Reinforcement Learning based Online Adaptive Flight Control for the Cessna Citation II(PH-LAB) Aircraft

Konatala, R.B.; van Kampen, E.; Looye, Gertjan H.N.

DOI
10.2514/6.2021-0883

Publication date
2021

Document Version
Final published version

Published in
AIAA Scitech 2021 Forum

Citation (APA)
Konatala, R. B., van Kampen, E., & Looye, G. H. N. (2021). Reinforcement Learning based Online Adaptive Flight Control for the Cessna Citation II(PH-LAB) Aircraft. In *AIAA Scitech 2021 Forum: 11–15 & 19–21 January 2021, Virtual Event* Article AIAA 2021-0883 American Institute of Aeronautics and Astronautics Inc. (AIAA). https://doi.org/10.2514/6.2021-0883

AIAA SciTech Forum
11–15 &amp; 19–21 January 2021, VIRTUAL EVENT
AIAA Scitech 2021 Forum

10.2514/6.2021-0883

Reinforcement Learning based Online Adaptive Flight Control for the Cessna Citation II(PH-LAB) Aircraft

Ramesh Konatala*, E. van Kampen†
Delft University of Technology, P.O. Box 5058, 2600 GB Delft, The Netherlands

Gertjan Looye‡
German Aerospace Center, 82234 Wessling, Germany

Online Adaptive Flight Control is interesting in the context of growing complexity of aircraft systems and their adaptability requirements to ensure safety. An Incremental Approximate Dynamic Programming (iADP) controller combines reinforcement learning methods, optimal control and online identified incremental model to achieve optimal adaptive control suitable for Nonlinear Time-Varying systems. The main contribution of this paper is twofold. Firstly, the iADP controller is designed to achieve automatic online rate control to track pilot commands via setpoints provided by the manual outer loop on Citation II Aircraft model. Secondly, to assess the controller performance in the presence of sensor dynamics and actuator dynamics, an analysis is carried out to identify causes of any performance degradation. The simulation results from iADP longitudinal control using full state feedback indicate that the discretization of sensor signals, sensor bias and transport delays did not have any significant effect on the controller performance or on the incremental model identification. However noisy signals and sensors delays are found to cause controller performance degradation. Appropriate filtering of signals resulted in better estimation of the incremental model subsequently improving the controller performance due to noisy signals. Control performance degradation due to sensor delays should be addressed in future before conducting flight tests on Citation II Aircraft.

Nomenclature

α, β	Angle of attack, Sideslip angle
X, U	State, Control input
R	Reward
V	Value function estimate
π	Policy
μ	Deterministic policy
J	Cost-to-go
P	Kernel matrix
$\delta_a, \delta_e, \delta_r$	Aileron, Elevator and Rudder deflections
γ	Discount factor
τ	Time constant
Q, R	Weighting matrices
Θ, Cov	Parameter matrix, Covariance matrix
p, q, r	Roll, Pitch and Yaw rate
ϕ, θ, ψ	Roll, Pitch and Yaw
V_{tas}, h, γ, n_z	True airspeed, Altitude, Flight path angle and Load factor
F_t, G_t	State matrix, Input matrix

*MSc Student, Control and Simulation Division, Faculty of Aerospace Engineering, Delft University of Technology
†Assistant Professor, Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology
‡Head, Department of Aircraft System Dynamics, Institue of System Dynamics and Control, German Aerospace Center

I. Introduction

The surge in the air traffic growth and increased complexity of the modern day aircraft systems in recent decades made flight safety a priority in the modern day aviation. According to the International Civil Aviation Organization (ICAO), a statistical analysis on risk category effecting the flight safety show that most number of fatalities are due to the Loss of Control-In Flight (LOC-I). Designing a Flight Control System (FCS) that is resilient to system failures, external disturbances, inappropriate control inputs by the crew and/or autoflight systems is one of the ways to ensure flight safety by preventing LOC-I[1].

Modern day FCS is designed using Fly-By-Wire (FBW) system and Flight Control Computer (FCC) which interprets the pilot's inputs as the desired outcome and converts these commands to the appropriate control surface actions for the actuators based on a Flight Control Law (FCL). However the underlying FCL should be both adaptive and robust to cope up with unforeseen situations or failure. Incremental model based NDI and backstepping versions viz,. Incremental Nonlinear Dynamic Inversion (INDI) and Incremental Backstepping (IBS) are some of the popular control methods designed with the aim of improving flight safety. In these incremental based methods, the model to be inverted is written in an incremental form using Taylor series expansion and an incremental control input is evaluated at every time step. These methods are found to increase the robustness against model uncertainties[2] [3] and similar observations are validated through flight tests in cooperation with the Aircraft System Dynamics department, DLR Oberpfaffenhofen on a fixed wing Cessna Citation II PH-Lab aircraft[4][5][6]. Another interesting approach in designing FCS is using Reinforcement Learning (RL) based FCL. Active research is going on in RL based control to achieve model free nonlinear optimal control with online learning capability. In RL based control, a control problem is defined as an objective to achieve and the optimal control law is achieved by solving for optimization using Dynamic Programming (DP) techniques[7]. In practice achieving control for real systems using DP is not viable as DP assumes a perfect known model of the system and high computational expense needed to solve optimization problem for larger state space. The former problem in the context of RL is referred to as *"Curse of Dimensionality"*.

Approximate Dynamic Programming (ADP) combines generalization methods like function approximators with DP techniques rendering these methods suitable to achieve optimal control for larger state space systems. The ADP methods are used to achieve feedback control for dynamical systems using a cost-to-go function with online learning capability using data observed along the system trajectories[8][9]. This ADP method is further extended to solve for reference tracking problem[10, 11]. Although these methods are model free, they assume a linear time-invariant model of the system to be controlled, thus making it difficult to extend these methods for nonlinear aerospace systems. Based on theory from INDI and IBS, an incremental version of ADP is proposed, which is referred to as Incremental Approximate Dynamic Programming (iADP) for stabilizing control problem using a quadratic cost function approximation. This method which uses an online identified local linearized model using Least squares or Recursive Least squares approach, making this method suitable for nonlinear time varying systems. This method is further extended to achieve more general reference tracking control[12] and two algorithms with full state feedback and output feedback are proposed and the control approach is verified on a F-16 aircraft model. However the iADP controller is yet to be verified on a real system.

The aim of this paper is to extend the RL based controller to Cessna Citation II aircraft and evaluate the viability of using this controller for a real system. To attend this objective, additional research has to carried out on integrating the RL based controller within FCS Cessna Citation II aircraft and study the effects of typical aircraft characteristics like sensor, actuator dynamics, time delays on the controller. The main contributions of this paper are as follows: Firstly, iADP controller is integrated into FCS of Citation II Aircraft to achieve automatic online rate control. Secondly, iADP controller performance is assessed considering sensor and actuator dynamics. The controller performance is evaluated for longitudinal control of the aircraft using full state feedback and output feedback.

The contents of this paper are structured as follows. In Section II, basic concepts of Reinforcement Learning are discussed followed by derivation of the iADP algorithms. In Section III, Cessna Citation II aircraft model along with sensor and actuator models used for simulations is discussed. FCS design of iADP controller for Citation II is explored in Section IV. Section V contains the results from the controller evaluation on the aircraft model. Finally, in Section VI main conclusions from this paper are presented.

II. Reinforcement Learning for optimal adaptive control

Optimal control design involves designing a controller to optimize a cost function that characterizes the desired behaviour of a system. Techniques like Linear Quadratic Gaussian(LQG) are often used to achieve optimal control through a quadratic cost function and a linear model of the system to be controlled. It is desirable to have a controller that does not completely rely on the model of the system as it is difficult to obtain a perfect model of the system due to modelling uncertainties. Optimal adaptive control methods address this issue by redesigning optimal controllers for varying models of the system which are identified using system identification techniques. A direct way to achieve this optimal adaptive control is a model free controller that learns the control scheme online using real time observations along system trajectories[13] and Reinforcement Learning(RL) schemes are found to be useful in designing this direct approach.

Reinforcement Learning

RL process essentially involves an agent interacting in an environment which learns to choose actions such that a certain goal/objective is reached. The RL agent achieves this through a trail and error search method and memorization of situations/states and suitable actions reinforced through the rewards yielded from the environment. Many of the RL algorithms adopt an actor-critic architecture as shown in Fig. (1) which enables online learning through real time observations. In an actor-critic setting, the actor does the job of control policy (mapping from system states to the control action inputs) implementation with the policy updates provided by the critic. The critic evaluates the current policy by updating the value associated with the current state using the cost information provided by the system/environment and updates the control policy for the actor such that the cost associated with the new policy is smaller than the previous one.

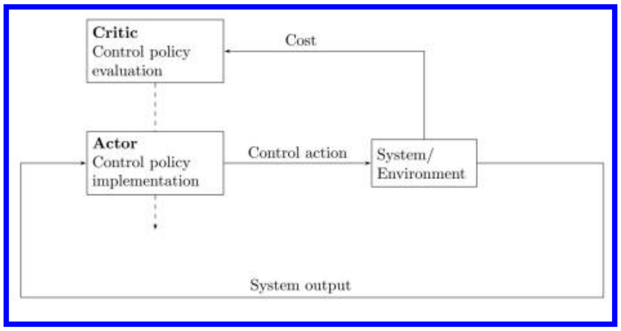

Fig. 1 Actor critic structure of RL agent

The RL problem is formalized mathematically as a Markov Decision Process(MDP) which assumes that the process obeys the memoryless Markov property, which is the concept that a future state is independent of the preceding states given the current state. The MDP is then solved for optimality using techniques like Dynamic Programming(DP).

To solve a DP problem it should have an optimal substructure and overlapping sub problems. The DP algorithms require the complete knowledge of the environment to estimate the state value functions. However it is not practical to have knowledge of the environment in all the cases and thus methods like Monte Carlo(MC) Methods which attain the optimal behaviour through experience can be used to attend RL problem. Monte Carlo refers to the use of random sampling methods to approximate numerical results. Monte Carlo methods in the context of RL refer to the learning of the agent in an environment through experience using sample returns observed. Thus instead of evaluating value function for all the states using a perfect known model of the environment we estimate the value of the states through some policy using the experience gained while visiting those states in an episode. The MC methods differ from DP methods in two ways. Firstly the agent learns from experience instead of state space sweep to estimate value functions and secondly the value functions are estimated directly from returns instead of other value estimates. Temporal Difference(TD) learning combines the advantage of sampling from experience in MC methods and learning from incomplete episodes in DP methods. While in the MC prediction we need to wait till we finish the episode to estimate the value function for a particular state, in a TD prediction we can instead estimate the value of the state by taking one step ahead and then using the value estimate of the new state that we have landed in. Another difference between TD and MC method is that the TD learning algorithm exploits the Markov property by first building an approximate model of the MDP and then converging the solution from the data for the estimated MDP.

Consider a MDP : (X, U, P, R) where X denotes set of states, U denotes set of control actions/inputs. The conditional probability of the MDP to transition from state $x \in X$ to $x' \in X$ by taking action $u \in U$ is $P^{u}_{xx'} = Pr\{x'|x, u\}$ and the expected immediate cost necessary for the transition is $R^{u}_{xx'}$. The control policy or action strategy $\pi(x, u) = Pr\{u|x\}$ is the mapping from states X to actions U. The policy can be stochastic $\pi(x, u)$ where there is non zero probability of selecting more than one control u or deterministic $\mu(x)$ policy which admits only one control given state x. The goal of the RL problem is to find the optimal policy π_*(or μ_* for deterministic optimal policy) which minimizes the expected future cost. Extending the MDP framework to a dynamical system which evolves through time we assume that the state transitions happen at discrete time steps : $k, k+1, k+2....$ The one step cost necessary for the transition $x_k \longrightarrow x_{k+1}$ by taking action u_k is defined by $r_k = r_k(x_k, u_k, x_{k+1})$. The discounted infinite horizon cost J_k provides measure of sum of future costs incurred by the dynamical system to evolve through time in the future and is given by

$$J_k = \sum_{i=k}^{\infty} \gamma^{i-k} r_i \tag{1}$$

where $0 \leq \gamma \leq 1$ discounts the costs incurred in further in future. Consider the RL agent selects control actions at every time step k following control policy $\pi_k(x_k, u_k)$. The value for a policy V^{π} is defined as the expected value of the future cost for a dynamical system starting from state x at time k and following the policy $\pi(x, u)$ subsequently, thus providing a measure of the value being in state x with the policy being followed as π. The value function is given by,

$$\begin{aligned} V^{\pi}(x) &= E_{\pi}\{J_k | x_k = x\} \\ &= E_{\pi}\left\{\sum_{i=k}^{\infty} \gamma^{i-k} r_i | x_k = x\right\} \\ &= E_{\pi}\left\{r_k + \gamma \sum_{i=k+1}^{\infty} \gamma^{i-(k+1)} r_i | x_k = x\right\} \end{aligned} \tag{2}$$

Here E_{π} denotes expectation of the value over all possible transitions conditional on policy π being followed. The Bellman equation is a fundamental concept in solving reinforcement learning problems which helps in arriving at optimal policies using experiences received further in time. The Bellman equation can be obtained from (2) as follows,

$$V^{\pi}(x) = \sum_{u} \pi(x, u) \sum_{x'} P^{u}_{xx'}[R^{u}_{xx'} + \gamma V^{\pi}(x')] \tag{3}$$

The value function should satisfy the Bellman equation at all stages of time. This equation can be interpreted as the relation between the current value of state $x = x_k$ and the value of state $x' = x_{k+1}$ whilst following policy $\pi(x, u)$. For an ergodic dynamical system it is proved that the MDP will have an deterministic optimal policy [14] to minimize the expected future cost. *Policy evaluation* is the procedure of arriving at the value of a policy which can be obtained using Bellman equation (3). If we know the value for a given policy $\pi(x, u)$ we can find another policy π' which is at least better than π and this step is referred to as *Policy improvement* which can be written as

$$\pi'(x, u) = \underset{u}{argmin} \sum_{x'} P^{u}_{xx'}[R^{u}_{xx'} + \gamma V^{\pi}(x')] \tag{4}$$

The optimality is reached when $\pi'(x, u) = \pi(x, u)$ and according to the Bellman's optimality principle[15] the optimal control policy and the optimal cost can be written as

$$u^{*} = \underset{u}{argmin} \sum_{x'} P^{u}_{xx'}[R^{u}_{xx'} + \gamma V^{*}(x')] \tag{5}$$

$$V^{*}(x) = \underset{\pi}{min} \sum_{x'} P^{u}_{xx'}[R^{u}_{xx'} + \gamma V^{*}(x')] \tag{6}$$

The objective of the RL problem is to arrive at the optimal control policy and this can be achieved using two iterative algorithms which use mapping between value and policy through policy evaluation and policy improvement steps. *Policy iteration(PI)* is the method of solving the RL problem through repeated sequence of policy evaluation and policy improvement steps until optimal solution is found. *Value iteration(VI)* is a special case of policy iteration method where instead of waiting for the exact convergence of policy evaluation, it is truncated to just one iteration and based on the approximate value function obtained policy improvement is done and the entire process is repeated till convergence. These DP based algorithms require the state transition probabilities $P^{u}_{xx'}$ and the cost $R^{u}_{xx'}$ of the MDP to arrive at the optimal control policy and can only be solved offline. To design optimal adaptive controllers it is desirable to have a method which does not rely on the full knowledge of the system. Temporal Difference(TD) is a model free RL method when applied for control systems has the capability of online learning using observed data measured along the system trajectories, which can be used to design optimal adaptive controllers.

