Infotech@Aerospace
26 - 29 September 2005, Arlington, Virginia

AIAA 2005-7037

Robust and Reconfigurable Flight Control by Neural Networks

Silvia Ferrari and Mark Jensenius

***Abstract*— A linear matrix inequalities (LMIs) framework for developing robust, adaptive nonlinear flight control systems is presented. The controller structure is that of a feedforward sigmoidal neural network that is both adaptive and reconfigurable, since the control law it approximates can be modified by updating its parameters during operation. The neural network controller is designed via LMIs to meet multiple control objectives, that include but are not limited to LQG and H_∞ performance, pole placement, and closed-loop stability, as dictated by the application of interest. Prior knowledge of the linearized equations of motion is utilized in order to guarantee that the neural network controller meets these objectives when the aircraft is operating in its linear-parameter varying (LPV) regime, or steady-state flight envelope. However, should unexpected changes or failures occur during flight, the controller also is capable to reconfigure according to the new and, possibly, nonlinear dynamics. The adaptation consists of a constrained optimization problem that is computationally feasible because it takes place incrementally over time, accounting for the new dynamics only if and when they arise. The LPV performance of the controller is preserved and guaranteed throughout adaptation, by means of a novel constrained-training technique for neural networks.**

I. INTRODUCTION

For many years the aerospace community has been interested in applying adaptive NNs to reconfigurable flight control, with the hope of handling unexpected failures or maneuvers. This problem is very challenging because, while the adaptation must impact a variety of operating conditions, the controller must maintain a high degree of safety at all times. On the other hand, the effects that advanced control systems can have on commercial and general aviation aircraft are far reaching. One goal of the industry is to produce a safer aircraft by eliminating loss of control and adapting to failure. Another goal is to build an aircraft that is easier to fly and cheaper to operate, by controlling its response to command inputs and minimizing control usage.

There is considerable precedent for applying adaptive inverse-dynamic designs [1]-[3] and approximate dynamic programming (ADP) [4]-[11] to reconfigurable control in order to handle unexpected system dynamics and failures. Typically, these designs superimpose an adaptive neural element onto a classical control structure in order to model and handle uncertainties, such as modeling errors or disturbances. Although these approaches have been extremely successful in a variety of applications, they lack the guarantees of closed-loop stability and robustness that characterize classical designs, such as, H_∞ controllers. Also, they often are based on a particular structure of the dynamic equations (e.g., for feedback linearization) that limits the level of performance they can achieve when the real dynamics deviate from this structure.

In this paper, the entire control system is comprised of a sigmoidal neural network, that is a universal function approximator. Hence, given a sufficient number of sigmoidal nonlinearities, it can produce any nonlinear control law on a compact space. The objective is to develop a nonlinear control system with enhanced adaptive and learning capabilities, that can adjust rapidly on line subject to the actual airplane dynamics. Numerical tests in [12] have shown that when the entire controller structure is updated, on-line learning takes place so rapidly as to considerably improve performance during maneuvers that are experienced by the controller for the first time, and that involve nonlinear coupled dynamics, control failures, and parameter variations. Then, the improvement brought about by on-line learning is so remarkable as to prevent loss of control during large-angle maneuvers (e.g., with $-70°$ bank angles), involving nonlinear effects that were not accounted for *a priori* in designing the control system. Although the simulations show that on-line learning can preserve prior control knowledge [12], closed-loop stability and robustness guarantees might still be required for its implementation on a real aircraft.

II. NEURAL NETWORK CONTROL LAW

The control law is formulated in terms of a nonlinear operator $\mathbf{\Phi}$ with repeated sigmoidal functions and it is a function of an augmented state χ defined in terms of control and system state variables,

$$\begin{cases} \dot{x}_K = A_K x_K + B_K y \\ u = u_N := v\mathbf{\Phi}(W\chi) \end{cases}, \tag{1}$$

where $\chi := [x^T \quad x_K^T]^T \in \mathbf{R}^{\nu\times 1}$. If $\chi = x$ the controller is said to be *static*, otherwise it is said to be *dynamic*. The $(1 \times l)$ vector v and the $(l \times \nu)$ matrix W are adjustable parameters to be determined. $\mathbf{\Phi}$ is a diagonal operator with repeated sigmoids,

$$\mathbf{\Phi}(n) := [\sigma(n_1) \cdots \sigma(n_l)]^T, \tag{2}$$

where n_i denotes the i^{th} component of a signal $n \in \mathbf{R}^{l\times 1}$. The sigmoidal function $\sigma(n_i) : \mathbf{R} \rightarrow \mathbf{R}$ is defined as a bounded measurable function on $\mathbf{R}$ for which $\sigma(n_i) \rightarrow 1$ as $n_i \rightarrow \infty$, and $\sigma(n_i) \rightarrow 0$ as $n_i \rightarrow -\infty$. It was shown in [13] and [14] that linear combinations of sigmoidal functions,

