Simulator Based Evaluation for Helicopter Load Factor Limit Avoidance with Concurrent Learning

Zeynep Unal, Gonenc Gursoy and Ilkay Yavrucuk

Abstract A neural network based adaptive limit detection algorithm concurrent learning is used in a simulator environment with an active side stick inceptor to test limit avoidance algorithms. The simulator environment uses a generic utility helicopter model, an active side stick controller and a generic engineering development cockpit. The active interceptor is used to give tactile cues to the pilot based on control margin predictions. In this paper the load factor limit for a fly-by-wire helicopter is studied. The adaptive limit detection algorithm uses Linearly Parameterized and Single Hidden Layer Neural Networks to estimate allowable control travels for the longitudinal cyclic input.

1 Introduction

Use of active interceptors for active pilot cueing to avoid flight envelope limits is studied in the literature [1], [2]. Active pilot cueing boosts confidence of pilots, this improves the effective operational flight envelope. Such active interceptor need to be used together with flight envelope protection systems. The aim is to estimate the allowable control travels within envelope limits and to give tactile cues to the pilot as the aircraft approaches its envelope boundaries. In order to give timely cues, correct estimations of control margins are required [3–5]. A sensitivity based method for control limit estimation is explored in [6, 7], where estimation is done by utilizing adaptive elements with concurrent learning [8]. However, control limit estimation alone is not enough for flight envelope protection. These algorithms need to be in-

Zeynep Unal
Middle East Technical University, Ankara,Turkey, e-mail: zeunal@metu.edu.tr

Gonenc Gursoy
Aerotim Engineering L. L. C., Ankara,Turkey, e-mail: gursoy@aerotim.com.tr

Ilkay Yavrucuk
Middle East Technical University, Ankara,Turkey, e-mail: yavrucuk@metu.edu.tr

tegrated with active pilot cueing systems for effective flight envelope protection.
This study uses the adaptive limit prediction method stated in [6] and [9] for load factor limit avoidance. Here, the critical limit parameter is the load factor of the helicopter and control margins for longitudinal channel is estimated online. First, the limit parameter is estimated with an approximate inverse model and the adaptive element at delayed time step. Then the control sensitivity is calculated for control margin estimation. The adaptive element is modeled with Linear in the Parameter Neural Network (LPNN) previously. This study also models the adaptive element with Single Hidden Layer Neural Network(SHL NN) for improved learning. Concurrent learning is used in weight update law for both neural networks.
For simulations, a high fidelity helicopter model equipped with automatic flight control system is used. Various pitch up and pitch down maneuvers are considered for test scenarios. Furthermore, the pilot is given force feedback cues through an active side stick [10]. The active side stick, the helicopter model and flight envelope protection algorithms are all integrated in a simulation environment that runs in real time.
This paper is organized as follows. In Section.2 control margin estimation method based on sensitivity calculations is explained. Additionaly, in this methodology section concurrent learning scheme for LPNN and SHL is briefly discussed. Then, in Section.3 implementation of this method to load factor limit avoidance problem is given. Simulation results are presented in Section.4. Finally, conclusions are stated in Section.5.

2 Methodology

Methodology given in [6] and [9] is followed in this paper, the reader may refer to [6] and [9] for detailed explanations on the methodology. The dynamics of the measured limit parameter $y_p \in \Re$ can be written as stated in [9] as:

$$y_p = h(\mathbf{x}_f, \mathbf{x}_s). \tag{1}$$

where, h is a nonlinear vector function, $x_f \in \Re^l$ are the fast states and $x_s \in \Re^{n-l}$ are the slow states of the plant.

Following the assumption that the nonlinear vector function h is invertable in [9] i.e. $\mathbf{x_f} = h^{-1}(y_p, \mathbf{x_s})$, y_p can be expressed as the summation of $\hat{g}_n$, approximation of model inverse and ξ, modeling error:

$$y_p = \hat{g}_n^{-1}(\mathbf{x}_s, \dot{\mathbf{y}}_p, u_e) + \xi. \tag{2}$$

Here, $\dot{\mathbf{y}}_p = [y_p^{(1)}, y_p^{(2)}, ..., y_p^{(n)}]^T \in \Re^n$ is the vector of derivatives and $u_e \in \Re$ is the control input.

For estimating y_p, an adaptive element, Δ, is introduced:

$$\hat{y}_p = \hat{g}_n^{-1}(\mathbf{x}_s, \dot{\mathbf{y}}_p, u_e) + \Delta(\mathbf{x}_s, \dot{\mathbf{y}}_p, u_e). \tag{3}$$

In order to estimate the derivative terms using past data with central difference operation as in [6, 9], Eq.(3) can be written at delayed time, d as:

$$\hat{y}_{p_d} = \hat{g}_n^{-1}(\mathbf{x}_{s_d}, \dot{\mathbf{y}}_{p_d}, u_{e_d}) + \Delta(\mathbf{x}_{s_d}, \dot{\mathbf{y}}_{p_d}, u_{e_d}) \tag{4}$$

Approximation error, e_d, is obtained by subtracting Eq.(4) from Eq.(2):

$$e_d = \xi_d - \Delta(\mathbf{x}_{s_d}, \dot{\mathbf{y}}_{p_d}, u_{e_d}). \tag{5}$$

Steady state value of the limit parameter, $\hat{y}_{p_{ss}}$,at the current time is found by inserting zeros to derivative terms i.e. $\dot{\mathbf{y}}_p = 0$:

$$\hat{y}_{p_{ss}} = \hat{g}_n^{-1}(\mathbf{x}_s, 0, u_e) + \Delta(\mathbf{x}_s, 0, u_e) + e_d. \tag{6}$$

2.1 Direct Adaptive Limit Margin Estimation

The method given in [6, 9] uses control sensitivities for control margin calculations.