In a TD method, the policy evaluation step is done using observed data collected along one sample path of MDP which the agent follows. The equation (3) now becomes a deterministic equation and the Bellman equation for TD can be written as

$$V^{\pi}(x_k) = r_k + \gamma V^{\pi}(x_{k+1}) \tag{7}$$

where the observed data is (x_k, r_k, x_{k+1}) at time step k. The TD error is given by the equation (8) and the objective is to update the value such that the TD error is minimized using PI or VI.

$$e_k = -V^{\pi}(x_k) + r_k + \gamma V^{\pi}(x_{k+1}) \tag{8}$$

For discrete systems the TD method provides exact solutions which can be arranged in a n-dimensional lookup table where n is the size of the state vector. In control systems we deal with continuous state and control spaces and when discretized, the state-space increases the number of states in the lookup table exponentially, a phenomenon referred to as "curse of dimensionality". This problem is addressed by approximating the value function using unknown parameters and suitable approximation structure. For a linear system the value function can be approximated to be quadratic in state[16] as shown in equation (9) which benefits from having one local/global minimum

$$V^{\pi}(x_k) = x_k^T P x_k \tag{9}$$

where P is a positive definite symmetric kernel matrix. These methods form class of Approximate Dynamic Programming(ADP) methods. The one-step cost r_k can be constructed based on the requirements of the control to be achieved viz, regulation, tracking with minimum control. A standard form is a quadratic energy function represented as equation (10) where Q,R are state weighting and control weighting matrices which provides trade off between objective to be achieved and the control effort required.

$$r_k = Q(x_k) + u_k^T R u_k \tag{10}$$

Because of the quadratic value function assumption, the ADP methods are suitable for dynamical systems which are Linear Time Invariant(LTI). However as most of the aerospace systems are nonlinear it is desirable to design controller which can deal with system nonlinearities and model uncertainties. Incremental model techniques approximate the original nonlinear dynamical system to linear time varying system around an operating point using first order Taylor series expansion. ADP methods are combined with incremental approach to design optimal controllers suitable for nonlinear systems referred to as Incremental Approximate Dynamic Programming (iADP) controllers[17, 18]. As these iADP controllers use only observed data for achieving the control iADP controllers can be classified as model free methods that has online learning capability.

A. Incremental Approximate Dynamic Programming for Tracking control

Here the methodology of extending iADP controllers to solve more general tracking control problems will be explained considering both the availability of full state observations and partial observability conditions.

1. Incremental model for Nonlinear system

Consider a Non-linear continuous system represented as follows:

$$\begin{aligned} \dot{x}(t) &= f[x(t), u(t)] \\ y(t) &= h[x(t)] \end{aligned} \tag{11}$$

where $f[x(t), u(t)] \in R^n$, $u(t) = R^m$ and output measurements are obtained using the measurement vector $h[x(t)] \in R^p$. As in practice we work with the discrete systems for achieving the control the above nonlinear system is discretized using a high sampling frequency and is represented as (12).

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= h(x_k) \end{aligned} \tag{12}$$

The objective is to design the iADP controller such that the system tracks a reference signal. Let the reference trajectory dynamics be represented for a discrete case as

$$\begin{aligned} r_{k+1} &= f_r(r_k) \\ y_k^r &= h_r(r_k) \end{aligned} \tag{13}$$

where $f_r(r_k) \in R^l$. By representing the reference signal in this form one can generate large class of reference trajectories. Augmenting the system dynamics with the reference dynamics we can generate the following augmented nonlinear system

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} f(x_k, u_k) \\ f_r(r_k) = \end{bmatrix} = t(X_k, u_k) \tag{14}$$

where $t(X_k, u_k) \in R^{n+l}$. The quadratic cost function will now be a quadratic in augmented state X_k. Linearizing the above augmented nonlinear discrete system around X_0, u_0 by taking the first order Taylor series expansion we get

$$X_{k+1} = t(X_k, u_k) \approx t(X_0, u_0) + \left.\frac{\partial t(X_k, u_k)}{\partial X_k}\right|_{X_0, u_0} (X_k - X_0) + \left.\frac{\partial t(X_k, u_k)}{\partial u_k}\right|_{X_0, u_0} (u_k - u_0) \tag{15}$$

As it is assumed that the discretization is done at a high sampling frequency we can consider Δt to be very small and can approximate $X_{k-1} \approx X_k$ and can replace X_0, u_0 with X_{k-1}, u_{k-1} to get (16)

$$\begin{aligned} X_{k+1} - X_k &\approx T(X_{k-1}, u_{k-1})(X_k - X_{k-1}) + G(X_{k-1}, u_{k-1})(u_k - u_{k-1}) \\ \Delta X_{k+1} &\approx T_{k-1} \Delta X_k + G_{k-1} \Delta u_k \end{aligned} \tag{16}$$

where $T_{k-1} = T(X_{k-1}, u_{k-1}) \in R^{(n+l) \times (n+l)}$ is the system matrix and $G_{k-1} = G(X_{k-1}, u_{k-1}) \in R^{(n+l) \times m}$ is the control effectiveness matrix. This regression model represented by F_{k-1}, G_{k-1} can be identified using Recursive Least Squares(RLS) techniques which provides a Linear Time Variant(LTV) approximation to the original model.

2. *Full state feedback*

For dynamical systems where full state measurements are available the observed measurements can be written as:

$$Y_k = \begin{bmatrix} y_k \\ y_k^r \end{bmatrix} = X_k \tag{17}$$

Using the utility function (10) for achieving tracking control and extending the concept of Bellman equation (7) to the incremental model we get the optimal Value function (18)

$$V^*(X_k) = \underset{\Delta u_k}{min}[(y_k - y_k^r)^T Q (y_k - y_k^r) + (u_{k-1} + \Delta u_k)^T R (u_{k-1} + \Delta u_k) + \gamma V^*(X_{k+1})] \tag{18}$$

Where the optimal control at time step k is given by (19)

$$\Delta u^* = \underset{\Delta u_k}{argmin}[(y_k - y_k^r)^T Q (y_k - y_k^r) + (u_{k-1} + \Delta u_k)^T R (u_{k-1} + \Delta u_k) + \gamma V^*(X_{k+1})] \tag{19}$$

using the quadratic value function approximation (9) we get (20)

$$\begin{aligned} X_k^T P X_k &= (y_k - y_k^r)^T Q (y_k - y_k^r) + u_k^T R u_k + \gamma X_{k+1}^T P X_{k+1} \\ &= (y_k - y_k^r)^T Q (y_k - y_k^r) + (u_{k-1} + \Delta u_k)^T R (u_{k-1} + \Delta u_k) + \\ &\quad \gamma (X_k + T_{k-1} \Delta X_k + G_{k-1} \Delta u_k)^T P (X_k + T_{k-1} \Delta X_k + G_{k-1} \Delta u_k) \end{aligned} \tag{20}$$

For optimal control we can set the derivative of the above cost function with respect to Δu_k to 0 and we get the optimal control law (21)

$$\Delta u_k = -(R + \gamma G_{k-1}^T P G_{k-1})^{-1} [R u_{k-1} + \gamma G_{k-1}^T P X_k + \gamma G_{k-1} P T_{k-1} \Delta X_k] \tag{21}$$

The VI algorithm for iADP-FS is given by the Algorithm (1)

Algorithm 1 iADP for tracking control using Full State Feedback[12]

```
Initialize a arbitrary control policy $\Delta u_k^0 = \mu(X_k)$
repeat
  Value Update Step: $X_k^T PX_k = (y_k - y_k^r)^T Q(y_k - y_k^r) + u_k^T Ru_k + \gamma X_{k+1}^T PX_{k+1}$
  Policy Improvement Step: $\Delta u_k = -(R + \gamma G_{k-1}^T PG_{k-1})^{-1}[Ru_{k-1} + \gamma G_{k-1}^T PX_k + \gamma G_{k-1} PT_{k-1}\Delta X_k]$
until Convergence
```

3. Output feedback

Often in practice, as measurement of full system states is not available, controller design using input-output measurement data over suitable time horizon is desirable. In this method the input output measurements are used to indirectly construct the state information, under the assumption that the system is observable. The measured data is then used to arrive at the optimal control using iADP method.

Consider we measure the data at N time steps between interval $[k - N, k]$, using equations (16) and (25) we can write (22),

$$\Delta X_k = \begin{bmatrix} \Delta x_k \\ \Delta r_k \end{bmatrix} \approx \begin{bmatrix} \tilde{F}_{k-2,k-N-1} & 0 \\ 0 & \tilde{D}_{k-2,k-N-1} \end{bmatrix} \begin{bmatrix} \Delta x_{k-N} \\ \Delta r_{k-N} \end{bmatrix} + \begin{bmatrix} U_N \\ 0 \end{bmatrix} \bar{\Delta} u_{k-1,k-N} \tag{22}$$

where $\tilde{F}_{k-a,k-b} = \Pi_{i=k-a}^{k-b} F_i$ and $\tilde{D}_{k-a,k-b} = \Pi_{i=k-a}^{k-b} D_i$, the input-output measurements captured over the time horizon [k-N,k] and the controllability matrix U_N are given by equations (23) and (24) respectively

$$\bar{\Delta} u_{k-1,k-N} = \begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \\ \Delta u_{k-N} \end{bmatrix} \in R^{mN}, \qquad \bar{\Delta} y_{k,k-N+1} = \begin{bmatrix} \Delta y_k \\ \Delta y_{k-1} \\ \vdots \\ \Delta y_{k-N+1} \end{bmatrix} \in R^{pN} \tag{23}$$

$$U_N = \left[G_{k-2} \quad F_{k-2} G_{k-3} \ldots \tilde{F}_{k-2,k-N} G_{k-N-1} \right] \in R^{n \times mN} \tag{24}$$

Linearizing the output of the nonlinear system(12) and the reference output of the system (13) using First order Taylor series expansion around x_{k-1} we get (25) and (26) respectively

$$\Delta y_k \approx H_{k-1} \Delta x_k \tag{25}$$

$$\Delta y_k^r \approx H_{k-1}^r \Delta r_k \tag{26}$$

where $H_{k-1} = \frac{\partial h(x)}{\partial x}|_{x_{k-1}} \in R^{p \times n}$ and $H_{k-1}^r = \frac{\partial h^r(x)}{\partial x}|_{x_{k-1}} \in R^{r \times n}$ are the observation matrices. Now using the input-output data from (22) we can write (25) and (26) as follows

$$\begin{aligned} \bar{\Delta} y_{k,k-N+1} &\approx V_N \Delta x_{k-N} + W_N \bar{\Delta} u_{k-1,k-N} \\ \bar{\Delta} y_{k,k-N+1}^r &\approx R_N \Delta r_{k-N} \end{aligned} \tag{27}$$

The matrices V_N, W_N and R_N are given by equations (28), (29) and (30) respectively

$$V_N = \begin{bmatrix} H_{k-1} \tilde{F}_{k-2,k-N-1} \\ H_{k-2} \tilde{F}_{k-3,k-N-1} \\ \vdots \\ H_{k-N} F_{k-N-1} \end{bmatrix} \in R^{pN \times n} \tag{28}$$

$$W_N = \begin{bmatrix} H_{k-1}G_{k-2} & H_{k-1}F_{k-2}G_{k-3} & H_{k-1}\tilde{F}_{k-2,k-3}G_{k-4} & \dots & H_{k-1}\tilde{F}_{k-2,k-N}G_{k-N-1} \\ 0 & H_{k-2}G_{k-3} & H_{k-2}F_{k-3}G_{k-4} & \dots & H_{k-2}\tilde{F}_{k-3,k-N}G_{k-N-1} \\ 0 & 0 & H_{k-3}G_{k-4} & \dots & H_{k-3}\tilde{F}_{k-4,k-N}G_{k-N-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & H_{k-N}G_{k-N-1} \end{bmatrix} \in R^{(p+m)\times N} \tag{29}$$

$$R_N = \begin{bmatrix} H^r_{k-1}\tilde{F}_{k-2,k-N-1} \\ H^r_{k-2}\tilde{F}_{k-3,k-N-1} \\ \vdots \\ H^r_{k-N}F_{k-N-1} \end{bmatrix} \in R^{rN\times l} \tag{30}$$