This work was supported by National Science Foundation, under grant ECS 0300236.

Silvia Ferrari is with the Faculty of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA sferrari@duke.edu
Mark Jensenius is with the Johns Hopkins University Applied Physics Laboratory, Laurel, MD, USA Mark.Jensenius@jhuapl.edu

such as (1), are universal function approximators, i.e., can approximate any continuous function on compact subsets of $\mathbf{R}^{\nu}$ uniformly well. In this paper, the sigmoidal function takes the form

$$\sigma(n_i) := \frac{e^{n_i} - 1}{e^{n_i} + 1} \quad (3)$$

It can be easily shown that (3) is *monotonically non-decreasing*, and *slope-restricted*, and that it *belongs to the sector* $[\alpha, \beta]$, with $\alpha = 0$ and $\beta = 1/2$ [15]. Thus, the nonlinear control law in (1) can be treated within the Integral Quadratic Constraints (IQCs) framework, for the purpose of analyzing its closed-loop stability and robustness.

III. DESIGN OBJECTIVES AND SPECIFICATIONS

The objective of this research is to develop adaptive nonlinear control systems that are both robust and reconfigurable. A controller whose parameters vary on line during operation is considered to be *adaptive* and can potentially accommodate for a higher degree of uncertainty than a fixed control structure. The controller in (1) consists of a universal function approximator that can represent different control laws by using different values of the parameters v and W. Hence, this controller also can be said to be *restructurable*, since its structure can be changed to account for uncontrollable changes in the dynamic characteristics of the system. Restructurability implies reconfigurability, and can provide satisfactory performance over a much larger range of operating conditions than robustness alone, since the control structure can be altered to address the specific changes as they arise during operation [16]. While robustness to unmodeled dynamics and parameter variations remains a desirable property of any control structure assumed by (1), by adapting according to desired optimization criteria this controller can deal with uncertainties more effectively, e.g., with lesser control usage and better nominal performance.

It is assumed that the actual aircraft dynamics can be captured by a nonlinear differential equation,

$$\dot{x}(t) = f[x(t), p_m(t), u(t)] \quad (4)$$

where $x \in \mathbf{X} \subset \mathbf{R}^{n \times 1}$, $u \in \mathbf{U} \subset \mathbf{R}^{m \times 1}$, and $\{\mathbf{X}, \mathbf{U}\}$ is said to be the *full flight envelope* of the aircraft. The parameters, p_m, and the structure of these equations are partially unknown and/or subject to change. For simplicity, the approach also assumes perfect knowledge of the state, based on error-free measurements of the output y. Nonlinear optimal state estimation and system identification of the actual dynamics in (4), are all important extensions of this approach to design.

Under typical operating conditions the aircraft dynamics can be closely approximated by a class of affine nonlinear systems

$$\begin{cases} \Delta\dot{x} &= F(x)\Delta x + G(x)\Delta u \\ \Delta y &= H_x(x)\Delta x + H_u(x)\Delta u \end{cases} \quad (5)$$

that evolve on a subset of the state space $\mathbf{X}_{LPV} \subset \mathbf{R}^{n \times 1}$, referred to as the linear-parameter-varying or *LPV regime* of the plant. Where, Δ denotes deviations from the nominal value of variable (or equilibrium). The functions F, G, H_x, and H_u can be approximated by a set of matrices $A_j \approx F(\zeta_j)$, $B_j \approx G(\zeta_j)$, $C_j \approx H_x(\zeta_j)$, and $D_j \approx H_u(\zeta_j)$, for p equilibria or *scheduling vectors* $\{\zeta_1, \zeta_2, \ldots, \zeta_p\} \in \mathbf{X}_{LPV}$. Then, for each of these scheduling vectors, there exists a linear-time-invariant (LTI) model,

$$G_j(s) : \begin{cases} \Delta\dot{x} &= A_j\Delta x + B_j\Delta u \\ \Delta y &= C_j\Delta x + D_j\Delta u \end{cases} \quad (6)$$

that can be considered a satisfactory representation of the system dynamics near $\zeta_j \in \mathbf{X}_{LPV}$.

