

Use of Artificial Intelligence Techniques for the Control of Cessna Citation X

**Ruxandra Mihaela Botez, Full Professor and Canada Research Chair Holder Tier 1 in
Aircraft Modeling and Simulation New Technologies**

École de Technologie Supérieure,

Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity

LARCASE,

Montreal Quebec, H3C 1K3

ruxandra.botez@etsmtl.ca

**The 3rd International Conference on Aviation Future: Challenges and Solutions
AFCS 2025**

10-12 December 2025, Ho Chi Minh City, Vietnam



Use of Artificial Intelligence Techniques for the Control of Cessna Citation X

- 1. Problem and Objectives**
- 2. Cessna Citation X Simulation Platforms**
- 3. Pitch Controller Design Methodology**
- 4. Model Simulation and Validation Results**
- 5. Discussion and Conclusion**

1. PROBLEM AND OBJECTIVES

Objectives

- Develop a **nonlinear adaptive control** algorithm for **longitudinal control** of the Cessna Citation X
- Apply the **C* control** architecture
- Use **Neural Networks** for adaptive control and **validate for the overall flight envelope** of the Cessna Citation X

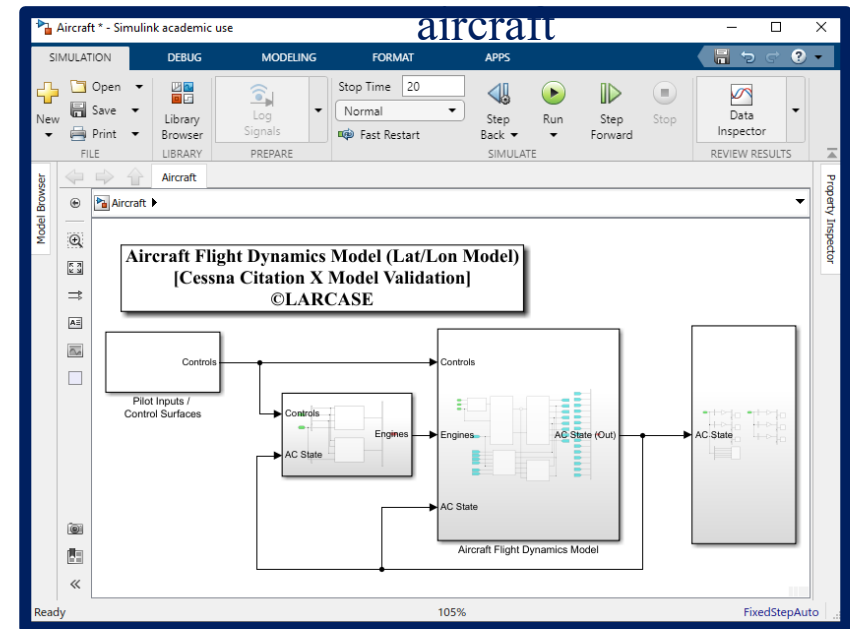
2. SIMULATION PLATFORMS

Research facilities

- Level D Flight simulator of the Cessna Citation X business aircraft



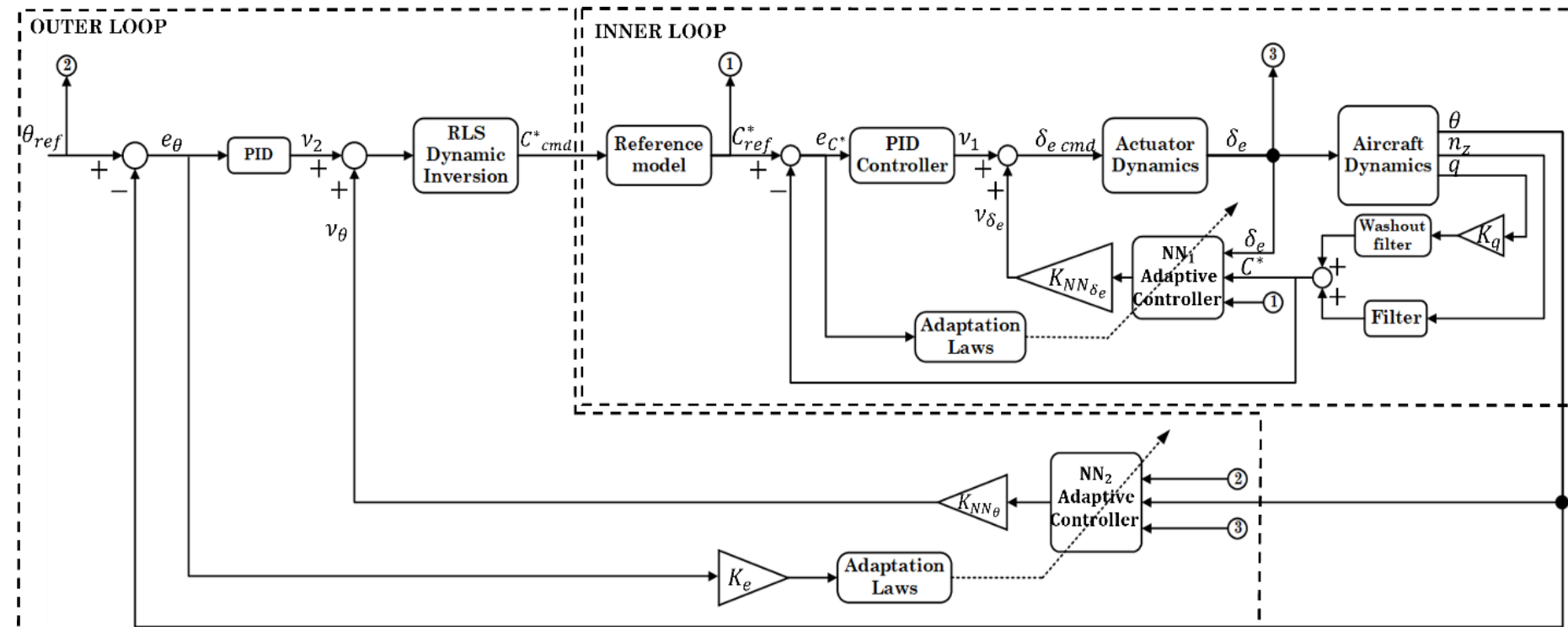
- High fidelity Simulink platform of the Cessna Citation X business



3. CONTROLLER DESIGN METHODOLOGY

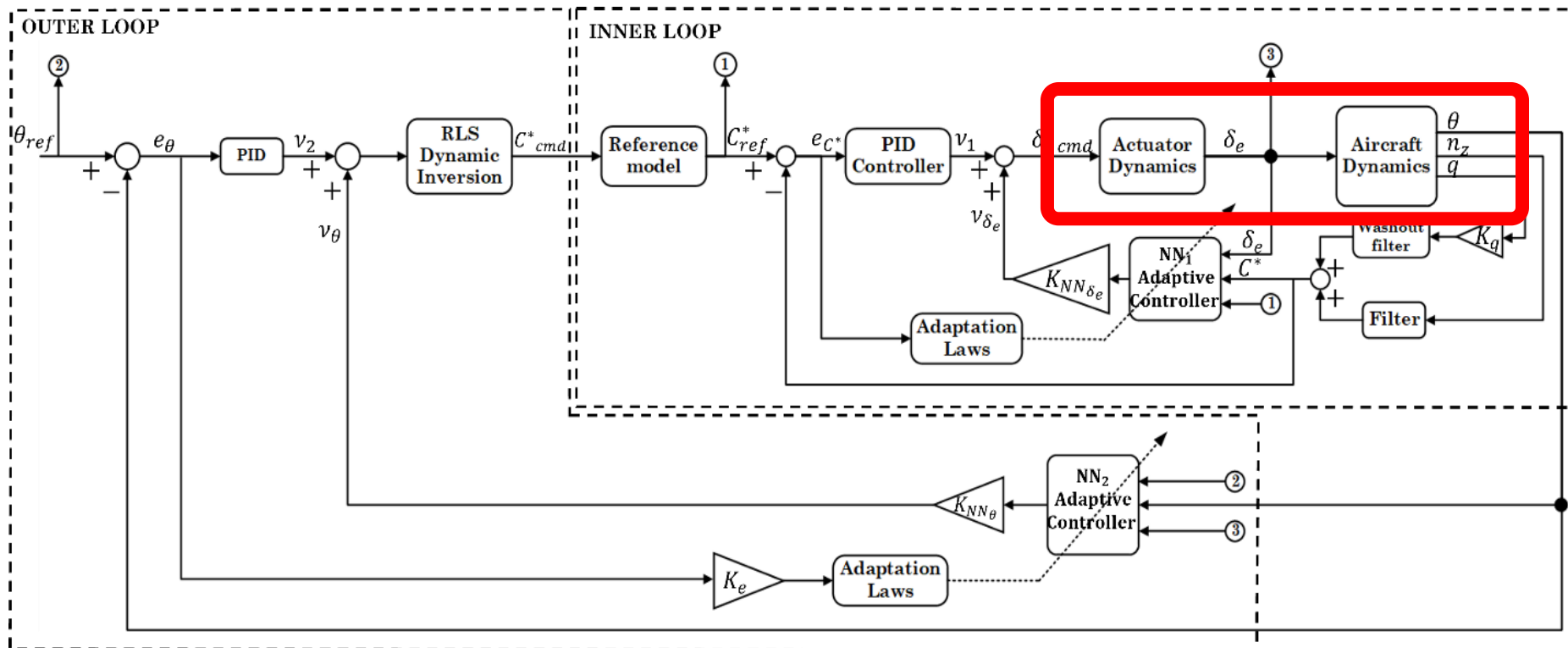
Pitch Control Architecture

- This architecture includes an **inner loop**, that controls the C^* parameter.
- Then, the **outer loop** controls the pitch angle θ , which follows a desired pitch reference model θ_{ref} . The design of the pitch reference model is described later and includes a **ramp modeling** that allows to limit the pitch rate q .



