ANALYTICAL MODELS FOR PARAMETER TRACKING ALGORITHMS USING HYBRID RECURRENT NETWORKS

D. AL-DABASS, D. EVANS and S. SIVAYOGANATHAN

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

Abstracts: Several explicit algorithms for tracking the parameters of second order models have been derived by the authors based on information available from the system time trajectory. In this paper the problem is recast in terms of recurrent integral-hybrid networks used in a hierarchical formation for both the reduced order model and to estimate the derivatives for parameter tracking. We relax the constant parameter condition by assuming linear time variation, the additional information is extracted from the system output trajectory by obtaining higher time derivatives which result in explicit functions to track the parameters online.

1. INTRODUCTION

Online dynamical algorithm (Al-Dabass et al 1999) have been developed to combine estimates of a given trajectory time derivatives, using data from several points on the trajectory, with explicit static non-linear functions to provide continuous parameter estimation in real time. For time varying parameters, the time separation between the points on the trajectory had a direct influence on estimation accuracy, where the assumption of constant parameters used in the derivation is no longer valid, and accuracy deteriorates with increasing rate of parameter variation.

Ultimately, the separation effect can only be eliminated if all the information needed for the estimation is obtained from a single time point. An algorithm was derived and proved, as expected, to be the most successful in coping with high rates of parameter variation. However, accurate tracking of parameters when two or more of them were varying simultaneously still proved problematical. The essential assumption of constant parameters in the derivation must clearly be the fundamental cause of these difficulties.

In this paper we relax the constant parameter condition by assuming a linear time variation, i.e. constant first derivative but zero second and higher time derivatives of parameters. As may be expected, more information is needed for this new case, which is to be extracted from the system output trajectory by obtaining higher time derivatives. Explicit functions of the parameters are still possible as well as those of their first time derivatives. For clarity, the problem is recast in terms of recurrent hybrid networks for both the system model that generates the trajectory and the parameter tracking algorithm.

2. ANALYTICAL MODELS AND ALGORITHMS

A reduced order model that can be used to describe the oscillatory behaviour of systems forms an initial value problem of the type:

$$\omega^{-2}.x''+2.\zeta\,.\,\omega^{-1}.\,x'+x=u \quad x(0)=x_0, \quad x'(0)=x'_0$$

Where ω is the natural frequency, ζ is the damping ratio, u is the input and x is the output of the system. The three parameters may be constants, variables or variable with dynamical behaviour.

2.1 Models Assuming Constant Parameters From Single Point Data

Consider using the 1[st] to 4[th] time derivatives at a single point. Given the second order system:

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

Differentiate with respect to t:

$$\omega^{-2}\,x''' + 2.\,\zeta.\omega^{-1}.x'' + x' = 0 \tag{2}$$

Divide by x":

$$\omega^{-2}\,x'''/\,x'' + 2.\,\zeta.\omega^{-1} + x'/\,x'' = 0 \tag{3}$$

Differentiate with respect to t again to give:

$$\omega^{-2}.[(x''.\,x'''' - x'''^2)\,/\,x''^2] + 0 + [(x''^2 - x'.\,x''')\,/\,x''^2] = 0 \tag{4}$$

We get expressions for estimated ω, estimated ζ, using (2), and estimated u:

$$E\omega^2 = [x''.\,x'''' - x'''^2]\,/\,[x'.\,x''' - x''^2] \tag{5}$$
$$E\zeta = -[E\omega^{-2}\,x''' + x']\,/\,[2.\,E\omega^{-1}.x''] \tag{6}$$
$$Eu = E\omega^{-2}.\,x'' + 2.\,E\zeta.\,E\omega^{-1}\,.\,x' + x \tag{7}$$

2.2 Models for Time Varying Parameters

We assume that the first time derivative of u to be non 0. For simplicity we still assume that both a and b (the coefficients of x" and x' to make symbol manipulation easier) to be constant and hence disappear on first differentiation. The extra information needed for u' to be non zero is extracted from the 5th time derivative of the trajectory.

