IN-63

1687

P9

Towards Practical Control Design Using Neural Computation

Terry Troudet
Sverdrup Technology, Inc.
Lewis Research Center
Brook Park, Ohio

Sanjay Garg
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

Duane Mattern
Sverdrup Technology, Inc.
Lewis Research Center
Brook Park, Ohio

Walter Merrill
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

NASA

TOWARDS PRACTICAL CONTROL DESIGN USING NEURAL COMPUTATION.

T. TROUDET,[2] S. GARG,[1] D. MATTERN,[2] W. MERRILL[1]

Abstract.

The objective of this paper is to develop neural network based control design techniques which address the issue of performance/control effort trade-off. Additionally, the control design needs to address the important issue of achieving adequate performance in the presence of actuator nonlinearities such as position and rate limits. These issues are discussed using the example of aircraft flight control. Given a set of pilot input commands, a feedforward net is trained to control the vehicle within the constraints imposed by the actuators. This is achieved by minimizing an objective function which is a weighted sum of the tracking errors, control input rates and control input deflections. A trade-off between tracking performance and control smoothness is obtained by varying, adaptively, the weights of the objective function. The neurocontroller performance is evaluated in the presence of actuator dynamics using a simulation of the vehicle. Appropriate selection of the different weights in the objective function resulted in good tracking of the pilot commands and smooth neurocontrol. An extension of the neurocontroller design approach is proposed to enhance its practicality.

I. Introduction. This paper addresses architecture and training issues in using neural computation towards practical control design. Such issues are discussed on a neural net architecture designed to control a simplified model of airframe/propulsion-system. Although many papers on application of neural networks to control design have appeared in the recent literature, e.g. Refs.[1-3], most of the applications considered are either for robotic systems or for control problems that are mainly of academic interest such as the inverted pendulum problem. The objective of this paper is to investigate the applicability of neural networks to the control of aerospace vehicles.

In the current literature on neurocontrol design for tracking target trajectories, a great emphasis is placed on minimizing tracking error without due emphasis on physical constraints on control inputs. For instance, in Ref.[4], a neurocontrol design is presented for a cart-pole system which nearly "exactly" tracks the reference cart position; however no information is provided on the control effort (force applied to cart) required to achieve the indicated performance. In general, such "exact" tracking of reference commands can only be achieved by using very large control input and control rates. A realistic control design problem consists of achieving a practical performance/control trade-off, i.e. "best" possible performance within the physical constraints of the actuators. Clearly then a better understanding needs to be developed on how to achieve this desired performance/control trade-off within the framework of neurocontrol design.

The approach taken in this paper is that of learning the neurocontrol by minimizing an objective function which is a weighted sum of tracking errors and control input commands and rates. The notion of weighting the control inputs in the objective function has previously been suggested by other researchers, see for example Refs.[2-3]. The process of adapting the weights of the objective function (*not to be confused with the synaptic weights of the neural net*), towards maximizing tracking performance within the physical limitation of the actuators while providing control smoothness, is described. Knowledge gained about the effect of each component of the objective function leads to extend the architecture towards more flexibility as needed for practical control design.

II. Vehicle Model. The vehicle model is a linear system of the form:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}_a, \quad \bar{z} = C\bar{x}; \tag{1}$$

where the state vector $\bar{x}$, defined in Ref.[5], consists of 5 airframe state variables (*aircraft body axis forward and vertical velocities, aircraft pitch rate, pitch angle and altitude*), and 4 propulsion system state variables (*engine fan speed, core compressor speed, engine mixing plane pressure, and engine high pressure turbine blade temperature*).

---

[1]S. Garg and W. Merrill, NASA Lewis Research Center.
[2]T. Troudet and D. Mattern, Sverdrup Technology, Inc., 2001 Aerospace Parkway, Brook Park, Ohio 44142.

The control input vector, $\bar{u}_a$, of interest is

$$\bar{u}_a = [WF,\ \delta TV]^T;\qquad(2)$$

where WF is the engine main burner fuel flow rate (#/hr), and δTV is the aft nozzle thrust vectoring angle (deg). The vehicle outputs to be controlled are

$$\bar{z} = [V,\ Q]^T\ ,\qquad(3)$$

where V is the aircraft velocity (ft/sec), and Q the pitch rate (deg/s). The system matrices A, B, and C are available in Ref.[5]. The open-loop vehicle system defined by (1) to (3) is unstable in pitch response, and is strongly coupled in the response of the controlled outputs $\bar{z}$ to control inputs $\bar{u}_a$.

