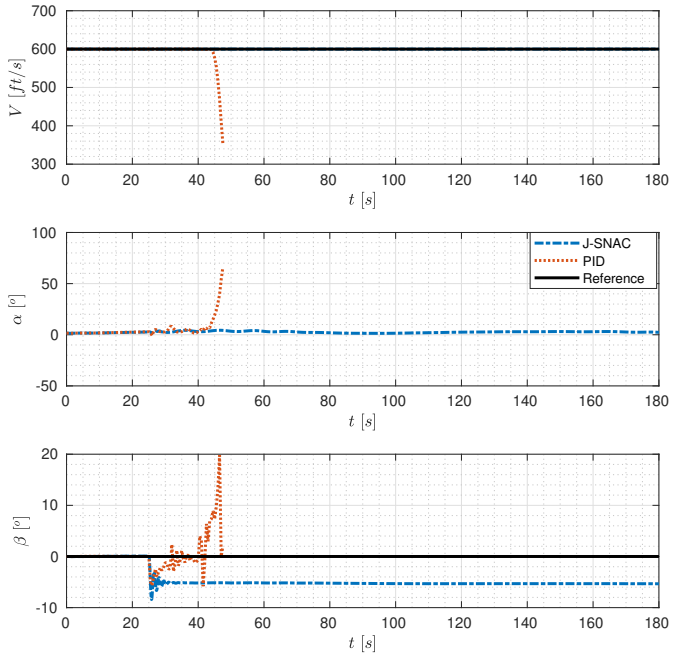


Figure 7-33: Effect of partial rudder failure on the airspeed and aerodynamic angles responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

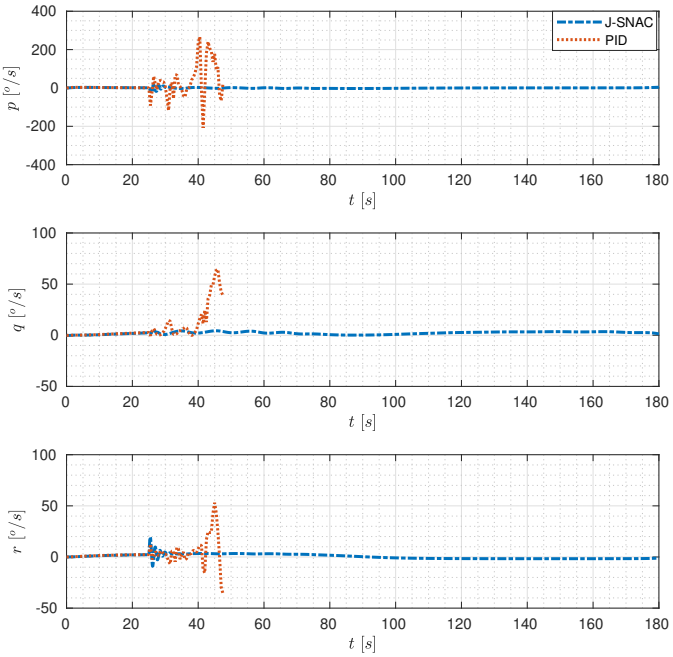


Figure 7-34: Effect of partial rudder failure on the angular rate responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

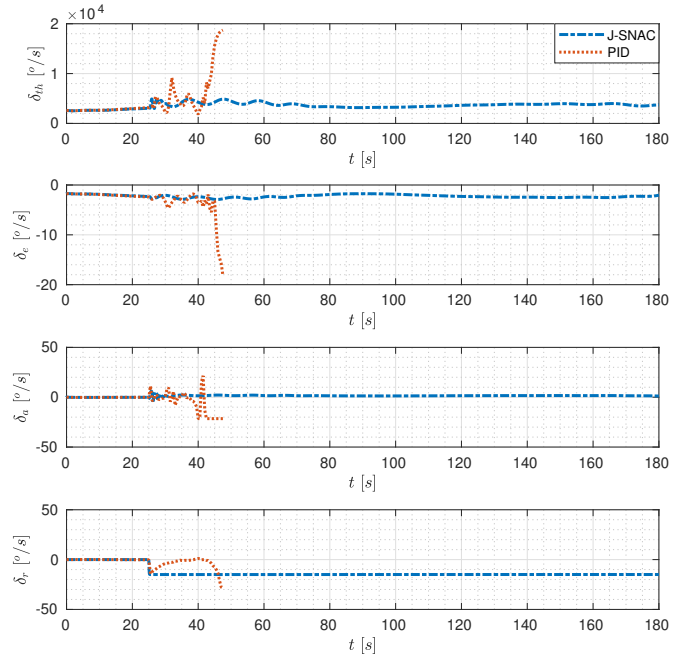


Figure 7-35: Effect of partial rudder failure on the actuator responses while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