For a known limit boundary, $y_{p_{lim}}$,the limit margin, $\hat{y}_{p_{marg_{ss}}}$, is:

$$\hat{y}_{p_{marg_{ss}}} = y_{p_{lim}} - \hat{y}_{p_{ss}}. \tag{7}$$

The sensitivity $S \in \Re$ of the limit parameter to the effective control input, u_e, can be calculated as:

$$S = \frac{\partial \hat{y}_{p_{ss}}}{\partial u_e} = \frac{\partial(\hat{h}_1^{-1}(x_s, 0, u_e) + \Delta(x_s, 0, u_e))}{\partial u_e}. \tag{8}$$

There is a linear relation between the limit and control margin:

$$\hat{y}_{p_{marg_{ss}}} = S\hat{u}_{e_{marg}}. \tag{9}$$

Note that, if $\hat{u}_{e_{marg}} = \hat{u}_{e_{lim}} - u_e$ then:

$$\hat{u}_{e_{lim}} = \frac{1}{S}\hat{y}_{p_{marg_{ss}}} + u_e \tag{10}$$

Some limit parameters may reach their limits during transient response. In such cases, Eq. (10) needs to be modified using the instant limit margin,$y_{p_{marg}} = y_{p_{lim}} - y_p$.

$$u_{e_{lim-upper}} = min\left(\left|\frac{1}{S}y_{p_{marg}}\right|, \left|\frac{1}{S}y_{p_{marg_{ss}}}\right|\right) + u_e \tag{11}$$

$$u_{e_{lim-lower}} = -min\left(\left|\frac{1}{S}y_{p_{marg}}\right|, \left|\frac{1}{S}y_{p_{marg_{ss}}}\right|\right) + u_e \tag{12}$$

Details on the control limit margin estimation methodology can be found in [6,9].

2.2 Neural Network Augmentation

2.2.1 Linear in the Parameter Neural Network

The adaptive element Δ can be constructed as:

$$\Delta(\bar{x}) = W^T \Phi(\bar{x}). \tag{13}$$

where, $\bar{x} = [x_s(t), \dot{\mathbf{y}}_p, u_e]^T \in \Re^r$ is the input vector, $W^* \in \Re^{m \times l}$ is the set of weights and, $\Phi(\bar{x}) = [\phi_1(\bar{x}), \phi_2(\bar{x}),\ ...,\ \phi_m(\bar{x})]$, where $\phi_i : \Re^r \to \Re$, $i = 1, 2, ..., m$ are known and bounded activation functions.

The weight update law is determined using concurrent learning [8] as:

$$\dot{W}(t) = \Gamma\big(\Phi(\bar{x})e^T + \sum_{j=1}^{p} \Phi_j(\bar{x})e_j^T\big) \tag{14}$$

where, Γ is a positive definite learning gain matrix. A proof of boundedness for the above weight update law is presented in [6].

2.2.2 Single Hidden Layer Neural Network

The adaptive element Δ can be constructed as:

$$\Delta(\bar{x}) = W^T \sigma(V^T \bar{x}). \tag{15}$$

where; $\bar{x} = [x_s(t), \dot{\mathbf{y}}_p, u_e]^T \in \Re^{r+1}$ is the input vector, $W^* \in \Re^{(m+1)\times l}$ is synaptic weights that connect hidden layer to output layer, $V^* \in \Re^{(r+1)\times m}$ is synaptic weights that connect input layer to hidden layer, and the sigmoidal activation functions vector is $\sigma(z) = [b_w, \sigma_1(z_1), \sigma_2(z_2),\ ...,\ \sigma_m(z_m)]$. Where,

$$\sigma_i(z_i) = \frac{1}{1 + e^{-a_i z_i}} \tag{16}$$

The approximation error, e, and the basis, $\bar{x}$ are recorded in history stack matrix and using the assumptions in [8]. The weight update law becomes:

$$\dot{W}(t) = \Gamma_W((\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})e^T + W_c \sum_{j=1}^{p} (\sigma(V^T\bar{x}_j) - \sigma'(V^T\bar{x}_j)V^T\bar{x}_j){e_j}^T) \tag{17}$$

$$\dot{V}(t) = \Gamma_V \bar{x} e^T W^T \sigma'(V^T\bar{x}) + V_c \sum_{j=1}^{p} \bar{x}_j {e_j}^T W^T \sigma'(V^T\bar{x}_j) \tag{18}$$

where;

$$W_c = I - \frac{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T}{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})} \tag{19}$$

$$V_c = I - \frac{\bar{x}\bar{x}^T}{\bar{x}^T\bar{x}} \tag{20}$$

A proof of boundedness for the above weight update law can be found in [8].

3 Implementation

3.1 Load Factor as the Critical Limit Parameter

As shown in Fig.1, an attitude stabilization system is used to help pilot to control the lateral dynamics. The critical limit parameter is taken as the load factor; thus, active controls are implemented on longitudinal channel.

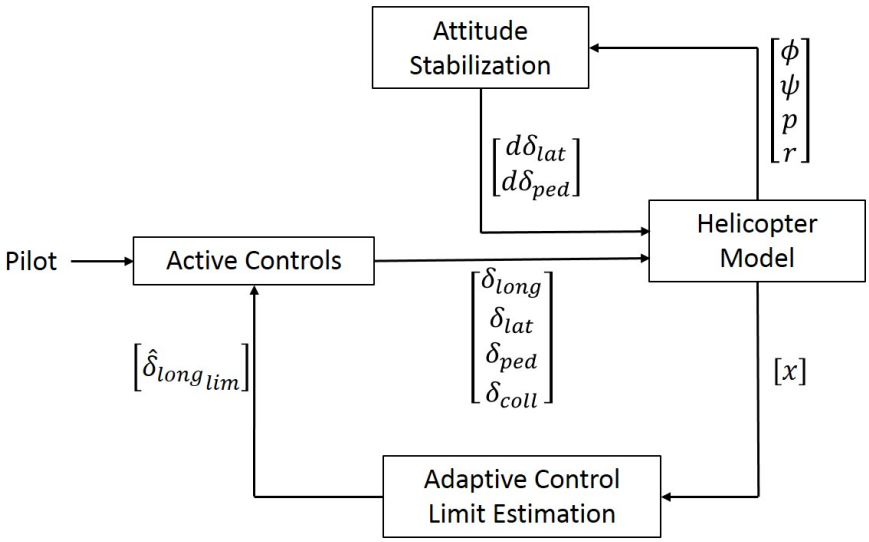

Fig. 1 Simulation Block Diagram

Load factor is given by:

$$n_z = 1 + \frac{Uq}{g}. \tag{21}$$

where, U is the forward speed, q is the pitch rate and g is the gravitational acceleration. Load factor dynamics are dominated by pitch rate dynamics. Therefore, pitch rate model is considered first. The fast states in longitudinal dynamics are the

pitch rate q and velocity in z-body w. The linear model for fast states can be written as:

$$\begin{bmatrix} \dot{q} \\ \dot{w} \end{bmatrix} = A_1 \begin{bmatrix} q \\ w \end{bmatrix} + B\delta_e \tag{22}$$

The pitch rate q, can be estimated in delayed time using the linear model inverse and an adaptive element [6]:

$$\hat{q}_d = A_1^{-1}([\hat{\dot{q}}_d \quad \hat{\dot{w}}_d] - B\delta_{e_d}) + \Delta(\hat{\dot{q}}_d, \hat{\dot{w}}_d, U_d, \theta_d, \delta_{e_d}, b_1) \tag{23}$$

Here, the derivative terms can be estimated using a smoothing method in delayed time, and Δ is the adaptive element.

