Development of a Nonlinear Reconfigurable F-16 Model and Flight Control Systems Using Multilayer Adaptive Neural Networks

David Torres Ocaña*, Hyo-Sang Shin*, Antonios Tsourdos*

** Institute of Aerospace Sciences, SATM, Cranfield University, Cranfield, UK, (email: david.torres.ocana@gmail.com [h.shin, a.tsourdos]@cranfield.ac.uk)*

Abstract: For flight control systems, this paper proposes an adaptive control approach based on a framework of Explicit Model Following Direct Adaptive Control scheme. As a first step, a modified F-16 dynamics model is developed to explore control surface redundancies, as well as to enable modelling of dynamics changes result from faults, failures and/or plant deviations. In this modified model, each control surface can be individually controlled. Next, this paper proposes a flight control framework that integrates adaptive neural network, non-linear dynamic inversion, control allocation, and model reference following to leverage their synergies. Then, the proposed approach is tested using the F-16 nonlinear model developed and its performance is validated via numerical simulations.

Keywords: Adaptive Neural Networks; Adaptive Flight Control; Nonlinear F-16 Model; Explicit Model Following; Lyapunov stability

1. INTRODUCTION

Current linear Flight Control Systems (FCS) algorithms are incapable of adapting to sudden changes in terms of aircraft configuration. It is well known that classical control approaches only provide a satisfying performance and robustness if the aircraft is close enough to the model assumed for control design. Any uncertainties or failures lead to degradations in stability and performance. Therefore, linear, model-based control techniques might require complete redesign of control if there are significant changes in aircraft configuration. As a result this tends to restrict the ability to alter the design or carry new equipment or to handle in-flight reconfigurations.

In order to cope with the large spectrum of uncertainties, either in the aircraft model parameters or atmospheric disturbances, classical linear control techniques mainly use robust control methods. These autopilots are designed in a very conservative way, where performance normally is sacrificed to increase closed-loop robustness (Stevens, 2003).

Current on-board processing capabilities of flight control computers allow the implementation of more sophisticated control which could alleviate the aforementioned issues with the classical linear control techniques. This fact leads to nonlinear control techniques in the autopilot design, where the inherent system nonlinearities exhibited by the aircraft are directly taken into account. Advancements in the field of nonlinear control methodology have led to a handful of simultaneous developments of guaranteed characteristics and metrics for nonlinear flight control systems. Employing these novel techniques in future flight control systems will further increase the level of safety and reliability for manned and unmanned aircraft when compared to linear control schemes (Edwards, et al., 2010).

Adaptive control theory offers the potential to design intelligent control structures, which are able to compensate the effects deriving from inevitable uncertainties, potential severe damages or other unanticipated failures that may occur during flight. These new control methods are able to stabilise the aircraft within its remaining physical capabilities in case of severe damage or failure. From previous researches and experiences (M. Gregory, et al., 2009) it has been proven that, even under severe damage, a safe continuation of the flight or mission is possible by taking the modified physical limits of the aircraft into account.

Therefore, this paper develops a control framework which leverages the nonlinear and adaptive control techniques. The proposed framework is based on explicit model following using Non-linear Dynamics Inversion (NDI) and Control Allocation (CA) similarly to (Calise, 2000), or (Ducard, 2009). The adaptation feature presented is performed by implementing adaptive Neural Networks (NN) designed by a Lyapunov stability analysis of the control loop and learning process.

In order to validate the designed approach, this paper firstly develops a reconfigurable 7 degrees-of-freedom (DoF) nonlinear model of the F-16 aircraft based on models presented in (Russell, 2003) and (Morelli, s.f.). This model allows the individual control of each control surface, as well as simulating changes in mass properties, aerodynamic surfaces areas, aerodynamic coefficients and control surfaces blockages, losses, floating and losses of effectiveness. The performance of the proposed control framework is examined via numerical simulations using this reconfigurable nonlinear model.

2. RECONFIGURABLE 7 DoF F-16 MODEL

The model here presented is based in models presented in (Russell, 2003) and (Sonnoeveldt, 2006) where the aircraft is modelled as a rigid body following the laws of Classical Mechanics.

The most important components of the external forces and moments are the aerodynamic components. The aerodynamic model is given in the form of aerodynamic coefficients in body axes and the aerodynamic forces are applied at x_{cg_r}. So, aerodynamic forces and moments are obtained by using an aerodynamic model as the following function, *f*:

$$\begin{aligned}&\{\mathrm{C_{X_T}}, \mathrm{C_{Y_T}}, \mathrm{C_{Z_T}}, \mathrm{C_{l_T}}, \mathrm{C_{m_T}}, \mathrm{C_{n_T}}\}\\&= f(V_{TAS}, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_{LEF})\end{aligned} \quad (1)$$

where V_{TAS} the true airspeed, α and β are body incidence angles, p, q and r angular rates in body axes. δ 's with subscripts represent control surfaces deflections (in degree) as shown in Table 1.