Modern fixed or gain-scheduled controllers are capable of providing satisfactory performance within the LTI regime of the aircraft ($\mathbf{X}_{LPV}$). However, they are not designed to control the aircraft in the event of an emergency maneuver or failure that causes it to abandon $\mathbf{X}_{LPV}$ and explore the remainder of the flight envelope ($\mathbf{X}$). Given that the full flight envelope is $(n + m)$-dimensional, with $n = 8$ and $m = 4$ or 5, and that there is a wide range of complex system dynamics and failures that can be experienced, it would be infeasible to design for all possible conditions *a priori*. The advantage of implementing a neural network controller, such as (1), is that it is accompanied by both batch and incremental learning algorithms for function approximation and refinement, respectively. These algorithms can be used to learn and reconfigure the control law on line, based on the actual aircraft dynamics, to accommodate for flight conditions outside of the LPV regime as they arise. Also, the neural network controller must provide the same performance and safety guarantees as classical controllers when the aircraft is in the LPV regime, which comprises the majority of maneuvers and flight conditions that the aircraft will experience over its lifetime.

In summary, the neural network controller in (1) must meet the following design objectives:

I) Provide the same performance and safety guarantees as an ideal fixed control structure, given *a-priori* knowledge of the aircraft dynamics in the LPV regime, $\mathbf{X}_{LPV}$.

II) Adapt on line to accommodate for aircraft dynamics that arise outside the LPV regime, i.e., in the complement set $\mathbf{X}_{NL} \equiv \bar{\mathbf{X}}_{LPV} = \{x : x \in \mathbf{X}, x \notin \mathbf{X}_{LPV}\}$, by optimizing the same performance and safety metrics used in (I).

III) Continue to satisfy (I) while adapting on line according to (II), to ensure that LPV performance is preserved at all times.

The following sections describe an approach to neural network control design that is aimed at satisfying all three objectives listed above.

IV. DESIGN METHODOLOGY

A. LPV Performance of the Neural Network Control System

At the onset of the design process, the neural network controller (1) is merely a function approximator that can be sized and shaped to represent any nonlinear function on the state space containing χ. Unlike those designs where the neural elements are superimposed onto a conventional control structure, this neural network controller must be trained to provide all of the control functionalities required by a particular application, such as, flight control. Also, these functionalities must be specified through a performance metric that can be optimized during the on-line adaptation. It was shown in [17] that a variety of design objectives, such as, $\mathcal{D}$-stability, H_∞ and H_2 performance, and pole placement, can all be approached simultaneously by formulating a multi-objective synthesis problem that results in sets of Linear Matrix Inequalities (LMIs) to be solved via convex optimization [18]. Then, the solution of this LMI problem, comprising a set of matrices, can be used to compute the state-space controller matrices, A_K, B_K, C_K, and D_K, for a linear dynamic controller,

$$K(s) : \begin{cases} \dot{x}_K = A_K x_K + B_K y \\ u = u_K := C_K x_K + D_K y \end{cases} \tag{7}$$

that achieves all of the desired design objectives for the family of affine LPV systems, such as (5).

Hence, the ideal LPV control performance can be specified through $K(s)$ over $\mathbf{X}_{LPV}$. It is shown in [15] that a feedback controller with $l = p$ nonlinearities in the form of (1) exists such that the ideal performance of $K(s)$ is met at p equilibria in $\mathbf{X}_{LPV}$, i.e.:

$$\begin{cases} \frac{\partial u_N}{\partial \dot{\chi}}(\zeta_j) = \frac{\partial u_K}{\partial \dot{\chi}}(\zeta_j) \\ u_N(\zeta_j) = u_K(\zeta_j) \end{cases} \quad j = 1, \ldots, p. \tag{8}$$