The adaptive element is expressed with LPNN as $\Delta = W^T \Phi_d$, where the basis is:

$$\begin{array}{c} \Phi(i) = \phi_i(\hat{\dot{q}}_d, \hat{\dot{w}}_d), \;\; i = 1:2 \\ \Phi(3:9) = [\phi_3(U\theta) \;\; \phi_4(\delta_e) \;\; \phi_5(U) \; ... \\ ... \;\; \phi_6(\theta) \;\; \phi_7(\delta_e U) \;\; \phi_8(\delta_e \theta) \;\; b_1]^T \end{array} \tag{24}$$

If the adaptive element is expressed with SHL NN as $\Delta = W^T \sigma(V^T \bar{x})$, then the basis becomes:

$$\begin{array}{c} \bar{x}(i) = (\hat{\dot{q}}_d, \hat{\dot{w}}_d), \;\; i = 1:2 \\ \bar{x}(3:6) = [U, \theta, \delta_e, b_1]^T \end{array} \tag{25}$$

The steady-state value, $\hat{q}_{SS}$, is given by:

$$\hat{q}_{SS} = -A_1^{-1} B\delta_e + \Delta(0,0,U,\theta,\delta_e,b_1) + e_d \tag{26}$$

where, the delayed approximation error is $e_d = q_d - \hat{q}_d$.

The sensitivity of the pitch rate with respect to the longitudinal cyclic input, S_q, is

$$S_q = \frac{\partial \hat{q}_{SS}}{\partial \delta_{long}} = -A_1^{-1} B + \left[\frac{\partial \hat{q}_{SS}}{\partial \delta_{long}} \right]_{\dot{q}=0, \dot{w}=0} \tag{27}$$

Similarly, sensitivity of the load factor with respect to longitudinal cyclic input,S_{n_z}, becomes:

$$S_{n_z} = \frac{\partial \hat{n}_{z_{SS}}}{\partial \delta_{long}} = \frac{U S_q}{g} \tag{28}$$

Hence, using Eqs.(11) and (12), the upper and the lower control limits become:

$$\hat{\delta}_{long_{lim-upper}} = min\left(\left| \frac{1}{S} \hat{n}_{z_{marg}} \right|, \left| \frac{1}{S} \hat{n}_{z_{marg_{SS}}} \right| \right) + \delta_{long}. \tag{29}$$

$$\hat{\delta}_{long_{lim-lower}} = -min\left(\left| \frac{1}{S} \hat{n}_{z_{marg}} \right|, \left| \frac{1}{S} \hat{n}_{z_{marg_{SS}}} \right| \right) + \delta_{long}. \tag{30}$$

where, $\hat{n}_{z_{marg_{SS}}}$ is the steady state limit margin and $\hat{n}_{z_{marg}}$ is the limit margin based on measured load factor.

3.2 Simulator Environment with Active Side Stick Inceptor

Fig. 2 Simulator Test Bench

Several components come together to construct the simulator environment is shown on Fig. 2. These components are; Stirling Dynamics Active Side Stick Inceptor [10], Flight Link Helicopter Package [11], Saitek pilot controllers [12] and two desktop computers. The computers are equipped with Windows operating systems and are connected to each other through TCP/IP connections. The computer 1 runs the simulation controller (SMC), Simulink model, and multi-function display (MFD). SMC initializes the simulator. Moreover, connections between hardware and software can also be checked through SMC. The simulink model and the MFD are fed calculated states from computer 2. Computer 2 runs the flight model which uses pilot inputs to calculate aircraft states. A high fidelity helicopter model is utilized in simulations [14]. Flight Link Advanced Helicopter Package is also connected to computer 2. The package includes a pilot seat with cyclic,collective and pedal controls. In ad-

dition to these controls, Saitek controls connected to computer 1 are also available. This optional controls enable different controller configurations and give versatility to the simulator environment. Furthermore, The Stirling Dynamics Next Generation Inceptor, which is an active side stick controller is also available in the simulator. The active interceptor feeds the pilot with tactile cues based on a nonlinear force map designed for effective pilot cueing (Fig.4). The force feedback depends on both stick angle and the control margin. A block diagram of the simulator environment is shown in Fig. 3. In this paper, Stirling active inceptor is used for lateral and longitudinal input channels and Flight Link is used for collective and pedals.