Table 1 Control surfaces deflections

Surface	Symbol	Surface	Symbol
Left elevator	δ_{el}	Right Aileron	δ_{ar}
Right elevator	δ_{er}	Rudder	δ_r
Left Aileron	δ_{al}	Left LEF	δ_{LEF_l}
		Right LEF	δ_{LEF_r}

A more extensive presentation of this model is given in (Russell, 2003). This paper modifies that model to enable individual control of each control surface deflection and changes in reconfiguration parameters.

The reconfiguration parameters are intended to represent blockages, floating, losses of surface, and losses of effectiveness. The parameters, $Blockage\_at$ and $Blockage\_now$, are implemented directly on the actuators dynamics. The usage of Effectiveness of control surfaces would be specified later in the aerodynamic model.

The rest of reconfiguration parameters bear on mass properties, aerodynamic surfaces areas and global aerodynamic properties. These parameters are entered directly to the model:$\Delta\, mass$,$\Delta\, S_L$,$\Delta\, S_R$,$\Delta\, S_{fin}$, x_{cg} , $\Delta\, I_i$, ΔC_D , ΔC_L, , ΔC_m.

Defining $S_{L_{baseline}} = S_{R_{baseline}} = \frac{S}{2}$, $S_{fin_{baseline}} = S_{fin}$ and performing an analysis of the areas that belong to the body and to the surfaces, the current surfaces areas can be approximately determined as:

$$S_{L_{current}} \approx S_{L_{baseline}} \cdot (0.3 + 0.7 \cdot (1 - \Delta\, S_L))$$

$$S_{R_{current}} \approx S_{R_{baseline}} \cdot (0.3 + 0.7 \cdot (1 - \Delta\, S_R))$$

$$S_{fin_{current}} \approx S_{fin_{baseline}} \cdot (0.1 + 0.9 \cdot (1 - \Delta\, S_{fin}))$$

Assuming that these losses of area are from tip to root, it can be defined the current Aerodynamic Centre y-position:

$$y_{AC} = \frac{(1 + 2 \cdot \gamma)}{3 \cdot c_{root} \cdot (1 + \gamma)^2} \cdot (S_{R_{current}} - S_{L_{current}})$$

where c_{root} is the root chord and γ the wing taper ratio. Based on the airplane geometry, the new x-position of the aerodynamic centre is $x_{cg_r} = 0.35 \cdot \bar{c} + 2.11 \cdot \left(\frac{y_{AC}}{b/2}\right)$

In order to separate the contributions of the dynamics and control surfaces deflections in the aerodynamic model with a general expression, the following technique is used:

$$f(V_{TAS}, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_{LEF}) = f_X + f_D$$

$$f_X = f(V_{TAS}, \alpha, \beta, p, q, r, \{\delta_e, \delta_a, \delta_r, \delta_{LEF}\} = 0)$$

$$f_D = f(V_{TAS}, \alpha, \beta, p, q, r, \{\delta_e, \delta_a, \delta_r, \delta_{LEF}\}) - f(V_{TAS}, \alpha, \beta, p, q, r, \{\delta_e, \delta_a, \delta_r, \delta_{LEF}\} = 0)$$

f_X refers to the contribution of the airframe or dynamics $(V_{TAS}, \alpha, \beta, p, q, r)$ to the aerodynamic coefficients and f_D that of the control surfaces deflections $\{\delta_e, \delta_a, \delta_r, \delta_{LEF}\}$. Then, f_D can be weighted with an effectiveness parameter for example.

A similar technique is used to separate the effects of each control surface deflection. It is especially necessary for elevators deflections, since the aerodynamic model is highly nonlinear w.r.t. the elevator deflection. Note that this separation is performed on all the coefficients and derivatives that are function of elevator deflections δ_e: $C_X, C_z, C_Y, C_m, C_n, C_l, \Delta C_m, \Delta C_{m_{ds}}, \Delta C_{n_\beta}, \Delta C_{l_\beta}$. The rest of the control surfaces components in the aerodynamic model, are linear with the parameters as shown in (2):

$$\begin{aligned}\delta'_{a_l} &= \frac{\delta_{a_l}}{20} \cdot \eta_{ail_l} \qquad \delta'_{a_r} = \frac{\delta_{a_r}}{20} \cdot \eta_{ail_r}\\ \delta'_r &= \frac{\delta_r}{30} \cdot \eta_{rudder}\\ \delta'_{LEF_l} &= \left(1 - \frac{\delta_{LEF_l}}{25}\right) \cdot \eta_{LEF_l} \qquad \delta'_{LEF_r} = \left(1 - \frac{\delta_{LEF_r}}{25}\right) \cdot \eta_{LEF_r}\end{aligned} \quad (2)$$

Asymmetric and symmetric effects of ailerons and LEF can be separated due to its linearity by using the variables in Table 2.

