N95-32434

NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS BY ARTIFICIAL NEURAL NETWORKS

Final Report

NASA/ASEE Summer Faculty Fellowship Program–1994

Johnson Space Center

Prepared By:	Andrew J. Meade, Jr., Ph.D
Academic Rank:	Assistant Professor
University & Department	Rice University Department of Mechanical Engineering and Materials Science Houston, Texas 77251-1892
NASA/JSC	
Directorate:	Information Systems
Division:	Information Technology
Branch:	Software Technology
JSC Colleague:	Robert O. Shelton, Ph.D
Date Submitted:	August 12, 1994
Contract Number:	NGT-44-005-803

Section 17 not used.

ABSTRACT

Conventionally programmed digital computers can process numbers with great speed and precision, but do not easily recognize patterns or imprecise or contradictory data. Instead of being programmed in the conventional sense, artificial neural networks (ANNs) are capable of self-learning through exposure to repeated examples. However, the training of an ANN can be a time consuming and unpredictable process.

A general method is being developed by the author to mate the adaptability of the ANN with the speed and precision of the digital computer. This method has been successful in building feedforward networks that can approximate functions and their partial derivatives from examples in a single iteration. The general method also allows the formation of feedforward networks that can approximate the solution to nonlinear ordinary and partial differential equations to desired accuracy without the need of examples. It is believed that continued research will produce artificial neural networks that can be used with confidence in practical scientific computing and engineering applications.

INTRODUCTION

Artificial neural networks (ANNs) have proven to be versatile tools for accomplishing what could be termed higher order tasks such as pattern recognition, classification, and visual processing. However, conventional wisdom has held that networks are unsuited for use in more purely computational tasks, such as mathematical modelling and physical analysis of engineering systems. Certainly the biological underpinnings of the neural network concept suggest that networks would perform best at tasks at which biological systems excel, and worse or not at all at other tasks.

Contrary to opinion the author believes that continued research into the approximation capabilities of networks will enable the neural network paradigm, with all of its advantages in behavior and adaptability, to be mated to the more purely computational paradigms of mathematically oriented scientific programming and analysis. Additionally, it is felt that the thorough investigation of network approximation capabilities will benefit the network field and connectionism in general.

In a field as conceptually difficult as the study of artificial neural networks, it is best to start investigation with supervised learning, test the established premises, and alter them to circumvent pitfalls in implimentation.

FUNCTION APPROXIMATION

Learning as Function Approximation

Central to the author's research approach is the view that supervised learning in artificial neural networks is equivalent to the problem of approximating a multivariate function and that learning should be able to be explained by approximation theory. Approximation theory deals with the problem of approximating or interpolating a multivariate function. This approach has been considered by other researchers in the field of ANNs [1]-[4]. However, the author extends this assumption of function approximation by assuming that ANNs can model discontinuous multivariate functions and should be at least as accurate and numerically efficient as existing computational techniques used in science and engineering. Also, ANN behavior and adaptation difficulties, from supervised learning to machine vision, should be amenable to the standard error analysis techniques used in numerical analysis.

In a general sense experiments, analytical methods, and computational methods can be considered to be forms of function approximation. The governing equations derived from analytical methods are a compact representation of the functions that model some particular phenomena observed in experiments. Computational techniques are used to approximate the function or functions that satisfy the governing equations. The graphs and tables

made from experiments are representations of the functions that underlie observed physical phenomena.

PROGRAMMABLE ARTIFICIAL NEURAL NETWORKS

If we are to assume that ANNs are as valid as established computational techniques, then ANNs should be evaluated in the same manner as are computational techniques. In evaluating the capabilities of a new numerical method, it is prudent to first apply it to the solution of algebraic and ordinary and partial differential equations of known behavior. This same approach can be used for ANNs since the solution of algebraic and differential equations can be viewed as the approximation of a function that must satisfy the equation in question subjected to boundary and/or initial conditions.

Applying an ANN to the solution of an algebraic or differential equation effectively uncouples the influences of the quality of data samples, network architecture, and transfer functions from the network approximation performance. The solution of equations also allows us to study the influence of constraining the connection weights. The most immediate benefit in this approach would be the construction of networks that can approximate the solution to desired equations without the need for examples. This would be of value in engineering applications since considerable effort may be saved if the equations governing a physical process can be directly incorporated into the neural network architecture without the need of examples, thereby shortening or even eliminating the learning phase.