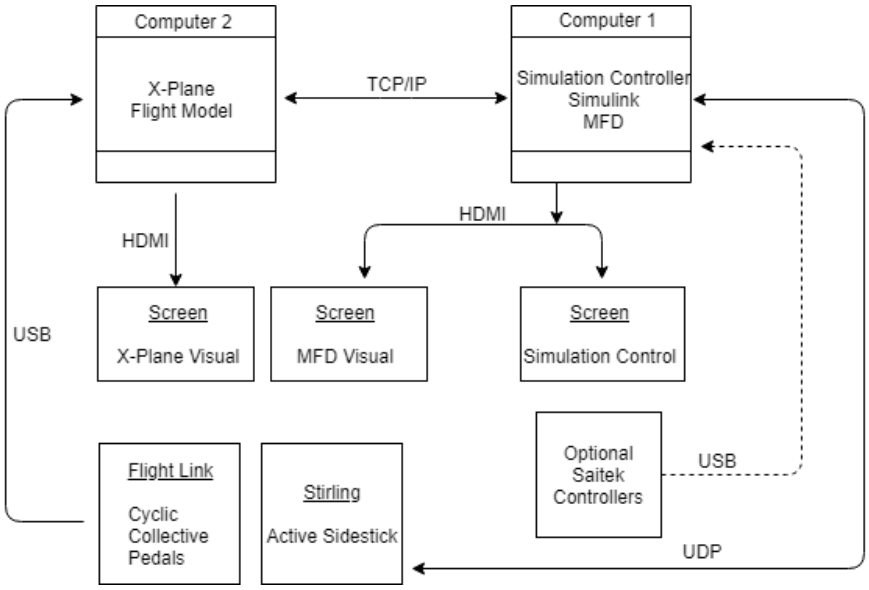

Fig. 3 Simulator Test Bench Block Diagram

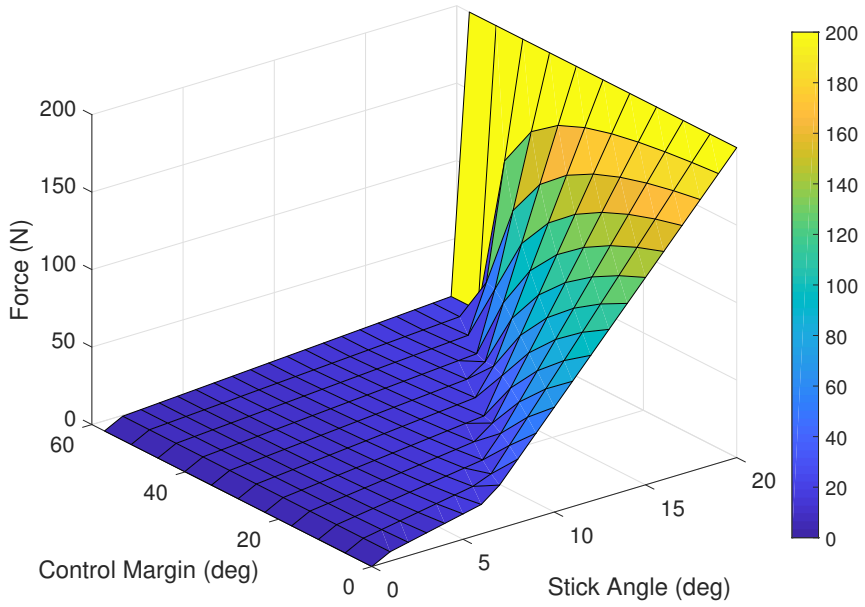

Fig. 4 Force Feedback Map

4 Simulation Results

The method explained in Sec. 3 is implemented on a high fidelity helicopter model [14]. 3 different test scenarios are simulated.
In the case1, various pitch up and pitch down maneuvers are considered on longitudinal channel. For this case, the adaptation is turned ON at $t = 10s$ and both LPNN and SHL NN is used separately. In Fig. 5 and Fig. 6 load factor output is plotted together with longitudinal cyclic input and control limits. Notice that, after adaptation is turned ON, control limits converge to their steady values. In Fig. 7 and Fig. 8 model error and the adaptive element, Δ, used for estimating the error is plotted together with minimum singular value of the history stack. Minimum singular value is maximized for data selection to history stack for concurrent learning. In Fig. 9, Fig. 10 weights of SHL NN and sensitivity is given. In Fig. 11 weights of LPNN is plotted with sensitivity. Note that sensitivity converges to its steady optimal value. Finally, in Fig. 12 helicopter states corresponding to this test maneuver is provided. Lateral dynamics are kept around their trim values.

