Neural Comput & Applic (1997)6:102–115

Neural Computing & Applications

Adaptive Neuro-Genetic Control of Chaos Applied to the Attitude Control Problem*

D.C. Dracopoulos and A.J. Jones

Department of Computer Science, Brunel University, London, Uxbridge, Middlesex, UK and Department of Computer Science, University of Wales, Cardiff, UK

Conventional adaptive control techniques have, for the most part, been based on methods for linear or weakly non-linear systems. More recently, neural network and genetic algorithm controllers have started to be applied to complex, non-linear dynamic systems. The control of chaotic dynamic systems poses a series of especially challenging problems. In this paper, an adaptive control architecture using neural networks and genetic algorithms is applied to a complex, highly nonlinear, chaotic dynamic system: the adaptive attitude control problem (for a satellite), in the presence of large, external forces (which left to themselves led the system into a chaotic motion). In contrast to the OGY method, which uses small control adjustments to stabilize a chaotic system in an otherwise unstable but natural periodic orbit of the system, the neuro-genetic controller may use large control adjustments and proves capable of effectively attaining any specified system state, with no a priori *knowledge of the dynamics, even in the presence of significant noise.*

Keywords: Adaptive control; Attitude control; Chaotic systems; Neurogenetic; Prediction

1. Introduction

The aim of this work is to demonstrate applications of techniques from neural networks and genetic algorithms to specific problems of adaptive nonlinear control. In a previous paper [1], we gave various examples of neuromodels used to make short range predictions for smooth dynamic systems of various complexities. The underlying principle used to construct these models is to train networks to model an input–output map of the form

$$(x(t), \dot{x}(t), \ldots, x(t - p\Delta t), \dot{x}(t - p\Delta t)) \rightarrow (x(t + \Delta t), \dot{x}(t + \Delta t)) \quad (1)$$

for some fixed $p \geq 1$, where $x(t)$ is the state of the system at time t, and the interval Δt is appropriately chosen. Thus, the model has as inputs the current state, and a number p of past states, and attempts to predict the state of the system a short time ahead. We call a neural network which acts in this way a *Locally Predictive Net* (LPN). The critical property of such models is the fact that, although trained on sample points from a *single* trajectory, the network is capable of accurate generalization on other, hitherto unseen, trajectories.

Building on the effectiveness of short range prediction using neuro-models, we recently proposed an adaptive *neuro-genetic* control architecture, a hybrid method of adaptive control using neural networks for modelling and genetic algorithms for control [2]. The neuro-genetic control system was demonstrated with reference to the *attitude control* problem for a rigid body (satellite) equipped with thrusters about each principal axis in the case of large slew angles, unknown dynamic characteristics, and stable and unstable target states.

This technique was designed to be applied to the adaptive control of smooth nonlinear dynamic systems. In [2], the neuro-genetic control architecture was applied to the adaptive satellite attitude

*This work was partly supported by SERC grant 90800355.

Correspondence and offprint requests to: D.C. Dracopoulos, Department of Computer Science, Brunel University, London, Uxbridge, Middlesex UB8 3PH, UK.

control problem in various cases where no external forces act upon the system (including the case where the system is in a limit cycle). Control of many nonlinear systems poses serious difficulties for conventional control techniques, since most are applicable to linear systems, weakly nonlinear systems, or require various assumptions about the plant dynamics [3,4]. Other problems of conventional control techniques appear when addressing physical effects such as friction, backlash, torque non-linearity (especially dead zone), high-frequency dynamics, and sensor noise [5]. Traditional adaptive control theory is based mainly around the model reference adaptive control method [6–9]. Despite the above, one must not underestimate or ignore conventional control (and adaptive control) theory, since it offers perhaps the only way of providing theoretical stability guarantees for the control of certain classes of nonlinear (even chaotic) dynamic systems (for example, see Refs [10–13]). This work and other work in the neurocontrol area aims to achieve control of dynamic systems for which existing control theory finds difficulties. However, for the time being, most of these new techniques cannot guarantee stability with theoretical proof and are based in simulation results. One of the exceptions is the work of Narendra of Yale University in neural adaptive control. His work is based in the classical adaptive control theory, but instead of matrices he uses neural networks [14–16]. In this paper, the hybrid neuro-genetic method introduced in [2] is further tested and evaluated for the control of *chaotic* dynamic systems.

Due to their extreme sensitivity to small perturbations, chaotic dynamic systems pose radical control system design problems, and the necessity of controlling such systems is usually avoided wherever possible. However, avoiding the problem is not always feasible (e.g. [17]); moreover, it has also been argued that in some situations designing a system to be chaotic might confer positive benefits [18].

The outline of this paper is as follows. Following the introduction, Sect. 2 gives a brief description of the neuro-genetic control architecture introduced in [2]. In Sect. 3 this is contrasted with the OGY method, one of the principal techniques used to control chaotic systems. Section 4 describes the satellite attitude control problem in a test case where perturbing torques induce chaotic motion and results for this case are presented in Sect. 5. Section 6 includes results for a more difficult test case in which chaotic perturbations are combined with the introduction of noise into the sensor system of the satellite. In Sect. 7 some aspects of the choice of objective function are discussed.

In all the examples presented, it is assumed that the dynamics are initially unknown to the controller which is required to adapt to the observed responses to control signals.

2. The Adaptive Neuro-Genetic Control Architecture

One major problem with the various schemes for direct inverse control using neural networks is that the inverse kinematic problem is frequently ill posed, i.e. there are many control inputs which might produce a given *sensed response*. In general, if any plant state has an inverse image that is a non-convex region in control space, then there exist sets of samples in the region whose average is outside the region. Therefore any learning procedure for an *inverse net* whose effect is to average over conflicting control signals (network outputs) for a given system state (network inputs) would be incorrect. Whilst such methods of *direct* learning inverse kinematics may be very effective in particular situations they cannot stand as a general methodology [3,19].

Jordan and Rumelhart [19] used a different type of inverse control to face the problem of non-uniqueness of inverse models. In their approach, they first train a forward model of the plant (able to give an estimation $\hat{y}[t]$ of the actual output $y[t]$ of the plant) and then they train a neural network controller by propagating back the errors $y^*[t] - y[t]$ of the system (where $y^*[t]$ is the desired response) through the forward model. In their work, when they deal with plants that are required to follow a particular trajectory, they use a modified algorithm, equivalent to that of backpropagation-through-time [20,21]. A similar approach was taken by Kawato et al. [22–24] and Psaltis [25]. Generally, however, it is not always possible to reach a desired next state, from any other dynamic system state (unless some planning is involved), and therefore their methods are not generally applicable to control problems. In particular, such an approach cannot be applied directly (although it can be modified) to the attitude control problem where the goal is to reach a target state starting from an initial state.

Another potential pitfall is that, even for smooth dynamic systems, suitable solutions to the inverse kinematics may be *discontinuous* functions of time. This is illustrated by many of our simulation results for the attitude control problem. If the initial angular velocities are large the controller tends to produce very rapidly varying control torques, despite the fact that the progression towards the goal state is quite smooth. As the angular velocities become smaller the control torques tend to become much smoother. Whilst these discontinuities do not make impossible for a neural network to learn such a functional transformation, they do make it very much harder (and more time-consuming) to efficiently train such a network.

For this reason we proposed [2] an alternative two part architecture. This paper does not give the full details of the algorithm, as these can be found elsewhere [2]. However, below we describe the main operation of the hybrid neural and genetic architecture.

The first part of the architecture is a LPN neuromodel that learns to predict a future system state given the few last states and control inputs (Fig. 1). Earlier results on neuromodelling [1,26], and the example given in [2], indicate that this is a realistic prospect. The time scale for such a prediction might be as high as 0·5–1 s.

With this predictive capability we can now evaluate a large number of hypothetical control inputs and select the best. This is not done directly, but by using a genetic algorithm to explore the space of hypothetical control inputs at any given moment and the LPN predictor to evaluate any member of the current population (of hypothetical control inputs).

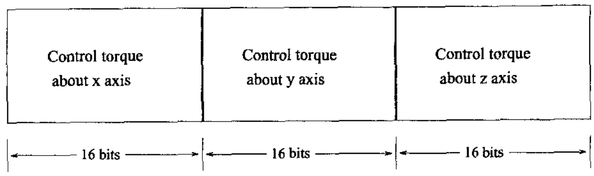

Fig. 2. The individual chromosomes for the Genetic controller.

Normally, the evaluation phase (computing the fitness of a member of the population) is the most time consuming aspect of the genetic algorithm, but a hardware neural network can be used for this phase (or as new much faster learning algorithms appear, these can be used), thus allowing the evaluation of many thousands of sets of control inputs over the relevant prediction interval which may be a significant fraction of a second.

Of itself the GA is very simple (see Fig. 3), and is described in full detail in [2]. The control signals are represented by binary strings (Fig. 2), with simple bit string manipulations for crossover and mutation. A variety of implementations could be used for the GA, ranging from execution by a serial

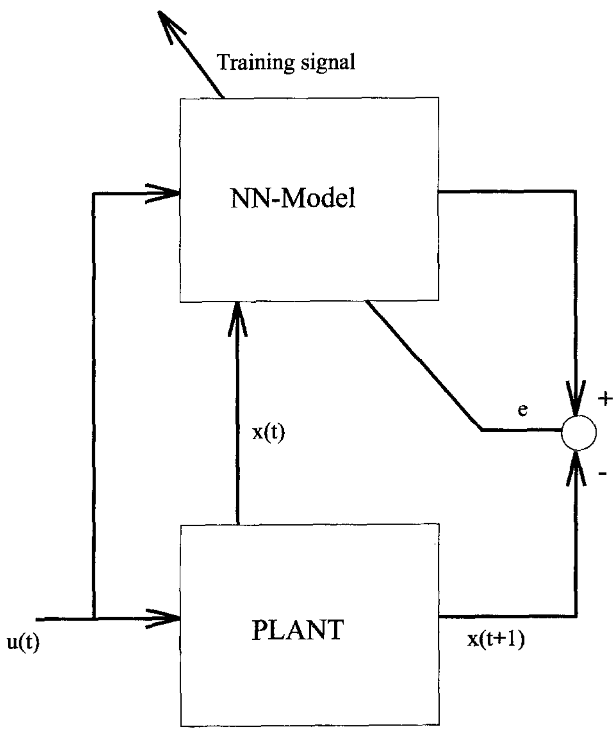

Fig. 1. Building the neuromodel for the plant. $x(t)$ represents the state of the plant at time t, $u(t)$ if the control inputs vector to the plant, and e is the error between plant state and predicted model state at time $t + 1$.

```
population_size := 50;
generation := 1;
Initialize population with random binary strings;
while generation ≤ 100 do
    Find the two best individuals of current population;
    Copy the two best individuals to new population;
    for i = 1 to population_size − 2 step 2 do
        Select two individuals based on fitness;
        Probabilistically mutate the two individuals;
        Probabilistically perform crossover;
        if crossover_performed then
            Copy the two offspring into new population;
        else
            Copy the two (mutated) individuals into new
            population;
        endif
    endfor
    generation := generation + 1;
endwhile
```

Fig. 3. Pseudocode of the genetic algorithm for the attitude control problem.

processor to gate arrays with additional memory and processors (to provide a stack, for example). Given that the evaluation time per member of the population is very fast, the rest of the GA is quite simple. The particular implementation chosen would depend heavily on the system to be controlled and the speed of events in the real world.

We call this mixture of techniques a *neuro-genetic control architecture*. The learning phase of the neural network can be likened to basic mastery of motor control to the extent that one can predict the immediate consequences of any given set of actions. The genetic component of the system is analogous to a series of 'mind experiments', using many of these predictions to choose an actual set of control signals.

The sequence of events is thus (see Fig. 4), $t = k\Delta t$ (with $\Delta t = 0{\cdot}01$ in the simulations):

At step $k - 1$: controller applies $u^*(k - 1)$—LPN predicts $x^q(k)$.

At times between $k - 1$ and k: based on this prediction, the genetic algorithm tries to choose $u^*(k)$ so as to optimize $x(k + 1)$. The genetic algorithm uses the LPN to predict the result $x^q(k + 1)$ of hypothetical control inputs $u^h(k)$, and hence evaluate the fitness of $u^h(k)$.

At step k: controller applies $u^*(k)$. LPN predicts $x^q(k + 1)$.

. . .

3. Chaotic Systems and the OGY Method

An entirely different method for controlling chaos is based on recent work by Ott, Grebogi and Yorke [18] (OGY). This method differs in significant respects from the neuro-genetic control architecture

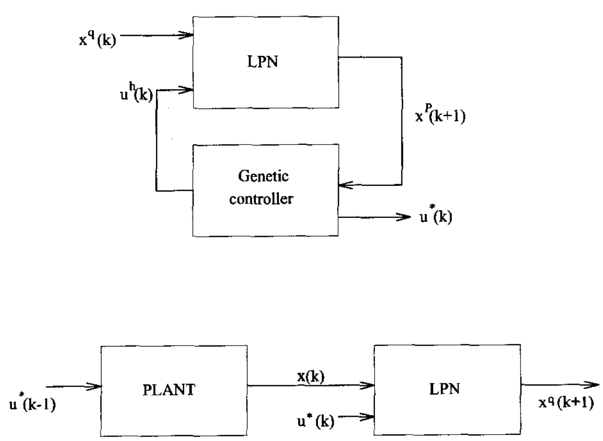

Fig. 4. The adaptive neuro-genetic architecture.

described above. A full review of the OGY method and other related techniques to control chaos are given in Ref. [27], but it seems worthwhile to briefly consider the differences between our method and the OGY algorithm.

In our earlier work on neuromodelling [1], input–output models such as (1) were constructed from observations along a trajectory of the system. The success of these neuro-modelling results almost certainly arises from the ability of a parametric model of the type (1) to capture the invariant features of the dynamics of the system rather than any particular properties of neural networks *per se*. In these models, all variables of the dynamic system were used to construct the models.

The OGY method is based on the key observation that a chaotic attractor typically has embedded within it an infinite number of unstable periodic orbits. To apply the method one must make the following assumptions [18,28]:

1. The dynamics of the system can be represented as derived from an n-dimensional nonlinear map (e.g. by a surface of section or time one return map). The map is given by $\xi_{n+1} = \mathbf{f}(\xi_n, c)$, where c is an accessible system parameter. This parameter is considered available for external adjustment with the aim of achieving control.
2. There is a specific periodic orbit of the map which lies in the attractor and around which we wish to stabilize the dynamics.
3. There is a maximum perturbation $\delta c_{\max}$ in the parameter c around the nominal value c_0.
4. The position of the periodic orbit is a function of p, but the local dynamics about it do not vary much with the allowed small changes in c.

These assumptions imply that for the control of a chaotic system the OGY method must be able to construct an accurate Poincaré map of the dynamic system.

To apply the method, one first has to locate the unstable fixed points of the chaotic system. A local linear approximation of the map $\mathbf{f}$ (around the desired periodic trajectory) is then constructed, by observing iterates of the map near the desired orbit. To control the system, the technique attempts to confine the iterates of the map to a small neighbourhood of the desired orbit. Once the system passes near the desired orbit the control input c is changed from c_0 to $c_0 + \delta c$. In this way, the location of the orbit and its stable manifold (associated with the unstable fixed point) is changed so as the next state will be forced back towards the stable manifold of the original orbit for $c = c_0$. However, if one application of the changed control input $(c_0 + \delta c)$ does not succeed, one has to wait until the system passes again from that state.

An example of how the OGY method works for a saddle fixed point is shown in Fig. 5. The system's state ξ_n passes near the fixed point $\xi_F(c_0)$ (Fig. 5a). A perturbation δc of the control input c (Fig. 5b) forces the next system state ξ_{n+1} onto the stable manifold of $\xi_F(c_0)$ (Fig. 5c).

From the above, one can see that the OGY method requires extracting the locations of unstable fixed points. Since it is based on linear approximation of the local dynamics of the system, the method will not be robust, especially in the presence of noise. Besides that, the original method waits until the system state passes through the neighbourhood of the unstable fixed point, something which might take a long time, if the attractor of the system is high-dimensional. Such a delay can be critical in many systems.

The OGY method relies on small variations of one or more control parameters to stabilize the system into an (otherwise unstable) periodic motion natural to the system. For this reason it is termed a 'weak control' method. In contrast, the neuro-genetic control architecture is a 'strong method' which aims to employ comparatively large variations of control signals to force the system to acquire any desired target state or to track any desired trajectory in state space. Of course, such a weak method can be desirable in many cases, since it does not expend so much 'control energy' to achieve stabilisation.

4. The Attitude Control Problem

Control of the orientation of a rigid body has important applications from pointing and slewing aircraft, helicopters, spacecraft and satellites, to the orientation of a rigid object held by single or multiple robot arms.

We briefly outline the mathematical system which describes the rotational motion and orientation of a rigid body and which is used in the simulator. The angular velocities are determined by a system of first order differential equations (Euler equations):

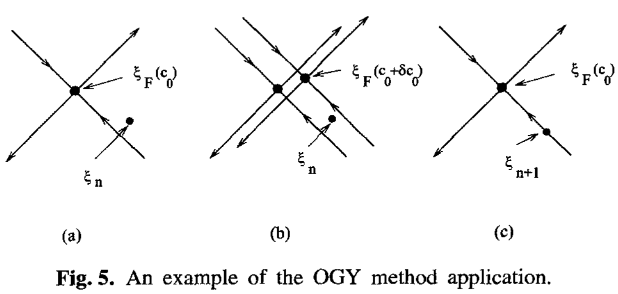

Fig. 5. An example of the OGY method application.

$$
\begin{aligned}
G_1 &= I_x\dot{\omega}_1 - (I_y - I_z)\omega_2\omega_3 \\
G_2 &= I_y\dot{\omega}_2 - (I_z - I_x)\omega_3\omega_1 \\
G_3 &= I_z\dot{\omega}_3 - (I_x - I_y)\omega_1\omega_2
\end{aligned} \tag{2}
$$

where I_x,I_y,I_z are the principal moments of inertia, $\omega_1,\omega_2,\omega_3$, are angular velocities about the principal x,y,z axes fixed in the rigid body, and G_1,G_2,G_3 are torques applied about these axes at time t.

Since the frame of reference adopted for the equations of motion is fixed to the rigid body, and moves with it, the position and orientation of the object cannot be described relative to this frame. As in Meyer [29], the orientation may be described locally by three angles ϕ,θ,ψ which represent consecutive rotations about a set of orthonormal axes **i,j,k** fixed in the body, with origin at the centre of mass of the body. By performing this sequence of rotations we bring the (**i,j,k**) frame into alignment with an inertial frame (**I,J,K**). We may then obtain the kinematic equation and derive the relationship between the inertial frame orientation angles and the body frame angular velocities. This yields

$$
\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \cdot \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{3}
$$

Although these angles do not coincide with the definition of Euler angles, they serve the same purpose and give rise to a convenient set of equations.

If we substitute $\omega_1,\omega_2,\omega_3$, as given by Eq. (3), into Eq. (2) we obtain a system of differential equations for ϕ, θ, ψ. The solution of these equations in general form presents a formidable problem, but the equations can easily be numerically integrated.

In general, the Euler equations are more complicated than Lorentz's system, and for certain choices of I_x,I_y,I_z and G_1,G_2,G_3 exhibit both strange attractors and limit cycles [30]. Specifically, if we choose $I_x = 3$, $I_y = 2$, $I_z = 1$ and

$$
\begin{pmatrix} G_1 \\ G_2 \\ G_3 \end{pmatrix} = \begin{pmatrix} -1.2 & 0 & \sqrt{6}/2 \\ 0 & 0.35 & 0 \\ -\sqrt{6} & 0 & -0.4 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \tag{4}
$$

the Euler equations produce a binary system of strange attractors for which the attractor of an orbit is determined by the location of the initial point of the orbit. The Poincaré section for the $\omega_1 - \omega_3$ plane of the system, for the trajectory starting at (3, 2, 1) is shown in Fig. 6. (This diagram was produced by numerical integration of the above equations, using

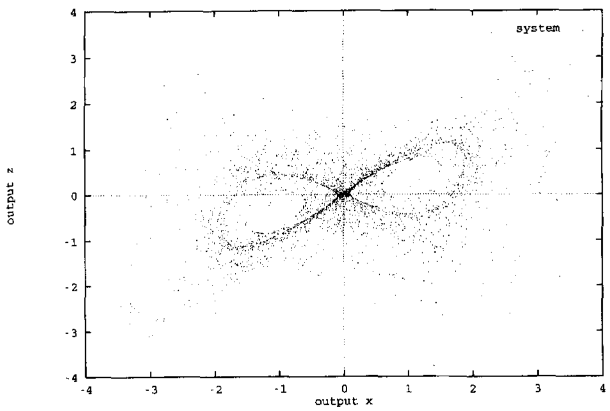

Fig. 6. Euler system. Poincaré section. Output x and output z refer to the x and z components of the ω vector.

the Runge–Kutta fourth order method, and implemented in C by the authors.)

5. Satellite Attitude Control Subject to External Forces Leading to Chaos

In this section the task of stabilizing a rigid body satellite is considered. The equations describing the orientation and rotational motion of a rigid body, combined with moments of inertia commensurate with those of a satellite are used.

The moments of inertia of the satellite are $I_x = 1160\,\text{kg·m}^2$, $I_y = 23\,300\,\text{kg·m}^2$ and $I_z = 24\,000\,\text{kg·m}^2$. We now suppose that deterministic external forces act upon the system for $t > 0$. Thus the mathematical description becomes

$$I_x\dot{\omega}_1 - (I_y - I_z)\omega_2\omega_3 = G_1 + T_1$$
$$I_y\dot{\omega}_2 - (I_z - I_x)\omega_3\omega_1 = G_2 + T_2$$
$$I_z\dot{\omega}_3 - (I_x - I_y)\omega_1\omega_2 = G_3 + T_3 \qquad (5)$$

where the external torques T_1,T_2,T_3 are given by the linear feedback equations

$$T_1 = -1200\omega_1 + 1000 \cdot \frac{\sqrt{6}}{2}\,\omega_3$$
$$T_2 = 350\omega_2$$
$$T_3 = -1000 \cdot \sqrt{6}\omega_1 - 400\omega_3 \qquad (6)$$

The above is a system in which the externally imposed torques would, left to themselves, result in a chaotic motion (see, for example [1] – the torques here are 1000 times greater because the moments of inertia are larger and we want the perturbing forces to be significant, but the dynamics are similar).

In the simulation results that follow, every time the system dynamics change in an unknown way, a new neuromodel describing the new dynamics of the plant has to be trained according to this methodology. Full details of how such a neuromodel can be constructed can be found elsewhere [1,2]. In practice, since vanilla backpropagation (adjusting weights with gradient descent and used here) converges rather slowly and sometimes unreliably, this would require a hardware implementation of backpropagation, or some other method for rapidly learning a dynamic model. As new and faster learning algorithms are derived, these can be used for speeding up the training of a network. For example a conjugate gradient based backpropagation converges much faster than the vanilla backpropagation but here we do not discuss such issues as these are discussed in [1,31] and elsewhere, and we concentrate in investigating the viability of the genetic part of the hybrid controller.

Here, the genetic controller uses data (i.e. evaluations of hypothetical control signals) provided by a previously trained (with 1000 training data coming from the plant) gradient descent based backpropagation.

The goal in this simulation is to acquire the state $(\omega_1,\omega_2,\omega_3) = (0{,}0{,}0)$ and $(\phi,\theta,\psi) = (0{,}0{,}0)$. The initial conditions are $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

The objective function for the genetic controller is specified to be

$$F(G_1,G_2,G_3) = |\dot{\phi} + \phi| + |\phi| + |\dot{\theta} + \theta| + |\theta| + |\dot{\psi} + \psi| + |\psi| \qquad (7)$$

Figures 7–9 show the evolution in time of the angular velocities ω_1,ω_2,ω_3 about the x,y,z body axes, respectively. The genetic controller soon leads these angular velocities to the prespecified values of zero despite the fact that during the control process the externally perturbing torques were as large as 44% of the maximum available control torques. When the target state is acquired the controller maintains the angular velocities.

In Figs 10–12 the reorientation of the satellite for the angles ϕ,θ,ψ during the application of the genetic controller is shown. Again, once achieved the controller maintains these angles.

The applied thrusts during the genetic control of the satellite are shown in Figs 13–15.

6. Control Subject to Sensor Noise and External Forces Leading to Chaos

We now add artificial noise, according to a uniform distribution, of 5% of the current sensors values. In

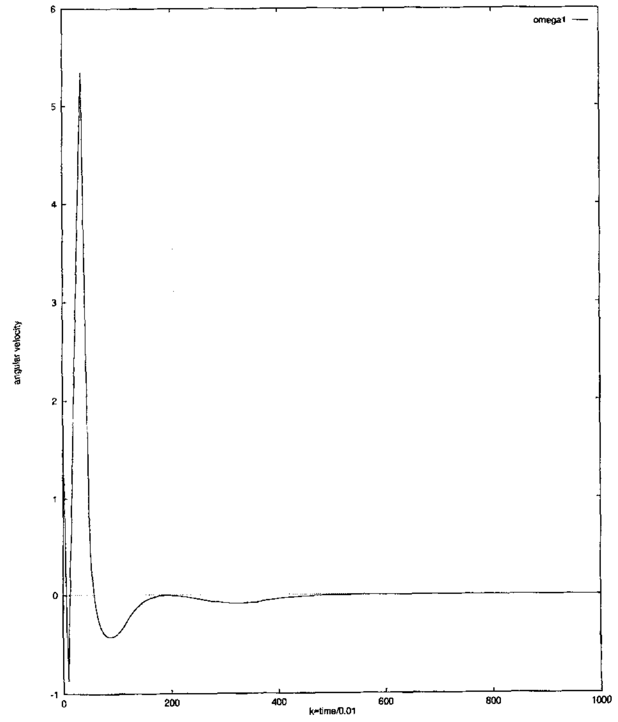

Fig. 7. Angular velocity ω_1 for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

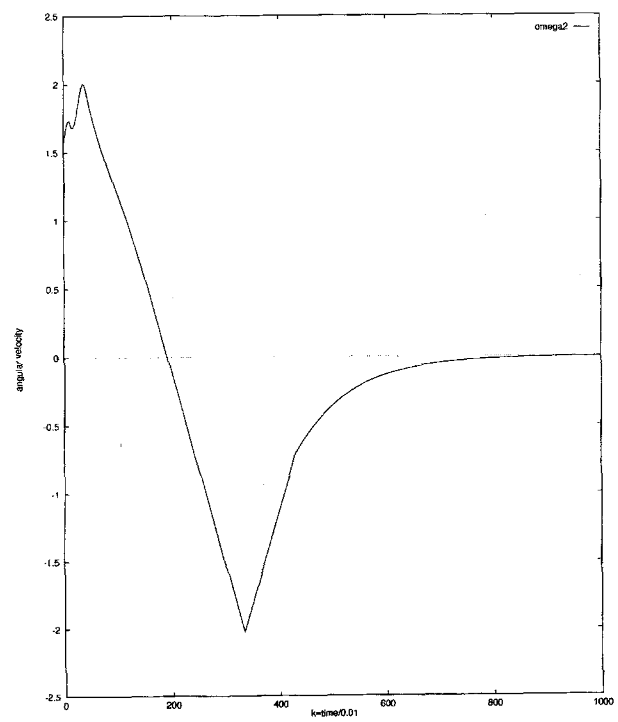

Fig. 8. Angular velocity ω_2 for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

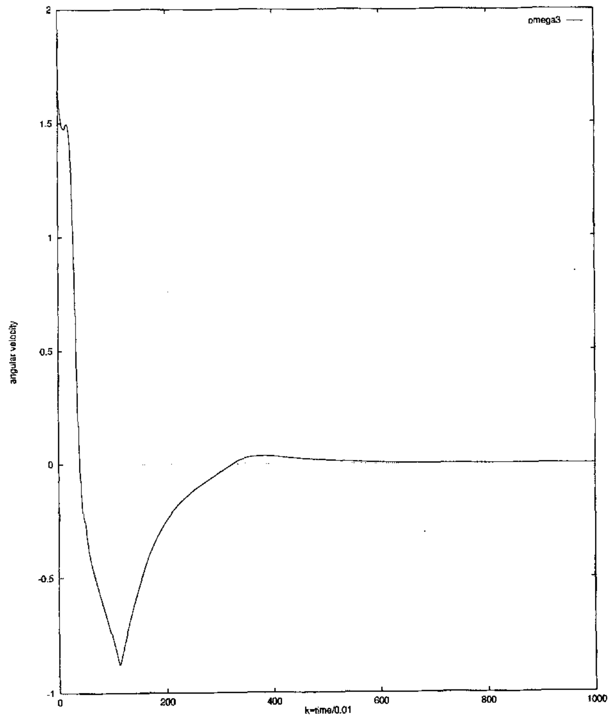

Fig. 9. Angular velocity ω_3 for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

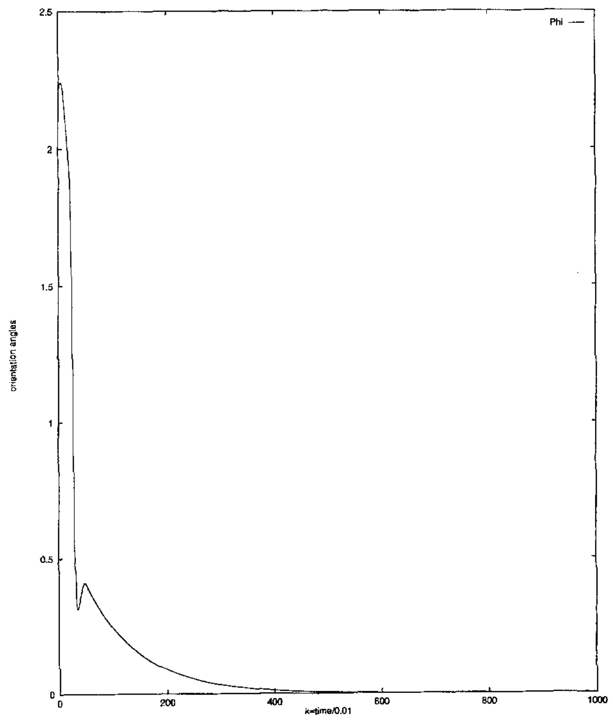

Fig. 10. Orientation angle ϕ for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

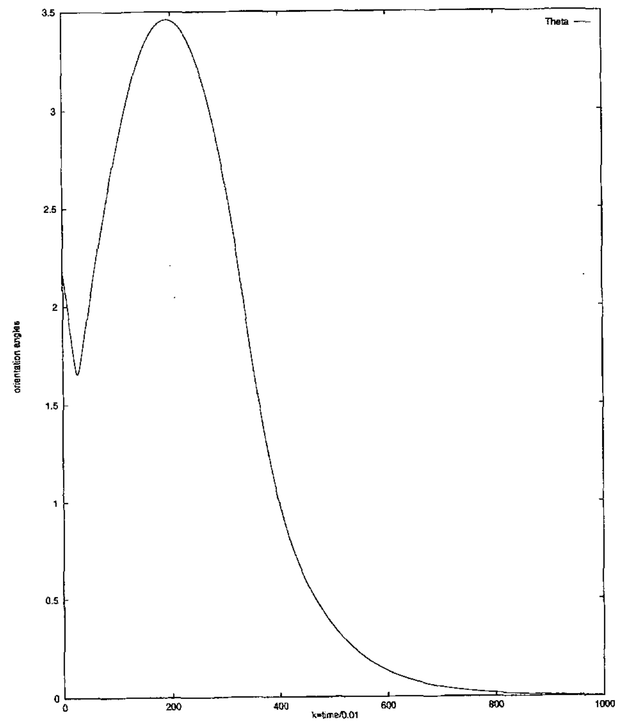

Fig. 11. Orientation angle θ for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

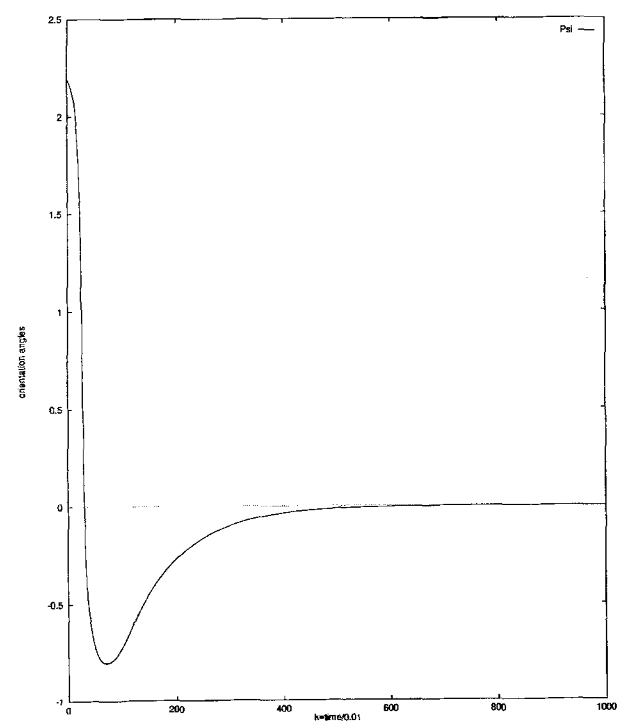

Fig. 12. Orientation angle ψ for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

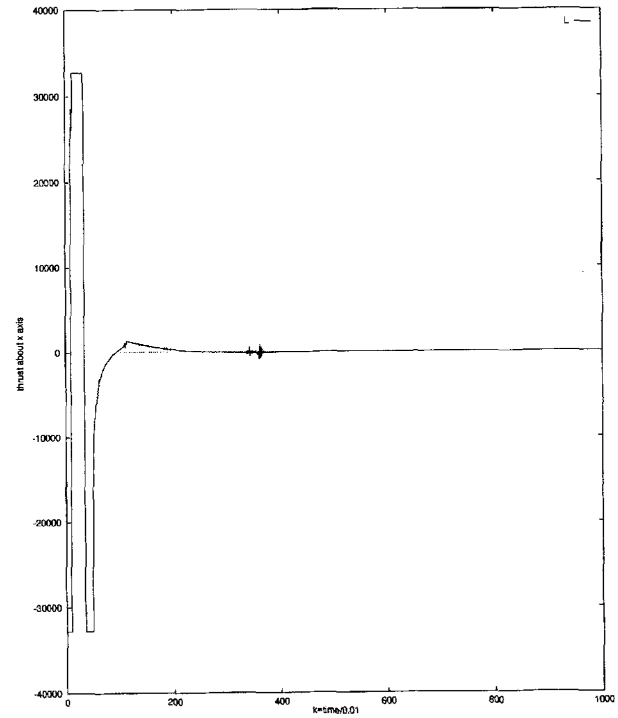

Fig. 13. Applied thrust G_1 about the x body axis for the satellite, in the presence of forces trying to lead it in a chaotic motion, during the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

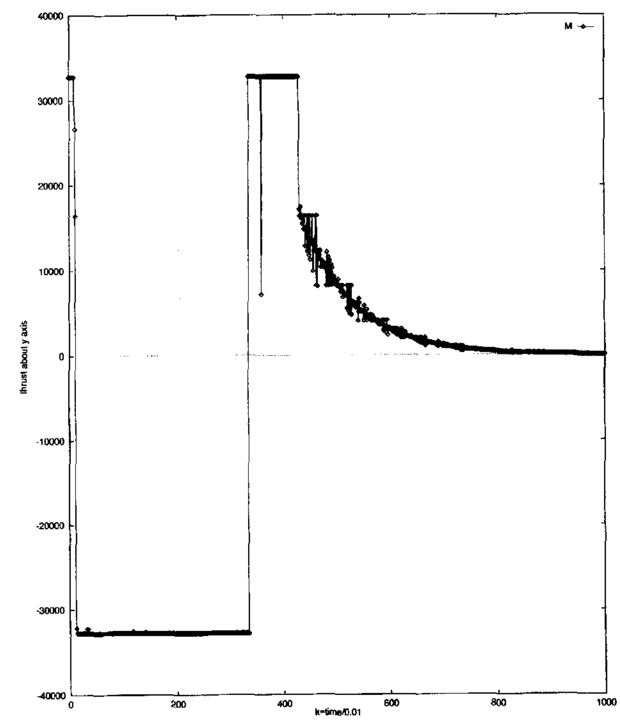

Fig. 14. Applied thrust G_2 about the y body axis for the satellite, in the presence of forces trying to lead it in a chaotic motion, during the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

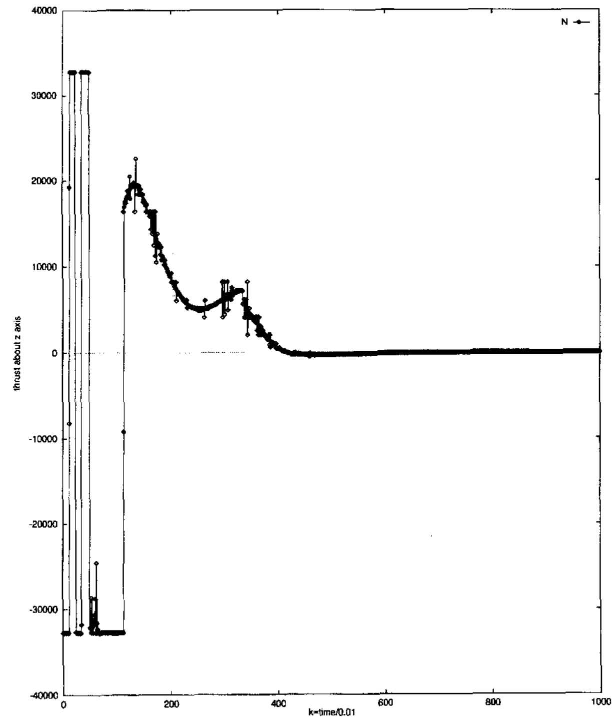

Fig. 15. Applied thrust G_3 about the z body axis for the satellite, in the presence of forces trying to lead it in a chaotic motion, during the application of the neuro-genetic controller. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}0,\ 1{\cdot}5,\ 1{\cdot}7)$ and $(\phi,\theta,\psi) = (2{\cdot}2,\ 2{\cdot}2,\ 2{\cdot}2)$.

addition, as in the previous example, externally imposed torques are acting upon the satellite for $t > 0$. The system is again described by the dynamic equations (5), (6). The moments of inertia of the satellite in this simulation are $I_x = 1200\,\mathrm{kg{\cdot}m^2}$, $I_y = 22\,000\,\mathrm{kg{\cdot}m^2}$ and $I_z = 25\,000\,\mathrm{kg{\cdot}m^2}$, but this is unknown to the controller as is the nature of the perturbing forces. For this reason, a new neuromodel is constructed (using 1000 training data), to describe the new plant dynamics.

The goal in this simulation is to acquire the state $(\omega_1,\omega_2,\omega_3) = (0,0,1)$ and $(\phi,\theta) = (0,0)$. The initial conditions are $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

Since the target state has a non-zero spin $\omega_3 = 1$, the forces will remain large in the vicinity of the target state and we expect to see asymptotically non-zero control torques as $t \longrightarrow \infty$.

The choice of the objective function needs some care. Since noise is present, when the system is near the target state much of the error will be due to noise and we would like to reduce 'hunting' (i.e. over-energetic control torques), and thus the long-term energy expenditure of maintaining the target state. Consequently, we want to smooth the angular acceleration near the target state. One way (amongst many) to achieve this is to introduce a smoothing term which forces $\ddot{\psi} \approx 1 - \dot{\psi}$ near the target state. Thus, our objective function is chosen to be

$$\begin{aligned} F(G_1,G_2,G_3) &= |\dot{\phi} + \phi| + |\phi| + |\dot{\theta} + \theta| \\ &\quad + |\theta| + |\dot{\psi} - 1{\cdot}0| + 0{\cdot}01 \cdot |\ddot{\psi} + \dot{\psi} - 1{\cdot}0| \end{aligned} \tag{8}$$

at the next sensor sample. This presupposes additional sensors for angular acceleration and would involve adding an extra output $\ddot{\psi}$ to the LPN predictor. For the present purpose, assuming the hypothetical thrusts we take a value of $\ddot{\psi}$ provided by the simulator.

Figures 16–18 show the evolution in time of the angular velocities $\omega_1,\omega_2,\omega_3$. Despite the presence of noise and large perturbing forces, the genetic controller manages to lead the system to the target state and once there maintain the target with thrusts comparable to the perturbing forces.

In Figs 19–21, the reorientation for the angles ϕ,θ,ψ during the application of the genetic controller is shown. Once again we see that the target orientation (0, 0) is acquired and maintained with acceptable accuracy (since there is 5% noise present, accuracy beyond a certain level cannot be achieved).

The applied thrusts during the application of the genetic controller are shown in Figs 22–24. Any particular set of control thrusts are applied for 0·01 s

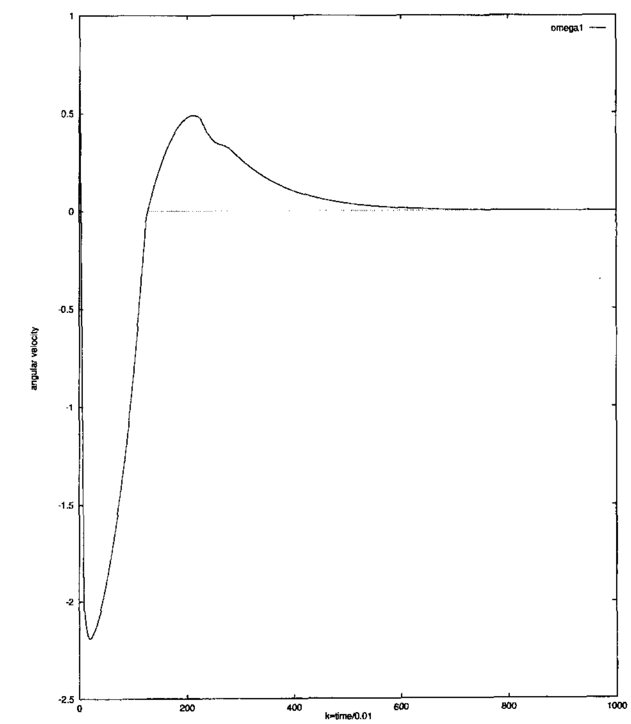

Fig. 16. Angular velocity ω_1 for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Presence of 5% noise in the sensors. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

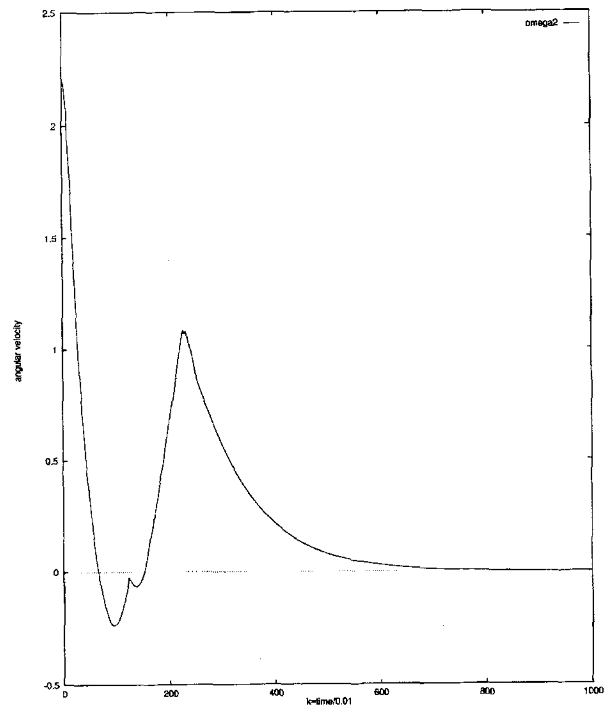

Fig. 17. Angular velocity ω_2 for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Presence of 5% noise in the sensors. Initital conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

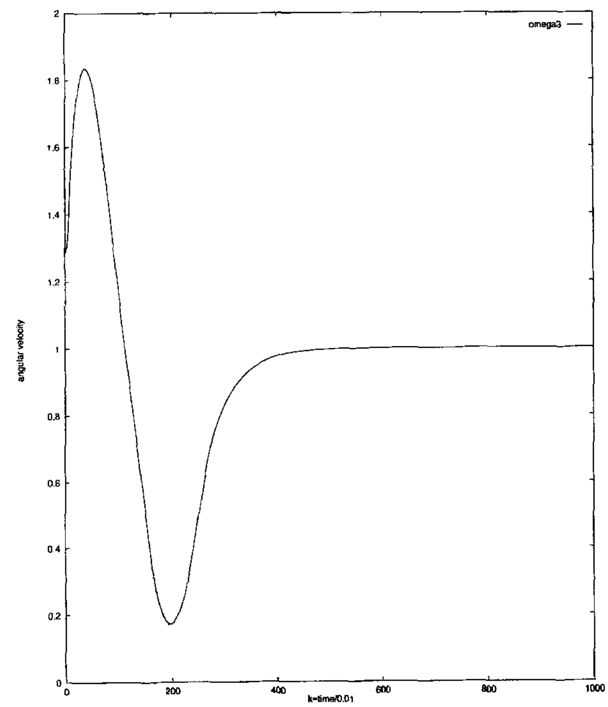

Fig. 18. Angular velocity ω_3 for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Presence of 5% noise in the sensors. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

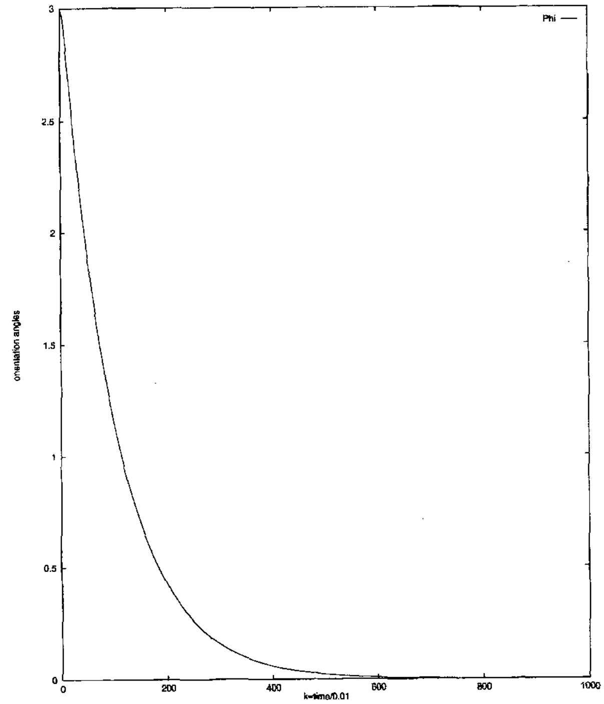

Fig. 19. Orientation angle ϕ for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the neuro-genetic controller. Presence of 5% noise in the sensors. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

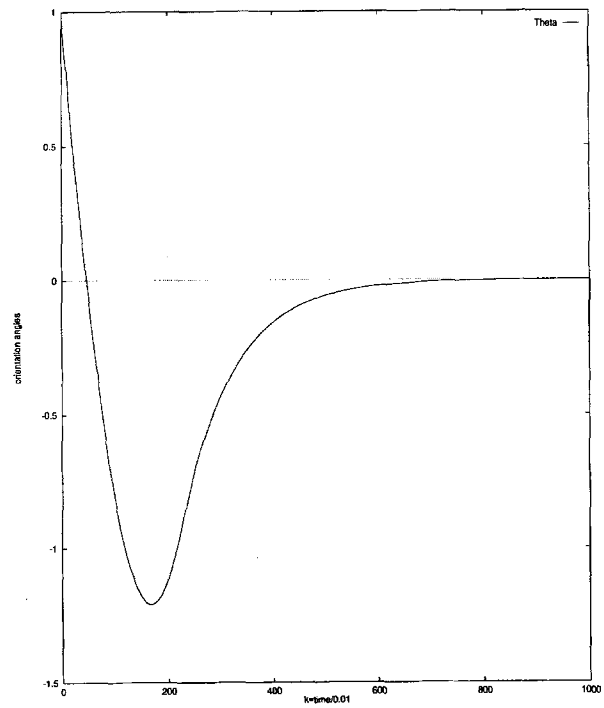

Fig. 20. Orientation angle θ for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the genetic controller. Presence of 5% noise in the sensors. Initital conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

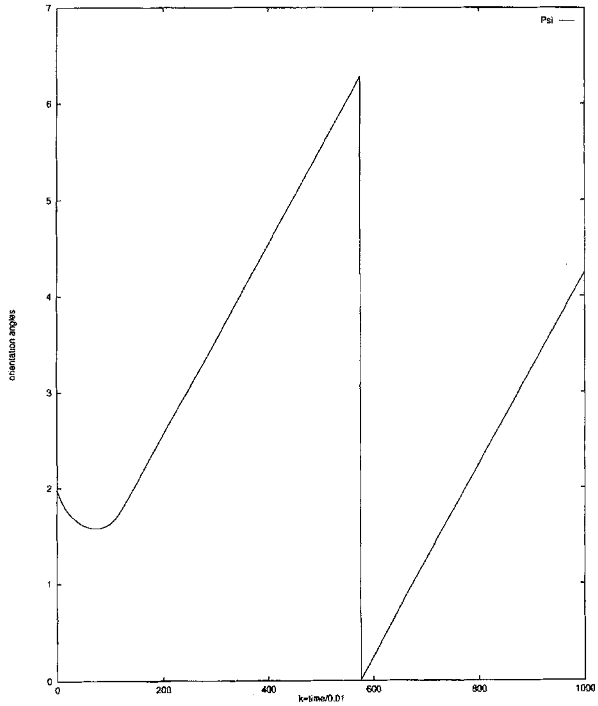

Fig. 21. Orientation angle ψ for the satellite, in the presence of forces trying to lead it in a chaotic motion, after the application of the genetic controller. Presence of 5% noise in the sensors. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

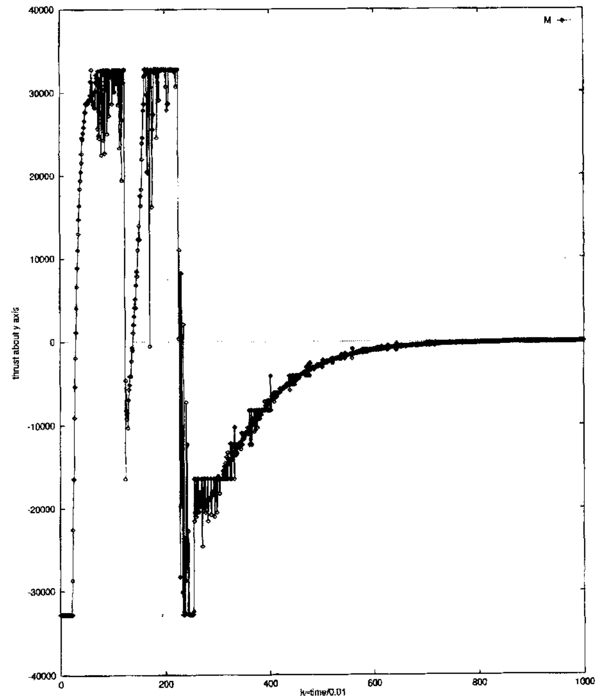

Fig. 23. Applied thrust G_2 about the y body axis for the satellite, in the presence of forces trying to lead it in a chaotic motion, during the application of the genetic controller. Presence of 5% noise in the sensors. Initital conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

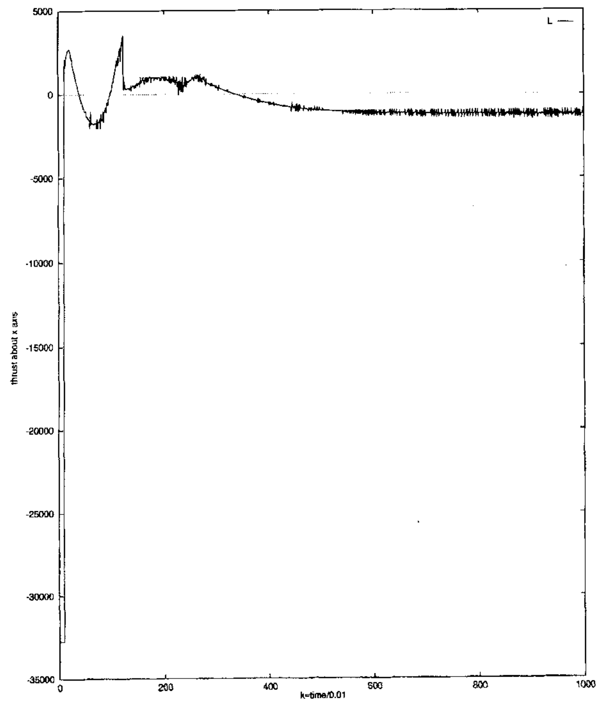

Fig. 22. Applied thrust G_1 about the x body axis for the satellite, in the presence of forces trying to lead it in a chaotic motion, during the application of the genetic controller. Presence of 5% noise in the sensors. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\theta,\phi,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

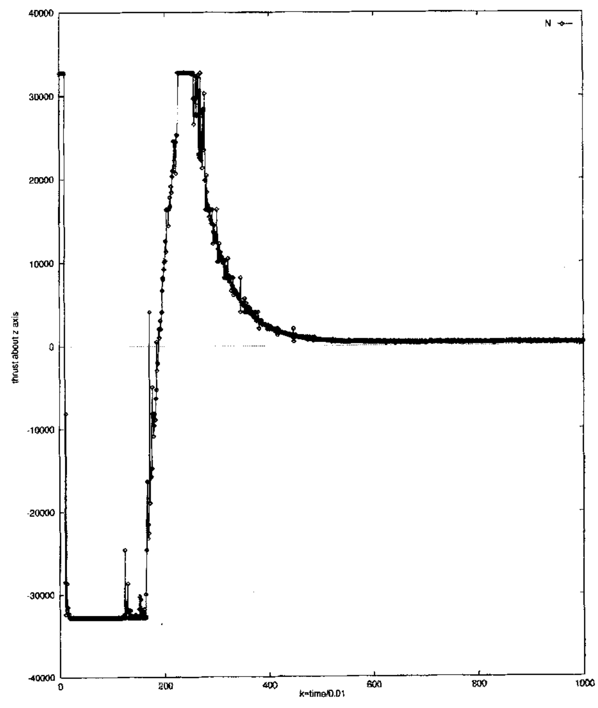

Fig. 24. Applied thrust G_3 about the z body axis for the satellite, in the presence of forces trying to lead it in a chaotic motion, during the application of the genetic controller. Presence of 5% noise in the sensors. Initial conditions $(\omega_1,\omega_2,\omega_3) = (2{\cdot}3,\ 1{\cdot}3,\ 1{\cdot}1)$ and $(\phi,\theta,\psi) = (2{\cdot}1,\ 1{\cdot}2,\ 2{\cdot}9)$.

(this is probably more frequently than would in reality be necessary) and the complete simulation runs for a time period of 10 s. These results illustrate that the proposed architecture is able to 'control chaos' even in the presence of noise.

7. Aspects of Objective Function Choice

The nature of a general objective function needs some care because not all the variables one could specify in various circumstances are independent. For example, if all three angular velocities are specified to be non-zero, then obviously the orientation angles cannot be specified as fixed values (except initially) because they will also be changing. Similarly, if the goal were to keep one axis pointing at an external relatively moving target, then all three orientation angles must be variable, according to some reference model, and (assuming the tracking is successful) all three angular velocities will change so as to maintain the desired attitude. In this case the relative speed with which the external target moves is critical in forming a tracking strategy. A satellite in Earth orbit required to keep one axis pointing at the Earth has a relatively simple tracking problem, requiring only small changes of attitude from an essentially equilibrium state to effect one rotation every orbit. In this case we can add acceleration smoothing terms to reduce energy consumption. If the target is changing relative orientation rapidly then energy considerations become secondary. We have not pursued these issues in detail but tracking and energy minimisation raise interesting questions. In general, if the target state is fixed rather than dynamic, energy minimisation obviously involves a tradeoff against time to target acquisition.

It is also possible that the initial angular velocities may be so high that the time window for the computation required by the controller may be too small. This situation is easily detected and, assuming sufficient energy is available, a simple feedback strategy designed to minimise kinetic energy can be employed until the angular velocities reduce to an acceptable magnitude.

8. Conclusions

In Ref. [2] we examined some simple simulation experiments in which a neuro-genetic architecture was applied to the adaptive attitude control problem for arbitrary transitions of system states. The experiments described there, tested the ability of the system to acquire and maintain an arbitrary target state with no prior knowledge of the system dynamics. This was achieved for both dynamically stable and unstable target states.

In the present paper, we have extended these simulations to very much harder scenarios: the presence of noise, large perturbing forces (which left to themselves lead the system to a chaotic motion), and a combination of the two. In all cases, the neuro-genetic controller performed well.

The suggested neuro-genetic architecture is quite general, in the sense that it is applicable to a wide range of control tasks, such as reaching a desired state or tracking an available reference model. Adaptive nonlinear control is achieved by constructing a plant model, using neural networks employing supervised learning. The generality of the method does not require simplifying assumptions for the plant, for example a linearization of the plant dynamics, but deals directly with the unknown nonlinearities of the dynamic system under consideration through the use of a genetic algorithm to solve the inverse kinematic problem. The neural network provides the evaluation of hypothetical control signals generated by the genetic algorithm (this would normally be the most time consuming part of a GA computation). It is supposed that the neural net learning and operation is executed in hardware. Whether or not it is necessary to implement the GA in hardware depends on the particular application but it would appear that in many cases a fast serial processor should suffice.