The control design objective is to design a control system to provide decoupled command tracking of velocity and pitch rate from pilot control inputs so as to make the overall closed-loop vehicle system acceptable for pilot controlled flight.

For a given input command $\bar{z}_{SEL} = [V_{SEL}, Q_{SEL}]^T$ selected by the pilot, the commanded variables $\bar{z}_c = [V_c, Q_c]^T$ that are to be tracked by the aircraft are solutions of

$$\dot{\bar{x}}_m = A_m \bar{x}_m + B_m \bar{z}_{SEL},\quad \bar{z}_c = C_m \bar{x}_m;\qquad(4)$$

where the matrices A_m, B_m and C_m represent the desired dynamics of the plant for a pilot selected input command. The numerical values for A_m, B_m and C_m for this example are as listed in Ref.[5]. These pre-filtering matrices are based on military specifications for level I ("good") flying qualities for piloted aircraft of the type being considered here (see Ref.[6] for example).

The dynamics of the fuel flow actuator are approximated by a second order system with transfer function

$$G_{WF}(s) = \frac{10}{10+s} \times \frac{50}{50+s};\qquad(5)$$

and with a maximum fuel flow rate $|WF_{max}| = 10,000\#/hr$ (perturbation around the nominal value), and a rate limit $|\dot{WF}_{max}| = 20,000\#/hr/s$. The thrust vectoring actuator is approximated by a first order system with transfer function

$$G_{\delta TV}(s) = \frac{15}{15+s};\qquad(6)$$

with a maximum thrust vector angle $|\delta TV_{max}| = 10deg$ (perturbation around the nominal value), and a rate limit $|\dot{\delta TV}_{max}| = 20deg/s$.

III. Training Architecture. The training architecture is represented in Fig.1. For each pilot selected trajectory $\bar{z}_{SEL}(t)$, a commanded trajectory $\bar{z}_c(t)$ is generated from (4). Prior to training, the commanded variables $\bar{z}_c(t)$ are discretized and scaled into $\bar{z}_c^s(t_k) = [V_c(t_k)/V_c{}^0, Q_c(t_k)/Q_c{}^0]^T$ where $V_c{}^0$ and $Q_c{}^0$ are of the order of magnitude of the maximum values of $V_c(t)$ and $Q_c(t)$ respectively. If $\bar{z}_c^s(t_{k+1})$ is the commanded scaled output of the desired dynamics at time t_{k+1}, the actual scaled output of the aircraft controlled by the neural network is obtained as follows.

As shown in Fig.1, the two control inputs ($\bar{u}$) are calculated by a two hidden-layer feedforward net that has eight input units (or four pairs of input units associated to the Q and V variables), and two neurons in the output layer. These pairs consist of the scaled output vector $\bar{z}^s(t_k)$; the tracking error $\bar{e}_z(t_k)$ between the scaled vehicle output vector $\bar{z}^s(t_k)$ and its desired scaled value at time t_{k+1} (i.e. $\bar{e}_z(t_k) = \bar{z}_c^s(t_{k+1}) - \bar{z}^s(t_k)$); the discrete time-derivative of the tracking error, $\dot{\bar{e}}_z(t_k)$; and the time-average of the tracking error, $1/t_k \int_0^{t_k} \bar{e}_z(t)dt$. The motivation behind using the combination of $\bar{z}^s(t_k)$ and $\bar{e}_z(t_k)$ as inputs to the neurocontroller, instead of $\bar{z}^s(t_k)$ and $\bar{z}_c^s(t_{k+1})$, is to allow the network to reconstruct the command without direct feedforward of the command (which would lead to a higher bandwith controller). The role of the error rates input $\dot{\bar{e}}_z(t_k)$ is to provide the net with lead information, and the integral error feedback $1/t_k \int_0^{t_k} \bar{e}_z(t)dt$ is to provide zero steady state tracking error for step commanded inputs. In Fig.1, Δ denotes a time-delay of length δt. Each neuron has the standard activation function:

$$y = tanh(x);\qquad(7)$$

2

which limits its output y to the interval $[-1, +1]$ for any input signal x. For a given set of weights of the neural network, the two output neurons yield the normalized control input vector

$$\bar{u}(t_k) = [\frac{WF}{WF_{max}}, \frac{\delta TV}{\delta TV_{max}}]^T .$$ (8)