The author has previously reported the use of the piecewise linear hard limit transfer function [5] in solving model algebraic and linear ordinary differential equations by the feedforward architecture [7]. In this report the author will use the recurrent artificial neural network (RANN) architecture to model a chaotic system by solving a nonlinear ordinary differential equation known as Duffing's equation.

APPROACH

A simple RANN consists of two layers of processing elements as illustrated in Fig. 1 for two coupled time-dependent variables u and v, where I denotes the input layer, and II denotes the output layer. Generally speaking, the inputs u and v, fed into layer I at time t^n, are operated on by the processing elements (PEs) and then multiplied by the constant connection weights a_{iq}. The coefficient a_{iq} is the connection weight between the ith neuron of layer I and the qth neuron of layer II. As a result, the outputs of layer I, multiplied by the connection weights, are input to layer II PEs. The output of layer II neurons are the values of the dependent variables at time t^{n+1}. These values are in turn fed back into the input layer I for the next iteration.

The dynamic behavior of such a RANN is governed by the value of the connection weights, the type of processing elements in the RANN, and the transfer functions used. In this study we restrict ourselves to using the piecewise linear transfer function in all layers of the RANN, since it is one of the simplest functions to implement on both digital and analog forms [8].

The hard limit function $\Upsilon_q^K(u)$ (Fig. 2) operating on the dependent variable u for the qth neuron of the Kth layer can be modelled by the following equations:

$$\Upsilon_q^K(u) = \begin{cases} -1.0 & \text{for} \quad \xi_q < -1.0 \\ \xi_q & \text{for} \quad -1.0 \leq \xi_q \leq 1.0 \\ +1.0 & \text{for} \quad \xi_q > 1.0 \end{cases}$$

where

$$\xi_q = \sum_i a_{iq} u_i \quad \text{for additive neurons}$$

$$\xi_q = \prod_i a_{iq} u_i \quad \text{for multiplicative neurons}$$

By correctly assigning different values to the RANN connection weights, one is able to integrate a system of coupled recurrent relations into the architecture without the need of training. We will now show how to obtain such recurrent relations from a system of ordinary differential equations.

Duffing's Equation

The inhomogeneous Duffing's equation describes the forced motion of a particle between two equilibrium states and can be written as the following nondimensional second-order ordinary differential equation

$$\frac{d^2x}{dt^2} + 2\mu\frac{dx}{dt} - \frac{1}{2}\left(x - x^3\right) = F_0 \cos(\omega t) \tag{1}$$

where t, x, μ, F_0, and ω represent time, displacement, damping coefficient, force amplitude, and frequency of excitation respectively. The importance of this equation is that its chaotic and nonchaotic behavior has been extensively examined by theoretical [9], experimental [10], and numerical methods [6], [11] - [13].

In order that Duffing's equation can be more easily approached by the RANN programming method and so that we may obtain particle displacement, velocity, and acceleration, Eq.(1) is reduced to a system of two first-order equations by the following change of variables

$$s = \frac{x}{\kappa_1} \quad , \quad y = \frac{1}{\kappa_2}\frac{dx}{dt} = \frac{1}{\kappa_2}\dot{x} \quad \text{and} \quad w = \frac{1}{\kappa_3}\frac{d^2x}{dt^2} = \frac{\kappa_2}{\kappa_3}\frac{dy}{dt}$$

where κ_1, κ_2, and κ_3 are constants used to scale the values of the new variables. Therefore, Eq.(1) becomes

$$\frac{ds}{dt} = \kappa_4 y \tag{2}$$

$$\frac{dy}{dt} = -2\mu y + \frac{1}{2\kappa_4}\left(s - {\kappa_1}^2 s^3\right) + \frac{F_0}{\kappa_2}\cos(\omega t) = \frac{\kappa_3}{\kappa_2} w \tag{3}$$

$$\text{where} \quad \kappa_4 = \frac{\kappa_2}{\kappa_1}.$$

It will now be demonstrated how the RANN architecture can be used to approximate the solution to Eq.(2) and Eq.(3) for arbitrary coefficients.

