ID 229

Limit Margin Prediction For Helicopters Using Long Term Learning Adaptive Neural Networks

Ilkay Yavrucuk* **Gonenc Gursoy****

Middle East Technical University, Ankara, 06531, Turkey

Abstract:
In this paper a methodology is presented for estimating helicopter limit margins using real-time learning dynamic models featuring long term (concurrent) learning adaptive neural networks. Limit margin estimations can be used in envelope protection systems of fly-by-wire helicopters. Linear models are compensated with adaptive neural networks to construct adaptive models of relevant aircraft dynamics. A stack of data collected during flight is used to update the network weights. The data stack for learning is made up of instantaneous measured data and recorded data. Rules for recording relevant data are established. It is observed that using recorded data in a stack can cancel out modeling errors faster and result in better predictions of approaching steady state limits compared to using instantaneous measured data only.

Keywords: neural network, long term learning, limit detection, envelope protection, pilot cueing

1. INTRODUCTION

Envelope protection is an area of research where the focus is to cue the pilot such that known envelope limits are not violated during flight. These systems can be used for fly-by-wire fixed and rotary wing aircraft in order to improve handling qualities and safety. Another objective of an Envelope Protection System (EPS) is to allow the aircraft to use its full flight envelope without exceeding its flight envelope limits. In literature, these type of cueing systems are also known as carefree maneuvering systems and exhibit *limit prediction* and *limit avoidance*. In limit prediction, possible violations of limits are detected with an effective lead time before the actual violation occurs. The difference between the current value of the limit variable and the actual limit is known as the *limit margin*, whereas the allowable control travel to reach that limit is the *control margin*. Such information can be used effectively in limit avoidance as a preventive action for pilot cueing.

A methodology is presented in [1,2,3] for estimating approaching limits and allowable control travel for VTOL aircraft. The method uses adaptive neural network based dynamic models for online estimation. Dynamic nonlinear models are build during flight to approximate the corresponding helicopter dynamics. The dynamic models are made up of local linear models with adaptive neural networks to compensate for the difference between the real dynamics and the approximate dynamics posed by the linear models. This information is then used to either cue the pilot or to adjust controller commands automatically. The method is applicable to limits that reach their maximum values at their steady state.

While the method is presented with promising results, in practice the adaptation capability is limited by the network structure, available training data and update laws. In particular, establishing network memory is not always straightforward -even for repeated maneuvers. A common observation is that there exists a re-learning process of the neural network. In other words, the online models are locally accurate and in practice do not exhibit much memory.

In this paper, we adopt the neural network update law presented in [4,5] for long term (concurrent) learning to the methods of [1,2,3] for limit detection. Here, the neural network update law is arranged such that both instantaneous sensor data and recorded data is used simultaneously in the update. This update law alleviates the rank-1 limitation in the update, a well known restriction in the update law using instantaneous sensor data only. The criteria to record adaptation data is modified to make use of data relevant for limit and control margin estimations.

The paper introduces the methodology and the modified neural network update laws for long term learning. Later a non-linear helicopter model is obtained using the Helidyn+ software program for a generic utility helicopter simulation. The code is embedded into Matlab/Simulink for batch simulations. Simulation results indicate a far better adaptation and more accurate estimation especially to repetitive maneuvers, when long term learning is used.

2. THEORETICAL DEVELOPMENT

The equations of motion of an aircraft are represented with the following nonlinear state equations:

* Assistant Professor, Department of Aerospace Engineering, email:yavrucuk@metu.edu.tr

**Graduate Student, Department of Aerospace Engineering, email:gonencgursoy@gmail.com

$$\dot{x} = f(x,u); \qquad x(t_0) = x_0, \qquad x \in \Re^n, u \in \Re^p, \quad (1)$$

where x is the state vector with known initial conditions, x_0, and u is a known control vector.

Similarly, the dynamics of a limit state of relative degree one can be represented as

$$\dot{y}_p = f(y_p, x, u); \qquad y_p(t_0) = y_{p_0} \quad (2)$$

where y_p is a variable representing the limit state. The limit state considered here is assumed to reach its maximum value asymptotically at its steady state, and is generally fast enough to be relevant for an EPS application.