The fuel flow rate and thrust vectoring angle calculated by the neural net are then applied to the vehicle model during a small time-interval $\delta t = t_{k+1} - t_k$, and change the state vector from $\bar{x}(t_k)$ to $\bar{x}(t_{k+1})$. To minimize the complexity involved in this preliminary analysis, the actuator dynamics were not explicitly included in the training architecture of Fig.1. However, as will be seen in the next section, their bandwidth limiting effect can be accounted for by training the net to minimize an objective function that includes tracking errors, control input deflections and control input rates

$$J(t_{k+1}) = \frac{1}{2}(\bar{\epsilon}_z^T(t_{k+1}).\bar{\bar{\rho}}.\bar{\epsilon}_z(t_{k+1}) \ + \ \bar{u}^T(t_k).\bar{\bar{\lambda}}.\bar{u}(t_k) \ + \ \dot{\bar{u}}^T(t_k).\bar{\bar{\mu}}.\dot{\bar{u}}(t_k) \ )$$ (9)

where $\bar{\epsilon}_z(t_{k+1})$ is the error between the scaled commanded vector $\bar{z}_c^s(t_{k+1})$ and the scaled vehicle output $\bar{z}^s(t_{k+1})$. The matrices $\bar{\bar{\rho}}$, $\bar{\bar{\lambda}}$ and $\bar{\bar{\mu}}$ are 2x2 diagonal matrices whose coefficients can be adapted so as to modify the characteristics of the neurocontroller, and allow it to accomodate for the physical limitations of the actuators when operating in closed-loop. It is proposed to use the backpropagation algorithm [7] to find the weights of the neural net that minimize the objective function (9) over the set of pilot input commands. In order to backpropagate (9), a neural net emulator (perceptron) was used in place of the vehicle model in the training architecture of Fig.1.

The commanded trajectories used to train the net were generated as follows. The pilot selected pitch rate was a doublet centered at a time t_c between 2.5s and 5s, with the characteristics: $Q_{SEL}(t) = Q_0$ for $t \le t_c$; $Q_{SEL}(t) = -Q_0$ for $2t_c \ge t > t_c$; $Q_{SEL}(t) = 0$ for $t > 2t_c$. Note that Q_{SEL} corresponds to pilot longitudinal stick deflection with units in inches. The pilot selected airframe velocity was a step function characterized by $V_{SEL}(t) = 0$ for $t \le 0$ and $V_{SEL}(t) = V_0$ for $t > 0$. The maximum intensities $|Q_0|$ and $|V_0|$ of the selected input commands were bounded by $Q_{max} = 0.5in$ and $V_{max} = 20ft/s$. This maximum value of Q_{SEL} corresponds to a maximum pitch rate command of about 3 deg/sec. Random sets of input trajectories were generated from uniform distributions of Q_0, t_c and V_0 over $[-Q_{max}, Q_{max}]$, $[2.5s, 5s]$ and $[-V_{max}, V_{max}]$ respectively. The commanded variables $Q_c(t)$ and $V_c(t)$ were filtered from $Q_{SEL}(t)$ and $V_{SEL}(t)$ over a period of 12s with a time-step $\delta t = 0.01s$. With these characteristics of the selected input commands, the scaling factors of the commanded variables $\bar{z}_c^s(t) = [V^s{}_c(t), Q^s{}_c(t)]^T$ are $V_c{}^0 = 20$ ft/sec and $Q_c{}^0 = 3$ deg/sec.

IV. Neurocontrol Performance. The evaluation architecture of the neurocontroller in closed-loop is shown in Figure 2. The neurocontroller was tested on *step* pitch rate input commands, different from the *doublets* used in training. The input commands chosen to illustrate the neurocontrol performance were defined by the step pitch rate command $Q_{SEL}(t) = 0.5$ inches (unit of the pilot shift stick) for $t \le 3sec$, $Q_{SEL}(t) = 0$ for $t > 3sec$; applied simultaneously with one of the following classes of step velocity commands: $V_{SEL}(t > 0) = -20ft/sec$ (case 1); $V_{SEL}(t > 0) = 20ft/sec$ (case 2); $V_{SEL}(t \ge 0) = 0ft/sec$ (case 3). Neurocontrol was applied to the vehicle-actuator system over a period of 12sec with a time-step $\delta \tau = 0.001sec$. For brevity, only the results of the evaluation with case 1 commands are presented in this paper.