It is of some interest to compare the present approach with the OGY method. This was the first practical method for controlling a chaotic system [18]. The methods proposed in the present paper offer the possibility of effectively acquiring and maintaining any physically realisable target state in both chaotic and more manageable systems. In contrast the OGY method really only applies to chaotic systems and before beginning control the OGY method (in its original form) was forced to wait until the chaotic system trajectory passed close to a specified unstable periodic orbit which is the target state. Thus the OGY method requires some knowledge of plant dynamics (e.g. the unstable periodic orbits) whilst the method discussed here requires no such knowledge. On the other hand, once implemented, the OGY method is computationally very simple whilst the demands of the neuro-genetic controller, although not excessive, are certainly significantly greater.

Finally, there remains the difficult question of control stability. Neuro, or neuro-genetic, control are important because these methods offer the possibility of enabling the automated control of systems which could not be controlled in the past, for two reasons: the physical cost of implementing a known control algorithm, or the difficulty of finding such an algorithm which is robust in complex, noisy, nonlinear problems [32]. In addition, neurocontrol might offer reduced development times for control algorithms and the possibility of real-time adaptive performance in a standardized module. Empirical results, such as those presented here, have been very encouraging.

However, existing adaptive control technology offers some cautionary advice for those who envisage the rapid introduction of neurocontrol. The first adaptive controls were proposed in the late 1950s, particularly for use in high performance aircraft. In fact such controls were not widely used in practice and a significant reason for this was the lack of a satisfactory mathematical theory for their operation and in particular the lack of a proper stability theory, which plays a key role in conventional (non-adaptive) control. It has taken some 30 years work to derive a satisfactory stability theory for conventional adaptive control (self tuning regulators and Model Reference Adaptive Control systems) [6–9].

It is hard to see how such guarantees could be offered with, for example the neuro-genetic paradigm. Everything depends on how well the genetic algorithm performs its task and there is little theory to guide us in this respect at present. Conceivably, approximation theorems of a probabilistic type might be proved for genetic algorithms, but is difficult to see how such a result could be independent of the problem being addressed.

Even assuming general theorems were forthcoming, for safety critical systems difficult issues are raised. We might forgive a civil aircraft manufacturer's sceptical response to the assertion that 'this control system will work with probability 1 if a control solution is feasible'! On the other hand if such systems could be shown to be reliable in normal operation and be demonstrated to have the potential to rapidly adjust to hitherto unseen situations and re-provide control where this is actually feasible then the argument becomes less clear cut.

For example, a frequently quoted illustration of aircraft recovery that a contemporary control system would never have achieved concerns a hydraulic failure on a Lockheed Tristar jet. The Tristar is a wide-bodied airliner, with an engine under each wing and a third mounted at the base of the vertical tail surface, i.e. above the fuselage. With a full complement of passengers, such a Tristar suffered primary and secondary hydraulic failure to the elevators, the surfaces that provide the pitch attitude control of the aircraft. The flight crew saved the day and landed the jet by (quickly) realising that they could restore some degree of pitch control by applying differential thrust from the engines, because the tail mounted engine thrust was above the centre of gravity and the wing mounted engines thrust below. There was nothing in the manual about such a form of control, it was invented at 38 000 ft. 'on the fly'.

Future work will concentrate on examining methods in which the genetic algorithm objective function, could serve in the role of a Lyapunov function for the whole controller. In this way, stability of the controller can be guaranteed from theory. In addition, we are interested in developing techniques which are closer to biological neural control, as the architecture developed here, does not seem to have anything to do with the operation in the brain.

Acknowledgements. The authors would like to express their debt to the late Professor Patrick Parks, Mathematical Institute, Oxford, for his valuable comments on this work. He was a kind friend and a wise teacher and will be sorely missed.

References

1. Dracopoulos DC, Jones AJ. Neuromodels of analytic dynamic systems. Neural Comput & Applic 1993; 1(4): 268–279
2. Dracopoulos DC, Jones AJ. Neuro-genetic adaptive attitude control. Neural Comput & Applic 1994; 2(4): 183–204
3. Miller T, Sutton RS, Werbos PJ, Eds. Neural Networks for Control. MIT Press, 1990
4. Astrom KJ. Towards intelligent control. IEEE Control Systems Magazine April 1988
5. Goldberg KY, Pearlmutter BA. Using backpropagation with temporal windows to learn the dynamics of the cmu direct drive arm II. Neural Information Processing Systems 1. Morgan Kaufmann, 1989
6. Landau YD. Adaptive Control: The Model Reference Approach. Marcel Dekker, 1979
7. Parks PC. Lyapunov redesign of model reference adaptive control systems. IEEE Trans Automatic Control 1966; 11: 362–367
8. Narendra KS, Annaswamy AM. Stable Adaptive Systems. Prentice Hall, 1989
9. Astrom KJ, Wittenmark B. Adaptive Control. Addison-Wesley, 1989
10. Qammar HK, Mossayebi F. System identification and model-based control of a chaotic system. Int J Bifurcation and Chaos 1994; 4: 843–851
11. Fradkov AL, Pogromsky A, Markov A. Adaptive control of chaotic continuous-time systems. Proc Third Euro Control Conf, Rome, 1995, pp. 3062–3067
12. Ryan EP. A universal adaptive stabilizer for a class of nonlinear systems. Systems and Control Lett 1991; 16: 209–218

13. Ilchmann A, Ryan EP. Universal λ-tracking for nonlinearity-perturbed systems in the presence of noise. Automatica 1994; 30(2): 337–346
14. Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. Neural Networks 1990; 1(1): 4–27
15. Narendra KS, Mukhopadhyay S. Intelligent control using neural networks. IEEE Control Systems Mag 1992: 11–18
16. Narendra KS. Adaptive control of dynamical systems using neural networks. In: Handbook of Intelligent Control, DA White, DA Sofge, Eds. Van Nostrand Reinhold, 1992
17. Rajarshi R, Murphy Jr TW, Maier TD, Gills Z, Hunt ER. Dynamical control of a chaotic laser: Experimental stabilisation of a globally coupled system. Phys Rev Lett 1992; 68(9): 1259–1262
18. Ott E, Grebogi C, Yorke J. Controlling chaos. Phys Rev Lett 1990; 64(11)
19. Jordan MI, Rumelhart DE. Forward models: Supervised learning with a distal teacher. Cognitive Sci 1992; 16: 307–354
20. Rumelhart D, McClelland J. PDP research group. Parallel Distributed Processing – Explorations in the Microstructure Cognition, vol. 1. MIT Press, 1986
21. Werbos PJ. Backpropagation through time: What it does and how to do it. Proc IEEE 1990; 78(10)
22. Kawato M. Computational schemes and neural network models for formation and control of multijoint arm trajectory. In: Neural Networks for Control, S Miller, DJ Werbos, Eds. MIT Press, 1990
23. Kawato M, Uno T, Isobe M, Suzuki R. Hierarchical neural network model for voluntary movement with application. IEEE Control Systems Mag 1988
24. Katayama M, Kawato M. Learning trajectory and force control of an artificial muscle arm by parallel-hierarchical neural network model. Neural Information Processing Systems 3. Morgan Kaufmann, 1991
25. Psaltis D, Sideris A, Yamamura AA. A multilayered neural network controller. IEEE Control Systems Mag 1988: 17–21
26. Dracopoulos DC. Neuromodelling, Adaptive Neurocontrol and the Attitude Control Problem. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, March 1994
27. Chen G, Dong X. From chaos to order – perspectives and methodologies in controlling chaotic nonlinear dynamical systems. Int J Bifurcation and Chaos 1993; 3(6): 1363–1409
28. Ditto WL, Rauseo SN, Spano ML. Experimental control of chaos. Phys Rev Lett 1990; 65(26)
29. Meyer G. Design and global analysis of spacecraft attitude control systems. Technical Report TR R-361, NASA, 1971
30. Leipnik RB, Newton TA. Double strange attractors in rigid body motion with linear feedback control. Physics Lett 1981; 86A: 63–67
31. Lapedes A, Farber R. How neural nets work. Proc IEEE Denver Conf Neural Nets, 1987
32. Werbos P. Neurocontrol and related techniques. In: Handbook of Neural Computing Applications, A Maren Ed. Academic Press, 1990, pp. 345–381