The neural network parameters W and v that meet the above specifications must satisfy a set of linear systems involving the controller and the LPV plant state-space matrices, as shown in [15]. Similarly to gain-scheduling, the local control performance at the p equilibria is automatically generalized to intermediate conditions thanks to the interpolation capabilities of sigmoidal neural networks [19]. However, due to the sigmoidal nonlinearities, Φ, the stability and robustness guarantees of $K(s)$ are not necessarily inherited by the neural network controller. Therefore, an additional set of LMIs is imposed on the parameters W and v such that the closed-loop LPV system comprised of the neural network controller and the LTI model (6),

$$\begin{cases} \Delta\dot{x}_{cl}(t) = A\Delta x_{cl}(t) + Bv\Phi[Wy_a(t)] \\ y_a(t) = C\Delta x_{cl}(t) \end{cases} \tag{9}$$

can be shown to be both stable and robust.

In fact, the neural-network controlled system in (9) is in the same form as the basic feedback interconnection suitable for IQC analysis, with Φ as a bounded, causal diagonal operator. The repeated sigmoidal nonlinearity in (2)-(3) is monotonically non-decreasing, slope-restricted, and belonging to the sector $[0, 1/2]$. Therefore, the IQCs developed in [20] can be applied to (9) for the purpose of stability and robustness analysis, as shown in [15]. The result is that the stability of the neural-network controlled system is guaranteed if there exists constant symmetric matrices $M, P \in \mathbf{R}^{\nu\times\nu}$, such that the following LMIs are satisfied:

$$\begin{array}{l} M_{ii} \geq \sum\limits_{j=1, j\neq i}^{\nu} |M_{ij}| \quad i = 1, \ldots, \nu \\ \begin{bmatrix} PA + A^T P & PB_N + C_N^T M \\ (PB_N)^T + MC_N & -2M \end{bmatrix} < 0 \end{array} \tag{10}$$

Where, ν is the dimension of the augmented state vector, χ, A is Hurwitz, $B_N = Bv$, and $C_N = WC$. Since these LMIs contain the adjustable parameters W and v, an adequate neural network controller in the form of (1), and meets the ideal LPV performance as stated in (I), can be obtained by finding the values of W and v that satisfy (8) and (10), simultaneously. Therefore, this design step consists of solving the corresponding LMIs for the matrices M, P, W and v [15]. The closed-loop system robustness can be similarly analyzed.

B. On-line Adaptation of the Neural Network Control System

The neural network controller obtained in the previous section, with parameters held fixed, provides satisfactory performance over the LPV envelope of the aircraft. Hence, it can be considered equivalent to a conventional control system that is designed to achieve multiple design objectives, such as, $\mathcal{D}$-stability, H_∞ and H_2 performance, and pole placement, for a plant that operates in a LPV regime. Depending on the application, the ideal control system may be static or dynamic, fixed or gain-scheduled. In every case, an equivalent neural network controller comprised solely of a linear superposition of sigmoidal nonlinearities can be derived using the technique described in the previous section. The advantage of the neural network controller over conventional designs, such as, linear quadratic regulators (LQR), H_∞, and μ-synthesis control, is that it is both nonlinear and restructurable, thanks to the approximation and learning capabilities of sigmoidal neural networks.

Several *training* algorithms, including resilient backpropagation (RPROP) and Levenberg-Marquardt, to name a few, have been developed for these networks. The error between the network output and an ideal target is propagated backwards through the equations representing the hidden (sigmoidal) layer, such that an unknown function can be learned from sampled input/output data or *examples*. In on-line learning, the data become available one pair at a time, as the actual aircraft state is observed from on-board measurements. Therefore, incremental training techniques are required to assimilate the data and improve the neural network approximation of the control law over time. Some of the key issues that must be resolved in order for the adaptation to be successful are: (1) the initial surface approximated by the neural network at the on-set of the adaptation; (2) the output target to be provided based on the observations of the state (the input) in order to improve the neural-network control performance through incremental training; and, (3) the incremental training algorithm.