The following approximate dynamics for the limit state dynamics is used:

$$\dot{\hat{y}}_p = \hat{f}(y_p, x_r) + \Delta(y_p, x, u) + K(y_p - \hat{y}_p), \quad (3)$$

A 'hat' denotes an estimation of a variable, and K is called the observer gain matrix [1]. x_r are selected states depending on the chosen linear approximation.

Following [2] the error dynamics becomes

$$\dot{e} = -Ke + \xi - \Delta. \quad (4)$$

where $e = y_p - \hat{y}_p$ and ξ is the true modeling error. Thus, when the modeling error, ξ, can be cancelled through Δ, with a positive definite matrix K, the modeling error will decay asymptotically to zero.

Since the maximum value of the limit variable is assumed to be its steady state value, for a given control input the maximum value of the limit vector is estimated when

$$\dot{\hat{y}}_p = 0 \quad (5)$$

Using the relationship in eqn.(3) an estimate of the steady state condition, $\hat{y}_{p_{ss}}$, can be found, which is in this case the maximum value to be reached for a given control input and flight condition:

$$\hat{f}_1(\hat{y}_{p_{ss}}, x_r, u) + \Delta(\hat{y}_{p_{ss}}, x, u) + Ke = 0. \quad (6)$$

Therefore, for a known value of the flight envelope limit, $y_{p_{lim}}$, an estimate of the limit margin becomes:

$$\hat{y}_{marg} = y_{p_{lim}} - \hat{y}_{p_{ss}} \quad (7)$$

2.1 Single Hidden Layer Neural Network Augmentation

Single hidden layer neural networks (SHLNN) are used to approximate the function Δ mentioned above. SHLNNs are known to be universal approximators and consist of an input layer, a hidden layer and an output layer. The estimator dynamics can be written following [6]:

$$\dot{\hat{y}}_p = A[\hat{y}_p \;\; x_r]^T + B_1 u + W^T\beta(V^T\bar{x}) + K(y_p - \hat{y}_p) \quad (8)$$

Here β is the activation function vector of the neural network and $\bar{x}$ is the neural network input vector which is defined as $\bar{x} = [1 \; y_p \; x \; u]$.

The classic update law for ultimately bounded error and weight signals is [1]:

$$\dot{\hat{W}} = -(\beta - \beta^{'}\hat{V}^T\bar{x})(e^T P)L_{R_W} \quad (9)$$

$$\dot{\hat{V}} = -L_{R_V}\bar{x}(e^T P)\hat{W}^T\beta^{'}(V^T\bar{x}) \quad (10)$$

where, L_{R_V} and L_{R_W} are the corresponding learning rates. Assume that P is the solution of the following Lyapunov equation:

$$(-K)^T P + P(-K) = -Q \quad (11)$$

where $Q > 0$.

Long Term Learning Through Recorded Data

Rewrite the model tracking error dynamics as:

$$\dot{e} = -Ke + \xi - W^T\beta(V^T\bar{x}) \quad (12)$$

$$r = \xi - W^T\beta(V^T\bar{x}) \quad (13)$$

In eqn.(12), the difference between the current modeling error (ξ) and the adaptive neural network output $W^T\beta(V^T\bar{x})$ is called the residual signal r and is used for online learning. The residual signal in the form of eqn.(13) doesn't contain any past information. The residual signal can be written in a more general form [4,5]:

$$r_{c_i} = \xi_i - W^T\beta(V^T\bar{x}_i) \quad (14)$$

Variables with subscript 'i' refer to the information of the i^{th} stored data point of the history stack. The long term learning law makes use of the residuals of a past time history. Using the residual signal of eqn.(14), the long term learning law attempts to reduce the difference between the stored estimate of the modeling error (ξ_i) and the neural network output $W^T\beta(V^T\bar{x}_i)$ which is also based on stored state information. As a result, recorded data is used in the training of the networks in addition to instantaneous sensor data. In literature, the use of time history data in the network weight update law is referred to as long term, concurrent learning or background learning [4,5].

Selection of Data Points for Long Term Learning

A key part in the design is to select the relevant past data to be used in long term learning. A common criteria used is to simply record data that are sufficiently different from each other [4,5,7]:

$$\frac{(\bar{x} - \bar{x}_p)^T(\bar{x} - \bar{x}_p)}{\bar{x}^T\bar{x}} > \epsilon_{\bar{x}} \quad (15)$$

where p denotes the last point stored in the history stack. The data selection criteria given in eqn.(15) indicates recording data that is 'sufficiently different' from each other. Since most of it happens in the transient response, we call data recorded using eqn.(15) as the transient data stack.