We can extract Δx_{k-N} and Δr_{k-N} from (27) as follows

$$\begin{aligned} \Delta x_{k-N} &\approx V_N^+ (\bar{\Delta y}_{k,k-N+1} - W_N \bar{\Delta u}_{k-1,k-N}) \\ \Delta r_{k-N} &\approx R_N^+ \bar{\Delta y}^r_{k,k-N+1} \end{aligned} \tag{31}$$

where $V_N^+ = (V_N^T V_N)^{-1} V_N^T$ and $R_N^+ = (R_N^T R_N)^{-1} R_N^T$ are the pseudo inverses of the respective matrices. Substituting (31) in (22) we get

$$\begin{aligned} \Delta X_k &\approx \begin{bmatrix} \tilde{F}_{k-2,k-N-1}V_N^+ & 0 \\ 0 & \tilde{D}_{k-2,k-N-1}R_N^+ \end{bmatrix} \begin{bmatrix} \bar{\Delta y}_{k,k-N+1} \\ \bar{\Delta y}^r_{k,k-N+1} \end{bmatrix} + \begin{bmatrix} U_N - \tilde{F}_{k-2,k-N-1}V_N^+ W_N \\ 0 \end{bmatrix} \bar{\Delta u}_{k-1,k-N} \\ &\approx \begin{bmatrix} U_N - \tilde{F}_{k-2,k-N-1}V_N^+ W_N & \tilde{F}_{k-2,k-N-1}V_N^+ & 0 \\ 0 & 0 & \tilde{D}_{k-2,k-N-1}R_N^+ \end{bmatrix} \begin{bmatrix} \bar{\Delta u}_{k-1,k-N} \\ \bar{\Delta y}_{k,k-N+1} \\ \bar{\Delta y}^r_{k,k-N+1} \end{bmatrix} \\ &\approx \begin{bmatrix} M_{\Delta u} & M_{\Delta y} & M_{\Delta y^r} \end{bmatrix} \bar{\Delta Z}_{k,k-N} \end{aligned} \tag{32}$$

Thus augmented state can be reconstructed using the input output data over a certain time horizon using above equation. It can also be shown that the output increments can also be constructed using past data measurements as follows:

$$\begin{aligned} \Delta y_{k+1} &\approx \underline{G}_k \bar{\Delta u}_{k-1,k-N} + \underline{F}_k \bar{\Delta y}_{k-1,k-N} \\ &\approx \underline{G}_{k,11}\Delta u_k + \underline{G}_{k,12}\bar{\Delta u}_{k-1,k-N+1} + \underline{F}_k \bar{\Delta y}_{k,k-N+1} \end{aligned} \tag{33}$$

where $\underline{G}_k \in R^{p\times Nm}$ is the extended control effectiveness matrix, $\underline{F}_k \in R^{p\times Np}$ is the extended system matrix, $\underline{G}_{k,11} \in R^{p\times m}$ and $\underline{G}_{k,12} \in R^{p\times(N-1)m}$ are partitioned matrices from $\underline{G}_k$.

Similarly Δy^r_{k+1} can also be constructed as:

$$\Delta y^r_{k+1} \approx \underline{F}^r_k \bar{\Delta y}_{k-1,k-N} \tag{34}$$

where $\underline{F}^r_k \in R^{r\times Nr}$. We define the quadratic cost to go function using a kernel matrix and the measured data as follows

$$V(\bar{Z}_{k,k-N+1}) = \bar{Z}^T_{k,k-N+1}\bar{P}\bar{Z}_{k,k-N+1} \tag{35}$$

where $\bar{Z}$ contains the input output data given by

$$\bar{Z}_{k,k-N+1} = \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k,k-N+1} \\ \bar{y}^r_{k,k-N+1} \end{bmatrix} \in R^{(m+p+r)N} \tag{36}$$

Extending the Bellman equation for tracking control with the incremental model representation, using the input output data we get

$$\bar{Z}^T_{k,k-N+1}\bar{P}\bar{Z}_{k,k-N+1} = (y_k - y^r_k)^T Q(y_k - y^r_k) + (u_{k-1} + \Delta u_k)^T R(u_{k-1} + \Delta u_k) + \gamma \bar{Z}^T_{k+1,k-N+2}\bar{P}\bar{Z}_{k+1,k-N+2} \tag{37}$$

where,

$$\bar{Z}^T_{k+1,k-N+2}\bar{P}\bar{Z}_{k+1,k-N+2} = \begin{bmatrix} u_{k-1} + \Delta u_k \\ \bar{u}_{k-1,k-N+1} \\ \bar{y}_k + \Delta y_{k+1} \\ \bar{y}_{k,k-N+2} \\ \bar{y}^r_k + \Delta y^r_{k+1} \\ \bar{y}^r_{k,k-N+2} \end{bmatrix}^T \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & P_{26} \\ P_{31} & P_{32} & P_{33} & P_{34} & P_{35} & P_{36} \\ P_{41} & P_{42} & P_{43} & P_{44} & P_{45} & P_{46} \\ P_{51} & P_{52} & P_{53} & P_{54} & P_{55} & P_{56} \\ P_{61} & P_{62} & P_{63} & P_{64} & P_{65} & P_{66} \end{bmatrix} \begin{bmatrix} u_{k-1} + \Delta u_k \\ \bar{u}_{k-1,k-N+1} \\ \bar{y}_k + \Delta y_{k+1} \\ \bar{y}_{k,k-N+2} \\ \bar{y}^r_k + \Delta y^r_{k+1} \\ \bar{y}^r_{k,k-N+2} \end{bmatrix} \tag{38}$$

The optimal control policy in terms of the measured data is now given by (39)

$$\begin{aligned} \Delta u_k &= \underset{\Delta u_k}{argmin}[(y_k - y^r_k)^T Q(y_k - y^r_k) + (u_{k-1} + \Delta u_k)^T R(u_{k-1} + \Delta u_k) + \gamma \bar{Z}^T_{k+1,k-N+2}\bar{P}\bar{Z}_{k+1,k-N+2}] \\ &= -[R + \gamma P_{11} + \gamma(\underline{G}^T_{k,11})P_{33}\underline{G}^T_{k,11} + \gamma P_{13}\underline{G}_{k,11} + \gamma(P_{13}\underline{G}_{k,11})^T]^{-1} \\ &\quad [[R + \gamma P_{11} + \gamma(\underline{G}_{k,11})^T P^T_{13}]u_{k-1} + \gamma[\underline{G}^T_{k,11}P_{33} + P_{13}]y_k \\ &+ \gamma[P_{12} + (\underline{G}_{k,11})^T P_{23}]\bar{u}_{k-1,k-N+1} + \gamma[P_{14} + (\underline{G}_{k,11})^T P_{34}]\bar{y}_{k,k-N+2} \\ &+ \gamma[(\underline{G}_{k,11})^T P_{33} + P_{13}]((\underline{G}_{k,12}\Delta \bar{u}_{k-1,k-N+1} + \underline{F}_k \bar{\Delta} y_{k,k-N+1})) \\ &+ \gamma[P_{15} + (\underline{G}_{k,11})^T P_{35}]y^r_k + \gamma[P_{15} + (\underline{G}_{k,11})^T P_{35}]\underline{F}^r_k \bar{\Delta} y^r_{k,k-N+1})) + \gamma[P_{16} + (\underline{G}_{k,11})^T P_{36}]\bar{y}^r_{k,k-N+2} \end{aligned} \tag{39}$$

The VI algorithm for iADP using output feedback is given by the Algorithm (2)

Algorithm 2 VI algorithm for iADP using output feedback[12]

```
Initialize a arbitrary control policy $\Delta u^0_k = \mu(\bar{Z}_{k,k-N+1})$
repeat
  Value Update Step:
  $\Delta u_k = -[R + \gamma P_{11} + \gamma(\underline{G}^T_{k,11})P_{33}\underline{G}^T_{k,11} + \gamma P_{13}\underline{G}_{k,11} + \gamma(P_{13}\underline{G}_{k,11})^T]^{-1}$
      $[[R + \gamma P_{11} + \gamma(\underline{G}_{k,11})^T P^T_{13}]u_{k-1} + \gamma[\underline{G}^T_{k,11}P_{33} + P_{13}]y_k$
      $+ \gamma[P_{12} + (\underline{G}_{k,11})^T P_{23}]\bar{u}_{k-1,k-N+1} + \gamma[P_{14} + (\underline{G}_{k,11})^T P_{34}]\bar{y}_{k,k-N+2}$
      $+ \gamma[(\underline{G}_{k,11})^T P_{33} + P_{13}]((\underline{G}_{k,12}\Delta \bar{u}_{k-1,k-N+1} + \underline{F}_k \bar{\Delta} y_{k,k-N+1}))$
      $+ \gamma[P_{15} + (\underline{G}_{k,11})^T P_{35}]y^r_k + \gamma[P_{15} + (\underline{G}_{k,11})^T P_{35}]\underline{F}^r_k \bar{\Delta} y^r_{k,k-N+1})) + \gamma[P_{16} + (\underline{G}_{k,11})^T P_{36}]\bar{y}^r_{k,k-N+2}$
until Convergence
```

B. Online Incremental Model Identification

The Incremental Model is identified in real time using Recursive Lease Squares (RLS) method assuming high sampling rate. The RLS is a recursive variant of Ordinary Least Squares (OLS) method which consists of simple matrix operations, whereas the OLS has a matrix inversion step[19]. Avoiding matrix inversion is ideal as the online model identification is done through some excitation signal and during phase of no excitation matrix, inversion might lead to numerical instability. The RLS method can also deal with time-varying systems and they demand small computational requirements which makes them suitable for online implementation. The derivation of Incremental model identification using RLS method with Full State measurements is adopted from[20].

1. Full State Measurements

For the implementation of iADP algorithm in section II.A.2, the augmented state transition matrix T_{k-1} and input distribution matrix G_{k-1} has to be identified online. The equation (15) can be segmented by row as follows:

$$\Delta x_{r,k+1} = \begin{bmatrix} \Delta x_k^T & \Delta u_k^T \end{bmatrix} \begin{bmatrix} f_{r,k-1}^T \\ g_{r,k-1}^T \end{bmatrix} \tag{40}$$

where $\Delta x_{r,k+1}$ is the r^{th} state increment yielding $f_{r,k-1}^T$ and $g_{r,k-1}^T$, the r^{th} row elements of F_{k-1} and G_{k-1} respectively. We can construct the parameter matrix Θ_{k-1} as follows

$$\Theta_{k-1} = \begin{bmatrix} F_{k-1}^T \\ G_{k-1}^T \end{bmatrix} \in \mathbb{R}^{(n+m)\times m} \tag{41}$$

The state prediction in terms of parameter matrix is:

$$\Delta \hat{x}_{k+1}^T = W_k^T \hat{\Theta}_{k-1}, \qquad W_k = \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} \in \mathbb{R}^{(n+m)\times 1} \tag{42}$$

The parameter matrix is updated as follows:

$$\begin{aligned} \epsilon_k &= \Delta x_{k+1}^T - \Delta \hat{x}_{k+1}^T \\ \hat{\Theta}_k &= \hat{\Theta}_{k-1} + \frac{Cov_{k-1} W_k}{\gamma^{RLS} + W_k^T Cov_{k-1} W_k} \epsilon_k \\ Cov_k &= \frac{1}{\gamma^{RLS}} \left(Cov_{k-1} + \frac{Cov_{k-1} W_k W_k^T Cov_{k-1}}{\gamma^{RLS} + X_k^T Cov_{k-1} X_k} \right) \in \mathbb{R}^{(n+m)\times(n+m)} \end{aligned} \tag{43}$$

where ϵ_k is the innovation or state prediction error, Cov is the estimation Covariance matrix and $\gamma^{RLS} \in [0, 1]$ is the forgetting factor.

Similar procedure can be adopted for identifying reference dynamics to obtain F_{k-1}^r. Now the system transition matrix can be created for the augmented system as follows

$$T_{k-1} = \begin{bmatrix} F_{k-1} & 0 \\ 0 & F_{k-1}^r \end{bmatrix}$$

2. Output Feedback

The incremental model in input output data in equation (33) can also be constructed using RLS method. Here as the full state measurements are not available, the incremental model is constructed using incremental input output measurements over a time horizon. Equation (40) is now modified to include historical incremental data instead of state measurements as follows :

$$\Delta y_{r,k+1} = \begin{bmatrix} \bar{\Delta} y_{k,k-N+1}^T & \bar{\Delta} u_{k,k-N+1}^T \end{bmatrix} \begin{bmatrix} f_{r,k}^T \\ g_{r,k}^T \end{bmatrix}$$

Using the similar procedure mentioned above RLS can be used to estimate F_k, G_k, F_k^r.

III. Cessna Citation II PH-Lab Research Platform

The Cessna Citation II(Model 550) twin-jet business aircraft is a pressurized, low-wing monoplane that is certified for up to 10 persons including two pilots[21], is jointly operated by TU Delft and National Aerospace Laboratory (NLR). The aircraft has maximum operating altitude 13 km and maximum cruising speed of 710 km/h. The aircraft is modified as a airborne research platform(PH-lab) and flight tests are organized in cooperation with external partners like DLR, Oberpfaffenhofen to test the FCL developed by Aircraft System Dynamics Department. The aircraft has a mechanically linked Flight Control System (FCS), an autopilot system facilitated by FCC, a Flight Test Instrumentation System (FTIS) and an experimental Fly-By-Wire (FBW) system. The control system of the Cessna Citation II consists of cables that are connected to the control surfaces. The movements of the control surfaces are converted to electronic signals and the FCC determines these signals based on expected actuator response and provides them to the servo amplifiers of the actuators that deflect the control surfaces. The FTIS consists of a data acquisition computer and signal conditioning unit that can process information from sensors and can provide measurements at a high sample rate of upto 1000Hz that can be available for controllers[22]. Some of the sensor signals made available for FTIS are Attitude Heading and Reference System (AHRS), Digital Air Data Computer (DADC), air data boom, control surface synchros which measure deflection angles of the control surfaces. Angle of attack is also available from a body mounted vane sensor. The FBW is developed based on the existing original autopilot system of the aircraft[23] which uses the position setpoint values from the FCL and feed back signals from servo which command the actuators. The overall PH-Lab aircraft diagram integrated with components useful for flight testing is shown in Fig. 2.