In this approach to design, the initial surface approximated by the neural network at the on-set of the adaptation is obtained from $K(s)$, according to the procedure described in Section IV-A. Since this can be considered as the ideal performance over the LPV envelope, the adaptation takes effect only when the aircraft leaves the LPV regime and enters $\mathbf{X}_{NL}$. The target performance in this non-LPV regime is determined from the actual state of the aircraft by means of approximate dynamic programming (ADP). Dynamic programming relies on the *principle of optimality* [21] to find a control law,

$$u(t) = c[x(t)] \tag{11}$$

that optimizes a desired *cost function* subject to the dynamic equations (4). The cost function measures performance with respect to design objectives that are expressed by an integral function of the state and control input,

$$J = \int_{t_0}^{t_f} \mathcal{L}[x(\tau), u(\tau)]d\tau \tag{12}$$

The control law that minimizes (12) subject to (4), denoted by c^*, is said to be *optimal*, and it is to be determined. Since the system dynamics are nonlinear and, possibly, unknown, the optimal control law cannot be computed *a priori*, before the changes and uncertainties in the aircraft dynamics are revealed. However, if these changes are reflected in the aircraft state at the time they arise, the optimal control law can be approximated on line, by learning implicitly from a so-called *value function* or *cost-to-go*.

At any moment in time, $t_0 \leq t \leq t_f$, a value function expressing the cost to be accumulated over all future times ($\tau \geq t$) can be defined based on (12)

$$V[x(t), c] = \int_{t}^{t_f} \mathcal{L}[x(\tau), u(\tau)]d\tau \tag{13}$$

By the principle of optimality, the minimization of J can be imbedded in the minimization of V, with the advantage that at any time t, V can be minimized independently of the control history prior to t. In [22], Howard showed that if c and V are modified by a Policy-Improvement Routine and a Value-Determination Operation, respectively, they converge to the optimal control law, c^*, and value function, V^*. Although in the case of full-envelope flight control one does not expect to continue the optimization until these optimal counterparts are found, this algorithm can be used very effectively to optimize the neural network controller, such that its sub-optimal performance in $\mathbf{X}_{NL}$ is improved over time.

1) Policy-Improvement Routine: Given a value function $V(\cdot, c_\ell)$ corresponding to a sub-optimal control law c_ℓ, an improved control law $c_{\ell+1}$ can be obtained as follows,

$$\begin{aligned} c_{\ell+1}[x(t)] \quad = \quad & \arg\min_{u(t)} \{\mathcal{L}[x(t), u(t)] \\ & + V[f[x(t), p_m(t), u(t)], c_\ell]\} \end{aligned} \tag{14}$$

such that $V[x(t), c_{\ell+1}] \leq V[x(t), c_\ell]$, for $\forall x(t) \in \mathbf{X}$. Furthermore, the sequence of functions $\mathcal{C} = \{c_\ell | \ell = 0, 1, 2, \ldots\}$ converges to the optimal control law, c^*.

2) Value-Determination Operation: Given a control law c, the value function can be updated according to the rule,

$$V_{\ell+1}[x(t), c] = \mathcal{L}[x(t), u(t)] + V_\ell[f[x(t), p_m(t), u(t)], c] \tag{15}$$

such that the sequence of functions $\mathcal{V} = \{V_\ell | \ell = 0, 1, 2, \ldots\}$ converges to V^*.

The two sequences of functions generated by the above algorithm are used to train two neural network function approximators, referred to as *actor* and *critic*, that over time converge to the optimal functions. The neural network controller in (1) (the actor) approximates the sequence of control laws $\mathcal{C}$ by updating the parameters W and v at every cycle of the Policy-Improvement Routine,

$$\begin{cases} \dot{x}_K = A_K x_K + B_K y \\ u_N = v_\ell \mathbf{\Phi}(W_\ell \chi) = \tilde{c}[x(t), \upsilon_\ell] \approx c_\ell[x(t)] \end{cases} \tag{16}$$

where, $\chi = [x^T \quad x_K^T]^T$ and, for convenience, W_ℓ and v_ℓ are arranged in a single vector of adjustable actor parameters υ_ℓ. The values of the parameters W and v determined in Section IV-A are assigned to W_0 and v_0, respectively, assuming that the aircraft begins its flight in the LPV region where these parameters provide optimal performance. Subsequently, they are updated by an incremental training algorithm that adapts the neural network controller to reflect the changes in the control functional c_ℓ.

The parametric structure referred to as the critic approximates the sequence of value functions $\mathcal{V}$,

$$\tilde{V}[x(t), \omega_\ell] \approx V_\ell[x(t), \tilde{c}] \tag{17}$$

by updating the vector of parameters ω_ℓ at every cycle. Research in [12] showed that a sigmoidal neural network is appropriate for approximating the value function as well as the control law. This research also showed that modified-RPROP allows for rapid and effective incremental training, provided few training epochs are used during every cycle of the algorithm in order to avoid overfitting. More recently, this RPROP algorithm was further modified to incorporate algebraic constraints, according to the approach outlined in the following section (and detailed in [23]).