Table 2 Symmetric/Asymmetric deflections

Symmetric deflections	Asymmetric deflections
$\delta_a^S = \frac{\delta'_{al} + \delta'_{ar}}{2}$	$\delta_a^A = \frac{\delta'_{al} - \delta'_{ar}}{2}$
$\delta_{LEF}^S = \frac{\delta'_{LEF_l} + \delta'_{LEF_r}}{2}$	$\delta_{LEF}^A = \frac{\delta'_{LEF_l} - \delta'_{LEF_r}}{2}$

In order to compute effect of wing surface loss, which mainly affect to lift, drag and pitching moment coefficients, the following method is used, in a step by step manner:

1st	$C_L = -C_{Z_X} \cdot cos(\alpha) + C_{X_X} \cdot sin(\alpha)$ $C_D = -C_{X_X} \cdot cos(\alpha) - C_{Z_X} \cdot sin(\alpha)$
2nd	$C'_L = C_L \cdot \left(\mathcal{P}_L + (1 - \mathcal{P}_L) \cdot \frac{S_{current}}{S}\right)$ $C'_D = C_D \cdot \left(\mathcal{P}_D + (1 - \mathcal{P}_D) \cdot \frac{S_{current}}{S}\right)$ $Note\, S_{current} = S_{R_{current}} + S_{L_{current}}$

3rd	$C'_{X_X} = -(C'_D + \Delta C_D) \cdot cos(\alpha) + (C'_L + \Delta C_L) \cdot sin(\alpha)$ $C'_{Z_X} = -(C'_D + \Delta C_D) \cdot sin(\alpha) - (C'_L + \Delta C_L) \cdot cos(\alpha)$ $C'_{m_X} = C_{m_X} \cdot \left(\mathcal{P}_m + (1 - \mathcal{P}_m) \cdot \frac{S_{current}}{S}\right) + \Delta C_m$
4th	$C'_{n_X} = C_{n_X} \cdot \left(\mathcal{P}_n + (1 - \mathcal{P}_n) \cdot \frac{S_{fin_{current}}}{S_{fin}}\right)$ $C'_{l_X} = C_{l_X} \cdot \left(\mathcal{P}_l + (1 - \mathcal{P}_l) \cdot \frac{S_{fin_{current}}}{S_{fin}}\right)$

Here, $\mathcal{P}$ parameters are introduced for the effect of the fuselage and rest or airframe. These and rest of vehicle related parameters can be found in the full paper and in (Torres Ocaña, 2014), together with the full Aerodynamics coefficients model. Note the cross effects of asymmetric wing surface loss on the yawing and rolling moments produced by the cross terms $\frac{y_{AC}}{b} \cdot C_{Z_X}{}'$ and $-\frac{y_{AC}}{b} \cdot C_{X_X}{}'$.

3. ADAPTIVE CONTROL BASED ON MULTILAYER ADAPTIVE NEURAL NETWORKS

3.1. Explicit Model Following Controller Framework

The control technique used in this paper is an Explicit Model Following using NDI and CA. This technique is widely used in FCS as in (Ducard, 2009), (Calise, 2000), (McFarland & Calise, 2000), (Chowdhary, et al., 2010) etc.

The Model Reference, that can be found better explained in the full paper, used in this paper is based on (Ducard, 2009) and (Torres Ocaña, 2014). 1st order desired dynamics models $\left(\frac{y_c}{y_{ref}} = \frac{K_{ref_i}}{s+K_{ref_i}}\right)$ are considered with ($K_{ref_\alpha}, K_{ref_q}, K_{ref_p}, K_{ref_\beta}, K_{ref_r}$) as design parameters. Proportional controllers are implemented with $(K_\alpha, K_q, K_p, K_\beta, K_r)$ as gains values and feedforward between those desired responses time derivatives are implemented. Non-linear transformations between reference models are used.

The variables being tracked are (α, β, p, q, r), so the tracking error is given by:

$$e = \begin{pmatrix} e_p \\ e_\alpha \\ e_q \\ e_\beta \\ e_r \end{pmatrix} = \begin{pmatrix} p_c - p_{meas} \\ \alpha_c - \alpha_{meas} \\ q_c - q_{meas} \\ \beta_c - \beta_{meas} \\ r_c - r_{meas} \end{pmatrix} \quad (3)$$

where $(\alpha_{meas}, \beta_{meas}, p_{meas}, q_{meas}, r_{meas})$ denotes assuming the measurements or estimates on (α, β, p, q, r). The actual plant rotational dynamics, which is in reality unknown, can be represented by a set of nonlinear differential equations:

$$\dot{\omega} = \varphi(x, \delta_i) \quad (4)$$

where $\dot{\omega} = (\dot{p} \quad \dot{q} \quad \dot{r})^T$ is the time derivative of the body rates, that is angular accelerations, $x \in \mathbb{R}^n$ is a general representation of the dynamics state, and δ_i's represent all the control inputs.