In addition to the transient data stack criteria we impose the following criteria:

$$\epsilon_{y_1} < ([x_l(k) - x_l(k-1)]^2 + [x_l(k) - x_l(k-2)]^2 + \\ ... + [x_l(k) - x_L(k-N)]^2)^{1/2} < \epsilon_{y_2} \quad (16)$$

and

$$\epsilon_{z_1} < ([\delta_l(k) - \delta_l(k-1)]^2 + [\delta_l(k) - \delta_l(k-2)]^2 + \\ ... + [\delta_l(k) - \delta_l(k-N)]^2)^{1/2} < \epsilon_{z_2} \quad (17)$$

Here x_l is the one dimensional limiting state in consideration, $x_l(k)$ is the k^{th} and current state, consequently

$x_l(k-N)$ represents the state information of N time steps before the current state. δ's are the corresponding controls. Here, the algorithm checks for a steady state condition for the limiting state and controls, and when the criteria is met data is added into the recorded stack to be used in long term learning. We call the data recorded using eqns. (16),(17) as the steady state data stack.

In [8,9] transient and steady state stacks are compared in the scope of the limit detection problem. It is observed that using steady state data on top of the transient data increases the performance of steady state predictions over time. In this paper we use both stacks for long term learning algorithm. Stack size of 100 points are selected, half of the stack is used for storing transient data and the rest is used for steady state data.

Once a flight condition meets any of the selection criteria all relevant information of that flight condition, such as the state vector, control vector, modeling error, etc. is stored. If the stack becomes large the oldest data in the history stacks is replaced with more recent data.

Long Term Learning Weight Update Law

For the simulations presented in this paper the following NN weight adaptation law is used [5]:

$$\begin{aligned}\dot{W} &= -(\beta - \beta' V^T \bar{x}) r^T L_{R_W} \\ &-W_c \sum_{i=1}^{p} (\beta_i - \beta_i' V^T \bar{x}_i) r_{c_i} L_{R_{W_s}}\end{aligned} \tag{18}$$

$$\begin{aligned}\dot{V} &= -L_{R_V} \bar{x} r^T W^T \beta' (V^T \bar{x}) \\ &-V_c \sum_{i=1}^{p} L_{R_{V_s}} \bar{x}_i r_{c_i}^T W^T \beta' (V^T \bar{x}_i)\end{aligned} \tag{19}$$

Here, $L_{R_{V_s}}$ and $L_{R_{W_s}}$ are the learning rates of the history stack (including both transient and steady state data). W_c and V_c are orthogonal projection operators and are defined as:

$$W_c = (I - \frac{\beta \beta^T}{\beta^T \beta}) \tag{20}$$

$$V_c = (I - \frac{L_{R_W} \bar{x} \bar{x}^T L_{R_W}}{\bar{x}^T L_{R_W} L_{R_W} \bar{x}}) \tag{21}$$

By using orthogonal projections, it is possible to constrain background learning to the nullspace of the online adaptation with instantaneous data only [5].

3. SIMULATION RESULTS

In this section, we show results by estimating limit margins using adaptive long term learning. The goal is to estimate the steady state value of a limit variable immediately after the controls are applied. Note that the the limit variable is chosen to have its maximum value at its steady state. Moreover, we compare results with cases where classic adaptation based on instantaneous data only is used.

Helicopter Flight Dynamics Model

A nonlinear helicopter model is build using the helicopter flight dynamics environment called HeliDyn+ [10]. HeliDyn+ allows the user to generate a helicopter math model by selecting model libraries for various helicopter components, such as models for the main rotor, inflow, tail rotor, fuselage, landing gear etc. The tool, then allows the user to extract a dynamic link library (.dll) that can be integrated into a C/C++ simulation environment, Matlab /Simulink, and FlightGear. The helicopter model used in this work is based on the geometry of the Uh-1h utility helicopter. The model is nonlinear, and uses a Peters-He inflow model, second order flap dynamics, Bailey's tail rotor model, vertical, horizontal tail models, ground effect models, ground reactions, etc. The model is extracted in Helidyn+ and integrated into Matlab/Simulink for this study.