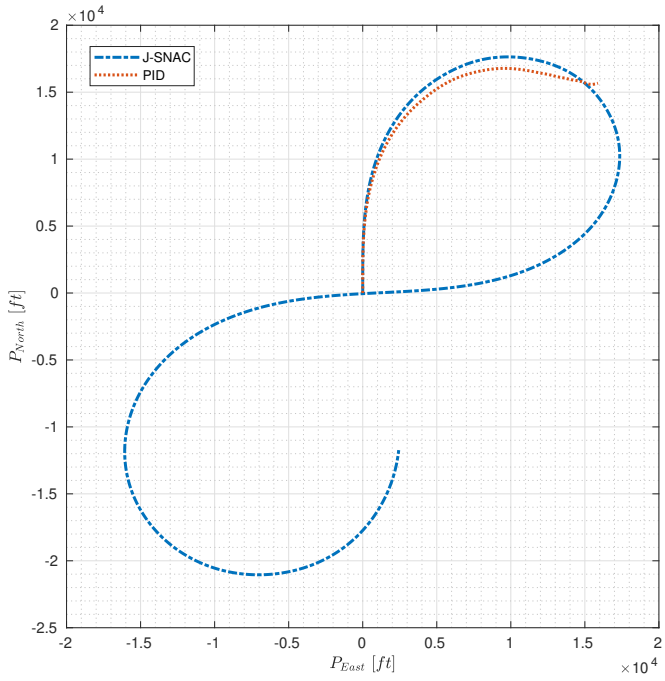


Figure 7-36: Effect of partial rudder failure on the ground track while tracking sinusoid reference signal with PID, and trained J-SNAC controllers.

Part IV

Closures

Chapter 8

Conclusions

The design, implementation, and evaluation of a *Reinforcement Learning (RL) Lateral-Directional Flight Controller* has been presented. The proposed controller and its assessment contribute to the field of Fault-Tolerant-Flight-Control systems, by validating the use of an improved RL algorithm for flight control system design and being one of the early attempts to developing lateral-directional flight control system with RL.

The objective of this thesis was to **"Improve the fault-tolerance of fixed-wing aircraft by investigating the applicability of J-SNAC algorithm for the design of an adaptive lateral-directional flight controller"**. Three main research questions were posed to structure the research and organize the process towards achieving the stated objective. The first research question, **R.Q.1**, concerned with investigating on Adaptive Critic (AC) algorithms, their use in flight control systems and delineating a design problem from the state-of-the-art literature. The second research question, **R.Q.2**, concerned with the finding of the conditions for successful implementation of a J-SNAC controller. Finally the last research question, **R.Q.3**, concerned with the extent to which J-SNAC based flight controller improves the performance and survivability of fixed-wing aircraft.

R.Q.1 has been answered in Chapter 3. This chapter presented various AC algorithms, explored existing applications of AC in Flight Control System design and compared these two aspects to conclude that there is a growing interest in the use of single network adaptive critic algorithms for the flight control system design. Next **R.Q.2** has been answered in Chapter 4. This chapter presented the development of a controller for an under-actuated pendulum with the J-SNAC algorithm. This development work had elucidated on the use of J-SNAC in the controller for a nonlinear system, found the hyper-parameters of the J-SNAC algorithm and analyzed the sensitivity the controller performance to changes in hyper-parameters. The results in the Article and in the Chapters 5 and 6 helped to answer the **R.Q.3**. In these chapters and article, the control performance of the proposed J-SNAC based flight controller was compared to that of a PID control law. The comparison showed that the proposed controller has a slight delay in sending commands to the actuators and lower side-slip attenuation in comparison to the PID controller. However, unlike the PID control law, the proposed flight controller is adaptable to unanticipated changes and gives an

improved performance under parametrically changed conditions. From the comparisons, it can be concluded that the use of J-SNAC improves the fault-tolerance of fixed-wing aircraft by bringing the capacity to learn control policies that can fly degraded or damaged aircraft. Verification and validation studies have been recommended to generalize the performance of the proposed flight controller for all fixed-wing aircraft.

Chapter 9

Recommendations

The performance of the proposed controller is dependent on the selected hyperparameters. Further improvement on the proposed flight controller could be made by automating the hyper-parameter tuning. Additionally, the adaptation of hidden parameters of the function approximators can improve the proposed controller's performance further. Changing the function approximator from NRBF network to multi-variable splines may improve the performance, as they have similar properties but more manageable to adapt.