Integration of Duffing's Equation

For this problem the initial value problem of Eqs.(2) and (3) will be integrated by the explicit MacCormack method [14] which is a finite-difference, predictor-corrector scheme commonly used in the solution of time-dependent fluid dynamics equations. The application of the MacCormack technique to a general first order ordinary differential equation

$$\frac{du}{dt} - f(u,t) = 0 \ ,$$

where f is some arbitrary function of the dependent variable u and independent variable t, results in

Predictor: $\quad u^* = u^n + \Delta t f(u^n, t^n)$

Corrector: $\quad u^{n+1} = \frac{1}{2}(u^n + u^* + \Delta t f(u^*, t^*)) = u^n + \frac{\Delta t}{2}(f(u^n, t^n) + f(u^*, t^*))$

where the superscripts denote the time level and $t^* = t^n + \Delta t$.

Application of the MacCormack method to Eqs.(2) and (3) results in

$$\begin{aligned} s^* &= s^n + \Delta t f_1(s^n, y^n, t^n) \\ s^{n+1} &= s^n + \frac{\Delta t}{2}(f_1(s^n, y^n, t^n) + f_1(s^*, y^*, t^*)) \\ y^* &= y^n + \Delta t f_2(s^n, y^n, t^n) \\ y^{n+1} &= y^n + \frac{\Delta t}{2}(f_2(s^n, y^n, t^n) + f_2(s^*, y^*, t^*)) \end{aligned}$$

where

$$f_1(s,y,t) = \kappa_4 y \quad \text{and} \quad f_2(s,y,t) = -2\mu y + \frac{1}{2\kappa_4}\left(s - {\kappa_1}^2 s^3\right) + \frac{F_0}{\kappa_2}\cos(\omega t)$$

Substitution of the expressions for f_1, f_2, s^*, and y^* into the equations for s^{n+1} and y^{n+1} results in the following algebraic system,

$$\begin{aligned} p^n &= (s^n)^3 && (4)\\ r^n &= (s^*)^3 = (s^n + \kappa_4 \Delta t\, y^n)^3 && (5)\\ s^{n+1} &= H_1 s^n + H_2 y^n + H_3 p^n + H_4 \cos(n\omega\Delta t) && (6)\\ y^{n+1} &= H_5 s^n + H_6 y^n + H_7 p^n + H_8 r^n + H_9 \cos(n\omega\Delta t) + \\ &\quad H_{10} \cos((n+1)\omega\Delta t) && (7) \end{aligned}$$

The coefficients H_1 through H_{10} are

$$H_1 = 1 + \frac{\Delta t^2}{4} \quad , \quad H_2 = \kappa_4 \Delta t\,(1 - \mu\Delta t) \quad , \quad H_3 = -\frac{\kappa_1^2 \Delta t^2}{4}$$

$$H_4 = \frac{F_0 \Delta t^2}{2\kappa_1} \quad , \quad H_5 = \frac{\Delta t}{2\kappa_4}(1 - \mu\Delta t) \quad , \quad H_6 = 1 - 2\mu\Delta t + 2\mu^2\Delta t^2 + \frac{\Delta t^2}{4}$$

$$H_7 = -\frac{{\kappa_1}^3}{\kappa_2}\frac{\Delta t}{4}(1 - 2\mu\Delta t) \quad , \quad H_8 = -\frac{{\kappa_1}^3}{\kappa_2}\frac{\Delta t}{4} \quad , \quad H_9 = \frac{F_0}{\kappa_2}\frac{\Delta t}{2}(1 - 2\mu\Delta t)$$

$$H_{10} = \frac{F_0}{\kappa_2}\frac{\Delta t}{2} \quad , \qquad (8)$$

If Δt is required to be constant, then the time-dependent coefficients of Eq.(8) become constants for specific values of μ, F_0, κ_1, κ_2, and κ_3. Equations (4) through (7) are the linear and nonlinear algebraic equations that approximate Duffing's equation and must be modelled by the RANN.

