

# **ANALYTICAL MODELS FOR PARAMETER TRACKING ALGORITHMS USING HYBRID RECURRENT NETWORKS**

DAVID AL-DABASS, DAVID EVANS and  
SIVA SIVAYOGANATHAN

*Faculty of Computing and Technology  
Nottingham Trent University  
Nottingham NG1 4B  
[david.al-dabass@ntu.ac.uk](mailto:david.al-dabass@ntu.ac.uk)*

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.

# **OUTLINE OF THE PRESENTATION**

## **1. INTRODUCTION**

## **2. ANALYTICAL MODELS AND ALGORITHMS**

### **2.1 Models Assuming Constant Parameters From Single Point Data**

### **2.2 Models for Time Varying Parameters**

## **3. HYBRID RECURRENT NETWORK MODELS**

### **3.1 Models of Hierarchical Recurrent Nodes**

## **4. RECURRENT MODELS FOR DERIVATIVES ESTIMATION**

## **5. RESULTS AND DISCUSSION**

### **5.1 Algorithm using Constant Parameters**

### **5.2 Results For The New Algorithm**

## **6. CONCLUSIONS AND FUTURE WORK**

## 1. INTRODUCTION

- Online dynamical algorithms 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had a direct influence on estimation accuracy, where the assumption of constant parameters used in the derivation is no longer valid, and accuracy deteriorates with increasing rate of parameter variation.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:

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

Where  $\omega$  is the natural frequency,  $\zeta$  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<sup>st</sup> to 4<sup>th</sup> time derivatives at a single point. Given the second order system:

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

Differentiate with respect to  $t$ :

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

Divide by  $x''$ :

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

Differentiate with respect to t again to give:

$$\omega^{-2} \cdot [(x'' \cdot x'''' - x''''^2) / x''^2] + 0 + [(x''^2 - x' \cdot x''') / x''^2] = 0 \quad (4)$$

We get expressions for estimated  $\omega$ , estimated  $\zeta$ , using (2), and estimated u:

$$E\omega^2 = [x'' \cdot x'''' - x''''^2] / [x' \cdot x''' - x''^2] \quad (5)$$

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

$$Eu = E\omega^{-2} \cdot x'' + 2 \cdot E\zeta \cdot E\omega^{-1} \cdot x' + x \quad (7)$$

## 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 5<sup>th</sup> time derivative of the trajectory.

$$a \cdot x'' + b \cdot x' + x = u \quad (1-a)$$

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

$$a \cdot x''' + b \cdot x'' + x' = u' \quad (8)$$

Differentiate again and set  $u'' = 0$  gives:

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

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

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

Differentiate again to eliminate  $b$ :

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

Re-arranging for  $a$  gives:

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

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:

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

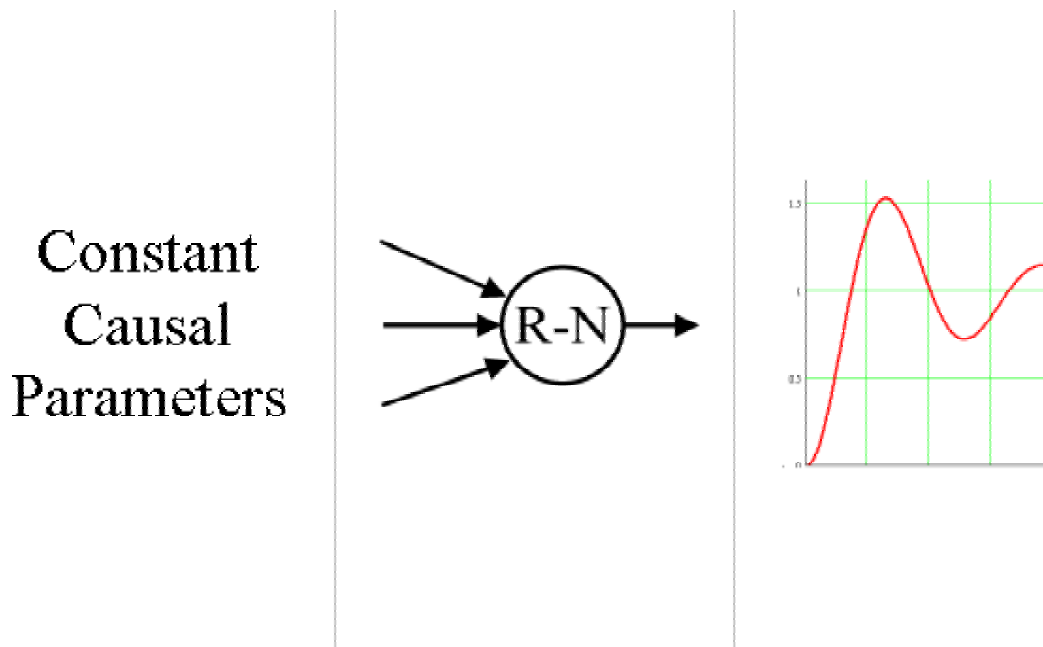
We can now substitute these values for  $a$  and  $b$  into Equation 1 to solve for  $u$ ,