Also, investigations are required on combing of the critics of all three controllers into one and use the same critic to adapt all three actor functions in a hierarchical order. This combination would resolve any conflicts among the sub-controllers. Furthermore, the proposed controller can be merged with the previously designed RL longitudinal flight controllers to synthesize a global flight controller.

Further progress can be made by validating this control law in a physical setup and figuring out the additional limitations. In the development of the controller, it was assumed that actuator saturation limit and physical limits for roll rate and yaw rate were known. However, for a truly intelligent controller, this parameter must be learned online. Furthermore, safe exploration and intelligent flight envelope protection scheme can be incorporated with the controller to have a general purpose intelligent flight controllers.

Adaptive critic flight controller can further be improved by separating the states that differ in timescale. Instead of using only longitudinal dynamics or lateral dynamics, one can use the three loops structure used in nonlinear dynamic inversion controller (innermost loop concerns with control of body rotation rate, central loop concerns with control of aerodynamic angles and outer-loop concerns navigational states). This separation also helps to merge control allocation methods with ACD based flight control system. Also, the load on computations in learning can be reduced by incorporating known non-changing dynamics. For example, in aircraft dynamics, the kinematic relations never changes. Therefore it does not add further values by learning this relationship online, the thing that is to be tracked is inertial properties such as the mass moment of inertia and inertial mass; such features are already being identified in modern aircraft with the various identification techniques. Also in the domain of aircraft

control, the control effectiveness is already being learned. Instead of learning the dynamics of aircraft online, this control effectiveness can be used in ACD algorithms. Another further comment is on the design of reward function. There already exists a vast amount of literature on incorporating handling and control quality in optimal control of aircraft. The objective function of this optimal control problems can directly be used as reward function to have desired handling and control quality.

In essence, ACD works based on simple mechanics. However, the challenge arises when the optimal actions are to be calculated. The search for optimal action requires the identification of the control effectiveness matrix and making sure that the learned value function remains bounded and have a maximum value in the origin of the concerned state-space. This task is related to system identification methods. Therefore the improvement in system identification techniques improves ACD based flight controllers. Moreover, the stability of the learning scheme has been left out in this work. Guaranteeing stability of the learning scheme needs to be incorporated into the value function learning scheme to increase the reliability of the proposed flight controllers.

Bibliography

[1] Boeing Airplane Safety. Statistical summary of commercial jet aircraft accidents: Worldwide operations, 1959-2016. *Boeing Commercial Airplane, Seattle, WA*, 2017.

[2] ICAO Safety. Safety Report. techreport, International Civil Aviation Organization, Montreal, Canada, 2017. URL `https://www.icao.int/safety/Documents/ICAO_SR_2017_18072017.pdf`.

[3] Thomas Jan Jozef Lombaerts. *Fault Tolerant Flight Control, A Physical Model Approach*. phdthesis, Technische Universiteit Delft, May 2010.

[4] Thomas Lombaerts, Hafid Smaili, and Jan Breeman. Introduction. In Christopher Edwards, Hafid Smaili, and Thomas Lombaerts, editors, *Fault tolerant flight control–a benchmark challenge*, pages 3–43. Springer, 1 edition, 2010. ISBN 978-3-642-11689-6.

[5] Christine M Belcastro and John V Foster. Aircraft loss-of-control accident analysis. In *Proceedings of AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Paper No. AIAA-2010-8004*, 2010.

[6] Xunhong Lv, Bin Jiang, Ruiyun Qi, and Jing Zhao. Survey on nonlinear reconfigurable flight control. *Journal of Systems Engineering and Electronics*, 24(6):971–983, 2013.

[7] Gang Tao. Multivariable adaptive control: A survey. *Automatica*, 50(11):2737–2764, 2014.

[8] Jin Jiang. Fault-tolerant control systems – an introductory overview. *Acta Automatica Sinica*, 31(1):161–174, 2005.

[9] Ron J Patton. Fault-tolerant control: the 1997 situation. *IFAC Proceedings Volumes*, 30 (18):1029–1051, 1997.

[10] Afef Fekih. Fault diagnosis and fault tolerant control design for aerospace systems: A bibliographical review. In *American Control Conference (ACC), 2014*, pages 1286–1291. IEEE, 2014.