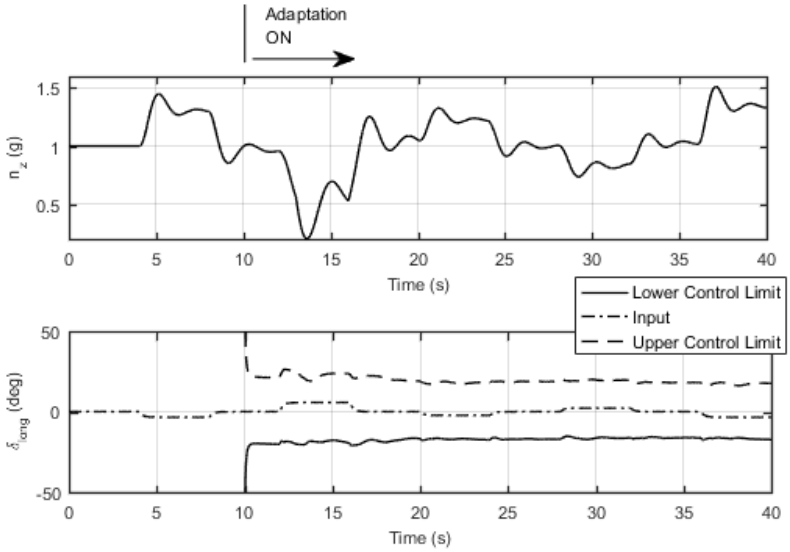

Fig. 5 Case 1: Load Factor and Control Limits with SHL NN

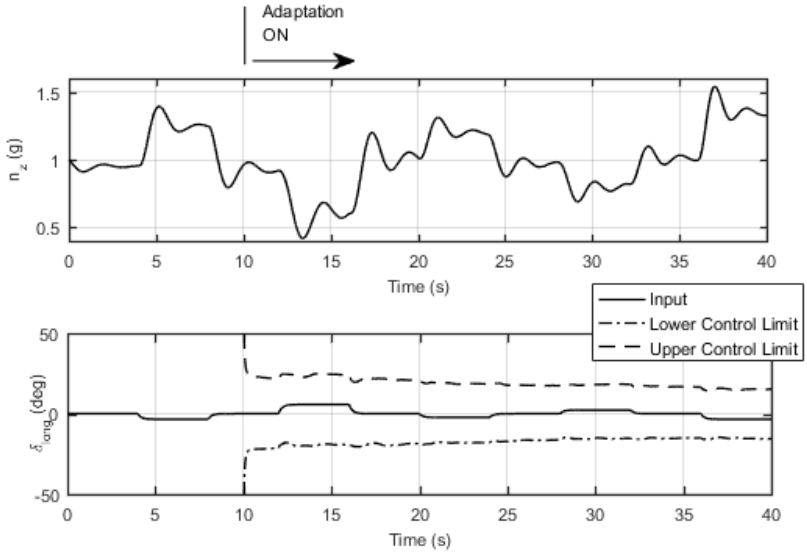

Fig. 6 Case 1: Load Factor and Control Limits with LPNN

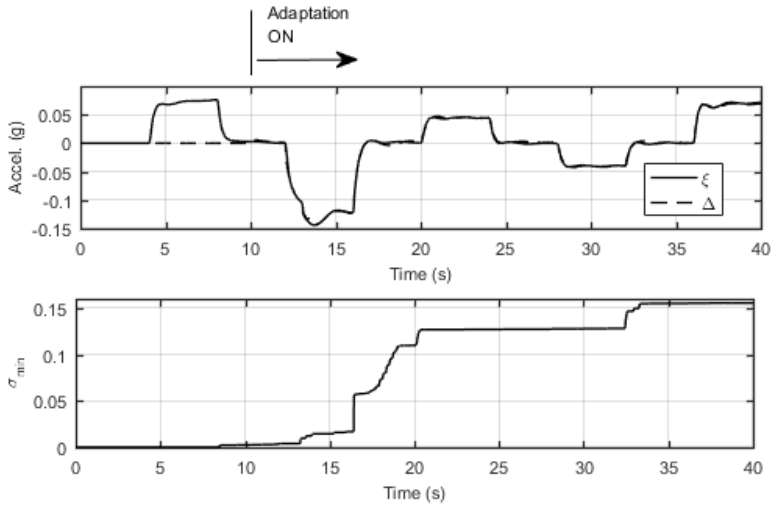

Fig. 7 Case 1: Model Error and SVD with LPNN

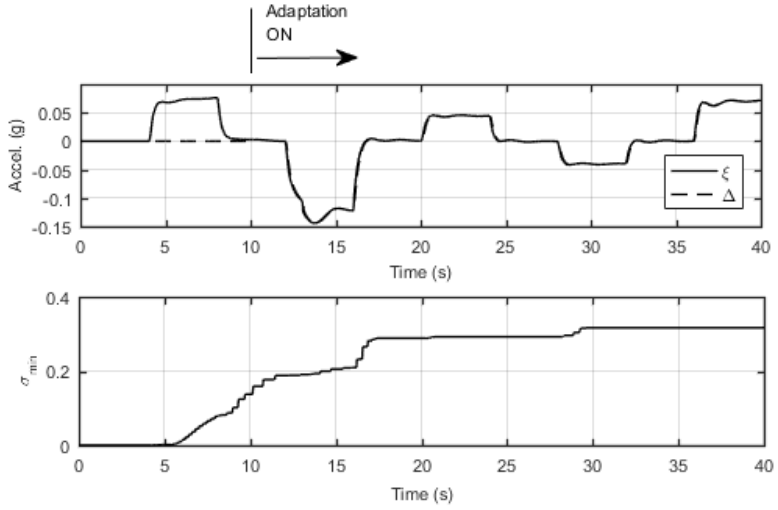

Fig. 8 Case 1: Model Error and SVD with SHL NN

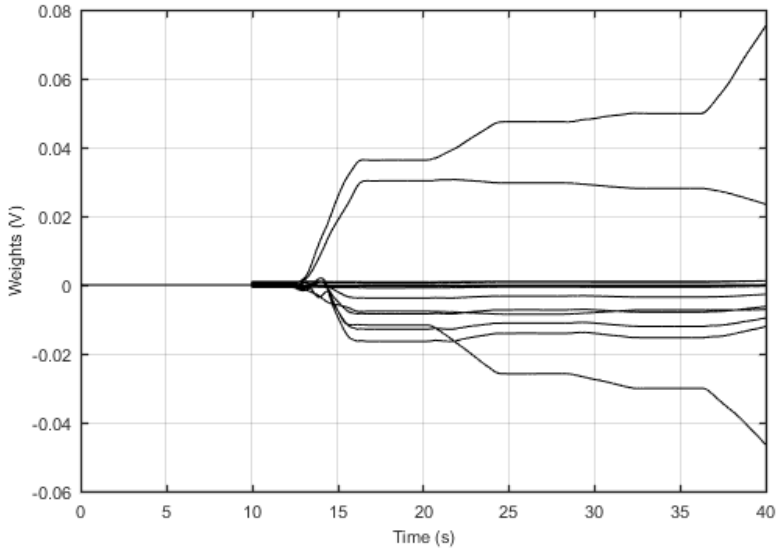

Fig. 9 Case 1: Weights connecting input layer and hidden layer

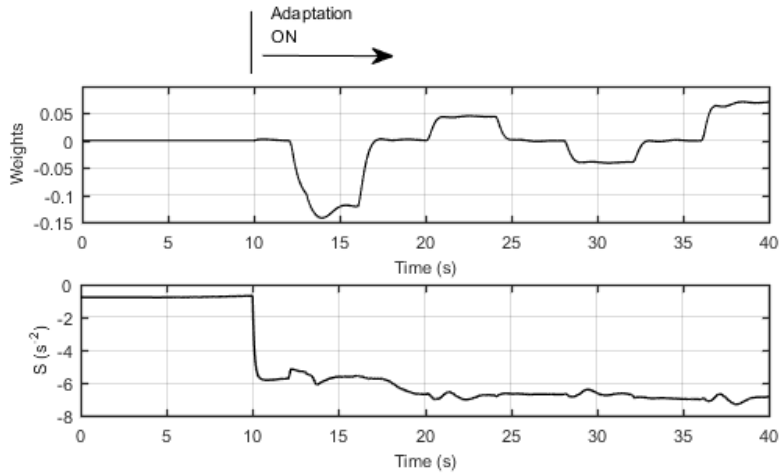

Fig. 10 Case 1: Sensitivity and Weights connecting hidden layer and output layer

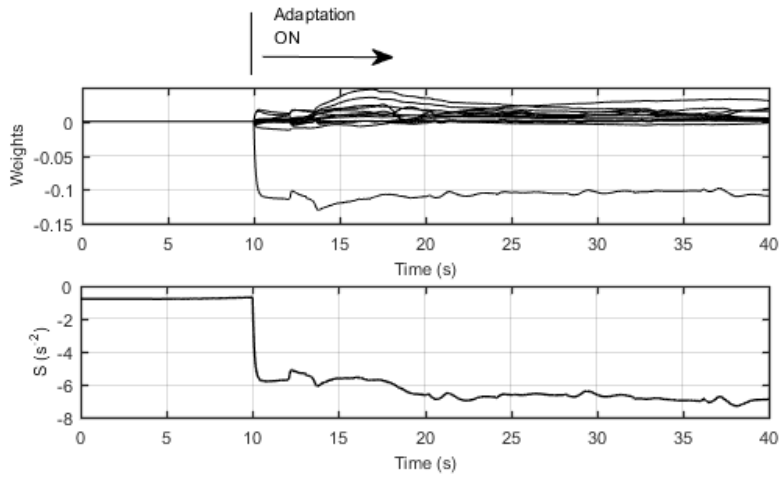

Fig. 11 Case 1: Sensitivity and Weights for LPNN

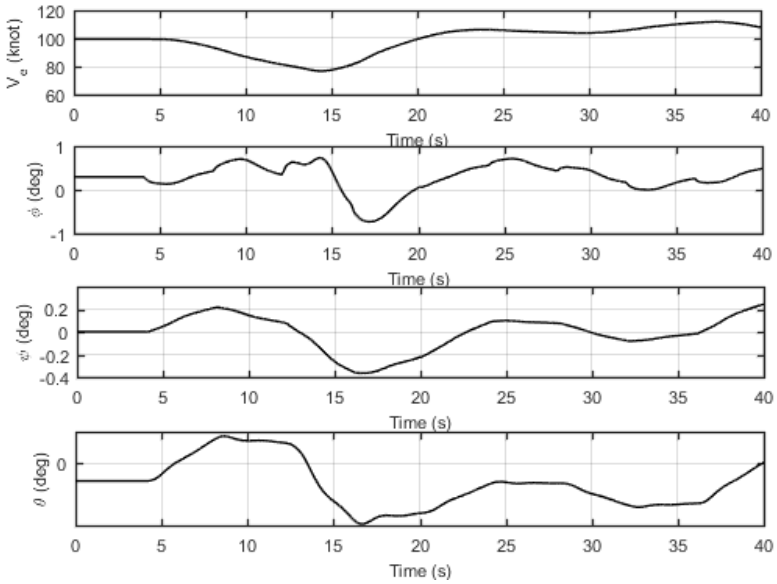

Fig. 12 Case 1: Helicopter States

In case2, immediately follows case1 and an aggressive maneuver where load factor violates its limit value is given after $t = 40s$. In Fig. 13 and Fig. 14 load factor and longitudinal cyclic input is plotted for aggressive input. Notice that, in both cases, the input violates the control limits before limit parameter reaches its limit value with SHL NN predicting limit violation slightly earlier than LPNN.

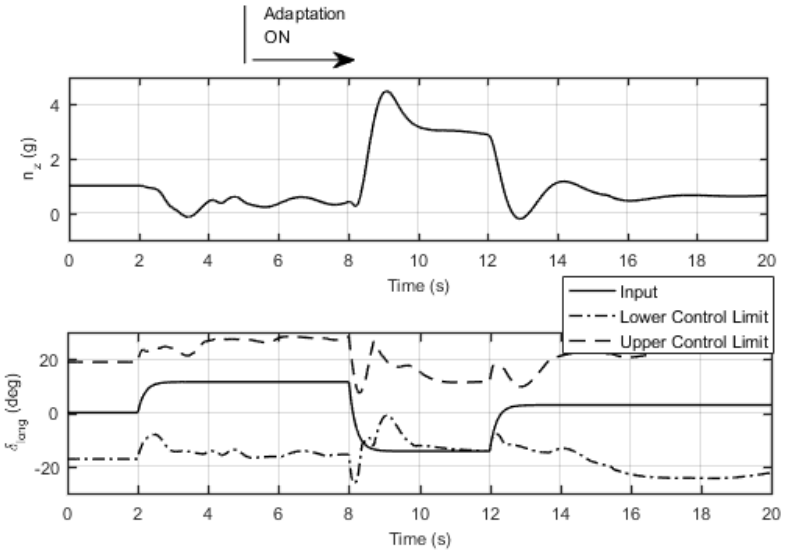