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

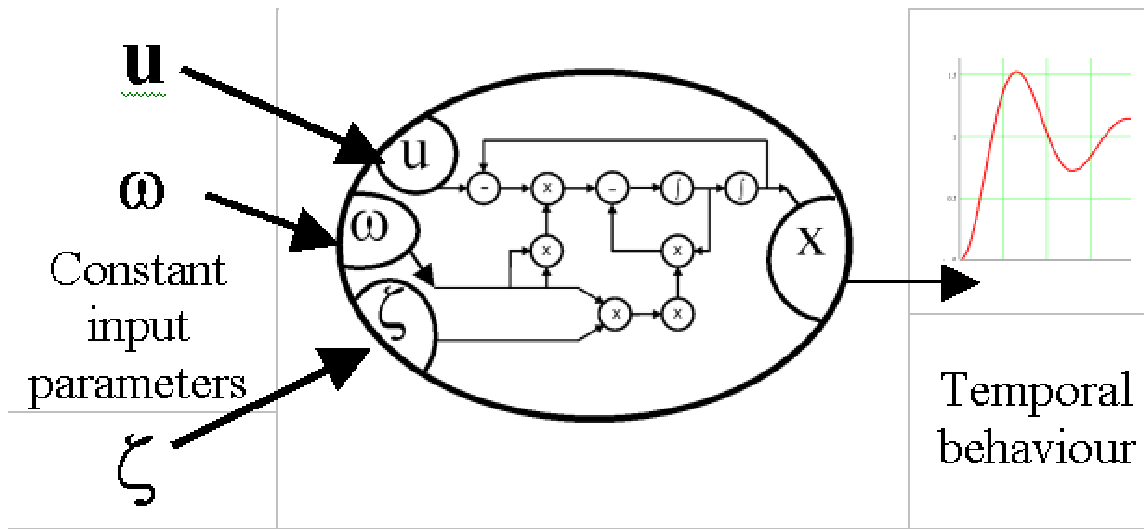
### 3. HYBRID RECURRENT NETWORK MODELS

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.



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



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

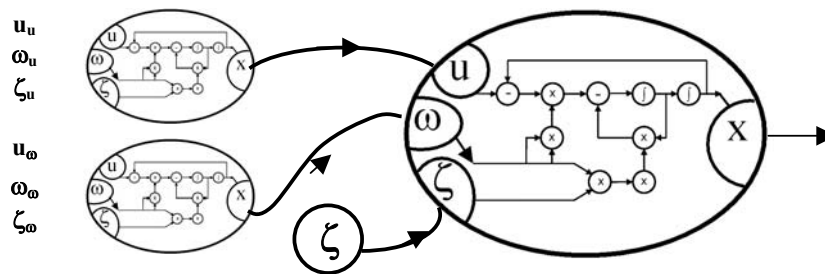
To model this oscillatory behaviour we propose a second order integral hybrid model shown in Fig. 2. This model is based on the well known second order dynamical system which has the following form:

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

Where  $x$  is the output of the node and  $\omega$ ,  $\zeta$  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 2<sup>nd</sup> order dynamical characteristics, as shown in Fig. 3.