When training with $\bar{\bar{\lambda}} = \bar{\bar{\mu}} = \bar{\bar{0}}$, the neural net learns only to minimize the tracking error $\bar{\epsilon}_z(t_k)$ without giving any consideration to the cost associated with high control requirements and high control rates. Training was performed over a set of 2000 commanded trajectories with a network configuration of 15 neurons in the first hidden layer, and 20 neurons in the second hidden layer. The synaptic weights were updated at every time $t_k (= k\delta t = k \times 0.01sec)$ after backpropagating $J(t_k)$ through the network. This was done once for each trajectory of the training set with a steepest-descent coefficient $\alpha = 0.03$. For $\bar{\bar{\rho}} = diag[\rho_V, \rho_Q] = diag[2000, 20]$, the net learns to track the commanded outputs nearly perfectly in the absence of actuators (see Fig.3a), but with extremely high control input and control rate requirements (see Fig.3b). When the actuator dynamics are included in the closed-loop evaluation, the tracking performance deteriorates significantly, as shown in Fig.4, with highly oscillatory pitch rate response and a limit cycle behavior in velocity/fuel-flow response.

The limit cycle in the velocity response is due to the large commanded fuel flow from the neurocontroller. In order to constrain control commands during training, non zero values of the control weights $\bar{\bar{\lambda}}$ were chosen in the objective function to be minimized (see (9)). A study of the tracking-performance/control-requirement

3

trade-off was conducted by training the network with $\bar{\bar{\lambda}}$ of the form $\bar{\bar{\lambda}} = \lambda\bar{\bar{I}}$, where $\bar{\bar{I}}$ is the 2x2 identity matrix and λ, a scalar, was varied from 0.01 to 0.1, with the same training characteristics and the same matrix elements of $\bar{\bar{\rho}}$ as before. The results from this trade-off study are shown in Fig.5 in terms of mean-square velocity tracking error, e_V, plotted against the mean-square fuel flow rate for the trained network tracking the commands of case 1, with actuators included in the evaluation. As seen from Fig.5, weighting the control commands in the cost error function (9) by the small value $\lambda = 0.01$ results in a significant decrease in commanded control activity while simultaneously improving tracking performance. Increasing λ beyond 0.01 does not result in any further significant decrease in control activity while the tracking performance starts to degrade noticeably.

The results of the closed-loop neurocontrol with $\lambda = 0.01$ are shown in Fig.6. Clearly, constraining the control activity during training results in a neurocontroller with stable and much improved velocity tracking capability, with zero steady state errors for both pitch rate and velocity, and low fuel flow requirements. However, the pitch rate response is still oscillatory, and it is evident from the the thrust vectoring (δTV) requirement plot that the command generated by the neurocontroller is riding the actuator rate limit. Such ringing of the actuator would lead to premature actuator wear and would also result in undesirable aircraft response. A smooth evolution of the control inputs is therefore a practical requirement of control design.

In order to enhance the *control smoothness* within the architecture of Fig.1, the control input rates were constrained during training by choosing a non-zero $\bar{\bar{\mu}}$ in the cost error function to be minimized during training. A performance/control-rate trade-off similar to that for the control activity constraint was also performed for the control rate constraint. Training was performed in two phases for a network configuration of 15 neurons in the first hidden layer, and 10 neurons in the second hidden layer. In the gross-tuning phase of the training, a set of 100 commanded trajectories was randomly generated, and the synaptic weights were updated following a moving-window scheme: at every time t_k, the weights were incremented after backpropagating through the network the time- integral of the objective function calculated over one second, i.e. $\sum_{i=1}^{100} J(t_{k+i})$. This was done once for each trajectory of the training set with a steepest-descent coefficient $\alpha = 0.0001$. In the fine-tuning phase of the training, the objective function was sampled every 0.1 sec. over a single commanded trajectory during a 12 sec. period. The changes in the synaptic weights were calculated for each sample, but the weights were updated only after summing the calculated changes over all the samples of the training trajectory. This procedure was repeated 50 times with $\alpha = 0.0002$. The results with $\bar{\bar{\mu}} = 0.01\bar{\bar{I}}$ and $\bar{\bar{\rho}}$, $\bar{\bar{\lambda}}$ as defined above, are shown in Fig.7. The pitch rate response no longer exhibits oscillatory behavior, and the deviation from the ideal response is small. The velocity command tracking is just as good as for the previously discussed case (Fig.6), while the control requirements WF and δTV are both much more smooth due to the control rate weighting in the performance index.