Load Factor Steady State Estimation

For the simulations presented below the helicopter model is exposed to longitudinal cyclic inputs starting from a trimmed flight condition. The goal is to predict the maximum value of a limit variable, the load factor, at the time when the longitudinal cyclic control, δ_e, is applied. A controller is designed to keep the lateral dynamics in equilibrium.

An approximation of the load factor dynamics is represented with the following equation:

$$\begin{aligned}\dot{\hat{n}}_z = [a_1 \ a_2][\hat{n}_z \ V]^T + [b](\delta_e) + Ke... \\ ... + \Delta(\hat{n}_z, \delta_e, V, 1)\end{aligned} \tag{22}$$

Here, n_z is the load factor and V represents the forward speed. The maneuvering steady states are found using an iterative solution for $\hat{n}_{z_{ss}}$:

$$\begin{aligned}0 = [a_1 \ a_2][\hat{n}_{z_{ss}} \ V]^T + [b](\delta_e) + Ke... \\ ... + \Delta(\hat{n}_{z_{ss}}, \delta_e, V, 1)\end{aligned} \tag{23}$$

The a_1, a_2 and b_1 constants are chosen with rather large modeling errors ($a_1 = -3$, $a_2 = 0.01$, $b = -1.5$).

The neural network input vector is $\bar{x} = [1 \ n_z \ V \ \delta_e]$. Gaussian and complementary Gaussian activation functions are used in the hidden layer of the neural network.

Three sets of simulations are performed: First, instantaneous learning rates are chosen low and the benefit of additional long term learning is demonstrated. Next, intentionally high instantaneous learning rates are applied. A well known fact is the chattering in the limit state estimation when high learning rates are chosen. The additional long term learning is shown to improve estimations when high learning rates are used. As a third case moderate instantaneous learning rates are chosen. This time, new artificial modeling errors are introduced during the simulation and prediction responses are compared. Also, for the third case we compare the predictions of the two methods after freezing the weights.

Learning rates and observer gains of the three scenarios are presented in Tables 1,2 and 3.

Table 1. Design Parameters of Scenario-1

$K = 30$	$L_{R_W} = 10$	$L_{R_V} = 10$
$L_{R_{W_s}} = 1$	$L_{R_{V_s}} = 1$	

Table 2. Design Parameters of Scenario-2

$K = 30$	$L_{R_W} = 5000$	$L_{R_V} = 5000$
$L_{R_{W_s}} = 1$	$L_{R_{V_s}} = 1$	

Table 3. Design Parameters of Scenario-3

$K = 30$	$L_{R_W} = 1000$	$L_{R_V} = 1000$
$L_{R_{W_s}} = 1$	$L_{R_{V_s}} = 1$	

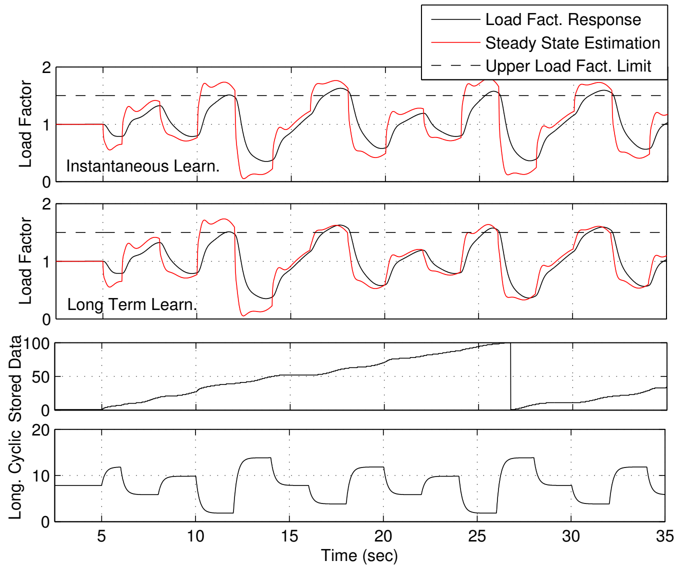

Fig. 1. Comparison of Steady State Est. (Scenario-1)