Network Approximation of Duffing's Equation

To compare against previous numerical studies of Duffing's equation, we require only that the RANN generate values for particle displacement and velocity. Figures 3 - 5 show the connections required in a constrained RANN modelling Eqs.(4) through (7) respectively. It is understood that the network output is fed back to the input layer though the connections are not shown in the figures. An initial time $t^0 = 0$ is assumed.

Note that the magnitude of all inputs and outputs are scaled to be less than value 1 so that hard limits may be used in the processing elements. This constraint on the magnitude of the dependent variables s^n, y^n, p^n, and r^n requires that

$$\kappa_1 > |x|_{max} + \Delta t \left|\frac{dx}{dt}\right|_{max} \quad , \quad \kappa_2 > \left|\frac{dx}{dt}\right|_{max} \quad , \quad \kappa_3 > \left|\frac{d^2x}{dt^2}\right|_{max} \qquad (9)$$

Equations (4) and (5) possess cubic time-dependent unknowns that are approximated by the RANN using the multiplicative neuron model also known as the Pi neuron (Fig. 3) [5]. In traditional ANNs, the Pi neurons are used to achieve greater processing power by using more complex connections.

The fully assembled RANN is shown in Fig. 6. Updated values of the cosine functions ($\cos(n\omega\Delta t)$ and $\cos((n+1)\omega\Delta t)$) must be input at each iteration in the RANN.

RESULTS

The RANN of Fig. 6 was constructed and run with a fixed damping value of $\mu = 0.084$ and fixed forcing amplitude of $F_0 = 0.178$. Previous results of the two-well potential system [6] indicate displacement ranging between $\pm$ 2.0 and velocity ranging between $\pm$ 1.0. Assuming that $\Delta t < |x|_{max}$ Eq.(9) allows the use of $\kappa_1 = 4.0$, $\kappa_2 = 2.0$, and $\kappa_3 = 1.0$. Initial conditions were set for $x(0) = 1.0$ and $\dot{x}(0) = 0.0$, except for Fig. 8 where $x(0) = 0$ and $\dot{x}(0) = 1.0$ for the coexisting global attractor. To ensure that steady-state solutions were displayed, approximations were run for 900 periods, T, based on the forcing frequency ω ($T = 2\pi/\omega$). All computations were done in double precision and all test cases were run in less than 40 real time seconds on a Sparc 10 SX Model 512 workstation.

RANN Simulation Results

Figures 7 - 9 compare the constructed RANN output ($x = \kappa_1 s$ and $\dot{x} = \kappa_2 y$) with the numerical results of Masoud and Asfar [6], for the same parameter values and initial conditions. Here, as with reference [6] the damping and excitation amplitude are based on the work presented by Pezeshki and Dowell [13]. Throughout the simulation the time step used in the RANN was identical to that used in the Masoud and Asfar study, $\Delta t = T/100$. For clarity, plots of time histories have been excluded. Phase plane trajectories, as well as Poincaré maps are shown and compared in cases where they are shown by reference [6].

Figure 7 starts with $\omega = 1.020$ at which the period of the oscillator is equal to 1 (period 1 oscillation) and doubles as ω decreases. Thus we observe period 2, period 4, and period 8 oscillations, until the system evolves into a coexisting global period 5 motion (Fig. 8) at which the system oscillates between the two equilibrium states $(\dot{x}, x)$ respectively $(0, -1)$ and $(0, 1)$. Figure 9 describes the first region of chaotic oscillations starting at $\omega = 0.865$ and ending at $\omega = 0.780$.

CONCLUSIONS

Using the piecewise linear transfer function and additive and multiplicative neurons the author was able to construct a recurrent artificial neural network that was capable of accurately solving a complex nonlinear problem. The solution of the inhomogenous Duffing's equation was shown as a numerical

example and compared well with the computational results of Masoud and Asfar. This was done without the need for data or conventional training. By establishing a direct link between a well known numerical method and the operation of a RANN we believe it may be possible to link common numerical methods from computational mechanics to the operation of the more popular ANN paradigms.

REFERENCES

[1] Barron, A.R. and Barron, R.L., "Statistical learning networks: a unifying view," *Symposium on the Interface: Statistics and Computing Science,* Reston, Virginia, April 1988.