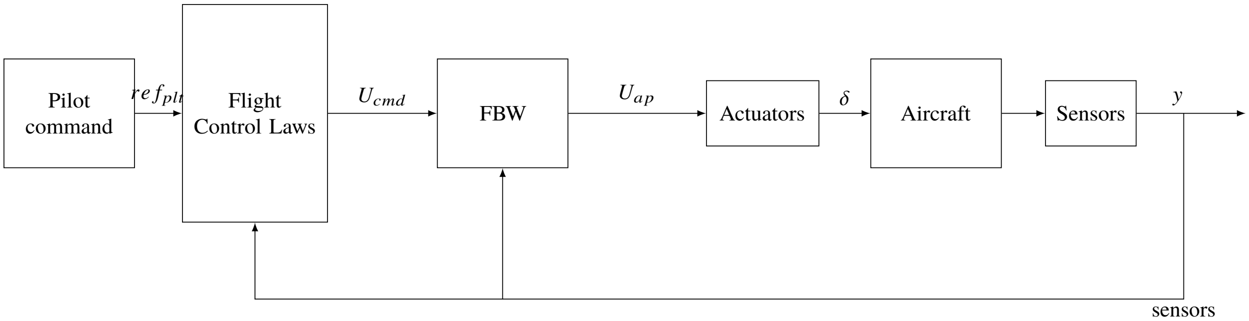

Fig. 2 Overview of PH-Lab integrated with Flight Control Laws and Fly-By-Wire system

Aircraft, sensor and actuator models that are available for testing the FCL's are discussed here.

A. DASMAT Aircraft Model

A simulation model for Cessna 500 aircraft was designed as a standard Flight CAD package referred to as DASMAT [24] and further improvements using this baseline model were made through flight tests on Citation II [25]. The baseline model is based on a generic nonlinear aircraft model with aerodynamic,propulsion and engine models. Models of external conditions like atmospheric wind and turbulence are also available which can be interfaced with the aircraft model. These models use the 6-DOF combined translational and rotational nonlinear equations of motion for the rigid body aircraft. The state vector is constructed using these 6-DOF equations and without the engine model the following 12 state variables and 8 Aerodynamic control inputs are available.

$$x = \begin{bmatrix} p & q & r & V_{tas} & \alpha & \beta & \phi & \theta & \psi & h_e & x_e & y_e \end{bmatrix}$$

$$u = \begin{bmatrix} \delta_e & \delta_a & \delta_r & \delta_{te} & \delta_{ta} & \delta_{tr} & \delta_f & lg_{sw} & \end{bmatrix}$$

where δ_f denotes flap control surface, lg_{sw} denotes the landing gear. Further the model also provides observations which will be useful for control design viz,. aircraft states, their derivatives, accelerations, force and moment components from aerodynamic and propulsion models. An accurate mass model was developed and adopted to Citation II which provides aircraft mass, inertia and center of gravity position[4].

B. Sensor model

The sensors instrumentation model of the PH-Lab is identified using flight test data[4][26]. These are modelled taking into account the practical phenomenon like bias, noise, delays, resolution and sampling rate. The noise of sensors are modelled as Gaussian white noise with zero mean. The sensor characteristics available from different sensor systems are shown in the Table 1.

C. Actuator model

A high fidelity actuator model is developed and adopted for use within Citation II with better estimation along elevator and aileron channels[6]. As this model assumes smaller control system movements which cannot be guaranteed

Table 1 PH-LAB Sensor characteristics [6]

Signal	Noise (σ^2)	Bias	Resolution	Delay[ms]	Sampling rate [Hz]
$p, q, r, \dot{\phi}, \dot{\theta}, \dot{\psi}[rad/s]$	4.0×10^{-7}	3.0×10^{-5}	6.8×10^{-7}	90	52
$\theta, \phi[rad]$	1.0×10^{-9}	4.0×10^{-3}	9.6×10^{-7}	90	52
$A_x, A_y, A_z[g]$	1.5×10^{-5}	2.5×10^{-3}	1.2×10^{-4}	117	52
$V_{TAS}, V_{CAS}[m/s]$	8.5×10^{-4}	2.5	3.2×10^{-2}	300	16,8
$\delta_a, \delta_e, \delta_r[rad]$	5.5×10^{-7}	2.4×10^{-3}	–	0	100
$\alpha_{boom}, \beta_{boom}[rad]$	7.5×10^{-8}	1.8×10^{-3}	9.6×10^{-5}	100	100
$\alpha_{body}[rad]$	4.0×10^{-10}	–	1.0×10^{-5}	280	1000

during the iADP controller training process, a low fidelity first order actuator model is chosen. This low fidelity actuator model is developed using flight test data to accommodate the dynamics of FBW[4] system. It is modelled as a first order system with a lag component, actuator deflection and rate saturation limits and transport delay as follows:

$$\dot{\delta}(t) = sat_{\dot{\delta}}\{\tau_{act}^{-1}\delta_{com}(t - \lambda_{act}) - \tau_{act}^{-1} sat_{\delta}[\delta(t)]\} \tag{44}$$

Where sat_{δ} and $sat_{\dot{\delta}}$ represent the saturation function for actuator deflection and rate respectively. τ_{act} is the time lag component identified using a step input response and the delay between FBW and the control surface deflection is modelled as the transport delay λ_{act}. The actuator model characteristics are listed in the Table 2.

Table 2 PH-LAB Actuator characteristics [4]

	$\delta_{max}[°]$	$\delta_{min}[°]$	$\dot{\delta}_{max}[°/s]$	$\lambda_{act}[ms]$	$\tau_{act}[ms]$
Aileron	15	-19	19.7	39.8	84
Elevator	15	-17			
Rudder	22	-22			

IV. iADP Control Law Design

This section presents the control law design for integrating iADP controller within the FCL's of Citation II aircraft. iADP controller is used for automatic control of inner loop while the outer loop is based upon the previously designed manual control[4] laws. The output from the slow outer loop is used as the reference signal to be tracked by the faster inner loop. The outer loop contains Command and reference model and a side-slip controller. The command module ensures safety through attitude flight envelope protection. The reference model is a second order model which converts the commanded signals from pilot into values achievable by aircraft. Smooth reference signals are necessary for iADP controller to ensure any sharp increase on cost function which might result in numerical instability in updating the kernel matrix P. A coordinated flight is desirable due to the absence of FBW for yaw channel. Thus, an outer loop controller for yaw channel is designed which generates reference for yaw rate such that any side slip angle is rejected. Standard flight maneuvers like 3211 for pitch tracking, bank to bank maneuvers for roll are simulated and are provided as input pilot commands to the outer loop.

A coupled longitudinal and lateral rate iADP controller is designed for the automatic inner loop which can learn to control all the available control surfaces at the same time. The advantage of having a combined longitudinal and lateral control is that it can learn control parameters without neglecting any coupling effects.

A. iADP-FS Online Rate Control Design

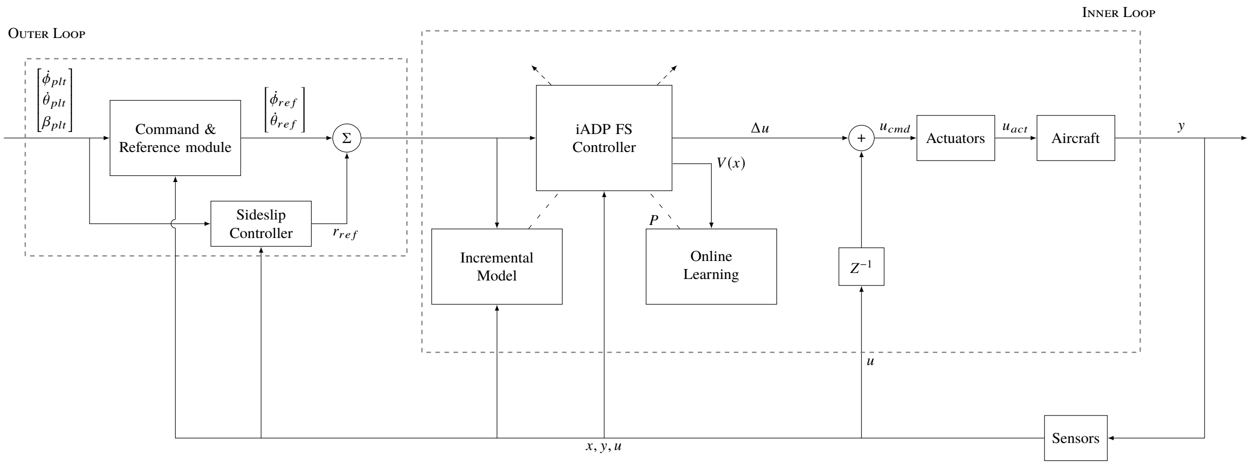

Fig. 3 Controller architecture with online training using Full State Feedback for combined longitudinal and lateral control of Cessna Citation II

The control system architecture with integrated iADP controller using full state feedback is shown in Fig. 3. iADP is used for rate control design as the rate control has least learning complexity. This is due to the fact that the effect of control surfaces input on aircraft angular rates are faster as compared to that of control surface effect on angle of attack, due to time scale principle. However in theory iADP should be able to achieve control involving slower dynamic variables even as it does not assume time scale separation[20]. Assuming actuators are modelled as first order systems the order of learning complexity according to time scale separation is as follows:

$$p, q, r < \phi, \theta, \psi, \alpha, \beta < V, \gamma, \chi \tag{45}$$

The states, control input, output and reference vectors used by the iADP controller are

$$x = \begin{bmatrix} p & q & r & \alpha & \beta & \phi & \theta \end{bmatrix}^T, u = \begin{bmatrix} \delta_a & \delta_e & \delta_r \end{bmatrix}^T$$

$$y = \begin{bmatrix} \dot{\phi} & \dot{\theta} & r \end{bmatrix}^T, y_r = \begin{bmatrix} \dot{\phi}_{ref} & \dot{\theta}_{ref} & r_{ref} \end{bmatrix}^T$$

The air speed information is not included in the state vector due to slower local variations in the airspeed variable which might effect the incremental model identification. The previous implementations are designed using a air speed controller, however this is not possible on Cessna citation because of lack of auto throttle functionality. The effect of variations in air speed on the controller are mitigated by choosing the reference commands to track such that variations in air speed are restricted to smaller values. Also as the controller is implemented with online learning capability this might reduce effects of air speed variations further.

The incremental model for the system is provided with state information, actuator position measurements(x, u). The incremental model for the reference dynamics is identified using the reference signal from the outer loop. Online incremental model for the system as well as the reference dynamics is identified online using RLS approach and the identified model coefficients are provided to the iADP controller. The iADP controller calculates the control increments using the model information from the incremental model and the measurements from the system. For online controller adaptation the kernel matrix P is updated at every time step using a Least Squares method with the data collected along the system trajectory. Recalling equation (20) we can write:

$$\begin{aligned} X_k^T P X_k &= (y_k - y_k^r)^T Q (y_k - y_k^r) + u_k^T R u_k + \gamma X_{k+1}^T P X_{k+1} \\ X_k^T P X_k &= V(X_k) \\ (X_k \otimes X_k)^T \vec{P} &= V(X_k) \\ X^{kr} \vec{P} &= V(X_k) \\ \vec{P} &= X^{kr+}.V(X_k) \end{aligned} \tag{46}$$

where $X^{kr} = (X_k \otimes X_k)^T$ is the Kronecker product, $\vec{P}$ is the kernel matrix reorganized as a vector, X^{kr+} is the pseudo inverse of X^{kr}. The online learning block stores the cost function estimate $V(X)$ and the Kronecker product($X^{kr} = (X_K \otimes X_k)^T \in \mathbb{R}^{(1\times(n+l)^2)}$) of augmented state vector X from the iADP controller over a certain time window $t_{ol} = N_{ol} \times f$, where N_{ol} are the number of samples collected during this window and f is the frequency of simulation. The collected cost estimate and state vector are stacked as follows:

$$\begin{aligned} \bar{V}(x) &= \begin{bmatrix} V(X_k) & \dots & V(X_{k-N_{ol}}) \end{bmatrix}^T \in \mathbb{R}^{(N_{ol}\times 1)} \\ \bar{X}^{kr} &= \begin{bmatrix} X_k^{kr} & \dots & X_{k-N_{ol}}^{kr} \end{bmatrix}^T \in \mathbb{R}^{(N_{ol}\times(n+l)^2)} \end{aligned} \tag{47}$$

Now the kernel matrix can be updated recursively using the data observed over this window as :

$$\vec{P} = \bar{X}^{kr+}.\bar{V}(X_k) \tag{48}$$

It is assumed for the iADP combined control design, clean measurements from sensors are available, thus effects of sensor dynamics like noise,delays, bias and quantization are neglected.