V. Conclusion. Insight was provided into issues related to practical control design using neural networks. An aircraft control design example with two control inputs and two controlled outputs was used to illustrate these issues. The control design problem was set up as that of following the trajectories generated from a model of the desired vehicle response dynamics from pilot command inputs. The training of the network was done without any actuators, to simplify the computation, while resulting neurocontrollers were evaluated with the *non-linear* actuator dynamics including position and rate limits. In most previous studies of neurocontrol design for such tracking problems, network training was done with emphasis only on minimizing the tracking error. It was shown in this paper that the neurocontrol learned with such a training results in very high control and control rate commands. Although nearly perfect tracking of pilot commands was achieved with such a control law when the control input was unconstrained, the closed-loop system was found to be unstable when the actuator dynamics and position and rate limits were included in the evaluation. Performance/control effort trade-off is an important consideration in practical control design, and an approach for achieving this trade-off within the framework of neural network based control was suggested and investigated in this paper. In this approach, neurocontrol is learned by minimizing an objective function which is a weighted sum of tracking errors, control inputs and control input rates. Appropriate selection of the different weights in the objective function resulted in good tracking of the pilot commands and *smooth* neurocontrol. The possibility of achieving further improvement in performance by including the actuator constraints during training is currently being investigated.

4

References.

[1] *Special Section on Neural Networks for Control Systems*, IEEE Control Systems Magazine, Vol.9, No.3, pp. 25-59, April 1989.

[2] *Special Issue on Neural Networks in Control Systems*, IEEE Control Systems Magazine, Vol.10, No.3, April 1990.

[3] M. I. Jordan: *Generic Constraints On Underspecified Trajectories*, Int. Joint Conf. On Neural Networks, Vol.I, p.217, Washington D.C., June 1989.

[4] W. Goldenthal and J. Farrell: *Application of Neural Networks to Automatic Control*, AIAA Paper 90-3438. Guidance, Navigation and Control Conference, Portland, OR, August 1990.

[5] S. Garg, D. Mattern and R. Bullard: *Integrated Flight/Propulsion Control System Design Based on a Centralized Approach*, Guidance, Navigation and Control Conference, AIAA, Boston, Aug. 1989.

[6] *"Military Specification - Flying Qualities of Piloted Airplanes"*, MIL-F-8785C, USAF, Wright Patterson AFB, OH, Nov. 1980.

[7] D. E. Rumelhart, J. L. McClelland, et al.: *Parallel Distributed Processing - Exploration in the Microstructure of Cognition, Volume 1: Foundations.*, MIT Press, Cambridge, MA, 1986.
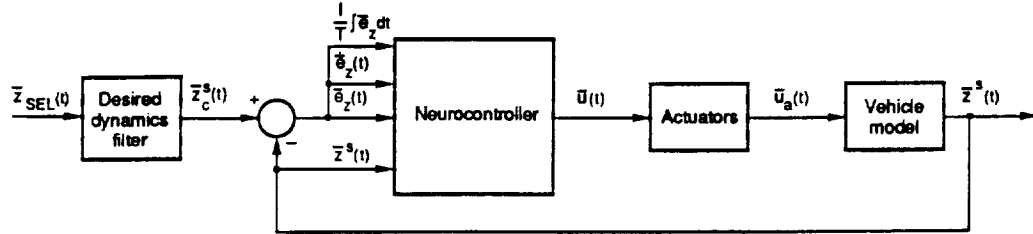
Figure 1.—Training architecture.

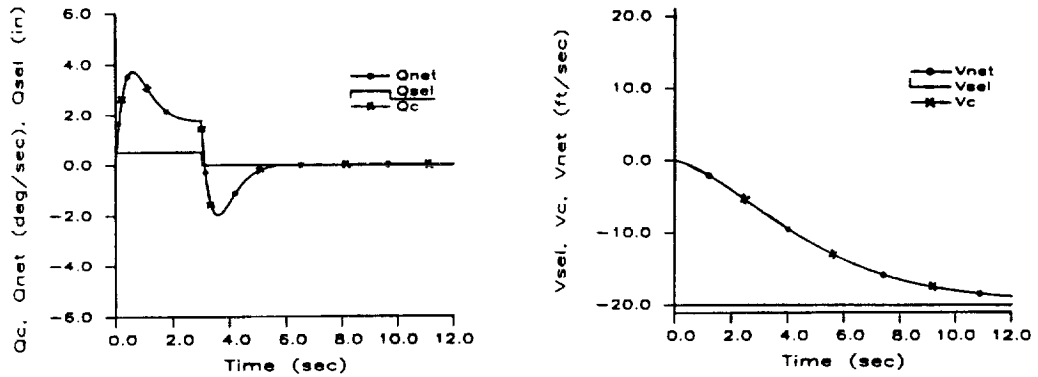Figure 2.—Evaluation architecture of closed-loop neurocontroller.