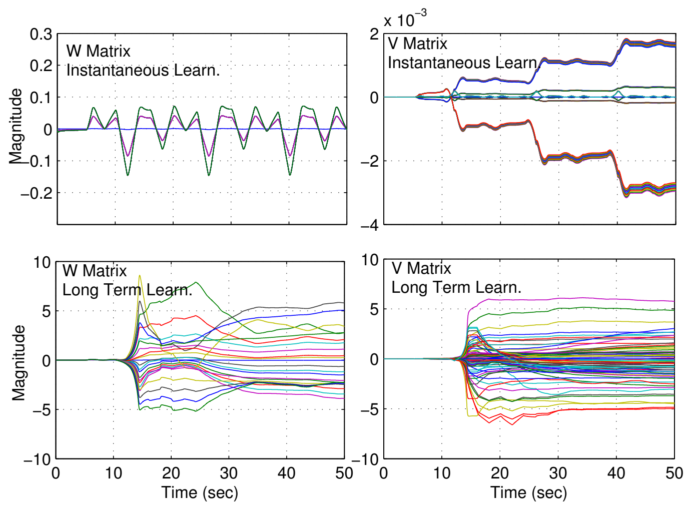

Fig. 2. Comparison of Weight Updates (Scenario-1)

In Fig.1 the load factor response for the first scenario is plotted along with dynamic trim predictions using long term learning and instantaneous learning. In this scenario, learning rates L_{R_W} and L_{R_V} are chosen low. As a result, using only instantaneous adaptation the steady state estimates are erroneous. It is observed that using data from a history stack improves the estimations over time. The network weight history of these simulations can

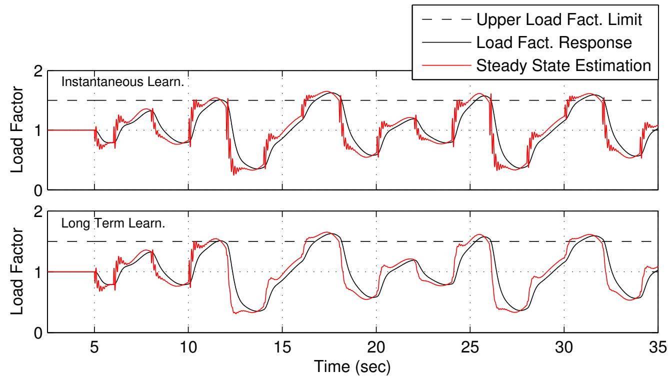

Fig. 3. Comparison of Steady State Est. (Scenario-2)

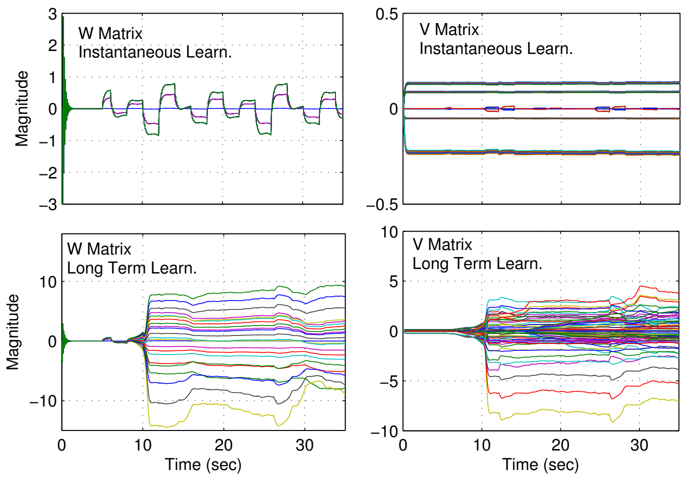

Fig. 4. Comparison of Weight Updates (Scenario-2)

be seen in Fig.2. Using long term learning, neural network weights converge to steady states much faster, whereas weights of the instantaneous learning network seems to re-learn the maneuver. The data storing activity of long term learning is shown in Fig.1.

Note that the steady state estimates are available at the time when the control is applied, while the actual state is still in its transient phase. Here, an artificial load factor upper limit is set to 1.5g (dotted line in Fig.1). The difference between this limit and the steady state prediction is an estimate of the limit margin This information becomes valuable when it is desired to timely cue to pilot on an approaching limit, as it will be available at the time the pilot introduces the input.