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

The 2<sup>nd</sup> 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 2<sup>nd</sup> order

dynamics. The input  $\mathbf{u}$  is the output of the following 2<sup>nd</sup> order system:

$$\omega_u^{-2} \mathbf{u}'' + 2 \cdot \zeta_u \cdot \omega_u^{-1} \cdot \mathbf{u}' + \mathbf{u} = \mathbf{u}_u$$

The natural frequency  $\omega$  is the output of the following 2<sup>nd</sup> order system:

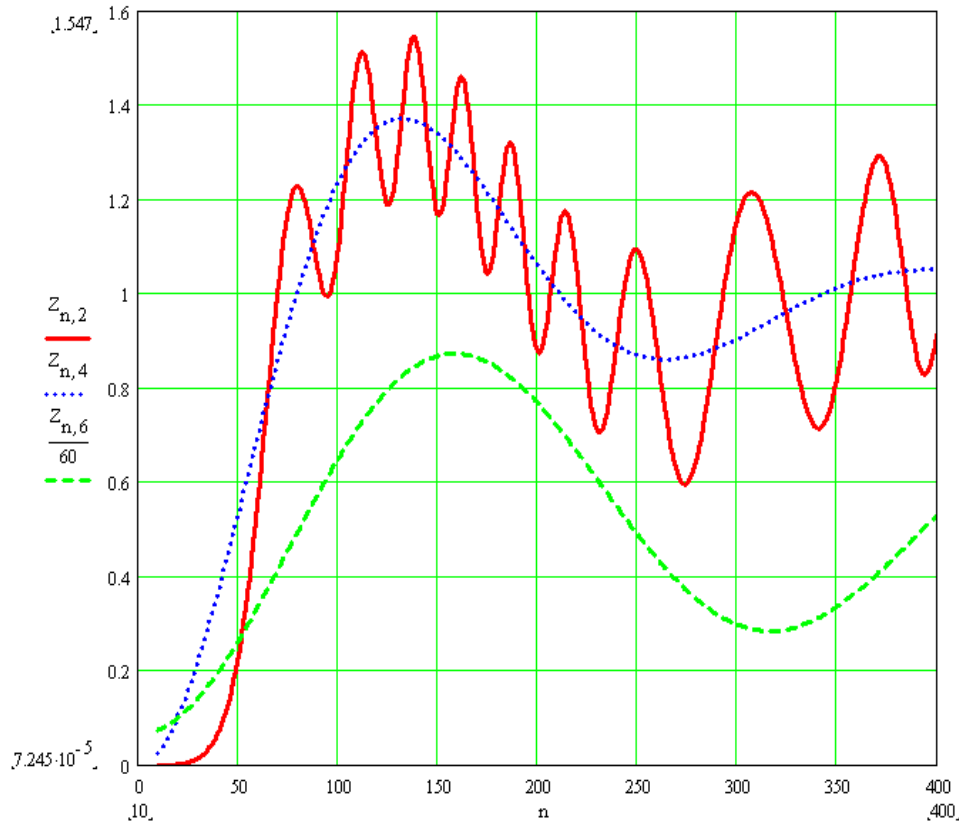
$$\omega_\omega^{-2} \omega'' + 2 \cdot \zeta_\omega \cdot \omega_\omega^{-1} \cdot \omega' + \omega = \mathbf{u}_\omega$$

The behaviour trajectory is generated by the 6<sup>th</sup> order vector differential equation (using Runge Kutta in Mathcad for this example), see Fig. 4

$$D(t, \mathbf{x}) := \begin{bmatrix} x_2 \\ x_5 \cdot x_5 \cdot x_3 - 2 \cdot z x_5 \cdot x_2 - x_5 \cdot x_5 \cdot x_1 \\ x_4 \\ (wuwuu) - 2 \cdot zuwux_4 - wuwux_3 \\ x_6 \\ (wwwuw) - 2 \cdot zwwwx_6 - wwwwx_5 \end{bmatrix}$$

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

Where  $x_1$  and  $x_2$  represent the  $x$  and  $x'$ ,  $x_3$  and  $x_4$  represent  $u$  and  $u'$ , and  $x_5$  and  $x_6$  represent  $\omega$  and  $\omega'$  respectively.