In order to perform feedback inversion, NDI and CA use a model of the plant to be controlled:

$$\dot{\hat{\omega}} = \hat{\varphi}(x, \delta_i) \quad (5)$$

Due to multiple reasons as stated in Section 1, this model could differ from the actual plant, which results in uncertainty Δ as shown in (6).

$$\Delta = \varphi(x, \delta_i) - \hat{\varphi}(x, \delta_i) \quad (6)$$

This uncertainty causes inversion errors preventing the control system to minimize (3) and destabilise the system in the worst case. In order to deal with this uncertainty, an adaptive term υ_{ad} is subtracted to the inputs commands to NDI, as shown Fig. 1. The commanded angular acceleration to NDI or "pseudo-control" input υ is obtained as:

$$\upsilon = (\dot{p}_{des} \quad \dot{q}_{des} \quad \dot{r}_{des})^T - \upsilon_{ad} \quad (7)$$

To obtain the actual control inputs, the inversion is then carried out, by performing NDI and CA on the model $\hat{\varphi}$ with the "pseudo-control" input υ and the actual dynamics state x, i.e. $\delta_i = \hat{\varphi}^{-1}(x, \upsilon)$.

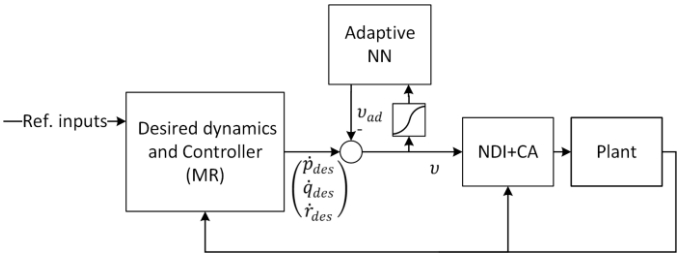

Fig. 1 Adaptive controller scheme

Under assumption that the inner control loops are much faster than the outer loops, from (4), (5), (6) and (7), the time derivative of the tracking error is obtained as:

$$\dot{e} = A \cdot e + B \cdot (\upsilon_{ad} - \Delta) \quad (8)$$

where:

$$A = \begin{pmatrix} -K_p & 0 & 0 & 0 & 0 \\ 0 & -K_\alpha & 1 & 0 & 0 \\ 0 & 0 & -K_q & 0 & 0 \\ 0 & 0 & 0 & -K_\beta & 1 \\ 0 & 0 & 0 & 0 & -K_r \end{pmatrix}, B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

The matrix A becomes Hurwitz if adequate controller's gains are selected.

A positive definite matrix P can be found by using the following Lyapunov equation:

$$A^T \cdot P + P \cdot A = -2 \cdot \Phi \cdot I \quad (10)$$

with $\Phi \geq 1, \Phi \in \mathbb{N}$ which is a designing parameter.

The error used in the NN learning process is defined as:

$$\xi = e^T \cdot P \cdot B \quad (11)$$

3.2. Adaptive Control Design

Fig. 1 shows how the adaptive NN are introduced in the control scheme. The output υ_{ad} is intended to compensate the uncertainty Δ which is an inversion error, as shown in (8).

There are many approaches to obtain v_{ad} as there are many possible designs of adaptive NN, for instance, a linear-in-the-parameters NN in (Kim & Calise, 1997). In this paper, Lyapunov stability analysis is used to obtain the adaptation laws for any number of layers and neurons in NN.

There are different notations when NN are implemented. Table 3 represents a general algorithm to propagate the output v_{ad} from the input $\bar{x}$ depending on the notations used.

Table 3 Forward propagation

Forward propagation	
L+1 Layers notation	**L Layers notation**
$a_1 = \bar{x}$ $z_1 = \Theta_1 \cdot a_1$ ***for*** $i = 2{:}L$ $a_i = \sigma_{i-1}(z_{i-1})$ $z_i = \Theta_i \cdot a_i$ ***end*** $v_{ad} = z_L$	$a_1 = \bar{x}$ ***for*** $i = 2{:}L$ $z_i = \Theta_{i-1} \cdot a_{i-1}$ $a_i = \sigma_i(z_i)$ ***end*** $v_{ad} = a_L$

Here, "L+1 layers" or "L layers" mean the total number of layers of the NN including input and output layers. For the given regression problem, the output layer of the NN should not implement any activation function, so "L+1 layers" notation is preferred and used in this paper.