In the second scenario, adaptations are made faster choosing higher learning rates. Neural network adaptations are sensitive to learning gains. In this case the high network gains resulted in jittery steady state predictions. In Fig.3 steady state predictions using long term learning and instantaneous learning are compared. The jittery behavior of the estimations are compensated using long term learning and the estimations become more accurate. Weight updates of this simulation are presented in Fig.4. Long term learning neural network weights converge to their steady states, whereas the ones obtained using instantaneous

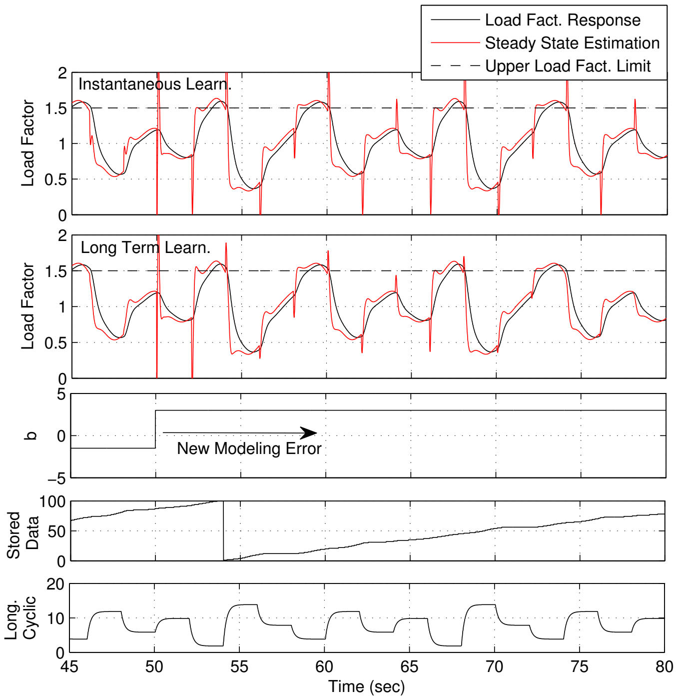

Fig. 5. Comparison of Steady State Est. (Scenario-3)

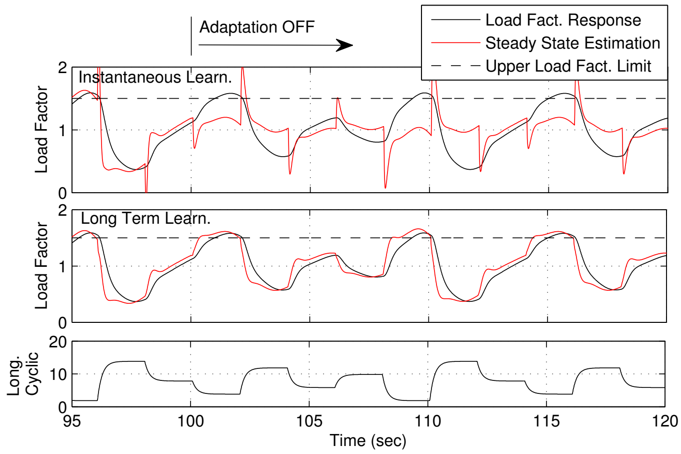

Fig. 6. Comparison of Steady State Est. (Scenario-3)

learning do not. The re-learning process of instantaneous learning is observed in Fig.4.

As a third scenario, learning rates of the network using instantaneous data are chosen to be moderate compared to the other scenarios. This time we try to compare prediction performance of the two methods to newly introduced modeling errors. In Fig.5 a new modeling error is introduced at about 50s into the simulation by changing the sign and the magnitude of the constant b of the approximate linear model. Figure 5 shows that using long term learning, the new modeling error is much faster compensated as maneuvers are repeated, hence, more accurate predictions of the steady state values are obtained. On the other hand, instantaneous learning only results in inaccurate predictions.

In order to demonstrate the fact that the weights actually converge to their true values when long term learning is used all network weights are frozen at about 100 seconds.

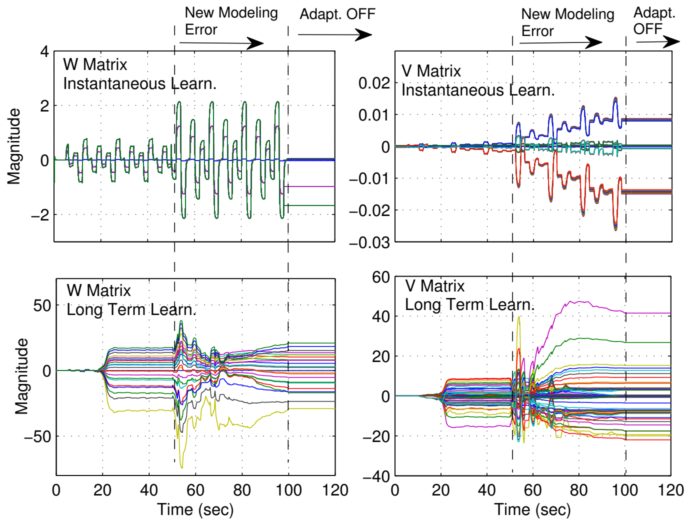

Fig. 7. Comparison of Weight Updates (Scenario-3)

Figure 6 presents simulation results after freezing the network weights. Here, results using long term learning are still accurate predictions. Weight updates of the third scenario is presented in Fig.7.

4. CONCLUSION

In this paper, simulation results are presented to estimate the future response of the load factor with acceptable lead time of a generic utility helicopter model. The steady state value of the load factor is estimated at the time the controls are applied. We compare results using a neural network with traditional update law using instantaneous data only and with the case where recorded data is also used for adaptation. It is observed that the network compensates some of the short comings of the traditional update using instantaneous data only. In particular, the weight convergence is much faster and more probable as the rank-1 limitation is lifted. Simulation results almost always have demonstrated a significant improvement when history data stacks are used in adaptation. While the benefit of using history data stacks is apparent, establishing the right data stack is crucial in accurate predictions. Updating the prediction law to include steady state conditions as well provided better results. The results in this paper are demonstrated for a particular maneuver in the longitudinal channel. Future research could focus on maneuvers where multiple controls and multiple limits are considered.

REFERENCES

[1] Yavrucuk, I., Prasad, J., and Unnikrishnan, S. Envelope Protection for Autonomous Unmanned Aerial Vehicles *Journal of Guidance, Control and Dynamics*, Vol. 32, No. 1,pp. 248–261., 2009.

[2] Yavrucuk, I., Prasad, J. V. R., Calise, A. J., and Unnikrishnan, S. Adaptive Limit Margin Prediction and Limit Avoidance *Proceedings of the American Helicopter Society 58th Annual Forum*,Montreal, Canada, Vol. 2, pp. 856–866, 2002.

[3] Yavrucuk, I., Prasad, J. V. R., and Calise, A. J. Adaptive Limit Detection and Avoidance for Carefree Maneuvering *In Proceedings of the AIAA Atmospheric Flight Mechanics Conference*,Montreal, Canada, 2001.

[4] Chowdhary, G. and Johnson, E. N. Adaptive Neural Network Flight Control Using both Current and Recorded Data *In Proceedings of AIAA Guidance, Navigation and Control Conference*,Hilton Head, South Carolina, 2007.

[5] Chowdhary, G. and Johnson, E. N. Theory and Flight Test Validation of Long Term Adaptive Flight Controller *In Proceedings of AIAA Guidance, Navigation and Control Conference*,Honolulu, Hawaii, 2008.

[6] Yavrucuk, I. Adaptive Limit Detection and Avoidance *PhD Thesis, Georgia Institute of Technology*, 2003.

[7] Johnson, E. N. and Oh, S. M. Adaptive Control Using Combined Online and Background Learning Neural Network *In Proceedings of the 43rd IEEE Conference on Decision and Control*,Atlantis, Paradise Island, Bahamas, 2004.

[8] Gursoy, G. and Yavrucuk, I. Concurrent Learning Enabled Adaptive Limit Detection for Active Pilot Cueing *In Proceedings of AIAA Atmospheric Flight Mechanics Conference*, AIAA-2010-7945, Toronto, Ontario, Aug. 2-5, 2010

[9] Gursoy, G. Neural Network Based Online Estimation of Maneuvering Steady States and Control Limits *M.S. Thesis, Middle East Technical University*, 2010.

[10] Yavrucuk, I., Tarimci, O., Katircioglu, M., Kubali, E., Yilmaz, D. A new Helicopter Simulation and Analyisi Tool: Helidyn+ *In Proceedings of European Rotorcraft Forum*, Paris, 2010.