3. CONTROLLER DESIGN METHODOLOGY

Aircraft longitudinal dynamics – The C* inner loop for actuators dynamics



3. CONTROLLER DESIGN METHODOLOGY

Aircraft longitudinal dynamics

- State vector : **longitudinal speed, horizontal speed, pitch rate and pitch angle**

$$\mathbf{x} = [u \quad w \quad q \quad \theta]^T$$

- Control variable : **elevators**

$$\eta = \delta_e$$

- Output variable : **pitch rate**

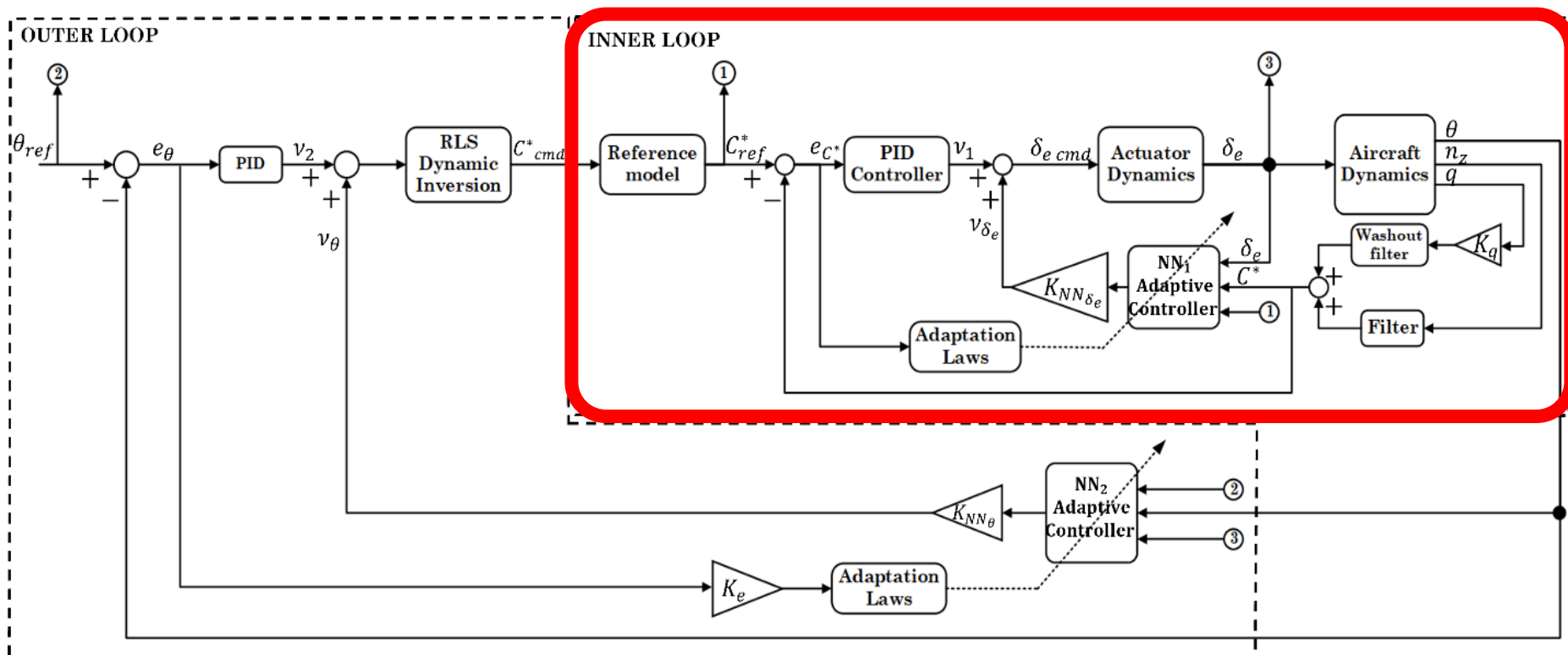
$$y = q = \mathbf{C} \cdot \mathbf{x}$$

Actuator dynamics

- **Second degree nonlinear transfer function**

3. CONTROLLER DESIGN METHODOLOGY

The C^* inner loop controller



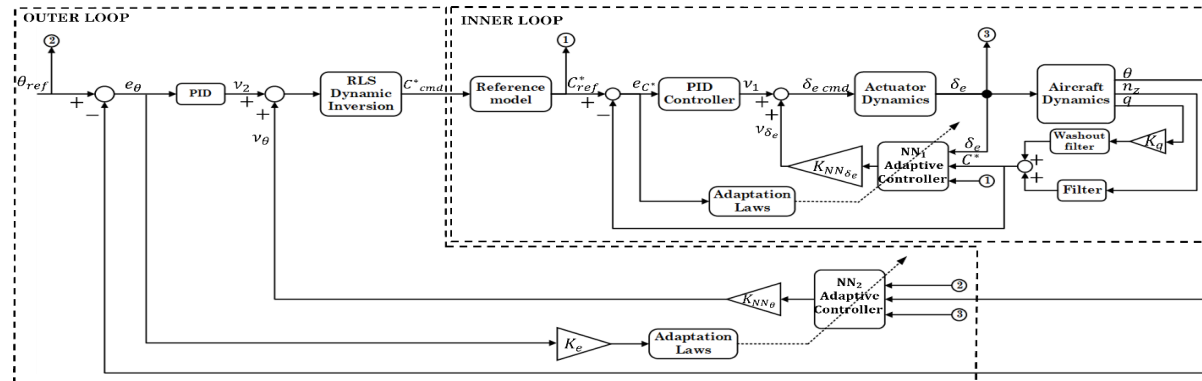
3. CONTROLLER DESIGN METHODOLOGY

The C^* inner loop controller

- The parameter C^* is defined as a combination of the **pitch rate** q and the **normal load factor** n_z

$$C^* = \Delta n_z + \beta \Delta q$$

- $\beta = \frac{V_c}{g_0}$, where V_c is the **crossover airspeed**. For most commercial aircraft, $V_c = 400$ ft/s, therefore $\beta = 12.4$
- The C^* control law allows **neutral static stability** at any time



3. CONTROLLER DESIGN METHODOLOGY

The C^* inner loop controller

- **C^* reference model** : second degree transfer function

$$\frac{C_{ref}^*}{C_{cmd}^*} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

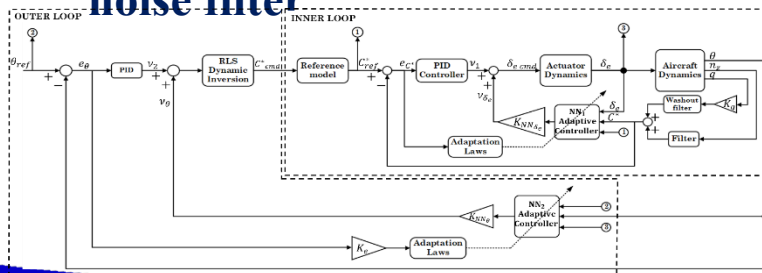
- **PID controller** with fixed gains

$$v_1 = \left(K_P + \frac{K_I}{s} + K_D s \right) e_{C^*}$$

- **Washout filter** : is a **high-pass filter** that rejects signals below the cutoff frequency

$$W(s) = \frac{s + a}{\tau_W s + b}$$

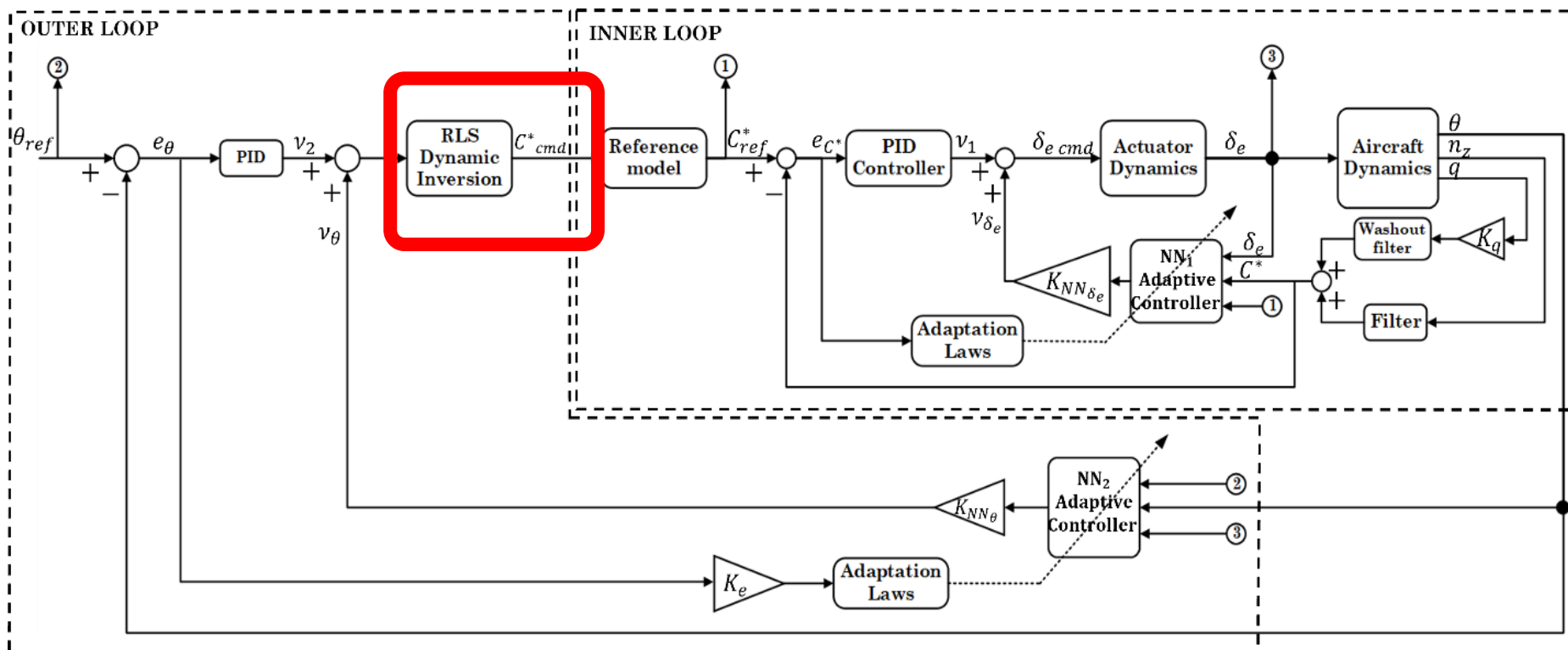
- **First Order Filter**: is a **low-pass filter** added to the load factor feedback and serves as a **noise filter**



$$N(s) = \frac{K}{\tau_N s + 1}$$

3. CONTROLLER DESIGN METHODOLOGY

The C^* inner loop controller - Recursive Least Square (RLS) Dynamic Inversion



3. CONTROLLER DESIGN METHODOLOGY

The C* inner loop controller

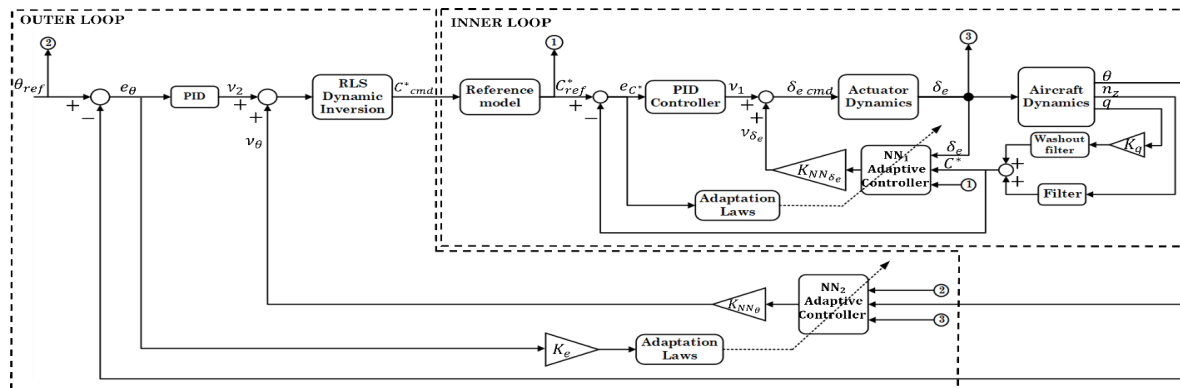
Nonlinear Dynamic Inversion - Recursive Least Square RLS algorithm