The weight matrices Θ_i's are the main parameters of the NN and preferably need to be determined or adapted in real time. Activation functions suitable for the given problem are sigmoidal function $\sigma(z) = \frac{1}{1+e^{-a \cdot z}}$. Equation (12) shows the inputs used in this paper where h and M denote the altitude and the Mach number:

$$\bar{x}^T = (1, h, M, M^2, \alpha, \alpha^2, \beta, p, r, Roll_{Var}^T, Long_{Var}^T, Yaw_{Var}^T, v$$

$$Roll_{Var} = \begin{pmatrix} e_p \\ p_c \\ \dot{p}_c \end{pmatrix}, Long_{Var} = \begin{pmatrix} e_\alpha \\ e_q \\ \alpha_c \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix}, Yaw_{Var} = \begin{pmatrix} e_\beta \\ e_r \\ \beta_c \\ \dot{\beta}_c \\ \dot{r}_c \end{pmatrix} \quad (12)$$

Note the complexity of the NN, that is the number of layers and neurons, depends on the input variables used and how well they represent the dynamics.

Defining the output of a general NN as $v_{ad} = f_{NN}(\Theta_i, \bar{x})$ and assuming that the normalized input is $\bar{x} \in \mathcal{D}$, where $\mathcal{D}$ is a bounded domain, there exists a set of ideal weights Θ_i^* such that:

$$v_{ad}^*(\Theta_i^*, \bar{x}) - \Delta = f_{NN}(\Theta_i^*, \bar{x}) - \Delta = \epsilon \quad (13)$$

In (13), $\|\epsilon\| \leq \bar{\epsilon}$ and $\bar{\epsilon} \geq 0$. These ideal weight matrices can be seen as the solution of the problem in (14) which has not solution in this case:

$$\{\Theta_i^*\} = \underset{\{\Theta_1, \ldots \Theta_L\}}{\operatorname{argmin}} \|\epsilon\|^2 = \underset{\{\Theta_1, \ldots \Theta_L\}}{\operatorname{argmin}} \|f_{NN}(\Theta_i, \bar{x}) - \Delta\|^2 \quad (14)$$

The error in the weight matrices $\widetilde{\Theta}_i$, that defines the convergence of the learning process, is defined:

$$\widetilde{\Theta}_i = \Theta_i - \Theta_i^* \quad (15)$$

With this framework, the Lyapunov candidate function of the controller presented in Fig. 1 including the learning process is given by:

$$V = \frac{1}{2} \cdot e^T \cdot P \cdot e + \frac{1}{2} \cdot \sum_{i=1}^{L} tr\{\widetilde{\Theta}_i \cdot \Gamma_i^{-1} \cdot \widetilde{\Theta}_i^T\} \quad (16)$$

where $\Gamma_i > 0$ are the learning rates, which are design parameters of the NN in the adaptation laws.

It is assumed the Frobenius norm of Z, defined in (17), is bounded, i.e. $[\![Z]\!] \leq \bar{Z}$, where $[\![.]\!]$ denotes Frobenius norm.

$$Z = \begin{pmatrix} \Theta_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Theta_L \end{pmatrix} \quad (17)$$

The error of Z is defined $\tilde{Z} = Z - Z^*$ and the derivative of the activation function is $\sigma_i'(z) = \frac{\partial\, \sigma_i(z)}{\partial\, z}$:

Consequently, the online adaptation laws defining the weight matrices are obtained as:

Table 4 Adaptation laws for the general NN "L+1 layers" notation

$\delta_L = \psi_L = \xi$
for $i = L - 1{:}1$
$\delta_i = \Theta_{i+1}^T \cdot \delta_{i+1} \cdot \sigma_i'(z_i)$
$\psi_i = \sigma_i'(z_i) \cdot (\psi_{i+1})^T$
end
$\dot{\Theta}_L = -\Gamma_L \cdot \left[\xi \cdot a_L^T - \sum_{i=1}^{L-1} (\psi_i \cdot z_i)^T + \lambda \cdot \|\xi\| \cdot \Theta_L\right]$
for $i = 1{:}L - 1$
$\dot{\Theta}_i = -\Gamma_i \cdot \left[\delta_i \cdot a_i^T + \lambda \cdot \|\xi\| \cdot \Theta_i\right]$
end

Table 5 Adaptation laws for the general NN "L layers" notation

$\delta_L = \psi_L = \xi$
for $i = L - 1{:}2$
$\delta_i = \Theta_i^T \cdot \delta_{i+1} \cdot \sigma_i'(z_i)$
$\psi_i = \sigma_i'(z_i) \cdot (\psi_{i+1})^T$
end
$\dot{\Theta}_{L-1} = -\Gamma_{L-1} \cdot \left[\xi \cdot a_{L-1}^T - \sum_{i=2}^{L-1} (\psi_i \cdot z_{i+1})^T + \lambda \cdot \|\xi\| \cdot \Theta_{L-1}\right]$
for $i = 1{:}L - 2$
$\dot{\Theta}_i = -\Gamma_i \cdot \left[\delta_{i+1} \cdot a_i^T + \lambda \cdot \|\xi\| \cdot \Theta_i\right]$
end

Note that, in the "L+1 layers" notation, $\sigma_i(z)$ is the activation function of the $(i+1)^{th}$ layer. Weights matrices in Table 3 are obtained then $\Theta_i = \Theta_{i_0} + \int_0^t \dot{\Theta}_i \cdot dt$ where Θ_{i_0} can be randomly initialized or the last known stable matrix.