[2] Omohundro, S., "Efficient algorithms with neural network behaviour," *Complex Systems*, **1**, 237, 1987.

[3] Poggio, T. and Girosi, F., "A Theory for Networks for Approximation and Learning," *Artificial Intelligence Laboratory, Massachusetts Institute of Technology*, A.I. Memo No. 1140, July 1989.

[4] Girosi, F. and Poggio, T., "Networks for learning: a view from the theory of approximation of functions," *Proceedings of the Genoa Summer School on neural networks and their applications*, Prentice Hall, 1989.

[5] Maren, A., Harston, C. and Pap, R., *Handbook of Neural Computing Applications*, 48, Addison-Wesley Publishing, 1990.

[6] Masoud, K. K. and Asfar, K. R., "Period Doublings, Bifurcations and Strange Attractors in a Two-Well Potential Oscillator," *European Journal of Mechanics, A/Solids*, Vol. 12, No. 3, 1993, pp. 417-428.

[7] Meade, A. J., "Development of Programmable Artificial Neural Networks," *NASA Contractor Report*, Contract Number NGT-44-001-800, 1993, pp. 21-1 - 21-15.

[8] Chua, L. O., Desoer, C. A. and Kuh, E. S., *Linear and Nonlinear Circuits*, McGraw Hill, New York, 1987. pp. 76-83, 171-212.

[9] Holmes, P., "A Nonlinear Oscillator with a Strange Attractor," *Phil. Trans. The Roy. Soc. London*, Vol. 292, 1979, pp. 419-448.

[10] Moon, F. C., "Experiments on Chaotic Motions of a Forced Nonlinear Oscillator: Strange Attractors," *ASME Journal of Applied Mechanics*, Vol. 47, 1980, pp. 638-644.

[11] Ueda, Y., "Randomly Transitional Phenomena in the System Governed by Duffing's Equation," *Journal of Statistical Physics*, Vol. 20, No. 2, 1979, pp. 181-196.

[12] Dowell, E. H. and Pezeshki, C., "On the Understanding of Chaos in Duffings Equation Including a Comparison With Experiment," *Journal of Applied Mechanics*, Vol. 53, 1986, pp. 5-9.

[13] Pezeshki, C.,and Dowell, E. H., "On Chaos and Fractal Behaviour in a Generalised Duffing's System," *Physica D*, Vol. 32, 1988, pp. 194-209.

[14] Anderson, D. A., Tannehill, J. C. and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, New York, 1984, pp. 163-164.

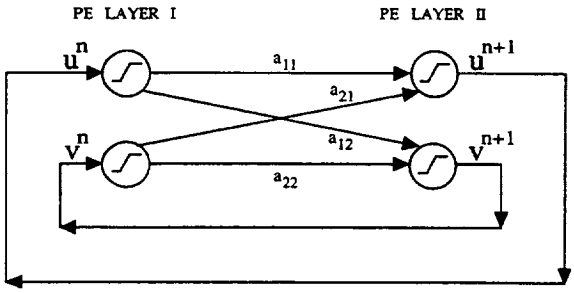

Figure 1: Basic RANN architecture.

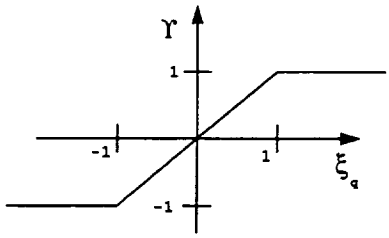

Figure 2: Hard limit transfer function.

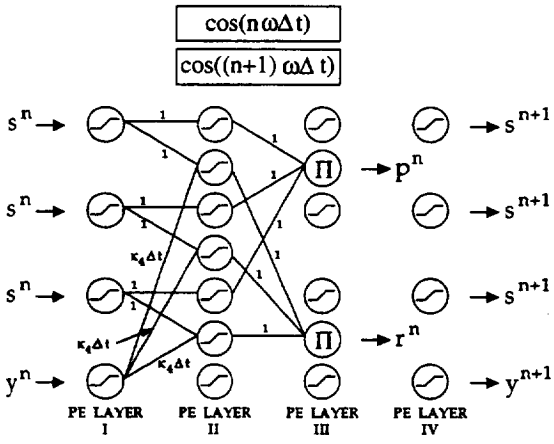

Figure 3: RANN representation of the nonlinear equations for p^n and r^n.