Fig. 13 Case 2: Load Factor and Control Limits with SHL NN

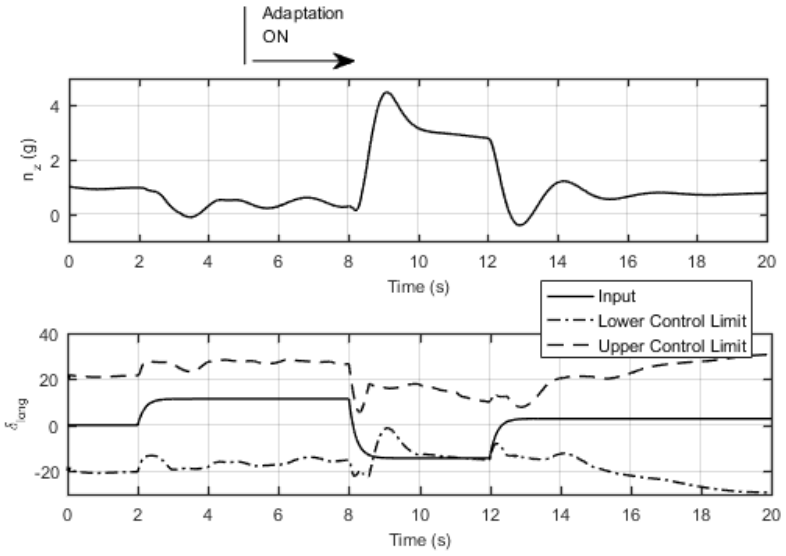

Fig. 14 Case 2: Load Factor and Control Limits with LPNN

Finally, in case 3, the active side stick is used for limit avoidance with SHL NN. The side stick is programmed to give different force feedbacks at different angles based on the control margins as shown in Fig. 4. Fig. 15 shows the load factor response and control input with control limits. For this scenario, first the helicopter speeds up with a nose down maneuver, then with a sudden pull up the helicopter approaches its limit. Next, an even more aggressive pull up is initiated in attempt to exceed limits; however, stick force prevents limit avoidance. In Fig. 16 how model error is approximated by adaptive element Δ can be seen. Weights of the SHL NN and sensitivity are provided in Fig. 17 and Fig. 18. Helicopter's response to this maneuver is plotted in Fig. 19 where lateral dynamics are always kept at their trim values. Finally, Fig. 20 stick force is plotted. Note that, stick force first increases slowly as the helicopter approaches the envelope boundary, then makes a stiffer increase in order to prevent limit violation. Furthermore, more aggressive inputs generate greater force feedback.