Note as well, that the products $\psi_i \cdot z_{i+1}$ and $\delta_{i+1} \cdot \sigma_i'(z_i)$ are colum-wise products and the regularization parameter $\lambda > 0$ which is another design parameter.

3.2.1. Stability analysis

Now, let us analyse the stability of the designed adaptation laws.

From (8) and (13), we have:

$$\dot{e} = A \cdot e + B \cdot (f_{NN}(\Theta_i, \bar{x}) - f_{NN}(\Theta_i^*, \bar{x}) + \epsilon) \quad (18)$$

Using (18) with "L+1 layers notation", the tracking error dynamics is obtained as:

$$\dot{e} = A \cdot e + B \cdot \left[\Theta_{\mathrm{L}} \cdot a_L(\Theta_{\mathrm{i}}, \bar{x}) - (\Theta_{\mathrm{L}} - \widetilde{\Theta}_L) \cdot a_L(\Theta_{\mathrm{i}}^*, \bar{x}) + \epsilon\right] \quad (19)$$

The ideal output can be represented as a Taylor expansion:

$$a_L(\Theta_i^*, \bar{x}) = a_L(\Theta_i, \bar{x}) + \sum_{i=1}^{L-1} \left.\frac{\partial\, a_L(\Theta_i, \bar{x})}{\partial\, \Theta_i}\right|_{\substack{\Theta_i=\Theta_i \\ \bar{x}=\bar{x}}} \cdot (\Theta_i^* - \Theta_i) + \mathcal{O}\left(\left[\!\left[\tilde{Z}\right]\!\right]\right)^2 \quad (20)$$

Substituting (20) into (19) the time derivative of the tracking error is simplified:

$$\dot{e} = A \cdot e + B \cdot \left[\widetilde{\Theta}_L \cdot \left(a_L(\Theta_i, \bar{x}) - \sum_{i=1}^{L-1} \left.\frac{\partial\, a_L(\Theta_i, \bar{x})}{\partial\, \Theta_i}\right|_{\substack{\Theta_i=\Theta_i \\ \bar{x}=\bar{x}}} \cdot \Theta_i\right) - \Theta_L \cdot \sum_{i=1}^{L-1} \left.\frac{\partial\, a_L(\Theta_i, \bar{x})}{\partial\, \Theta_i}\right|_{\substack{\Theta_i=\Theta_i \\ \bar{x}=\bar{x}}} \cdot \tilde{\Theta}_i + \epsilon + \omega\right] \quad (21)$$

with $\omega = \widetilde{\Theta}_L \cdot \sum_{i=1}^{L-1} \left.\frac{\partial\, a_L(\Theta_{\mathrm{i}}, \bar{x})}{\partial\, \Theta_{\mathrm{i}}}\right|_{\substack{\Theta_{\mathrm{i}}=\Theta_{\mathrm{i}} \\ \bar{x}=\bar{x}}} \cdot \Theta_{\mathrm{i}}^* + \mathcal{O}\left(\left[\!\left[\tilde{Z}\right]\!\right]\right)^2$.

From the adaptation laws in (16) and (21), the time derivative of the Lyapunov function is given by:

$$\dot{V} = -2 \cdot \Phi \cdot \|e\|^2 + \xi \cdot (\epsilon + \omega) - \lambda \cdot \|\xi\| \cdot tr\{\tilde{Z}^T \cdot Z\} \quad (22)$$

In (22), $\|\omega\| \leq \mathrm{c}_0 + c_1 \cdot \left[\!\left[\tilde{Z}\right]\!\right] + c_2 \cdot \left[\!\left[\tilde{Z}\right]\!\right]^2$ where c_i 's are dependent of the number of layers and that of neurons.