Aircraft nonlinear dynamics

- $\dot{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{x}, \boldsymbol{\eta})$ - state space equation with $\boldsymbol{\varphi}$ – nonlinear function; $\dot{\mathbf{x}}$ - variation of states; $\mathbf{x}$ – state vector; $\boldsymbol{\eta}$ – control input vector
- $y = h(\mathbf{x}) = \theta$ – pitch angle output equation

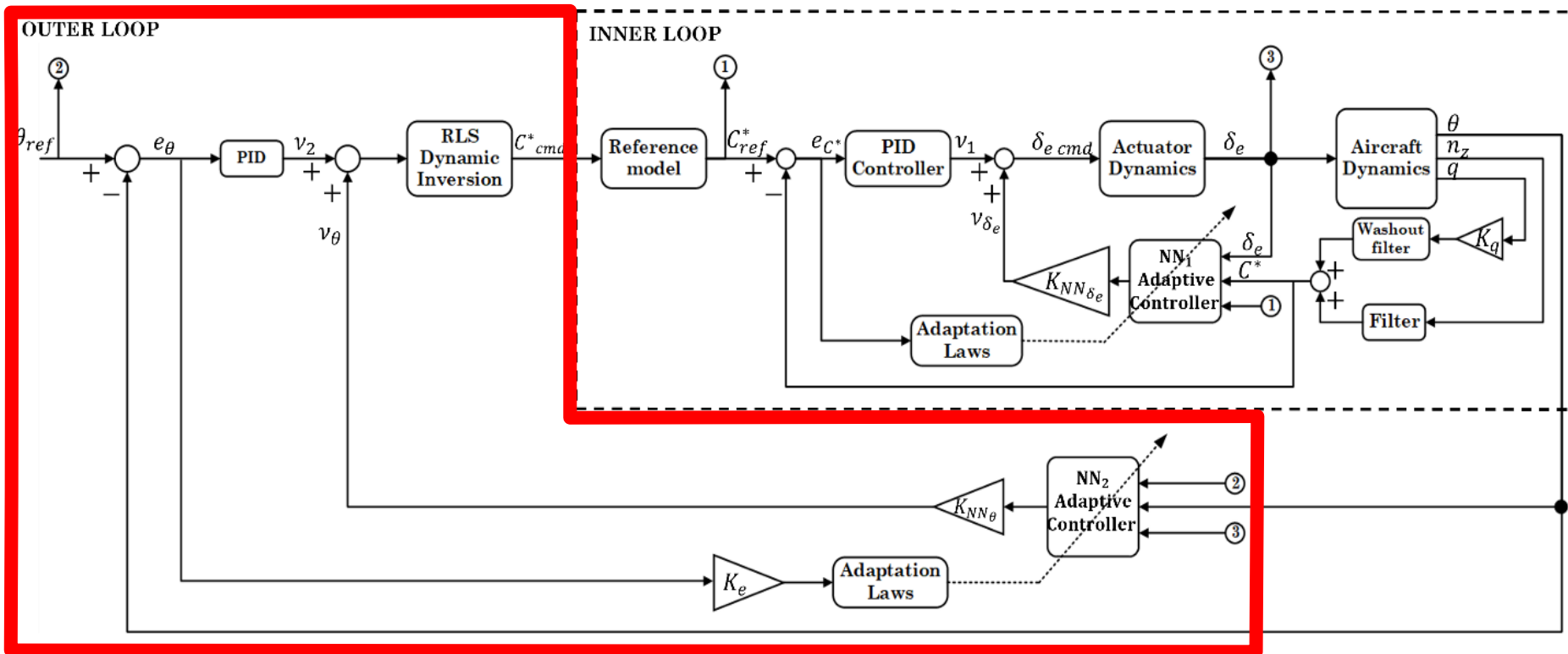
Nonlinear Dynamic inversion

- $\boldsymbol{\eta} = \boldsymbol{\varphi}^{-1}(\dot{\mathbf{x}}, \mathbf{x})$; with $\boldsymbol{\varphi}^{-1}$ – inverse function of $\boldsymbol{\varphi}$



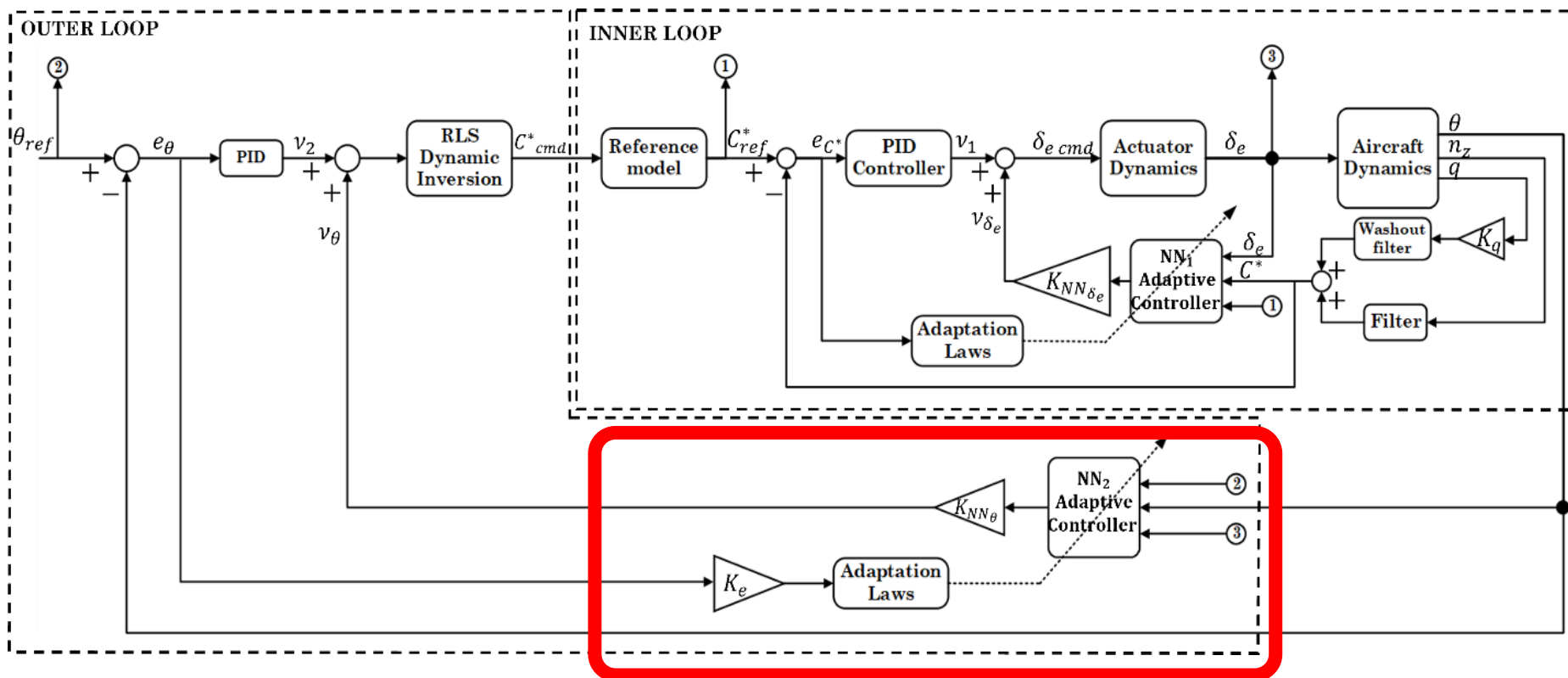
3. CONTROLLER DESIGN METHODOLOGY

Pitch Outer loop controller



3. CONTROLLER DESIGN METHODOLOGY

Pitch Adaptive Neural Network Controller



3. CONTROLLER DESIGN METHODOLOGY

Pitch Adaptive Neural Network Controller

In practice, using estimates may reduce the system's calculations, but it may also conduct to prediction errors ε .