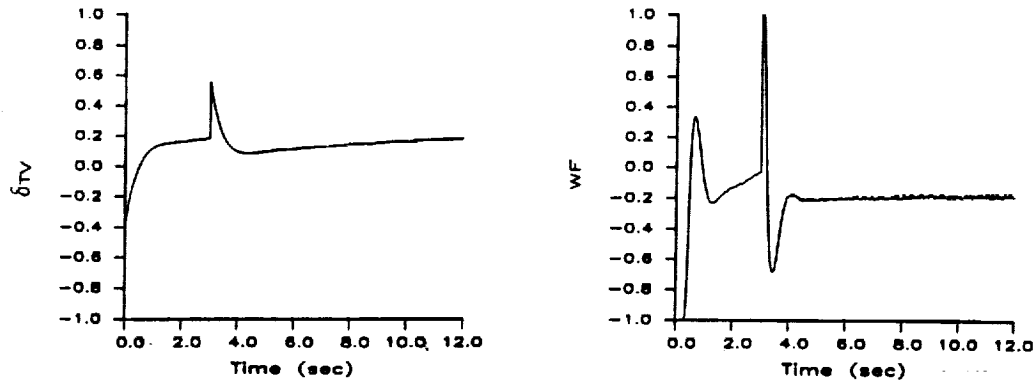Figure 3a. Closed-Loop Neurocontroller *without* Actuators; case 1 with $\bar{\bar{\lambda}} = \bar{\bar{\mu}} = \bar{\bar{0}}$.

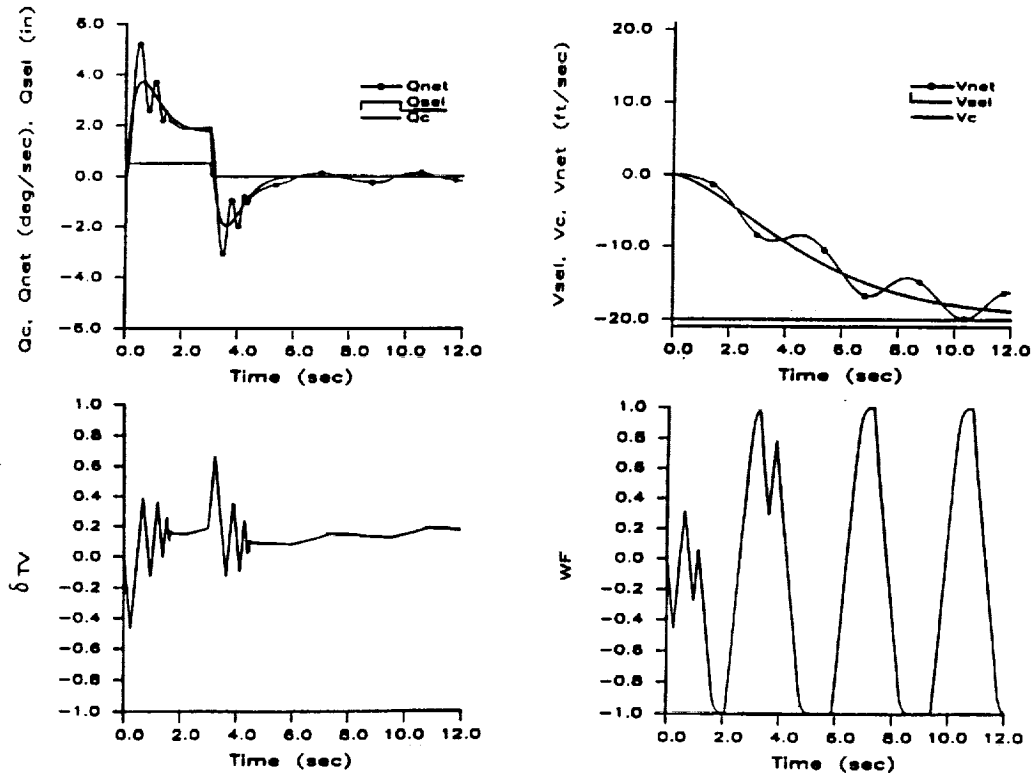Figure 3b. Closed-Loop Neurocontroller *without* Actuators; case 1 with $\bar{\bar{\lambda}} = \bar{\bar{\mu}} = \bar{\bar{0}}$.