The correspondent inequalities and the definition of trace $2 \cdot tr\{A \cdot \mathrm{B}\} = [\![A + B]\!]^2 - [\![B]\!]^2 - [\![A]\!]^2 \geq -[\![B]\!]^2 - [\![A]\!]^2$ on (22) yield:

$$\dot{V} \leq -2 \cdot \Phi \cdot \|e\|^2 + \|\xi\| \cdot \left(\bar{\epsilon} + c_0 + c_1 \cdot \left[\!\left[\tilde{Z}\right]\!\right] + c_2 \cdot \left[\!\left[\tilde{Z}\right]\!\right]^2\right) + \frac{1}{2} \cdot \lambda \cdot \|\xi\| \cdot \left(\left[\!\left[\tilde{Z}\right]\!\right]^2 + \bar{Z}^2\right) \quad (23)$$

So, either of the expressions in (24) ensures $\dot{V}$ to be negative semidefinite $\forall \bar{x} \in \mathcal{D}$:

$$\Phi \cdot \|e\| \geq \frac{\|PB\|}{2} \cdot \left[(\bar{\epsilon} + c_0) + c_1 \cdot \left[\!\left[\tilde{Z}\right]\!\right] + \left(c_2 + \frac{\lambda}{2}\right) \cdot \left[\!\left[\tilde{Z}\right]\!\right]^2 + \frac{\lambda}{2} \cdot \bar{Z}^2\right]$$

$$\left[\!\left[\tilde{Z}\right]\!\right] \geq \frac{c_1 + \sqrt{c_1^2 - 4 \cdot \left(c_2 + \frac{\lambda}{2}\right) \cdot \left((\bar{\epsilon} + c_0) + \frac{\lambda}{2} \cdot \bar{Z}^2\right)}}{2 \cdot \left(c_2 + \frac{\lambda}{2}\right)} \quad (24)$$

Note that Γ_i, Φ and λ are the design parameters and determine a trade-off between performance and robustness.

4. NUMERICAL EXAMPLES

In order to investigate the performance of the control scheme developed, two different tests are performed. Flight altitude in both cases is 4000m and a True Air Speed (TAS) 200m/s. The TAS is controlled by a PI controller, so that it can be assumed as constant in all manoeuvres.

The NN implemented in 4.1 and 4.2 is a three layers design, with a Learning rate of 20, regularization parameter of 1 and a hidden layer with 50 neurons.

4.1. Deviations accommodation

The case presented simulates the incorporation of two external fuel tanks, under the assumption that neither the fuel tanks nor their incorporation was taken into account at the modelling stage.

New fuel tanks mass is 500kg and each of them is located at 3.5m of the symmetry plane of the aircraft, which causes changes in mass, moments of inertia, drag, pitching moments and CG location. Estimations of these changes are presented in Table 6.

Table 6 New fuel tanks estimated changes

ΔJ_x(%)	ΔJ_z (%)	Δm (%)	ΔC_D	ΔC_{M_0}	x_{cg}
27.2%	14.3%	10.8%	0.02	−0.03	0.35

Fig. 2 shows the responses of the aircraft in terms of normal acceleration, angle-of-attack (AoA), roll rate and SS angle.

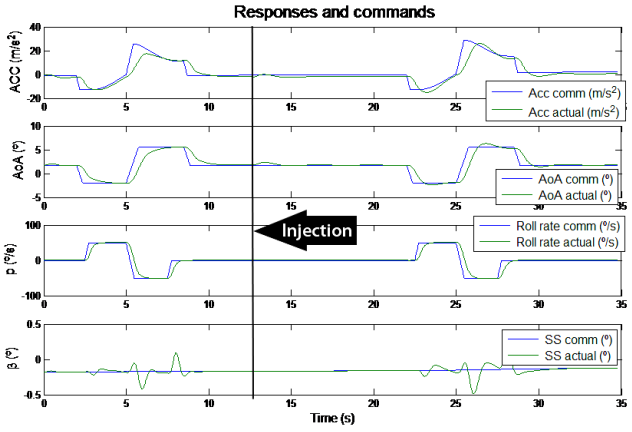

Fig. 2 Responses under mass properties changes

Fig. 3 represents the behaviour of the control system including the proposed adaptation law from the outputs of the NN (υ_{ad}) and the total inputs (Angular accelerations) of NDI (υ).

As shown in Figures, an injection is performed at 12.5s. This causes a pitching moment (long channel signal), but the NN immediately adapt to the change and consequently the proposed control system corrects this change.

The second set of manoeuvres is the same as the first set, where the responses of the aircraft are similar, showing a similar performance in terms of tracking. NN outputs in the second set of manoeuvres are more aggressive and bigger than those in the first set, which improves the tracking performance and helps to deal with the mass properties changes.