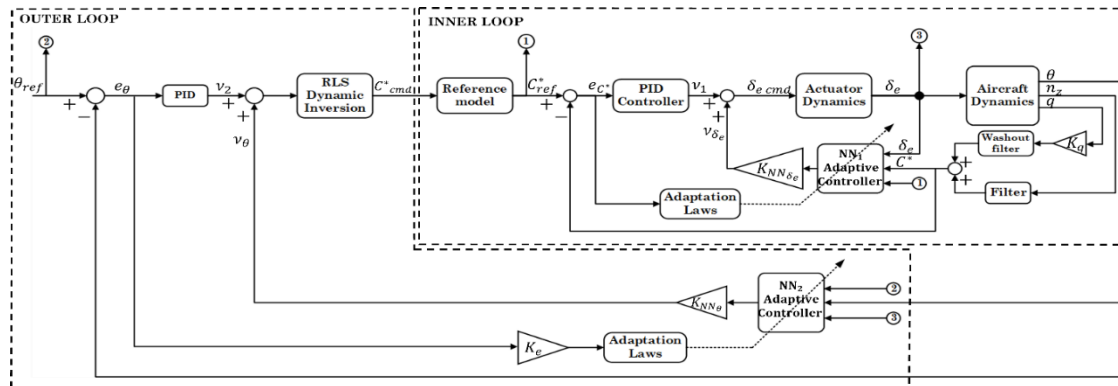
$$\triangleright \Delta \hat{\mathbf{y}}_{k+1} = \Delta \mathbf{y}_{k+1} - \varepsilon$$

ε may be compensated using adaptive elements such as **neural networks** so that :

$$\triangleright \varepsilon = v_\theta$$

Therefore, the control inputs may be written as follows:

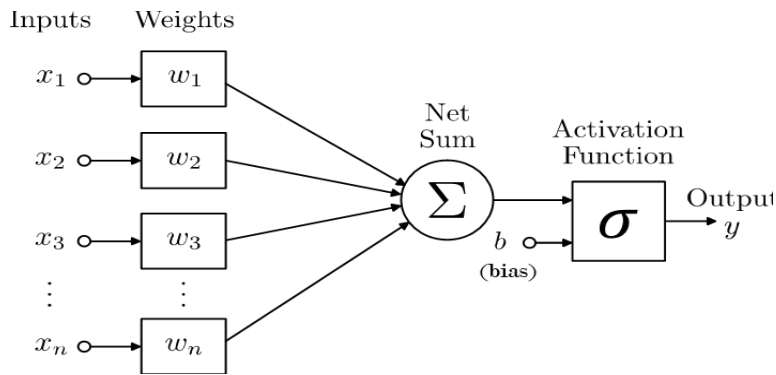
$$\triangleright \Delta \eta_k = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1} [\dot{r} + v_2 - \mathbf{C}\hat{\mathbf{F}}_k \Delta \mathbf{x} - \varepsilon]$$



3. CONTROLLER DESIGN METHODOLOGY

Pitch Adaptive Neural Network Controller

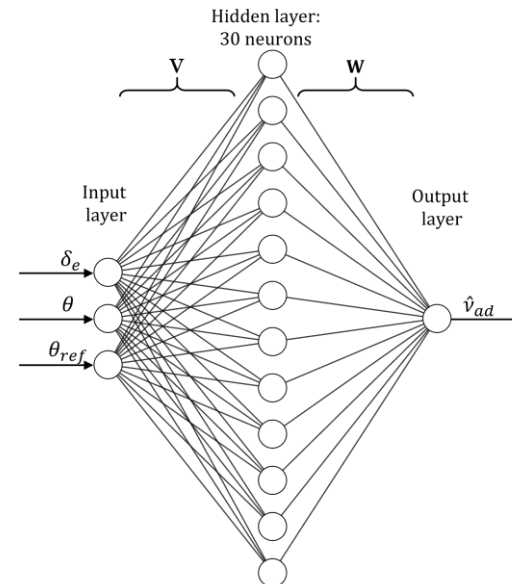
➤ Artificial neuron



➤ Activation function: *tangent sigmoid*

$$\sigma_i(z) = \frac{1}{1 + \exp(-z)}$$

➤ Feedforward Neural Network (FFNN)



➤ Number of hidden layer : 1

➤ Number of neurons : 30

3. CONTROLLER DESIGN METHODOLOGY

Pitch Adaptive Neural Network Controller

Adaptation Law are given on the weight updates, **Johnson and Calise, 2003**

- $\dot{\mathbf{W}} = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}'\mathbf{V}^T\bar{\mathbf{x}})e_\theta^T + \lambda\|e_\theta\|\mathbf{W}]\boldsymbol{\Gamma}_2$
- $\dot{\mathbf{V}} = -\boldsymbol{\Gamma}_2[\bar{\mathbf{x}}e_\theta^T\mathbf{W}^T\boldsymbol{\sigma}' + \lambda\|e_\theta\|\mathbf{V}]$

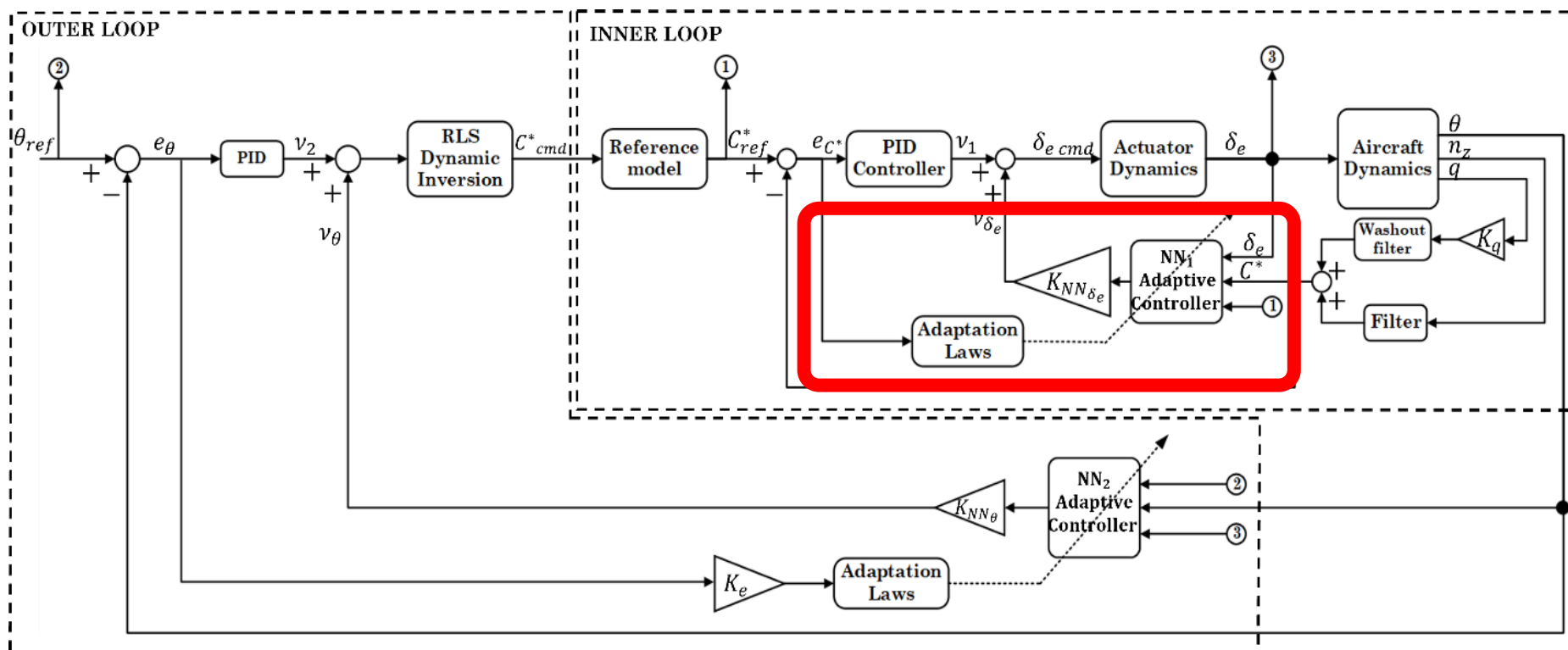
$\dot{\mathbf{W}}$, and $\dot{\mathbf{V}}$ are the variations of the weight matrices, $\boldsymbol{\Gamma}_2$ is the learning rates, $\boldsymbol{\sigma}'$ is the gradient of $\boldsymbol{\sigma}$

The adaptive control signal is then formulated as follows :

- $\hat{v}_{ad(n \times 1)} = \mathbf{W}_{(N \times n)}^T \boldsymbol{\sigma}_{(N \times N)} \left(\mathbf{V}_{(m \times N)}^T \bar{\mathbf{x}}_{2(m \times 1)} \right)$
- $\bar{\mathbf{x}}_2 = [\delta_e \ \theta \ \theta_{ref}]^T$ is the **NN2 input vector**, m is the number of network inputs, n is the number of layers and is equal to 1, and N is the number of neurons

3. CONTROLLER DESIGN METHODOLOGY

C* Adaptive Neural Network Controller



3. CONTROLLER DESIGN METHODOLOGY

C^* Adaptive Neural Network Controller

Analogically, the adaptation laws for the C^* Adaptive Neural Network controller may be written as follows :

- $\dot{\mathbf{W}} = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}'\mathbf{V}^T\bar{\mathbf{x}}_1)e_{C^*}^T + \lambda\|e_{C^*}\|\mathbf{W}]\boldsymbol{\Gamma}_1$
- $\dot{\mathbf{V}} = -\boldsymbol{\Gamma}_1[\bar{\mathbf{x}}e_{C^*}^T\mathbf{W}^T\boldsymbol{\sigma}' + \lambda\|e_{C^*}\|\mathbf{V}]$

$\dot{\mathbf{W}}$, and $\dot{\mathbf{V}}$ are the variations of the weight matrices, $\boldsymbol{\Gamma}_1$ is the learning rates, $\boldsymbol{\sigma}'$ is the gradient of $\boldsymbol{\sigma}$

The adaptive control signal is then formulated as follows

- $v_{\delta_e} = K_{NN\delta_e}\hat{v}_{ad} = K_{NN\delta_e}\mathbf{W}^T\sigma(\mathbf{V}^T\bar{\mathbf{x}}_1)$
- $\bar{\mathbf{x}}_1 = [\delta_e \ C^* \ C^*_{ref}]^T$ is the **NN1 input vector**, m is the number of network inputs, n is the number of layers and is equal to 1, and N is the number of neurons

3. CONTROLLER DESIGN METHODOLOGY

Overall Control Laws

➤ Elevator command $\Delta\delta_e$

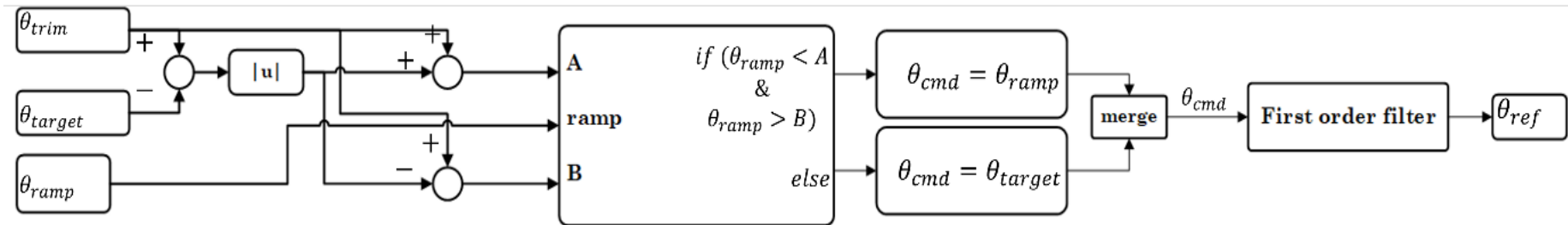
$$\Delta\delta_{e\,cmd} = v_1 + v_{\delta_e} = \underbrace{\left(K_P + \frac{K_I}{s} + K_D s\right) e_{C^*}}_{\text{PID controller}} + \underbrace{K_{NN_{\delta_e}} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}_1)}_{\text{NN1 controller}}$$