$$a.x'' + b.x' + x = u \qquad (1\text{-a})$$

Differentiate wrt to t and assume u' is non zero to give:

$$a.x''' + b.x'' + x' = u' \qquad (8)$$

Differentiate again and set u'' = 0 gives:

$$a.x'''' + b.x''' + x'' = 0 \qquad (9)$$

Divide Equation 4 by x''' to isolate b:

$$a.x''''/x''' + b + x''/x''' = 0 \qquad (10)$$

Differentiate again to eliminate b:

$$a.(x'''''.x'' - x''''^2)/x'''^2 + (x'''^2 - x''.x'''')/x'''^2 = 0 \qquad (11)$$

Re-arranging for a gives:

$$a = (x''.x'''' - x'''^2)/(x'''''.x''' - x''''^2) \qquad (12)$$

Solve for *b* by substituting a from equ. 12 into equ. 10:

$$b = -x''/x''' - a.x''''/x'''$$

which after substituting for a and manipulating gives:

$$b = (x''.x''''' - x'''.x'''')/(x''''^2 - x'''.x''''') \qquad (13)$$

We can now substitute these values for *a* and *b* into Equation 1to solve for u,

$$u = a.x'' + b.x' + x$$

3. 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 input parameters themselves may be the output of other nodes, which may either be recurrent nodes or static nodes,- the later 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:

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

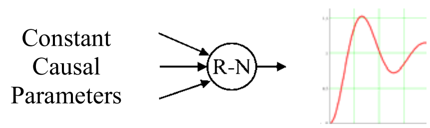


Figure 1: A Recurrent Node (R-N) exhibits an oscillatory behaviour at the output despite having constant input parameters.

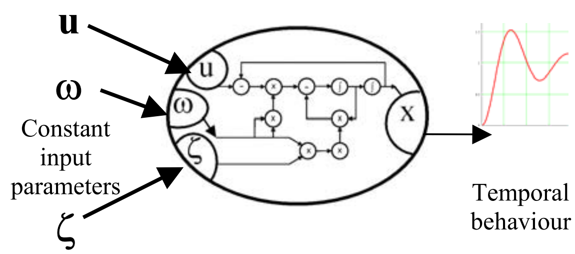


Figure 2: Hybrid integral-recurrent net to model the temporal behaviour of the node in Fig. 1

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.

3.1 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.

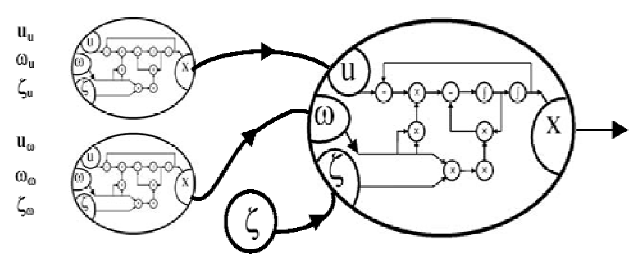


Fig. 3. Two of the input parameters of the final node are time varying modelled as 2nd order hybrid integral recurrent nets.

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:

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

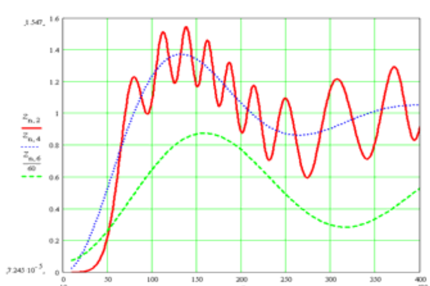
The natural frequency $\boldsymbol{\omega}$ is the output of the following 2nd order system:

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

Thus the behaviour trajectory is generated by the following 6th order vector differential equation (using Runge Kutta in Mathcad for this example), see Fig. 4.

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

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



Figurer 5: Simulated trajectory of a hierarchical recurrent node (red trace), with 2 variable inputs: u (blue) and omega (green)

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 ω_u = 5 rad/s with ζ_u = 0.3. For the ω subsystem, ω started from 4 rad/s aiming at u_ω = 32 rad/s at a rate of ω_ω = 4 rad/s with ζ_ω = 0.1. The resulting compound trajectory of x (red), together with the trajectories for u (blue dotted) and ω (green dotted) are shown in the graph below.

4. RECURRENT MODELS FOR DERIVATIVES ESTIMATION

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.

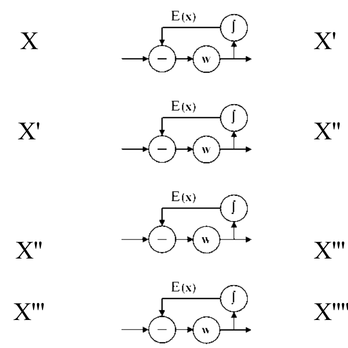


Figure 6: A 4th order recurrent network to abduct 1st to 4th time derivatives.