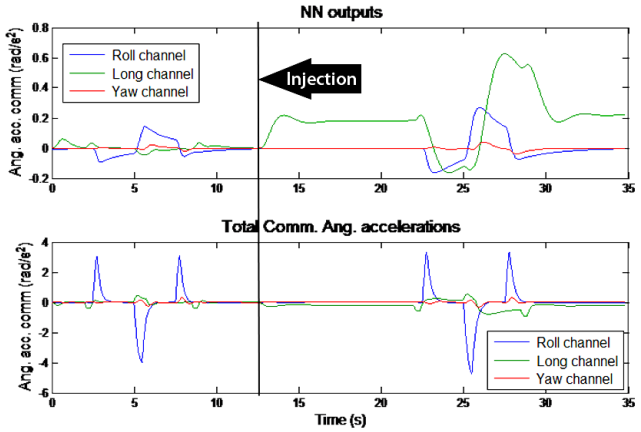

Fig. 3 Commands adaptation under mass properties changes

4.2. Fault receiving

This session considers a typical fault situation, the loss or floating of a control surface.

The commanded manoeuvres presented here are several roll manoeuvres in both directions, at a constant altitude and speed. Fault is injected at 5s while performing a roll manoeuvre.

Fig. 4 shows responses in roll rate, NN outputs (υ_{ad}) and the total inputs (Angular accelerations) of NDI (υ).

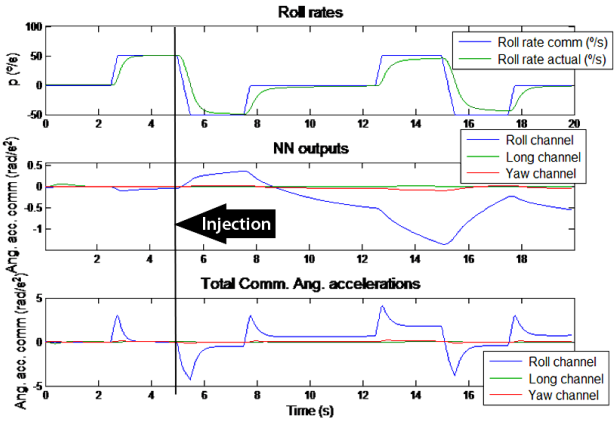

Fig. 4 Adaptation under loss of an aileron

Fig. 4 illustrates how the performance in roll rate after injection is degraded but similar to the undamaged plant, with similar rise time but a degraded settling time.

It can be seen how NN outputs increase when tracking errors appears. This increase implies that the action the NN is to eliminate these steady state errors in tracking. This action is similar to an integral action, so it can be understood as an "adaptive integral action".

5. CONCLUSION

This paper firstly developed a nonlinear 7dof of control reconfigurable model which is used to develop an Explicit Model Following Direct Adaptive controller with Adaptive Neural Networks and using NDI and CA. It also designed a Multilayer Adaptive Neural Network and undertook its stability analysis.

The controller presented and the three layers Adaptive NN implemented show notable performance under deviations and faults in the plant dynamics.

The main advantage of the proposed approach is incorporation of changes or deviations in the plant on/off-flight while ensuring stability, manoeuvrability, and performance. This could reduce development efforts in modelling, control design and tuning stages.

Moreover, under the event of faults or major changes in the plant dynamics, this paper shows that the proposed approach can guarantee stability during the recovery and relax the need to develop extensive and precise SI and/or FDI algorithms.

REFERENCES

Calise, A., 2000. *Development of a reconfigurable flight control law for the X-36 tailless fighter aircraft.* s.l., s.n.

Chowdhary, G. et al., 2010. Autonomous guidance and control of Airplanes under Actuator Failures and Severe structural Damage. *American Institute of Aeronautics and Astronautics.*

Ducard, G. J., 2009. *Fault-tolerant Flight Control and Guidance Systems.* s.l.:Springer.

Edwards, C., Lombaerts, T. & Smaili, H., 2010. *Fault Tolerant Flight Control - A Benchmark Challenge.* s.l.:Springer.

Kim, B. & Calise, A., 1997. Nonlinear Flight Control using Neural Networks.

M. Gregory, I. et al., 2009. *L1 Adaptive Control Design for NASA AirSTAR Flight Test Vehicle.* Chicago, AIAA Guidance, Navigation and Control Conference.

McFarland, M. B. & Calise, A. J., 2000. Adaptive Nonlinear Control of Agile Antiair Missiles Using Neural Networks. *IEEE Transaction on Control Systems Technology* , 8(5), p. 749.

Morelli, E. A., n.d. *Global nonlinear parametric modeling with application to F-16 aerodynamics,* Hampton, Virginia: NASA Langley Research Center.

Russell, R. S., 2003. Non-linear F-16 Simulation using Simulink and Matlab. *University if Minnesota.*

Sonnoeveldt, L., 2006. Nonlinear F-16 Model Description. *Control and Simulation Division at Faculty of Aerospace Engineering. Delft University.*

Stevens, B. L. F., 2003. *Aircraft Control and Simulation.* 2nd ed. s.l.:John Wiley & Sons.

Torres Ocaña, D., 2014. *Adaptive and Fault Tolerant Flight Control Systems.* Cranfield University: MSc Thesis.