➤ The pitch controller gives a C^* command signal C_{cmd}^*

$$C_{cmd}^* = \underbrace{(\mathbf{C}\hat{\mathbf{G}}_k)^{-1}}_{\text{DI}} \left[\dot{r} + \underbrace{\left(K_P + \frac{K_I}{s} + K_D s\right) e_{\theta}}_{\text{PID controller}} - \underbrace{\mathbf{C}\hat{\mathbf{F}}_k \Delta \mathbf{x}}_{\text{DI}} + \underbrace{K_{NN_{\theta}} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}_2)}_{\text{NN2 controller}} \right]$$

3. CONTROLLER DESIGN METHODOLOGY

Pitch reference modeling



- The **pitch reference** θ_{ref} is generated for each trim condition.
- A fixed **ramp** value θ_{ramp} is set, which indirectly controls the pitch rate
- A **pitch target** value θ_{target} is set for each flight condition.
- The **pitch command** θ_{cmd} is smoothen using a **first order filter**, such that :

$$\frac{\theta_{ref}}{\theta_{cmd}} = \frac{\omega_c}{s + \omega_c}$$

where ω_c is the **cut-off frequency** and may be tuned by the designer.

4. MODEL SIMULATION AND VALIDATION RESULTS

Controller Tuning Parameters

Inner loop controller		
Controller element	Parameter	Value
C* reference model	ω_n, ξ	1 rad/s, 0.5
PID controller	K_p, K_i, K_d	-15, -2, -1
NN_1 controller	Learning rate $\Gamma_1, K_{NN\delta_e}$	0.5, 30
	Number of layers	1
	Number of neurons	30
Pitch rate gain	K_q	12.4
Noise filter	K, τ_N	0.125, 0.125 sec
Washout filter	a, b, τ_W	1.6, 6.7, 1.6 sec
Outer loop controller		
Controller element	Parameter	Value
First order filter	ω_c	2 rad/s
PID controller	K_P, K_I, K_D	-1.5, -3, -1.05
NN_2 controller	Learning rate $\Gamma_2, K_{NN\theta}, K_e$	1.2, 1.5, 0.1
	Number of layers	1
	Number of neurons	30
RLS Dynamic Inversion controller	$\kappa, \mathbf{L}_0, \hat{\mathbf{F}}_0, \hat{\mathbf{G}}_0$	1.015, $10^7 \mathbf{I}_{5,5}, \mathbf{I}_{4,4}, -\mathbf{0.1}_{4,1}$

Controller Tuning Parameters

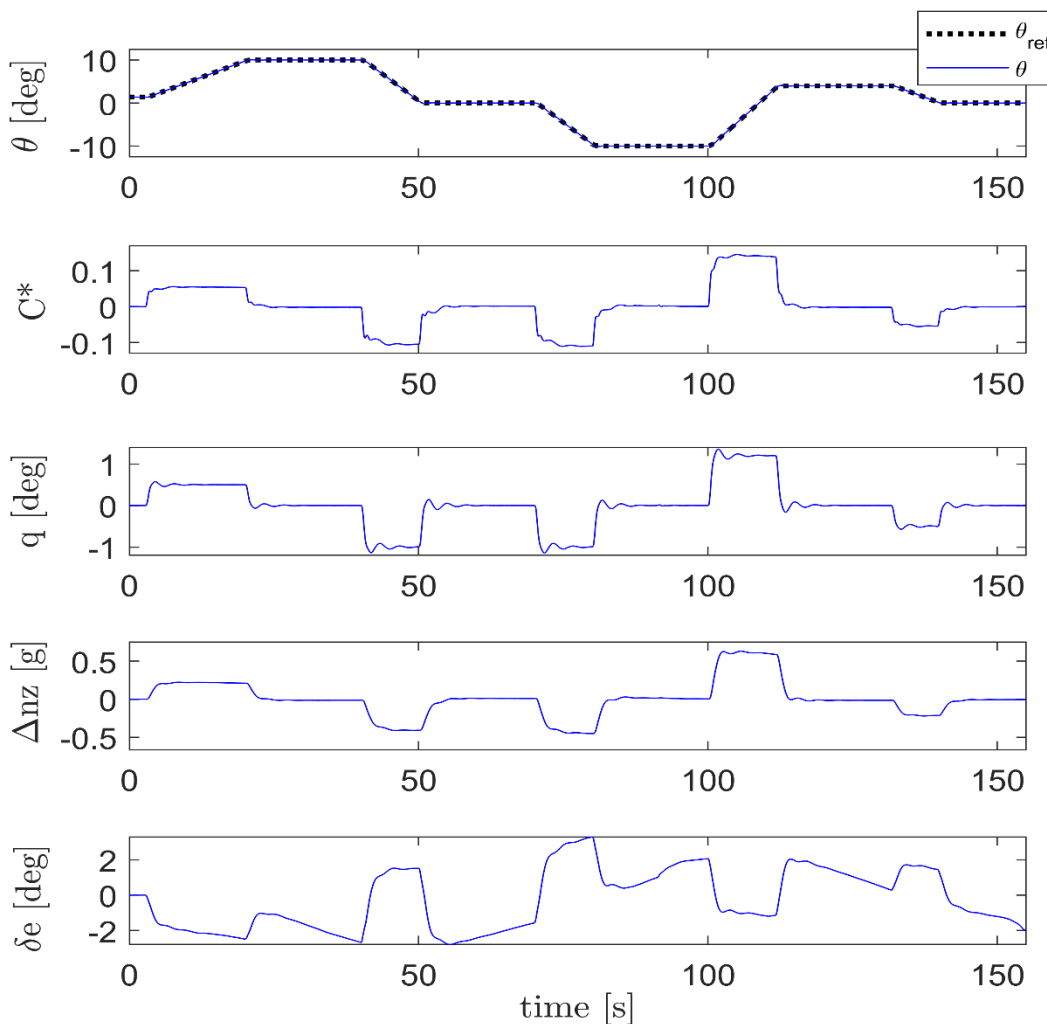
- The tuning of each control element follows **an iterative process**, for which the objective consists of **reducing the pitch angle tracking error e_θ**
- The **inner loop control parameters** were tuned **first**, then the **outer loop control parameters** were tuned.

Online parameter estimation using Recursive Least Square algorithm

➤ **1 flight operating condition is simulated:**

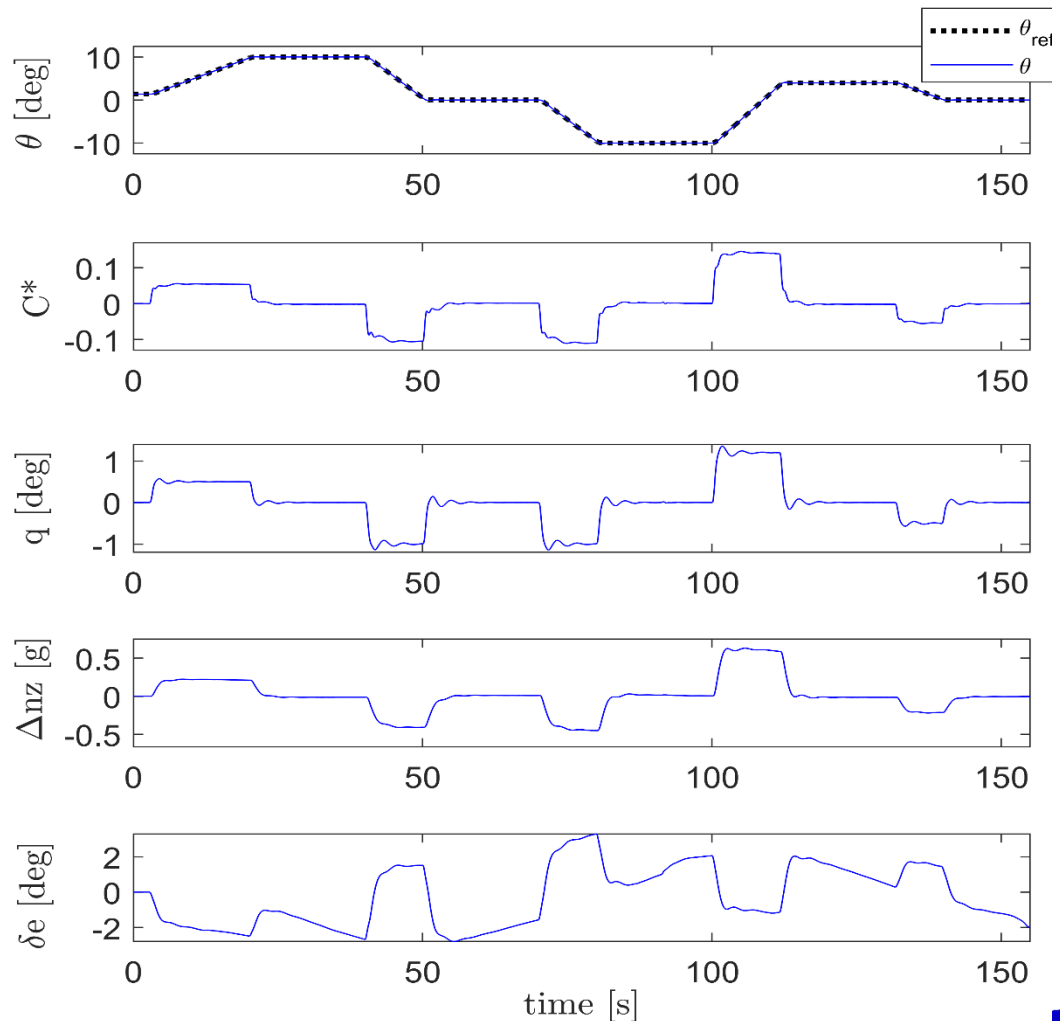
- $CAS = 290$ knots
- $h = 35,000$ ft
- $x_{cg} = 0.25$
- $m = 30,000$ lb

- The figure shows the pitch angle, the C^* parameter, the pitch rate, the load factor and the elevator commands.

