Intelligent System Modelling and Simulation using Hybrid Recurrent Networks

D. Al-Dabass, D. Evans and S. Sivayoganathan

Department of Computing & Mathematics
Faculty of Computing and Technology
Nottingham Trent University
Nottingham NG1 4B
david.al-dabass@ntu.ac.uk

Abstract

Numerous intelligent systems in practice exhibit complex behaviour that cannot be easily modelled using simple nets. In this paper we re-cast this problem in terms of hybrid recurrent nets, which consist of combinations of static nodes, either logical or arithmetic, and recurrent nodes. The behaviour of a typical recurrent node is modelled as a second order dynamical system. The causal parameters of such a recurrent node may themselves exhibit temporal tendencies that can be modelled in terms of further recurrent nodes. Layers of recurrent nodes are added until a complete account of the behaviour of the system has been achieved. Algorithms are given to abduct the values of the parameters of these models from behaviour trajectories of intelligent systems. One novel aspect of the work lies in having a simple hierarchical 6^{th} order linear model to represent a fairly complicated behaviour encountered in numerous real examples in finance, biology and engineering.

1. Introduction

To model the behaviour of complex natural and physical systems, the authors (1, 2, 6, 7 and 8) have recently developed a number of explicit static algorithms to estimate the parameters of recurrent second order models that approximate the behaviour of these complex higher order systems. These algorithms rely on the availability of the time derivatives of the trajectory. A cascaded recurrent network architecture is used to 'abduct' these derivatives in successive stages. These derivatives form the input into non-linear functions to calculate the causal parameters of the model that simulates the behaviour of the real system under study. The technique is tested successfully on a wide range of parameter tracking algorithms ranging from the constant parameter algorithm that only requires derivatives up to order 4 to an algorithm that tracks two variable parameters and requires up to the 8^{th} time derivatives. The significance of the work relies on the use of a carefully structured 2-layer hierarchy of 3 second order models whose parameters can be easily estimated using the techniques outlined in this paper. The behaviour of the resulting 6^{th} order linear model may be used to approximate fairly complicated signals found in many applications (3 and 5).

2. Hybrid Recurrent Network Models

Many physical, economical and biological phenomena exhibit a temporal behaviour even when the input 'causal' parameters are constant, Fig. 1.

The causal parameters themselves may be the output of other nodes, which may either be recurrent nodes or static nodes,- the latter may be logical or arithmetic.

To model this oscillatory behaviour we propose a second order integral hybrid model shown in Fig. 2. This model is based on the well known second order dynamical system which has the following form:

$\omega^{-2} x'' + 2.\ \zeta.\omega^{-1}.x' + x = u$

Where x is the output of the node and ω, ζ and u are the natural frequency, damping ratio and input respectively, which represent the 3 causal parameters that form the input. To configure this differential model as a recurrent network, a twin integral elements are used to form a hybrid integral-recurrent net as shown in Fig. 2.

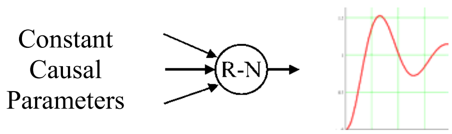

Fig. 1. A Recurrent Node (R-N) exhibits a temporal behaviour at the output despite having constant causal parameters.

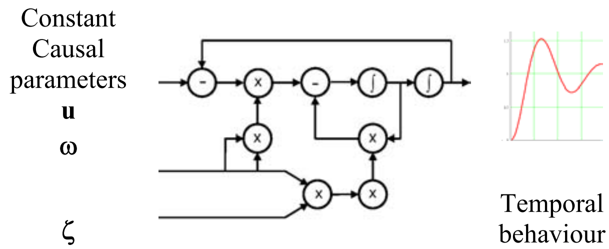

Fig. 2. Hybrid integral-recurrent net to model the temporal behaviour of the node in Fig. 1

3. Models of Hierarchical Recurrent Nodes

The output trajectory of the system may be more complex than can be represented by a simple second order differential model. In this case each causal parameter may itself be modelled as having a dynamical behaviour, which may or may not be oscillatory. One such case is where two of the 3 causal parameters have 2nd order dynamical characteristics, as shown in Fig. 3.

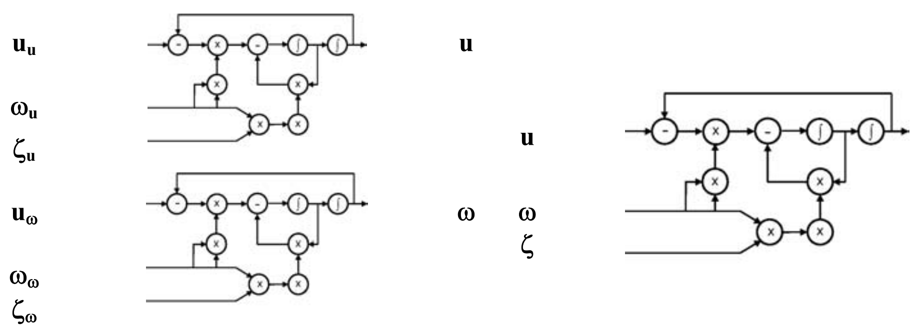

Fig. 3. Two of the causal parameters of the final node have temporal behaviour modelled as 2nd order hybrid integral recurrent nets.

The 2nd order model of a node in a given layer in the hierarchy is given by:

$$\omega^{-2}\, x'' + 2.\, \zeta.\omega^{-1}.x' + x = u$$

To model the complicated behaviour of intelligent systems let both u and omega have their own 2nd order dynamics. The input **u** is the output of the following 2nd order system:

$$\omega_u^{-2}\, \mathbf{u''} + 2.\, \zeta_u\, .\, \omega_u^{-1}.\mathbf{u'} + \mathbf{u} = \mathbf{u_u}$$

The natural frequency **ω** is the output of the following 2nd order system:

$$\omega_\omega^{-2}\, \boldsymbol{\omega''} + 2.\, \zeta_\omega\, .\, \omega_\omega^{-1}.\boldsymbol{\omega'} + \boldsymbol{\omega} = \mathbf{u_\omega}$$

Thus the behaviour trajectory is generated by the following 6th order vector differential equation (using Runge Kutta in Mathcad for this example).

$$D(t,x) := \begin{bmatrix} x_2 \\ x_5 \cdot x_5 \cdot x_3 - 2 \cdot z \cdot x_5 \cdot x_2 - x_5 \cdot x_5 \cdot x_1 \\ x_4 \\ (wu \cdot wu \cdot uu) - 2 \cdot zu \cdot wu \cdot x_4 - wu \cdot wu \cdot x_3 \\ x_6 \\ (ww \cdot ww \cdot uw) - 2 \cdot zw \cdot ww \cdot x_6 - ww \cdot ww \cdot x_5 \end{bmatrix}$$

Fig. 4. The derivative vector for generating the intelligent system output using subsystems for u and omega.

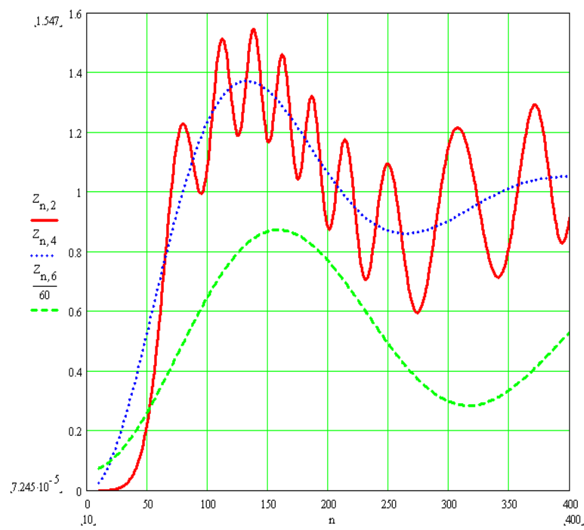

Fig. 5. Simulated trajectory of a hierarchical recurrent node (oscillatory trace), with 2 variable inputs: u (upper trace) and omega (lower trace)

Where x1 and x2 represent the x and x', x3 and x4 represent u and u', and x5 and x6 represent ω and ω' respectively. To generate the trajectory shown in Fig. 5, the following values were used: for the u subsystem, u started from 0 aiming at uu=1 at a rate of $\omega_u = 5$ rad/s with $\zeta_u = 0.3$. For the ω subsystem, ω started from 4 rad/s aiming at $u_\omega = 32$ rad/s at a rate of $\omega_\omega = 4$ rad/s with $\zeta_\omega = 0.1$. The resulting compound trajectory of x (oscillatory trace), together with the trajectories for u (upper trace) and ω (lower trace) are shown in the graph below.

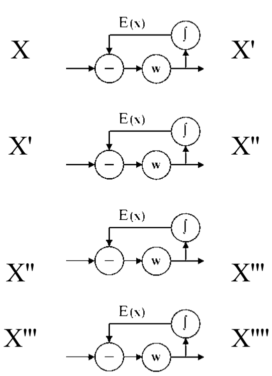

Fig. 6. A 4th order recurrent network to abduct 1st to 4th time derivatives.

$$D(t,x) := \begin{bmatrix} x_2 \\ x_5 \cdot x_5 \cdot x_3 - 2 \cdot z \cdot x_5 \cdot x_2 - x_5 \cdot x_5 \cdot x_1 \\ x_4 \\ (wu \cdot wu \cdot uu) - 2 \cdot zu \cdot wu \cdot x_4 - wu \cdot wu \cdot x_3 \\ x_6 \\ (ww \cdot ww \cdot uw) - 2 \cdot zw \cdot ww \cdot x_6 - ww \cdot ww \cdot x_5 \\ G \cdot (x_1 - x_7) \\ G1 \cdot [G \cdot (x_1 - x_7) - x_8] \\ G2 \cdot [G1 \cdot [G \cdot (x_1 - x_7) - x_8] - x_9] \\ G3 \cdot [G2 \cdot [G1 \cdot [G \cdot (x_1 - x_7) - x_8] - x_9] - x_{10}] \\ G4 \cdot [G3 \cdot [G2 \cdot [G1 \cdot [G \cdot (x_1 - x_7) - x_8] - x_9] - x_{10}] - x_{11}] \\ G5 \cdot [G4 \cdot [G3 \cdot [G2 \cdot [G1 \cdot [G \cdot (x_1 - x_7) - x_8] - x_9] - x_{10}] - x_{11}] - x_{12}] \end{bmatrix}$$

Fig. 7 A recurrent node model, top 2 rows, with the input u modelled in rows 3 and 4, and omega in rows 5 an 6; rows 7 to 12 model the derivative estimation 1st order recurrent nodes

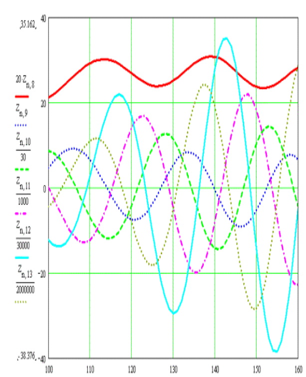

Fig. 8. A typical set of a recurrent node trajectory (top) and 5 higher order time derivatives x' to x''''' abducted from a simulation of an intelligent system trajectory.

4. Abduction of Derivatives

To estimate the values of the derivatives a cascade of 1st order recurrent networks is used, Fig. 6. The output of each cell feeds the input to the next one to generate the next higher order time derivative. The output of the system and the cascade of 1st order recurrent network filters were simulated using the 4th order Runge-Kutta method in Mathcad. The derivatives vector for producing derivatives up to fifth is shown in Fig. 7. Figure 8 shows a typical set of high order time derivatives abducted from the output trajectory of a system displaying temporal behaviour.

5. Calculation of Model Parameters

To acquire the values of the model parameters, several Parameter estimation algorithms are available, [1,2,3,4]. The simplest algorithm assumes constant parameters in the derivation, while the more complicated ones attempt to fit successively higher order polynomial to at least one of the parameters, u being the most tractable analytically. They are given here without proof. For the constant parameter case they are:

$E\omega^2 = [x''. x'''' - x'''^2] / [x'. x''' - x''^2]$; $E\zeta = -[E\omega^{-2} x''' + x'] / [2. E\omega^{-1}.x'']$; and $Eu = E\omega^{-2}. x'' + 2. E\zeta. E\omega^{-1} . x' + x$

For the first order parameter case they are given in terms of a generalised set of parameters a, b and u:

$a = (x''.x'''' - x'''^2)/(x'''''.x''' - x''''^2)$; $b = -x''/x''' - a.x''''/x'''$; and $u = a.x'' + b.x' + x$.

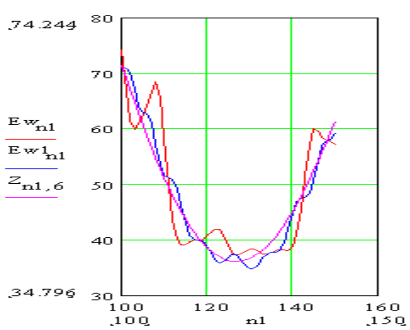

Fig. 9. Estimated Omega assuming constant parameters (Ew, very jagged trace) and first order parameters (Ew1, jagged trace) compared to actual (smooth trace).

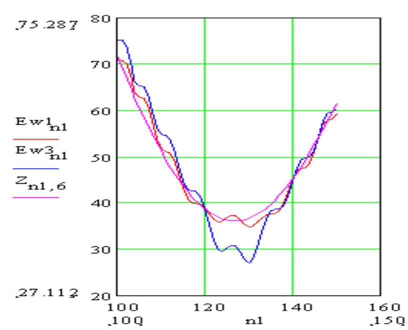

Fig. 10. Deterioration of accuracy of estimated Omega with higher order algorithms: 3rd order estimate (Ew3, very jagged trace) is worse than 1st order estimate (Ew1, jagged), actual (smooth).

Results: For a given range of parameters the algorithms worked well, being able to estimate the two causal parameters u and omega with their temporal behaviour, i.e. track them while they are changing. The 1st order algorithm worked better than the constant one, Fig. 9 shows a comparison of the two algorithms tracking omega. Algorithms of higher order than 1st showed marginal improvement but in certain cases showed a deteriorating behaviour, Fig. 10 shows a 3rd order algorithm deviating quite markedly from the true trajectory compared to a 1st order algorithm. This is likely to be due to an accumulation of errors in higher derivative values used in the former algorithm.

6. Conclusions and Future Work

A model consisting of hybrid linear recurrent nets was proposed to emulate the complex behaviour of intelligent systems. A 6th order 2 layer structure based on this model was used as an example to demonstrate its use. To estimate the values of the model parameters, the values of higher order time derivatives had to be estimated first. A cascade of 1st order recurrent networks were used to estimate these, followed by non-linear functions to estimate the parameter values. Several of these were tested, the one based on a 1st order parameters proved to be the most accurate overall. Higher order algorithms gave marginally more accurate results but their accuracy deteriorated under certain conditions such as mismatch in initial conditions.

Future work will extend the technique to a third layer to abduct the causal parameters of the temporal characteristics of the first level parameters. Application to compression and encoding will be attempted.

Biography. David Al-Dabass graduated from Imperial College in 1966 with BSc in electrical engineering, worked for Redifon Flight Simulation until 1972, completed a PhD in Parallel Processing at Staffordshire University in 1975 and held post-doctoral and advanced research fellowships (76-82) at the Control Systems Centre, UMIST. He joined The Nottingham Trent University in 1983 as a Principal Lecturer in the Department of Computing. For more details see his website: http://ducati.doc.ntu.ac.uk/uksim/dad/webpage.htm

References

1. Al-Dabass, D., Evans, D. J., & Sivayoganathan, K. (2002). Derivative Abduction using a Recurrent Network Architecture for Parameter Tracking Algorithms, IEEE 2002 Joint Int. Conference on Neural networks, World Congress on Computational Intelligence, May 12-17, Hawaii.

2. Al-Dabass, D., Evans, D. J., & Sivayoganathan, K. (2002). A Recurrent Network Architecture for Non-linear Parameter Tracking Algorithms, The Second International Conference On Neural, Parallel, And Scientific Computations, Morehouse College, Atlanta, U.S.A., August 07-10.

3. Seveance, F. L. (2001). Systems Modelling and Simulation, An Introduction. Wiley.

4. Al-Dabass, D. (2002). Modelling the Complexity of Concept Dynamics, 47th Meeting Of The International Society For The Systems Sciences, 2nd – 6th August, Shanghai International Convention Centre, Shanghai, China.

5. Close, C. M. & Fredrick, D. K., (1994). Modelling and Analysis of Dynamic Systems, 2nd edn. Wiley.

6. Al-Dabass, D., Zreiba, A., Evans D.,& Sivayoganathan, K. (1999). Simulation of Three Parameter Estimation Algorithms for Pattern Recognition Architecture, *UKSIM'99, Conference Proceedings of the UK Simulation Society*, St Catharine's College, Cambridge, 7-9 April, pp170-176, http://ducati.doc.ntu.ac.uk/uksim/papers/moller/dad.doc, ISBN 0-905488-38-5.

7. Al-Dabass, D., Zreiba, A., Evans D.,& Sivayoganathan, K. (1999). Simulation of Noise Sensitivity of Parameter Estimation Algorithms, Simulation'99 Workshop, UCL, London, 29 October, pp32-35.

8. Zreiba, A., Mphil Thesis (2000). Simulation of Real Time Parameter Estimation Algorithms for Time varying Systems, March 2000, Department of Computing, the Nottingham Trent University.

9. Goodwin, C, "Real Time Recursive Block Parameter Estimation of Second Order Systems", PhD Thesis, Dept. of Computing, The Nottingham Trent University, Nottingham, 1997.

10. Mathcad 8 Professional Program.