[11] Michel Verhaegen, Kanev Stoyan, Redouane Hallouzi, Colin Jones, Jan Maciejowski, and Hafid Smail. Fault tolerant flight control – a survey. In Christopher Edwards, Hafid Smaili, and Thomas Lombaerts, editors, *Fault tolerant flight control–a benchmark challenge*, pages 43–84. Springer, 1 edition, 2010. ISBN 978-3-642-11689-6.

[12] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.

[13] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *IFAC Proceedings Volumes*, 36(5):257–268, 2003.

[14] Frank L Lewis, Draguna Vrabie, and Kyriakos G Vamvoudakis. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems*, 32(6):76–105, 2012.

[15] Danil V Prokhorov and Donald C Wunsch. Adaptive critic designs. *IEEE transactions on Neural Networks*, 8(5):997–1007, 1997.

[16] Silvia Ferrari and Robert F Stengel. An adaptive critic global controller. In *American Control Conference, 2002. Proceedings of the 2002*, volume 4, pages 2665–2670. IEEE, 2002.

[17] Paul J Werbos. Reconfigurable flight control via neurodynamic programming and universally stable adaptive control. In *American Control Conference, 2001. Proceedings of the 2001*, volume 4, pages 2896–2900. IEEE, 2001.

[18] Russell Enns and Jennie Si. Helicopter flight-control reconfiguration for main rotor actuator failures. *Journal of Guidance Control and Dynamics*, 26(4):572–584, 2003.

[19] Dongchen Han and SN Balakrishnan. Adaptive critic-based neural networks for agile missile control. *Journal of Guidance Control and Dynamics*, 25(2):404–406, 2002.

[20] Dongchen Han and SN Balakrishnan. Robust adaptive critic based neural networks for speed-constrained agile missile control. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 1999.

[21] Dongchen Han and SN Balakrishnan. Adaptive critic based neural networks for control-constrained agile missile control. In *American Control Conference, 1999. Proceedings of the 1999*, volume 4, pages 2600–2604. IEEE, 1999.

[22] Dongchen Han and SN Balakrishnan. State-constrained agile missile control with adaptive-critic-based neural networks. *IEEE Transactions on Control Systems Technology*, 10(4):481–489, 2002.

[23] Silvia Ferrari and Robert F Stengel. Online adaptive critic flight control. *Journal of Guidance Control and Dynamics*, 27(5):777–786, 2004.

[24] E Van Kampen, QP Chu, and JA Mulder. Online adaptive critic flight control using approximated plant dynamics. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 256–261. IEEE, 2006.

[25] Wilfred Nobleheart, Geethalakshmi Shivanapura Lakshmikanth, Animesh Chakravarthy, and James E Steck. 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*, page 5003, 2013.

[26] F.M. Pohl. Adaptive-critic designs for aircraft control. Master's thesis, Delft University of Technology, 2017.

[27] Kenji Doya. Reinforcement learning in continuous time and space. *Neural computation*, 12(1):219–245, 2000.

[28] Jie Ding, Ali Heydari, and S.N. Balakrishnan. *Single Network Adaptive Critics Networks-Development, Analysis, and Applications*, chapter 5, pages 98–118. John Wiley & Sons, Inc., Hoboken, New Jersey, 2013.

[29] Richard S Russell. Non-linear f-16 simulation using simulink and matlab. *University of Minnesota, Tech. paper*, 2003.

[30] ICAO Safety. Doc 10004, global aviation safety plan, 2017-2019. techreport, International Civil Aviation Organisation, 999 Robert-Bourassa Boulevard, Montral, Quebec, Canada H3C 5H7, 2016. URL `https://www.icao.int/publications/Documents/10004_en.pdf`.

[31] Christine M Belcastro, John V Foster, Gautam H Shah, Irene M Gregory, David E Cox, Dennis A Crider, Loren Groff, Richard L Newman, and David H Klyde. Aircraft loss of control problem analysis and research toward a holistic solution. *Journal of Guidance, Control, and Dynamics*, 2017.

[32] Diederick Alwin Joosten. *Constrained and reconfigurable flight control*. PhD thesis, The Dutch Institute of Systems and Control, The Netherland, 4 2017.

[33] Richard Ernest Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[34] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[35] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[36] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement learning and dynamic programming using function approximators*, volume 39. CRC press, 2010.

[37] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[38] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[39] Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, P.O Box 931, Bellmount, Massachusetts, 1996.

[40] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.

[41] Paul J Werbos. Neurocontrol and supervised learning: An overview and evaluation. *Handbook of intelligent control*, 65:89, 1992.

[42] Derong Liu. Approximate dynamic programming for self-learning control. *Acta Automatica Sinica*, 31(1):13–18, 2005.