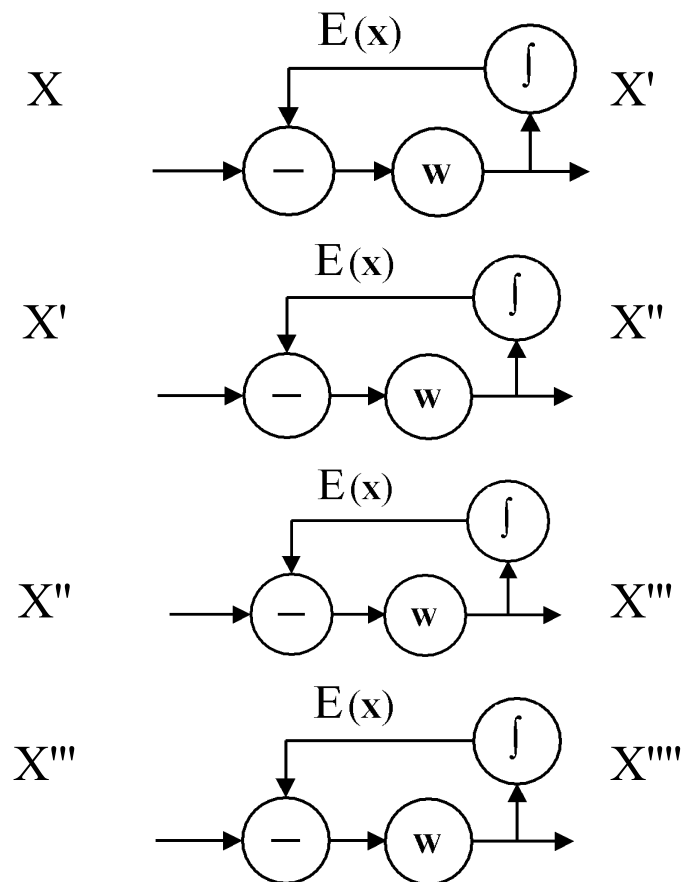


**Fig. 5.** Simulated trajectory of a hierarchical recurrent node (red trace), with 2 variable inputs:  $u$  (blue) and  $\omega$  (green)

- for the  $u$  subsystem,  $u$  started from 0 aiming at  $u=1$  at a rate of  $\omega_u = 5$  rad/s with  $\zeta_u = 0.3$ .
- For the  $\omega$  subsystem,  $\omega$  started from 4 rad/s aiming at  $u_\omega = 32$  rad/s at a rate of  $\omega_\omega = 4$  rad/s with  $\zeta_\omega = 0.1$ .

## 4. RECURRENT MODELS FOR DERIVATIVES ESTIMATION

- To estimate the values of the derivatives a cascade of 1<sup>st</sup> 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 1<sup>st</sup> order recurrent network filters were simulated using the 4<sup>th</sup> order Runge-Kutta method in Mathcad.