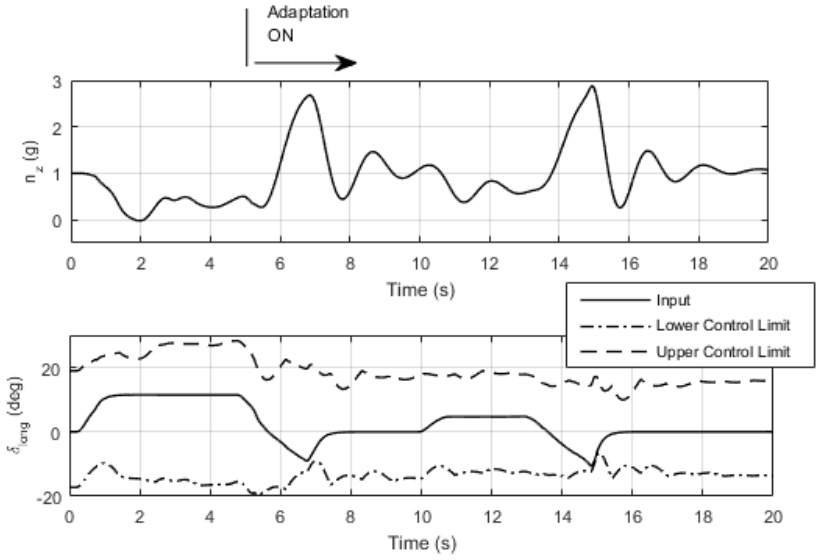

Fig. 15 Case 3: Load Factor and Control Limits with SHL NN

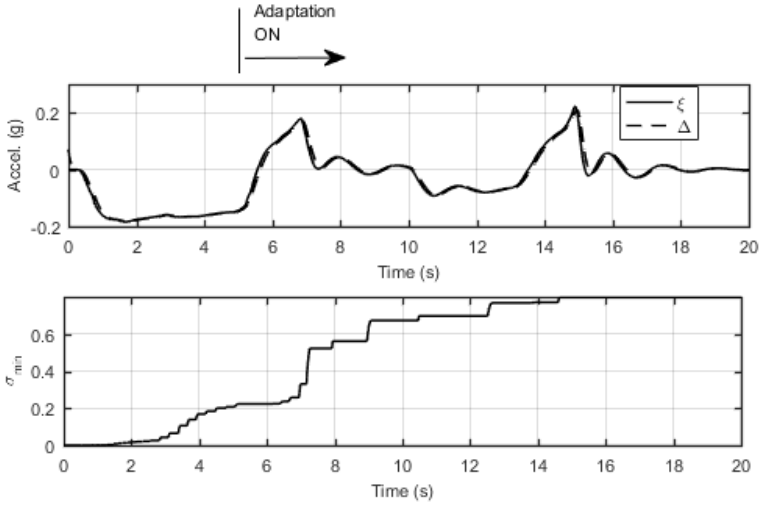

Fig. 16 Case 3: Model Error and SVD with SHL NN

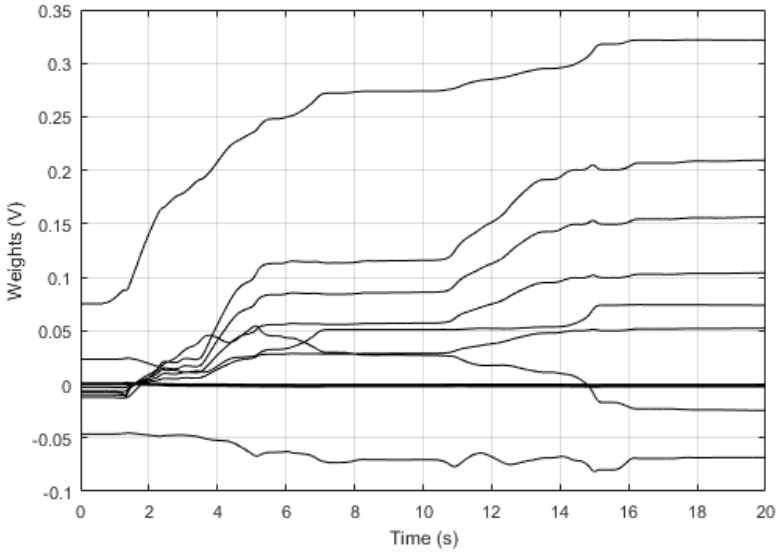

Fig. 17 Case 3: Weights connecting input layer and hidden layer

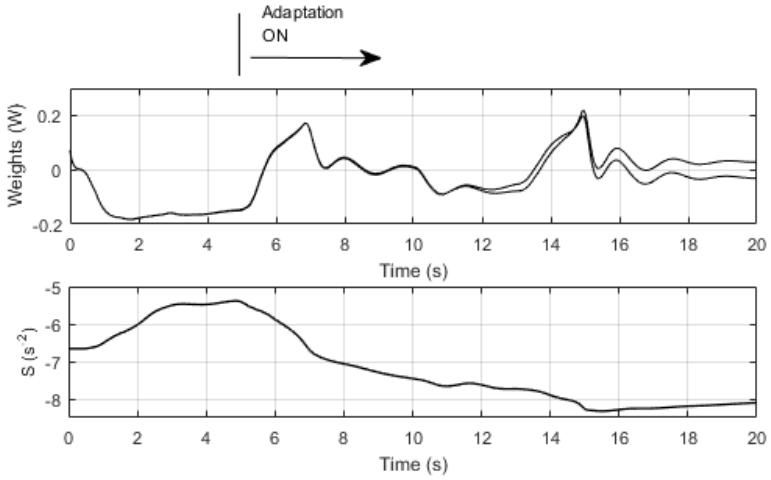

Fig. 18 Case 3: Sensitivity and Weights connecting hidden layer and output layer

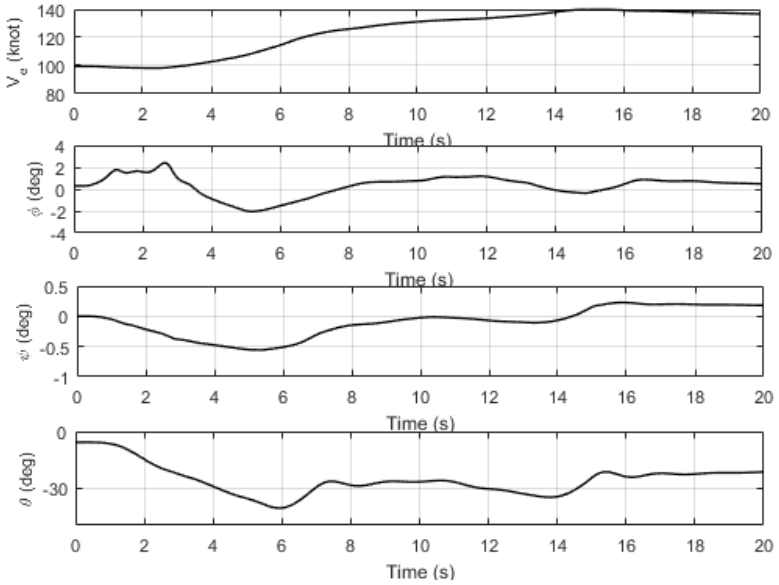

Fig. 19 Case 3: Helicopter States

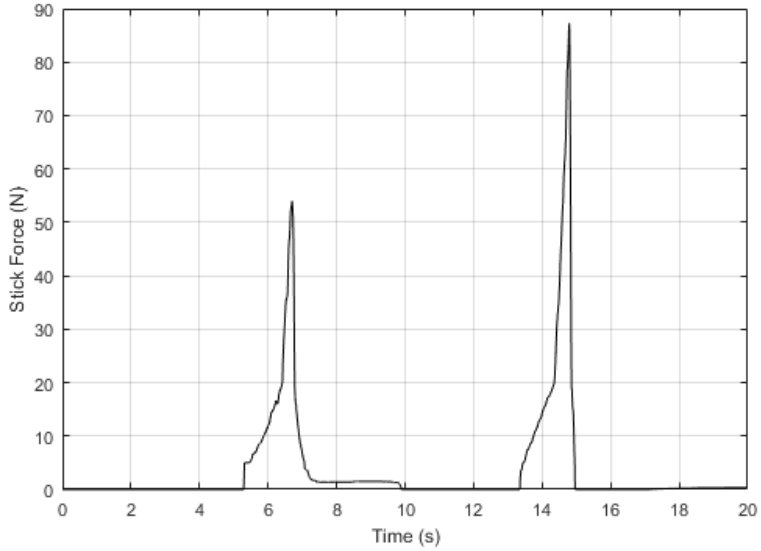

Fig. 20 Case 3: Stick Force

5 Conclusions and Future Work

In this paper, load factor limit avoidance for helicopters is done using the control sensitivity based adaptive limit prediction method stated in [6] and [9]. Control margins for longitudinal channel is estimated online with concurrent learning using both Linear in the Parameter Neural Network (LPNN) and Single Hidden Layer Neural Network(SHL NN). Furthermore, limit prediction algorithms are utilized in a simulation environment that consist of a high fidelity helicopter model and an active side stick capable of giving force feedback cues to the pilot based on the limit margin estimations.

Contributions of this work is twofold. First, SHL NN implemented with existing methods in [6, 9] and comparison of LPNN and SHL is done on the same platform. No major differences between two neural networks is observed. Our chosen approximate model behaves similarly to the real dynamics of the system and it generates small model error that could be indeed compensated with both neural networks. In the future, a more challenging approximate model can be selected to better observe the differences between LPNN and SHL NN. Secondly, envelope protection algorithms are implemented on a simulator with an active interceptor. Force feedback cues are generated based on the estimations done in the algorithm. The stick force increases as the helicopter approaches its limits; this is so called *soft stop*. This approach not only provides limit avoidance, as the force becomes too great at the limit, but also informs the pilot that the helicopter is approaching its limits. In the future, other force feedback schemes can be explored and tested with pilots in order to better understand which approach gives a better *feeling*.

References