C. Robust Adaptation in non-LPV Regimes

One main difficulty of on-line or incremental learning techniques is that the neural networks have tendency to *forget* the information assimilated during prior training sessions or epochs. In regard to the neural network control system presented in this paper, this would imply that the control knowledge imparted for the LPV regime might be altered or lost during the adaptation in $\mathbf{X}_{NL}$, due to the global support exhibited by sigmoidal neural networks. Assuming the optimal control law over $\mathbf{X}$ is smooth and well-behaved, global support is a desirable characteristic for the neural network controller to have. Therefore, an on-line training technique that modifies $\tilde{c}[x(t), \upsilon_\ell]$ for $x \in \mathbf{X}_{NL}$, but that preserves its shape and properties for $x \in \mathbf{X}_{LPV}$ is devised.

The control design objectives to be optimized on line are defined through a set of q cost indices $\{J_1, \ldots, J_q\}$, where

$$J_i = \lim_{t_f \to \infty} \left\{ \frac{1}{t_f} \int_{t_0}^{t_f} \xi^T(\tau)\xi(\tau)d\tau \right\} \quad (18)$$

For example, LQG performance and pole placement each can be addressed by a cost index with,

$$\xi_i(t) = \begin{bmatrix} Q_i^{1/2} & 0 \\ 0 & R_i^{1/2} \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \quad (19)$$

where, Q_i and R_i are weighting matrices that can be determined by classical multivariable control techniques (see [24] for a comprehensive review). Also, a stability index can be obtained by observing that the quadratic function $V_s \equiv \Delta x_{cl}^T P \Delta x_{cl}$, with P being the matrix obtained from the IQC problem in (10), is a Lyapunov function for (9).

Depending on the control objective, each cost index must be minimized or constrained below a prescribed value γ_i in order to optimize the neural network control system performance through the adaptation in $\mathbf{X}_{NL}$. Suppose r cost indices are constrained, and $(q - r)$ indices are combined into a single cost function J_0 that takes the form of (12) and, similarly, must be minimized with respect to the control law $c[x(t)]$. Then, the minimization of J_0 can be viewed as a nonlinear optimal control problem that can be approached by the ADP technique described in Section IV-B, with added inequality constraints adjoint to the policy-improvement routine (14):

$$\text{minimize} \quad \xi_0^T(\cdot)\xi_0(\cdot) + \tilde{V}[f(\cdot), \omega_\ell] \quad (20)$$

$$\text{subject to} \quad J_i \leq \gamma_i, \quad i = 1, \ldots, r \quad (21)$$

$$F_0 + \sum_j p_j F_j, \quad j = 1, \ldots, N \quad (22)$$

During each adaptation cycle indexed by ℓ, all functions in (20) are evaluated at the present state and control values, $x(t)$ and $u_N(t) = \tilde{c}[x(t), \upsilon_\ell]$. The LMIs in (22) represent the IQC requirements in (10) and the linear equations corresponding to (8). Therefore, the above constrained minimization problem is to be solved with respect to the neural network parameters, and, possibly, for the matrices P and M in (22).

The value-determination operation is carried out by updating the critic parameters, ω_ℓ, according to the following equation, obtained from (15) and (18):

$$\tilde{V}[x(t), \omega_{\ell+1}] = \xi_0^T(\cdot)\xi_0(\cdot) + \tilde{V}[f(\cdot), \omega_\ell] \quad (23)$$

By cycling between the above operations, the actor and critic progressively are improved, and the neural network control system performance in $\mathbf{X}_{NL}$ is optimized over time. The constrained-training algorithm developed for updating the sigmoidal neural networks is described in [23].

V. CONCLUSIONS

An LMI framework for designing reconfigurable controllers with sigmoidal nonlinearities, also referred to as artificial neural networks, is presented. The approach is directed to addressing the need for flight controllers that provide strong safety and performance guarantees under ”typical” (LPV) flight conditions, and that can adapt rapidly in the event of changes or failures that cannot be accounted for *a priori*. In order to provide the learning capabilities and flexibility required to accommodate for a variety of situations and maneuvers, while retaining LPV performance, the control system is comprised of sigmoidal neural network where all of the parameters can be updated as soon as a new observation of the state becomes available. A constrained-training technique is developed to prevent forgetting prior LPV control knowledge during incremental on-line training outside of the LPV envelope.