Figure 4. Closed-Loop Neurocontroller *with* Actuators; case 1 with $\bar{\bar{\lambda}} = \bar{\bar{\mu}} = \bar{\bar{0}}$.
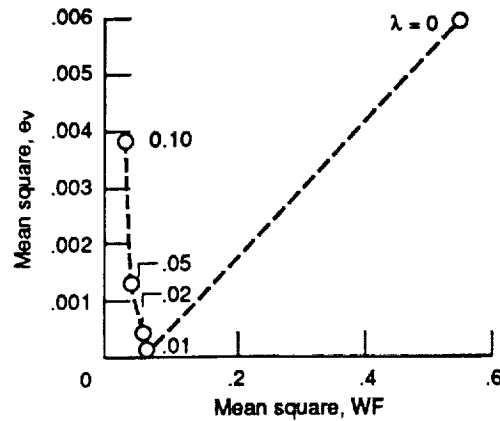

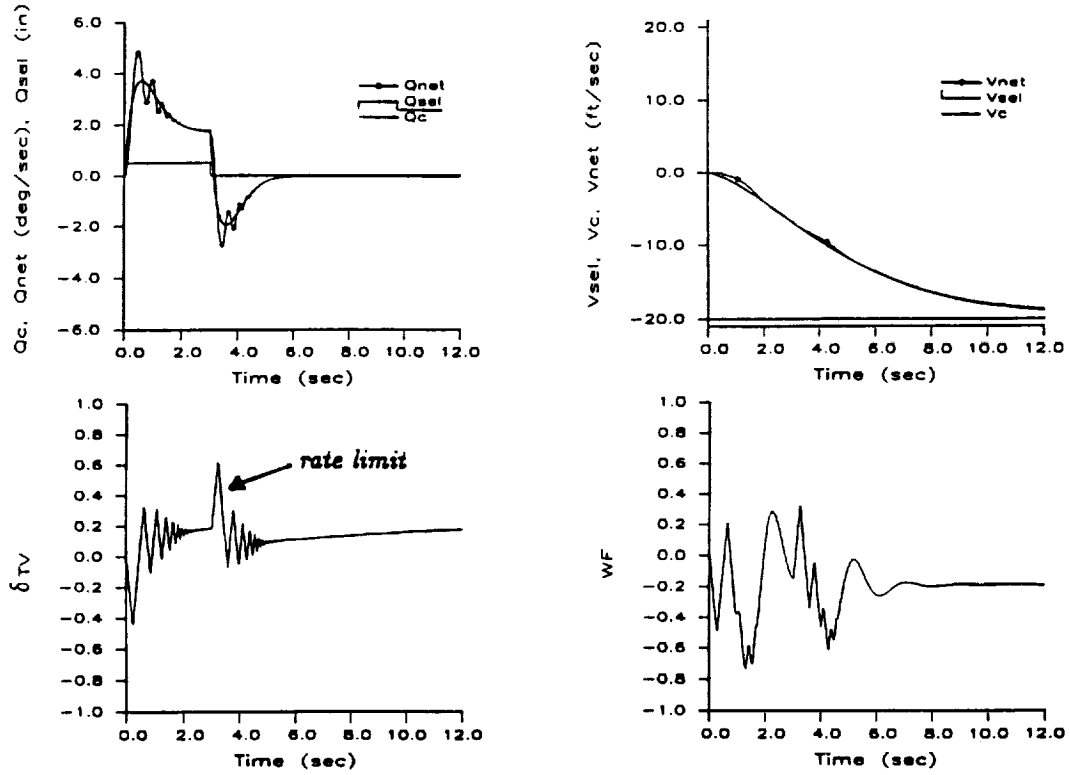Figure 5.—Performance/control trade-off; effect of varying λ for case 1.