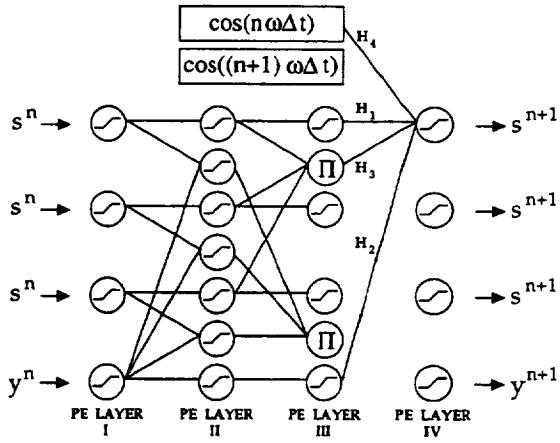

Figure 4: RANN representation of the recurrent equation for s^{n+1}.

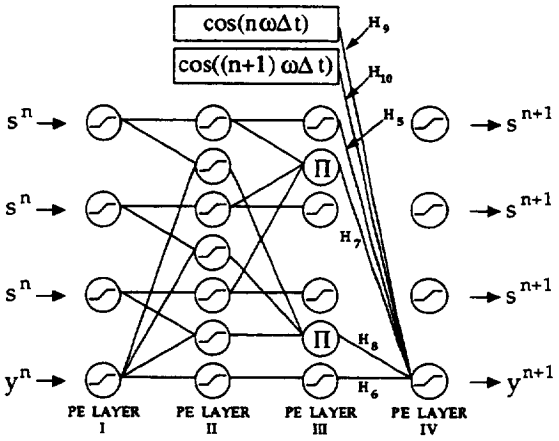

Figure 5: RANN representation of the recurrent equation for y^{n+1}.

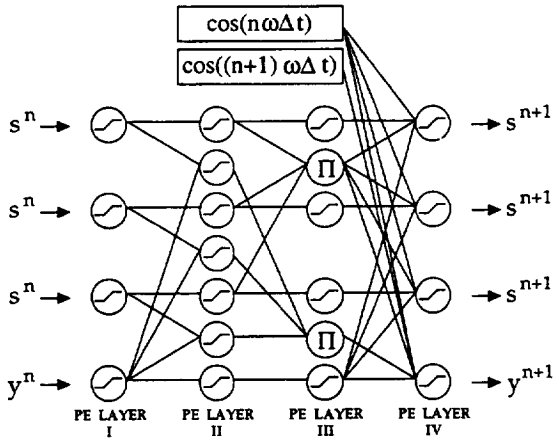

Figure 6: Full RANN assembly for the MacCormack scheme applied to Duffing's equation.

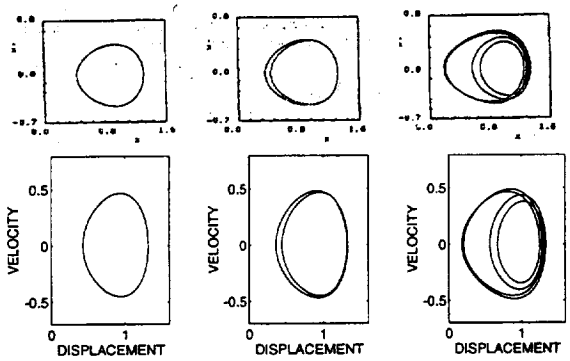

Figure 7: Comparison of phase space trajectories from Masoud and Asfar (top set) and the RANN (bottom set). From left to right: (a) $\omega = 1.020$ period 1, (b) $\omega = 1.000$ period 2, (c) $\omega = 0.963$ period 4.