1. Whalley, M. S., Hindson, W., and Thiers, G., A Comparison of Active Sidestick and Conventional Inceptors for Helicopter Flight Envelope Tactile Cueing, *Proceedings of the American Helicopter Society 56th Annual Forum*, AHS International, Alexandria, VA, May 2000, pp. 181204.
2. Einthoven, P., Miller, D., Nicholas, J., and Margetich, S., Tactile Cueing Experiments with a Three Axis Sidestick, *Proceedings of the American Helicopter Society 57th Annual Forum*, AHS International, Alexandria, VA, May 2001.
3. Yavrucuk, I., and Prasad, J., Online Dynamic Trim and Control Limit Estimation, *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 16471656. doi:10.2514/1.59116.
4. J.F.Horn, Calise, A., and Prasad, J., Flight Envelope Cueing on a Tilt-Rotor Aircraft Using Neural Network Limit Prediction, *Journal of the American Helicopter Society*, Vol. 46, No. 1, Jan 2001, pp. 2331. doi:10.4050/JAHS.46.23.
5. J.F.Horn, Calise, A., and Prasad, J., Flight Envelope Limit Detection and Avoidance for Rotorcraft, *Journal of the American Helicopter Society*, Vol. 47, No. 4, Oct. 2002, pp. 253262. doi:10.4050/JAHS.47.253.
6. Gursoy, G., and Yavrucuk, I., Direct Adaptive Limit and Control Margin Estimation with Concurrent Learning, *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 6, June 2016. doi:10.2514/1.G001515.

7. Gursoy, G., and Yavrucuk, I., Concurrent Learning Enabled Adaptive Limit Detection for Active Pilot Cueing, *Journal of Aerospace Information Systems*, Vol. 11, No. 9, 2014, pp. 542550. doi:10.2514/1.I010205.
8. Chowdhary, G., Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, 2010.
9. Gursoy, G., and Yavrucuk, I., Non-Iterative Adaptive Vertical Speed Limit and Control Margin Prediction for Fly-By-Wire Helicopter, *Proceedings of the American Helicopter Society 71th Annual Forum*, AHS International, May 2015.
10. Stirling Dynamics website: https://www.stirlingdynamics. com/ accurate as of Sept 2018.
11. Flight Link website: http://www.flightlink.com/ accurate as of Sept 2018.
12. Saitek website: http://www.saitek.com/ accurate as of Sept 2018.
13. Yavrucuk, I., Tarimci, O., Katircioglu, M., Kubali, E., and Yilmaz, D., A new Helicopter Simulation and Analysis Tool: Helidyn+, *Proceedings of the Thirty- Sixth European Rotorcraft Forum*, ERF International, Paris, France, Sept. 2010.
14. Yavrucuk, I. Tarimci, O., Katircioglu., M., Kubali, E., Yilmaz, D., A new Helicopter Simulation and Analysis Tool: Helidyn+, September 2010, *In Proceedings of the Thirty- Sixth European Rotorcraft Forum*, Paris, France.