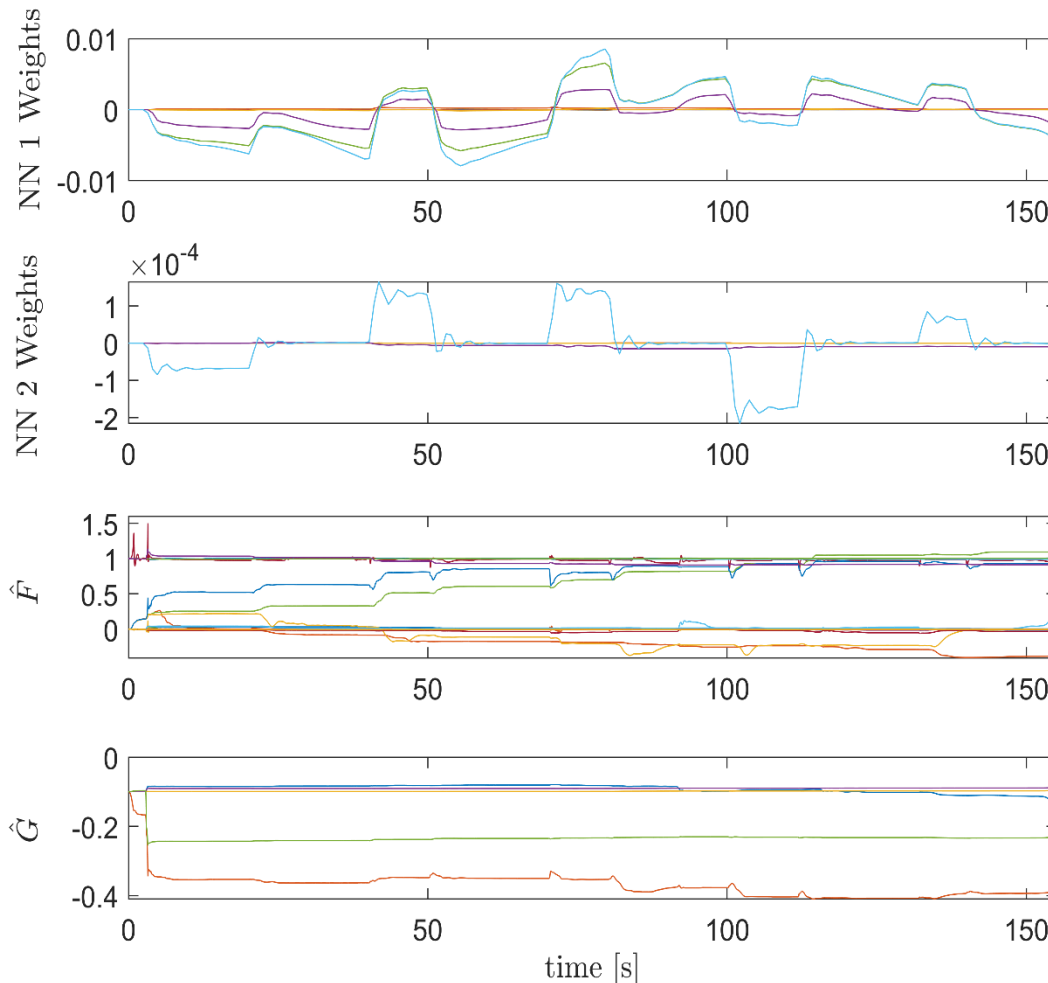


Online parameter estimation using Recursive Least Square algorithm

- The controller is able to follow the pitch angle reference with different **step** and **ramp** values
- The C^* parameter remain stable
- The pitch rate and the normal load factor are well damped, and **also remain stables**
- the elevators δ_e are smooth and **not saturated**, and they remain in a good range of $\pm 2.5^\circ$.

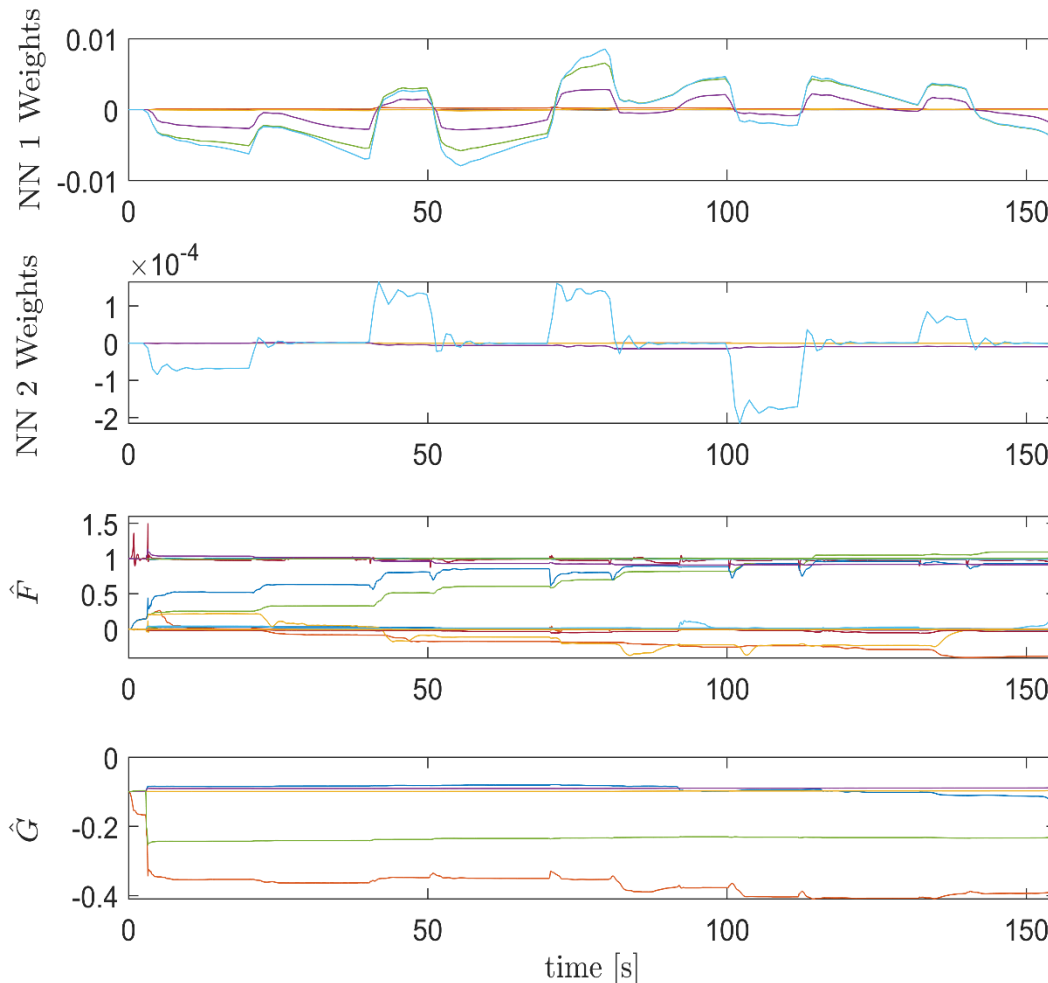


Online parameter estimation using Recursive Least Square algorithm



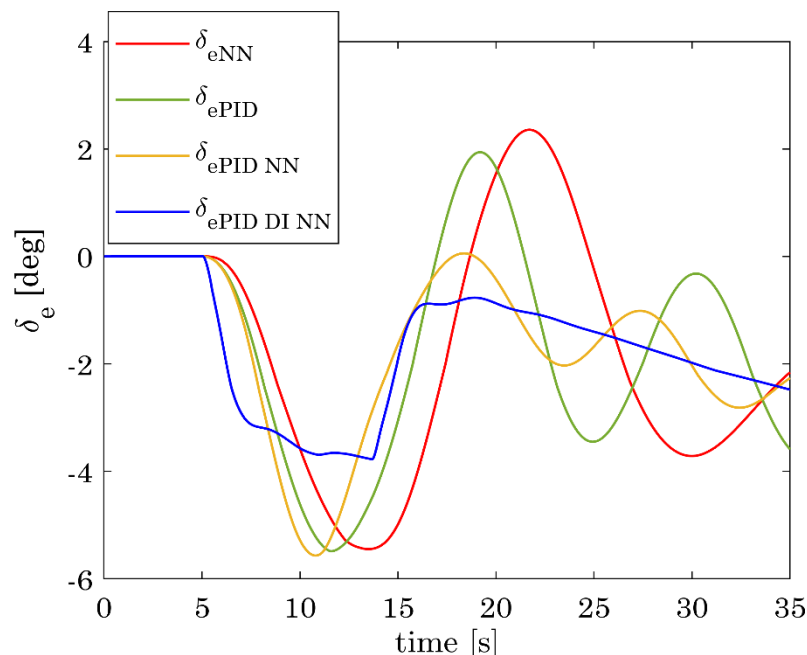
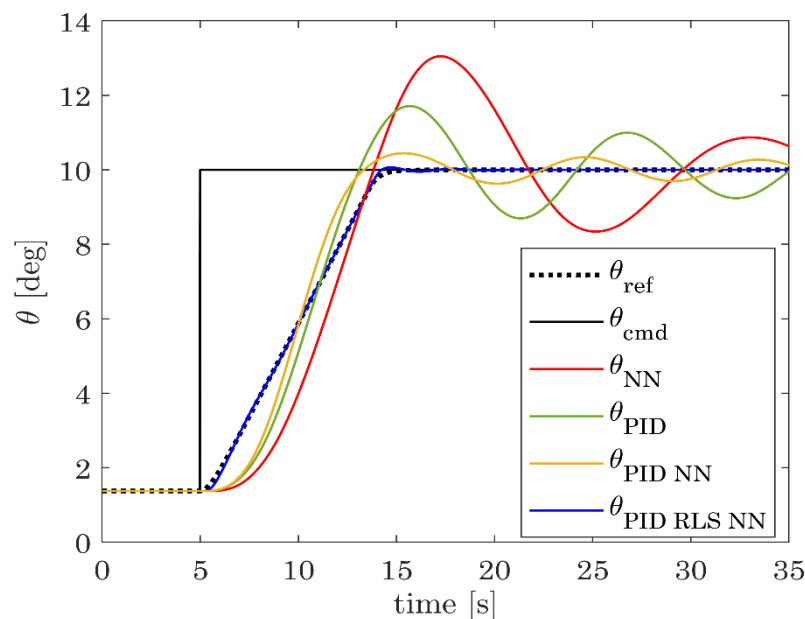
- The **NN weights** are well adapted for the first and second neural network controllers.
- The **NN weights** are **converging** to stable values, and **none** of the weights are diverging.