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

Figure 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.

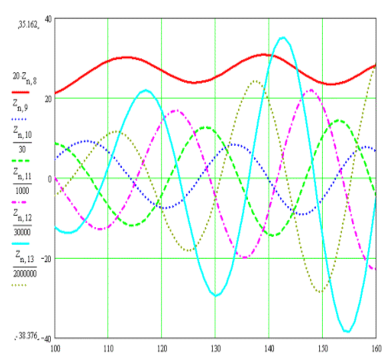


Figure 8: A typical set of a recurrent node trajectory (red) and 5 higher order time derivatives x' to x'''' abducted from a simulation of an intelligent system trajectory.

5. RESULTS AND DISCUSSION

5.1 Algorithm using Constant Parameters

This algorithm uses a single time point but two further time derivatives. The filter cascade is increased by one again to provide a continuous estimate of the 4th time derivative x''''. The separation problem disappears altogether now to provide a continuous estimate of all parameters at each point on the trajectory. Program 3 [Zreiba, 1999, Appendix A] was run, and the result of the estimation are given in Fig. 9; which shows fast and accurate convergence.

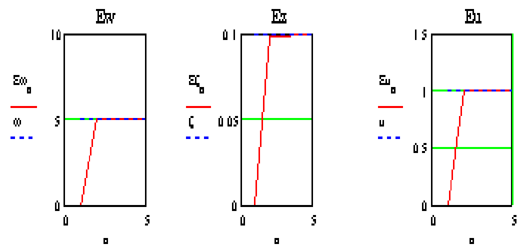


Figure 9: Estimated constant omega, zeta and u.

5.3 Results For The New Algorithm

Mathcad routines were set up to generate the input u as second order system with its own parameters of natural frequency, damping ratio and input. The input subsystem damping ratio was set to 0.05 to generate an oscillatory behaviour for long enough to test the parameter tracking algorithm thoroughly. The frequency of the input was set to 16 radians per second, one quarter of the frequency of the object natural frequency. The derivative generation cascade was increased by one to produce the fifth time derivative. The results are shown in Fig. 10 below.

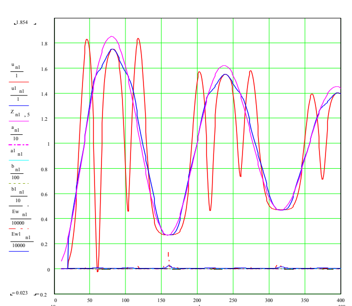


Figure 10: Results of the fourth algorithm for one second integration time.

The actual input is shown in pink, which gives approximately two and a half cycles over a period of a second as expected, i.e. 16 radians/s = 2.546 Hertz. The red trace shows the results from the previous constant u derivation algorithm which is failing completely to track the input parameter. The blue trace shows the result of the new algorithm, which is managing to track the input much more closely; however it start to diverge slightly near the peak of the cycle but then returns to track it well right down and round the lower trough of the input trajectory.

It is clear that tracking remain stable. It is interesting to note that the old algorithm while completely failing to track the upper half of the input trajectory it seems to track it well during the its lower half but not as well as the new algorithm.

Tracking Two Parameters: For a given range of parameters the algorithm worked well, being able to estimate the two input parameters u and omega with their time varying behaviour, i.e. track them while they are changing. The 1st order algorithm worked better than the constant one, Fig. 11 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. 12 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.

Figure 11: Estimated Omega assuming constant parameters (Ew, red tarce) and first order parameters (Ew1, blue trace) compared to actual shown in pink

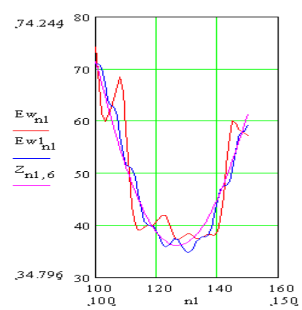
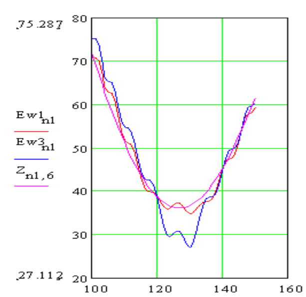


Figure 12: Deterioration of accuracy of estimated Omega with higher order algorithms: 3rd order estimate (Ew3 in blue) is worse than 1st order estimate (Ew1 in red), actual in pink.