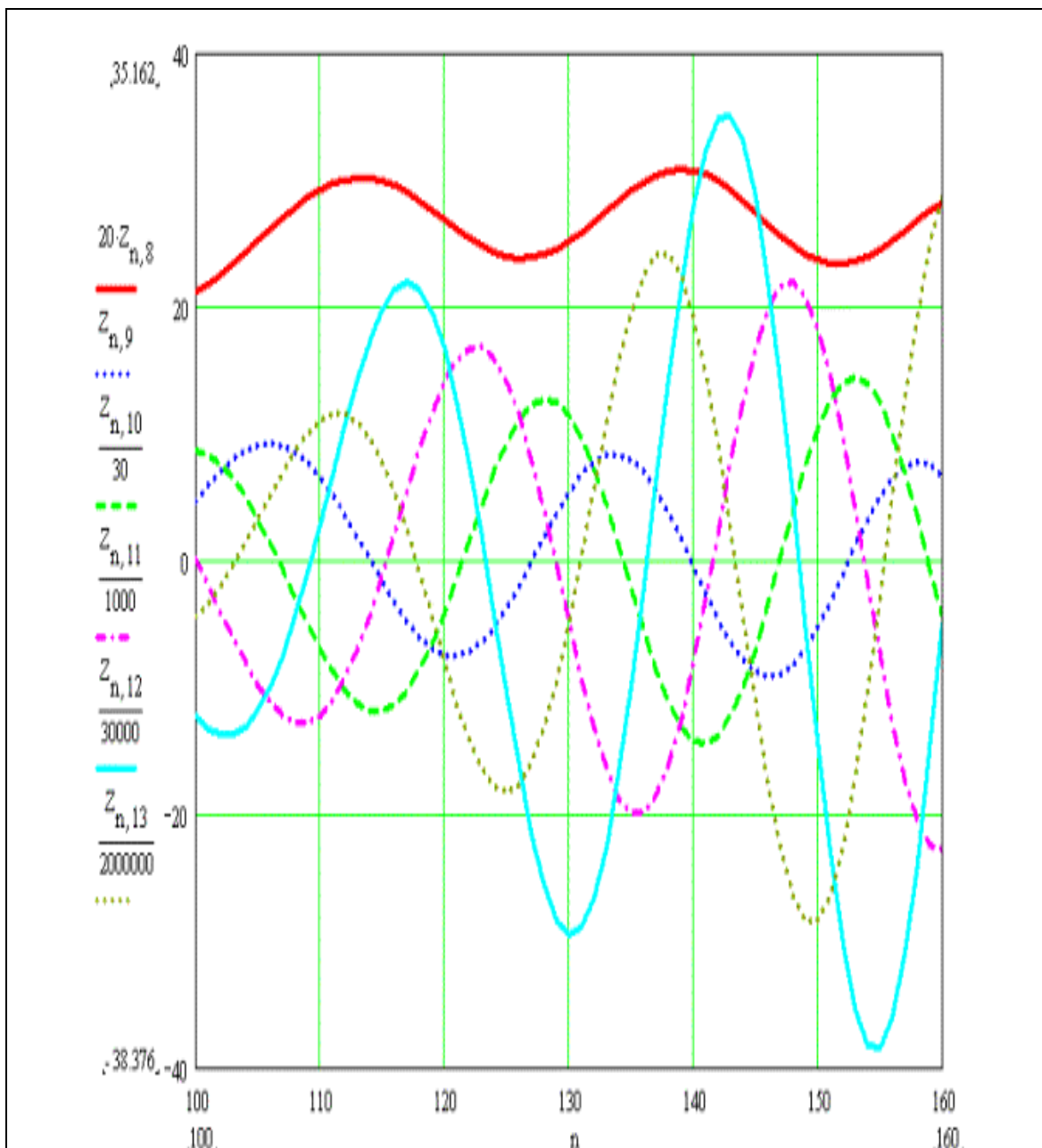
**Fig. 6.** A 4<sup>th</sup> order recurrent network to abduct 1<sup>st</sup> to 4<sup>th</sup> time derivatives.

- The derivatives vector for producing derivatives up to fifth is shown in Fig. 7

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

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

Figure 8 shows a typical set of high order time derivatives abducted from the output trajectory of a system displaying temporal behaviour.

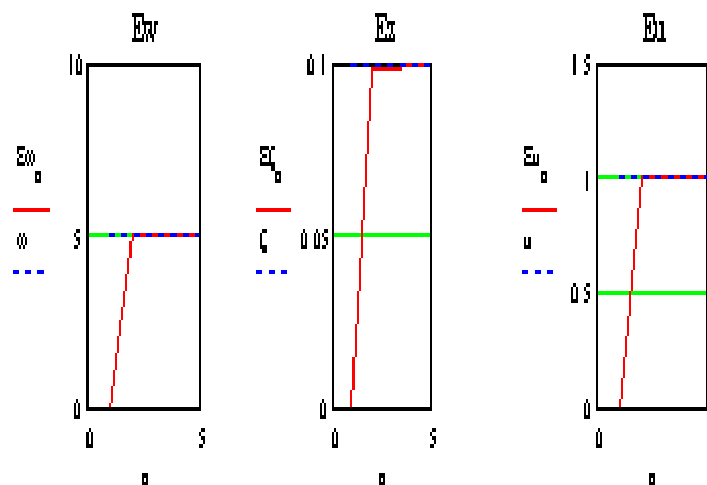


**Fig. 8.** A typical set of a recurrent node trajectory (red) and 5 higher order time derivatives  $x'$  to  $x''''$  abducted from a simulation of an intelligent system trajectory

## 5. RESULTS AND DISCUSSION

### 5.1 Algorithm using Constant Parameters

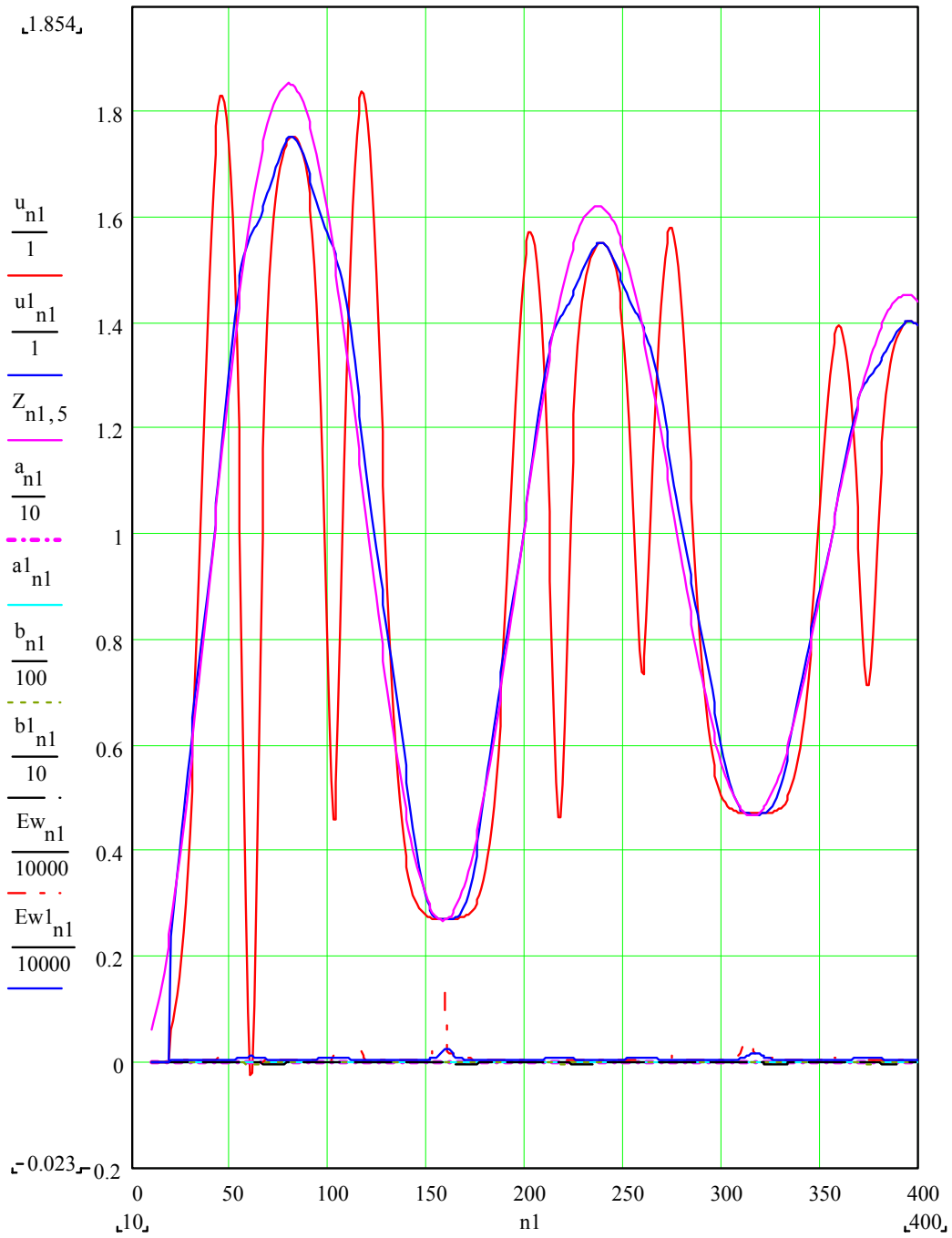
- This algorithm uses a single time point but two further time derivatives.
- The filter cascade is increased by one again to provide a continuous estimate of the 4<sup>th</sup> time derivative  $x''''$ .
- The separation problem disappears altogether now to provide a continuous estimate of all parameters at each point on the trajectory.
- The result of the estimation are given in Fig. 9; which shows fast and accurate convergence.



**Fig. 9.** Estimated constant omega, zeta and u.

## 5.2 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.



**Fig. 10.** Results of the fourth algorithm for one second integration time.

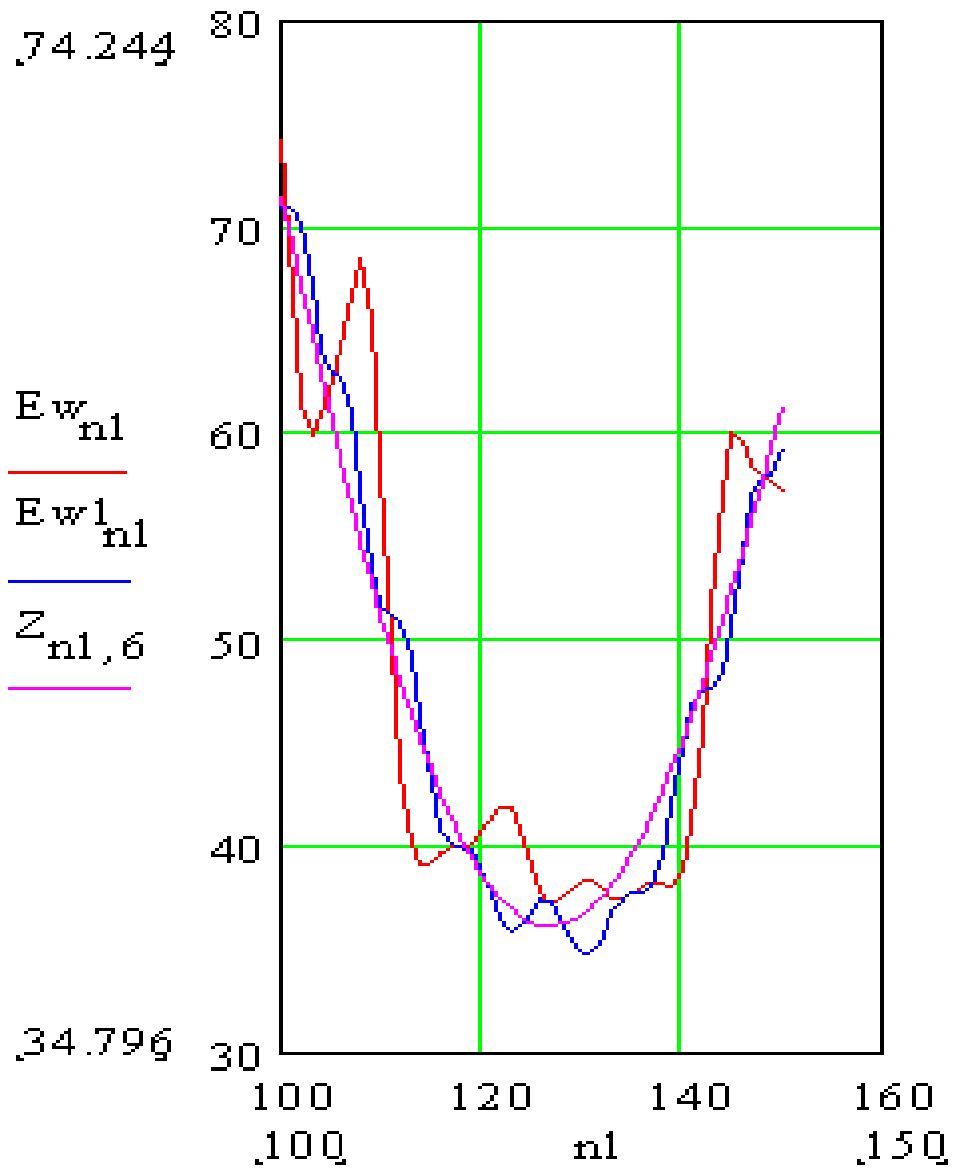
### Figure 10

- 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.  $16 \text{ radians/s} = 2.546 \text{ Hertz}$ .
- 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.

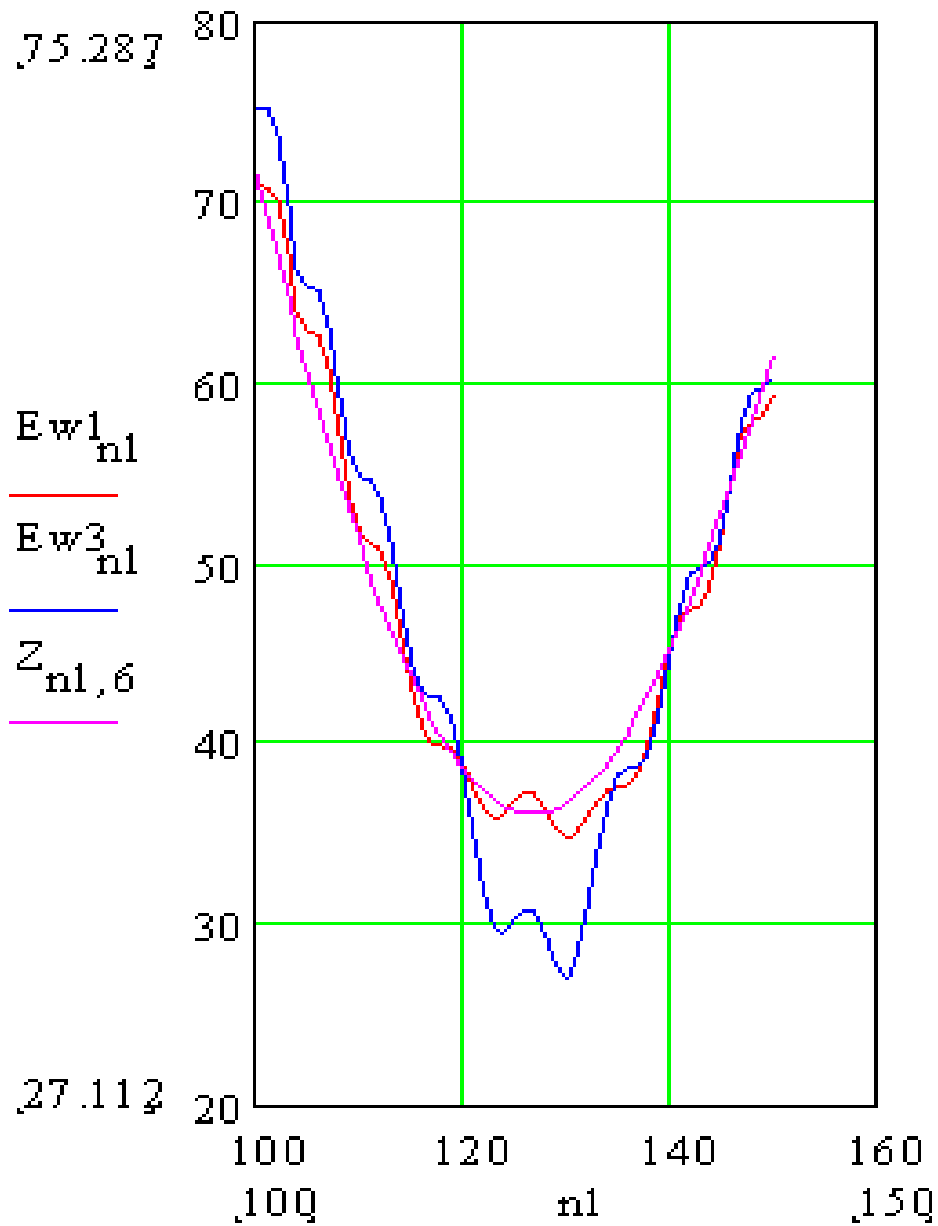
## Tracking Two Parameters:

- 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 1<sup>st</sup> order algorithm worked better than the constant one, Fig. 11 shows a comparison of the two algorithms tracking  $\omega$ .



**Fig. 11.** Estimated Omega assuming constant parameters ( $E_w$ , red trace) and first order parameters ( $E_{w1}$ , blue trace) compared to actual shown in pink

- Algorithms of higher order than 1<sup>st</sup> showed marginal improvement but in certain cases showed a deteriorating behaviour.
- Fig. 12 shows a 3<sup>rd</sup> order algorithm deviating quite markedly from the true trajectory compared to a 1<sup>st</sup> order algorithm.
- This is likely to be due to an accumulation of errors in higher derivative values used in the former algorithm.



**Fig. 12.** Deterioration of accuracy of estimated Omega with higher order algorithms: 3<sup>rd</sup> order estimate (Ew3 in blue) is worse than 1<sup>st</sup> 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 1<sup>st</sup> 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 1<sup>st</sup> 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.

<http://ducati.doc.ntu.ac.uk/uksim/dad/webpage.htm>