[43] Fei-Yue Wang, Huaguang Zhang, and Derong Liu. Adaptive dynamic programming: An introduction. *IEEE computational intelligence magazine*, 4(2), 2009.

[44] SN Balakrishnan, Jie Ding, and Frank L Lewis. Issues on stability of adp feedback controllers for dynamical systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):913–917, 2008.

[45] Xin Xu, Lei Zuo, and Zhenhua Huang. Reinforcement learning algorithms with function approximation: Recent advances and applications. *Information Sciences*, 261:1–31, 2014.

[46] Ding Wang, Haibo He, and Derong Liu. Adaptive critic nonlinear robust control: a survey. *IEEE transactions on cybernetics*, 47(10):3429–3451, 2017.

[47] Paul J Werbos. Approximate dynamic programming for real-time control and neural modeling, 1992.

[48] Jennie Si. *Handbook of learning and approximate dynamic programming*, volume 2. John Wiley & Sons, 2004.

[49] Frank L Lewis and Derong Liu. *Reinforcement learning and approximate dynamic programming for feedback control*, volume 17. John Wiley & Sons, 2013.

[50] Radhakant Padhi, Nishant Unnikrishnan, Xiaohua Wang, and SN Balakrishnan. A single network adaptive critic (snac) architecture for optimal control synthesis for a class of nonlinear systems. *Neural Networks*, 19(10):1648–1660, 2006.

[51] Jie Ding, SN Balakrishnan, and Frank L Lewis. 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*, pages 1–8. IEEE, 2010.

[52] Thomas Hanselmann, Lyle Noakes, and Anthony Zaknich. Continuous adaptive critic designs. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 5, pages 3001–3006. IEEE, 2005.

[53] Frank L Lewis and Draguna Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE circuits and systems magazine*, 9(3), 2009.

[54] Ganesh K Venayagamoorthy, Ronald G Harley, and Donald C Wunsch. Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator. *IEEE Transactions on Neural Networks*, 13(3):764–773, 2002.

[55] Russell Enns and Jennie Si. Helicopter trimming and tracking control using direct neural dynamic programming. *IEEE Transactions on Neural Networks*, 14(4):929–939, 2003.

[56] Kalmanje Krishnakumar, Greg Limes, Karen Gundy-Burlet, and Don Bryant. An adaptive critic approach to reference model adaptation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 5790, 2003.

[57] SN Balakrishnan and Victor Biega. Adaptive-critic-based neural networks for aircraft optimal control. *Journal of Guidance, Control, and Dynamics*, 19(4):893–898, 1996.

[58] Sergio Esteban Roncero and SN Balakrishnan. Nonlinear flight control system with neural networks. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit Proceedings (1-9) Montreal: American Institute of Aeronautics and Astronautics*. American Institute of Aeronautics and Astronautics, 2001.

[59] Zhongwu Huang and S Balakrishnan. Robust adaptive critic based neurocontrollers for missiles with model uncertainties. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4159, 2001.

[60] Katie Grantham. Adaptive critic neural network based terminal area energy management/entry guidance. In *41st Aerospace Sciences Meeting and Exhibit*, page 305, 2003.

[61] James Steck, Geethalakshmi Lakshmikanth, and John Watkins. Adaptive critic optimization of dynamic inverse control. In *Infotech@ Aerospace 2012*, page 2408. 2012.

[62] Geethalakshmi Shivanapura Lakshmikanth, Scott Reed, John Watkins, and James E Steck. Single network adaptive critic aided nonlinear dynamic inversion with optimal control modification for fast adaptation of mrac flight control. In *AIAA Infotech@ Aerospace (I@ A) Conference*, page 5208, 2013.

[63] Wilfred Nobleheart, Animesh Chakravarthy, and James Steck. Single network adaptive critic (snac) design for a morphing aircraft. In *AIAA Guidance, Navigation, and Control Conference*, page 4614, 2012.

[64] Scott Reed and James E Steck. Adaptive control for fault tolerant autonomous carrier recovery. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 0871, 2018.

[65] Jie Ding and Sivasubramanya Balakrishnan. An online nonlinear optimal controller synthesis for aircraft with model uncertainties. In *AIAA Guidance, Navigation, and Control Conference*, page 7738, 2010.

[66] Jie Ding and SN Balakrishnan. Intelligent constrained optimal control of aerospace vehicles with model uncertainties. *Journal of guidance, control, and dynamics*, 35(5): 1582–1592, 2012.

[67] D Moorhouse and R Woodcock. Us military specification mil–f–8785c. Technical report, US Department of Defense Arlington County, 1980.