6. CONCLUSIONS AND FUTURE WORK

Analytical models for parameter estimation were recast in terms of hybrid recurrent nets to track the parameters of complex systems in real time. To estimate the values of the parameters, the values of higher order time derivatives had to be estimated first. A cascade of 1st order recurrent networks were used to for this, 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. Future work will extend the technique to a further layer to track the parameters of a third level input model. 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. D. Al-Dabass, A. Zreiba, D. J. Evans, S. Sivayoganathan, "Parameter Estimation Algorithms for Hierarchical Distributed Systems", I. J. of Computer Mathematics, Vol. 79, No. 1, January 2002, pp65-88, ISSN 0020-7160.
2. D. Al-Dabass, D. Evans and S. Sivayoganathan, "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.
3. D. Al-Dabass, D. Evans and S. Sivayoganathan, "A Recurrent Network Architecture for Non-linear Parameter Tracking Algorithms", research report, January 2002, Dept of Computing & mathematics, Nottingham Trent University, Nottingham, NG1 4BU.
4. D. Al-Dabass, "Modelling the Complexity of Concept Dynamics", 47th Meeting Of The International Society For The Systems Sciences, 2nd – 6th August 2002, Shanghai International Convention Centre, Shanghai, China.
5. Richard Cant, Julian Churchill, David Al-Dabass, "Using Hard And Soft Artificial Intelligence Algorithms To Simulate Human Go Playing Techniques", Int. J. of Simulation, Vol. 2, No.1, June 2001, pp 31-49, ISSN 1473-804x Online, ISSN 1473-8031 Print.
6. Manling Ren, David Al-Dabass, "Simulation Of Fuzzy Possibilistic Algorithms For Recognising Chinese Characters", Int. J. of Simulation, Vol. 2, No.1, June 2001, pp 1-13, ISSN 1473-804x Online, ISSN 1473-8031 Print.
7. A Zreiba, MPhil, Simulation of Real Time Parameter Estimation Algorithms for Time varying Systems, March 2000.
8. D. Al-Dabass, A. Zreiba, D. Evans and K Sivayoganathan, "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 1999, pp170-176, http://ducati.doc.ntu.ac.uk/uksim/papers/moller/dad.doc, ISBN 0-905488-38-5.
9. D. Al-Dabass, A. Zreiba, D. Evans, K. Sivayoganathan., "Simulation of Noise Sensitivity of Parameter Estimation Algorithms, Simulation'99 Workshop, UCL, London, 29 October 1999, pp32-35.
10. Goodwin, C, "Real Time Recursive Block Parameter Estimation of Second Order Systems", PhD Thesis, Dept. of Computing, The Nottingham Trent University, Nottingham, 1997.
11. Kailath, T, "Lectures on Linear Least-Squares Estimation", CISM courses and lectures No. 140, Spring-Verlag, New York, 1978.
12. Gersch, W, "Least Squares Estimates of Structural System Parameters using Covariance Function Data", EEE Trans. On auto. Control, 19(6), 1974.
13. Man, Z, "Parameter-Estimation of Continuous Linear Systems using Functional Approximation", Computers and Electrical Eng. Vol. 21, No. 3, pp. 183-187 (1995).
14. Cawley, P, "The reduction of Bias Error in Transfer Function Estimates using FFT-based Analysers", Journal of Vibration, Acoustics, Stress and Reliability in Design, pp.29-35 (1984).
15. Dewolf, D, and D. Wiberg, "An Ordinary Differential-Equation Technique for Continuous Time Parameter Estimation", IEEE Trans. On Auto. Control, Vol. 38, No. 4, PP. 514-528 (1993).
16. Kalman, R, "A New Approach to Linear Filtering and Prediction Problems", Tans. Of SAME: Journal of Basic Eng., series D, 82, PP. 35-45 (1960).
17. Mathcad 7 Professional Program.
18. Zreiba, A, "MathCad Programs for Parameter Estimation", Research Report, Dept of Computing, The Nottingham Trent University, Nottingham, 1999.