Online parameter estimation using Recursive Least Square algorithm



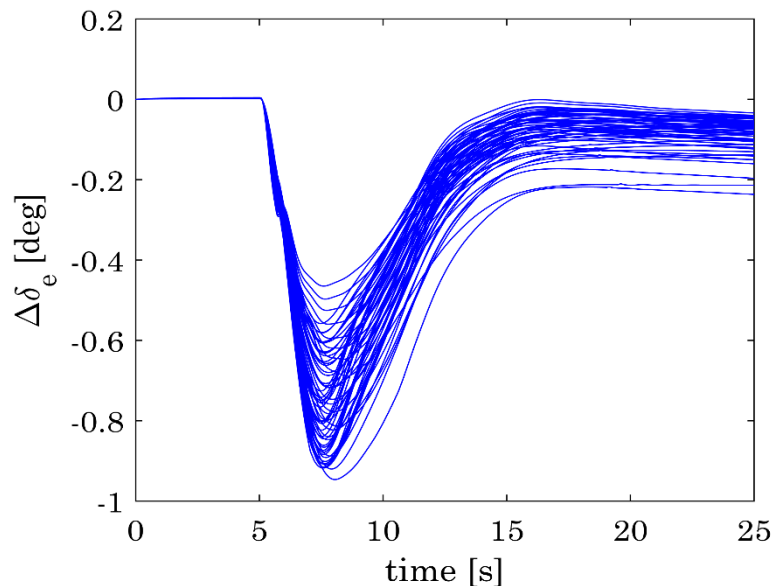
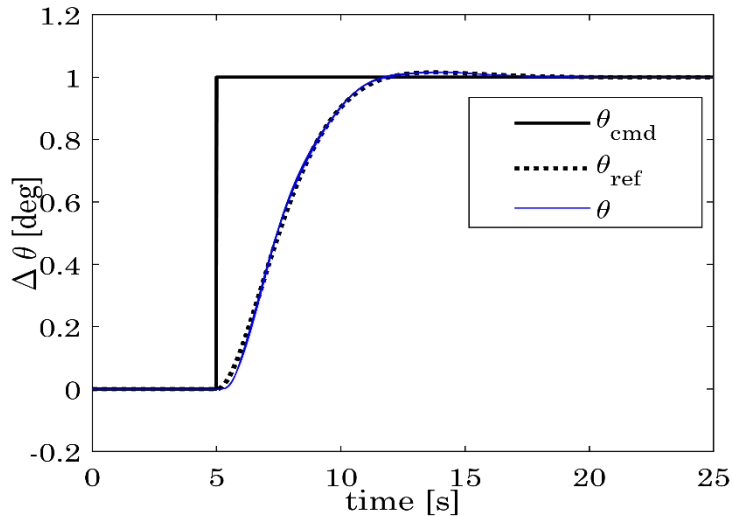
- Matrices $\hat{F}$ and $\hat{G}$ are well estimated.
- The convergence of $\hat{F}$ and $\hat{G}$ is sensitive to the forgetting factor κ , and the initial matrices L_0, Θ_0 .
- The covariance matrix L_0 is reinitialized when the error ϵ is bigger than the threshold value
 $\epsilon > \text{threshold}$

Comparison of different controller configurations



- The **pitch angle** and **elevator deflection** responses are shown.
- The PID – NN configuration alone is **not enough** to achieve the desired performance
- The **PID – RLS – NN** shows the **best performance**

4. MODEL SIMULATION AND VALIDATION RESULTS



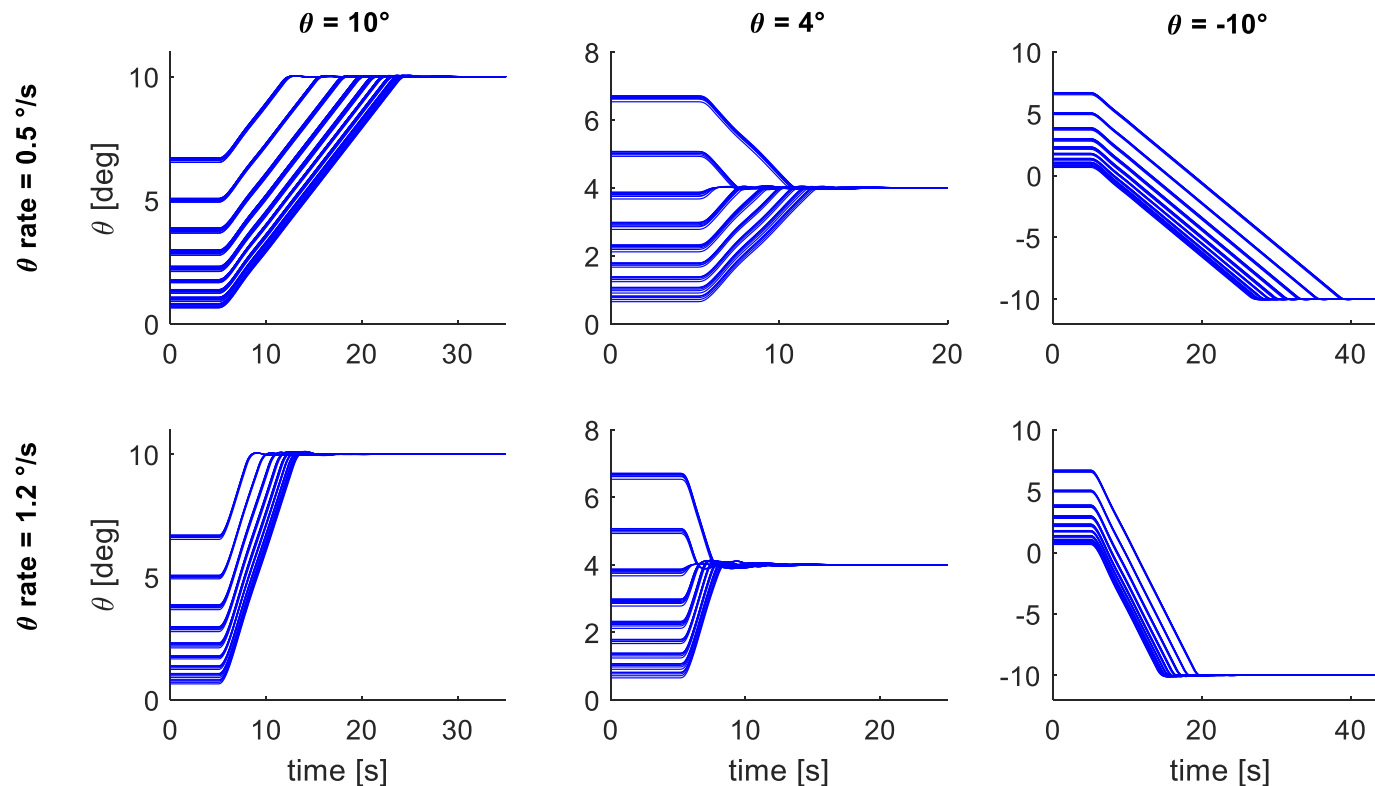
Validation on step signal over the flight envelope

➤ 63 flight operating conditions :

- Cruise phase
- $CAS = 170 - 350$ knots
- $h = 5,000 - 40,000$ ft
- $x_{cg} = 0.25$
- $m = 30,000$ lb

- The controller is perfectly capable of following the pitch reference **step signal** for **all cruise** conditions.

Pitch Controller Validation over the Flight Envelope for ideal conditions

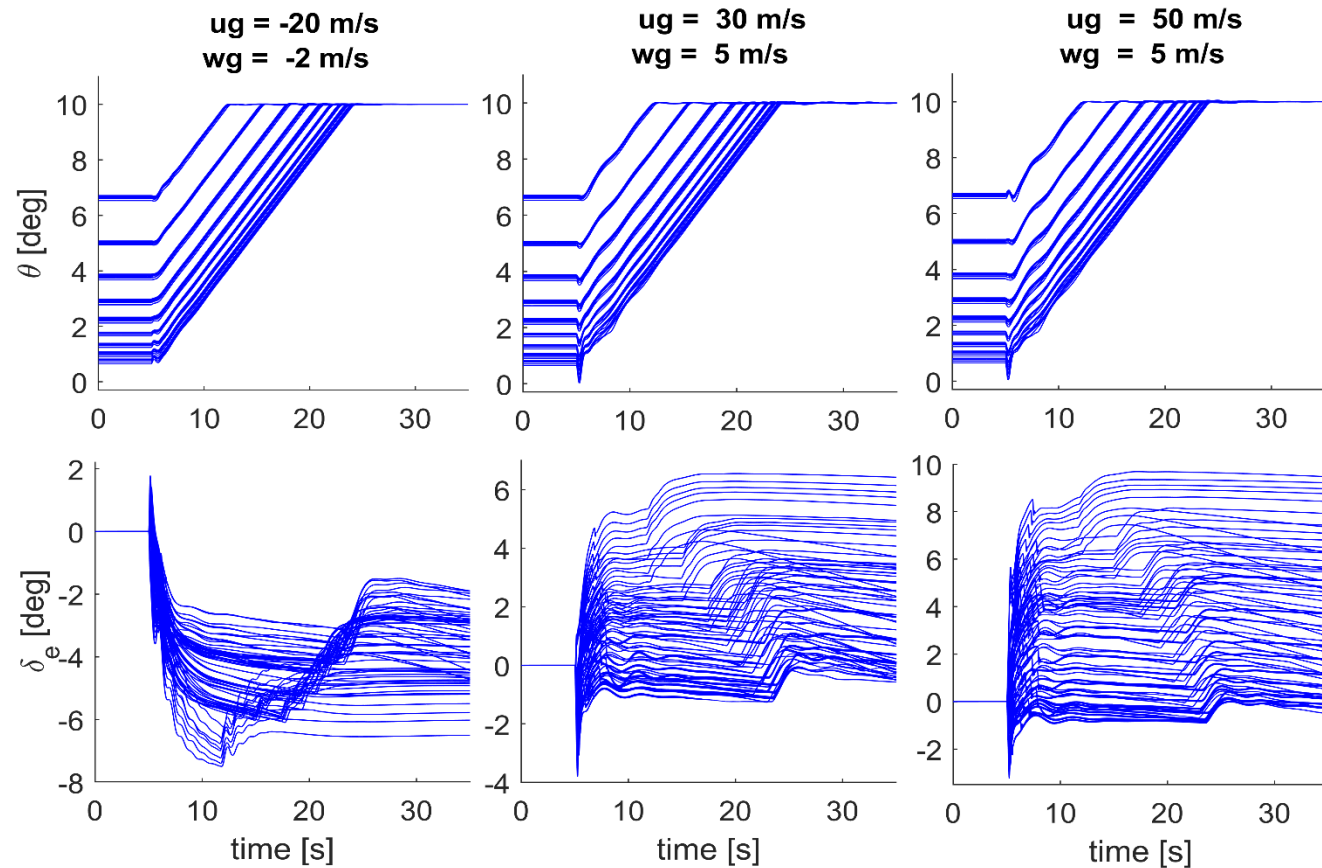


- Two **pitch rates** were applied at $0.5^\circ/\text{s}$ and $1.2^\circ/\text{s}$ for three **pitch target values**
- The controller is able to reach the **target pitch** starting from any given initial pitch trim angle θ_{trim} .