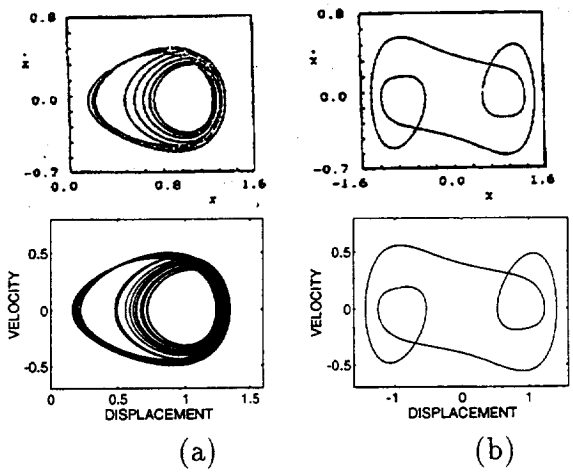

Figure 8: Phase space trajectories from Masoud and Asfar (top set) and the RANN (bottom set). From left to right: coexisting (a) $\omega = 0.957$ period 8, (b) $\omega = 0.957$ global period 5.

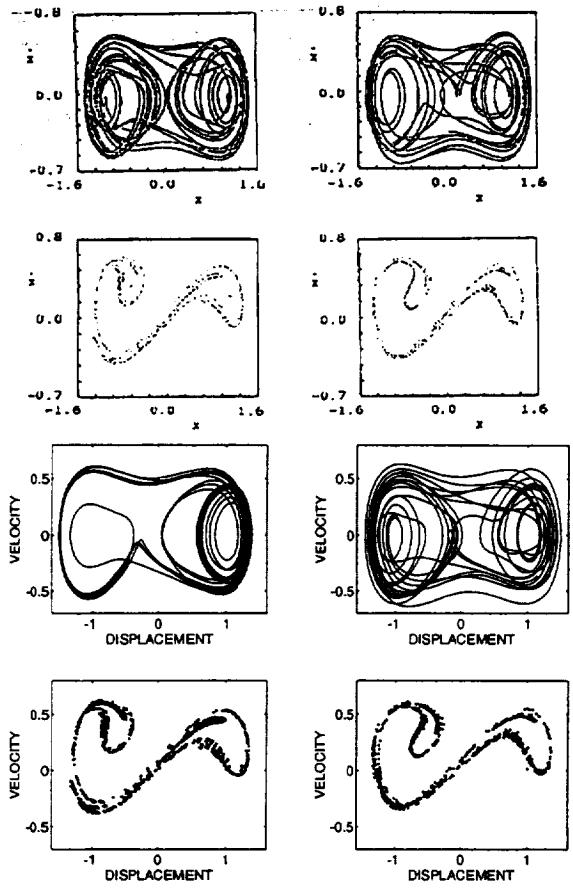

Figure 9: Comparison of phase space trajectories and Poincaré maps in the first chaotic region from Masoud and Asfar (top set) and the RANN (bottom set). From left to right: (a) $\omega = 0.865$ chaotic higher frequency boundary, (b) $\omega = 0.780$ chaotic lower frequency boundary.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1995	Contractor Report

4. TITLE AND SUBTITLE

National Aeronautics and Space Administration (NASA)/American Society for Engineering Education (ASEE) Summer Faculty Fellowship Program - 1994, Volume 1

5. FUNDING NUMBERS

6. AUTHOR(S)

Richard Bannerot* and Donn G. Sickorez, Editors

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lyndon B. Johnson Space Center
Houston, Texas 77058

8. PERFORMING ORGANIZATION REPORT NUMBERS

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

CR-188410

11. SUPPLEMENTARY NOTES

*University of Houston, Houston, Texas

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified/Unlimited
Available from the NASA Center for AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390 Subject Category: 99

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The JSC NASA/ASEE Summer Faculty Fellowship Program was conducted by Texas A&M University and JSC. The objectives of the program, which began nationally in 1964 and at JSC in 1965 are to (1) further the professional knowledge of qualified engineering and science faculty members, (2) stimulate an exchange of ideas between participants and NASA, (3) enrich and refresh the research and teaching activities of participants' institutions, and (4) contribute to the research objectives of the NASA centers. Each faculty fellow spent at least 10 weeks at JSC engaged in a research project in collaboration with a NASA JSC colleague. This document is a compilation of the final reports on the research projects completed by the faculty fellows during the summer of 1994.

14. SUBJECT TERMS

information transfer, research, research projects, urban research, engineering, science, universities, university program

15. NUMBER OF PAGES
202

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited