A RECURRENT ARTIFICIAL NEURAL NETWORK MODEL OF DUFFING'S EQUATION WITHOUT TRAINING

Andrew J. Meade, Jr.* and Rafael Moreno
Department of Mechanical Engineering
and Materials Science
Rice University
Houston, Texas, 77251-1892, USA
Phone: (713) 527-8101 ext. 3590
E-mail: meade@rice.edu

International Journal of Smart Engineering System Design

---

* Author to whom correspondence should be addressed.
This work is supported under NASA grant NAG 9-719 and ONR grant N00014-95-1-0741.

A RECURRENT ARTIFICIAL NEURAL NETWORK MODEL OF DUFFING'S EQUATION WITHOUT TRAINING

Andrew J. Meade, Jr.* and Rafael Moreno†
Department of Mechanical Engineering
and Materials Science,
Rice University
Houston, TX 77251-1892, USA

Abstract

A method is developed for constructing recurrent artificial neural networks to model physical systems. The construction requires imposing certain constraints on the values of the input, bias, and output weights. The attribution of certain roles to each of these parameters allows for mapping a polynomial approximation into an artificial neural network architecture. Attention is focused on a second-order nonlinear ordinary differential equation, which governs the well known Duffing's oscillator. The nonlinear ordinary differential equation is modelled by the recurrent artificial neural network architecture in conjunction with the popular hyperbolic tangent transfer function. Moreover, this approach is shown to be capable of incorporating other smooth neuron transfer functions, as long as they can be described by a Taylor series expansion. Numerical examples are presented illustrating the accuracy and utility of the method.

Key Words: recurrent artificial neural networks, neural computation, differential equations, chaos, network training

1 INTRODUCTION

Presently, an area of increasing interest is the emulation and control of nonlinear dynamic systems by the recurrent artificial neural network (RANN) architecture. These particular networks use supervised learning algorithms (training algorithms) such as recurrent

* Associate Professor, E-mail: meade@rice.edu
† Research Assistant
This work is supported under NASA grant NAG 9-719 and ONR grant N00014-95-1-0741.

backpropagation [1]-[3], which allow the network to simulate a dynamic system without knowledge of the governing equations.

The supervised training of a RANN system can be formulated as the procedure of selecting the optimal network parameters so that the response of the neural network is "close" to the available data, as measured by an error criterion. In terms of the connectionist literature, numerical minimization of this error criterion by the steepest descent method constitutes the backpropagation algorithm [1]. The conjugate gradient method, the Levenberg-Marguardt method, and other nonlinear optimization methods have also been used for network training. However, it has been found that even the most advanced optimization algorithms can show unreliable convergence in training and can be highly dependent on nonlinear parameter constraints and their initial values [4]. In modelling physical systems, the initial values for the neural network parameters are commonly selected by random sampling [5]. If a satisfactory solution is not obtained during training, the nonlinear optimization procedure is restarted with new randomly selected values [6]. Clearly this approach can require prohibitive computational resources for some practical problems.

With the many computational mechanics tools available to the typical engineer, it is doubtful that he or she would resort to data driven modelling by a RANN unless an analytical description of acceptable fidelity were unavailable. Add to this the previously mentioned difficulties in training, and it is apparent why artificial neural networks are not used more widely in engineering. However, unlike soft-computing problems, even the most highly complex physical problem possesses a course analytical description that can be derived from semi-empirical relations or the conservation of mass, momentum and energy. Since it is well known that training algorithms either reduce to, or are, nonlinear optimization techniques, it is reasonable to assume that they will benefit from physically meaningful initial values when modelling a physical system.

In this regard, a new connectionist training initialization method is proposed that can be developed from accurate solutions of the time-dependent differential equations, using only the equations of interest and the initial conditions. This proposed initialization method is based on three assumptions. Firstly, an artificial neural network system can be treated as a general approximation scheme [7] capable of approximating differential equations. Secondly, polynomial bases, which are a traditional method for approximating functions and analyzing experimental data [8], [9], can be mapped into neural network architectures through constraints on the network parameters. Thirdly, a-priori mathematical models of engineering systems, linear and nonlinear, can be utilized to obtain initial values for the network parameters.

Note that this proposed approach would also allow the initialization of a connectionist training algorithm without data; this feature could be especially useful for on-line RANN modelling systems. Also, the method could be useful in obtaining preliminary bounds on the network performance and estimating the accuracy of network approximations. For example, by exploiting the analogy between ANN systems and polynomial approximations, the effect of augmenting networks with additional processing elements could be evaluated. In addition, initiating training with a mathematical model of sufficient fidelity significantly reduces problems associated with the inadequacy of a small training set in describing a physical system.

However, the proposed network initialization method is critically dependent on demonstrating that a conventional RANN, using nonlinear transfer functions, can accurately approximate a physical model in the form of differential equations and that the accuracy of the constructed RANN can be controlled by the user. Consequently, the purpose of this paper is to show how a conventional RANN can be constructed to model a nonlinear time-dependent ordinary differential equation without the need for training or resorting to special architectures. The connection weights of this network are determined by reducing the governing equations to systems of both linear and nonlinear algebraic equations and approximating the dependent variables by linear combinations of transfer functions. The algorithms used were relatively straightforward and can be run on conventional hardware. The resulting network is architecturally identical to those that use conventional training techniques, yet it possesses bounded errors that can be controlled by the network parameters in a straight-forward manner.

As a numerical example a RANN using a single input and output layer, with additive neurons utilizing the hyperbolic tangent activation function, has been constructed to approximate the solution to the inhomogeneous Duffing's equation. The Duffing's equation describes the forced oscillations of a particle in a two-well potential [10]. This is considered to be an interesting and practical example since the equation exhibits both chaotic and nonchaotic behavior for specific parameter combinations. The network output, in the form of phase plane trajectories and Poincaré maps, compares well with the results of previous computational investigations for both chaotic and nonchaotic conditions.

2 APPROACH

Since the RANN construction method is to ultimately serve as a supervised learning initialization technique, the development of the construction method is given in the framework of supervised training.

2.1 RANN Architecture

A recurrent artificial neural network (RANN) can consist of several layers of processing elements as illustrated in Fig. 1. It is understood that the network output is fed back to the input layer, though the connections are not shown in the figure. The variables $u_i(t)$, $i = 1, ..., M$ denote the input states of the RANN at the time t while the variables $v_j(t+1)$, $j = 1, ..., N$ denote the output states of the network at the time $t+1$ which in turn are fed back as some of the input variables used to determine the RANN response at the next time period. The first P components of the input state vector $\overline{u}(t)$ are selected as the input variables of the physical system, $P < M$. Similarly, the first L components of the vector $\overline{v}(t+1)$ specify the response of the corresponding physical system, $L \leq N$. The output at time $t+1$ of a RANN, consisting of k layers of nonlinear processing elements, can be determined by the equation

$$v_j(t+1,\overline{\beta}) \;=\; \beta^{N+1}_{k,j}\Upsilon_k\left(\sum_l \beta^j_{(k-1),l}\Upsilon_{(k-1)}\left(...\Upsilon_2\left(\sum_i \beta^p_{1,i}\Upsilon_1\left(u_i(t)\right)\right)\right)\right), \tag{1}$$

where $\overline{\beta}$ represents the set of network parameters, and Υ_j is the nonlinear transfer function of neurons associated with the j-th intermediate layer of the network.

Several alternative functions have been used as the processing elements in RANNs, but the more popular training procedures, such as backpropagation, require that the functions be continuously differentiable and bounded. In this regard, sigmoidal neurons are often selected for network applications. The corresponding functions are continuous and are characterized by the following limits

$$\lim_{\xi\to\infty}\Upsilon(\xi)=1 \quad , \quad \lim_{\xi\to-\infty}\Upsilon(\xi)=-1 \quad , \tag{2}$$

The function

$$\Upsilon(\xi)=\tanh\left(\xi\right) \tag{3}$$

is a paradigmatic example of such sigmoidal functions.

Figure 1 presented the general architecture of a RANN. However, in an actual application the user must specify the number of neuron layers as well as the number of neurons in each layer and their connections. These user specifications may be based on intrinsic considerations. In this study a simple RANN with two layers of nonlinear processing elements is used since this type of RANN is arguably the most common in dynamic network applications [11]. The transfer functions of these layers are taken as the hyperbolic tangent function of Eq. (3). The response of this type of RANN can be written as

$$v_j(t+1) \;=\; \beta^{N+1}_{2,j}\tanh\left(\sum_i \beta^j_{1,i}\tanh\left(u_i(t)\right)\right) \quad . \tag{4}$$

2.2 RANN Parameter Initialization Using Mathematical Models

2.2.1 Linear versus Non-Linear Training

The training of the RANN system is performed by adjusting the linear and nonlinear network parameters in $\overline{\beta}$ by nonlinear optimization procedures. As previously mentioned, the success in training depends to a great extent on an appropriate selection of the initial values of the network parameters. Note that the network response from Eq.(4) depends linearly only on the parameters $\beta_{2,j}^{N+1}$, other parameters affect the network response in a nonlinear manner. Consequently, initial values of the network parameters that would be adequate for successful network training can be determined efficiently by establishing a set of constraints on the nonlinear parameters. The response of these types of neural networks can then be rewritten as a linear combination of basis functions $\Phi_k\left(\overline{u}(t)\right)$. Specifically,

$$v_j(t+1) \cong \sum_k c_k^j \Phi_k\left(\overline{u}(t)\right) \quad . \tag{5}$$

In this paper global bases will be formed from linear combinations of transfer functions. The linear coefficients (connection weights) associated with the bases will be determined in the process of approximating the dependent variables.

2.2.2 Sigmoidal Networks and Polynomial Bases Approximation

The idea of mapping a conventional approximation scheme, with well established approximation properties, into neural network architecture has been pursued in the literature [12]-[16] for theoretical studies regarding the density and approximation properties of ANNs and the dependence of the network approximation error on the number of neurons. In this paper, the mapping approach is used with a new and practical intent. Specifically, the well known method of polynomial approximation, commonly used in data analysis and computational mechanics applications, will be mapped into a network architecture for the emulation of mechanical systems. Attention is drawn to the following formula

$$\psi_\wp\left(\overline{w},\overline{x}\right) = \frac{\partial^{|\wp|}}{\partial w_1^{p_1}...\partial w_s^{p_s}}\psi\left(\overline{w}^T\overline{x}+\theta\right) = \overline{x}^\wp\psi^{(|\wp|)}\left(\overline{w}^T\overline{x}+\theta\right) \quad . \tag{6}$$

where $(\cdot)^T$ denotes a vector transpose, $\overline{x}^\wp = {x_1}^{p_1}...\ {x_s}^{p_s}$, $\wp$ is a multi-index such that $|\wp| = p_1 + ... + p_s$, ψ is a nonlinear and non-polynomial function and superscript $(\cdot)$ represents the order of the ordinary derivative of ψ. Equation (6) shows that a polynomial

$\overline{x}^{\wp}$ can be readily expressed as

$$\overline{x}^{\wp} \quad = \quad \left(\psi^{(|\wp|)}(\theta)\right)^{-1} \psi_{\wp}\left(0, \overline{x}\right) \quad . \tag{7}$$

where the bias θ is selected by the user. Therefore, by replacing the partial derivative $\psi_{\wp}\left(0, \overline{x}\right)$ by an adequate finite difference approximation, one can accurately approximate the polynomial $\overline{x}^{\wp}$ by a finite linear combination of functions in the form $\psi\left(\overline{w}^T \overline{x} + \theta\right)$. The following finite difference scheme is given as an example

$$\psi_{\wp}\left(0, \overline{x}\right) \quad \cong \quad \Psi\left(\overline{x}\right) = \left(2\Delta w\right)^{-|\wp|} \sum_{0 \leq \zeta \leq \wp} \left(-1\right)^{|\zeta|} \begin{pmatrix} \wp \\ \zeta \end{pmatrix} \psi\left(\left(2\zeta - \wp\right)^T \Delta w \overline{x} + \theta\right) \quad , \tag{8}$$

where Δw is a small positive constant and the multi-integer binomial is defined as

$$\begin{pmatrix} \wp \\ \zeta \end{pmatrix} \quad = \quad \prod_{j=1}^{s} \begin{pmatrix} p_j \\ r_j \end{pmatrix} \quad . \tag{9}$$

Note that the accuracy of the finite difference scheme of Eq. (8), within a bounded domain of $\overline{x}$, is given by

$$\max\left|\Psi\left(\overline{x}\right) - \psi_{\wp}\left(0, \overline{x}\right)\right| \leq C\left(\Delta w\right)^2 \quad . \tag{10}$$

By combining Eqs. (8) and (9) and replacing the function $\psi\left(\overline{x}\right)$ by the neuron transfer function $\Upsilon\left(\overline{x}\right)$ it can be shown that the polynomial $\overline{x}^{\wp}$ can be approximated by a network layer of $\prod_{j=1}^{s}\left(p_j + 1\right)$ neurons. The error of this approximation can be made arbitrarily small by the selection of Δw and θ. For example, the linear function x can be approximated by x_a, where

$$x_a = \left(2\Delta w \quad \Upsilon^{(1)}(\theta)\right)^{-1} \left[\Upsilon\left(\Delta w x + \theta\right) - \Upsilon\left(-\Delta w x + \theta\right)\right] \quad . \tag{11}$$

This approach is also valid for constructing products of dependent variables,

$$\begin{aligned} x_a \; y_a = \left(4\Delta w_x \; \Delta w_y \; \Upsilon^{(2)}(\theta)\right)^{-1} & \left[\Upsilon\left(\Delta w_x x + \Delta w_y y + \theta\right) + \Upsilon\left(-\Delta w_x x - \Delta w_y y + \theta\right) + \right. \\ & \left. -\Upsilon\left(-\Delta w_x x + \Delta w_y y + \theta\right) - \Upsilon\left(\Delta w_x x - \Delta w_y y + \theta\right)\right] \end{aligned} \tag{12}$$

and multilayered approximations

$$x_{a2} = \left(2\Delta w \;\; \Upsilon^{(1)}(\theta)\right)^{-1} \left[\Upsilon\left(\Delta w x_{a1} + \theta\right) - \Upsilon\left(-\Delta w x_{a1} + \theta\right)\right] \quad , \tag{13}$$

where x_{a1} is the approximation from Eq.(11) and x_{a2} is the result from the second layer of neurons.

In the next subsection this approach will be combined with the mathematical model of a physical system to obtain a reliable and accurate scheme for RANN construction for training initialization.

2.3 Developing A RANN Model of Duffing's Equation

In this paper we will assume that an a-priori mathematical model of a mechanical system can be described by the following second-order ordinary differential equation

$$\frac{d^2x}{dt^2} + 2\mu\frac{dx}{dt} - \frac{1}{2}\left(x - x^3\right) = F_0 \;\cos\left(\omega t\right) \tag{14}$$

where t, x, μ, F_0, and ω represent the time, displacement, damping coefficient, force amplitude, and frequency of excitation, respectively. Equation (14) is the inhomogeneous Duffing's equation, which describes the forced motion of a particle between two equilibrium states. The importance of this equation is that its chaotic and nonchaotic behavior has been extensively examined by theoretical [17], experimental [18], and numerical methods [10], [19] - [21].

To make the Duffing's equation more easily approachable by the RANN construction method, and to allow the user to obtain particle displacement and velocity, Eq.(14) is reduced to a system of two first-order equations by the following change of variables

$$s = \frac{x}{\kappa_1} \quad , \quad y = \frac{1}{\kappa_2}\frac{dx}{dt} = \frac{1}{\kappa_2}\dot{x} \quad , \tag{15}$$

where κ_1 and κ_2 are constants used to scale the new variables. Therefore, Eq.(14) becomes

$$\frac{ds}{dt} \;\; = \;\; \kappa_3 y \tag{16}$$

where $\kappa_3 = \dfrac{\kappa_2}{\kappa_1}$ and

$$\frac{dy}{dt} = -2\mu y + \frac{1}{2\kappa_3}\left(s - {\kappa_1}^2 s^3\right) + \frac{F_0}{\kappa_2}\cos(\omega t) \quad . \tag{17}$$

2.3.1 RANN Model of the Duffing's Equation

For this paper integration of the initial value problem of Eqs. (16) and (17) will be made, for arbitrary coefficients, by the explicit MacCormack method [22]. The MacCormack method is a finite-difference, predictor-corrector scheme commonly used in the solution of time-dependent fluid dynamics equations. It should be noted that the RANN programming approach can use any time marching technique that can be put into explicit form (e.g. Runge-Kutta method). 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: $\quad u^{n+1} = \dfrac{1}{2}\left(u^n + u^* + \Delta t f(u^*, t^*)\right) = u^n + \dfrac{\Delta t}{2}\left(f(u^n, t^n) + f(u^*, t^*)\right)$

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

Application of the MacCormack method to Eqs. (16) and (17) results in

$$\begin{aligned} s^* &= s^n + \Delta t f_1(s^n, y^n, t^n) \\ s^{n+1} &= s^n + \frac{\Delta t}{2}\left(f_1(s^n, y^n, t^n) + f_1(s^*, y^*, t^*)\right) \end{aligned} \tag{18}$$

$$\begin{aligned} y^* &= y^n + \Delta t f_2(s^n, y^n, t^n) \\ y^{n+1} &= y^n + \frac{\Delta t}{2}\left(f_2(s^n, y^n, t^n) + f_2(s^*, y^*, t^*)\right) \end{aligned} \tag{19}$$

where

$$f_1(s,y,t) \;=\; \kappa_3 y \quad \text{and} \quad f_2(s,y,t) = -2\mu y + \frac{1}{2\kappa_3}\left(s - {\kappa_1}^2 s^3\right) + \frac{F_0}{\kappa_2}\cos(\omega t)$$

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

$$z^n = (s^n + \kappa_3 \Delta t \; y^n) \tag{20}$$

$$s^{n+1} = H_1 s^n + H_2 y^n + H_3 (s^n)^3 + H_4 \cos(n\omega\Delta t) \tag{21}$$

$$y^{n+1} = H_5 s^n + H_6 y^n + H_7 (s^n)^3 + H_8 (z^n)^3 + H_9 \cos(n\omega\Delta t) - H_{10}\sin(n\omega\Delta t) \tag{22}$$

$$\sin((n+1)\omega\Delta t) = H_{11}\sin(n\omega\Delta t) + H_{12}\cos(n\omega\Delta t) \tag{23}$$

$$\cos((n+1)\omega\Delta t) = H_{11}\cos(n\omega\Delta t) - H_{12}\sin(n\omega\Delta t) \tag{24}$$

The coefficients H_1 through H_{12} are

$$H_1 = 1 + \frac{\Delta t^2}{4} \quad , \quad H_2 = \kappa_3 \Delta t\,(1 - \mu\Delta t) \quad , \quad H_3 = -\frac{\kappa_1^2 \Delta t^2}{4}$$

$$H_4 = \frac{F_0 \Delta t^2}{2\kappa_1} \quad , \quad H_5 = \frac{\Delta t}{2\kappa_3}(1 - \mu\Delta t) \quad , \quad H_6 = 1 - 2\mu\Delta t + 2\mu^2\Delta t^2 + \frac{\Delta t^2}{4}$$

$$H_7 = -\frac{{\kappa_1}^3}{\kappa_2}\frac{\Delta t}{4}(1 - 2\mu\Delta t) \quad , \quad H_8 = -\frac{{\kappa_1}^3}{\kappa_2 {\kappa_4}^3}\frac{\Delta t}{4} \quad , \quad H_9 = \frac{F_0}{\kappa_2}\frac{\Delta t}{2}(1 - 2\mu\Delta t + \cos(\omega\Delta t))$$

$$H_{10} = -\frac{F_0}{\kappa_2}\frac{\Delta t}{2}\sin(\omega\Delta t) \quad , \quad H_{11} = \cos(\omega\Delta t) \quad , \quad H_{12} = \sin(\omega\Delta t) \tag{25}$$

If it is required that Δt be constant then the time dependent coefficients of Eq.(25) become constants for specific values of μ, F_0, κ_1 and κ_2. Equations (20) through (24) are the linear and nonlinear algebraic equations that approximate Duffing's equation and that must be modelled by the RANN.

For the sake of convenience the magnitude of all dependent variables are scaled to lie in the closed interval $[-1, 1]$. To satisfy this constraint on the magnitude of the dependent variables s^n, y^n and z^n it is required that

$$\kappa_1 > |x| \quad , \quad \kappa_2 > \left|\frac{dx}{dt}\right| \quad \text{and} \quad \kappa_4 \geq 1 + \Delta t \kappa_3 \quad , \quad \text{respectively.} \tag{26}$$

Using the results of section 2.2.2 the time dependent unknowns of the nonlinear set of algebraic equations can be approximated by the linear combination of transfer functions Υ with a user specified order of accuracy. Assuming ξ^n represents the dependent variables s^n, y^n, z^n, $\sin(n\omega\Delta t)$ and $\cos(n\omega\Delta t)$, an eighth-order accurate scheme was chosen for the approximation (${\xi_a}^n$) of the the linear and cubic terms, with $\theta = 0$ and assuming that $\Upsilon(-\xi) = -\Upsilon(\xi)$, $\Upsilon^{(1)}(0) \neq 0$ and $\Upsilon^{(3)}(0) \neq 0$.

$$\begin{aligned} {\xi_a}^n = \left(5\Delta w \ \Upsilon^{(1)}(0)\right)^{-1} \Big[& 8\Upsilon(\Delta w\xi^n) - 2\Upsilon(2\Delta w\xi^n) + (21)^{-1}(8)\Upsilon(3\Delta w \ \xi^n) \\ & -(28)^{-1}\Upsilon(4\Delta w\xi^n)\Big] + O\left(\Delta w^8\right) \end{aligned} \tag{27}$$

$$\begin{aligned} ({\xi_a}^n)^3 = \left(2520\Delta w^3 \ \Upsilon^{(3)}(0)\right)^{-1} [& -11683\Upsilon(\Delta w\xi^n) + 8738\Upsilon(2\Delta w\xi^n) \\ & -(2)^{-1}(4869)\Upsilon(3\Delta w \ \xi^n) + (3)^{-1}(1261)\Upsilon(4\Delta w\xi^n) \\ & -(12)^{-1}(410)\Upsilon(5\Delta w\xi^n)\Big] + O\left(\Delta w^8\right) \end{aligned} \tag{28}$$

Since the updated dependent variables must each exit a single node in the second layer, the following approximation was used:

$$m{\xi_a}^{n+1} \quad = \quad m\left(\Delta w \ \Upsilon^{(1)}(0)\right)^{-1} \Upsilon(\Delta w \ \xi^{n+1}) + O\left(\Delta w^2\right) \quad . \tag{29}$$

where m is an integer constant.

Using these approximations with $\Upsilon(\xi) = \tanh(\xi)$, Eqs. (20) through (24) become

$$m_1 \Delta w {s_a}^{n+1} \quad = \quad m_1 \tanh\left[\Delta w \ H_1 {s_a}^n + ... + \Delta w \ H_4 \ \cos_a(n\omega\Delta t)\right] \tag{30}$$

for $m_1 = 1, ..., 5$,

$$m_2 \Delta w {y_a}^{n+1} \quad = \quad m_2 \tanh\left[\Delta w \; H_5 {s_a}^n + ... - \Delta w \; H_{10} \; \sin_a(n\omega\Delta t)\right] \tag{31}$$

for $m_2 = 1, ..., 4$,

$$m_3 \Delta w {z_a}^{n+1} \quad = \quad m_3 \tanh\left[\Delta w \; {s_a}^n + \kappa_3 \Delta t \; {y_a}^n\right] \quad \text{for } m_3 = 1, ..., 5 \; , \tag{32}$$

$$m_4 \Delta w \; \sin_a((n+1)\omega\Delta t) \quad = \quad m_4 \tanh\left[\Delta w \; H_{11} \; \sin_a(n\omega\Delta t) + \Delta w \; H_{12} \; \cos_a(n\omega\Delta t)\right] \tag{33}$$

for $m_4 = 1, ..., 4$ and

$$m_5 \Delta w \; \cos_a((n+1)\omega\Delta t) \quad = \quad m_5 \tanh\left[\Delta w \; H_{11} \; \cos_a(n\omega\Delta t) - \Delta w \; H_{12} \; \sin_a(n\omega\Delta t)\right] \tag{34}$$

for $m_5 = 1, ..., 4$.

Figure 2 is a sketch of the fully connected two-layer, twenty two node, RANN given by Eqs. (30) through (34), which are third-order accurate. An initial time $t^0 = 0$ is assumed.

3 RESULTS

The motion of the oscillator can be characterized by the number of times (k) a cycle of oscillation is completed in one period T of the forcing function about the total number of equilibrium states of the system. Such an oscillation is known as a global-period-k oscillation. If the oscillator completes k cycles of motion in one period of excitation T about a number of equilibrium states less than the total number, the oscillator is said to have completed a local-period-k oscillation, or simply a period-k oscillation. With these definitions in mind one can immediately classify the observed motions of the Duffing's oscillator (Fig. 3). Figure 3 is assembled from works of Pezeshki and Dowell [21] and Masoud and Asfar [10] for the damping value and excitation amplitude of $\mu = 0.084$ and $F_0 = 0.178$, respectively.

The RANN of Fig. 2 was constructed with the previously mentioned values of μ and F_0 and run for the same forcing frequency range ($\omega = 0.633$ to $\omega = 1.02$) illustrated in Fig.

3. Previous results of the two well potential system [10] indicate a maximum displacement between ± 2.0 and maximum velocity between ± 1.0. As a result $\kappa_1 = 4.0$ and $\kappa_2 = 2.0$. Initial conditions were set for $x(0) = 1.0$ and $\dot{x}(0) = 0.0$. To ensure that steady-state solutions were displayed, approximations were run for 900 periods, T, based on the forcing frequency ω ($\mathrm{T} = 2\pi/\omega$). Throughout the simulation the time step ($\Delta t = \mathrm{T}/200$) used in the RANN was half that used in the Masoud and Asfar study. All computations were done in single precision with $\Delta w = 1 \times 10^{-4}$ and all test cases were run in less than 40 seconds on a Sparc 20 workstation.

3.1 RANN Simulation Results

Figures 4 - 11 compare the constructed RANN output ($x = \kappa_1 s$ and $\dot{x} = \kappa_2 y$) with the numerical results of Masoud and Asfar. For clarity, independent plots of position time histories and velocity time histories have been excluded. In their place, phase space plots, as well as Poincaré maps are shown and compared. In Figs. 4 - 11 the results of Masoud and Asfar are shown on upper row while the those from the RANN are shown on the lower.

Figures 4 and 5 compare the $\dot{x}$ vs. x phase space trajectory of the oscillator over time as the forcing frequency ω decreases. Starting with $\omega = 1.02$ at which the period of the oscillator is equal to 1, i.e. period-1 oscillation, the period of the oscillator doubles as ω decreases, so period-2, period-4, and period-8 oscillations are observed until the system evolves into a global-period-5 motion at which the system oscillates between the two equilibrium states $(\dot{x}, x)$ which are $(0, -1)$ and $(0, 1)$. Note in Fig. 5 that $x(0) = 0$ and $\dot{x}(0) = 1.0$ were used to display the coexisting global attractor.

Figures 6 and 7 show the transition from the symmetric global-period-5 motion ($\omega = 0.957$) to the asymmetric global-period-5 motion ($\omega = 0.877$), to the first onset of chaotic oscillation ($\omega = 0.870$).

Figure 8 describes the first region of chaotic oscillations starting at $\omega = 0.865$ and ending close to $\omega = 0.780$. As the forcing frequency decreases further, the system evolves from chaotic motion into a new global-period-3 motion at $\omega = 0.75$, as can be seen in Fig.

9. Between $\omega = 0.738$ and $\omega = 0.634$ the oscillator enters a new region of chaotic motion as seen in Fig. 10 until at $\omega = 0.633$, a new period-1 oscillation appears (Fig. 11).

For almost all values of ω treated in this study, results from the constructed RANN compared well to the numerical results of Masoud and Asfar. However, mis-matches were present in Fig. 4 and Fig. 11. Specifically, Fig. 4(c) shows that the RANN simulation of $\omega = 0.963$ is almost identical to that given in the reference except for what seems to be a rotation of the figure about the horizontal axis. Since most of the other results were satisfactory, we suspect a minor error in the reference. Also, though the RANN simulation of $\omega = 0.633$ in Fig. 11(b) was able to produce and match the stable phase plane trajectory for positive displacement, it was unable to find the stable phase plane trajectory for negative displacement with any combination of initial displacement and velocity.

3.2 Stability

Since the MacCormack method used for the temporal integration is conditionally stable there is a limit at which the value of Δt will be too large and result in complete instability of the constructed RANN. The stability criterion is dependent on the system to be solved, and so in the case of Duffing's equation it is a function of the parameter values. Figure 12 shows the effect of increasing the value of Δt from 0.0616 to 0.6776 for $\omega = 1.020$ and $\Delta w = 1 \times 10^{-4}$. Note that the simulation changes from periodic motion to one resembling chaos even though at $\omega = 1.020$ Duffing's equation should only give periodic motion. In effect, as Δt increases the RANN is responding to numerical error and appears to be simulating an incorrect, though stable, system. For $\omega = 1.020$, the limiting value of Δt for which the RANN becomes completely unstable was found to be approximately 0.73. Similar stability considerations also exist for Δw.

This information is included primarily for those wishing to use the RANN with supervised learning for the emulation of a chaotic dynamic system. The user should be concerned with the value of Δt at which data is being taken for training, as the RANN may display numerical error rather than an accurate simulation of the dynamic system. Higher order explicit time marching techniques could be used in this case (with an increase in the number of nodes in the RANN).

4 CONCLUSIONS AND FUTURE WORK

The adaptive data-driven emulation and control of mechanical systems are popular applications of recurrent artificial neural networks in engineering. However, because RANN training is a nonlinear optimization problem it is difficult to apply this attractive modelling tool to engineering applications. As a remedy, a new initialization method was proposed for connectionist training algorithms. This proposed initialization would be obtained from accurate solutions of the time-dependent differential equations, using only the equations of interest and the initial conditions. However, the proposed network initialization method is critically dependent on demonstrating that a conventional RANN, using nonlinear transfer functions, can accurately approximate a physical model in the form of differential equations and that the accuracy of the constructed RANN can be controlled by the user.

By theory and numerical example, it has been shown that it is possible to construct a conventional RANN from commonly used components and select the connection pattern and values of connection weights to approximate the solution to a time-dependent nonlinear ordinary differential equation. This RANN construction method is based on reducing the differential equation to a series of linear and nonlinear algebraic equations, forming global polynomial bases from linear combinations of smooth transfer functions that can be expanded in a Taylor series, and then mapping the polynomial bases into a network architecture.

Specifically, it has been shown that a system of coupled differential equations, describing periodic and chaotic motion, could be reduced to a set of nonlinear recurrent algebraic equations utilizing the explicit MacCormack finite difference scheme. The set of nonlinear recurrent algebraic equations was then represented in a RANN architecture, using hyperbolic transfer functions, whose accuracy of approximation can be controlled by the user through the connection weights.

The authors chose to map the algebraic equations, rather than the explicit differential equations, because it is thought that this will give greater utility to the engineer. The more popular numerical techniques, such as finite difference and finite elements, reduce differential equation to algebraic form for use on digital computers. As a result, the engineer can use these familiar and well understood techniques in the process of programming the RANN.

Since it has been shown that it is possible to describe the operation of a RANN with explicit finite difference techniques, and in light of the extensive literature that exists on finite difference, it should be possible to continue work along these lines to better understand the stability and convergence properties of RANNs in general. Work will also continue on increasing the RANN approximation accuracy of variables exiting the output layer, modelling time dependent partial differential equations and investigating the RANN initialization technique's effect on conventional RANN training.

5 ACKNOWLEDGEMENTS

This work was supported under NASA Grant NAG 9-719 and ONR grant N00014-95-1-0741. The authors would like to thank Robert O. Shelton of the NASA Johnson Space Center and Boris Zeldin of Rice University for their many helpful suggestions.

REFERENCES

[1] Rumelhart, D. E. and McClelland, J. L. (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* (MIT Press, Cambridge MA) pp. 318-362.

[2] Williams, R. J. and Zipser, D. (1989) A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, **1**, pp. 270-280.

[3] Pineda, F. J. (1989) Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation, *Neural Computation*, **1**, pp. 161-172.

[4] Saarinen, S., Bramley, R. and Cybenko, G. (1993) Ill-Conditioning in Neural Network Training Problems, *SIAM J. Sci. Comput.*, **14**, No. 3, pp. 693-714.

[5] Igelnik, B. and Pao, Y.-H. (1995) Stochastic Choice of Basis Function in Adaptive Function Approximation and the Function-Link Net, *IEEE Transactions on Neural Networks*, **6**, No. 6, pp. 1320-1329.

[6] Schidt, W.F., Raudys, S., Kraaijveld, M.A., Skurikhina, M. and Duin, R.P.W. (1993) Initializations, Back-Propagation and Generalization of Feed-Forward Classifiers, Proceedings of IEEE International Conference on Neural Networks, ICNN-93, Vol. 1, pp. 598-604.

[7] Poggio, T. and Girosi, F. (1990) Networks for Approximation and Learning, Proceedings of the IEEE, **78**, No. 9, pp. 1481-1497.

[8] Canto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A. (1988) *Spectral Methods in Fluid Dynamics*, (Springer-Verlag, Berlin Heidelberg), pp. 31-75.

[9] Box, G.E.P. and Draper, N.R. (1987) *Empirical Model-Building and Response Surface* (Wiley, New York).

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

[11] Hecht-Nielsen, R. (1990) *Neurocomputing* (Addison-Wesley, Inc., New York), pp. 183-191.

[12] Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, **2**, pp. 359-366.

[13] Hornik, K., Stinchcombe, M. and White, H. (1990) Universal Approximation of an Unknown Mapping and its Derivatives Using Multilayer Feedforward Networks, *Neural Networks*, **3**, pp. 551-560.

[14] Ito, Y. (1991) Representation of Functions by Superpositions of a Step or Sigmoidal Function and Their Applications to Neural Network Theory, *Neural Networks*, **4**, pp. 385-394.

[15] Leshno, M., Lin, V.Y., Pinkus, A. and Schocken, S. (1993) Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function, *Neural Networks*, **6**, pp. 861-867.

[16] Mhaskar, H.N. (1995) Neural Networks for Optimal Approximation of Smooth and Analytic Functions, *Neural Computation*, **8**, pp. 164-177.

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

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

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

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

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

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

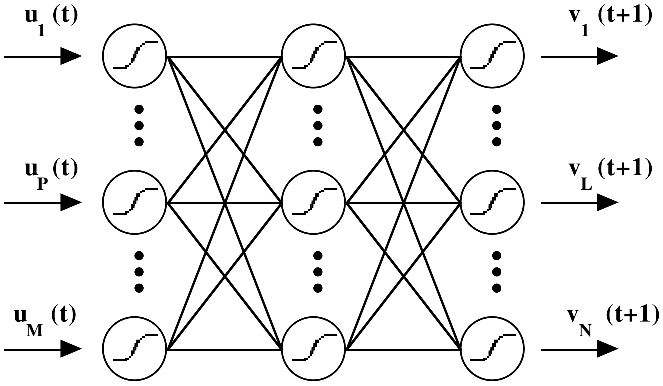

Figure 1: General RANN architecture.

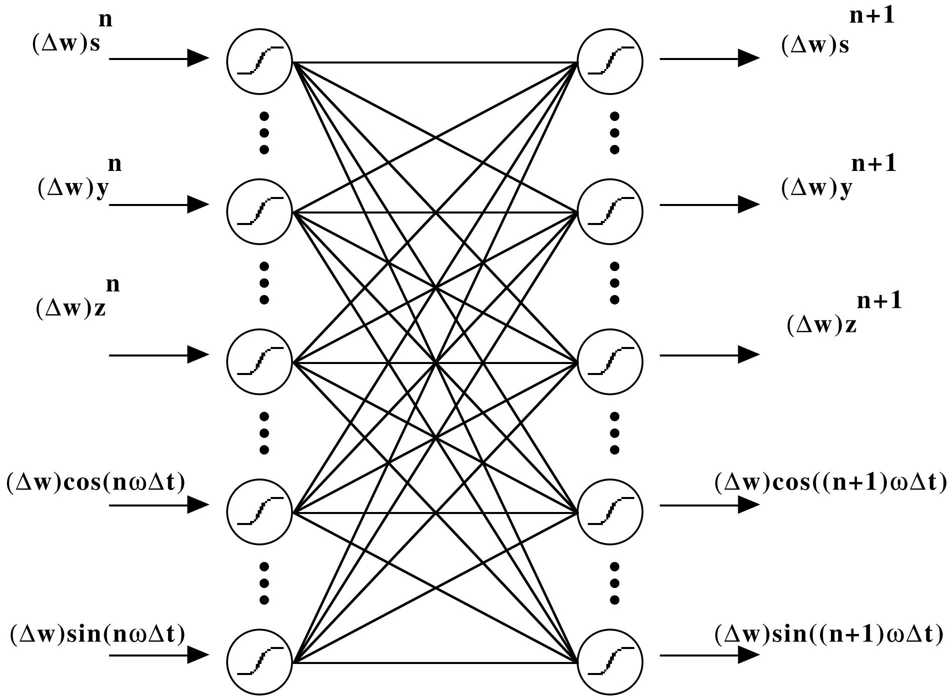

Figure 2: Complete two-layer twenty two node RANN assembly from the network construction scheme applied to Duffing's equation.

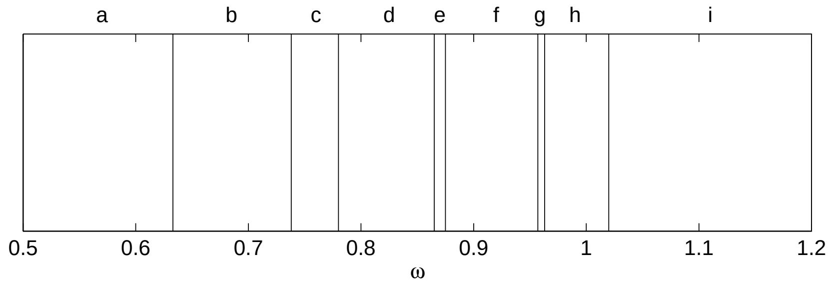

Figure 3: Classification of frequency response for the inhomogenous Duffing's equation: (a) and (i) period-1 motion; (b) second chaotic region; (c) evolution of period-3 attractor into chaos; (d) first chaotic region; (e) period doubling bifurcations in the period-5 attractor; (f) bifurcations from the symmetric period-5 attractor to asymmetric one; (g) coexistence of the period-5 attractor; (h) period doubling bifurcations.

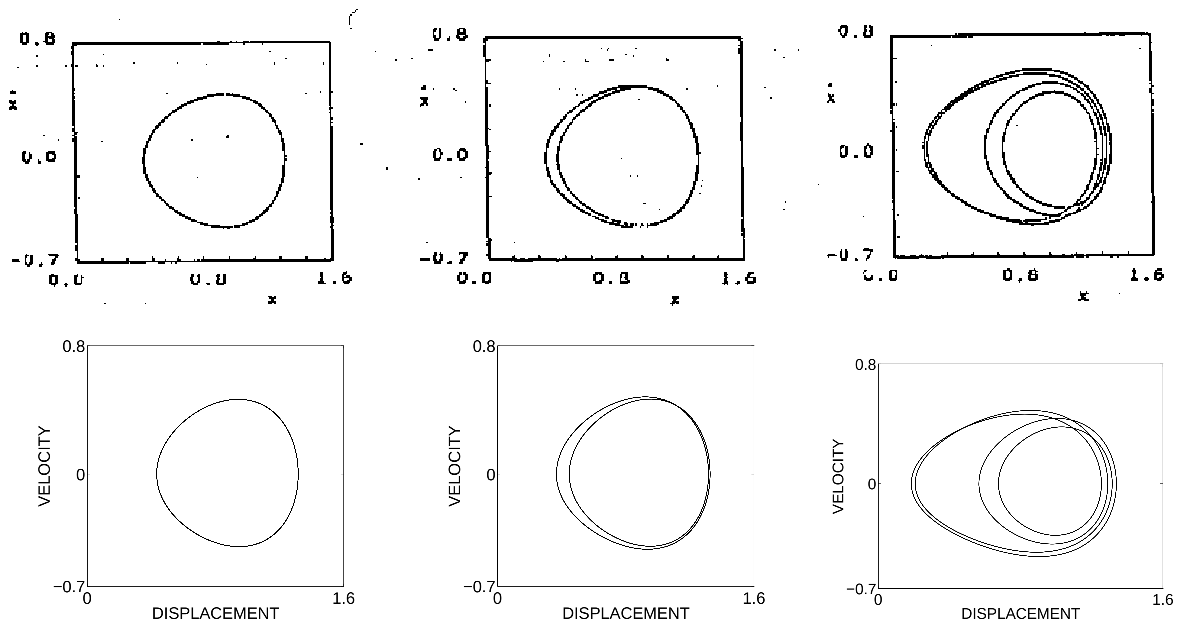

Figure 4: Phase space trajectories of the forced two well potential oscillator. From left to right: (a) $\omega = 1.02$ period-1, (b) $\omega = 1.0$ period-2, (c) $\omega = 0.963$ period-4.

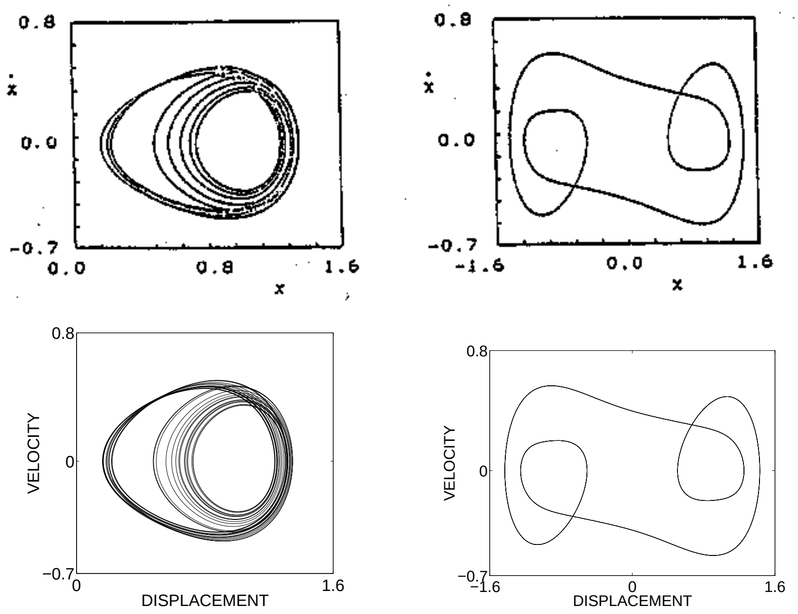

Figure 5: Phase space trajectories of the forced two well potential oscillator. From left to right: coexisting (a) $\omega = 0.957$ period-8, (b) $\omega = 0.957$ global-period-5.

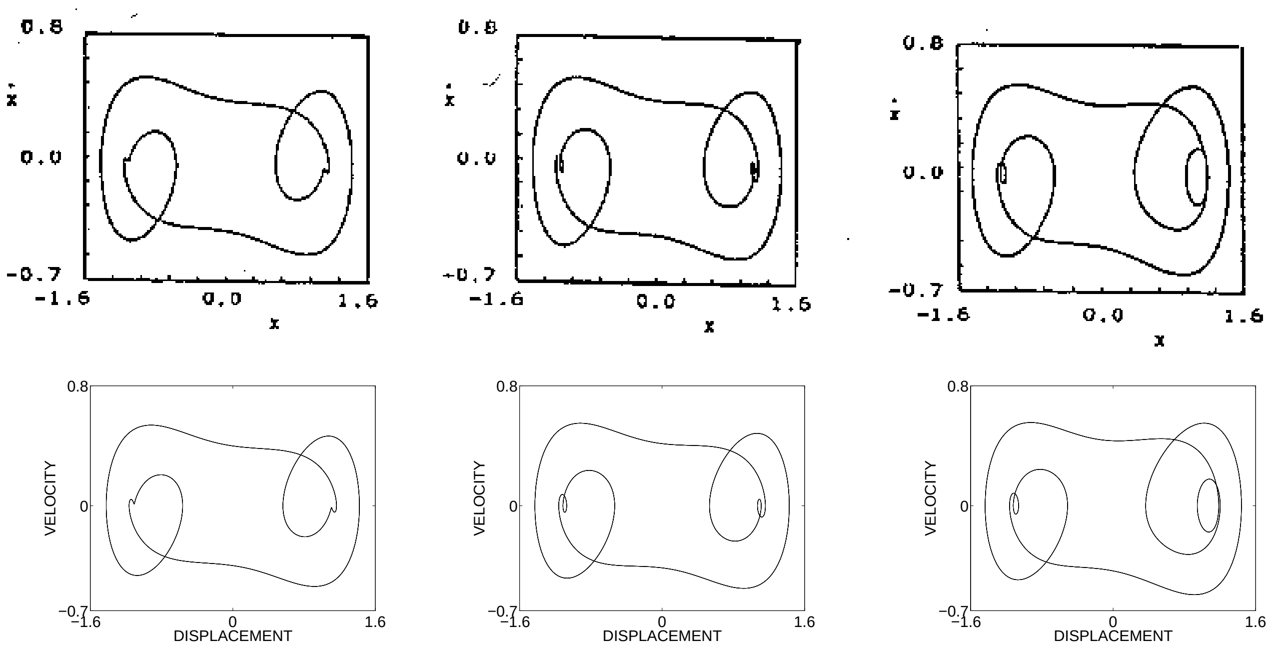

Figure 6: Phase space trajectories of the forced two well potential oscillator. From left to right: (a) $\omega = 0.91$ symmetric global-period-5, (b) $\omega = 0.9$ symmetric global-period-5, (c) $\omega = 0.877$ asymmetric global-period-5.

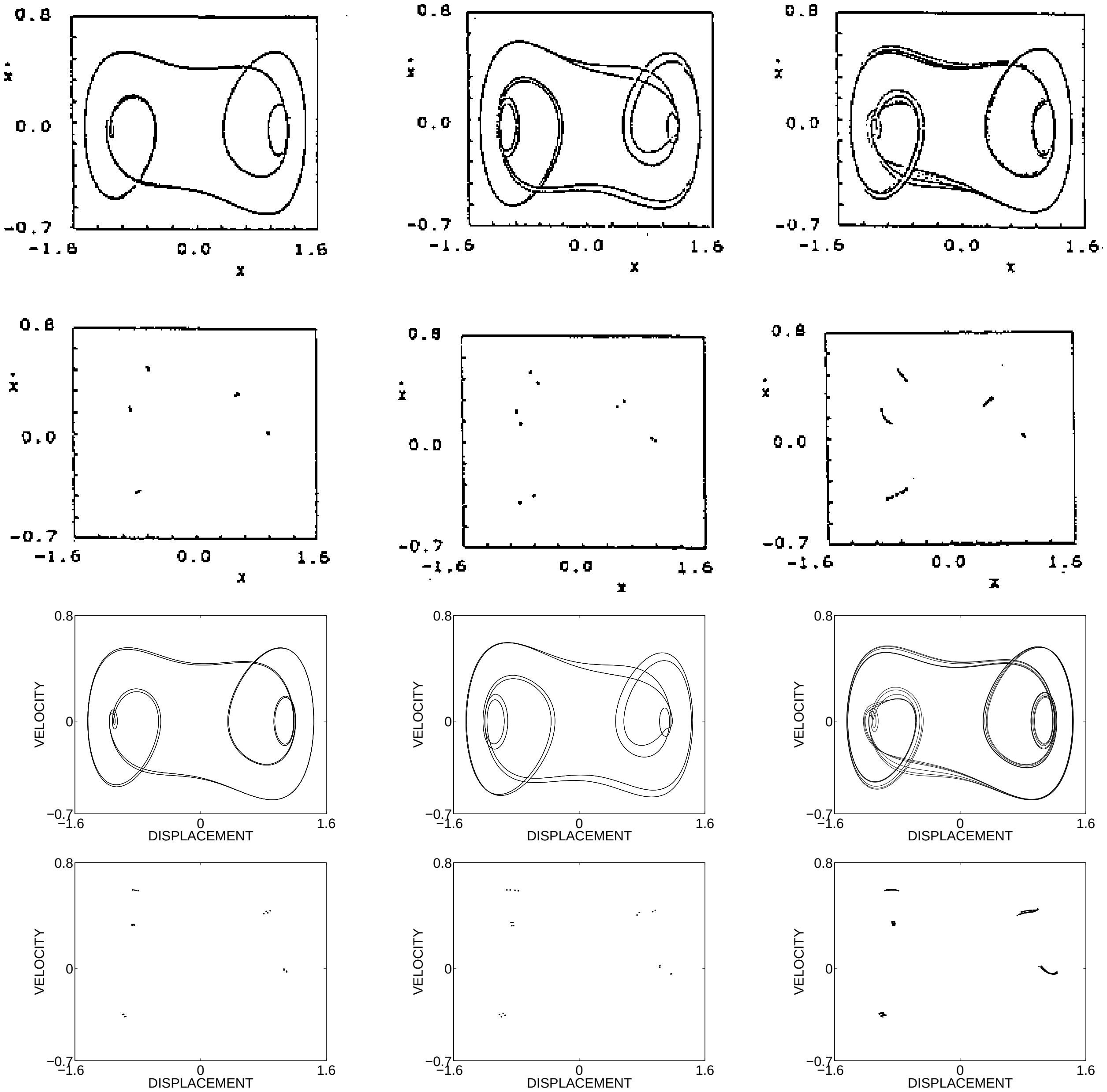

Figure 7: Phase space trajectories and Poincaré maps of the forced two well potential oscillator. From left to right: (a) $\omega = 0.875$ asymmetric global-period-5, (b) $\omega = 0.871$ asymmetric global-period-10, (c) $\omega = 0.87$ onset of chaos.

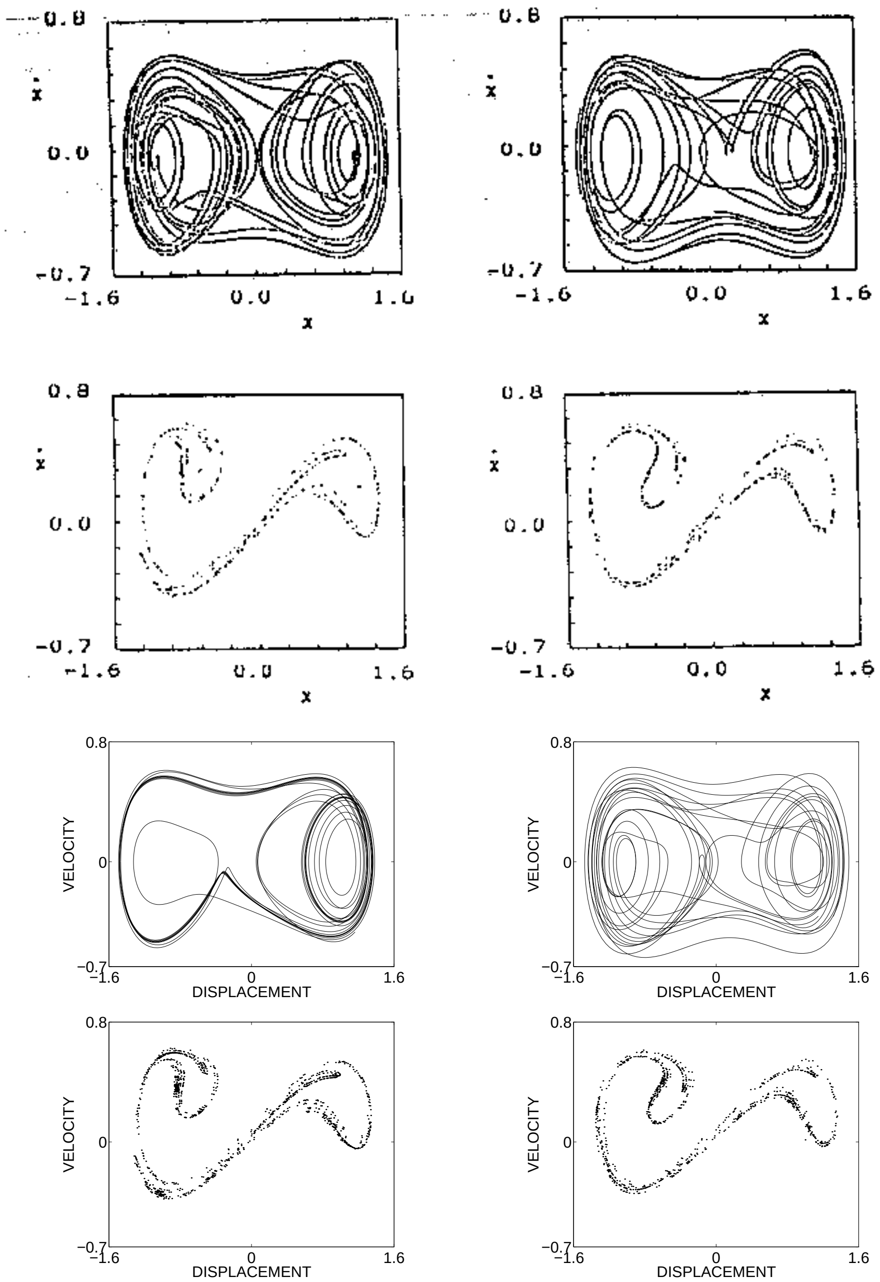

Figure 8: Phase space trajectories and Poincaré maps of the forced two well potential oscillator in the first chaotic region. From left to right: (a) $\omega = 0.865$ chaotic higher frequency boundary, (b) $\omega = 0.78$ chaotic lower frequency boundary.

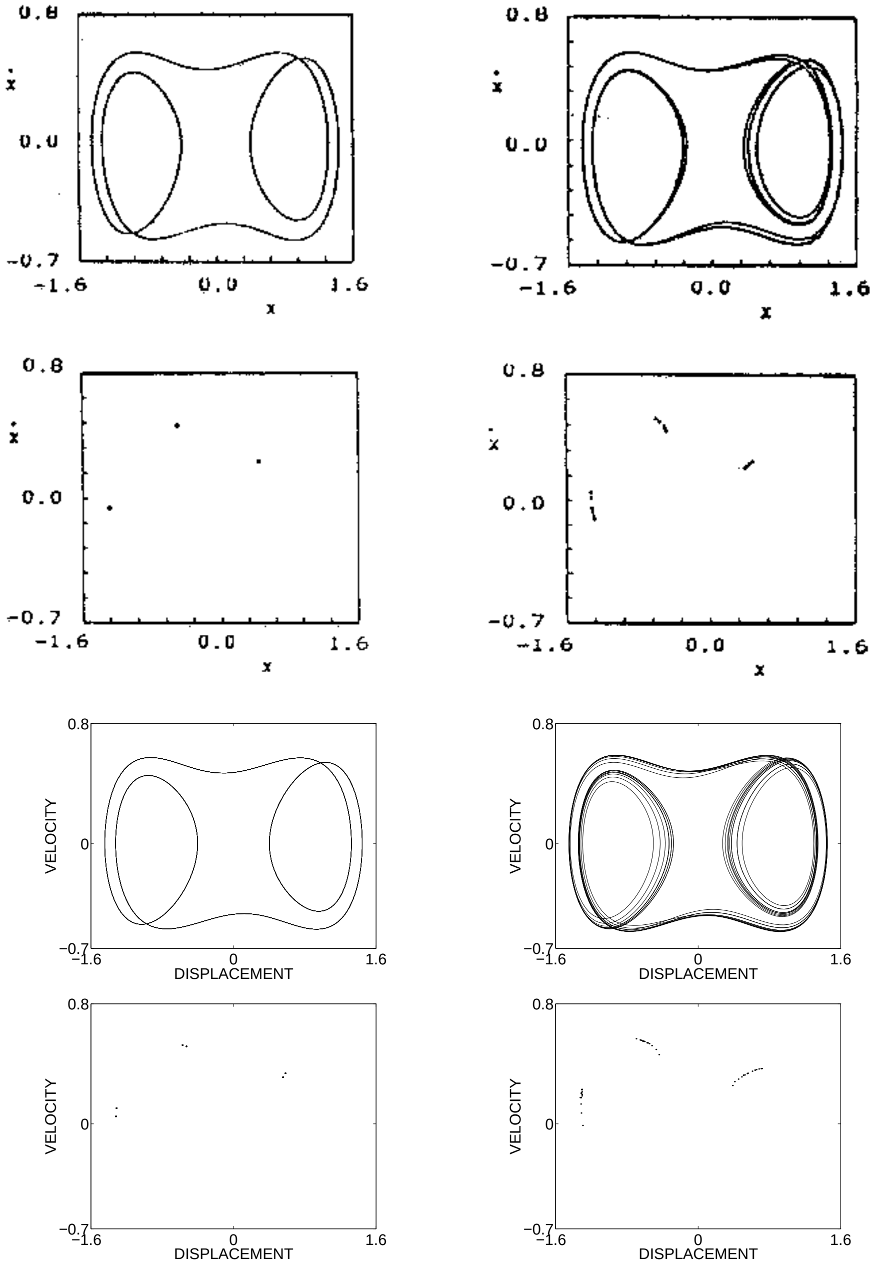

Figure 9: Phase space trajectories of the forced two well potential oscillator. From left to right: (a) $\omega = 0.75$ period-3, (b) $\omega = 0.74$ onset of chaos.

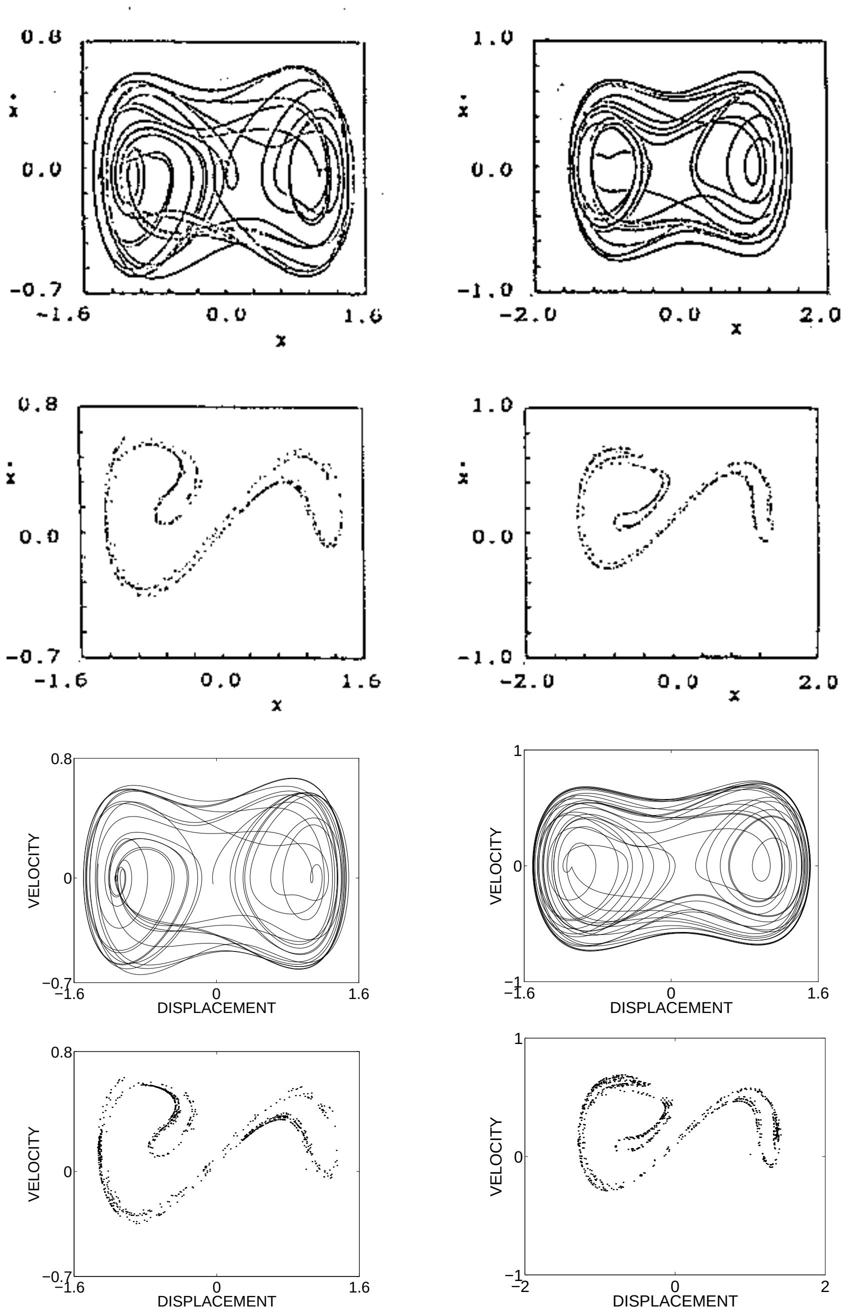

Figure 10: Phase space trajectories and Poincaré maps of the forced two well potential oscillator in the second chaotic region. From left to right: (a) $\omega = 0.738$ chaotic higher frequency boundary, (b) $\omega = 0.634$ chaotic lower frequency boundary.

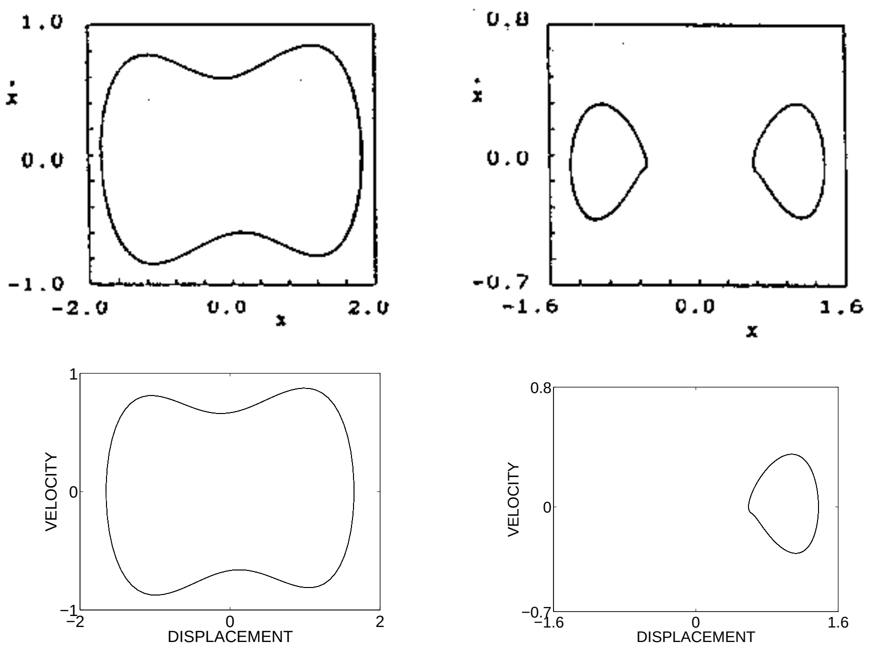

Figure 11: Phase space trajectories of the forced two well potential oscillator. From top to bottom: (a) $\omega = 0.633$ coexisting period-1 oscillations both global and local, (b) $\omega = 0.633$ local-period-1 oscillation.

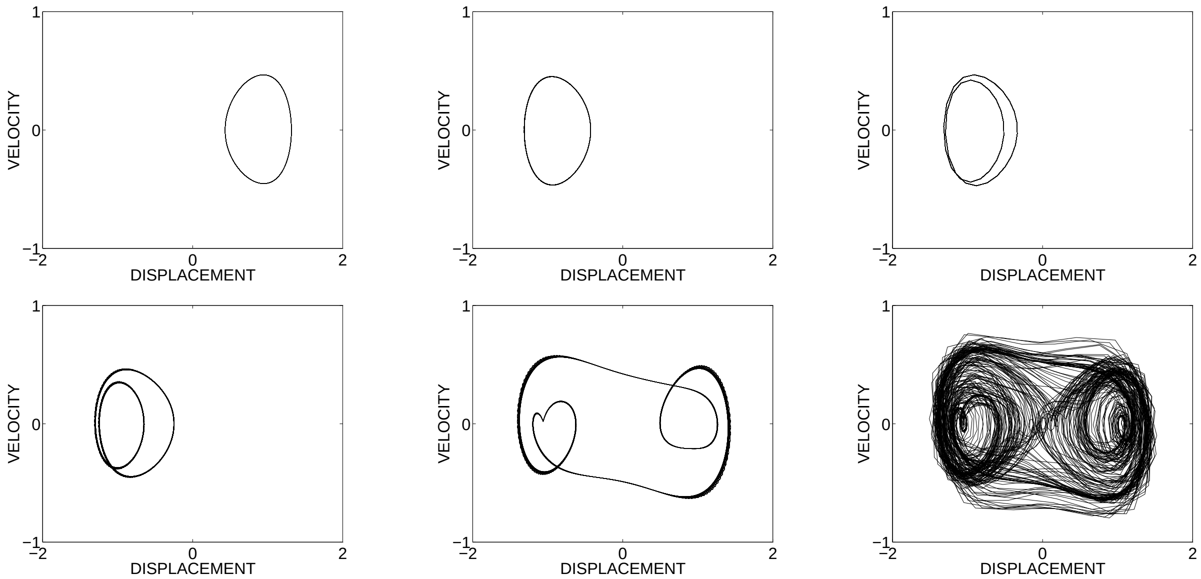

Figure 12: Phase space trajectories illustrating the solution dependence on the time step Δt at $\omega = 1.02$. From left to right and top to bottom: Δt=0.0616, 0.1848, 0.3080, 0.4312, 0.5544, and 0.6776.