APPENDIX I

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation, under grant ECS 0300236.

REFERENCES

[1] A. J. Calise, “Neural Networks in Nonlinear Aircraft Flight Control,” *IEEE Aerospace and Electronics Systems Magazine*, Vol. 11, No. 7, pp. 5-10, 1996.

[2] B. S. Kim and A. J. Calise, “Nonlinear Flight Control Using Neural Networks,” *J. Guidance, Control, and Dynamics*, Vol. 20, No. 1, pp. 26-33, 1997.

[3] A. J. Calise and R. T. Rysdyk, “Nonlinear Adaptive Flight Control Using Neural Networks,” *IEEE Control Systems Magazine*, pp. 14-25, December 1998.

[4] *Handbook of Intelligent Contro*l, D. A. White and D. A. Sofge, Eds., pp. 65-86, Van Nostrand Reinhold, New York, NY, 1992.

[5] S. N. Balakrishnan and V. Biega, “Adaptive-Critic-Based Neural Networks for Aircraft Optimal Control,” *J. Guidance, Control, and Dynamics*, Vol. 19, No. 4, pp. 893-898, 1996.

[6] K. Krishnakumar and J. Neidhoefer, “Immunized Adaptive Critics for Level 2 Intelligent Control,” *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, FL, pp. 856-861, 1997.

[7] G. G. Lendaris, C. Paintz, and T. Shannon, “More on Training Strategies for Critic and Action Neural Nets in Dual Heuristic Programming Method,” *Proc. of IEEE-SMC ’97*, Orlando, October, 1997.

[8] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, “Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator,” *IEEE Transactions on Neural Networks*, Vol. 13, No. 3, pp. 764-773, 2002.

[9] A. Barto, R. Sutton, and C. Anderson, “Neuronlike Elements that Can Solve Difficult Learning Control Problems,” *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 3, No. 5, pp. 834-846, 1983.

[10] P. J. Werbos, “Advanced Forecasting Methods for Global Crisis Warning and Models of Intelligence,” *General Systems Yearbook*, 1997.

[11] Z. Huang and S. N. Balakrishnan, “Robust Adaptive Critic Based Neurocontrollers for Systems with Input Uncertainties,” *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, Vol. 3, pp. 67-72, 2000.

[12] S. Ferrari and R.F. Stengel, “On-line Adaptive Critic Flight Control,” *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, 2004, pp. 777-786.

[13] G. Cybenko, ”Approximation by Superposition of a Sigmoidal Function”, *Math. Contr., Signals, Syst.*, vol. 2, 1989, pp. 359-366.

[14] K. Hornik, M. Stichcombe, and H. White, ”Multilayer Feedforward Networks are Universal Approximators,” *Neural Networks*, vol. 2, no. 5, 1989, pp. 359-366.

[15] S. Ferrari, “Stable Neural Network Control Design by Linear Matrix Inequalities,” under review.

[16] R. F. Stengel and C. Huang, “Restructurable control using proportional-integral implicit model following,” *Journal of Guidance, Control, and Dynamics*, Vol.13, No.2, pp. 303-309, 1990.

[17] M. Chilali, P. Ghainet, and P. Apkarian, ”Robust Pole Placement in LMI Regions,” *IEEE Trans. on Automatic Control*, vol. 44, no. 12, 1999, pp. 2257-2270.

[18] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox, for Use with MATLAB*, The Mathworks, 1995.
[19] S. Ferrari and R.F. Stengel, "Classical/Neural Synthesis of Nonlinear Control Systems," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 3, 2002, pp. 442-448.
[20] F. J. D'Amato, M. A. Rotea, A. V. Megretski, U. T. Jönsson, "New Results for Analysis of Systems with Repeated Nonlinearities," *Automatica*, vol. 37, 2001, pp. 739-747.
[21] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.
[22] R. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA, 1960.
[23] M. Jensenius, *Constrained Learning in Neural Control Systems*, M.S. Thesis, Duke University, 2005.
[24] R. F. Stengel, *Optimal Control and Estimation*, Dover Publications, Inc., New York, 1994.