Pitch Controller Validation over the Flight Envelope under uncertainties

Wind Gust tests

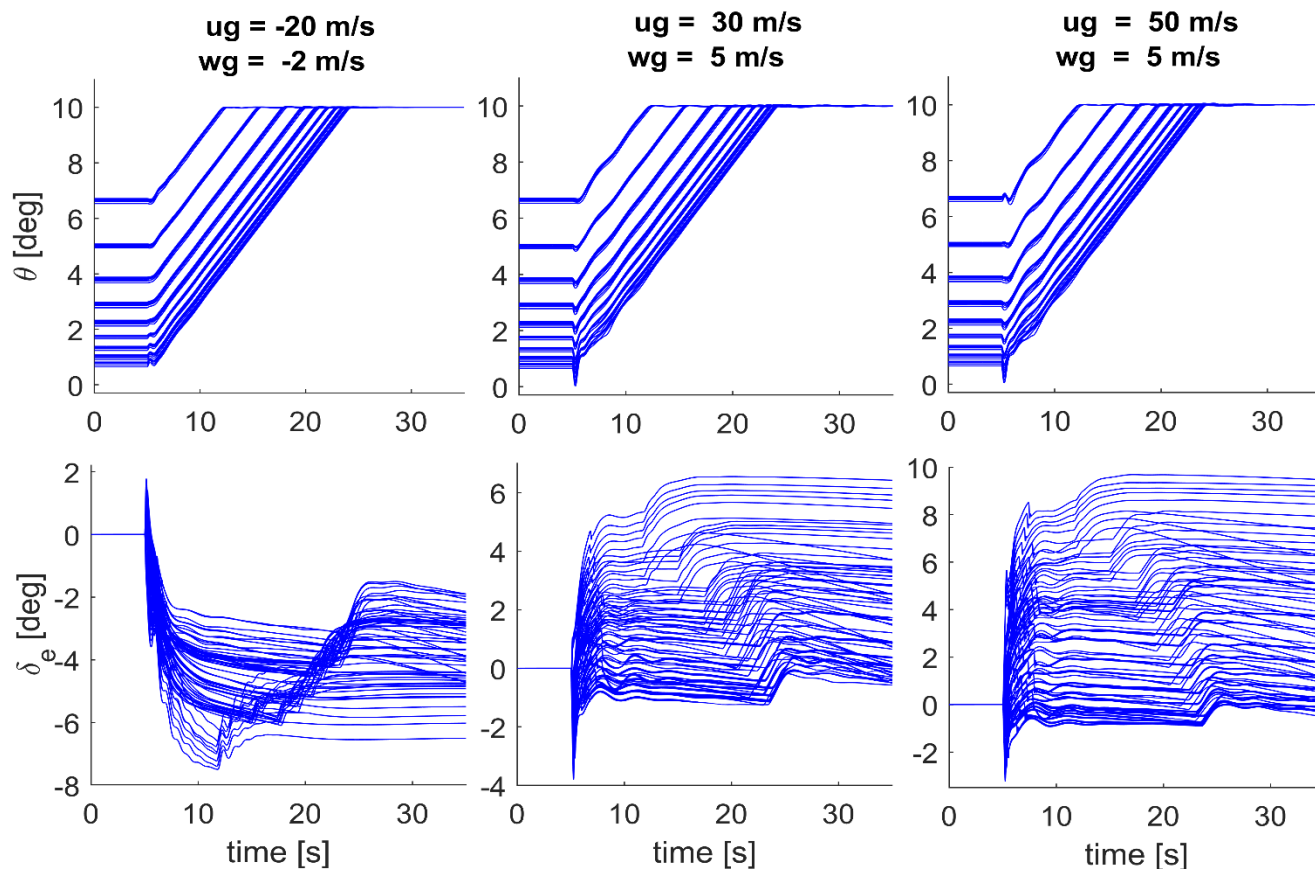
- Only longitudinal u_g and vertical w_g wind gust components are simulated
- The wind gust starts at **5 seconds** for all flight conditions



Pitch Controller Validation over the Flight Envelope under Uncertainties

Wind Gust tests

- The wind gust velocities are **added** directly into the aircraft dynamics
- The controller is able to give an **adaptive command signal** to the elevators in the presence of wind gusts

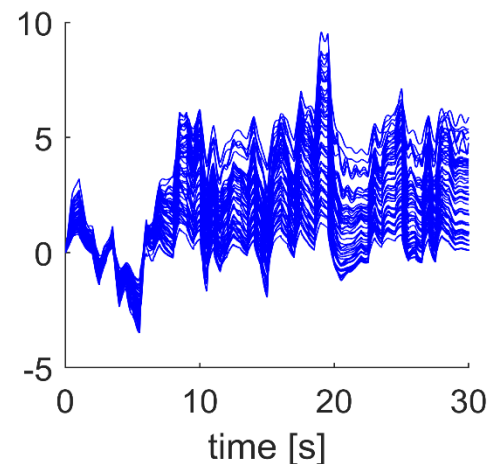
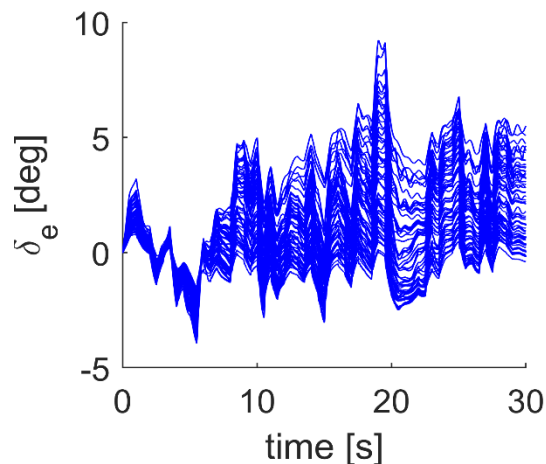
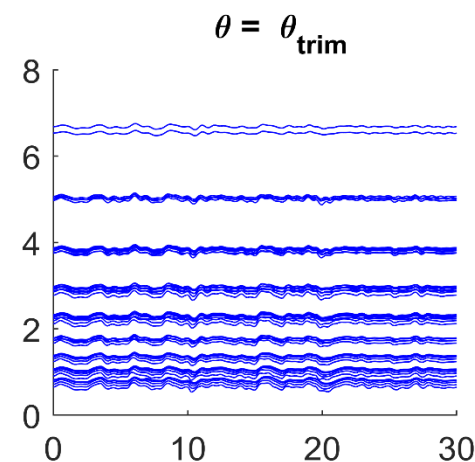
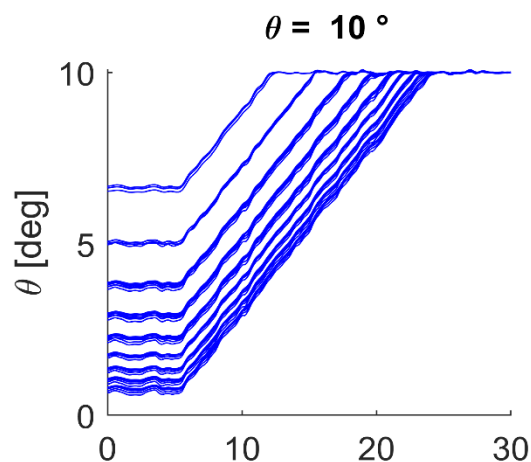


4. MODEL SIMULATION AND VALIDATION RESULTS

Pitch Controller Validation over the Flight Envelope under uncertainties

Turbulence tests

- The **Dryden turbulence** model was simulated.
- Only the **longitudinal** and **vertical** turbulence components were simulated, with **moderate intensities**.
- The controller was able to follow the pitch angle reference in the presence of turbulence,



5. DISCUSSION AND CONCLUSION

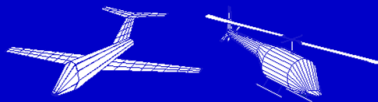
- A **fully adaptive** nonlinear flight controller has been **successfully designed** and validated for **63 flight conditions** in the cruise phase.
- The controller demonstrated excellent capability in tracking a **specified reference** signal for **pitch angle control** for the Cessna Citation X.
- The resulting controller was able to provide very good command signals to the elevators.
- The controller included an **inner loop** architecture which controlled the C^* parameter, allowing **neutral static stability** at **high-speed** conditions
- The controller included 3 main features : a **baseline PID controllers**, **two adaptive neural networks** and a **recursive least square dynamic inversion**
- **no prior knowledge** of the aircraft dynamics was needed. The control input was calculated using **state increments** and **online state approximation** for Dynamic Inversion

5. DISCUSSION AND CONCLUSION

- A **dual adaptation** was fulfilled, first using **online estimation and dynamic inversion**, and second by using **adaptive Neural Networks**.
- The PID gains remain **fixed**, therefore, the PID controller serves only as an **error selector**
- **Additional elements** (noise and washout filters) were added for robustness.
- The adaptive features allow to control the aircraft in the presence of **uncertainties**, such as **wind gusts** and **turbulence**,
- The Recursive Least Square **initial conditions** have a significant impact on the dynamic inversion.
- The **pitch reference model** improved the controller performance, allowing to keep the pitch rate value within the aircraft limitations

5. DISCUSSION AND CONCLUSION

- The controller design is an **iterative approach**, especially during validation. Each control element (PID, NN and Dynamic Inversion) needs to be validated separately and then together to **guaranty convergence and stability**.



Thank you !

Ruxandra Mihaela Botez
ruxandra.botez@etsmtl.ca



LARCASE, ETS,
1100 Notre Dame Ouest
Montréal, Quebec, CANADA
larcase.etsmtl.ca