B. iADP-FS Online Longitudinal Rate Control Design

A simple longitudinal control design is considered to analyze the effects of real world phenomenon on the controller performance as this design needs only limited sensor measurements of states/outputs and actuators to study the individual effects of different phenomenon. The control architecture is as shown in Fig. 4 where only variables related to longitudinal rate control are considered. The manual outer loop provides the reference signal to be tracked to the automatic inner rate control loop. The iADP controller is used for the inner rate control with full state feedback. The states, outputs, reference and control vectors used by the iADP controller are

$$x = \begin{bmatrix} q & \alpha \end{bmatrix}^T, u = \delta_e, y = \dot{\theta}, y_r = \dot{\theta}_{ref}$$

.

Further as the primary aim of this analysis is to study the influence of real world phenomenon on controller performance, effects of sensor dynamics like bias, noise, delays, transport delay and quantization effects are considered. Transport delay for citation model is added in the control channel. To mitigate the effect of sensor noise, processing of noisy signals is done through signal filtering as shown in Fig. 4 and filtered signals ($\hat{x}, \hat{y}, \hat{u}$) are used by the iADP controller and also for the incremental model identification. Similar to the combined control approach, the online learning is achieved by updating the kernel matrix P at every time step using the data within a window.

C. iADP-OPFB Longitudinal Rate Control Design

Similar to previous section a simple longitudinal control design is considered to evaluate the output feedback algorithm. The control architecture is as shown in Fig. 5 where it is assumed that the full state feedback is not available and the task of the iADP is to achieve longitudinal rate control using only output measurements over a time horizon. As the effects of sensor dynamics are considered, the noisy sensor measurements are processed through signal filtering. Due to higher learning complexity of the algorithm the controller is trained offline to arrive at a baseline controller P which is used to evaluate the controller. The offline training is done by certain number of episodes where the kernel matrix is updated at the end of every episode. Every episode is initialized with kernel matrix P that is carried on from the previous episode and the aircraft is reset to steady wing level flight condition at the beginning of episode. The incremental model identifies the model coefficients necessary predict the next incremental output. The output,reference and control vectors used by the iADP controller are

$$y = \dot{\theta}, y_r = \dot{\theta}_{ref}, u = \delta_e$$

D. Signal filtering

The iADP controller is a model free controller which is based only on the measurements that are obtained along the system trajectory. For Full State Feedback controller as shown in Fig. 3 the controller needs full state measurement

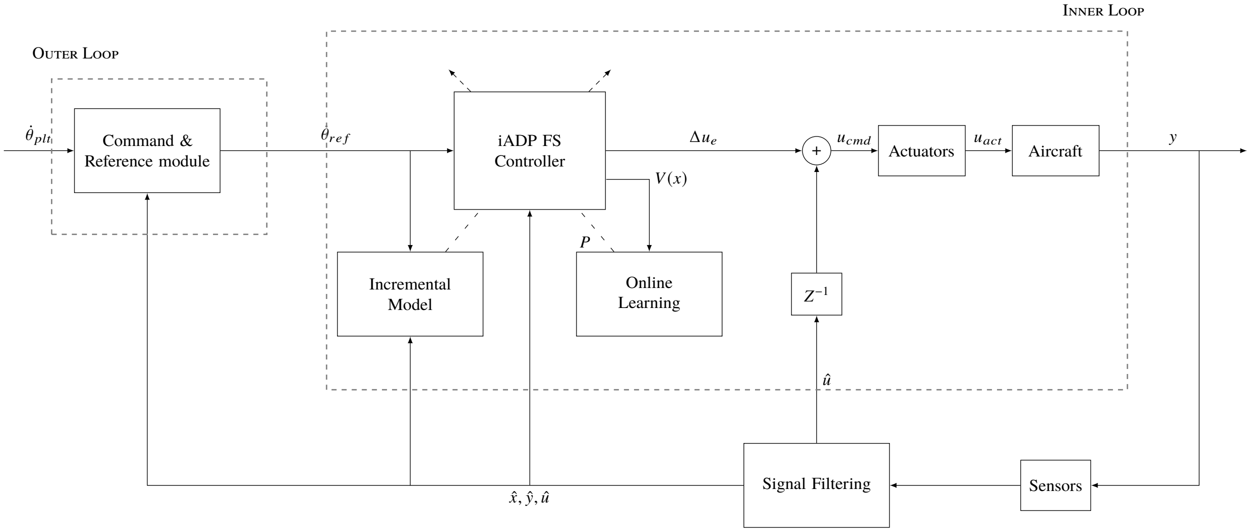

Fig. 4 Controller architecture with online learning using Full State Feedback for longitudinal control of Cessna Citation II

and the actuator deflection measurements. State information is made available through various sensors like AHRS for attitude and angular rates while angle of attack and side slip angle are measured using vane type sensors. Control surface deflection measurements are available for Citation-II aircraft which are measured using synchros. This alleviates the need for an actuator model to estimate the actuator position.

To mitigate the effects of noisy sensor measurements appropriate signal filtering is required. All the signals are filtered using a first order low-pass filter (49) unless specified with a cut off frequency of $\omega_n = 20 rad/s$.

$$H(s) = \frac{\omega_n}{s + \omega_n} \tag{49}$$

E. Implementation issues

The following section provides discussion on some of the implementation issues related to the iADP Controller viz, Incremental model identification, Persistent excitation and parameter tuning.

1. Online Incremental Model Identification

The iADP controller performance is dependent on good identification of the incremental model parameters. As the controller do not have any prior knowledge of the model, procedures similar to online system identification have to be adopted for incremental model identification. Online system identification typically involves exciting the aircraft through specific control inputs along different channels. Some of the common flight maneuvers used for system identification are listed in [19] and some of these maneuvers and necessary control input parameters are adopted here.

(a) For estimation of parameters related to longitudinal motion a short period motion is selected as this motion can provide most information for parameter estimation related to vertical and pitching motion. A multi step input like 3211, which consists of alternative positive and negative steps with relative duration of 3,2,1,1 respectively is chosen . The duration of the time step can be tuned to excite mode of interest.

(b) For lateral motion parameter estimation, banking roll maneuver and dutch roll motion are chosen. The banking maneuver is achieved through multi step aileron pulses while the dutch roll is excited through a doublet input.

The control inputs are chosen such that the aircraft can come back to steady state condition and are skewed in time across the channels such that different dynamic motions across different axes are excited. Another advantage of these control inputs is that the pilot can easily provide these inputs during flight avoiding automated control input generation.

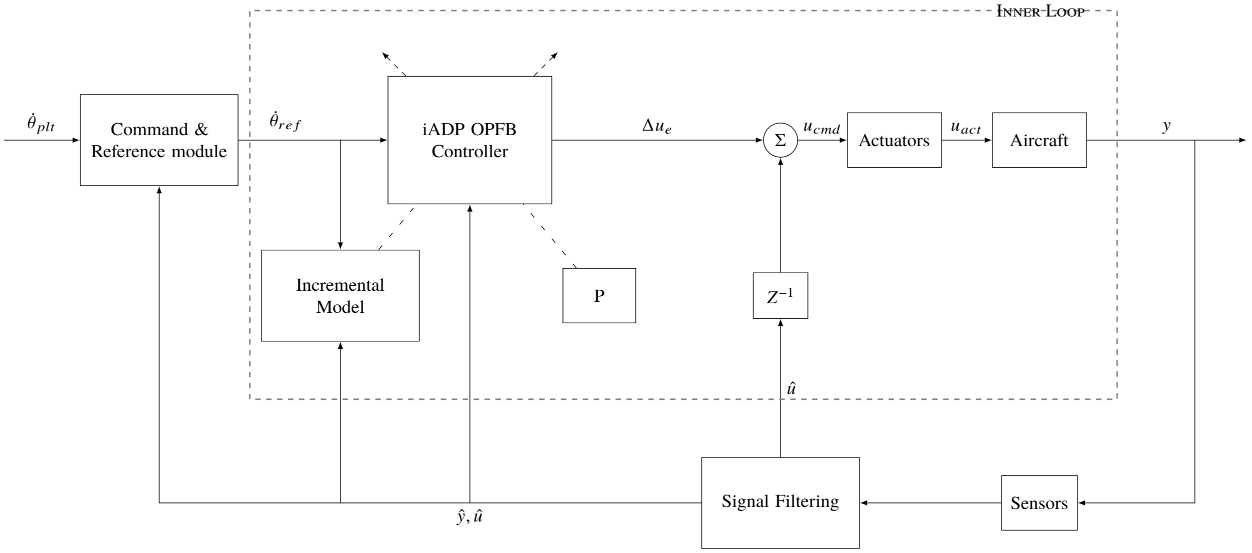

Fig. 5 iADP controller architecture with offline learning using Output Feedback for longitudinal control of Cessna Citation

2. Persistent Excitation

The need for Persistent excitation is two fold. One for online system identification such that all the modes of the system are excited continuously so that the model parameters can be updated. Second it serves for state space exploration which is a necessary condition for RL algorithm to arrive at good policies that can minimize the cost function. Although the input signals mentioned above are useful in identifying the initial model parameters, such signals cannot be implemented continuously. To aid continuous learning for model identification and controller state space exploration, white noise is added to the control input along with the incremental control input across the three channels. Thus the control input becomes

$$u_k = u_{k-1} + \Delta u_k + PE$$

where PE is white noise. During initial mode excitation phase, PE is white noise combined with control input used for model identification mentioned in previous section. The reference tracking problem includes a feedback loop from output y to u. For systems involving feedback loop it is also necessary that the reference signal is also persistently exciting to estimate the model parameters[27].

3. Parameter tuning

For optimal control performance parameter tuning is essential. For online incremental model identification the following hyperparameters are involved viz,. forgetting factor γ_{RLS}, initial covariance matrix Cov_0 and the initial parameter matrix Θ_0. For the *iadp* controller the following hyperparameters are involved viz,. forgetting factor γ, weighting matrices Q and R, initial kernel matrix P_0.

The RLS algorithm for model identification uses forgetting factor γ_{RLS} which provides control over the importance of data and its recency. For time-varying models γ can be chosen to be a value less than 1 to rely on the most recent data. However this makes the parameter updates more sensitive to the noise in the data. This is important if the real time data is obtained from noisy sensors. Typical range of values for the forgetting factor are $0.95 \leq \gamma^{RLS} \leq 1$[27]. The Covariance matrix(Cov) is a measure of confidence in the parameter matrix Θ. It is generally initialized as an identity matrix scaled by a factor. During the phases of poor excitation, covariance matrix parameters might grow exponentially leading to covariance wind up[28] causing numerical instability errors. The initial parameter matrix(Θ_0) contain the control effectiveness matrix and dynamic model information. These are typically initialized as zero matrices.

The hyperparameters of the iADP controller directly influence the controller performance. The discount factor $gamma \in [0, 1]$ is a measure of importance of cost information from future states. Typically, γ is initialized to value smaller than 1 to contain the infinite horizon cost to a finite value. Another advantage of discount factor is that it can reduce the bias effects that is introduced by the white noise during persistent excitation and the effects of improper initial conditions[13]. The weighting matrices Q and R provide a trade off between the tracking performance and control input energy[12]. The choice of Q and R will also effect the stability and robustness of the controller and hence should be initialized to appropriate values. Initialization of Q and R is done through trial and error such that satisfactory performance can be achieved. Fine tuning of Q and R might be essential for systems involving multiple inputs and outputs. In such scenarios fine tuning of parameters can be achieved through optimization techniques like Multi Objective Parameter Synthesis (MOPS)[29] such that certain control design requirements like overshoot, settling time, rise time and tracking error are met. However tuning control parameters to meet control design requirement is beyond the scope of this paper. Finally the kernel matrix is typically initialized as a identity matrix scaled by a small factor.

V. Control Law Evaluation

This section presents the results of the iADP controller implementation on Cessna Citation II aircraft. The results are simulated in MatLab/Simulink environment using a Cessna Citation model that simulates the data required for the controller evaluation. The simulations are performed at 100 Hz sampling frequency using Heun's second order fixed step solver. As data is sampled from sensors at fixed time steps in real time applications, using a fixed time solver is ideal to evaluate the controller learning performance. Unless specified the simulations are started from a steady straight and level trim flight condition which is a wings-level constant flight path condition. The trimming conditions of the aircraft at the start of the simulation are provided in Tables 3 and 4. PLA stands for power lever angle and an equal constant throttle power is provided to left and right engines.

Table 3 State Trim conditions

State	$p[°/s]$	$q[°/s]$	$r[°/s]$	$V_{tas}[m/s]$	$\alpha[°]$	$\beta[°]$	$\phi[°]$	$\theta[°]$	$\psi[°]$	$h_e[m]$	$x_e[m]$	$y_e[m]$
Value	0	0	0	90	3.76	0	0	3.76	0	2000	0	0

Table 4 Actuator Trim conditions

Actuator	$\delta_a[°]$	$\delta_e[°]$	$\delta_r[°]$	$PLA_{1,2}$
Value	0	−1.727	0	0.6335

A. iADP-FS Online Rate Control

The simulation results for combined Online rate control design using iADP full state feedback controller are presented in this section. The control law is designed based on the procedure discussed in section IV.A and clean sensor measurements are assumed. The hyperparameters are tuned according the principles mentioned before. The parameters for incremental model identification and iADP controller used in this simulation are presented in Tables 5 and 6 respectively. The forgetting factor γ_{RLS} is chosen to be one to provide better convergence of model parameters. The weighting matrices are manually tuned such that a satisfactory controller performance can be achieved. Typically values of weighting matrices R are fixed and Q are varied to achieve satisfactory performance. Higher weightage is given to the yaw rate control to maintain zero side slip to avoid adverse yaw. The task of the controller is two fold. Firstly an incremental model of the aircraft has to be identified online ensuring that the aircraft retains the steady state flying condition. Secondly aircraft should learn the control parameters online and perform a defined flight maneuver and achieve satisfactory tracking performance. The flight maneuver involves a combined longitudinal and lateral motion, where the aircraft performs a bank to bank roll maneuver and tracking a 3211 reference in the longitudinal direction, while ensuring coordinated turn using rudder. The necessary pilot commands to achieve this flight maneuver are simulated which are then fed to the manual control loop. The manual control loop provides the set points of the reference signals to be tracked for the automatic inner loop by ensuring that the aircraft stays within safe flight envelope and the reference commands are achievable by the inner loop.

Figures 6 and 7 shows the time responses and control inputs(u_{act}) acting on the aircraft. During the first 30 seconds, control inputs are generated such that online incremental model can be identified. Firstly, a 3211 input is commanded by elevator to excite short period dynamics. After the aircraft has reached steady state then the lateral dynamics are excited

Table 5 Parameters for Incremental Model Identification

Parameter	γ^{RLS}	Cov_0	Θ_0
Value	1	$1000I$	0

Table 6 Controller tuning parameters

Parameter	$diag(Q_p, Q_q, Q_r)$	$diag(R_{\delta_a}, R_{\delta_e}, R_{\delta_r})$	γ	P_0
Value	$diag(80, 100, 200)$	$diag(1, 1, 0.25)$	0.4	$0.001I$

through a banking motion by commanding a pulsed input through aileron and dutch roll motion is excited through doublet by rudder. As the reference model also needs to be identified online reference signals are provided during this period. The controller is then activated at t = 30 seconds and the parameters of the kernel matrix are updated online from there after. The iADP controller needs data over a window period of t_{ol} before the parameters of the kernel matrix are updated. A window of 20 seconds is chosen in this case where the data is collected during this window viz,. $\bar{V}(x)$, $\bar{X}^{kr}$and used to update the kernel matrix P at every time step according to equation (48). From the time responses we can see that the controller is able learn the policy online using data collected from just 2000 samples in the 20 second window period and is able to track the reference signals with satisfactory performance. The small oscillations observed at the control surface is due to the persistent excitation which is required for continuous online learning. Also the small peaks at control surfaces are visible at t = 30 seconds due to the kernel matrix update however the deflections are found to be within the actuator limits. High control activity in the rudder can be seen which is due to less weight($R_{\delta r}$) given to rudder. Observing the pitch attitude rate tracking response, we can see that the controller is able to adapt to time varying reference signals. Higher aileron control activity can be seen whenever there is a non zero pitch rate command, implying the controller is able to learn the coupling effects as the designed controller does not assume a decoupled controller for longitudinal and lateral dynamics.

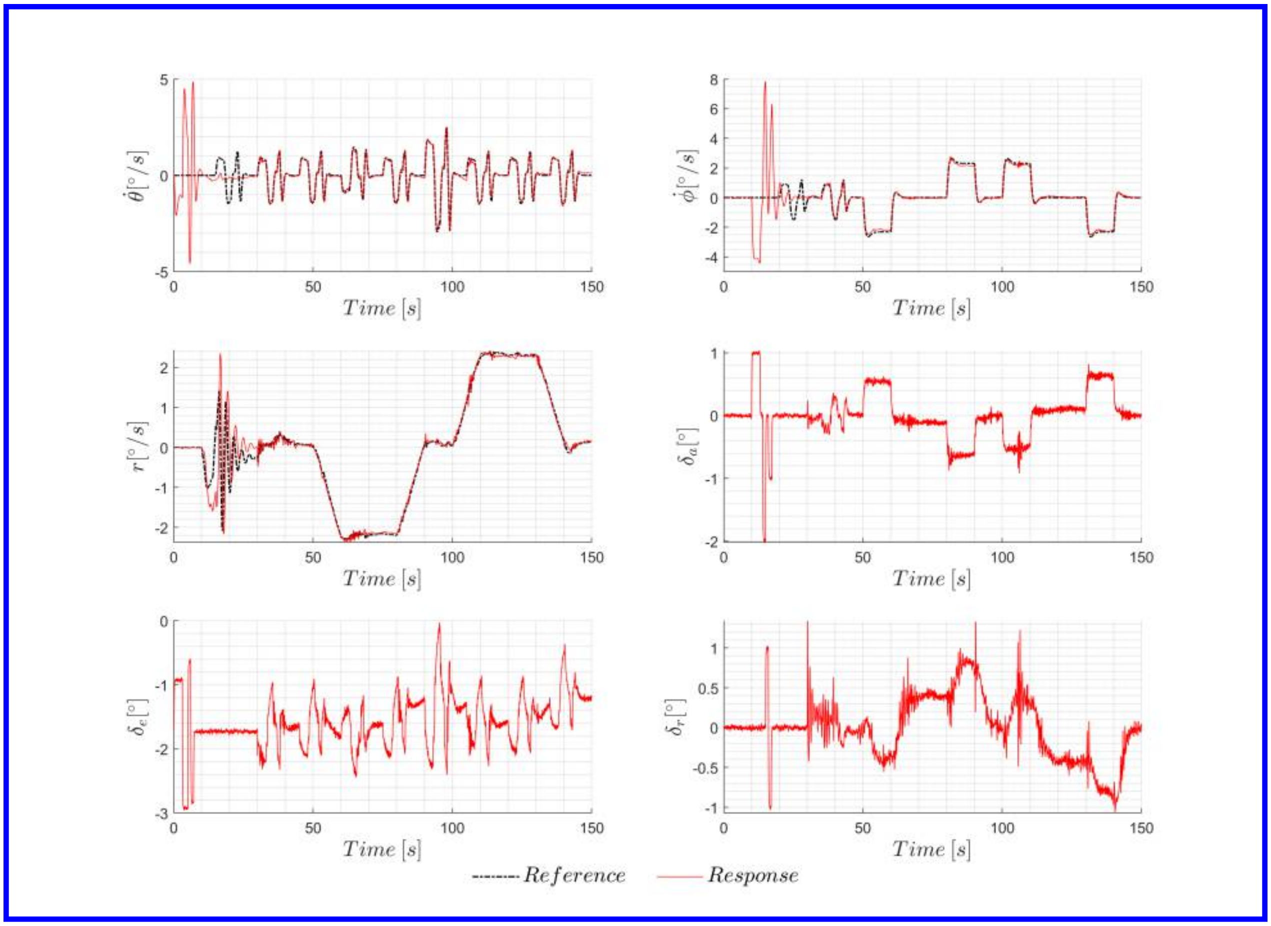

Fig. 6 Time responses of the Cessna Citation with combined iADP Online rate control using Full State Feedback

From Fig. 7 we can see that the aircraft is able to perform bank to bank(±20[°]) manoeuvre while restricting the side slip angle to value ±0.5[°]. Safety critical parameters like α and load factor n_z are found to be within safety limits throughout the flight maneuver including the initial model identification phase. Good longitudinal tracking response is seen even when the velocity conditions are changing which might be due to the online learning aided with persistent excitation. However for large changes in velocity we might have to excite the dynamic modes of the aircraft again if any performance degradation is observed.

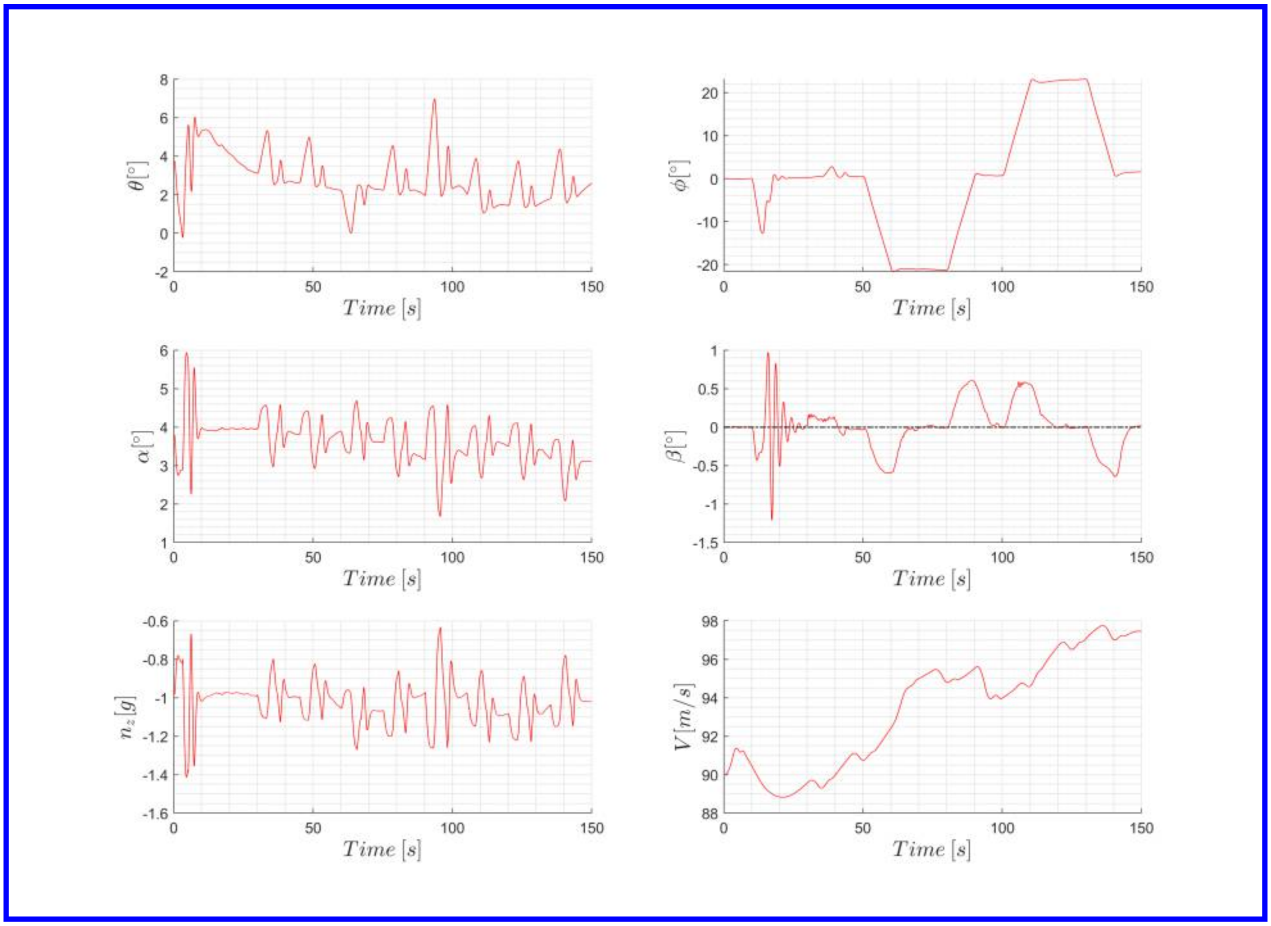

Fig. 7 Time responses of the Cessna Citation with combined iADP Online rate control using Full State Feedback

Fig. 8 shows the evolution of incremental model parameters viz,. state transition matrix F_t, control effectiveness matrix G_t and diagonal values of the kernel matrix parameters. The incremental model parameters have converged after the initial model identification phase of 30 seconds. We can see the effect of different modes of aircraft being excited at different times from the diagram. The kernel matrix parameters have converged within a short time after activating the kernel matrix at 30 seconds. However some of the parameters are fluctuating during the flight maneuver which might be because the controller is finding the need to update its policy for changing flight conditions and the coupling effects not encountered before.

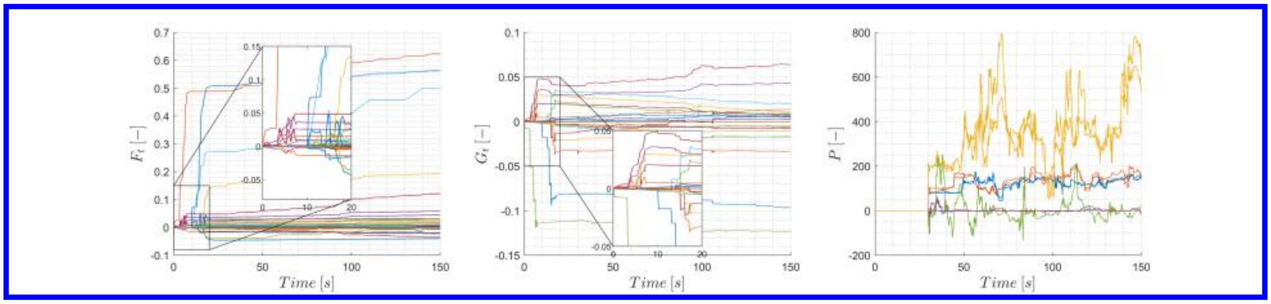

Fig. 8 Evolution of model coefficients and kernel matrix parameters during online learning

B. iADP-FS Online Longitudinal Rate Control

To assess the effect of sensor dynamics a simple longitudinal control task is considered. The time response and the control input is shown in Fig. 9. First 25 seconds is the model identification phase through short period dynamics excitation and the controller is activated after 25 seconds. The evolution of model coefficients and kernel matrix parameters are shown in Fig. 10. The model coefficients have converged after 10 seconds and for the remainder of this section the converged model coefficients will be considered as a measure to evaluate the model identification. Controller performance is assessed by considering three metrics viz, Root Mean Square Error (RMSE) between reference and actual pitch attitude rate, max absolute elevator deflection angle($max(\delta_e)$) and max elevator deflection rate($max(\dot{\delta_e})$). The rate saturation limit of the elevator is 20 deg/s. To minimize the transient effects of controller learning process, these metrics are evaluated between 40 - 80 seconds period.

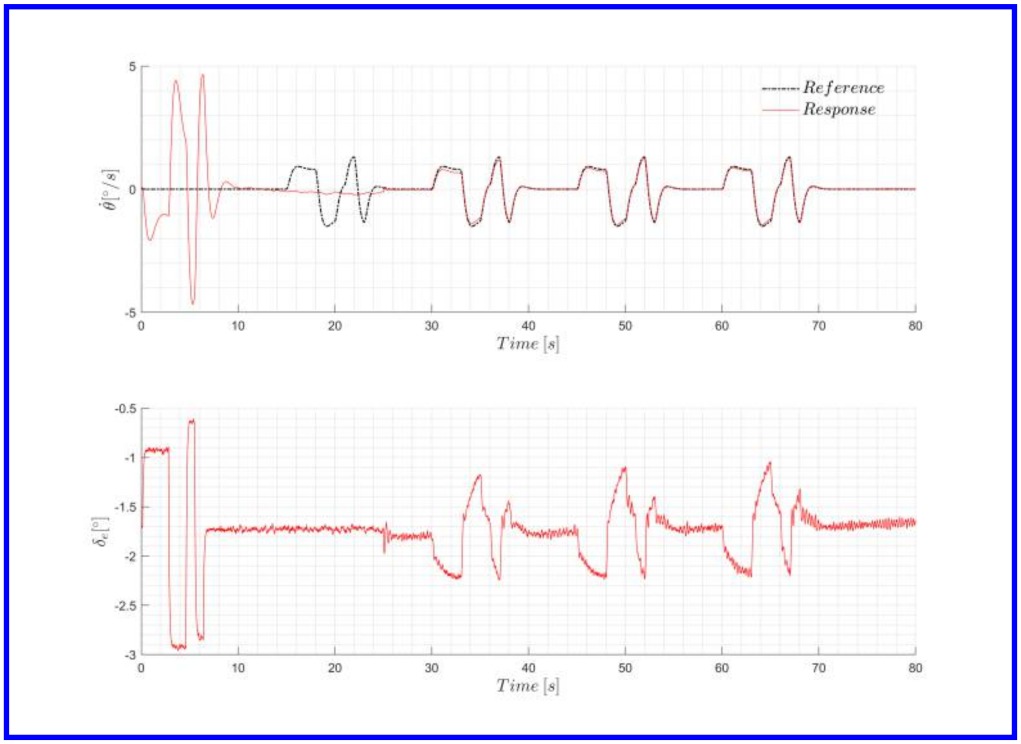

Fig. 9 Time responses of the Cessna Citation with longitudinal iADP Online rate control using Full State Feedback

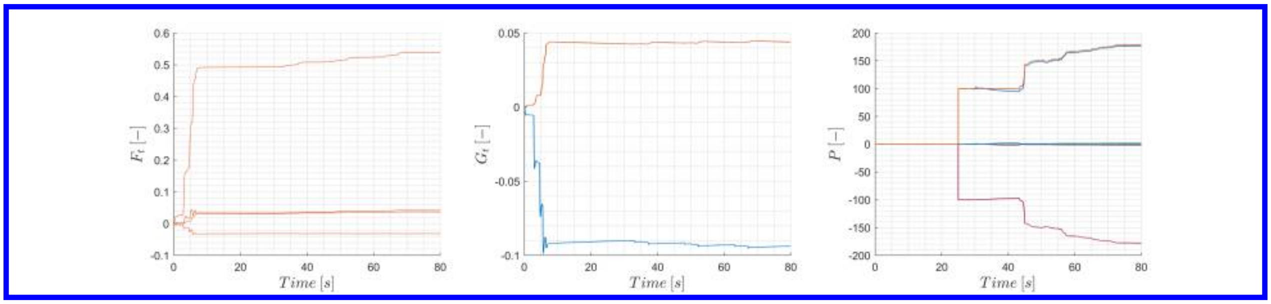

Fig. 10 Evolution of model coefficients and kernel matrix parameters during online learning

1. Selection of Weighting matrices

Before studying the effect of sensor dynamics a robustness analysis of controller is performed to assess the stability of the controller against change in weighting parameters. Fig. 11 shows the variation of the controller performance metrics against the weighting matrices. It is clear that the tracking error is minimum for higher values of Q and lower values of R. However the tracking error seems to increase after a certain limit indicating oscillatory behaviour after certain threshold. This is confirmed from the elevator deflection rate graph as the actuator rate saturation limits are found to have reached beyond this threshold. For constant values of Q an increase in R will reduce the control activity but will increase the RMSE. For constant values of R an increase in Q results in less RMSE but it is also increasing the control activity. To ensure stability, lower values of Q and higher values of R are preferred.

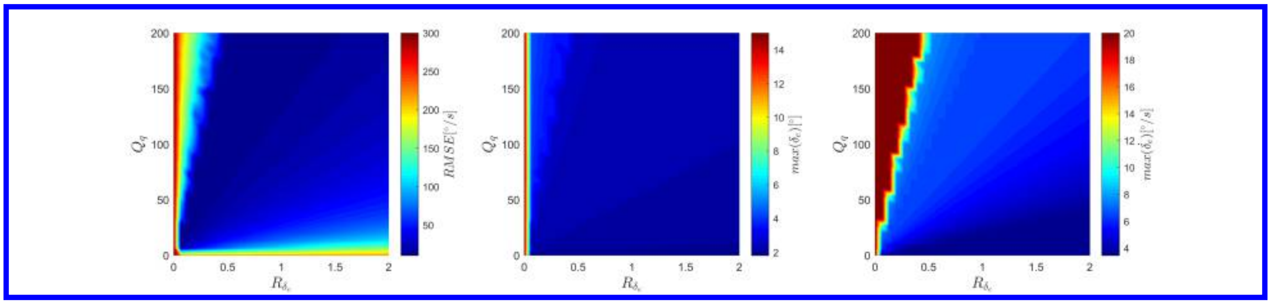

Fig. 11 Effect of weighting matrices on tracking performance and control activity

2. Real world phenomenon investigation

Before studying the effects of sensor dynamics the weighting matrices are readjusted to Q = 20 and R = 1 so that influence of weighting matrices is minimized during this analysis. The effect of sensor dynamics on the performance metric namely RMSE, actuator maximum deflection and rate and converged model parameters are evaluated which is listed in Table 7.

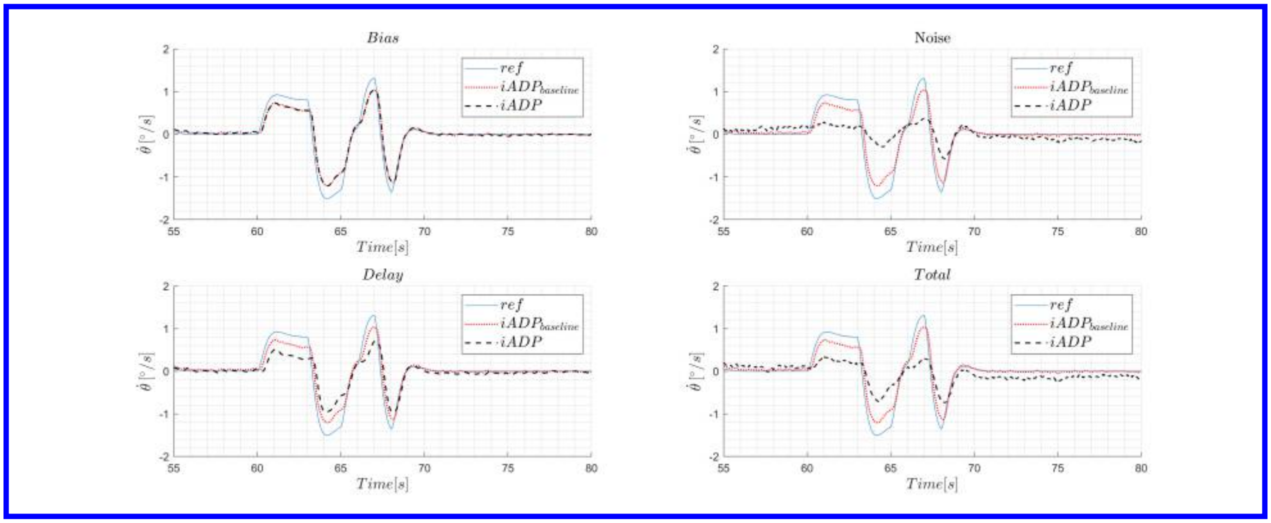

Fig. 12 Effect of sensor dynamics on tracking performance

For a comparison a baseline controller performance is included where sensor dynamics are neglected. Each phenomenon is considered separately based on one factor at a time method. Finally controller performance is evaluated considered combined phenomenon referenced as Total in the table. Comparing with baseline performance, we can see that the controller performance is not effected by discretization. The model parameters are identified without any difference. The effect of sensor bias on controller performance is also minimal, however small changes in the model parameters in the presence of bias is observed. Transport delay has also minimal effect on the controller performance. Noise and delays are found to degrade the controller performance and RLS algorithm is unable to identify the model parameters in the presence of noise/delays. These effects can be visualized from Fig. 12.

Note that the baseline performance is not optimal as the weighting matrices are retuned such that the controller performance is stable. Noise has degraded controller performance considerably. In the presence of delays, although the

Table 7 Effect of Sensor dynamics on Controller performance and Model Identification

	$RMSE[^\circ/s]$	$max(\delta_e)[^\circ]$	$max(\dot{\delta_e})[^\circ/s]$	$Ft[-]$	$Gt[-]$
Baseline	57.45	2.11	3.58	[0.492, -0.032, 0.034, 0.032]	[-0.092, 0.044]
Discretization	57.27	2.11	3.60	[0.492, -0.032, 0.034, 0.032]	[-0.092, 0.044]
Bias	58.73	2.09	3.55	[0.492, -0.034, 0.034, 0.035]	[-0.089, 0.045]
Noise	193.75	1.85	4.47	[0.410, -0.006, 0.016, -0.015]	[-0.046, -0.064]
Sensor Delay	110.82	2.04	3.46	[0.501, -0.011, 0.026, 0.032]	[-0.035, 0.044]
Transport Delay	55.21	2.11	3.50	[0.492, -0.032, 0.034, 0.032]	[-0.093, 0.045]
Total	170.70	1.90	2.40	[0.409, -0.010, 0.017, -0.017]	[-0.049, -0.063]

controller performance has degraded, it is observed that the controller is still trying to track the reference signal but with higher tracking error. Oscillatory behaviour can also be observed.

Table 8 Effect of Sensor dynamics on Controller performance and Model Identification with filtering

	$RMSE[^\circ/s]$	$max(\delta_e)[^\circ]$	$max(\dot{\delta_e})[^\circ/s]$	$Ft[-]$	$Gt[-]$
Baseline	57.53	2.11	4.16	[0.481, -0.030, 0.034, 0.032]	[-0.092, 0.044]
Discretization	57.51	2.11	4.15	[0.481, -0.030, 0.034, 0.032]	[-0.092, 0.044]
Bias	59.19	2.09	4.08	[0.481, -0.032, 0.035, 0.035]	[-0.088, 0.045]
Noise	61.49	2.10	4.03	[0.480, -0.026, 0.034, 0.032]	[-0.091, 0.042]
Sensor Delay	94.77	2.12	3.73	[0.488, -0.014, 0.027, 0.033]	[-0.046, 0.047]
Transport Delay	57.87	2.11	4.05	[0.481, -0.030, 0.034, 0.032]	[-0.092, 0.044]
Total	101.30	2.09	2.94	[0.487, -0.011, 0.027, 0.036]	[-0.045, 0.046]

To mitigate the effect of noise, signals are filtered using a first order low pass filter. Table 8 lists the effect of filtering on the controller performance. Using filtering, the controller performance degradation has been reduced considerably and RLS is able to learn the model parameters with improved accuracy. The improvement in the controller performance with filtering is shown in the time response plots in Fig. 13.

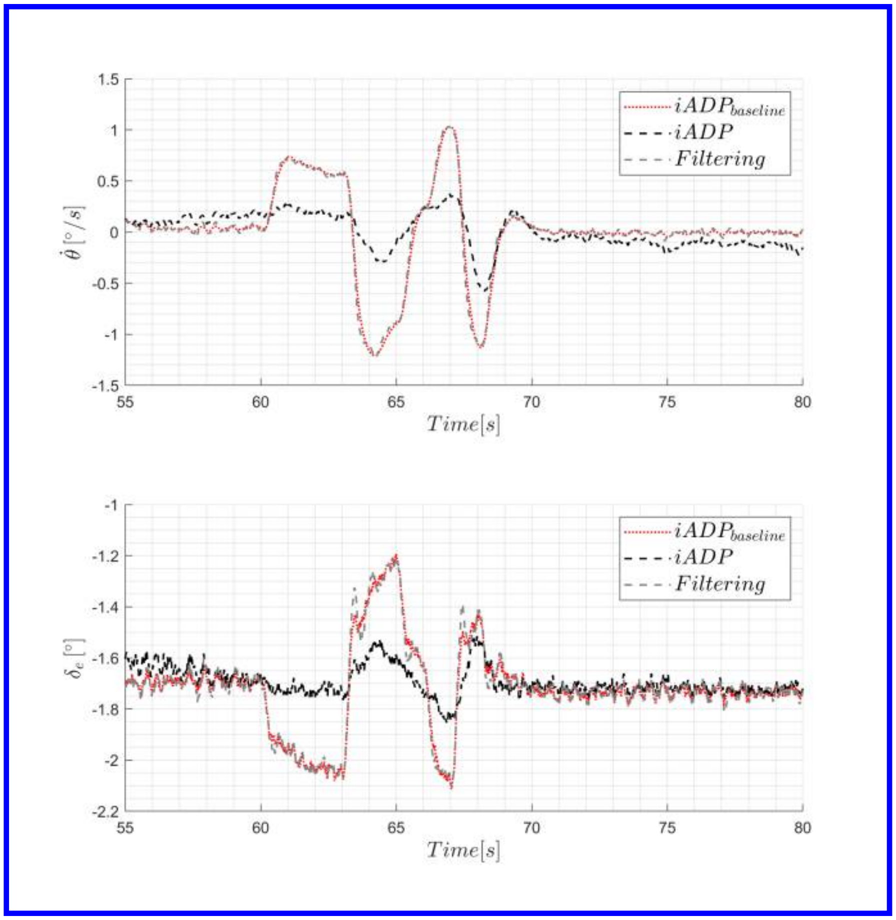

Fig. 13 Effect of Filtering on tracking performance

C. iADP-OPFB Longitudinal Rate Control

This section presents the results of iADP-OPFB controller evaluation to achieve longitudinal rate control of Citation II. As mentioned before, full state information is not provided to the controller and the controller has to achieve longitudinal tracking by using the input output measurements over a certain time horizon. Recalling the methodology from Section II.A.3, N=2 samples is used to construct the state. The weighting matrices($Q = 10, R = 1$) are tuned to achieve satisfactory performances. The results are summarized in Fig. 5. The model is identified using a 3211 maneuver in every episode and the objective of the controller is to track the 3211 signals commanded by the pilot. The evolution of kernel matrix parameters over episodes is shown in Fig. 15. The parameters have converged after 13 iterations. The controller performance using the converged kernel matrix parameters for tracking is shown in Fig. 14. The controller is able to track the reference in the presence of noisy signal measurements and time delays when state information is not provided to the controller. Small steady state errors are visible which might due to the sensor bias and as the full state information is not available for this controller the effects of bias is higher compared to the controller where full state feedback is available.

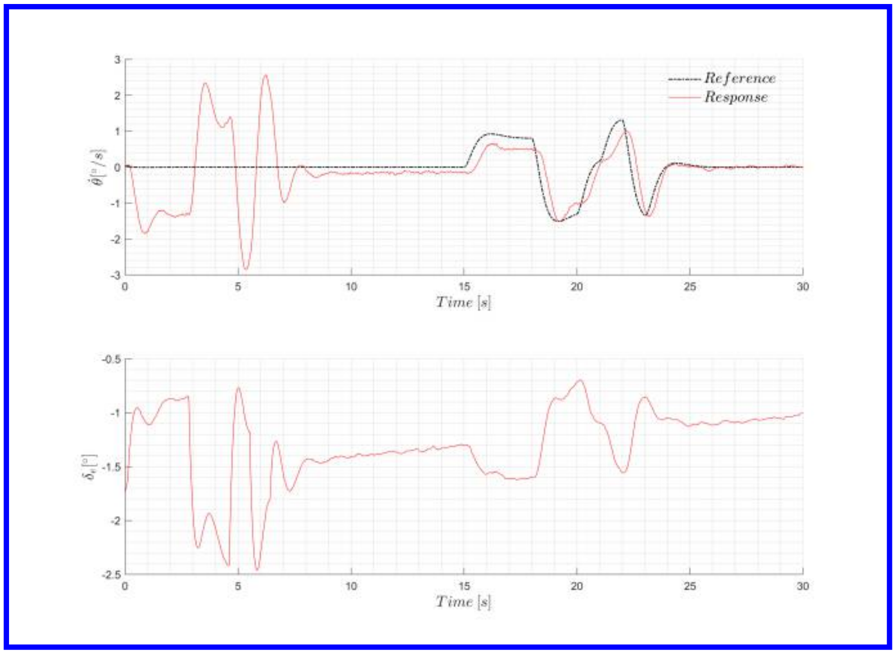

Fig. 14 Time responses of Cessna Citation with longitudinal iADP rate control using Output Feedback

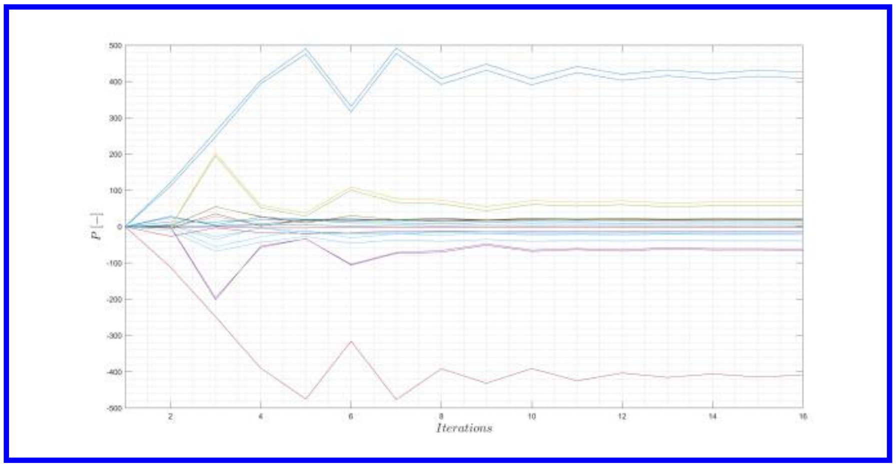

Fig. 15 Evolution of kernel matrix parameters of iADP rate control using Output Feedback

VI. Conclusion

This paper presents design of Incremental Approximate Dynamic Programming based flight control law for Cessna Citation II aircraft. The iADP controller is designed to achieve automatic online rate control to track pilot commands via setpoints provided by the manual outer loop. The incremental model necessary for the iADP controller is identified online through appropriate system excitation using control surfaces. The simulation results show that the combined rate control using full state feedback is able to learn online to control the aircraft without any knowledge of the system but just using the data collected along the system trajectories. To assess the controller performance in the presence of sensor dynamics and actuator dynamics, an analysis is carried out to identify causes of any performance degradation. The simulation results from iADP longitudinal control design using full state feedback indicate that the discretization of sensor signals, sensor bias and transport delays did not have any significant effect on the controller performance or on the incremental model identification while noisy signals and delays in sensors are found to effect the controller performance. The noisy signals resulted in incorrect estimation of incremental model parameters affecting the controller performance.

It is observed that appropriate filtering of signals resulted in better estimation of the incremental model subsequently improving the controller performance. Sensor delays also resulted in incorrect estimation of model parameters, however the controller is found to track the reference inspite of the incorrect incremental model parameters but with reduced tracking performance. Finally an iADP controller using output feedback is designed to achieve longitudinal control in the absence of full state information. The controller is trained offline due to higher learning complexity and performance is evaluated in the presence of sensor and actuator dynamics. The results from output feedback method show the controller can achieve satisfactory tracking control but with reduced tracking performance.

For a successful implementation of iADP controller on Cessna Citation II aircraft, further research needs to be conducted to validate this controller on a real system through flight tests. The effect of sensor delays on controller performance should be investigated in future by conducting stability and robustness analysis as they are found to degrade the controller performance.

References

[1] Belcastro, C. M., Foster, J. V., Newman, R. L., Groff, L., Crider, D. A., and Klyde, D. H., "Aircraft Loss of Control: Problem Analysis for the Development and Validation of Technology Solutions," *AIAA Guidance, Navigation, and Control Conference*, 2016. doi:10.2514/6.2016-0092, URL `https://arc.aiaa.org/doi/abs/10.2514/6.2016-0092`.

[2] Sieberling, S., Chu, Q. P., and Mulder, J. A., "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, 2010, pp. 1732–1742. doi:10.2514/1.49978, URL `https://doi.org/10.2514/1.49978`.

[3] Acquatella, P., Van Kampen, E.-J., and Chu, Q., "Incremental backstepping for robust nonlinear flight control," 2013.

[4] Grondman, F., Looye, G., Kuchar, R. O., Chu, Q. P., and Kampen, E.-J. V., "Design and Flight Testing of Incremental Nonlinear Dynamic Inversion-based Control Laws for a Passenger Aircraft," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018. doi:10.2514/6.2018-0385, URL `https://arc.aiaa.org/doi/abs/10.2514/6.2018-0385`.

[5] Keijzer, T., Looye, G., Chu, Q. P., and Kampen, E.-J. V., "Design and Flight Testing of Incremental Backstepping based Control Laws with Angular Accelerometer Feedback," *AIAA Scitech 2019 Forum*, 2019. doi:10.2514/6.2019-0129, URL `https://arc.aiaa.org/doi/abs/10.2514/6.2019-0129`.

[6] Pollack, T., Looye, G., and Linden, F., "Design and flight testing of flight control laws integrating incremental nonlinear dynamic inversion and servo current control," 2019. doi:10.2514/6.2019-0130.

[7] Bertsekas, D., *Dynamic Programming and Optimal Control*, No. Bd. 2 in Athena Scientific optimization and computation series, Athena Scientific, 2012. URL `https://books.google.de/books?id=H-PSMwEACAAJ`.

[8] Lewis, F., and Vrabie, D., "Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control," *Circuits and Systems Magazine, IEEE*, Vol. 9, 2009, pp. 32 – 50. doi:10.1109/MCAS.2009.933854.

[9] Lewis, F. L., and Vamvoudakis, K. G., "Reinforcement Learning for Partially Observable Dynamic Processes: Adaptive Dynamic Programming Using Measured Output Data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 41, No. 1, 2011, pp. 14–25. doi:10.1109/TSMCB.2010.2043839.

[10] Kiumarsi, B., Lewis, F. L., Naghibi-Sistani, M., and Karimpour, A., "Optimal Tracking Control of Unknown Discrete-Time Linear Systems Using Input-Output Measured Data," *IEEE Transactions on Cybernetics*, Vol. 45, No. 12, 2015, pp. 2770–2779. doi:10.1109/TCYB.2014.2384016.

[11] van Kampen, E. J., Chu, Q. P., and Mulder, J. A., "Continuous Adaptive Critic Flight Control Aided with Approximated Plant Dynamics," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2006. doi:10.2514/6.2006-6429.

[12] Dias, P. M., Zhou, Y., and Kampen, E.-J. V., *Intelligent Nonlinear Adaptive Flight Control using Incremental Approximate Dynamic Programming*, 2019. doi:10.2514/6.2019-2339, URL `https://arc.aiaa.org/doi/abs/10.2514/6.2019-2339`.

[13] Lewis, F. L., and Vamvoudakis, K. G., "Optimal adaptive control for unknown systems using output feedback by reinforcement learning methods," *2010 8th IEEE International Conference on Control and Automation, ICCA 2010*, 2010, pp. 2138–2145. doi:10.1109/ICCA.2010.5524211.

[14] Bertsekas, D., and Tsitsiklis, J., *Neuro-Dynamic Programming*, Vol. 27, 1996. doi:10.1007/978-0-387-74759-0_440.

[15] Bellman, R., "Dynamic Programming," *Science*, Vol. 153, No. 3731, 1966, pp. 34–37. doi:10.1126/science.153.3731.34, URL `https://science.sciencemag.org/content/153/3731/34`.

[16] Bradtke, S., Ydstie, B., and Barto, A., "Adaptive Linear Quadratic Control Using Policy Iteration," *Proceedings of the American Control Conference*, Vol. 3, 1994. doi:10.1109/ACC.1994.735224.

[17] Heyer, S., Kroezen, D., and Van Kampen, E. J., "Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft," *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, 2020.

[18] Zhou, Y., van Kampen, E. J., and Chu, Q. P., "Incremental model based online dual heuristic programming for nonlinear adaptive control," *Control Engineering Practice*, Vol. 73, 2018, pp. 13–25. doi:10.1016/j.conengprac.2017.12.011.

[19] Jategaonkar, R., *Flight Vehicle System Identification: A Time Domain Methodology*, Vol. 216, 2006. doi:10.2514/4.866852.

[20] Zhou, Y., van Kampen, E.-J., and Chu, Q., "Incremental Approximate Dynamic Programming for Nonlinear Adaptive Tracking Control with Partial Observability," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 12, 2018, pp. 2554–2567. doi:10.2514/1.G003472, URL `https://doi.org/10.2514/1.G003472`.

[21] "EASA.IM.A.207: Cessna 500, 550, S550, 560 and 560XL," , 2019. URL `https://www.easa.europa.eu/documents/type-certificates/aircraft-cs-25-cs-22-cs-23-cs-vla-cs-lsa/easaima207`.

[22] Oliveira, J., "DEVELOPMENT OF A FLEXIBLE FLIGHT TEST INSTRUMENTATION SYSTEM," 2006.

[23] Zaal, P., Pool, D., Postema, F., Veld, A., Mulder, M., Van Paassen, M. M., and Mulder, J., "Design and Certification of a Fly-By-Wire System with Minimal Impact on the Original Flight Controls," 2009. doi:10.2514/6.2009-5985.

[24] der Linden, V., "DASMAT-Delft University Aircraft Simulation Model and Analysis tool," 1996.

[25] Hoek, M., De Visser, C., and Pool, D., *Identification of a Cessna Citation II Model Based on Flight Test Data*, 2018, pp. 259–277. doi:10.1007/978-3-319-65283-2_14.

[26] van 't Veld, R., Kampen, E.-J. V., and Chu, Q. P., "Stability and Robustness Analysis and Improvements for Incremental Nonlinear Dynamic Inversion Control," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018. doi:10.2514/6.2018-1127, URL `https://arc.aiaa.org/doi/abs/10.2514/6.2018-1127`.

[27] Söderström, T., and Stoica, P. (eds.), *System Identification*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[28] Vahidi, A., Stefanopoulou, A., and Peng, H., "Recursive least squares with forgetting for online estimation of vehicle mass and road grade: Theory and experiments," *Vehicle System Dynamics - VEH SYST DYN*, Vol. 43, 2005, pp. 31–55. doi:10.1080/00423110412331290446.

[29] Joos, H.-D., "A methodology for multi-objective design assessment and flight control synthesis tuning," *Aerospace Science and Technology*, Vol. 3, No. 3, 1999, pp. 161 – 176. doi:https://doi.org/10.1016/S1270-9638(99)80040-6, URL `http://www.sciencedirect.com/science/article/pii/S1270963899800406`.