Figure 6. Closed-Loop Neurocontroller *with* Actuators; case 1 with $\bar{\lambda} = 0.01\bar{I}$ and $\bar{\mu} = \bar{0}$.
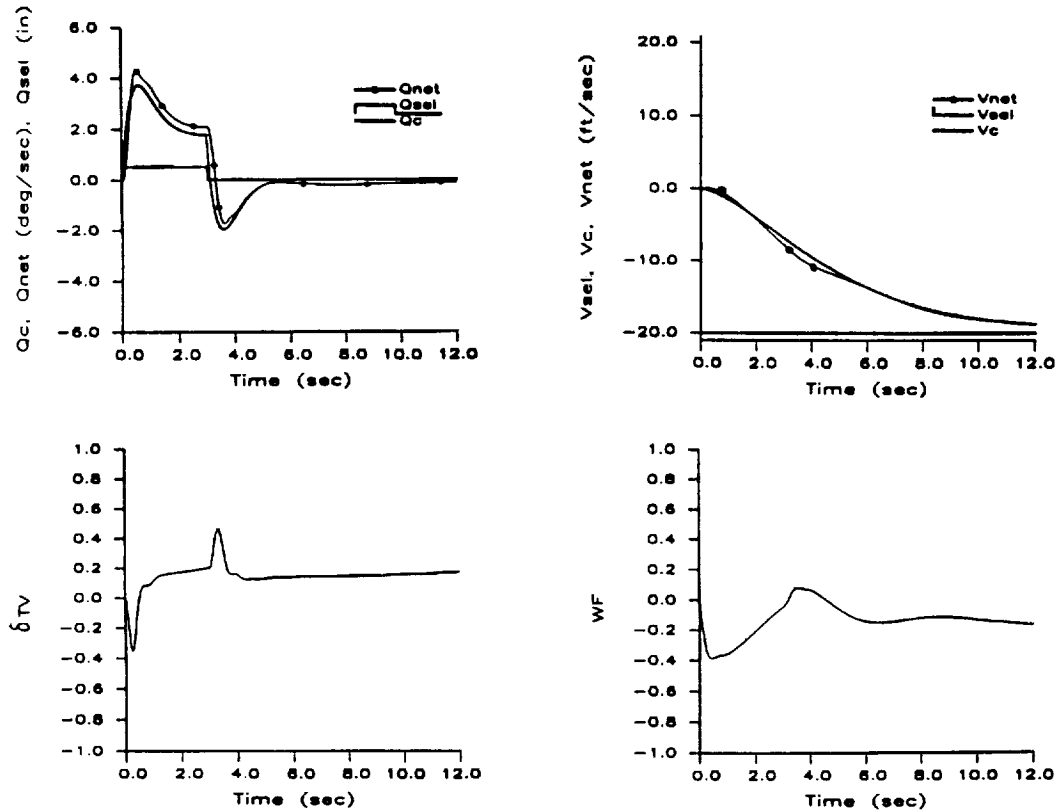
Figure 7. Closed-Loop Neurocontroller *with* Actuators; case 1 with $\bar{\lambda} = \bar{\mu} = 0.01\bar{I}$.

NASA

National Aeronautics and
Space Administration

Report Documentation Page

1. Report No. NASA TM - 103785	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle		5. Report Date
Towards Practical Control Design Using Neural Computation		
		6. Performing Organization Code

7. Author(s)		8. Performing Organization Report No.
Terry Troudet, Sanjay Garg, Duane Mattern, and Walter Merrill		E - 6061
		10. Work Unit No. 582 - 01 - 11

9. Performing Organization Name and Address		11. Contract or Grant No.
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135 - 3191		
		13. Type of Report and Period Covered

12. Sponsoring Agency Name and Address		Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546 - 0001		14. Sponsoring Agency Code

16. Abstract

The objective of this paper is to develop neural network based control design techniques which address the issue of performance/control effort trade-off. Additionally, the control design needs to address the important issue of achieving adequate performance in the presence of actuator nonlinearities such as position and rate limits. These issues are discussed using the example of aircraft flight control. Given a set of pilot input commands, a feedforward net is trained to control the vehicle within the constraints imposed by the actuators. This is achieved by minimizing an objective function which is a weighted sum of the tracking errors, control input rates and control input deflections. A trade-off between tracking performance and control smoothness is obtained by varying, adaptively, the weights of the objective function. The neurocontroller performance is evaluated in the presence of actuator dynamics using a simulation of the vehicle. Appropriate selection of the different weights in the objective function resulted in good tracking of the pilot commands and smooth neurocontrol. An extension of the neurocontroller design approach is proposed to enhance its practicality.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Control Neural networks Neurocontrol Supervised learning	Unclassified - Unlimited Subject Category 63

19. Security Classif. (of the report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	8	A02