Derivative Abduction using a Recurrent Network Architecture for Parameter Tracking Algorithms

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: To model the behaviour of complex natural and physical systems, the authors 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. In this paper a cascaded recurrent network architecture is proposed to 'abduct' these derivatives in successive stages. 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.

I. INTRODUCTION

Several explicit algorithms for the 3 usual parameters characterising the behaviour of second order models have been derived [1] based on information available from the systems time trajectory. Leaving the 2^{nd} order model in its 2^{nd} time derivative form and using 3 points on the trajectory, each providing position, velocity and acceleration, a set of 3 simultaneous algebraic equations were solved to yield estimates of input, natural frequency and damping ratio. An online dynamical algorithm was then configured to combine estimates of the trajectory time derivatives with these explicit static non-linear functions to provide continuous parameter estimation in real time.

For time varying parameters, the time separation between the 3 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had a direct influence on estimation accuracy, where the assumption of constant parameters used in the derivation is no longer valid, and accuracy deteriorates with increasing rate of parameter variation. To reduce the separation effect, a second algorithm was derived that relied on two points only but, to compensate for this reduction in data, it needed more information from each point in the form of a higher, 3^{rd}, time derivative. This resulted in better estimation of higher rate parameter variation despite the fact that higher derivatives are more sensitive to trajectory measurement noise and estimation errors.

Ultimately, the separation effect can only be eliminated if all the information needed for the estimation is derived from a single time point. This was successfully carried out by deriving a third algorithm, termed algorithm 3, which obtained the additional information from a higher, 4^{th}, time derivative. An online algorithm would then consist of two subsystems: i) a cascade of recurrent networks, and ii) static non-linear functions of these derivatives to produce continuous estimates of the 3 parameters. The first subsystem performs state estimation, as a set of first order observers, to generate continuous trajectories of all time derivatives up to 4^{th}, and simultaneously provides noise filtering. As all the information needed to estimate the parameters are obtained from a single time point on the trajectory, this algorithm 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. A set of 3 equations, one for each parameter, is formulated and numerically computed in real time together with the state estimation vector observer to yield continuous trajectories of the parameters. This is a different technique to that of augmenting the state derivative vector with the parameter derivatives,- instead of driving these derivatives with some function of the error between the system and model output, we provide an explicit function that should aid successful and speedy convergence to actual parameter values and provide continuous tracking. This should hold even when the parameters are changing rapidly compared to the system's natural frequency or time constant.

MathSoft Mathcad

Mathcad 8 Professional from MathSoft is the industry standard calculation software for technical professionals that will be used to test the algorithm and create the necessary graphical outputs. Mathcad can be used to display equations in a format similar to that seen on a blackboard or reference book and allows the user to solve equations symbolically or numerically.

II. PROBLEM FORMULATION

Definitions

Many physical problems are described by the solution of an initial value problem of the form:

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

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.

Algorithm For Constant Parameters From Single Point Data

This was termed algorithm 3 in reference 1.

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

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

Differentiate with respect to t:

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

divide by x":

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

and differentiate with respect to t again to give:

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

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

$$E\omega^2 = [x''.\, x'''' - x'''^2] / [x'.\, x''' - x''^2] \qquad (5)$$

$$E\zeta = -[E\omega^{-2}\, x''' + x'] / [2.\, E\omega^{-1}.x''] \qquad (6)$$

$$Eu = E\omega^{-2}.\, x'' + 2.\, E\zeta.\, E\omega^{-1}\, .\, x' + x \qquad (7)$$

III. HIGH ORDER ALGORITHMS

We assume that the first and higher 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', u", u'" and u"" to be non zero is extracted from the 5^{th}, 6^{th}, 7^{th} and 8^{th} time derivatives of the trajectory. Only the case for the u' is shown here, the others for u" etc are simple extensions of the idea and are left as an exercise for the reader.

$$a.x'' + b.x' + x = u \qquad (1)$$

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,

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

IV. SIMULATIONS AND PROGRAMS

Programs implemented in Mathcad were presented in reference 1 for the 3 categories of parameters: constant and variable with 1st and 2nd order dynamics, and are used to illustrate the generation of time derivatives. In the next section the 3 algorithms will be applied for each parameter category to test their effectiveness.

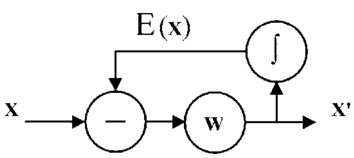

Figure 1: A single stage recurrent sub-net using an integrator in the feedback path to abduct the derivative x' = w(x-E(x))

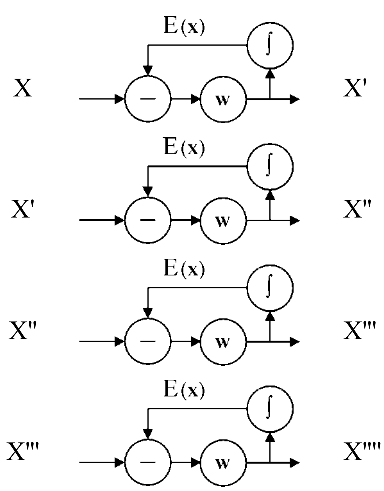

Figure 2: A 4th order recurrent network to abduct 1st to 4th time derivatives.

A Recurrent Architecture to Abduct Time Derivatives

The structure of each cell of the recurrent network is shown in Figure 1. The output of each cell feeds the input to the next one to generate the next higher order tiome 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 is shown in Figure 2.

$$x := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad D(t,x) := \begin{bmatrix} x_2 \\ -\omega^2 \cdot x_1 - 2\cdot\zeta\cdot\omega\cdot x_2 + \omega^2\cdot u \\ G\cdot(x_1 - x_3) \\ G\cdot\left[G\cdot(x_1 - x_3) - x_4\right] \\ G\cdot\left[G\cdot\left[G\cdot(x_1 - x_3) - x_4\right] - x_5\right] \\ G\cdot\left[G\cdot\left[G\cdot\left[G\cdot(x_1 - x_3) - x_4\right] - x_5\right] - x_6\right] \\ G\cdot\left[G\cdot\left[G\cdot\left[G\cdot\left[G\cdot(x_1 - x_3) - x_4\right] - x_5\right] - x_6\right] - x_7\right] \end{bmatrix}$$

$$Z := \mathrm{Rkadapt}(x, t0, t1, N, D)$$

Figure 3: A cascade of 5 recurrent cells plus the 2nd order trajectory model.

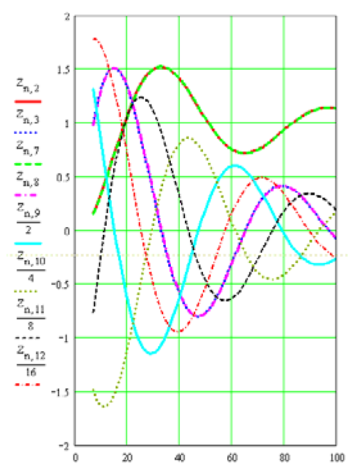

Figure 4: A typical set of time derivatives abducted from the trajectory of an oscillatory 2nd order dynamical system

V. RESULTS AND DISCUSSION

Algorithm 3 using Constant Parameters

This algorithm uses a single time point but two further time derivatives compared to Algorithm-1. 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 figure 5; which shows fast and accurate convergence.

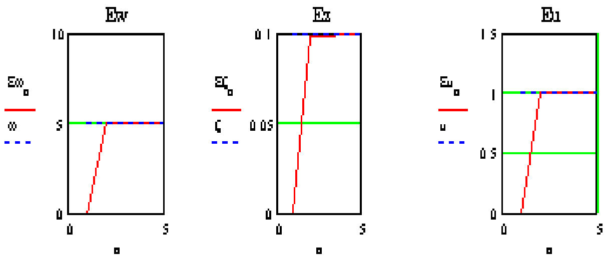

Figure 5: Estimated constant omega, zeta and u

Discussion

Estimated values for constants parameters were close to the desired set values. The derived algorithms estimated ω, ζ and u for a good range of values: ω from 1 to10, ζ between +/- (0.01 to 1), and u between +/- (0.5-40), and gave accurate estimates. Estimation errors decreased as ω increased, particularly for small ζ (less than 0.5): where oscillation provided wide variation in the variables to decrease errors. The differences between the (simulated) system time derivatives (x, x' and x'') and their estimates from the filter cascade depended on G (the cut-off frequency): high G provided more accurate estimation of derivatives but made the algorithms prone to noise and vice versa. Another disadvantage of high G from the simulation point of view is that simulation time increased considerably due to the integration routine adapting to ever smaller steps. The algorithms provided progressively faster convergence with Algorithm-3 being the fastest to converge.

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 Figure 6 below.

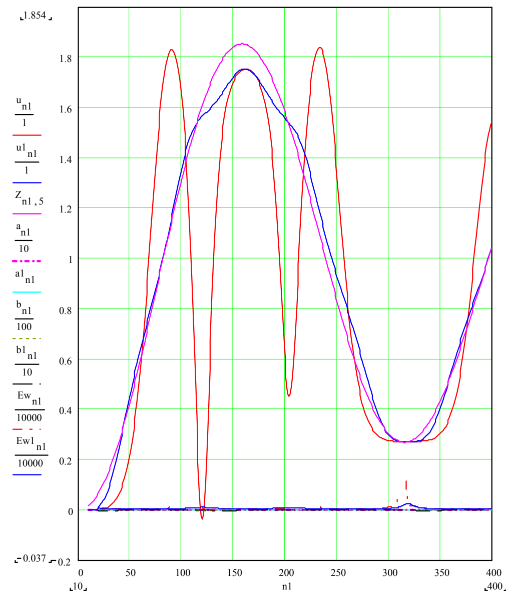

Figure 6: Results of the high order algorithm

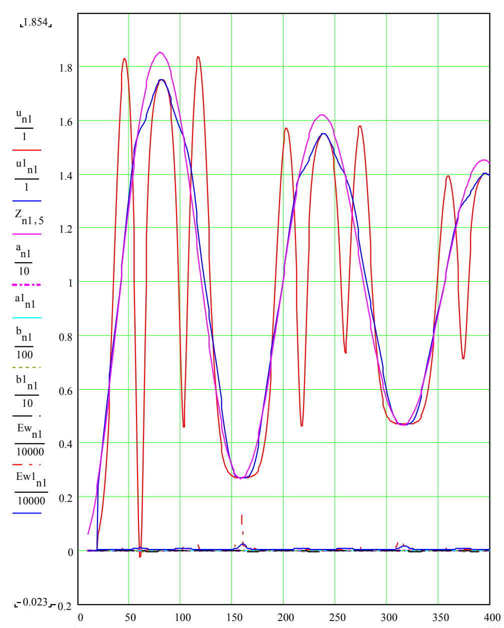

Figure 7: Results of the high order algorithm for one second integration time.

The actual input is shown in pink, which gives approximately one and one quarter cycles over a period of half 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.

To check the quality of tracking as time progresses, a second set of results was obtained with integration time extended to one second to give two and a half cycles. The results are shown in Figure 7. 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.

VI. CONCLUSIONS AND FUTURE WORK

A recurrent architecture to abduct the high order derivatives from the time trajectory of a system was proposed. These time derivatives were then used to derive and test a new high order algorithm to track the input parameter for a reduced order model. The test involved the generation of the output trajectory of a lightly damped second order system. The results showed the algorithm maintaining good tracking over an extended period of time. The new algorithm was far superior to a previous one, which relied on the assumption of constant input in the derivation.

The technique needs to be extended in two directions: a) to estimate continuously changing values of the other two parameters of natural frequency omega and damping ratio zeta, and b) explore even better estimation algorithms by assuming non zero second time derivatives of the parameters.

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. Between 1976 and 1982 he held post-doctoral and advanced research fellowships at the Control Systems Centre, University of Manchester Institute of Science & Technology (UMIST). He joined The Nottingham Trent University in 1983 as a Principal Lecturer in the Department of Computing and mathematics. For more details see his website:
http://ducati.doc.ntu.ac.uk/uksim/dad/webpage.htm

REFERENCES

[1] 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.

[2] 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.

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

[4] Kailath, T, "Lectures on Linear Least-Squares Estimation", CISM courses and lectures No. 140, Spring-Verlag, New York, 1978.

[5] Gersch, W, "Least Squares Estimates of Structural System Parameters using Covariance Function Data", EEE Trans. On auto. Control, 19(6), 1974.

[6] Man, Z, "Parameter-Estimation of Continuous Linear Systems using Functional Approximation", Computers and Electrical Eng. Vol. 21, No. 3, pp. 183-187 (1995).

[7] 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).

[8] 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).

[9] 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).

[10] Mathcad 7 Professional Program.

[11] Zreiba, A, "MathCad Programs for Parameter Estimation", Research Report, Dept of Computing, The Nottingham Trent University, Nottingham, 1999.